RKHSMetaMod: An R Package to Estimate the Hoeffding Decomposition of a Complex Model by Solving RKHS Ridge Group Sparse Optimization Problem

by Halaleh Kamari, Sylvie Huet and Marie-Luce Taupin

Abstract In this paper, we propose an R package, called **RKHSMetaMod**, that implements a procedure for estimating a meta-model of a complex model. The meta-model approximates the Hoeffding decomposition of the complex model and allows us to perform sensitivity analysis on it. It belongs to a reproducing kernel Hilbert space that is constructed as a direct sum of Hilbert spaces. The estimator of the meta-model is the solution of a penalized empirical least-squares minimization with the sum of the Hilbert norm and the empirical L^2-norm. This procedure, called RKHS ridge group sparse, allows both to select and estimate the terms in the Hoeffding decomposition, and therefore, to select and estimate the Sobol indices that are non-zero. The **RKHSMetaMod** package provides an interface from the R statistical computing environment to the C++ libraries **Eigen** and **GSL**. In order to speed up the execution time and optimize the storage memory, except for a function that is written in R, all of the functions of this package are written using the efficient C++ libraries through **RcppEigen** and **RcppGSL** packages. These functions are then interfaced in the R environment in order to propose a user-friendly package.

1 Introduction

Consider a phenomenon described by a model m depending on d input variables $X = (X_1, ..., X_d)$. This model m from $\mathbb{R}^d$ to $\mathbb{R}$ may be a known model that is calculable in all points of X, i.e. $Y = m(X)$, or it may be an unknown regression model defined as follows:

$$Y = m(X) + \varepsilon, \tag{1}$$

where the error ε is assumed to be centered with a finite variance, i.e. $E(\varepsilon) = 0$ and $\text{var}(\varepsilon) < \infty$. The components of X are independent with a known law $P_X = \prod_{a=1}^{d} P_{X_a}$ on $\mathcal{X}$, a subset of $\mathbb{R}^d$. The number d of components of X may be large. The model m may present high complexity as strong non-linearities and high order interaction effects, and it is assumed to be square-integrable, i.e. $m \in L^2(\mathcal{X}, P_X)$. Based on the data points $\{(X_i, Y_i)\}_{i=1}^{n}$, we estimate a meta-model that approximates the Hoeffding decomposition of m. This meta-model belongs to a reproducing kernel Hilbert space (RKHS), which is constructed as a direct sum of the Hilbert spaces leading to an additive decomposition including variables and interactions between them (Durrande et al., 2013). The estimator of the meta-model is calculated by minimizing an empirical least-squares criterion penalized by the sum of two penalty terms: the Hilbert norm and the empirical norm (Huet and Taupin, 2017). This procedure allows us to select the subsets of variables $X_1, ..., X_d$ that contribute to predict Y. The estimated meta-model is used to perform sensitivity analysis, and therefore, to determine the influence of each variable and groups of them on the output variable Y.

In the classical framework of the sensitivity analysis, the function m is calculable in all points of X, and one may use the method of Sobol (1993) to perform the sensitivity analysis on m. Let us briefly introduce this method:
The independency between the components of X leads to write the function m according to its Hoeffding decomposition (Sobol, 1993; Van der Vaart, 1998):

$$m(X) = m_0 + \sum_{a=1}^{d} m_a(X_a) + \sum_{a<a'} m_{a,a'}(X_a, X_{a'}) + ... + m_{1,...,d}(X). \tag{2}$$

The terms in this decomposition are defined using the conditional expected values:

$$m_0 = E_X(m(X)),\ m_a(X_a) = E_X(m(X)|X_a) - m_0;$$
$$m_{a,a'}(X_a, X_{a'}) = E_X(m(X)|X_a, X_{a'}) - m_a(X_a) - m_{a'}(X_{a'}) - m_0, \cdots$$

These terms are known as the constant term, main effects, interactions of order two and so on. Let $\mathcal{P}$ be the set of all subsets of $\{1, ..., d\}$ with dimension 1 to d. For all $v \in \mathcal{P}$ and $X \in \mathcal{X}$, let X_v be the vector

with components X_a, $a \in v$. For a set A let $|A|$ be its cardinality, and for all $v \in \mathcal{P}$, let $m_v : \mathbb{R}^{|v|} \to \mathbb{R}$ be the function associated with X_v in Equation (2). Then Equation (2) can be expressed as follows:

$$m(X) = m_0 + \sum_{v \in \mathcal{P}} m_v(X_v). \tag{3}$$

This decomposition is unique, all terms m_v, $v \in \mathcal{P}$ are centered, and they are orthogonal with respect to $L^2(\mathcal{X}, P_X)$. The functions m and m_v, $v \in \mathcal{P}$ in Equation (3) are square-integrable. As any two terms of decomposition (3) are orthogonal, by squaring (3) and integrating it with respect to the distribution of X, a decomposition of the variance of $m(X)$ is obtained as follows:

$$\text{var}(m(X)) = \sum_{v \in \mathcal{P}} \text{var}(m_v(X_v)). \tag{4}$$

The Sobol indices associated with the group of variables X_v, $v \in \mathcal{P}$ are defined by:

$$S_v = \text{var}(m_v(X_v))/\text{var}(m(X)). \tag{5}$$

For each v, the S_v expresses the fraction of the variance of $m(X)$ explained by X_v. For all $v \in \mathcal{P}$, when $|v| = 1$, the S_vs are referred to as the first order indices; when $|v| = 2$, i.e. $v = \{a, a'\}$ and $a \neq a'$, they are referred to as the second order indices or the interaction indices of order two (between X_a and $X_{a'}$); and the same holds for $|v| > 2$.

The total number of the Sobol indices to be calculated is equal to $|\mathcal{P}| = 2^d - 1$, which raises exponentially with the number of the input variables d. When d is large, the evaluation of all the indices can be computationally demanding and even not reachable. In practice, only the indices of order not higher than two are calculated. However, only the first and second order indices may not provide a good information on the model sensitivities. In order to provide better information on the model sensitivities, Homma and Saltelli (1996) proposed to calculate the first order and the total indices defined as follows:
Let $\mathcal{P}_a \subset \mathcal{P}$ be the set of all the subsets of $\{1, ..., d\}$ including a, then $S_{T_a} = \sum_{v \in \mathcal{P}_a} S_v$. For all $a \in \{1, ..., d\}$, the S_{T_a} denotes the total effect of X_a. It expresses the fraction of variance of $m(X)$ explained by X_a alone and all the interactions of it with the other variables. The total indices allow us to rank the input variables with respect to the amount of their effect on the output variable. However, they do not provide complete information on the model sensitivities as do all the Sobol indices.

The classical computation of the Sobol indices is based on the Monte Carlo methods (see for example: Sobol (1993) for the main effect and interaction indices, and Saltelli (2002) for the main effect and total indices). For models that are expensive to evaluate, the Monte Carlo methods lead to a high computational burden. Moreover, in the case where d is large, m is complex and the calculation of the variances (see Equation (4)) is numerically complicated or not possible (as in the case where the model m is unknown) the methods described above are not applicable. Another approach consists in approximating m by a simplified model, called a meta-model, which is much faster to evaluate and to perform sensitivity analysis on it. Beside the approximations of the Sobol indices of m at a lower computational cost, a meta-model provides a deeper view of the input variables effects on the model output. Among the meta-modelling methods proposed in the literature, the expansion based on the polynomial Chaos (Wiener, 1938; Schoutens, 2000) can be used to approximate the Hoeffding decomposition of m (Sudret, 2008). The principle of the polynomial Chaos is to project m onto a basis of orthonormal polynomials. The polynomial Chaos expansion of m is written as (Soize and Ghanem, 2004):

$$m(X) = \sum_{j=0}^{\infty} h_j \phi_j(X), \tag{6}$$

where $\{h_j\}_{j=0}^{\infty}$ are the coefficients, and $\{\phi_j\}_{j=0}^{\infty}$ are the multivariate orthonormal polynomials associated with X which are determined according to the distribution of the components of X. In practice, expansion (6) shall be truncated for computational purposes, and the model m may be approximated by $\sum_{j=0}^{v_{max}} h_j \phi_j(X)$, where v_{max} is determined using a *truncation scheme*. The Sobol indices are obtained then by summing up the squares of the suitable coefficients. Blatman and Sudret (2011) proposed a method for truncating the polynomial Chaos expansion and an algorithm based on the least angle regression for selecting the terms in the expansion. In this approach, according to the distribution of the components of X, a unique family of orthonormal polynomials $\{\phi_j\}_{j=0}^{\infty}$ is determined. However, this family may not be necessarily the best functional basis to approximate m well.

Gaussian Process (GP) can also be used to construct meta-models as highlighted in Welch et al. (1992), Oakley and O'Hagan (2004), Kleijnen (2007, 2009), Marrel et al. (2009), Durrande et al. (2012), and Le Gratiet et al. (2014). The principle is to consider that the prior knowledge about the function $m(X)$, can be modelled by a GP $\mathcal{Z}(X)$ with a mean $m_{\mathcal{Z}}(X)$ and a covariance kernel $k_{\mathcal{Z}}(X, X')$. To

perform sensitivity analysis from a GP model, one may replace the model $m(X)$ with the mean of the conditional GP and deduce the Sobol indices from it. A review on the meta-modelling based on the polynomial Chaos and the GP is presented in Le Gratiet et al. (2017).

Durrande et al. (2013) considered a class of the functional approximation methods similar to the GP and obtained a meta-model that satisfies the properties of the Hoeffding decomposition. They proposed to approximate m by functions belonging to a RKHS $\mathcal{H}$ which is a direct sum of the Hilbert spaces. Their RKHS $\mathcal{H}$ is constructed in a way that the projection of m onto $\mathcal{H}$, denoted f^*, is an approximation of the Hoeffding decomposition of m. The function f^* is defined as the minimizer over the functions $f \in \mathcal{H}$ of the criterion $E_X(m(X) - f(X))^2$.

Let $\langle .,. \rangle \mathcal{H}$ be the scalar product in $\mathcal{H}$, let also k and k_v be the reproducing kernels associated with the RKHS $\mathcal{H}$ and the RKHS $\mathcal{H}_v$ respectively. The properties of the RKHS $\mathcal{H}$ insures that any function $f \in \mathcal{H}$, $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ is written as the following decomposition:

$$f(X) = \langle f, k(X,.) \rangle \mathcal{H} = f_0 + \sum_{v \in \mathcal{P}} f_v(X_v), \tag{7}$$

where f_0 is constant, and $f_v : \mathbb{R}^{|v|} \to \mathbb{R}$ is defined by $f_v(X) = \langle f, k_v(X,.) \rangle_{\mathcal{H}}$. For more details on the RKHS construction and the definition of the Hilbert norm see Section "RKHS construction" in the Appendix (supplementary materials).

For all $v \in \mathcal{P}$, the functions $f_v(X_v)$ are centered and for all $v \neq v'$, the functions $f_v(X_v)$ and $f_{v'}(X_{v'})$ are orthogonal with respect to $L^2(\mathcal{X}, P_X)$. Therefore, the decomposition of the function f presented in Equation (7) is its Hoeffding decomposition. As the function f^* belongs to the RKHS $\mathcal{H}$, it is decomposed as its Hoeffding decomposition, $f^* = f_0^* + \sum_{v \in \mathcal{P}} f_v^*$, and each function f_v^* approximates the function m_v in Equation (3). The number of the terms f_v^* that should be estimated in the Hoeffding decomposition of f^* is equal to $|\mathcal{P}| = 2^d - 1$, which may be huge since it rises very quickly by increasing d. In order to deal with this problem, in the regression framework, one may estimate f^* by a sparse meta-model $\widehat{f} \in \mathcal{H}$. To this end, the estimation of f^* is done on the basis of n observations by minimizing a least-squares criterion suitably penalized in order to deal with both the non-parametric nature of the problem and the possibly large number of functions that have to be estimated. In the classical framework of the sensitivity analysis one may calculate a sparse approximation of f^* using least-squares penalized criterion as it is done in the non-parametric regression framework. In order to obtain a sparse solution of a minimization problem, the penalty function should enforce the sparsity. There exists various ways of enforcing sparsity for a minimization (maximization) problem, see for example Hastie et al. (2015) for a review. Some methods, such as the Sparse Additive Models (SpAM) procedure (Ravikumar et al., 2009; Liu et al., 2009) are based on a combination of the l_1-norm with the empirical L^2-norm: $\|f\|_{n,1} = \sum_{a=1}^d \|f_a\|_n$, where $\|f_a\|_n^2 = \sum_{i=1}^n f_a^2(X_{ai})/n$, is the squared empirical L^2-norm of the univariate function f_a. The Component Selection and Smoothing Operator (COSSO) method developed by Lin and Zhang (2006) enforces sparsity using a combination of the l_1-norm with the Hilbert norm: $\|f\|_{\mathcal{H},1} = \sum_{a=1}^d \|f_a\|_{\mathcal{H}_a}$. Instead of focusing on only one penalty term, one may consider a more general family of estimators, called the *doubly penalized estimator*, which is obtained by minimizing a criterion penalized by the sum of two penalty terms. Raskutti et al. (2009, 2012) proposed a *doubly penalized estimator*, which is the solution of the minimization of a least-squares criterion penalized by the sum of a sparsity penalty term and a combination of the l_1-norm with the Hilbert norm:

$$\gamma \|f\|_{n,1} + \mu \|f\|_{\mathcal{H},1}, \tag{8}$$

where $\gamma, \mu \in \mathbb{R}$ are the tuning parameters that should be suitably chosen.

Meier et al. (2009) proposed a related family of estimators, based on the penalization with the empirical L^2-norm. Their penalty function is the sum of the sparsity penalty term, $\|f\|_{n,1}$, and a smoothness penalty term. Huet and Taupin (2017) considered the same approximation functional spaces as Durrande et al. (2013), and obtained a *doubly penalized estimator* of a meta-model which approximates the Hoeffding decomposition of m. Their estimator is the solution of the least-squares minimization penalized by the penalty function defined in Equation (8) adapted to the multivariate setting,

$$\gamma \|f\|_n + \mu \|f\|_{\mathcal{H}}, \text{ with } \|f\|_n = \sum_{v \in \mathcal{P}} \|f_v\|_n, \ \|f\|_{\mathcal{H}} = \sum_{v \in \mathcal{P}} \|f_v\|_{\mathcal{H}_v}. \tag{9}$$

This procedure, called RKHS ridge group sparse, estimates the groups v that are suitable for predicting f^*, and the relationship between f_v^* and X_v for each group. The obtained estimator, called RKHS meta-model, is used then to estimate the Sobol indices of m. This approach renders it possible to estimate the Sobol indices for all groups in the support of the RKHS meta-model, including the interactions of possibly high order, a point known to be difficult in practice.

In this paper, we introduce an R package, called **RKHSMetaMod**, that implements the RKHS ridge group sparse procedure. The functions of this package allows us to:

(1) calculate the reproducing kernels and their associated Gram matrices (see Section Calculation of the Gram matrices),

(2) implement the RKHS ridge group sparse procedure and a special case of it, called the RKHS group lasso procedure (when $\gamma = 0$ in the penalty function (9)), in order to estimate the terms f_v^* in the Hoeffding decomposition of the meta-model f^* leading to an estimation of the function m (see Section Optimization algorithms),

(3) choose the tuning parameters μ and γ (see Equation (9)), using a procedure that leads to obtain the *best* RKHS meta-model in terms of the prediction quality,

(4) estimate the Sobol indices of the function m (see Section Estimation of the Sobol indices).

The current version of the package supports uniformly distributed input variables on $\mathcal{X} = [0,1]^d$. However, it could be easily adapted to datasets with input variables from another distribution by making a small modification to one of its functions (see Remark 3 of Section Calculation of the Gram matrices).

Let us give a brief overview of the related existing statistical packages to the **RKHSMetaMod** package. The R package **sensitivity** is designed to implement sensitivity analysis methods and provides the approaches for numerical calculation of the Sobol indices. In particular, Kriging method can be used to reduce the number of the observations in global sensitivity analysis. The function `sobolGP` of the package **sensitivity** builds a Kriging based meta-model using the function `km` of the package **DiceKriging** (Roustant et al., 2012), and estimates its Sobol indices. This procedure can also be done using the function `km` and the function `fast99` of the package **sensitivity** (see Section 4.5. of Roustant et al. (2012)). In this case, the idea is once again to build a Kriging based meta-model using the function `km` and then estimate its Sobol indices using the function `fast99`. In both cases the true function is substituted by a Kriging based meta-model and then its Sobol indices are estimated. In the `sobolGP` function, the Sobol indices are estimated by the Monte Carlo integration, while the `fast99` function estimates them using the extended-FAST method (Saltelli et al., 1999). To reduce the computational burden when dealing with large datasets and complex models, in **RKHSMetaMod** package, we propose to use the empirical variances to estimate the Sobol indices (see Section Estimation of the Sobol indices). Besides, the estimation of the Sobol indices in the **RKHSMetaMod** package is done based on the RKHS meta-model which is a sparse estimator. It is beneficial since instead of calculating the Sobol indices of all groups $v \in \mathcal{P}$, only the Sobol indices associated with the groups in the support of the RKHS meta-model are computed (see Section Estimation of the Sobol indices). Moreover, the functions `sobolGP` and `fast99` provide the estimation of the first order and the total Sobol indices only, while the procedure in the **RKHSMetaMod** package makes it possible to estimate the high order Sobol indices. The R packages **DiceKriging** and **DiceOptim** (Deep Inside Computer Experiments Kriging/Optim) (Roustant et al., 2012) implement the Kriging based meta-models to estimate complex models in the high dimensional context. They provide different GP (Kriging) models corresponding to the Gaussian, Matérn, Exponential and Power-Exponential correlation functions. The estimation of the parameters of the correlation functions in these packages relies on the global optimizer with gradient `genoud` algorithm of the package **rgenoud** (Mebane and Sekhon, 2011). These packages do not implement any method of the sensitivity analysis themselves. However, some authors (see Section 4.5. of Roustant et al. (2012) for example) perform sensitivity analysis on their estimated meta-models by employing the functions of the package **sensitivity**. The R package **RobustGaSP** (Robust Gaussian Stochastic Process) (Gu et al., 2019) approximates a complex model by a GP meta-model. This package implements marginal posterior mode estimation of the GP model parameters. The estimation method in this package insures the robustness of the parameter estimation in the GP model, and allows one also to identify input variables that have no effect on the variability of the function under study. The R package **mlegp** (maximum likelihood estimates of Gaussian processes) (Dancik and Dorman, 2008) provides functions to implement both meta-modelling approaches and sensitivity analysis methods. It obtains maximum likelihood estimates of the GP model for the output of costly computer experiments. The GP models are built either on the basis of the Gaussian correlation function or on the basis of the first degree polynomial trend. The sensitivity analysis methods implemented in this package include Functional Analysis of Variance (FANOVA) decomposition, plot functions to obtain diagnostic plots, main effects, and second order interactions. The prediction quality of the meta-model depends on the quality of the estimation of its parameters and more precisely the estimation of parameters in the correlation functions (Kennedy and O'Hagan, 2000). The maximum likelihood estimation of these parameters often produce unstable results, and as a consequence, the obtained meta-model may have an inferior prediction quality (Gu et al., 2018; Gu, 2019). The **RKHSMetaMod** package is devoted to the meta-model estimation on the RKHS $\mathcal{H}$. It implements the convex optimization algorithms to calculate meta-models; provides the functions to compute the prediction error of the obtained meta-models; performs the sensitivity analysis on the obtained meta-models and more precisely calculate their Sobol

indices. The convex optimization algorithms used in this package are all written using C++ libraries, and are adapted to take into account the problem of high dimensionality in this context. This package is available from the Comprehensive R Archive Network (CRAN) (Kamari, 2019).

The organization of the paper is as follows: In the next Section, we describe the estimation method. In Section Algorithms, we present in details the algorithms used in the **RKHSMetaMod** package. Section RKHSMetaMod through examples includes two parts: In the first part, Section Simulation study, the performance of the **RKHSMetaMod** package functions is validated through a simulation study. In the second part, Section Comparison examples, the comparison in terms of the predictive accuracy between the RKHS meta-model and the Kriging based meta-models from **RobustGaSP** (Gu et al., 2019) and **DiceKriging** (Roustant et al., 2012) packages is given through two examples.

2 Estimation method

In this Section, we present: the RKHS ridge group sparse and the RKHS group lasso procedures (see RKHS ridge group sparse and RKHS group lasso procedures), the strategy of choosing the tuning parameters in the RKHS ridge group sparse algorithm (see Choice of the tuning parameters), and the calculation of the empirical Sobol indices of the RKHS meta-model (see Estimation of the Sobol indices).

RKHS ridge group sparse and RKHS group lasso procedures

Let us denote by n the number of observations. The dataset consists of a vector of n observations $Y = (Y_1, ..., Y_n)$, and a $n \times d$ matrix of features X with components $(X_{ai}, i = 1, ..., n, a = 1, ..., d) \in \mathbb{R}^{n\times d}$. For some tuning parameters γ_v, μ_v, $v \in \mathcal{P}$, the RKHS ridge group sparse criterion is defined by,

$$\mathcal{L}(f) = \frac{1}{n}\sum_{i=1}^{n}\Big(Y_i - f_0 - \sum_{v\in\mathcal{P}} f_v(X_{vi})\Big)^2 + \sum_{v\in\mathcal{P}}\gamma_v\|f_v\|_n + \sum_{v\in\mathcal{P}}\mu_v\|f_v\|_{\mathcal{H}_v}, \tag{10}$$

where X_v represents the matrix of variables corresponding to the v-th group, i.e. $X_v = (X_{vi}, i = 1, ..., n, v \in \mathcal{P}) \in \mathbb{R}^{n\times|\mathcal{P}|}$, and where $\|f_v\|_n$ is the empirical L^2-norm of f_v defined by the sample $\{X_{vi}\}_{i=1}^n$ as $\|f_v\|_n = \sqrt{\sum_{i=1}^n f_v^2(X_{vi})/n}$.

The penalty function in the criterion (10) is the sum of the Hilbert norm and the empirical norm, which allows us to select few terms in the additive decomposition of f over sets $v \in \mathcal{P}$. Moreover, the Hilbert norm favours the smoothness of the estimated f_v, $v \in \mathcal{P}$.
Let $\mathcal{F} = \{f : f = f_0 + \sum_{v\in\mathcal{P}} f_v,\ f_v \in \mathcal{H}_v,\ \|f_v\|_{\mathcal{H}_v} \le r_v,\ r_v \in \mathbb{R}^+\}$ be the set of functions. Then the RKHS meta-model is defined by,

$$\widehat{f} = \arg\min_{f\in\mathcal{F}} \mathcal{L}(f). \tag{11}$$

According to the Representer Theorem (Kimeldorf and Wahba, 1970), the non-parametric functional minimization problem described above is equivalent to a parametric minimization problem. Indeed, the solution of the minimization problem (11) belonging to the RKHS $\mathcal{H}$ is written as $f = f_0 + \sum_{v\in\mathcal{P}} f_v$, where for some matrix $\theta = (\theta_{vi}, i = 1, ..., n, v \in \mathcal{P}) \in \mathbb{R}^{n\times|\mathcal{P}|}$ we have for all $v \in \mathcal{P}$,

$$f_v(.) = \sum_{i=1}^{n}\theta_{vi}k_v(X_{vi}, .), \text{ and } \|f_v\|^2_{\mathcal{H}_v} = \sum_{i,i'=1}^{n}\theta_{vi}\theta_{vi'}k_v(X_{vi}, X_{vi'}). \tag{12}$$

Let $\|.\|$ be the Euclidean norm (called also L^2-norm) in $\mathbb{R}^n$, and for each $v \in \mathcal{P}$, let K_v be the $n \times n$ Gram matrix associated with the kernel $k_v(.,.)$, i.e. $(K_v)_{i,i'} = k_v(X_{vi}, X_{vi'})$. Let also $K^{1/2}$ be the matrix that satisfies $t(K^{1/2})K^{1/2} = K$, and let $\widehat{f}_0$ and $\widehat{\theta}$ be the minimizers of the following penalized least-squares criterion:

$$C(f_0, \theta) = \|Y - f_0 I_n - \sum_{v\in\mathcal{P}} K_v\theta_v\|^2 + \sqrt{n}\sum_{v\in\mathcal{P}}\gamma_v\|K_v\theta_v\| + n\sum_{v\in\mathcal{P}}\mu_v\|K_v^{1/2}\theta_v\|.$$

Then the estimator $\widehat{f}$ defined in Equation (11) satisfies,

$$\widehat{f}(X) = \widehat{f}_0 + \sum_{v\in\mathcal{P}}\widehat{f}_v(X_v) \text{ with } \widehat{f}_v(X_v) = \sum_{i=1}^{n}\widehat{\theta}_{vi}k_v(X_{vi}, X_v).$$

Remark 1 *The constraint $\|f_v\|_{\mathcal{H}_v} \le r_v$ is crucial for theoretical properties, but the value of r_v is generally*

unknown and has no practical usefulness. In this package, it is not taken into account in the parametric minimization problem.

For each $v \in \mathcal{P}$, let γ'_v and μ'_v be the weights that are chosen suitably. We define $\gamma_v = \gamma \times \gamma'_v$ and $\mu_v = \mu \times \mu'_v$ with $\gamma, \mu \in \mathbb{R}^+$.

Remark 2 *This formulation simplifies the choice of the tuning parameters since instead of tuning $2 \times |\mathcal{P}|$ parameters γ_v and μ_v, $v \in \mathcal{P}$, only two parameters γ and μ are tuned. Moreover, the weights γ'_v and μ'_v, $v \in \mathcal{P}$ may be of interest in practice. For example, one can take weights that increase with the cardinal of v in order to favour the effects with small interaction order between variables.*

For the sake of simplicity, in the rest of this paper for all $v \in \mathcal{P}$ the weights γ'_v and μ'_v are assumed to be set as one, and the RKHS ridge group sparse criterion is then expressed as follows:

$$C(f_0, \theta) = \|Y - f_0 I_n - \sum_{v \in \mathcal{P}} K_v \theta_v\|^2 + \sqrt{n}\gamma \sum_{v \in \mathcal{P}} \|K_v \theta_v\| + n\mu \sum_{v \in \mathcal{P}} \|K_v^{1/2} \theta_v\|. \tag{13}$$

If we consider only the second part of the penalty function in the criterion (13) (i.e. set $\gamma = 0$), we obtain the RKHS group lasso criterion as follows:

$$C_g(f_0, \theta) = \|Y - f_0 I_n - \sum_{v \in \mathcal{P}} K_v \theta_v\|^2 + n\mu \sum_{v \in \mathcal{P}} \|K_v^{1/2} \theta_v\|, \tag{14}$$

which is a group lasso criterion (Yuan and Lin, 2006) up to a scale transformation.

In the **RKHSMetaMod** package, the RKHS ridge group sparse algorithm is initialized using the solutions obtained by solving the RKHS group lasso algorithm. Indeed, the penalty function in the RKHS group lasso criterion (14) insures the sparsity in the solution. Therefore, for a given value of μ, by implementing the RKHS group lasso algorithm (see Section RKHS group lasso), a RKHS meta-model with few terms in its additive decomposition is obtained. The support and the coefficients of a RKHS meta-model which is obtained by implementing the RKHS group lasso algorithm will be denoted by $\widehat{S}_{\widehat{f}_{\text{Group Lasso}}}$ and $\widehat{\theta}_{\text{Group Lasso}}$, respectively. From now on, we denote the tuning parameter in the RKHS group lasso criterion by:

$$\mu_g = \sqrt{n}\mu. \tag{15}$$

Choice of the tuning parameters

While dealing with an optimization problem of a criterion of the form (13), one of the essential steps is to choose the appropriate tuning parameters. Cross-validation is generally used for that purpose. Nevertheless in the context of high-dimensional complex models, the computational time for a cross-validation procedure may be prohibitively high. Therefore, we propose a procedure based on a single testing dataset:

- we first choose, a grid of values of the tuning parameters μ and γ;
  Let $\mu_{\max}$ be the smallest value of μ_g (see Equation (15)), such that the solution to the minimization of the RKHS group lasso problem for all $v \in \mathcal{P}$ is $\theta_v = 0$. We have,

  $$\mu_{\max} = \max_v \left(2\|K_v^{1/2}(Y - \overline{Y})\| \right) / \sqrt{n}. \tag{16}$$

  In order to set up the grid of values of μ, one may find $\mu_{\max}$ and then a grid of values of μ could be defined by $\mu_l = \mu_{\max} / (\sqrt{n} \times 2^l)$ for $l \in \{1, ..., l_{\max}\}$. The grid of values of γ is chosen by the user.
- next, for the grid of values of μ and γ, we calculate a sequence of estimators. Each estimator associated with the pair (μ, γ) in the grid of values of μ and γ, denoted by $\widehat{f}_{(\mu,\gamma)}$, is the solution of the RKHS ridge group sparse optimization problem or the RKHS group lasso optimization problem if $\gamma = 0$.
- finally, the obtained estimators $\widehat{f}_{(\mu,\gamma)}$ are evaluated using a testing dataset, $\{(Y_i^{\text{test}}, X_i^{\text{test}})\}_{i=1}^{n^{\text{test}}}$. The prediction error associated with each estimator $\widehat{f}_{(\mu,\gamma)}$ is calculated by,

  $$\text{ErrPred}(\mu, \gamma) = \sum_{i=1}^{n^{\text{test}}} (Y_i^{\text{test}} - \widehat{f}_{(\mu,\gamma)}(X_i^{\text{test}}))^2 / n^{\text{test}},$$

where for $S_{\widehat{f}}$ being the support of the estimator $\widehat{f}_{(\mu,\gamma)}$ we have,

$$\widehat{f}_{(\mu,\gamma)}(X^{\text{test}}) = \widehat{f}_0 + \sum_{v\in S_{\widehat{f}}} \sum_{i=1}^{n} \widehat{\theta}_{vi} k_v(X_{vi}, X_v^{\text{test}}).$$

The pair $(\widehat{\mu},\widehat{\gamma})$ with the smallest value of the prediction error is chosen, and the estimator $\widehat{f}_{(\widehat{\mu},\widehat{\gamma})}$ is considered as the *best* estimator of the function m, in terms of the prediction error.

In the **RKHSMetaMod** package, the algorithm to calculate a sequence of the RKHS meta-models, the value of $\mu_{\max}$, and the prediction error are implemented as `RKHSMetMod`, `mu_max`, and `PredErr` functions, respectively. These functions are described in Section "Overview of the RKHSMetaMod functions" (supplementary materials), and illustrated in Example 1, Example 2, and Examples 1, 2, 3, respectively.

Estimation of the Sobol indices

The variance of the function m is estimated by the variance of the estimator $\widehat{f}$. As the estimator $\widehat{f}$ belongs to the RKHS $\mathcal{H}$, it admits the Hoeffding decomposition and,

$$\text{var}(\widehat{f}(X)) = \sum_{v\in\mathcal{P}} \text{var}(\widehat{f}_v(X_v)), \text{ where } \forall v \in \mathcal{P},\ \text{var}(\widehat{f}_v(X_v)) = E_X(\widehat{f}_v^2(X_v)) = \|\widehat{f}_v\|_2^2.$$

In order to reduce the computational cost in practice, one may estimate the variances of $\widehat{f}_v(X_v)$, $v \in \mathcal{P}$ by their empirical variances. Let $\widehat{f}_{v.}$ be the empirical mean of $\widehat{f}_v(X_{vi})$, $i = 1, ..., n$, then:

$$\widehat{\text{var}}(\widehat{f}_v(X_v)) = \frac{1}{n-1}\sum_{i=1}^{n}(\widehat{f}_v(X_{vi}) - \widehat{f}_{v.})^2.$$

For the groups v that do not belong to the support of $\widehat{f}$, we have $\widehat{S}_v = 0$ and for the groups v that belong to the support of $\widehat{f}$, the estimators of the Sobol indices of m are defined by,

$$\widehat{S}_v = \widehat{\text{var}}(\widehat{f}_v(X_v))/\sum_{v\in\mathcal{P}} \widehat{\text{var}}(\widehat{f}_v(X_v)).$$

In the **RKHSMetaMod** package, the algorithm to calculate the empirical Sobol indices $\widehat{S}_v$, $v \in \mathcal{P}$ is implemented as `SI_emp` function. This function is described in Section "Companion functions" (supplementary materials) and illustrated in Examples 1, 2, 3.

3 Algorithms

The **RKHSMetaMod** package implements two optimization algorithms: the RKHS ridge group sparse (see Algorithm 2) and the RKHS group lasso (see Algorithm 1). These algorithms rely on the Gram matrices K_v, $v \in \mathcal{P}$ that have to be positive definite. Therefore, the first and essential step in this package is to calculate these matrices and insure their positive definiteness. The algorithm of this step is described in the next Section. The second step is to estimate the RKHS meta-model. In the **RKHSMetaMod** package, two different objectives based on different procedures are considered to calculate this estimator:

1. The RKHS meta-model with the *best* prediction quality.
   The procedure to calculate the RKHS meta-model with the *best* prediction quality has been described in Section Choice of the tuning parameters: a sequence of values of the tuning parameters (μ,γ) is considered, and the RKHS meta-models associated with each pair of the values of (μ,γ) are calculated. For $\gamma = 0$, the RKHS meta-model is obtained by solving the RKHS group lasso optimization problem, while for $\gamma \neq 0$ the RKHS ridge group sparse optimization problem is solved to calculate the RKHS meta-model. The obtained estimators are evaluated by considering a new dataset and the RKHS meta-model with the minimum value of the prediction error is chosen as the *best* estimator.
2. The RKHS meta-model with at most $qmax$ groups in its support, i.e. $|S_{\widehat{f}}| \leq qmax$.
   First, the tuning parameter γ is set as zero. Then, a value of μ for which the number of groups $v \in \mathcal{P}$ in the solution of the RKHS group lasso optimization problem is equal to $qmax$, is computed. This value of μ will be denoted by μ_{qmax}. Finally, the RKHS meta-models containing at most $qmax$ groups in their support are obtained by implementing the RKHS ridge group

sparse algorithm for a grid of values of $\gamma \neq 0$ and μ_{qmax}. This procedure is described in more details in Section RKHS meta-model with *qmax* active groups.

Calculation of the Gram matrices

The available kernels in the **RKHSMetaMod** package are: Gaussian kernel, Matérn 3/2 kernel, Brownian kernel, quadratic kernel and linear kernel. The usual presentation of these kernels is given in Table 1. The choice of kernel, that is done by the user, determines the functional approximation space.

Kernel type	Mathematical formula for $u \in \mathbb{R}^n, v \in \mathbb{R}$	RKHSMetaMod name
Gaussian	$k_a(u,v) = \exp(-\|u-v\|^2/2r^2)$	`"gaussian"`
Matérn 3/2	$k_a(u,v) = (1+\sqrt{3}\|u-v\|/r)\exp(-\sqrt{3}\|u-v\|/r)$	`"matern"`
Brownian	$k_a(u,v) = \min(u,v)+1$	`"brownian"`
Quadratic	$k_a(u,v) = (u^T v+1)^2$	`"quad"`
Linear	$k_a(u,v) = u^T v+1$	`"linear"`

Table 1: List of the reproducing kernels used to construct the RKHS $\mathcal{H}$. The range parameters r in the Gaussian and Matérn 3/2 kernels are assumed to be fixed and set as $1/2$ and $\sqrt{3}/2$, respectively. The value 1 is added to the Brownian kernel to relax the constraint of nullity at the origin (Durrande et al., 2013).

For a chosen kernel, the algorithm to calculate the Gram matrices K_v, $v \in \mathcal{P}$ in the **RKHSMetaMod** package, is implemented as `calc_Kv` function. This algorithm is based on three essential points:

(1) Modify the chosen kernel:
In order to satisfy the conditions of constructing the RKHS $\mathcal{H}$ described in Section "RKHS construction" of the Appendix (supplementary materials), these kernels are modified according to Equation "(2)" (see the Appendix (supplementary materials)). Let us take the example of the Brownian kernel:
The RKHS associated with the Brownian kernel $k_a(X_a, X_a') = \min(X_a, X_a') + 1$ is well known to be $\mathcal{H}_a = \{f : [0,1] \to \mathbb{R} \text{ is absolutely continuous, and } f(0) = 0,\ \int_0^1 f'(X_a)^2 dX_a < \infty\}$, with the inner product $\langle f, h\rangle_{\mathcal{H}_a} = \int_0^1 f'(X_a)h'(X_a)dX_a$. Easy calculations lead to obtain the Brownian kernel as follows,

$$k_{0a} = \min(X_a, X_a') + 1 - (3/4)(1 + X_a - X_a^2/2)(1 + X_a' - X_a'^2/2).$$

The RKHS associated with kernel k_{0a} is the set $\mathcal{H}_{0a} = \{f \in \mathcal{H}_a : \int_0^1 f(X_a)dX_a = 0\}$, and we have $\mathcal{H} = \mathbb{1} + \sum_{v\in\mathcal{P}} \mathcal{H}_v = \{f : [0,1]^d \to \mathbb{R} : f = f_0 + \sum_{v\in\mathcal{P}} f_v(X_v),\ \text{with } f_v \in \mathcal{H}_v\}$.

Remark 3 *In the current version of the package, we consider the input variables $X = (X_1, ..., X_d)$ that are uniformly distributed on $[0,1]^d$. In order to consider the input variables that are not distributed uniformly, it suffices to modify a part of the function* `calc_Kv` *related to the calculation of the kernels k_{0a}, $a = 1, ..., d$. For example, for $X = (X_1, ..., X_d)$ being distributed with law $P_X = \prod_{a=1}^d P_a$ on $\mathcal{X} = \bigotimes_{a=1}^d \mathcal{X}_a \subset \mathbb{R}^d$, the kernel k_{0a} associated with the Brownian kernel is calculated as follows,*

$$k_{0a} = \min(X_a, X_a') + 1 - \frac{(\int_{\mathcal{X}_a}(\min(X_a, U)+1)dP_a)(\int_{\mathcal{X}_a}(\min(X_a', U)+1)dP_a)}{(\int_{\mathcal{X}_a}\int_{\mathcal{X}_a}(\min(U,V)+1)dP_a dP_a)}.$$

The other parts of function `calc_Kv` *remain unchanged.*

(2) Calculate the Gram matrices K_v for all v:
First, for all $a = 1, ..., d$, the Gram matrices K_a associated with kernels k_{0a} are calculated using Equation "(2)" (see the Appendix (supplementary materials)), $(K_a)_{i,i'} = k_{0a}(X_{ai}, X_{ai'})$. Then, for all $v \in \mathcal{P}$, the Gram matrices K_v associated with kernel $k_v = \prod_{a\in v} k_{0a}$ are computed by $K_v = \bigodot_{a\in v} K_a$.

(3) Insure the positive definiteness of the matrices K_v:
The output of function `calc_Kv` is one of the input arguments of the functions associated with the RKHS group lasso and the RKHS ridge group sparse algorithms. Throughout these algorithms we need to calculate the inverse and the square root of the matrices K_v. In order to avoid the numerical problems and insure the invertibility of the matrices K_v, it is mandatory to have these matrices positive definite. One way to render the matrices K_v positive definite is to add a nugget effect to them. That is, to modify matrices K_v by adding a diagonal with a constant term, i.e. $K_v + \text{epsilon} \times I_n$. The value of epsilon is computed based on the data and through a part of the algorithm of the function `calc_kv`. Let us briefly explain this part of the algorithm:

For each group $v \in \mathcal{P}$, let $\lambda_{v,i}, i = 1, ..., n$ be the eigenvalues associated with the matrix K_v. Set $\lambda_{v,\max} = \max_i \lambda_{v,i}$ and $\lambda_{v,\min} = \min_i \lambda_{v,i}$. For some fixed value of tolerance `tol`, and for each matrix K_v, if "$\lambda_{v,\min} < \lambda_{v,\max} \times$ `tol`", then, the eigenvalues of K_v are replaced by $\lambda_{v,i}+$`epsilon`, with `epsilon` being equal to $\lambda_{v,\max} \times$ `tol`. The value of `tol` is set as $1e^{-8}$ by default, but one may consider a smaller or a greater value for it depending on the kernel chosen and the value of n.

The function `calc_Kv` is described in Section "Companion functions" (supplementary materials) and illustrated in Example 2.

Optimization algorithms

The RKHS meta-model is the solution of one of the optimization problems: the minimization of the RKHS group lasso criterion presented in Equation (14) (if $\gamma = 0$), or the minimization of the RKHS ridge group sparse criterion presented in Equation (13) (if $\gamma \neq 0$). In the following, the algorithms to solve these optimization problems are presented.

RKHS group lasso

A popular technique for doing group wise variable selection is group lasso. With this procedure, depending on the value of the tuning parameter μ, an entire group of predictors may drop out of the model. An efficient algorithm for solving group lasso problem is the classical block coordinate descent algorithm (Boyd et al., 2011; Bubeck, 2015). Following the idea of Fu (1998), Yuan and Lin (2006) implemented a block wise descent algorithm for the group lasso penalized least-squares under the condition that the model matrices in each group are orthonormal. A block coordinate (gradient) descent algorithm for solving the group lasso penalized logistic regression is then developed by Meier et al. (2008). This algorithm is implemented in the R package **grplasso** available from CRAN (Meier, 2020). Yang and Zou (2015) proposed a unified algorithm named group wise majorization descent for solving the general group lasso learning problems by assuming that the loss function satisfies a quadratic majorization condition. The implementation of their work is done in the **gglasso** R package available from CRAN (Yang et al., 2020).

In order to solve the RKHS group lasso optimization problem, we use the classical block coordinate descent algorithm. The minimization of criterion $C_g(f_0, \theta)$ (see Equation (14)) is done along each group v at a time. At each step of the algorithm, the criterion $C_g(f_0, \theta)$ is minimized as a function of the current block's parameters, while the parameters values for the other blocks are fixed to their current values. The procedure is repeated until convergence. This procedure leads to Algorithm 1 (see the Appendix (supplementary materials) for more details on this procedure). In the **RKHSMetaMod**

Algorithm 1 RKHS group lasso algorithm:

Set $\theta_0 = [0]_{|\mathcal{P}| \times n}$
repeat
  Calculate $f_0 = \text{argmin}_{f_0} C_g(f_0, \theta)$
  for $v \in \mathcal{P}$ **do**
    Calculate $R_v = Y - f_0 - \sum_{v \neq w} K_w \theta_w$
    if $2\|K_v^{1/2} R_v\| / \sqrt{n} \leq \mu_g$ **then**
      $\theta_v \leftarrow 0$
    else
      $\theta_v \leftarrow \text{argmin}_{\theta_v} C_g(f_0, \theta)$
    end if
  end for
until convergence

package, the Algorithm 1 is implemented as `RKHSgrplasso` function. This function is described in Section "Companion functions" (supplementary materials) and illustrated in Example 2.

RKHS ridge group sparse

In order to solve the RKHS ridge group sparse optimization problem, we propose an adapted block coordinate descent algorithm. This algorithm is provided in two steps:

Step 1 Initialize the input parameters by the solutions of the RKHS group lasso algorithm for each value of the tuning parameter μ, and implement the RKHS ridge group sparse algorithm through the

active support of the RKHS group lasso solutions until it achieves convergence. This step is provided in order to decrease the execution time. In fact, instead of implementing the RKHS ridge group sparse algorithm over the set of all groups $\mathcal{P}$, it is implemented only over the groups in the support of the solution of the RKHS group lasso algorithm, $\widehat{S}_{\widehat{f}_{\text{Group Lasso}}}$.

Step 2 Re-initialize the input parameters with the obtained solutions of Step 1 and implement the RKHS ridge group sparse algorithm through all groups in $\mathcal{P}$ until it achieves convergence. This second step makes it possible to verify that no group is missing in the output of Step 1.

This procedure leads to Algorithm 2 (see the Appendix (supplementary materials) for more details on this procedure). In the **RKHSMetaMod** package the Algorithm 2 is implemented as `pen_MetMod`

Algorithm 2 RKHS ridge group sparse algorithm:

Step 1:
Set $\theta_0 = \widehat{\theta}_{\text{Group Lasso}}$ and $\widehat{\mathcal{P}} = \widehat{S}_{\widehat{f}_{\text{Group Lasso}}}$
repeat
  Calculate $f_0 = \text{argmin}_{f_0} C(f_0, \theta)$
  for $v \in \widehat{\mathcal{P}}$ **do**
    Calculate $R_v = Y - f_0 - \sum_{v \neq w} K_w \theta_w$
    Solve $J^* = \text{argmin}_{\widehat{t}_v \in \mathbb{R}^n} \{J(\widehat{t}_v),$ such that $\|K_v^{-1/2} \widehat{t}_v\| \leq 1\}$
    if $J^* \leq \gamma$ **then**
      $\theta_v \leftarrow 0$
    else
      $\theta_v \leftarrow \text{argmin}_{\theta_v} C(f_0, \theta)$
    end if
  end for
until convergence
Step 2:
Implement the same procedure as **Step 1** with $\theta_0 = \widehat{\theta}_{\text{old}}$, $\widehat{\mathcal{P}} = \mathcal{P} \triangleright \widehat{\theta}_{\text{old}}$ is the estimation of θ in **Step 1**.

function. This function is described in Section "Companion functions" (supplementary materials) and illustrated in Example 2.

RKHS meta-model with at most $qmax$ groups in its support

By considering some prior information about the data, one may be interested in a RKHS meta-model $\widehat{f}$ with the number of groups in its support not greater than some "$qmax$". In order to obtain such an estimator, we provide the following procedure in the **RKHSMetaMod** package:

- First, the tuning parameter γ is set as zero and a value of μ for which the solution of the RKHS group lasso algorithm, Algorithm 1, contains exactly $qmax$ groups in its support is computed. This value is denoted by μ_{qmax}.
- Then, the RKHS ridge group sparse algorithm, Algorithm 2, is implemented by setting the tuning parameter μ equal to μ_{qmax} and a grid of values of the tuning parameter $\gamma > 0$.

This procedure leads to Algorithm 3. This algorithm is implemented in the **RKHSMetaMod** package, as function `RKHSMetMod_qmax` (see Section "Main RKHSMetaMod functions" (supplementary materials) for more details on this function).

Remark 4 *As both terms in the penalty function of criterion (13) enforce sparsity to the solution, the estimator obtained by solving the RKHS ridge group sparse associated with the pair of the tuning parameters $(\mu_{qmax}, \gamma > 0)$ may contain a smaller number of groups than the solution of the RKHS group lasso optimization problem (i.e. the RKHS ridge group sparse with $(\mu_{qmax}, \gamma = 0)$). And therefore, the estimated RKHS meta-model contains at most "qmax" groups in its support.*

4 RKHSMetaMod through examples

Simulation study

Let us consider the g-function of Sobol (Saltelli et al., 2009) in the Gaussian regression framework, i.e. $Y = m(X) + \varepsilon$. The error term ε is a centered Gaussian random variable with variance σ^2, and m is the

Algorithm 3 Algorithm to estimate RKHS meta-model with at most *qmax* groups in its support:

1: Calculate $\mu_{\max} = \max_v 2\|K_v^{1/2}(Y - \overline{Y})\|/\sqrt{n}$
2: Set $\mu_1 = \mu_{\max}$ and $\mu_2 = \mu_{\max}$/`rat` ▷ `rat` is setted by user.
3: **repeat**
4: Implement RKHS group lasso algorithm, Algorithm 1, with $\mu_i = (\mu_1 + \mu_2)/2$
5: Set $q = |\widehat{S}_{\widehat{f}_{\text{Group Lasso}}}|$
6: **if** $q > qmax$ **then**
7: Set $\mu_1 = \mu_1$ and $\mu_2 = \mu_i$
8: **else**
9: Set $\mu_1 = \mu_i$ and $\mu_2 = \mu_2$
10: **end if**
11: **until** $q = qmax$ or $i >$ `Num` ▷ `Num` is setted by user.
12: Implement RKHS ridge group sparse algorithm, Algorithm 2, with $(\mu = \mu_{q_{max}}, \gamma > 0)$

g-function of Sobol defined over $[0,1]^d$ by,

$$m(X) = \prod_{a=1}^{d} \frac{|4X_a - 2| + c_a}{1 + c_a},\ c_a > 0. \tag{17}$$

The Sobol indices of the g-function can be expressed analytically:

$$\forall v \in \mathcal{P},\ S_v = \frac{1}{D}\prod_{a\in v} D_a,\ D_a = \frac{1}{3(1+c_a)^2},\ D = \prod_{a=1}^{d}(D_a + 1) - 1.$$

Set $c_1 = 0.2$, $c_2 = 0.6$, $c_3 = 0.8$ and $(c_a)_{a>3} = 100$. With these values of coefficients c_a, the variables X_1, X_2 and X_3 explain 99.98% of the variance of function $m(X)$ (see Table 4).

In this Section, three examples are presented. In all examples, the value of Dmax is set as three. Example 1 illustrates the use of the `RKHSMetMod` function by considering three different kernels, `"matern"`, `"brownian"`, and `"gaussian"` (see Table 1), and three datasets of $n \in \{50, 100, 200\}$ observations and $d = 5$ input variables. The larger datasets with $n \in \{1000, 2000, 5000\}$ observations and $d = 10$ input variables are studied in Examples 2 and 3. In each example, two independent datasets are generated: (X, Y) to estimate the meta-models, and (XT, YT) to estimate the prediction errors. The design matrices X and XT are the Latin Hypercube Samples of the input variables that are generated using `maximinLHS` function of the package **lhs** available at CRAN (Carnell, 2021):

```
library(lhs); X <- maximinLHS(n, d); XT <- maximinLHS(n, d)
```

The response variables Y and YT are calculated as $Y = m(X) + \varepsilon$ and $YT = m(XT) + \varepsilon_T$, where ε, and ε_T are centered Gaussian random variables with $\sigma^2 = (0.2)^2$:

```
a <- c(0.2, 0.6, 0.8, 100, 100, 100, 100, 100, 100, 100)[1:d]
g=1; for (i in 1:d) g = g*(abs(4*X[,i]-2)+a[i])/(1+a[i])
sigma <- 0.2
epsilon <- rnorm(n, 0, sigma^2); Y <- g + epsilon
gT=1; for (i in 1:d) gT = gT*(abs(4*XT[,i]-2)+a[i])/(1+a[i])
epsilonT <- rnorm(n, 0, sigma^2); YT <- gT + epsilonT
```

Example 1 *RKHS meta-model estimation using* `RKHSMetMod` *function:*

In this example, three datasets of n points `maximinLHS` over $[0,1]^d$ with $n \in \{50, 100, 200\}$ and $d = 5$ are generated, and a grid of five values of tuning parameters μ and γ is considered as follows:

$$\mu_{(1:5)} = \mu_{max}/(\sqrt{n} \times 2^{(2:6)}),\quad \gamma_{(1:5)} = (0.2, 0.1, 0.01, 0.005, 0).$$

For each dataset, the experiment is repeated $N_r = 50$ times. At each repetition, the RKHS meta-models associated with the pair of tuning parameters (μ, γ) are estimated using the `RKHSMetMod` function:

```
Dmax <- 3; kernel <- "matern" # kernel <- "brownian" # kernel <- "gaussian"
gamma <- c(0.2, 0.1, 0.01, 0.005, 0); frc <- 1/(0.5^(2:6))
res <- RKHSMetMod(Y, X, kernel, Dmax, gamma, frc, FALSE)
```

These meta-models are evaluated using a testing dataset. The prediction errors are computed for them using the `PredErr` function. The RKHS meta-model with minimum prediction error is chosen to be the

best estimator for the model. Finally, the Sobol indices are computed for the *best* RKHS meta-model using the function `SI_emp`:

```
Err <- PredErr(X, XT, YT, mu, gamma, res, kernel, Dmax)
SI <- SI_emp(res, Err)
```

The vector `mu` is the values of the tuning parameter μ that are calculated throughout the function `RKHSMetMod`. It could be recovered from the output of the `RKHSMetMod` function as follows:

```
mu <- vector()
l <- length(gamma); for(i in 1:length(frc)){mu[i] <- res[[(i-1)*l+1]]$mu}
```

The performance of this method for estimating a meta-model is evaluated by considering a third dataset $(m(X_i^{third}), X_i^{third})$, $i = 1, ..., N$, with $N = 1000$. The global prediction error is calculated as follows:

Let $\widehat{f}_r(.)$ be the *best* RKHS meta-model obtained in the repetition r, $r = 1, ..., N_r$, then

$$GPE = \frac{1}{N_r}\sum_{r=1}^{N_r}\frac{1}{N}\sum_{i=1}^{N}(\widehat{f}_r(X_i^{third}) - m(X_i^{third}))^2.$$

The values of GPE obtained for different kernels and values of n are given in Table 2. As expected

n	50	100	200
GPE_m	0.13	0.07	0.03
GPE_b	0.14	0.10	0.05
GPE_g	0.15	0.10	0.07

Table 2: Example 1: The columns of the table correspond to the different datasets with $n \in \{50, 100, 200\}$ and $d = 5$. Each line of the table, from up to down, gives the value of GPE obtained for each dataset associated with the "`matern`", "`brownian`" and "`gaussian`" kernels, respectively.

the value of GPE decreases as n increases. The lowest values of GPE are obtained when using the "`matern`" kernel.

In order to sum up the behaviour of the procedure for estimating the Sobol indices, we consider the mean square error (MSE) criterion obtained by $\sum_v(\sum_{r=1}^{N_r}(\widehat{S}_{v,r} - S_v)^2/N_r)$, where for each group v, S_v denotes the true values of the Sobol indices, and $\widehat{S}_{v,r}$ is the empirical Sobol indices of the *best* RKHS meta-model in repetition r. The obtained values of MSE for different kernels and values of n are given in Table 3. As expected, the values of MSE are smaller for larger values of n. The smallest values are

n	50	100	200
MSE_m	75.12	46.72	28.22
MSE_b	110.71	84.99	41.06
MSE_g	78.22	94.67	67.02

Table 3: Example 1: The columns of the table correspond to the different datasets with $n \in \{50, 100, 200\}$ and $d = 5$. Each line of the table, from up to down, gives the value of MSE obtained for each dataset associated with the "`matern`", "`brownian`" and "`gaussian`" kernels, respectively.

obtained when using the "`matern`" kernel.

The means of the empirical Sobol indices of the *best* RKHS meta-models through all repetitions for $n = 200$ and "`matern`" kernel are displayed in Table 4. It appears that the estimated Sobol indices

v	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$	sum
S_v	43.30	24.30	19.20	5.63	4.45	2.50	0.57	99.98
$\widehat{S}_{v,.}$	46.10	26.33	20.62	2.99	2.22	1.13	0.0	99.39

Table 4: Example 1: The first line of the table gives the true values of the Sobol indices $\times 100$. The second line gives the mean of the estimated empirical Sobol indices $(\widehat{S}_{v,.} = \sum_{r=1}^{N_r}\widehat{S}_{v,r}/N_r) \times 100$ greater than 10^{-2} calculated over fifty simulations for $n = 200$ and "`matern`" kernel. The sum of the Sobol indices is displayed in the last column.

are close to the true ones, nevertheless, they are overestimated for the main effects, i.e. groups $v \in \{\{1\}, \{2\}, \{3\}\}$, and underestimated for the interactions of order two and three, i.e. groups $v \in \{\{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$. Note that the strategy of choosing the tuning parameters is based

on the minimization of the prediction error of the estimated meta-model, which may not minimize the error of estimating the Sobol indices.

Taking into account the results obtained for this Example 1, the calculations in the rest of the examples is done using only the "matern" kernel.

Example 2 *A time-efficient strategy to obtain the "optimal" tuning parameters when dealing with large datasets:*

A dataset of n points maximinLHS over $[0,1]^d$ with $n = 1000$ and $d = 10$ is generated. First, we use functions calc_Kv and mu_max to compute the eigenvalues and eigenvectors of the positive definite matrices K_v, and the value of μ_{max}, respectively:

```
kernel <- "matern"; Dmax <- 3
Kv <- calc_Kv(X, kernel, Dmax, TRUE, TRUE)
mumax <- mu_max(Y, Kv$kv)
```

Then, we consider the two following steps:

1. Set $\gamma = 0$ and, $\mu_{(1:9)} = \mu_{max}/(\sqrt{n} \times 2^{(2:10)})$. Calculate the RKHS meta-models associated with the values of $\mu_g = \mu \times \sqrt{n}$ by using the function RKHSgrplasso. Gather the obtained RKHS meta-models in a list, res_g (while this job could be done using the function RKHSMetMod by setting $\gamma = 0$, in this example, we use the function RKHSgrplasso in order to avoid the re-calculation of K_v's at the next step). Thereafter, for each estimator in res_g, the prediction error is calculated by considering a new dataset and using the function PredErr. The value of μ with the smallest error of prediction in this step is denoted by μ_i. Let us implement this step:
   For a grid of values of μ_g, a sequence of the RKHS meta-models are calculated and gathered in the res_g list:

   ```
   mu_g <- c(mumax*0.5^(2:10))
   res_g <- list(); resg <- list()
   for(i in 1:length(mu_g)){
     resg[[i]] <- RKHSgrplasso(Y, Kv, mu_g[i], 1000, FALSE)
     res_g[[i]] <- list("mu_g"=mu_g, "gamma"=0, "MetaModel"=resg[[i]])
   }
   ```

   Output res_g contains nine RKHS meta-models and they are evaluated using a testing dataset:

   ```
   gamma <- c(0); Err_g <- PredErr(X, XT, YT, mu_g, gamma, res_g, kernel, Dmax)
   ```

   The prediction errors of the RKHS meta-models obtained in this step are displayed in Table 5. It appears that the minimum prediction error corresponds to the solution of the RKHS group lasso algorithm with $\mu_g = 0.041$, so $\mu_i = 0.041/\sqrt{n}$.

μ_g	1.304	0.652	0.326	0.163	0.081	0.041	0.020	0.010	0.005
$\gamma = 0$	0.197	0.156	0.145	0.097	0.063	0.055	0.056	0.063	0.073

Table 5: Example 2: Prediction errors associated with the RKHS meta-models computed in step 1.

2. Choose a smaller grid of values of μ, $(\mu_{(i-1)}, \mu_i, \mu_{(i+1)})$, and set a grid of values of $\gamma > 0$. Calculate the RKHS meta-models associated with each pair of the tuning parameters (μ, γ) by the function pen_MetMod. Calculate the prediction errors for the new sequence of the RKHS meta-models using the function PredErr. Compute the empirical Sobol indices for the *best* estimator. Let us go back to the implementation of the example and apply this step 2:
   The grid of the values of μ in this step is $(0.081, 0.041, 0.020)/\sqrt{n}$. The RKHS meta-models associated with this grid of the values of μ are gathered in a new list resgnew. We set $\gamma_{(1:4)} = (0.2, 0.1, 0.01, 0.005)$, and we calculate the RKHS meta-models for this new grid of the values of (μ, γ) using pen_MetMod function:

   ```
   gamma <- c(0.2, 0.1, 0.01, 0.005); mu <- c(mu_g[5], mu_g[6], mu_g[7])/sqrt(n)
   resgnew <- list()
   resgnew[[1]] <- resg[[5]]; resgnew[[2]] <- resg[[6]]; resgnew[[3]] <- resg[[7]]
   res <- pen_MetMod(Y, Kv, gamma, mu, resgnew, 0, 0)
   ```

   The output res is a list of twelve RKHS meta-models. These meta-models are evaluated using a new dataset, and their prediction errors are displayed in Table 6. The minimum prediction error is associated with the pair $(0.020/\sqrt{n}, 0.01)$, and the *best* RKHS meta-model is then $\widehat{f}_{(0.020/\sqrt{n}, 0.01)}$.

μ	$0.081/\sqrt{n}$	$0.041/\sqrt{n}$	$0.020/\sqrt{n}$
$\gamma = 0.2$	0.153	0.131	0.119
$\gamma = 0.1$	0.098	0.079	0.072
$\gamma = 0.01$	0.065	0.054	0.053
$\gamma = 0.005$	0.064	0.054	0.054

Table 6: Example 2: Obtained prediction errors in step 2.

The performance of this procedure for estimating the Sobol indices is evaluated using the relative error (RE) defined as follows:
For each v, let S_v be the true value of the Sobol indices displayed in Table 4 and $\widehat{S}_v$ be the estimated empirical Sobol indices. Then

$$RE = \sum_v |\widehat{S}_v - S_v| / S_v. \tag{18}$$

In Table 7 the estimated empirical Sobol indices, their sum, and the value of RE are displayed.

v	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$	sum	RE
$\widehat{S}_v$	42.91	25.50	20.81	4.40	3.84	2.13	0.00	99.60	1.64

Table 7: Example 2: The estimated empirical Sobol indices $\times 100$ greater than 10^{-2}. The last two columns show $\sum_v \widehat{S}_v$ and RE, respectively.

The obtained RE for each group v is smaller than 1.64%, therefore, the estimated Sobol indices in this example are very close to the true values of the Sobol indices displayed in the first row of Table 4.

Example 3 *Dealing with larger datasets:*

Two datasets of n points `maximinLHS` over $[0,1]^d$ with $n \in \{2000, 5000\}$ and $d = 10$ are generated. In order to obtain one RKHS meta-model associated with one pair of the tuning parameters (μ, γ), the number of coefficients to be estimated is equal to $n \times$vMax$= n \times 175$. Table 8 gives the execution time for different functions used throughout the Examples 1-3. In all examples we used a cluster of computers with: 2 Intel Xeon E5-2690 processors (2.90GHz) and 96Gb Ram (6x16Gb of memory 1600MHz). As we can see, the execution time increases fast as n increases. In Figure 1 the plot

(n,d)	`calc_Kv`	`mu_max`	`RKHSgrplasso`	`pen_MetMod`	$\lvert S_{\widehat{f}}\rvert$	sum
(100,5)	0.09s	0.01s	1s	2s	18	$\sim$ 3s
			2s	3s	19	$\sim$ 5s
(500,10)	33s	9s	247s	333s	39	$\sim$ 10min
			599s	816s	64	$\sim$ 24min
(1000,10)	197s	53s	959s	1336s	24	$\sim$ 42min
			2757s	4345s	69	$\sim$ 2h
(2000,10)	1498s	420s	3984s	4664s	12	$\sim$ 2h:56min
			12951s	22385s	30	$\sim$ 10h:20min
(5000,10)	34282s	6684s	38957s	49987s	11	$\sim$ 36h:05min
			99221s	111376s	15	$\sim$ 69h:52min

Table 8: Example 3: The kernel used is "`matern`". The execution time for the functions `RKHSgrplasso` and `pen_MetMod` is displayed in each row for two pairs of values of the tuning parameters $(\mu_1 = \mu_{max}/(\sqrt{n} \times 2^7), \gamma = 0.01)$ on up, and $(\mu_2 = \mu_{max}/(\sqrt{n} \times 2^8), \gamma = 0.01)$ on below. In the column $|S_{\widehat{f}}|$, the number of groups in the support of each estimated RKHS meta-model is displayed.

of the logarithm of the time (in seconds) versus the logarithm of n is displayed for the functions `calc_Kv`, `mu_max`, `RKHSgrplasso` and `pen_MetMod`. It appears that, the algorithms of these functions are of polynomial time $O(n^\alpha)$ with $\alpha \backsimeq 3$ for the functions `calc_Kv` and `mu_max`, and $\alpha \backsimeq 2$ for the functions `RKHSgrplasso` and `pen_MetMod`.

Taking into account the results obtained for the prediction error and the values of $(\widehat{\mu}, \widehat{\gamma})$ in Example 2, in this example, only two values of the tuning parameter $\mu_{(1:12)} = \mu_{max}/(\sqrt{n} \times 2^{(7:8)})$, and one value of the tuning parameter $\gamma = 0.01$ are considered. The RKHS meta-models associated with the pair of values (μ_i, γ), $i = 1, 2$ are estimated using the `RKHSMetMod` function:

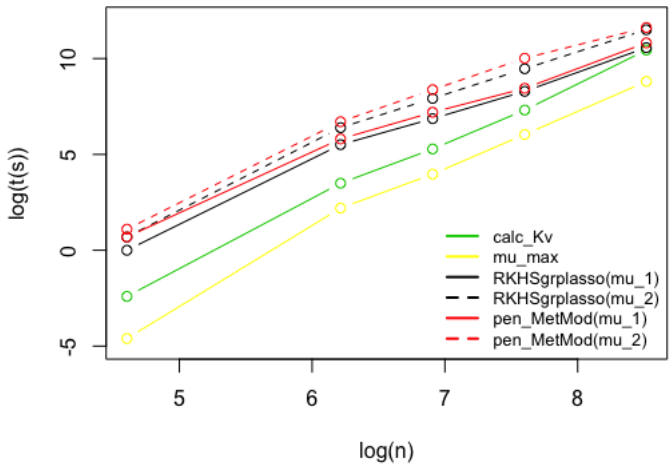

Figure 1: Example 3: Timing plot for $d = 10$, $n \in \{100, 300, 500, 1000, 2000, 5000\}$, and different functions of the **RKHSMetaMod** package. The logarithm of the execution time (in seconds) for the functions `RKHSgrplasso` and `pen_MetMod` is displayed for two pairs of values of the tuning parameters $(\mu_1 = \mu_{max}/(\sqrt{n} \times 2^7), \gamma = 0.01)$ in solid lines, and $(\mu_2 = \mu_{max}/(\sqrt{n} \times 2^8), \gamma = 0.01)$ in dashed lines.

```
kernel <- "matern"; Dmax <- 3
gamma <- c(0.01); frc <- 1/(0.5^(7:8))
res <- RKHSMetMod(Y, X, kernel, Dmax, gamma, frc, FALSE)
```

The prediction error and the empirical Sobol indices are then calculated for the obtained meta-models using the functions `PredErr` and `SI_emp`:

```
mu <- vector(); mu[1] <- res[[1]]$mu; mu[2] <- res[[2]]$mu
Err <- PredErr(X, XT, YT, mu, gamma, res, kernel, Dmax)
SI <- SI_emp(res, NULL)
```

Table 9 gives the estimated empirical Sobol indices as well as their sum, the values of RE (see Equation (18)), and the prediction errors associated with the obtained estimators. For $n = 5000$ we obtained the

n	v	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$	sum	RE	Err
2000	$\widehat{S}_{v;(\mu_1,\gamma)}$	45.54	24.78	21.01	3.96	3.03	1.65	0.00	99.97	2.12	0.052
	$\widehat{S}_{v;(\mu_2,\gamma)}$	45.38	25.07	19.69	4.36	3.66	1.79	0.00	99.95	1.79	0.049
5000	$\widehat{S}_{v;(\mu_1,\gamma)}$	44.77	25.39	20.05	4.49	3.38	1.90	0.00	99.98	1.81	0.049
	$\widehat{S}_{v;(\mu_2,\gamma)}$	43.78	24.99	19.56	5.43	3.90	2.32	0.00	99.98	1.29	0.047

Table 9: Example 3: The estimated empirical Sobol indices $\times 100$ greater than 10^{-2} associated with each estimated RKHS meta-model is printed. The last three columns show $\sum_v \widehat{S}_v$, RE, and the prediction error (Err), respectively. We have $\mu_1 = \mu_{max}/(\sqrt{n} \times 2^7)$, $\mu_2 = \mu_{max}/(\sqrt{n} \times 2^8)$ and $\gamma = 0.01$.

smaller values of RE and prediction error (Err). So as expected, the estimation of the Sobol indices as well as the prediction quality are better for larger values of n.

In Figure 2 the result of the estimation quality and the Sobol indices for the dataset with n equal to 5000, d equal to 10, and (μ_2, γ) are displayed. The line $y = x$ in red crosses the cloud of points as long as the values of the g-function are smaller than three. When the values of the g-function are greater than three, the estimator $\widehat{f}$ tends to underestimate the g-function. Concerning the Sobol indices obtained by the estimator $\widehat{f}$, as illustrated in the right-hand plot, with the exception of groups $\{1\}$, $\{2\}$, $\{3\}$, $\{1,2\}$, $\{1,3\}$, and $\{2,3\}$ for which we obtained significant values of the sobol indices, for all other groups the estimated sobol indices are very small and almost zero.

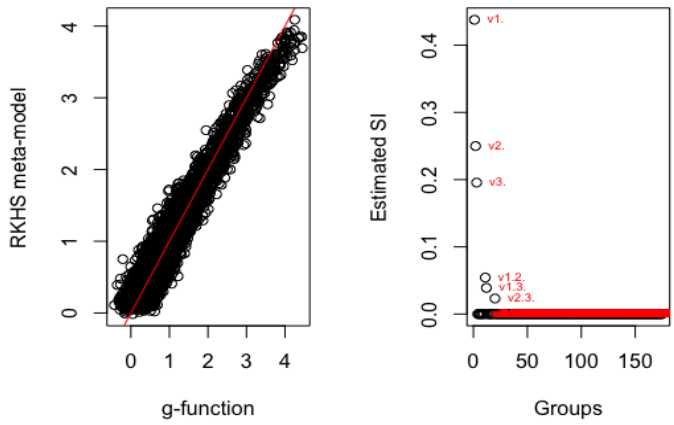

Figure 2: Example 3: On the left, the RKHS meta-model versus the g-function is plotted. On the right, the empirical Sobol indices in the y-axis and vMax= 175 groups in the x-axis are displayed.

Comparison examples

This section includes two examples. In the first example we reproduce an example from paper Gu et al. (2019) and compare the prediction quality of the RKHS meta-model with the GP (Kriging) based meta-models from the **RobusGaSP** (Gu et al., 2019) and **DiceKriging** (Roustant et al., 2012) packages. The objective is to evaluate the quality of the RKHS meta-model and to compare it with methods recently proposed for approximating complex models. In the first example we consider one-dimensional model and focus on the comparison between the true model and the estimated meta-model. In the second example we reproduce an example from paper Roustant et al. (2012) which allows us to compare the prediction quality of the RKHS meta-model with the Kriging based meta-model from **DiceKriging** package, as well as the estimation quality of the Sobol indices in our package with the well-known package **sensitivity**. For the sake of comparison between the three methods, the meta-models are calculated using the same experimental design and outputs, and the same kernel function available in three packages is used. However, in packages **RobustGaSP** and **DiceKriging** the range parameter r (see Table 1) in the kernel function is estimated by marginal posterior modes with the robust parameterization and by MLE with upper and lower bounds, respectively, while it is assumed to be fixed and set as $\sqrt{3}/2$ in the **RKHSMetaMod** package.

Example 4 *"The modified sine wave function":*

We consider the 1-dimensional modified sine wave function defined by $m(X) = 3\sin(5\pi X) + \cos(7\pi X)$ over $[0,1]$. The same experimental design as described in Gu et al. (2019) is considered: the design matrix X is a sequence of 12 equally spaced points on $[0,1]$, and the response variable Y is calculated as $Y = m(X)$:

```
X <- as.matrix(seq(0,1,1/11)); Y <- sinewave(X)
```

where `sinewave` function is defined in Gu et al. (2019). We build the GP based meta-models by the **RobustGaSP** and the **DiceKriging** packages using the constant mean function and kernel Matérn $3/2$:

```
library(RobustGaSP)
res.rgasp <- rgasp(design=X, response=Y, kernel_type="matern_3_2")
library(DiceKriging)
res.km <- km(design=X, response=Y, covtype="matern3_2")
```

As $d = 1$, we have $Dmax = 1$. We consider the grid of values of $\mu_{(1:9)} = \mu_{max}/(\sqrt{n} \times 2^{(2:10)})$ and $\gamma_{(1:5)} = (0.2, 0.1, 0.01, 0.005, 0)$. The RKHS meta-models associated with the pair of values (μ_i, γ_j), $i = 1, \cdots, 9$, $j = 1, \cdots, 5$ are estimated using the `RKHSMetMod` function:

```
kernel <- "matern"; Dmax <- 1
gamma <- c(0.2, 0.1, 0.01, 0.005,0); frc <- 1/(0.5^(2:10))
res <- RKHSMetMod(Y, X, kernel, Dmax, gamma, frc, FALSE)
```

Given a testing dataset (XT, YT), the prediction errors associated with the obtained RKHS meta-models are calculated using the `PredErr` function, and the *best* RKHS meta-model is chosen to be the estimator of the model $m(X)$:

```
XT <- as.matrix(seq(0,1,1/11)); YT <- sinewave(XT)
Err <- PredErr(X, XT, YT, mu, gamma, res, kernel, Dmax)
```

To compare these three estimators in terms of the prediction quality, we perform prediction on 100 test points, equally spaced in $[0, 1]$:

```
predict_X <- as.matrix(seq(0,1,1/99))
#prediction with the GP based meta-models:
rgasp.predict <- predict(res.rgasp, predict_X)
km.predict <- predict(res.km, predict_X, type='UK')
#prediction with the best RKHS meta-model:
res.predict <- prediction(X, predict_X, kernel, Dmax, res, Err)
```

The prediction results are plotted in Figure 3. The black circles that correspond to the prediction from the **RKHSMetMod** package are closer to the real output than the green and the blue circles corresponding to the predictive means from the **RobustGaSP** and **DiceKriging** packages. The meta-

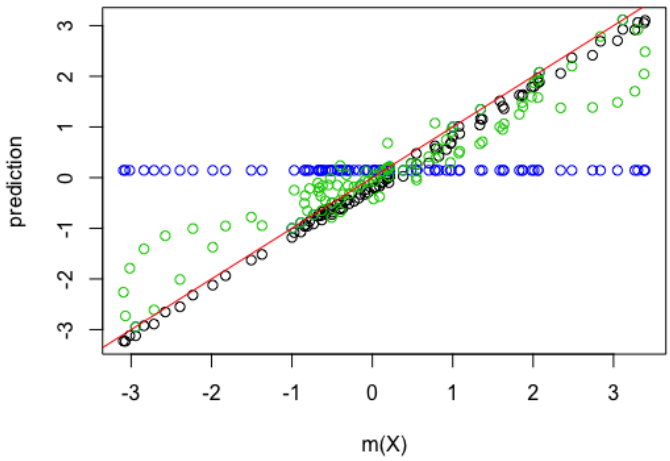

Figure 3: Example 4: Prediction of the modified sine wave function with 100 test points, equally spaced in $[0, 1]$. The x-axis is the real output and the y-axis is the prediction. The black circles are the prediction from **RKHSMetMod**, the green circles are the predictive mean from **RobustGaSP**, and the blue circles are the predictive mean from **DiceKriging**.

model results are plotted in Figure 4. The prediction from the **RKHSMetaMod** package plotted as the black curve is much more accurate as an estimate of the true function (plotted in red) than the predictive mean from the **RobustGaSP** and **DiceKriging** packages plotted as the blue and green curves, respectively. As already noted by Gu et al. (2019), for that sine wave example, the meta-model from the DiceKriging package "degenerates to the fitted mean with spikes at the design points".

Example 5 *"A standard SA 8-dimensional example":*

We consider the 8-dimensional g-function of Sobol implemented in the package **sensitivity**: the function $m(X)$ as defined in Equation (17) with coefficients $c_1 = 0$, $c_2 = 1$, $c_3 = 4.5$, $c_4 = 9$, $(c_a)_{a=5,6,7,8} = 99$. With these values of coefficients c_a, the variables X_1, X_2, X_3 and X_4 explain 99.96% of the variance of function $m(X)$ (see Table 10).

We consider the same experimental design as described in Roustant et al. (2012): the design matrices X and XT are 80-point optimal Latin Hypercube Samples of the input variables generated by the `optimumLHS` function of package **lhs**, and the response variables Y and YT are calculated as $Y = m(X)$, and $YT = m(XT)$ using `sobol.fun` function of the package **sensitivity**:

```
n <- 80; d <- 8
library(lhs); X <- optimumLHS(n, d); XT <- optimumLHS(n, d)
library(sensitivity); Y <- sobol.fun(X); YT <- sobol.fun(XT)
```

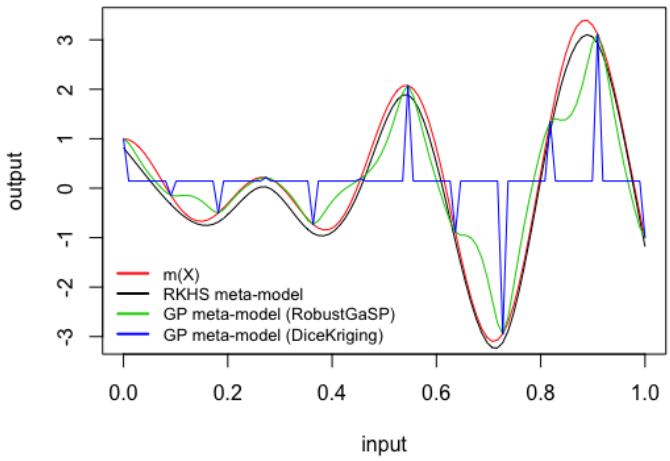

Figure 4: Example 4: The red curve is the graph of the modified sine wave function with 100 test points, equally spaced in $[0,1]$. The black curve is the prediction produced by the **RKHSMetaMod** package. The blue curve is the predictive mean by the **DiceKriging** package, and the green curve is the predictive mean produced by the **RobustGaSP** package.

Let us first consider the RKHS meta-model method. We set Dmax= 3, and we consider the grid of values of $\mu_{(1:9)} = \mu_{max}/(\sqrt{n} \times 2^{(2:10)})$, and $\gamma_{(1:5)} = (0.2, 0.1, 0.01, 0.005, 0)$. The RKHS meta-models associated with the pair of values $(\mu_i, \gamma_j), i = 1, \cdots, 9, j = 1, \cdots, 5$ are estimated using the `RKHSMetMod` function:

```
kernel <- "matern"; Dmax <- 3
gamma <- c(0.2, 0.1, 0.01, 0.005,0); frc <- 1/(0.5^(2:10))
res <- RKHSMetMod(Y, X, kernel, Dmax, gamma, frc, FALSE)
```

Given the testing dataset (XT, YT), the prediction errors associated with the obtained RKHS meta-models are calculated using `PredErr` function, and the *best* RKHS meta-model is chosen to be the estimator of the model $m(X)$. Finally, the Sobol indices are computed for the *best* RKHS meta-model using the function `SI_emp`:

```
Err <- PredErr(X, XT, YT, mu, gamma, res, kernel, Dmax)
SI <- SI_emp(res, Err)
```

Secondly, let us build the GP based meta-model. We use the `km` function of the package **DiceKriging** with the constant mean function and kernel Matérn 3/2:

```
library(DiceKriging)
res.km <- km(design = X, response = Y, covtype = "matern3_2")
```

The Sobol indices associated with the estimated GP based meta-model are calculated using `fast99` function of the package **sensitivity**:

```
SI.km <- fast99(model = kriging.mean, factors = d, n = 1000,
                      q = "qunif", q.arg = list(min = 0, max = 1), m = res.km)
```

where `kriging.mean` function is defined in Roustant et al. (2012).

The result of the estimation with the *best* RKHS meta-model and the Kriging based meta-model is drawn in Figure 5. The black circles that correspond to the *best* RKHS meta-model are closer to the real output than the blue circles corresponding to the GP based meta-model from the **DiceKriging** package. Another way to evaluate the prediction quality of the estimated meta-models is to consider the mean square error of the fitted meta-model computed by $\sum_{i=1}^{80}(m(X_i) - \widehat{f}(X_i))^2/80$. We obtained 3.96% and 0.07% for the Kriging based meta-model and the RKHS meta-model, respectively, which confirms the good behavior of the RKHS meta-model.

The estimated Sobol indices associated with the RKHS meta-model and the Kriging based meta-model are given in Table 10. As shown, with RKHS meta-model, we obtained non-zero values for the interactions of order two. Concerning the main effects, excepting the first one, the estimated Sobol

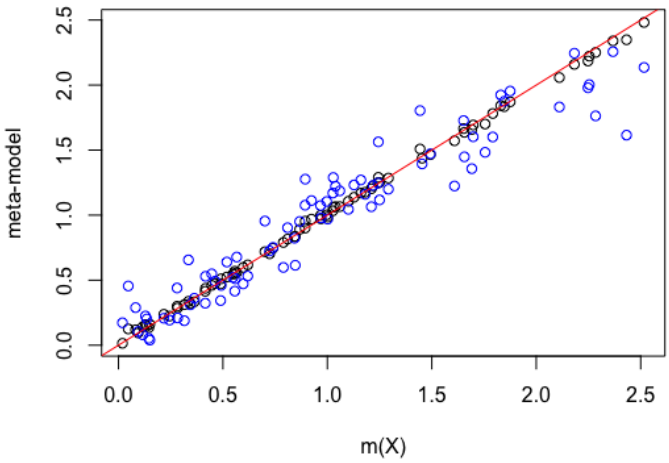

Figure 5: Example 5: The x-axis is the real output and the y-axis is the fitted meta-model. The black circles are the meta-model from **RKHSMetMod** and the blue circles are the meta-model from **DiceKriging**.

v	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$	$\{1,2,3\}$	$\{1,2,4\}$	sum
S_v	71.62	17.90	2.37	0.72	5.97	0.79	0.24	0.20	0.06	0.07	0.02	99.96
$\widehat{S}_v$	75.78	17.42	1.71	0.47	4.00	0.05	0.07	0.28	0.09	0.00	0.00	99.87
$\widehat{S}_{km_v}$	71.18	15.16	1.42	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	88.20

Table 10: Example 5: The true values of the Sobol indices $\times 100$ greater than 10^{-2} are given in the first raw. The estimated Sobol indices associated with the RKHS meta-model ($\widehat{S}_v$) and the Kriging based meta-model ($\widehat{S}_{km_v}$) are given in second and third rows, respectively.

indices with the RKHS meta-model are closer to the true ones. However, the interactions of order three are ignored by both meta-models. For a general comparison of the estimation quality of the Sobol indices, one may consider the criterion RE defined in Equation (18), which is equal to 7.95 for the Kriging based meta-model, and 5.59 for the RKHS meta-model. Comparing the values of RE, we can point out that the Sobol indices are better estimated with the RKHS meta-model in that model.

5 Summary and discussion

In this paper, we proposed an R package, called **RKHSMetaMod**, that estimates a meta-model of a complex model m. This meta-model belongs to a reproducing kernel Hilbert space constructed as a direct sum of Hilbert spaces (Durrande et al., 2013). The estimation of the meta-model is carried out via a penalized least-squares minimization allowing both to select and estimate the terms in the Hoeffding decomposition, and therefore, to select the Sobol indices that are non-zero and estimate them (Huet and Taupin, 2017). This procedure makes it possible to estimate the Sobol indices of high order, a point known to be difficult in practice. Using the convex optimization tools, **RKHSMetaMod** package implements two optimization algorithms: the minimization of the RKHS ridge group sparse criterion (13) and the RKHS group lasso criterion (14). Both of these algorithms rely on the Gram matrices K_v, $v \in \mathcal{P}$ and their positive definiteness. Currently, the package considers only uniformly distributed input variables. If one is interested by another distribution of the input variables, it suffices to modify the calculation of the kernels k_{0a}, $a = 1, ..., d$ in the function `calc_Kv` of this package (see Remark 3). The available kernels in the **RKHSMetaMod** package are: Gaussian kernel (with the fixed range parameter $r = 1/2$), Matérn kernel (with the fixed range parameter $r = \sqrt{3}/2$), Brownian kernel, quadratic kernel and linear kernel (see Table 1). With regard to the problem being under study, one may consider other kernels or kernels with different values of the range parameter r and add them easily to the list of the kernels in the `calc_Kv` function. For the large values of n and d the calculation and storage of eigenvalues and eigenvectors of all the Gram matrices K_v, $v \in \mathcal{P}$ require a lot of time and a very large amount of memory. In order to optimize the execution time and also

the storage memory, except for a function that is written in R, all of the functions of **RKHSMetaMod** package are written using the efficient C++ libraries through **RcppEigen** and **RcppGSL** packages. These functions are then interfaced with the R environment in order to contribute a user friendly package.

Bibliography

G. Blatman and B. Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of computational Physics*, 230:2345–2367, 2011. [p107]

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL https://doi.org/10.1561/2200000016. [p114]

S. Bubeck. Convex Optimization: Algorithms and Complexity. *Found. Trends Mach. Learn.*, 8(3-4): 231–357, Nov. 2015. ISSN 1935-8237. doi: 10.1561/2200000050. URL https://doi.org/10.1561/2200000050. [p114]

R. Carnell. *lhs: Latin Hypercube Samples*, 2021. URL https://CRAN.R-project.org/package=lhs. R package version 1.1.3. [p116]

G. M. Dancik and K. S. Dorman. mlegp: statistical analysis for computer models of biological systems using R. *Bioinformatics*, 24(17):1966, 2008. [p109]

N. Durrande, D. Ginsbourger, and O. Roustant. Additive Covariance kernels for high-dimensional Gaussian Process modeling. *Annales de la faculté des sciences de Toulouse Mathématiques*, 21(3):481–499, 4 2012. URL http://eudml.org/doc/251000. [p107]

N. Durrande, D. Ginsbourger, O. Roustant, and L. Carraro. ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis. *Journal of Multivariate Analysis*, 115:57 – 67, 2013. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2012.08.016. URL http://www.sciencedirect.com/science/article/pii/S0047259X1200214X. [p106, 108, 113, 124]

W. J. Fu. Penalized Regressions: The Bridge versus the Lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998. ISSN 10618600. URL http://www.jstor.org/stable/1390712. [p114]

M. Gu. Jointly Robust Prior for Gaussian Stochastic Process in Emulation, Calibration and Variable Selection. *Bayesian Analysis*, 14(3):857 – 885, 2019. doi: 10.1214/18-BA1133. URL https://doi.org/10.1214/18-BA1133. [p109]

M. Gu, X. Wang, and J. O. Berger. Robust Gaussian stochastic process emulation. *The Annals of Statistics*, 46(6A):3038 – 3066, 2018. doi: 10.1214/17-AOS1648. URL https://doi.org/10.1214/17-AOS1648. [p109]

M. Gu, J. Palomo, and O. Berger, James. RobustGaSP: Robust Gaussian Stochastic Process Emulation in R. *The R Journal*, 11(1):112, 2019. ISSN 2073-4859. doi: 10.32614/rj-2019-011. URL http://dx.doi.org/10.32614/rj-2019-011. [p109, 110, 121, 122]

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015. ISBN 1498712169, 9781498712163. [p108]

T. Homma and A. Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1):1 – 17, 1996. ISSN 0951-8320. doi: https://doi.org/10.1016/0951-8320(96)00002-6. URL http://www.sciencedirect.com/science/article/pii/0951832096000026. [p107]

S. Huet and M.-L. Taupin. Metamodel construction for sensitivity analysis. *ESAIM: Procs*, 60:27–69, 2017. doi: 10.1051/proc/201760027. URL https://doi.org/10.1051/proc/201760027. [p106, 108, 124]

H. Kamari. *RKHSMetaMod: Ridge Group Sparse Optimization Problem for Estimation of a Meta Model Based on Reproducing Kernel Hilbert Spaces*, 2019. URL https://CRAN.R-project.org/package=RKHSMetaMod. R package version 1.1. [p110]

M. C. Kennedy and A. O'Hagan. Bayesian Calibration of Computer Models. *Journal of the Royal Statistical Society, Series B, Methodological*, 63:425–464, 2000. [p109]

G. S. Kimeldorf and G. Wahba. A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines. *Ann. Math. Statist.*, 41(2):495–502, 04 1970. doi: 10.1214/aoms/1177697089. URL http://dx.doi.org/10.1214/aoms/1177697089. [p110]

J. P. C. Kleijnen. *Design and Analysis of Simulation Experiments*. Springer Publishing Company, Incorporated, 1st edition, 2007. ISBN 0387718125, 9780387718125. [p107]

J. P. C. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707 – 716, 2009. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2007.10.013. URL http://www.sciencedirect.com/science/article/pii/S0377221707010090. [p107]

L. Le Gratiet, C. Cannamela, and B. Iooss. A Bayesian Approach for Global Sensitivity Analysis of (Multifidelity) Computer Codes. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):336–363, 2014. doi: 10.1137/130926869. URL https://doi.org/10.1137/130926869. [p107]

L. Le Gratiet, S. Marelli, and B. Sudret. *Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes*, pages 1289–1325. Springer International Publishing, Cham, 2017. ISBN 978-3-319-12385-1. doi: 10.1007/978-3-319-12385-1_38. URL https://doi.org/10.1007/978-3-319-12385-1_38. [p108]

Y. Lin and H. H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *Ann. Statist.*, 34(5):2272–2297, 10 2006. doi: 10.1214/009053606000000722. URL https://doi.org/10.1214/009053606000000722. [p108]

H. Liu, L. Wasserman, and J. D. Lafferty. Nonparametric regression and classification with joint sparsity constraints. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 969–976. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3616-nonparametric-regression-and-classification-with-joint-sparsity-constraints.pdf. [p108]

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. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2008.07.008. URL http://www.sciencedirect.com/science/article/pii/S0951832008001981. [p107]

W. Mebane and J. Sekhon. Genetic Optimization Using Derivatives: The rgenoud Package for R. *Journal of Statistical Software, Articles*, 42(11):1–26, 2011. ISSN 1548-7660. doi: 10.18637/jss.v042.i11. URL https://www.jstatsoft.org/v042/i11. [p109]

L. Meier. *grplasso: Fitting User-Specified Models with Group Lasso Penalty*, 2020. URL https://CRAN.R-project.org/package=grplasso. R package version 0.4-7. [p114]

L. Meier, S. van de Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society. Series B*, 70(1):53–71, 2008. doi: 10.1111/j.1467-9868.2007.00627.x. [p114]

L. Meier, S. van de Geer, and P. Bühlmann. High-dimensional additive modeling. *Ann. Statist.*, 37(6B):3779–3821, 12 2009. doi: 10.1214/09-AOS692. URL https://doi.org/10.1214/09-AOS692. [p108]

J. E. Oakley and A. O'Hagan. Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, 2004. doi: 10.1111/j.1467-9868.2004.05304.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2004.05304.x. [p107]

G. Raskutti, M. J. Wainwright, and B. Yu. Lower bounds on minimax rates for nonparametric regression with additive sparsity and smoothness. In *Advances in Neural Information Processing Systems*, 2009. [p108]

G. Raskutti, M. J. Wainwright, and B. Yu. Minimax-optimal Rates for Sparse Additive Models over Kernel Classes via Convex Programming. *J. Mach. Learn. Res.*, 13(1):389–427, Feb. 2012. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2503308.2188398. [p108]

P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009. doi: 10.1111/j.1467-9868.2009.00718.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2009.00718.x. [p108]

O. Roustant, D. Ginsbourger, and Y. Deville. DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. *Journal of Statistical Software, Articles*, 51(1):1–55, 2012. ISSN 1548-7660. doi: 10.18637/jss.v051.i01. URL https://www.jstatsoft.org/v051/i01. [p109, 110, 121, 122, 123]

A. Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145(2):280 – 297, 2002. ISSN 0010-4655. doi: https://doi.org/10.1016/S0010-4655(02)00280-1. URL http://www.sciencedirect.com/science/article/pii/S0010465502002801. [p107]

A. Saltelli, S. Tarantola, and K. P.-S. Chan. A Quantitative Model-Independent Method for Global Sensitivity Analysis of Model Output. *Technometrics*, 41(1):39–56, 1999. doi: 10.1080/00401706.1999.10485594. URL https://www.tandfonline.com/doi/abs/10.1080/00401706.1999.10485594. [p109]

A. Saltelli, K. Chan, and E. Scott. *Sensitivity Analysis*. Wiley, 2009. ISBN 9780470743829. URL https://books.google.fr/books?id=gOcePwAACAAJ. [p115]

W. Schoutens. *Stochastic Processes and Orthogonal Polynomials*. Lecture Notes in Statistics. Springer New York, 2000. ISBN 9780387950150. URL https://books.google.fr/books?id=V2BS_Dmp0XoC. [p107]

I. M. Sobol. Sensitivity Estimates for Nonlinear Mathematical Models. In *Sensitivity Estimates for Nonlinear Mathematical Models*, 1993. [p106, 107]

C. Soize and R. Ghanem. Physical Systems with Random Uncertainties: Chaos Representations with Arbitrary Probability Measure. *SIAM Journal on Scientific Computing*, 26(2):395–410, 2004. doi: 10.1137/S1064827503424505. URL https://doi.org/10.1137/S1064827503424505. [p107]

B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964 – 979, 2008. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2007.04.002. URL http://www.sciencedirect.com/science/article/pii/S0951832007001329. Bayesian Networks in Dependability. [p107]

A. W. Van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998. doi: 10.1017/CBO9780511802256. [p106]

W. J. Welch, R. J. Buck, J. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris. Screening, Predicting, and Computer Experiments. *Technometrics*, 34(1):15–25, 1992. ISSN 00401706. URL http://www.jstor.org/stable/1269548. [p107]

N. Wiener. The Homogeneous Chaos. *American Journal of Mathematics*, 60(4):897–936, 1938. ISSN 00029327, 10806377. URL http://www.jstor.org/stable/2371268. [p107]

Y. Yang and H. Zou. A Fast Unified Algorithm for Solving Group-lasso Penalize Learning Problems. *Statistics and Computing*, 25(6):1129–1141, Nov. 2015. ISSN 0960-3174. doi: 10.1007/s11222-014-9498-5. URL http://dx.doi.org/10.1007/s11222-014-9498-5. [p114]

Y. Yang, H. Zou, and S. Bhatnagar. *gglasso: Group Lasso Penalized Learning Using a Unified BMD Algorithm*, 2020. URL https://CRAN.R-project.org/package=gglasso. R package version 1.5. [p114]

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006. doi: 10.1111/j.1467-9868.2005.00532.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00532.x. [p111, 114]

Halaleh Kamari
UMR 8071 Laboratoire de Mathématiques et Modélisation d'Évry (LaMME)
Bâtiment IBGBI de l'Université d'Evry Val d'Essonne, 23 Bd de France, 91037 Evry CEDEX
France
halala.kamari@gmail.com

Sylvie Huet
UR 1404 Mathématiques et Informatique Appliquées du Génome à l'Environnement (MaIAGE)
Bâtiment 210 de INRAE, Domaine de Vilvert, 78352 JOUY-EN-JOSAS Cedex
France
shuet.inra@gmail.com

Marie-Luce Taupin
UMR 8071 Laboratoire de Mathématiques et Modélisation d'Évry (LaMME)
Bâtiment IBGBI de l'Université d'Evry Val d'Essonne, 23 Bd de France, 91037 Evry CEDEX
France
marie-luce.taupin@univ-evry.fr