EUROPEAN
COMMISSION
Community research

PAMINA
Performance Assessment Methodologies in Application to Guide the Development of the Safety Case

(Contract Number: FP6-036404)

PERFORMING SENSITIVITY ANALYSIS OF CPU TIME CONSUMING MODELS USING METAMODELS
MILESTONE (N°:M2.1.D.5)

Authors: **Bertrand Iooss and Amandine Marrel**
(Commissariat à l'Énergie Atomique)

Date of issue of this report : **07/04/08**

Start date of project : **01/10/2006**

Duration : 36 Months

Project co-funded by the European Commission under the Euratom Research and Training Programme on Nuclear Energy within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
RE	Restricted to a group specified by the partners of the [PAMINA] project	
CO	Confidential, only for partners of the [PAMINA] project	

Foreword

The work presented in this report was developed within the Integrated Project PAMINA: **P**erformance **A**ssessment **M**ethodologies **IN** **A**pplication to Guide the Development of the Safety Case. This project is part of the Sixth Framework Programme of the European Commission. It brings together 25 organisations from ten European countries and one EC Joint Research Centre in order to improve and harmonise methodologies and tools for demonstrating the safety of deep geological disposal of long-lived radioactive waste for different waste types, repository designs and geological environments. The results will be of interest to national waste management organisations, regulators and lay stakeholders.

The work is organised in four Research and Technology Development Components (RTDCs) and one additional component dealing with knowledge management and dissemination of knowledge:

- In RTDC 1 the aim is to evaluate the state of the art of methodologies and approaches needed for assessing the safety of deep geological disposal, on the basis of comprehensive review of international practice. This work includes the identification of any deficiencies in methods and tools.

- In RTDC 2 the aim is to establish a framework and methodology for the treatment of uncertainty during PA and safety case development. Guidance on, and examples of, good practice will be provided on the communication and treatment of different types of uncertainty, spatial variability, the development of probabilistic safety assessment tools, and techniques for sensitivity and uncertainty analysis.

- In RTDC 3 the aim is to develop methodologies and tools for integrated PA for various geological disposal concepts. This work includes the development of PA scenarios, of the PA approach to gas migration processes, of the PA approach to radionuclide source term modelling, and of safety and performance indicators.

- In RTDC 4 the aim is to conduct several benchmark exercises on specific processes, in which quantitative comparisons are made between approaches that rely on simplifying assumptions and models, and those that rely on complex models that take into account a more complete process conceptualization in space and time.

The work presented in this report was performed in the scope of RTDC 2.

All PAMINA reports can be downloaded from http://www.ip-pamina.eu.

Direction de l'Énergie Nucléaire
Direction du CEA/Cadarache
Département d'Études des Réacteurs
Service d'Études des Systèmes Innovants
Laboratoire de Conduite et Fiabilité des Réacteurs

CEA/DEN/CAD/DER/SESI/LCFR
DO 25 21/03/08
08PPMO000037

BORDEREAU D'ENVOI

Date : 7 AVR. 2008 Réf. : CEA/DEN/CAD/DER/SESI/LCFR/NT DO 7 21/03/08

Designation : Note Technique « Performing sensitivity analysis of cpu time consuming models using metamodels »

Auteurs : B. IOOSS et A. MARREL

Destinataires	Nombre	Observations
GALSON SCIENCES LTD		Transmission par courriel du document complet
A. KHURSHEED, D. GALSON	2	
ENRESA : J. ALONSO	1	
NRG : J. GRUPA	1	
GRS-B : D.A. BECKER	1	
GRS-K : K. RÖHLIG	1	
JRC : R. BOLADO	1	
Facilia : R. AVILA	1	
SCK/CEN : J. MARIVOET	1	
Institut Kurchatov		
E. VOLKOVA	1	
CEA/DEN/SACLAY		
DSOE/SIMULATION : B. BRUN	1	
DM2S/SFME/LGLS : V. BERGEAU, J-M. MARTINEZ	2	
DM2S/SFME/MTMS : A. GENTY	1	
CEA/DEN/CADARACHE		
DTN/SMTM/LMTE :		
C. TIFFREAU, F. JOURDAIN, A. MARREL	3	
DER/SESI : J-C. GARNIER	1	
DER/SESI/LCFR :		
F. BERTRAND, B. IOOSS	2	
M. MARQUES, N. PEROT	2	
DER/SESI/LCFR	1	1 exemplaire papier en circulation
DER	1	Transmission par courriel du résumé
DER/SPEX – SPRC – SRES – SSTH	4	
SESI/LCSI - LESA	2	

Le Chef du DER/SESI : Jean-Claude GARNIER

Proposal/Contract no.: **FP6-036404**

Project acronym: PAMINA

Project title: **PERFORMANCE ASSESSMENT METHODOLOGIES IN APPLICATION TO GUIDE THE DEVELOPMENT OF THE SAFETY CASE**

Instrument: Integrated Project

Thematic Priority: Management of Radioactive Waste and Radiation Protection and other activities in the field of Nuclear Technologies and Safety

RTDC2 - WP2.1.D – Techniques for sensitivity and uncertainty analysis - CEA contribution

Due date of deliverable: 31.03.08
Actual submission date:

Start date of project: 01.10.2006
Duration: 36 months

Revision: 1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination level

PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	x
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

TITLE: PERFORMING SENSITIVITY ANALYSIS OF CPU TIME CONSUMING MODELS USING METAMODELS

AUTHOR(S) : Bertrand IOOSS and Amandine MARREL

ABSTRACT:

This report consists in deliverable CEA/DEN/DER for the component RTDC 2 of European project PAMINA (Performance Assessment Methodologies IN Application to guide the development of the safety case) 6th FP. This task concerns the presentation of new methods to perform sensitivity analysis for cpu time consuming computer codes. In this report, one restricts to methodological aspects. We describe a recent technique based on the use of a metamodel, i.e. a cpu time inexpensive mathematical function fitted and validated on a few simulations of the computer code. We show how to fit and use one of the most popular metamodels: the Gaussian process model which extends the kriging principles of geostatistics to numerical experiments. Its formulation allows to derive analytical formulas for the sensitivity indices without running other simulations of the computer code.

KEY WORDS: PAMINA, UNCERTAINTY, SENSITIVITY, METAMODEL, RESPONSE SURFACE, KRIGING, GAUSSIAN PROCESS

PAMINA Sixth Framework programme, 28.03.2008

CEA/DEN/CAD/DER/SESI/LCFR/NT
DO 7 21/03/08
08PPBT000010

DIRECTION DE L'ENERGIE NUCLÉAIRE
DIRECTION DU CENTRE DE CADARACHE
DÉPARTEMENT D'ÉTUDES DES RÉACTEURS
SERVICE D'ÉTUDES DES SYSTÈMES INNOVANTS
LABORATOIRE DE CONDUITE ET FIABILITE DES REACTEURS

NT	CEA/DEN/CAD/DER/SESI/LCFR/NT DO 7 21/03/08	0	5H62	A-SUPAR-02-03	DSOE/SIMUL/SUPAR	1/36
NATURE	CHRONO UNITÉ	INDICE	UNITÉ	ÉLÉMENT D'OTP	CLASSEMENT UNITÉ	PAGE

Note Technique

TITRE : REALISATION D'ANALYSES DE SENSIBILITE POUR DES CODES COUTEUX EN TEMPS DE CALCUL GRACE A L'UTILISATION DE METAMODELES

TITLE : PERFORMING SENSITIVITY ANALYSIS OF CPU TIME CONSUMING MODELS USING METAMODELS

AUTEUR(S) : B. IOOSS, A. MARREL (DTN/SMTM)

RESUME : Ce rapport constitue le livrable du CEA/DEN/DER pour le composant RTDC 2 du projet européen PAMINA (Performance Assessment Methodologies IN Application to guide the development of the safety case) du 6ème PCRD. Cette tâche concerne la présentation de nouvelles méthodes pour réaliser des analyses de sensibilité sur des modèles excessivement coûteux en temps de calcul. Dans ce rapport, on se restreint aux aspects méthodologiques. Nous décrivons une technique récente basée sur l'utilisation d'un métamodèle, i.e. une fonction mathématique, dont l'évaluation se fait avec un temps de calcul négligeable, ajustée et validée sur quelques simulations du code de calcul. Nous montrons comment ajuster et utiliser l'un des métamodèles les plus populaires : le modèle processus gaussien qui étend les principes du krigeage en géostatistique aux expériences numériques. Sa formulation permet d'obtenir des formules analytiques pour les indices de sensibilité des variables d'entrée sans effectuer de nouvelles simulations sur le code de calcul.

ABSTRACT : This report consists in deliverable CEA/DEN/DER for the component RTDC 2 of European project PAMINA (Performance Assessment Methodologies IN Application to guide the development of the safety case) 6th FP. This task concerns the presentation of new methods to perform sensitivity analysis for cpu time consuming computer codes. In this report, one restricts to methodological aspects. We describe a recent technique based on the use of a metamodel, i.e. a cpu time inexpensive mathematical function fitted and validated on a few simulations of the computer code. We show how to fit and use one of the most popular metamodels: the Gaussian process model which extends the kriging principles of geostatistics to numerical experiments. Its formulation allows to derive analytical formulas for the sensitivity indices without running other simulations of the computer code.

KEYWORDS : PAMINA, UNCERTAINTY, SENSITIVITY, METAMODEL, RESPONSE SURFACE, KRIGING, GAUSSIAN PROCESS

MOTS CLES : PAMINA, INCERTITUDE, SENSIBILITE, METAMODELE, SURFACE DE REPONSE, KRIGEAGE, PROCESSUS GAUSSIEN

(04/02 - JA-Document1)

	Rédacteur	Vérificateur	Vérificateur Qualité	Approbateur
Fonction			IQ du SESI	Le Chef du SESI/LCFR
visa				
NOM	Bertrand IOOSS	Michel MARQUES	Florence JOYER	Frédéric BERTRAND
Date	28/03/08	28/3/08	3/04/08	28/03/08

Propriété Industrielle	Cadre de réalisation	Classification					Qualité
		DR	CC	CD	SD	Ss	
En l'absence d'accord ou de contrat, la diffusion des informations contenues dans ce document auprès d'un organisme tiers extérieur au CEA est soumise à l'accord du Chef de Département.	DOB.................................DSOE Domaine .... Simulation Projet...............................SUPAR Contrat.............EUR 656 036 404					x	Document réalisé sous SMQ certifié AFAQ — ISO 9001

Indice	OBJET DES REVISIONS	DATE	RÉFÉRENCE GCAO
0	Emission initiale	28/03/08	CEA/DEN/CAD/DER/SESI/LCFR/NT DO 7 21/03/08

1. INTRODUCTION

This report consists in deliverable CEA/DEN/DER for the component RTDC 2 of European project PAMINA (Performance Assessment Methodologies IN Application to guide the development of the safety case) 6th FP. This task concerns the presentation of new methods to perform sensitivity analysis for cpu time consuming computer codes. In this report, one restricts to methodological aspects. We describe a recent technique based on the use of a metamodel, i.e. a cpu time inexpensive mathematical function fitted and validated on a few simulations of the computer code. We show how to fit and use one of the most popular metamodels: the Gaussian process model which extends the kriging principles of geostatistics to numerical experiments. Its analytical formulation allows to derive analytical formulas for the sensitivity indices without running other simulations of the computer code.

The chapter 1 describes an efficient algorithm for modelling complex computer codes with Gaussian processes. Indeed, when the number or random inputs is large (> 10), non parametric regression techniques are difficult to apply. The described methodology proposes a new variable selection technique while fitting the metamodel. This allows to obtain more predictive metamodels.

The chapter 2 shows how to compute variance-based sensitivity indices (that we call Sobol indices) with the Gaussian process model. Contrary to classical sensitivity indices (derivatives, Pearson and Spearman correlation coefficients, standardized regression coefficients, etc), the Sobol indices are valid without any linearity, monotonicity or regularity assumptions of the underlying numerical model. Computations of Sobol indices via a metamodel are ordinarily done by simple Monte-Carlo algorithms. We explain two different analytical ways of Sobol indices computations with the Gaussian process metamodel.

2 AN EFFICIENT METHODOLOGY FOR MODELING COMPLEX COMPUTER CODES WITH GAUSSIAN PROCESSES

2.1 ABSTRACT

Complex computer codes are often too time expensive to be directly used to perform uncertainty propagation studies, global sensitivity analysis or to solve optimization problems. A well known and widely used method to circumvent this inconvenience consists in replacing the complex computer code by a reduced model, called a metamodel, or a response surface that represents the computer code and requires acceptable calculation time. One particular class of metamodels is studied : the Gaussian process model that is characterized by its mean and covariance functions. A specific estimation procedure is developed to adjust a Gaussian process model in complex cases (non linear relations, highly dispersed or discontinuous output, high dimensional input, inadequate sampling designs, etc.). The efficiency of this algorithm is compared to the efficiency of other existing algorithms on an analytical test case. The proposed methodology is also illustrated for the case of a complex hydrogeological computer code, simulating radionuclide transport in groundwater.

2.2 INTRODUCTION

With the advent of computing technology and numerical methods, investigation of computer code experiments remains an important challenge. Complex computer models calculate several output values (scalars or functions) which can depend on a high number of input parameters and physical variables. These computer models are used to make simulations as well as predictions or sensitivity studies. Importance measures of each uncertain input variable on the response variability provide guidance to a better understanding of the modeling in order to reduce the response uncertainties most effectively (Saltelli et al. [25], Kleijnen [15], Helton et al. [11]).

However, complex computer codes are often too time expensive to be directly used to conduct uncertainty propagation studies or global sensitivity analysis based on Monte Carlo methods. To avoid the problem of huge calculation time, it can be useful to replace the complex computer code by a mathematical approximation, called a response surface or a surrogate model or also a metamodel. The response surface method (Box & Draper [5]) consists in constructing a function that simulates the behavior of real phenomena in the variation range of the influential parameters, starting from a certain number of experiments. Similarly to this theory, some methods have been developed to build surrogates for long running computer codes (Sacks et al. [24], Osio & Amon [22], Kleijnen & Sargent[17], Fang et al. [9]). Several metamodels are classically used : polynomials, splines, generalized linear models, or learning statistical models such as neural networks, support vector machines, ...(Hastie et al. [10], Fang et al. [9]).

For sensitivity analysis and uncertainty propagation, it would be useful to obtain an analytic predictor formula for a metamodel. Indeed, an analytical formula often allows the direct calculation of sensitivity indices or output uncertainties. Moreover, engineers and physicists prefer interpretable models that give some understanding of the simulated physical phenomena and parameter interactions. Some metamodels, such as polynomials (Jourdan & Zabalza-Mezghani [14], Kleijnen [16], Iooss et al. [13]), are easily interpretable but not always very efficient. Others, for instance neural networks (Alam et al. [3], Fang et al. [9]), are more efficient but do not provide an analytic predictor formula.

The kriging method (Matheron [19], Cressie [7]) has been developed for spatial interpolation problems ; it takes into account spatial statistical structure of the estimated variable. Sacks et al. [24] have extended the kriging principles to computer experiments by considering the correlation between two responses of a computer code depending on the distance between input variables. The kriging model (also called Gaussian process model), characterized by its mean and covariance functions, presents several advantages, especially the interpolation and interpretability properties. Moreover, numerous authors (for example, Currin et al. [8], Santner et al. [26] and Vazquez et al. [28]) show that this model can provide a statistical framework to compute an efficient predictor of code response.

From a practical standpoint, constructing a Gaussian process model implies estimation of several hyperparameters included in the covariance function. This optimization problem is particularly difficult for a model with many inputs and inadequate sampling designs (Fang et al. [9], O'Hagan [21]). In this paper, a special estimation procedure is developed to fit a Gaussian process model in complex cases (non linear relations, highly dispersed output, high dimensional input, inadequate sampling designs). Our purpose includes developing a procedure for parameter estimation via an essential step of input parameter selection. Note that we do not deal with the design of experiments in computer code simulations (i.e. choosing values of input parameters). Indeed, we work on data obtained in a previous study (the hydrogeological model of Volkova et al. [29]) and try to adapt a Gaussian process model as well as possible to a non-optimal sampling design. In summary, this study presents two main objectives : developing a methodology to implement and adapt a Gaussian process model to complex data while studying its prediction capabilities.

The next section briefly explains the Gaussian process modeling from theoretical expression to predictor formulation and model parameterization. In section 3, a parameter estimation procedure is introduced from the numerical standpoint and a global methodology of Gaussian process modeling implementation is presented. Section 4 is devoted to applications. First, the algorithm efficiency is compared to other algorithms for the example of an analytical test case. Secondly, the algorithm is applied to the data set (20 inputs and 20 outputs) coming from a hydrogeological transport model based on waterflow and diffusion dispersion equations. The last section provides some possible extensions and concluding remarks.

2.3 GAUSSIAN PROCESS MODELING

2.3.1 Theoretical model

Let us consider n realizations of a computer code. Each realization $y(x)$ of the computer code output corresponds to a d-dimensional input vector $x = (x_1, ..., x_d)$. The n points corresponding to the code runs are called an experimental design and are denoted as $X_s = (x^{(1)}, ..., x^{(n)})$. The outputs will be denoted as $Y_s = (y^{(1)}, ..., y^{(n)})$ with $y^{(i)} = y(x^{(i)}), i = 1, ..., n$. Gaussian process (Gp) modeling treats the deterministic response $y(x)$ as a realization of a random function $Y(x)$, including a regression part and a centered stochastic process. This model can be written as :

$$Y(x) = f(x) + Z(x). \tag{1}$$

The deterministic function $f(x)$ provides the mean approximation of the computer code. Our study is limited to the parametric case where the function f is a linear combination of elementary functions. Under this assumption, $f(x)$ can be written as follows :

$$f(x) = \sum_{j=0}^{k} \beta_j f_j(x) = F(x)\beta,$$

where $\beta = [\beta_0, \ldots, \beta_k]^t$ is the regression parameter vector and $F(x) = [f_0(x), \ldots, f_k(x)]$ is the regression matrix, with each f_j ($j = 0, \ldots, k$) an elementary function. In the case of the one-degree polynomial regression, $(d+1)$ elementary functions are used :

$$\begin{cases} f_0(x) = 1, \\ f_i(x) = x_i \text{ for } i = 1, \ldots, d. \end{cases}$$

In the following, we use this one-degree polynomial for the regression part, while our methodology can be extended to other bases of regression functions. The regression part allows the addition of an external drift. Without prior information on the relation between the model output and the input variables, this quite simple choice appears reasonable. Indeed, adding this simple external drift allows for a nonstationary global model even if the stochastic part Z is a stationary process. Moreover, on our tests of section 2.5, this simple model does not affect our prediction performance. This simplification is also reported by Sacks et al. [24].

The stochastic part $Z(x)$ is a Gaussian centered process fully characterized by its covariance function : Cov$(Z(\boldsymbol{x}), Z(\boldsymbol{u})) = \sigma^2 R(\boldsymbol{x}, \boldsymbol{u})$, where σ^2 denotes the variance of Z and R is the correlation function that provides interpolation and spatial correlation properties. To simplify, a stationary process $Z(\boldsymbol{x})$ is considered, which means that correlation between $Z(\boldsymbol{x})$ and $Z(\boldsymbol{u})$ is a function of the difference between $\boldsymbol{x}$ and $\boldsymbol{u}$. Our study is focused on a particular family of correlation functions that can be written as a product of one-dimensional correlation functions :

$$\text{Cov}(Z(\boldsymbol{x}), Z(\boldsymbol{u})) = \sigma^2 R(\boldsymbol{x} - \boldsymbol{u}) = \sigma^2 \prod_{l=1}^{d} R_l(x_l - u_l).$$

Abrahamsen [2], Sacks et al. [24], Chilès & Delfiner [6] and Rasmussen & Williams [23] give lists of correlation functions with their advantages and drawbacks. Among all these functions, we choose to use the generalized exponential correlation function :

$$R_{\boldsymbol{\theta},\boldsymbol{p}}(\boldsymbol{x} - \boldsymbol{u}) = \prod_{l=1}^{d} \exp(-\theta_l |x_l - u_l|^{p_l}) \text{ with } \theta_l \geq 0 \text{ and } 0 < p_l \leq 2,$$

where $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_d]^t$ and $\boldsymbol{p} = [p_1, \ldots, p_d]^t$ are the correlation parameters. Our motivations stand on the derivation and regularity properties of this function. Moreover, different choices of covariance parameters allow a wide spectrum of possible shapes (Figure 2.1); $p = 1$ gives the exponential correlation function and $p = 2$ the Gaussian correlation function.

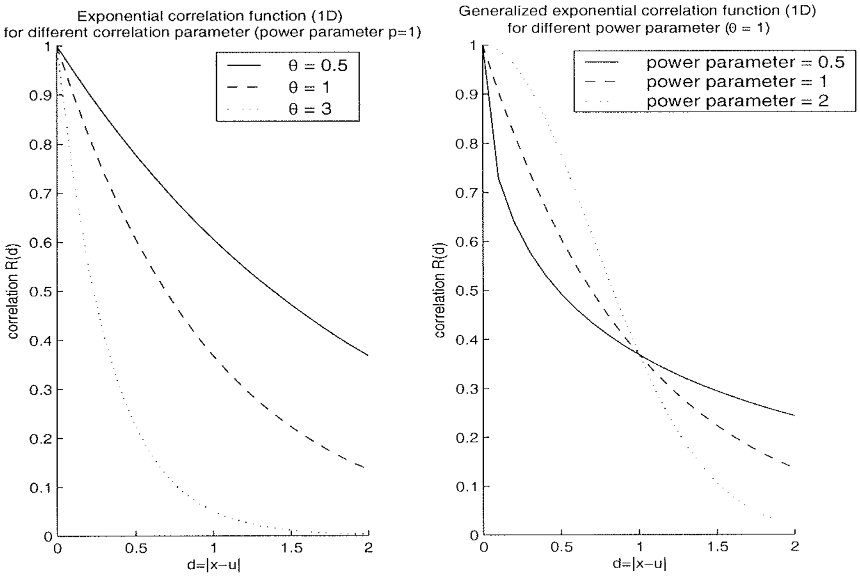

Fig. 2.1 – **Generalized exponential correlation function for different power and correlation parameters.**

Even for deterministic computational codes (i.e. outputs corresponding to the same inputs are identical), the outputs may be subject to noise (e.g. numerical noise). In this case, an independent white noise $U(\boldsymbol{x})$ is added in the stochastic part of the model :

$$Y(\boldsymbol{x}) = f(\boldsymbol{x}) + Z(\boldsymbol{x}) + U(\boldsymbol{x}), \tag{2}$$

where $U(\boldsymbol{x})$ is a centered Gaussian variable with variance $\varepsilon^2 = \sigma^2\tau$. In terms of covariance function, this white noise introduces a discontinuity at the origin called the nugget effect (Matheron [19]) :

$$\mathrm{Cov}(Y(\boldsymbol{x}), Y(\boldsymbol{u})) = \sigma^2 \left(R_{\boldsymbol{\theta},\boldsymbol{p}}(\boldsymbol{x}-\boldsymbol{u}) + \tau\delta(\boldsymbol{x}-\boldsymbol{u}) \right),$$

where $\delta(\boldsymbol{v}) = \begin{cases} 1 \text{ if } \boldsymbol{v} = 0, \\ 0 \text{ otherwise.} \end{cases}$

2.3.2 Joint and conditional distributions

Under the hypothesis of a Gp model, the learning sample Y_s follows the multivariate normal distribution

$$p(Y_s|X_s, \boldsymbol{\beta}, \sigma, \boldsymbol{\theta}, \boldsymbol{p}, \tau) = \mathcal{N}\left(F_s\boldsymbol{\beta}, \boldsymbol{\Sigma}_s\right),$$

where $F_s = [F(\boldsymbol{x}^{(1)})^t, \ldots, F(\boldsymbol{x}^{(n)})^t]^t$ is the regression matrix and

$$\boldsymbol{\Sigma}_s = \sigma^2 \left(R_{\boldsymbol{\theta},\boldsymbol{p}}\left(\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right)_{i,j=1\ldots n} + \tau \boldsymbol{I}_n \right)$$

is the covariance matrix with $\boldsymbol{I}_n$ the n-dimensional identity matrix.

If a new point $\boldsymbol{x}^* = (x_1^*, \ldots, x_d^*)$ is considered, the joint probability distribution of $(Y_s, Y(\boldsymbol{x}^*))$ is :

$$p(Y_s, Y(\boldsymbol{x}^*)|X_s, \boldsymbol{x}^*, \boldsymbol{\beta}, \sigma, \boldsymbol{\theta}, \boldsymbol{p}, \tau) = \mathcal{N}\left(\begin{bmatrix} F_s \\ F(\boldsymbol{x}^*) \end{bmatrix} \boldsymbol{\beta}, \begin{bmatrix} \boldsymbol{\Sigma}_s & \boldsymbol{k}(\boldsymbol{x}^*) \\ \boldsymbol{k}(\boldsymbol{x}^*)^t & \sigma^2(1+\tau) \end{bmatrix} \right), \tag{3}$$

with

$$\begin{aligned} \boldsymbol{k}(\boldsymbol{x}^*) &= (\, \mathrm{Cov}(y^{(1)}, Y(\boldsymbol{x}^*)), \ldots, \mathrm{Cov}(y^{(n)}, Y(\boldsymbol{x}^*))\,)^t \\ &= \sigma^2(\, R_{\boldsymbol{\theta},\boldsymbol{p}}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^*) + \tau\delta(\boldsymbol{x}^{(1)}, \boldsymbol{x}^*), \ldots, R_{\boldsymbol{\theta},\boldsymbol{p}}(\boldsymbol{x}^{(n)}, \boldsymbol{x}^*) + \tau\delta(\boldsymbol{x}^{(n)}, \boldsymbol{x}^*)\,)^t. \end{aligned} \tag{4}$$

By conditioning this joint distribution on the learning sample, we can readily obtain the conditional distribution of $Y(\boldsymbol{x}^*)$ which is Gaussian (von Mises [30]) :

$$\begin{aligned} & p(Y(\boldsymbol{x}^*)|Y_s, X_s, \boldsymbol{x}^*, \boldsymbol{\beta}, \sigma, \boldsymbol{\theta}, \boldsymbol{p}, \tau) \\ & = \mathcal{N}\left(\mathbb{E}[Y(\boldsymbol{x}^*)|Y_s, X_s, \boldsymbol{x}^*, \boldsymbol{\beta}, \sigma, \boldsymbol{\theta}, \boldsymbol{p}, \tau], \mathrm{Var}[Y(\boldsymbol{x}^*)|Y_s, X_s, \boldsymbol{x}^*, \boldsymbol{\beta}, \sigma, \boldsymbol{\theta}, \boldsymbol{p}, \tau]\right), \end{aligned} \tag{5}$$

with

$$\begin{cases} \mathbb{E}[Y(\boldsymbol{x}^*)|Y_s, X_s, \boldsymbol{x}^*, \boldsymbol{\beta}, \sigma, \boldsymbol{\theta}, \boldsymbol{p}, \tau] = F(\boldsymbol{x}^*)\boldsymbol{\beta} + \boldsymbol{k}(\boldsymbol{x}^*)^t \boldsymbol{\Sigma}_s^{-1}(Y_s - F_s\boldsymbol{\beta}), \\ \mathrm{Var}[Y(\boldsymbol{x}^*)|Y_s, X_s, \boldsymbol{x}^*, \boldsymbol{\beta}, \sigma, \boldsymbol{\theta}, \boldsymbol{p}, \tau] = \sigma^2(1+\tau) - \boldsymbol{k}(\boldsymbol{x}^*)^t \boldsymbol{\Sigma}_s^{-1} \boldsymbol{k}(\boldsymbol{x}^*). \end{cases} \tag{6}$$

The conditional mean (equation (6)) is used as a predictor. The variance formula corresponds to the mean squared error (MSE) of this predictor and is also known as the kriging variance. This analytical formula for MSE gives a local indicator of the prediction accuracy. More generally, Gp model provides an analytical formula for the distribution of the output variable at an arbitrary new point. This distribution formula can be used for sensitivity and uncertainty analysis, as well as for quantile evaluation (O'Hagan [21]). Its use can be completely or partly analytical and avoids costly methods based for example on a Monte Carlo algorithm. The variance expression can also be used in sampling strategies (Scheidt & Zabalza-Mezghani [27]). All these considerations and possible extensions of Gp modeling represent significant advantages (Currin et al. [8], Rasmussen & Williams [23]).

2.3.3 Parameter estimation

To compute the mean and variance of a Gp model, estimation of several parameters is needed. Indeed, the Gp model (2) is characterized by the regression parameter vector β, the correlation parameters $(\boldsymbol{\theta}, \boldsymbol{p})$ and the variance parameters (σ^2, τ). The maximum likelihood method is commonly used to estimate these parameters. Given a Gp model, the log-likelihood of Y_s can be written as :

$$\begin{aligned} l_{Y_s}(\beta, \boldsymbol{\theta}, \boldsymbol{p}, \sigma, \tau) = & -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln(\det(\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}} + \tau \boldsymbol{I}_n)) \\ & -\frac{1}{2\sigma^2}(Y_s - F_s\beta)^t(\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}} + \tau \boldsymbol{I}_n)^{-1}(Y_s - F_s\beta). \end{aligned}$$

Given the correlation parameters $(\boldsymbol{\theta}, \boldsymbol{p})$ and the variance parameter τ, the maximum likelihood estimator of β is the generalized least squares estimator :

$$\hat{\beta} = ({F_s}^t(\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}} + \tau \boldsymbol{I}_n)^{-1}F_s)^{-1} \; {F_s}^t(\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}} + \tau \boldsymbol{I}_n)^{-1}Y_s, \tag{7}$$

and the maximum likelihood estimator of σ^2 is :

$$\widehat{\sigma^2} = \frac{1}{n}(Y_s - F_s\hat{\beta})^t(\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}} + \tau \boldsymbol{I}_n)^{-1}(Y_s - F_s\hat{\beta}). \tag{8}$$

Remark 2.1 *If we consider the predictor built on the conditional mean (equation (6)), we replace β by its estimator $\widehat{\beta}$. The predictor writes now*

$$\widehat{Y(\boldsymbol{x}^*)}_{|Y_s, X_s, \boldsymbol{x}^*, \sigma, \boldsymbol{\theta}, \boldsymbol{p}, \tau} = F(\boldsymbol{x}^*)\widehat{\beta} + \boldsymbol{k}(\boldsymbol{x}^*)^t\Sigma_s^{-1}(Y_s - F_s\widehat{\beta})$$

and its MSE has consequently an additional component (Santner et al. [26]) :

$$Var[\widehat{Y(\boldsymbol{x}^*)}|Y_s, X_s, \boldsymbol{x}^*, \sigma, \boldsymbol{\theta}, \boldsymbol{p}, \tau] = \sigma^2(1+\tau) - \boldsymbol{k}(\boldsymbol{x}^*)^t\Sigma_s^{-1}\boldsymbol{k}(\boldsymbol{x}^*) + u(\boldsymbol{x}^*)({F_s}^t\Sigma_s^{-1}F_s)^{-1}u(\boldsymbol{x}^*)^t$$

with $u(\boldsymbol{x}^*) = F(\boldsymbol{x}^*) - \boldsymbol{k}(\boldsymbol{x}^*)^t\Sigma_s^{-1}F_s$.

Matrix $\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}}$ depends on $\boldsymbol{\theta}$ and $\boldsymbol{p}$. Consequently, $\hat{\beta}$ and $\widehat{\sigma^2}$ depend on $\boldsymbol{\theta}$, $\boldsymbol{p}$ and τ. Substituting $\hat{\beta}$ and $\widehat{\sigma^2}$ into the log-likelihood, we obtain the optimal choice $(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{p}}, \widehat{\tau})$ which maximizes :

$$\phi(\boldsymbol{\theta}, \boldsymbol{p}, \tau) = -\frac{1}{2}\left[n\ln(\widehat{\sigma^2}) + \ln(|\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}} + \tau \boldsymbol{I}_n|)\right] \text{ where } |\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}} + \tau \boldsymbol{I}_n| = \det(\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}} + \tau \boldsymbol{I}_n).$$

Thus, estimation of $(\boldsymbol{\theta}, \boldsymbol{p})$ and τ consists in numerical optimization of the function ψ defined as follows :

$$(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{p}}, \widehat{\tau}) = \underset{\boldsymbol{\theta},\boldsymbol{p},\tau}{\arg\min}\, \psi(\boldsymbol{\theta}, \boldsymbol{p}, \tau) \text{ with } \psi(\boldsymbol{\theta}, \boldsymbol{p}, \tau) = |\boldsymbol{R}_{\boldsymbol{\theta},\boldsymbol{p}} + \tau \boldsymbol{I}_n|^{\frac{1}{n}}\, \widehat{\sigma^2}.$$

Our study is focused on complex cases with large dimensions d for the input vector x ($d = 20$ in our second example in section 2.5), where the sampling design has not been chosen as a uniform grid. In this setting, minimizing function $\psi(\boldsymbol{\theta}, \boldsymbol{p}, \tau)$ is an optimization problem that is numerically costly and hard to solve. Several difficulties guide the choice of the algorithm. First, a large number of parameters imposes the use of a sequential algorithm, where different parameters are introduced step by step. Second, a large parameter domain due to the number of parameters and the lack of prior bounds requires an exploratory algorithm able to explore the domain in an optimal way. Finally, the observed irregularities of $\psi(\boldsymbol{\theta}, \boldsymbol{p}, \tau)$ due, for instance, to a conditioning problem induce local minima, which recommend the use of a stochastic algorithm rather than a descent algorithm.

Several algorithms have been proposed in previous papers. Welch et al. [31] use the simplex search method and introduce a kind of forward selection algorithm in which correlation parameters are added step by step to reduce function $\psi(\boldsymbol{\theta}, \boldsymbol{p}, \tau)$. In Kennedy and O'Hagan's GEM-SA software (O'Hagan [21]), which uses the Bayesian formalism, the posterior distribution of hyperparameters is maximized via a conjugate gradient method (the Powel method is used as the numerical recipe). The DACE Matlab free toolbox (Lophaven et al. [18]) introduces a powerful stochastic algorithm based on the Hooke & Jeeves method (Bazaraa et al. [4]), which unfortunately requires a starting point and some bounds to constrain the optimization. In complex applications, Welch's algorithm reveals some limitations and for high dimensional input, GEM-SA and DACE software cannot be applied directly on data including all the input variables. To solve this problem, we propose a sequential version (inspired by Welch's algorithm) of the DACE algorithm. It is based on the step by step inclusion of input variables (previously sorted). Our methodology allows progressive parameter estimation by input variables selection both in the regression part and in the covariance function. The complete description of this methodology is the subject of the next section.

Remark 2.2 *One of the problems we have to acknowledge in the evaluation of $\psi(\boldsymbol{\theta}, \mathbf{p}, \tau)$ is the condition number of the prior covariance matrix. This condition number affects the numerical stability of the linear system for the $\hat{\beta}$ determination and for the evaluation of the determinant. The degree of ill-conditioning not only depends on sampling design but is also sensitive to the underlying covariance model. Ababou et al. [1] showed, for example, that a Gaussian covariance ($p = 2$) implies an ill-conditioned covariance matrix (which leads to a numerically unstable system), while an exponential covariance ($p = 1$) gives more stability. Moreover, in our case, the experimental design cannot be chosen and numerical parameter estimation is often damaged by the ill-conditioning problem. The nugget effect represented by τ solves this problem. Although the outputs of the learning sample are no longer interpolated, this nugget effect improves the correlation matrix condition number and increases robustness of our estimation algorithm.*

2.4 MODELING METHODOLOGY

Let us first detail the procedure used to validate our model. Since the Gp predictor is an exact interpolator (except when a nugget effect is included), residuals of the learning data cannot be used directly. So, to estimate the mean squared error in a non-optimistic way, we use either a K-fold cross validation procedure (Hastie et al. [10]) or a test sample (consisting of new data, unused in the building process of the Gp model). In both cases, the predictivity coefficient Q_2 is computed. Q_2 corresponds to the classical coefficient of determination R^2 for a test sample, i.e. for prediction residuals :

$$Q_2(Y,\hat{Y}) = 1 - \frac{\sum_{i=1}^{n_{test}} \left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n_{test}} \left(\bar{Y} - Y_i\right)^2},$$

where Y denotes the n_{test} observations of the test set and $\bar{Y}$ is their empirical mean. $\hat{Y}$ represents the Gp model predicted values, i.e. the conditional mean (equation (6)) computed which the estimated values of parameters $(\widehat{\beta}, \widehat{\sigma}, \widehat{\boldsymbol{\theta}}, \widehat{\mathbf{p}}, \widehat{\tau})$. Other simple validation criteria can be used : the absolute error, the mean and standard deviation of the relative residuals, ... (see, for example, Kleijnen & Sargent [17]), which are all global measures. Some statistical and graphical analyses of residuals can provide more detailed diagnostics.

Our methodology consists in seven successive steps. A formal algorithmic definition is specified for each step. For $i = 1, \ldots, d$, let e_i denote the i^{th} input variable. $\mathcal{M}_0 = \left\{e_1^{(0)}, \ldots, e_d^{(0)}\right\}$ denotes the complete initial model (i.e. all the inputs in their initial ranking). $\mathcal{M}_1 = \left\{e_1^{(1)}, \ldots, e_d^{(1)}\right\}$ and $\mathcal{M}_2 = \left\{e_1^{(2)}, \ldots, e_d^{(2)}\right\}$ refer to the inputs in new rankings after sorting by different criteria (correlation coefficient or variation of Q_2). Finally, $\mathcal{M}_{cov}$ and $\mathcal{M}_{reg}$ denote the current covariance model and the current regression model; i.e. the list of selected inputs appearing in the covariance and regression functions.

Step 0 - Standardization of input variables

The appropriate procedure to construct a metamodel requires space filling designs with good optimality and orthogonality properties (Fang et al. [9]). However, we are not always able to choose the experimental design, especially in industrial studies when the data have been generated a long time ago. Furthermore, other restrictions can be imposed ; for example, a sampling design taking into account the prior distribution of input variables. This can have prejudicial consequences for hyperparameter estimation and metamodel quality.

So, to increase the robustness of our parameter estimation algorithm and to optimize the metamodel quality, we recommend to transform all the inputs into uniform variables. In order to get each transformed input variable following an uniform distribution $\mathcal{U}[0,1]$, the theoretical distribution (if known) or the empirical ones after a piecewise linear approximation is applied to the original inputs. This approximation is required to avoid transforming a future unsampled x^* to one of the transformed training sites, even if no element of x^* is equal to the corresponding element of any of the untransformed training sites. We empirically observed that this uniform transformation of the inputs seems well adapted to correctly estimate correlation parameters. Choices of bounds and starting points are also simplified by this standardization.

Step 1 - Initial input variables ranking by decreasing coefficient of correlation between e_i and Y

Sorting input variables is necessary to reduce the number of possible models, especially to dissociate regression and covariance models. Furthermore, direct estimation of all parameters without an efficient starting point gives bad results. So, as a sort criterion, we choose the coefficient of correlation between the input variable and the output variable under consideration. The correlation coefficients between the input parameters and the output variable are the simplest measures of the influence of inputs on the output (Saltelli et al. [25]). They are valid in the linear relation context, while in the nonlinear context, they give a first idea of the hierarchy among input variables, in terms of their influence on the output. Finally, this simple and intuitive choice does not require any modeling and appears a good initial method to sort the inputs when no other information is available.

For a strongly nonlinear computer code, it could be interesting to use a qualitative method, independent of the model complexity, in order to sort the inputs by influence order (Helton et al. [11]). Another possibility would be to fit an initial Gp model with a regression part limited to an intercept and all components of p equal to 1 or 2. Only the correlation coefficients vector θ has to be estimated. Then, sensitivity measures such as the Sobol indices (Saltelli et al. [25], Volkova et al. [29]) are computed and used to sort the inputs by influence order.

Algorithm

$$\mathcal{M}_0 = \left\{e_1^{(0)}, \ldots, e_d^{(0)}\right\} \Longrightarrow \mathcal{M}_1 = \left\{e_1^{(1)}, \ldots, e_d^{(1)}\right\}$$

$$\begin{cases} \mathcal{M}_{reg} = \mathcal{M}_1 \\ \mathcal{M}_{cov} = \mathcal{M}_1 \end{cases}$$

Step 2 - Initialization of the correlation parameter bounds and starting points for the estimation procedure

To constrain the ψ optimization, the DACE estimation procedure requires three following values for each correlation parameter : a lower bound, an upper bound and a starting point. These values are crucial for the success of the estimation algorithm, when it is used directly for all the input variables. However, using sequential estimation based on progressive introduction of input variables, we limit the problems associated with these three values, especially with the starting point value. Another way to reduce the importance of starting point and bounds is to increase the number of iterations in DACE estimation algorithm. However, in the case of a high number of inputs, increasing the number of iterations in DACE can become extremely time expensive ; a compromise has to be found. As the input variables have been previously transformed into standardized uniform variables, the initialization and the bounds of the correlation parameters can be the same for all the inputs :

- ◇ lower bounds for each component of $\boldsymbol{\theta}$ and $\boldsymbol{p}$: $lob_{\boldsymbol{\theta}} = 10^{-8}$, $lob_{\boldsymbol{p}} = 0$,
- ◇ upper bounds for each component of $\boldsymbol{\theta}$ and $\boldsymbol{p}$: $upb_{\boldsymbol{\theta}} = 100$, $upb_{\boldsymbol{p}} = 2$,
- ◇ starting points for estimation of each component of $\boldsymbol{\theta}$ and $\boldsymbol{p}$: $\theta^0 = 0.5$, $p^0 = 1$.

Step 3 - Successive inclusion of input variables in the covariance function

For each set of inputs included in the covariance function, all the inputs from the ordered set in the regression function are evaluated. Correlation and regression parameters are estimated by the DACE modified algorithm, with the values, estimated at the $(i-1)^{th}$ step for the same regression model, used as a starting point. More precisely, at step i, input variables numbered from 1 to i are included in the covariance function and the algorithm estimates pairs of the correlation parameters (θ_l, p_l) for $l = 1, \ldots, i$. As the starting point, the algorithm uses correlation parameters obtained at the $(i-1)^{th}$ step for the starting values of $((\theta_1, p_1), \ldots, (\theta_{i-1}, p_{i-1}))$. First starting value of (θ_i, p_i) is fixed to an arbitrary reference value. Then, at each step, selection of variables in the regression part is also made.

Hoeting et al. [12] recommends the corrected Akaike information criterion (AICC) for input selection in the regression model in order to take spatial correlations into account. Therefore, after the estimation of correlation and regression parameters, the AICC is computed :

$$\text{AICC} = -2l_{Y_s}\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}, \hat{\sigma}\right) + 2n\frac{m_1 + m_2 + 1}{n - m_1 - m_2 - 2},$$

where m_1 denotes the number of input variables in the regression function, m_2 those in the covariance function and l_Y the log-likelihood of the sample Y. The required model is the one minimizing this criterion.

Algorithm

```
For $i = 1 \ldots d$
  ◇ Step 3.1 : Variables in covariance function
    $\mathcal{M}_{i,cov} = \mathcal{M}_{cov}(1, \ldots, i)$
  ◇ Step 3.2 : Successive inclusion of input variables in regression function
    For $j = 1 \ldots d$
    – Regression Model :
      $\mathcal{M}_{j,reg} = \mathcal{M}_{reg}(1, \ldots, j)$
    – Parameter estimation :
      $\boldsymbol{\theta}^{init} = ({\theta_1}^{(i-1),j}, \ldots, {\theta_{i-1}}^{(i-1),j}, \theta^0)^t$
      $\boldsymbol{p}^{init} = ({p_1}^{(i-1),j}, \ldots, {p_{i-1}}^{(i-1),j}, p^0)^t$
      $[\theta^{i,j}, p^{i,j}] = \text{DACE estimation}(\mathcal{M}_{i,cov}, \mathcal{M}_{j,reg}, [\boldsymbol{\theta}^{init}, \boldsymbol{p}^{init}], [lob_{\boldsymbol{\theta}}, lob_{\boldsymbol{p}}], [upb_{\boldsymbol{\theta}}, upb_{\boldsymbol{p}}])$
    – AICC Criterion computation
      AICC$(i,j)$ = AICC$(\mathcal{M}_{i,cov}, \mathcal{M}_{j,reg})$
    End
  ◇ Step 3.3 : Optimal regression model selection :
    $j^{optim}(i) = \arg\min_j$ (AICC$(i,j)$)
  ◇ Step 3.4 : $Q_2$ evaluation by $K$-fold cross validation or on test data (with current correlation
    model and optimal regression model)
    $Q_2(i) = Q_2(\mathcal{M}_{i,cov}, \mathcal{M}_{j^{optim}(i),reg})$
End
```

This order (correlation outer, regression inner) can be justified by minimizing the computer time required for optimization. The selection procedure for the regression part is made by the minimization of AICC criterion which requires, at each step, only one parameter estimation. On the other hand, the covariance selection is made by the maximization of Q_2 which is often computed by a K-fold cross validation. This procedure requires, at each step, K estimation procedures. So, the loop on covariance selection is the more expensive, and consequently has to be outer. The choice of K depends on the number of observations of the data set, on the constraints in term of computer time and on the influence of the learning sample size on prediction quality. If few data are available, a leave-one-out cross-validation could be preferred to a K-fold procedure to avoid an undesirably negative effect of small learning sets on prediction quality.

Remark 2.3 *To avoid some biases on the choice of the optimal covariance model in the next two steps, the coefficient Q_2 has to be computed on a test sample (or by a cross validation procedure), different from the one used for the final validation of the Gp model at step 7.*

Other criteria often used in the optimization of the computer experiment designs (Sacks et al. [24], Santner et al. [26]) could be considered to select the optimal regression and covariance model. These criteria are based on the variance of Gp model : they produce a model that minimizes the maximum or the integral of predictive variance over input space. However, in the case of a high number of inputs, the optimization of these criteria can be very computer time expensive. The advantage of the Q_2 statistic is its relatively fast evaluation, while producing a final model that optimizes the predictive performance.

Step 4 (optional) - New input variables ranking in the covariance function based on the evolution of Q_2 (inputs sorted by decreasing "jumps" of Q_2)

This optional step improves the selection of inputs, particularly in the covariance function. For each input X_i, the increase of the Q_2 coefficient (denoted $\Delta Q_2(i)$) is computed when this i^{th} variable is added to the covariance function. This value is an indicator of the contribution of the i^{th} input to the accuracy of the Gp model. For this reason, it can be judicious to use values $\Delta Q_2(1), \dots, \Delta Q_2(d)$ to sort the inputs included in the correlation function. The inputs are sorted by decreasing of values $\Delta Q_2(i)$ and the procedure of parameter estimation is repeated with this new ranking of inputs for the covariance function (step 3 is rerun).

Algorithm

```
– Evaluation of $Q_2$ increase for each input variable included in the covariance function :
  $\Delta Q_2(k) = Q_2(1)$
  For $k = 2 \dots d$
      $\Delta Q_2(k) = Q_2(k) - Q_2(k-1)$
  end
– Sorting input variables by decreasing of $\Delta Q_2$
  $\mathcal{M}_1 \Longrightarrow \mathcal{M}_2$
```

Step 5 (optional) - Algorithm for parameter estimation with new ranking of input variables in the covariance function

This optional step improves the selection of inputs, particularly in the covariance function. The procedure of parameter estimation (step 3) is repeated with the inputs sorted by decreasing values of $\Delta Q_2(i)$ in the covariance function. Consequently, correlation parameters related to the inputs that are the most influential for the increase of the Gp model accuracy are estimated in the first place. Furthermore, we can also hope that the use of this new ranking allows a decrease in the number of inputs included in the covariance function and an optimal input selection. The use of this new ranking appears more judicious and justifiable for the covariance function than sorting by decreasing correlation coefficient (cf. step 1). However, the ranking of step 1 is kept for the regression function.

Algorithm

$$\begin{cases} \mathcal{M}_{reg} = \mathcal{M}_1 \\ \mathcal{M}_{cov} = \mathcal{M}_2 \end{cases}$$

Step 6 - Optimal covariance model selection

For each set of inputs in the covariance function, the optimal regression model is selected based on minimization of the AICC criterion (cf. step 3.3). Then, the predictivity coefficient Q_2 is computed either by cross validation or on a test sample (cf. step 3.4). Finally, the selected covariance model is the one corresponding to the highest Q_2 value.

Algorithm

$$i^{optim} = \arg\max_i \left(Q_2(i)\right)$$

$$\begin{cases} \mathcal{M}_{cov}^{\text{optim}} = \mathcal{M}_{cov}(1,\ldots,i^{optim}) \\ \mathcal{M}_{reg}^{\text{optim}} = \mathcal{M}_{reg}(1,\ldots,j^{optim}(i^{optim})) \end{cases}$$

Step 7 - Final validation of the optimal Gp model

After building and selecting the optimal Gp model, a final validation is necessary to evaluate the predictive performance and to eventually compare it to other metamodels. To do this, coefficient Q_2 is evaluated on a new test sample (i.e. data not used in the building procedure). If only few data are available, a cross validation procedure can be considered. So, two cross validation procedures are overlapped; one for building the model and one for its validation.

Algorithm

$$Q_2^{final} = Q_2(\mathcal{M}_{cov}^{\text{optim}}, \mathcal{M}_{reg}^{\text{optim}})$$

After all the steps of our algorithm (including the step 5), we can often link the inputs appearing in the covariance and regression functions with the nature of their effects on the output. Indeed, we can generally observed 4 cases : the inputs with only a linear effect which are supposed to appear only in the regression and excluded from the covariance with the step 5, the inputs with only a non-linear effect which are excluded from the regression and can then appear in the covariance with the re-ordering of $\mathcal{M}_{cov}$ at step 5, the inputs with both effects appearing in the regression and covariance functions and, finally, the inactive input variables excluded from both.

2.5 APPLICATIONS

2.5.1 Analytical test case

First, an analytical function called the g-function of Sobol is used to illustrate and justify our methodology. The g-function of Sobol is defined for d inputs uniformly distributed on $[0,1]^d$:

$$g_{\text{Sobol}}(X_1,\ldots,X_d) = \prod_{k=1}^{d} g_k(X_k) \text{ where } g_k(X_k) = \frac{|4X_k - 2| + a_k}{1 + a_k} \text{ and } a_k \geq 0.$$

Because of its complexity (strongly nonlinear and non-monotonic relationship) and the availability of analytical sensitivity indices, the g-function of Sobol is a well known test example in the studies of global sensitivity analysis algorithms (Saltelli et al. [25]). The contribution of each input X_k to the variability of the model output is represented by the weighting coefficient a_k. The lower this coefficient a_k, the more significant the variable X_k. For example :

$$\begin{cases} a_k = 0 \rightarrow X_k \text{ is very important,} \\ a_k = 1 \rightarrow X_k \text{ is relatively important,} \\ a_k = 9 \rightarrow X_k \text{ is non important,} \\ a_k = 99 \rightarrow X_k \text{ is non significant.} \end{cases}$$

For our analytical test, we choose $a_k = k$.

Applying our methodology to the g-function of Sobol, we illustrate its different steps, especially the importance of rerunning the estimation procedure after sorting the inputs by decreasing ΔQ_2

(cf. steps 4 and 5). At the same time, comparisons are made with other reference software like, for example, the GEM-SA software (O'Hagan [21], freely available at *http ://www.ctcd.group.shef.ac.uk/gem.html*)

To do this, different dimensions of inputs are considered, from 4 to 20 : $d = 4, 6, \dots, 20$. For each dimension d, we generate a learning sample formed by $N_{LS} = d \times 10$ simulations of the g-function of Sobol following the Latin Hypercube Sampling (LHS) method (McKay et al. [20]). Using these learning data, two Gp models are built : one following our methodology and one using the GEM-SA software. For each method, the Q_2 coefficient is computed on a test sample of $N_{TS} = 1000$ points. For each dimension d, this procedure is repeated 50 times to obtain an average performance in terms of the prediction capabilities of each method (mean of Q_2). The standard deviation of Q_2 is also a good indicator of the robustness of each method.

For each dimension d, the mean and standard deviation of Q_2 computed on the test sample using different methods are presented in Table 2.1. Three methods are compared : the GEM-SA software, our methodology without steps 4 and 5, and our methodology with steps 4 and 5.

g-Sobol simulations		GEM-SA software		Gp methodology without steps 4 and 5		Gp methodology with steps 4 and 5	
d	N_{LS}	$\overline{Q_2}$	sd	$\overline{Q_2}$	sd	$\overline{Q_2}$	sd
4	40	0.82	0.08	0.60	0.21	0.86	0.07
6	60	0.67	0.24	0.59	0.16	0.85	0.05
8	80	0.66	0.13	0.61	0.10	0.85	0.04
10	100	0.59	0.25	0.63	0.13	0.83	0.05
12	120	0.57	0.16	0.61	0.15	0.84	0.05
14	140	0.60	0.17	0.61	0.14	0.83	0.03
16	160	0.62	0.11	0.67	0.06	0.86	0.04
18	180	0.66	0.09	0.67	0.05	0.84	0.03
20	200	0.64	0.09	0.72	0.07	0.86	0.02

TAB. 2.1 – **Mean $\overline{Q_2}$ and standard deviation sd of the predictivity coefficient Q_2 for several implementations of the g-function of Sobol.**

For the values of d higher than 6, our methodology including double selection of inputs (with steps 4 and 5) clearly outperforms the others. More precisely, the pertinence of rerunning the estimation procedure after sorting the inputs by decreasing ΔQ_2 is obvious. The prediction accuracy is much more robust (lower standard deviation of Q_2).

The drawback of our methodology lies in the somewhat costly steps 4 and 5. Indeed, sequential estimation and rerunning of the procedure require many executions of the Hooke & Jeeves algorithm, particularly in the case of a double cross validation (cf. steps 3.4 and 7 of the algorithm). Consequently, this approach is much more computer time expensive than the GEM-SA software. For example, for a simulation with $d = 10$ and $N_{LS} = 100$, the computing time of our approach is on average ten times larger than that of the GEM-SA software.

For a practitioner, a compromise is usually made between the time to obtain the sampling design points and the time to build a metamodel. As a conclusion of this section, our methodology is interesting for high dimensional input models (more than ten), for inadequate or small sampling designs (a few hundreds) and when simpler methodologies have failed. The data presented in the next section fall into this scope.

Remark 2.4 *The Gp model used in the GEM-SA software has a gaussian covariance function. Our model uses a generalized exponential correlation function even if it requires the estimation of twice as many hyperparameters. Indeed, the sequential approach allows to estimate a large number of hyperparameters.*

2.5.2 Application on an hydrogeologic transport code

Our methodology is now applied to the data obtained from the modeling of strontium 90 (noted ^{90}Sr) transport in saturated porous media using the MARTHE software (developed by BRGM, the French Geological Survey). The MARTHE computer code models flow and transport equations in three-dimensional porous formations. In the context of an environmental impact study, this code is used to model ^{90}Sr transport in saturated media for a radwaste temporary storage site in Russia (Volkova et al. [29]). One of the final purposes is to determine the short-term evolution of ^{90}Sr transport in soils in order to help rehabilitation decision making. Only a partial characterization of the site has been made and, consequently, values of the model input parameters are not known precisely. One of the first goals is to identify the most influential parameters of the computer code in order to improve the characterization of the site in an optimal way. Because of large computing time of the MARTHE code, Volkova et al. [29] propose to construct a metamodel on the basis of the first learning sample. In the following, our Gp methodology is applied and its results are compared to the previous ones obtained with boosting regression trees and linear regression.

Data presentation

Data simulated in this study are composed of 300 observations. Each simulation consists of 20 inputs and 20 outputs. The 20 uncertain model parameters are permeability of different geological layers composing the simulated field (parameters 1 to 7), longitudinal dispersivity coefficients (parameters 8 to 10), transverse dispersivity coefficients (parameters 11 to 13), sorption coefficients (parameters 14 to 16), porosity (parameter 17) and meteoric water infiltration intensities (parameters 18 to 20). To study sensitivity of the MARTHE code to these parameters, simulations of these 20 parameters have been made by the LHS method.

For each simulated set of parameters, MARTHE code computes transport equations of ^{90}Sr and predicts the evolution of ^{90}Sr concentration. Initial and boundary conditions for the flow and transport models are fixed at the same values for all simulations. So, for an initial map of ^{90}Sr concentration in 2002 and a set of 20 input parameter values, MARTHE code computes a map of predicted concentrations in 2010. For each simulation, the 20 outputs considered are values of ^{90}Sr concentration, predicted for year 2010, in 20 piezometers located on the waste repository site.

Comparison of three different models

For each output, we choose to compare and analyze the results of three models :

▷ Linear regression : it represents a model that provides a reference for the contribution of the Gp model stochastic component to modeling quality. Indeed, comparison between simple linear regression and Gp model will show if considering spatial correlations has significant impact on the modeling results. Moreover, a selection based on the AICC criterion is implemented to optimize the results of the linear regression.

▷ Boosting of regression trees : this model was used in the previous study of the data (Volkova et al. [29]). The boosting trees method is based on sequential construction of weak models (here regression trees with low interaction depth), that are then aggregated. The MART algorithm (Multiple Additive Regression Trees), described in Hastie et al. [10], is used here. The boosting trees method is relatively complex, in the sense that, as with neural networks, it is a black box model, efficient but quite difficult to interpret. It is interesting to see if a Gp model, that is easier to interpret and offers a quickly computable predictor, can compete with a more complex method in terms of modeling and prediction quality. Note that the boosting trees algorithm also makes its proper input selection.

▷ Gaussian process : to implement this model, the methodology previously described in this paper is applied, with the input selection procedure.

Results

To compare prediction quality of the three different models presented above, the coefficient of predictivity Q_2 is estimated by a 6-fold cross validation. Note that for each model the results correspond to the optimal set of inputs included in the model. To avoid some bias in the results, the cross validation used to select variables in the Gp model (see step 6) differs from the cross validation used to validate and compare prediction capabilities of the three models. Indeed, at each cross validation step (used to validate), data are divided into a learning sample (denoted $LS1$) of 250 observations and a test sample ($TS1$) of the 50 remaining observations. For the Gp model, the procedure of variable selection is then performed by a second cross validation on $LS1$ (for example : a 4-fold cross validation, dividing $LS1$ into a learning set $LS2$ of 210 data and a test set $TS2$ of the 40 others). Then, an optimal set of variables is determined and a Gp model is built based on the 250 data of $LS1$ (with this optimal set of inputs previously selected). Finally, the model is validated on the test set $TS1$ that has never been used for the Gp model construction.

The results are presented in Table 2.2 and are taken up in a barplot (see Figure 2.2). Results obtained for the output 8 (piezometer p110) are not considered because of physically insignificant concentration values. For most outputs, the Gp performance is superior to linear regression and boosting, in many cases substantially so. Concerning the outputs 11 ($p27k$) and 19 ($p4a$), the performances of the Gp model are worse than the linear regression ones. However, for these two outputs, the prediction errors are very high and consequently the difference of performance between the two models can be considered as non-significant.

As expected, for most of the outputs, the linear regression presents the worst results. When this model is successfully adapted, the two others are also efficient. When linear regression fails (for example, for output number 12), Gp model presents a real interest, since it gives results as good as those of the boosting trees method. In fact, this is verified for all the ouputs and results are significantly better for several outputs (outputs 1, 2, 4, 9, 12, 13 and 16). To illustrate this, the Figure 2.3 shows the predicted values vs real values for the output 16, for the Gaussian process and boosting trees models. It clearly shows a better repartition of the Gp model residuals than the boosting trees model ones.

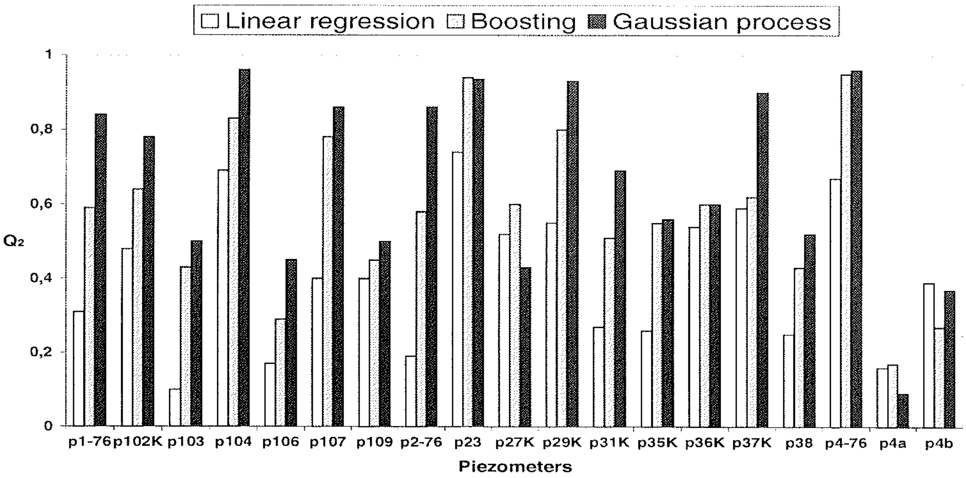

Fig. 2.2 – **Barplot of the predictivity coefficient Q_2 for the three different models.**

Furthermore, the estimator of MSE, that is expressed analytically (see Equation (6)), can be used

Output		Linear regression	boosting trees	Gaussian process
Denomination	Number	Q_2	Q_2	Q_2
p1-76	1	0.31	0.59	0.84
p102K	2	0.48	0.64	0.78
p103	3	0.10	0.43	0.5
p104	4	0.69	0.83	0.96
p106	5	0.17	0.29	0.45
p107	6	0.40	0.78	0.86
p109	7	0.40	0.45	0.5
p2-76	9	0.19	0.58	0.86
p23	10	0.74	0.94	0.935
p27K	11	0.52	0.60	0.43
p29K	12	0.55	0.80	0.93
p31K	13	0.27	0.51	0.69
p35K	14	0.26	0.55	0.56
p36K	15	0.54	0.60	0.60
p37K	16	0.59	0.62	0.90
p38	17	0.25	0.43	0.52
p4-76	18	0.67	0.95	0.96
p4a	19	0.16	0.17	0.09
p4b	20	0.39	0.27	0.37

TAB. 2.2 – **Predictivity coefficients Q_2 for the three different models of the MARTHE data.**

as a local prediction interval. To illustrate this, we consider 50 observations of the output 16. Figure 2.4 shows the observed values, the predicted values and the upper and lower bounds of the 95% prediction interval based on the MSE local estimator. It confirms the good adequacy of the Gp model for this output because all the observed values (except one point) are inside the prediction interval curves.

Analysis

These results confirm the potential of the Gp model and justify its application for computer codes. Application of our methodology to complex data also confirms the efficiency of our input selection procedure. For a fixed set of inputs in the covariance function, we can verify that this procedure selects the best set of inputs in regression part. Furthermore, the necessity of conducting sequential and ordered procedure estimation has been demonstrated. Indeed, if all the Gp parameters (i.e. considering the 20 inputs) are directly and simultaneously estimated with the DACE algorithm, they are not correctly determined and poor results in terms of Q_2 are obtained. So, in case of a complex model with a large number of inputs, we recommend using a selection procedure such as the

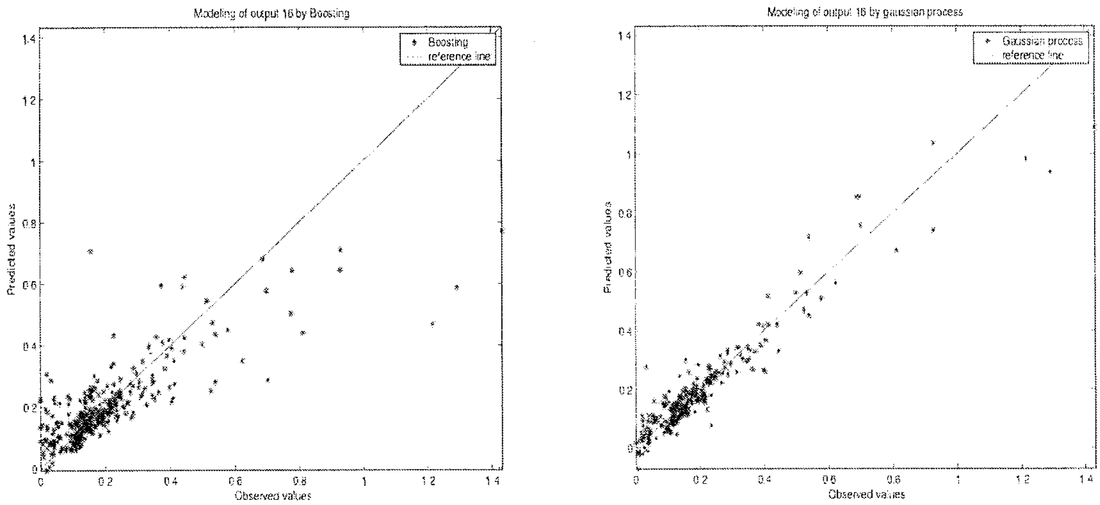

Fig. 2.3 – **Plot of predicted values vs real values for boosting trees (left) and Gaussian process (right).**

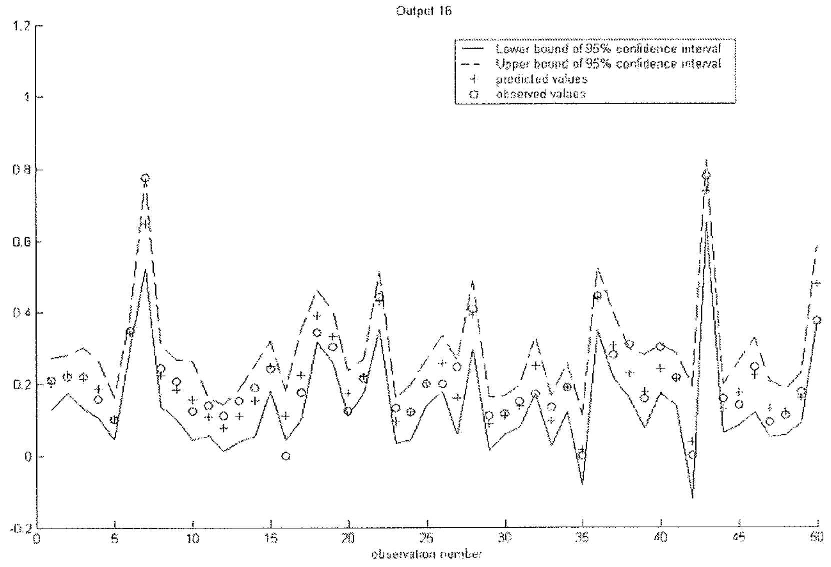

Fig. 2.4 – **Plot of observed and Gaussian process predicted values for the output 16 with the** 95% **prediction interval based on** $\widehat{MSE}$ **formula.**

algorithm of section 2.4.

The study of these data have motivated the choice of this methodology. At first, Welch's algorithm (see section 2.3.3) has been tried. Considering the poor results obtained, our methodology based on the DACE estimation algorithm has been developed. To illustrate this, let us detail the different results obtained on the output number 9. With our methodology based on the DACE estimation, the Q_2 coefficient (always computed by a 6-fold cross validation) is 0.86, while with Welch's algorithm (used in its basic version), Q_2 is close to zero. The difference in the results between the two methods can be explained by the value of estimated correlation parameters which are significantly different.

To minimize the number of correlation parameters and consequently reduce computer time required for estimation, the possible values of power parameters p_i $(i = 1, \dots, d)$ can be limited to 0.5, 1 and 2. It can be a solution to optimize computer time. It allows an exhaustive, quick and optimal representation of different kinds of correlation functions (two kinds of inflexion are represented). Furthermore, in many cases, estimation of power parameter with generalized exponential correlation converges to exponential ($p_i = 1$) or Gaussian ($p_i = 2$) correlation.

2.6 CONCLUSION

The Gaussian process model presents some real advantages compared to other metamodels : exact interpolation property, simple analytical formulations of the predictor, availability of the mean squared error of the predictions and the proved efficiency of the model. The keen interest in this method is testified by the publication of the recent monographs of Santner et al. [26], Fang et al. [9] and Rasmussen & Williams [23].

However, for its application to complex industrial problems, developing a robust implementation methodology is required. In this paper, we have outlined some difficulties arising from the parameter estimation procedure (instability, high number of parameters) and the necessity of a progressive model construction. Moreover, an a priori choice of regression function and, more important, of covariance function is essential to parameterize the Gaussian process model. The generalized exponential covariance function appears in our experience as a judicious and recommended choice. However, this covariance function requires the estimation of $2d$ correlation parameters, where d is the input space dimension. In this case, the sequential estimation and selection procedures of our methodology are more appropriate. This methodology is interesting when the computer model is rather complex (non linearities, threshold effects, etc.), with high dimensional inputs ($d > 10$) and for small size samples (a few hundreds).

Results obtained on the MARTHE computer code are very encouraging and place the Gaussian process as a good and judicious alternative to efficient but non-explicit and complex methods such as boosting trees or neural networks. It has the advantage of being easily evaluated on a new parameter set, independently of the metamodel complexity. Moreover, several statistical tools are available because of the analytical formulation of the Gaussian model. For example, the MSE estimator offers a good indicator of the model's local accuracy. In the same way, inference studies can be developed on parameter estimators and on the choice of the experimental input design. Finally, one possible improvement in our construction algorithm is based on the sequential approach of the choice of input design, which remains an active research domain (Sceidt & Zabalza-Mezghani [27] for example).

2.7 ACKNOWLEGMENTS

This work has been done with François Van Dorpe (CEA Cadarache) and Elena Volkova (Institut Kurchatov, Moscow).

2.8 REFERENCES

[1] R. Ababou, A.C. Bagtzoglou, and E.F. Wood. On the condition number of covariance matrices in kriging, estimation, and simulation of random fields. *Mathematical Geology*, 26 :99–133, 1994.

[2] P. Abrahamsen. A review of Gaussian random fields and correlation functions. Technical Report 878, Norsk Regnesentral, 1994.

[3] F.M. Alam, K.R. McNaught, and T.J. Ringrose. A comparison of experimental designs in the development of a neural network simulation metamodel. *Simulation Modelling Practice and Theory*, 12 :559–578, 2004.

[4] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear programming*. John Wiley & Sons, Inc, 1993.

[5] G.E. Box and N.R. Draper. *Empirical model building and response surfaces*. Wiley Series in Probability and Mathematical Statistics. Wiley, 1987.

[6] J-P. Chilès and P. Delfiner. *Geostatistics : Modeling spatial uncertainty.* Wiley, New-York, 1999.

[7] N.A.C. Cressie. *Statistics for spatial data.* Wiley Series in Probability and Mathematical Statistics. Wiley, 1993.

[8] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416) :953–963, 1991.

[9] K-T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments.* Chapman & Hall/CRC, 2006.

[10] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning.* Springer, 2002.

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

[12] J.A. Hoeting, R.A. Davis, A.A. Merton, and S.E. Thompson. Model selection for geostatistical models. *Ecological Applications*, 16 :87–98, 2006.

[13] B. Iooss, F. Van Dorpe, and N. Devictor. Response surfaces and sensitivity analyses for an environmental model of dose calculations. *Reliability Engineering and System Safety*, 91 :1241–1251, 2006.

[14] A. Jourdan and I. Zabalza-Mezghani. Response surface designs for scenario mangement and uncertainty quantification in reservoir production. *Mathematical Geology*, 36(8) :965–985, 2004.

[15] J. Kleijnen. Sensitivity analysis and related analyses : a review of some statistical techniques. *Journal of Statistical Computation and Simulation*, 57 :111–142, 1997.

[16] J. Kleijnen. An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research*, 164 :287–300, 2005.

[17] J. Kleijnen and R.G. Sargent. A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research*, 120 :14–29, 2000.

[18] S.N. Lophaven, H.B. Nielsen, and J. Sondergaard. DACE - A Matlab kriging toolbox, version 2.0. Technical Report IMM-TR-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark, 2002. <http ://www.immm.dtu.dk/∼hbn/dace>.

[19] G. Matheron. *La théorie des variables régionalisées, et ses applications.* Les Cahiers du Centre de Morphologie Mathématique de Fontainebleau, Fascicule 5. Ecole des Mines de Paris, 1970.

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

[21] A. O'Hagan. Bayesian analysis of computer code outputs : A tutorial. *Reliability Engineering and System Safety*, 91 :1290–1300, 2006.

[22] I.G. Osio and C.H. Amon. An engineering design methodology with multistage bayesian surrogates and optimal sampling. *Research in Engineering Design*, 8 :189–206, 1996.

[23] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning.* MIT Press, 2006.

[24] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4 :409–435, 1989.

[25] A. Saltelli, K. Chan, and E.M. Scott, editors. *Sensitivity analysis.* Wiley Series in Probability and Statistics. Wiley, 2000.

[26] T. Santner, B. Williams, and W. Notz. *The design and analysis of computer experiments.* Springer, 2003.

[27] C. Scheidt and I. Zabalza-Mezghani. Assessing uncertainty and optimizing production schemes : Experimental designs for non-linear production response modeling. an application to early water breakthrough prevention. In *ECMOR IX*, Cannes, France, August 2004.

[28] E. Vazquez, E. Walter, and G. Fleury. Intrinsic kriging and prior information. *Applied Stochastic Models in Business and Industry*, 21 :215–226, 2005.

[29] E. Volkova, B. Iooss, and F. Van Dorpe. Global sensitivity analysis for a numerical model of radionuclide migration from the RRC "Kurchatov Institute" radwaste disposal site. *Stochastic Environmental Research and Risk Assesment*, 22 :17–31, 2008.

[30] R. von Mises. *Mathematical Theory of Probability and Statistics.* Academic Press, 1964.

[31] 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.

3 CALCULATIONS OF SOBOL INDICES FOR THE GAUSSIAN PROCESS METAMODEL

3.1 ABSTRACT

Global sensitivity analysis of complex numerical models can be performed by calculating variance-based importance measures of the input variables, such as the Sobol indices. However, these techniques, requiring a large number of model evaluations, are often unacceptable for time expensive computer codes. A well known and widely used decision consists in replacing the computer code by a metamodel, predicting the model responses with a negligible computation time and making straightforward the estimation of Sobol indices. In this paper, we discuss about the Gaussian process model which gives analytical expressions of Sobol indices. Two approaches are studied to compute the Sobol indices : the first based on the predictor of the Gaussian process model and the second based on the global stochastic process model. Comparisons between the two estimates, made on analytical examples, show the superiority of the second approach in terms of convergence and robustness. Moreover, the second approach allows to integrate the modeling error of the Gaussian process model by directly giving some confidence intervals on the Sobol indices. These techniques are finally applied to a real case of hydrogeological modeling.

3.2 INTRODUCTION

Environmental risk assessment is often based on complex computer codes, simulating for instance an atmospheric or hydrogeological pollution transport. These computer models calculate several output values (scalars or functions) which can depend on a high number of input parameters and physical variables. To provide guidance to a better understanding of this kind of modeling and in order to reduce the response uncertainties most effectively, sensitivity measures of the input importance on the response variability can be useful (Saltelli et al. [24], Kleijnen [12], Helton et al. [9]). However, the estimation of these measures (based on Monte-Carlo methods for example) requires a large number of model evaluations, which is unacceptable for time expensive computer codes. This kind of problem is of course not limited to environmental modeling and can be applied to any simulation system.

To avoid the problem of huge calculation time in sensitivity analysis, it can be useful to replace the complex computer code by a mathematical approximation, called a response surface or a surrogate model or also a metamodel. The response surface method (Box & Draper [2]) consists in constructing a function from few experiments, that simulates the behavior of the real phenomenon in the domain of influential parameters. These methods have been generalized to develop surrogates for costly computer codes (Sacks et al. [23], Kleijnen & Sargent [13]). Several metamodels are classically used : polynomials, splines, generalized linear models, or learning statistical models like neural networks, regression trees, support vector machines (Chen et al. [3], Fang et al. [8]).

Our attention is focused on the Gaussian process model which can be viewed as an extension of the kriging principles (Matheron [18], Cressie [6], Sacks et al. [23]). This metamodel which is characterized by its mean and covariance functions, presents several advantages : it is an exact interpolator and it is interpretable (not a black-box function). Moreover, numerous authors (for example, Currin et al. [7], Santner et al. [25], Vazquez et al. [28], Rasmussen & Williams [22]) have shown how this model can provide a statistical basis for computing an efficient predictor of code response. In addition to its efficiency, this model gives an analytical formula which is very useful for sensitivity analysis, especially for the variance-based importance measures, the so-called Sobol indices (Sobol [26], Saltelli et al. [24]). To derive analytical expression of Sobol indices, Chen et al. [4] used tensor-product formulation and Oakley & O'Hagan [20] considered the Bayesian formalism of Gaussian processes.

We propose to compare these two analytical formulations of Sobol indices for the Gaussian process model : the first is obtained considering only the predictor, i.e. the mean of the Gaussian process model (Chen et al. [4]), while the second is obtained using all the global stochastic model (Oakley & O'Hagan [20]). In the last case, the estimate of a Sobol index is itself a random variable. Its standard deviation is available and we propose an original algorithm to estimate its distribution. Consequently, our method leads to build confidence intervals for the Sobol indices. To our knowledge, this information has not been proposed before and can be obtained thanks to the analytical formulation of the Gaussian process model error. This is particularly interesting in practice, when the predictive quality of the metamodel is not high (because of small learning sample size for example), and our confidence on Sobol index estimates via the metamodel is poor.

The next section briefly explains the Gaussian process modeling and the Sobol indices defined in the two approaches (predictor-only and global model). In section 3, the numerical computation of a Sobol index is presented. In the case of the global stochastic model, a procedure is developed to simulate its distribution. Section 4 is devoted to applications on analytical functions. First, comparisons are made between the Sobol indices based on the predictor and those based on the global model. The pertinence of simulating all the distribution of Sobol indices is therefore evaluated. Finally, Sobol indices and their uncertainty are computed for a real data set coming from a hydrogeological transport model based on waterflow and diffusion dispersion equations. The last section provides some possible extensions and concluding remarks.

3.3 SOBOL INDICES WITH GAUSSIAN PROCESS MODEL

3.3.1 Gaussian process model

Let us consider n realizations of a computer code. Each realization $y(x)$ of the computer code output corresponds to a d-dimensional input vector $x = (x_1, ..., x_d)$. The n input points corresponding to the code runs are called an experimental design and are denoted as $X_s = (x^{(1)}, ..., x^{(n)})$. The outputs will be denoted as $Y_s = (y^{(1)}, ..., y^{(n)})$ with $y^{(i)} = y(x^{(i)}), i = 1, ..., n$. Gaussian process (Gp) modeling treats the deterministic response $y(x)$ as a realization of a random function $Y(x)$, including a regression part and a centered stochastic process. The sample space Ω denotes the space of all possible outcomes ω, which is usually the Lebesgue-measurable set of real numbers. The Gp is defined on $R^d \times \Omega$ and can be written as :

$$Y(x, \omega) = f(x) + Z(x, \omega). \tag{1}$$

In the following, we use indifferently the terms Gp model and Gp metamodel.

The deterministic function $f(x)$ provides the mean approximation of the computer code. Our study is limited to the parametric case where the function f is a linear combination of elementary functions. Under this assumption, $f(x)$ can be written as follows :

$$f(x) = \sum_{j=0}^{k} \beta_j f_j(x) = F(x)\beta,$$

where $\beta = [\beta_0, \ldots, \beta_k]^t$ is the regression parameter vector, f_j $(j = 1, \ldots, k)$ are basis functions and $F(x) = [f_0(x), \ldots, f_k(x)]$ is the corresponding regression matrix. In the case of the one-degree polynomial regression, $(d+1)$ basis functions are used :

$$\begin{cases} f_0(x) = 1, \\ f_i(x) = x_i \text{ for } i = 1, \ldots, d. \end{cases}$$

In our applications, we use this one-degree polynomial as the regression part in order to simplify all the analytical numerical computation of sensitivity indices. This can be extended to other bases of regression functions. Without prior information on the relationship between the output and the inputs, a basis of one-dimensional functions is recommended to simplify the computations in sensitivity analysis and to keep one of the most advantages of Gp model (Martin & Simpson [17]).

The stochastic part $Z(\boldsymbol{x},\omega)$ is a Gaussian centered process fully characterized by its covariance function : $\text{Cov}_\Omega(Z(\boldsymbol{x},\omega),Z(\boldsymbol{u},\omega)) = \sigma^2 R(\boldsymbol{x},\boldsymbol{u})$, where σ^2 denotes the variance of Z and R is the correlation function that provides interpolation and spatial correlation properties. To simplify, a stationary process $(Z(\boldsymbol{x},\omega))$ is considered, which means that the correlation between $Z(\boldsymbol{x},\omega)$ and $Z(\boldsymbol{u},\omega)$ is a function of the difference between $\boldsymbol{x}$ and $\boldsymbol{u}$. Moreover, our study is restricted to a family of correlation functions that can be written as a product of one-dimensional correlation functions :

$$\text{Cov}_\Omega(Z(\boldsymbol{x},\omega),Z(\boldsymbol{u},\omega)) = \sigma^2 R(\boldsymbol{x}-\boldsymbol{u}) = \sigma^2 \prod_{l=1}^{d} R_l(x_l - u_l). \tag{2}$$

This form of correlation function is particularly well adapted to get some simplifications of the integrals in the future analytical developments : in the case of independent inputs, it implies the computation of only one or two-dimensional integrals to compute the Sobol indices. Indeed, as described in section 3.4.2, the application and the computation of the Sobol index formulae are simplified when the correlation function has the form of a one-dimensional product (Santner et al. [25]).

Among other authors, Chilès & Delfiner [5] and Rasmussen & Williams [22] give a list of correlation functions with their advantages and drawbacks. Among all these functions, our attention is devoted to the generalized exponential correlation function :

$$R_{\boldsymbol{\theta},\boldsymbol{p}}(\boldsymbol{x}-\boldsymbol{u}) = \prod_{l=1}^{d} \exp(-\theta_l |x_l - u_l|^{p_l}) \text{ with } \theta_l \geq 0 \text{ and } 0 < p_l \leq 2,$$

where $\boldsymbol{\theta} = [\theta_1,\ldots,\theta_d]^t$ and $\boldsymbol{p} = [p_1,\ldots,p_d]^t$ are the correlation parameters. This choice is motivated by the derivation and regularity properties of this function. Moreover, within the range of covariance parameters values, a wide spectrum of shapes are possible : for example $p = 1$ gives the exponential correlation function while $p = 2$ gives the Gaussian correlation function.

3.3.2 Joint and conditional distributions

Under the hypothesis of a Gp model, the learning sample Y_s follows a multivariate normal distribution $p_\Omega(Y_s\,|X_s)$:

$$p_\Omega(Y_s,\omega|X_s) = \mathcal{N}\left(F_s\boldsymbol{\beta},\Sigma_s\right),$$

where $F_s = [F(\boldsymbol{x}^{(1)})^t,\ldots,F(\boldsymbol{x}^{(n)t})]$ is the regression matrix and

$$\Sigma_s = \sigma^2 R_{\boldsymbol{\theta},\boldsymbol{p}}\left(\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right)_{i,j=1\ldots n}$$

is the covariance matrix.

If a new point $\boldsymbol{x}^* = (x_1^*,\ldots,x_d^*)$ is considered, the joint probability distribution of $(Y_s, Y(\boldsymbol{x}^*,\omega))$ is :

$$p_\Omega(Y_s, Y(\boldsymbol{x}^*,\omega)|X_s,\boldsymbol{x}^*,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}) = \mathcal{N}\left(\begin{bmatrix} F_s \\ F(\boldsymbol{x}^*)\end{bmatrix}\boldsymbol{\beta}, \begin{bmatrix} \Sigma_s & \boldsymbol{k}(\boldsymbol{x}^*) \\ \boldsymbol{k}(\boldsymbol{x}^*)^t & \sigma^2 \end{bmatrix}\right), \tag{3}$$

with

$$\begin{aligned} \boldsymbol{k}(\boldsymbol{x}^*) &= (\,\text{Cov}_\Omega(y^{(1)}, Y(\boldsymbol{x}^*,\omega)),\ldots,\text{Cov}_\Omega(y^{(n)}, Y(\boldsymbol{x}^*,\omega))\,)^t \\ &= \sigma^2(\,R_{\boldsymbol{\theta},\boldsymbol{p}}(\boldsymbol{x}^{(1)},\boldsymbol{x}^*),\ldots,R_{\boldsymbol{\theta},\boldsymbol{p}}(\boldsymbol{x}^{(n)},\boldsymbol{x}^*)\,)^t. \end{aligned} \tag{4}$$

By conditioning this joint distribution on the learning sample, we can readily obtain the conditional distribution of $Y(\boldsymbol{x}^*,\omega)$ which is Gaussian (von Mises [30]) :

$$\begin{aligned} &p_\Omega(Y(\boldsymbol{x}^*,\omega)|Y_s,X_s,\boldsymbol{x}^*,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}) \\ &= \mathcal{N}\left(\mathbb{E}_\Omega[Y(\boldsymbol{x}^*,\omega)|Y_s,X_s,\boldsymbol{x}^*,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}],\text{Var}_\Omega[Y(\boldsymbol{x}^*,\omega)|Y_s,X_s,\boldsymbol{x}^*,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}]\right), \end{aligned} \tag{5}$$

with

$$\mathbb{E}_\Omega[Y(\boldsymbol{x}^*,\omega)|Y_s,X_s,\boldsymbol{x}^*,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}] = F(\boldsymbol{x}^*)\boldsymbol{\beta} + \boldsymbol{k}(\boldsymbol{x}^*)^t\boldsymbol{\Sigma}_s^{-1}(Y_s - F_s\boldsymbol{\beta}), \tag{6}$$

$$\text{Var}_\Omega[Y(\boldsymbol{x}^*,\omega)|Y_s,X_s,\boldsymbol{x}^*,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}] = \sigma^2 - \boldsymbol{k}(\boldsymbol{x}^*)^t\boldsymbol{\Sigma}_s^{-1}\boldsymbol{k}(\boldsymbol{x}^*). \tag{7}$$

The conditional mean of Eq. (6) is used as a predictor. The conditional variance formula of Eq. (7) corresponds to the mean squared error (MSE) of this predictor and is also known as the kriging variance. As we obtained the distribution for a new point conditionally to the learning sample, we can consider the covariance between two new sites. A Gp conditional to the learning sample is obtained and denoted as follows :

$$\begin{aligned}(Y|Y_s,X_s,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}) \sim \text{Gp}(\ &\mathbb{E}_\Omega[Y(\boldsymbol{x}^*,\omega)|Y_s,X_s,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}],\\ &\text{Cov}_\Omega(Y(\boldsymbol{x}^*,\omega),Y(\boldsymbol{u}^*,\omega)|Y_s,X_s,\boldsymbol{x}^*,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}))\end{aligned} \tag{8}$$

with the same expression for the conditional mean than Eq. (6) and

$$\text{Cov}_\Omega\left(Y(\boldsymbol{x}^*,\omega),Y(\boldsymbol{u}^*,\omega)|Y_s,X_s,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p}\right) = \sigma^2\left(R_{\boldsymbol{\theta},\boldsymbol{p}}(\boldsymbol{x}^*,\boldsymbol{u}^*) - \boldsymbol{k}(\boldsymbol{x}^*)^t\boldsymbol{\Sigma}_s^{-1}\boldsymbol{k}(\boldsymbol{u}^*)\right). \tag{9}$$

The conditional Gp model (8) provides an analytical formula which can be directly used for sensitivity analysis, and more precisely to compute the Sobol indices. To simplify the notations, the conditional Gp $(Y|Y_s,X_s,\boldsymbol{\beta},\sigma,\boldsymbol{\theta},\boldsymbol{p})$ will now be written in a simplified form : $Y_{\text{Gp}|Y_s,X_s}(X,\omega)$.

3.3.3 Sobol indices

Methods based on variance decomposition aim at determining the part of the variance of the output $Y(x)$ resulting from each variable $x_i, i = 1,\dots,d$. A global measure of the sensitivity of $Y(x)$ to each input x_i is given by the first order Sobol index (Sobol [26], Saltelli et al. [24]) :

$$S_i = \frac{\text{Var}_{X_i}[\mathbb{E}_{X_1,\dots,X_d}Y|X_i]}{\text{Var}_{X_1,\dots,X_d}[Y]} \text{ for } i = 1,\dots,d.$$

These indices have been defined for deterministic functions Y of the inputs $X_1,\dots,X_d$ but, in the case of the conditional Gp model, we have a stochastic function of the inputs. A first solution is applying the Sobol index formula to the predictor, i.e. the mean of the conditional Gp (Eq. (6)) which is a deterministic function of the inputs. Analytical calculations are developed by Chen et al. [4]. The second approach that we consider consists in using the whole global conditional Gp by taking into account not only the mean of conditional Gp model but also its covariance structure as Oakley & O'Hagan [20] did. In this case, when the Sobol definition is applied to the global Gp model, a random variable is obtained and constitutes a new sensitivity measure. Its expectation can be then considered as a sensitivity index. Its variance and more generally its distribution can then be used as an indicator of sensitivity index accuracy.

To sum up, the two approaches can be defined as follows :

- Approach 1 : Sobol indices computed with the predictor-only

$$S_i = \frac{\text{Var}_{X_i}\mathbb{E}_{X_1,\dots,X_d}[\mathbb{E}_\Omega[Y_{\text{Gp}|Y_s,X_s}(X,\omega)]|X_i]}{\text{Var}_{X_1,\dots,X_d}\mathbb{E}_\Omega[Y_{\text{Gp}|Y_s,X_s}(X,\omega)]} \text{ for } i = 1,\dots,d. \tag{10}$$

- Approach 2 : Sobol indices computed with the global Gp model

$$\tilde{S}_i(\omega) = \frac{\text{Var}_{X_i}\mathbb{E}_{X_1,\dots,X_d}[Y_{\text{Gp}|Y_s,X_s}(X,\omega)|X_i]}{\mathbb{E}_\Omega\text{Var}_{X_1,\dots,X_d}[Y_{\text{Gp}|Y_s,X_s}(X,\omega)]} \text{ for } i = 1,\dots,d. \tag{11}$$

$\tilde{S}_i(\omega)$ is then a random variable; its mean can be considered as a sensitivity index and its variance as an indicator of its accuracy :

$$\left\{\begin{array}{ll} \mu_{\tilde{S}_i} = \dfrac{\mathbb{E}_\Omega \mathrm{Var}_{X_i} \mathbb{E}_{X_1,\dots,X_d}[Y_{\mathrm{Gp}|Y_s,X_s}(X,\omega)|X_i]}{\mathbb{E}_\Omega \mathrm{Var}_{X_1,\dots,X_d}[Y_{\mathrm{Gp}|Y_s,X_s}(X,\omega)]} & \text{for } i = 1,\dots,d. \\ \\ \sigma^2_{\tilde{S}_i} = \dfrac{\mathrm{Var}_\Omega \mathrm{Var}_{X_i} \mathbb{E}_{X_1,\dots,X_d}[Y_{\mathrm{Gp}|Y_s,X_s}(X,\omega)|X_i]}{(\mathbb{E}_\Omega \mathrm{Var}_{X_1,\dots,X_d}[Y_{\mathrm{Gp}|Y_s,X_s}(X,\omega)])^2} & \text{for } i = 1,\dots,d. \end{array}\right. \tag{12}$$

Our work focuses on the computation and the study of the sensitivity indices defined following the two approaches, respectively S_i and $\mu_{\tilde{S}_i}$. We will also propose a methodology to numerically simulate the probability distribution of $\tilde{S}_i$. Then, a study to compare the accuracy and the robustness of the two indices is made on several test functions and the use of the distribution of $\tilde{S}_i$ is illustrated to build confidence intervals.

3.4 IMPLEMENTATION OF SOBOL INDICES

3.4.1 Estimation of Gp parameters

First of all, to build the conditional Gp defined by Eq. (8), regression and correlation parameters (often called hyperparameters) have to be determined. Indeed, the Gp model is characterized by the regression parameter vector $\boldsymbol{\beta}$, the correlation parameters $(\boldsymbol{\theta}, \boldsymbol{p})$ and the variance parameter σ^2. The maximum likelihood method is commonly used to estimate these parameters from the learning sample (X_s, Y_s).

Several algorithms have been proposed in previous papers to numerically solve the maximization of likelihood. Welch at al. [31] use the simplex search method and introduce a kind of forward selection algorithm in which correlation parameters are added step by step to increase the log-likelihood function. In Kennedy and O'Hagan's GEM-SA software (O'Hagan [21]), which uses the Bayesian formalism, the posterior distribution of hyperparameters is maximized, using a conjugate gradient method (the Powel method is used as the numerical recipe). The DACE Matlab free toolbox (Lophaven et al. [14]) uses a powerful stochastic algorithm based on the Hooke & Jeeves method (Bazaraa et al. [1]), which requires a starting point and some bounds to constrain the optimization. In complex applications, Welch's algorithm reveals some limitations and for complex model with high dimensional input, GEM-SA and DACE software cannot be applied directly on data including all the input variables. To solve this problem, we use a sequential version (inspired by Welch's algorithm) of the DACE algorithm. It is based on the step by step inclusion of input variables (previously sorted). This methodology, described in details in Marrel et al. [15], allows progressive parameter estimation by input variables selection both in the regression part and in the covariance function.

3.4.2 Computation of Sobol indices for the two approaches

To perform a variance-based sensitivity analysis for time consuming computer models, some authors propose to approximate the computer code by a metamodel (neural networks in Martin & Simpson [16], polynomials in Iooss et al. [10], boosting regression trees in Volkova et al. [29]). For metamodels with sufficient prediction capabilities, the bias due to the use of the metamodel instead of the true model is negligible (Jacques [11]). The metamodel's predictor have to be evaluated a large number of times to compute Sobol indices via Monte Carlo methods. Recent works based on polynomial chaos expansions (Sudret [27]) have used the special form of this orthogonal functions expansion to derive analytical estimation of Sobol indices. However, the modeling error of this metamodel is not available and then has not been integrated inside the Sobol index estimates.

The conditional Gp metamodel provides an analytic formula which can be easily used for sensitivity analysis in an analytical way. Moreover, in the case of independent inputs and with a covariance which is a product of one-dimensional covariances (Eq. (2)), the analytical formulae of S_i and $\mu_{\tilde{S}_i}$ (respectively Eqs. (10) and (12)) lead to numerical integrals, more precisely to respectively one-dimensional and two-dimensional integrals. The accuracy of these numerical integrations can be easily controlled and are less computer time expensive than Monte Carlo simulations. Few analytical developments of Sobol indices computation (for S_i, $\mu_{\tilde{S}_i}$ and $\sigma^2_{\tilde{S}_i}$) can be found in Oakley & O'Hagan [20].

3.4.3 Simulation of the distribution of $\tilde{S}_i$

For the second approach where $\tilde{S}_i$ is a random variable, the distribution of $\tilde{S}_i$ is not directly available. By taking the mean related to all the inputs except X_i, the main effect of X_i is defined and denoted $A(X_i, \omega)$:

$$A(X_i, \omega) = \mathbb{E}_{X_1,\ldots,X_d}[Y_{\text{Gp}|Y_s,X_s}(X,\omega)|X_i].$$

$A(X_i, \omega)$ is still a Gaussian process defined on $R \times \Omega$ and characterized by its mean and covariance which can be determined in an analytical way by integrating the Gp model over all the inputs except X_i. In the case of independent inputs, one-dimensional integrals are obtained and can be numerically computed. Then, to obtain the Sobol indices, we consider the variance related to X_i of the Gaussian process defined by the centered main effect. This variance is written

$$\int_{a_i}^{b_i} \left(A(x_i, \omega) - \int A(x_i, \omega) d\eta_{x_i} \right)^2 d\eta_{x_i}$$

with $d\eta_{x_i}$ the probability density function of the input X_i defined on $[a_i\,;\,b_i]$. This last expression is a one-dimensional random integral which has to be discretized and approximated by simulations.

The discretization of this random integral over the space of X_i leads to a Gaussian vector of n_{dis} elements :

$$V_{n_{\text{dis}}}(\omega) = \left(A(a_i, \omega),\ A(a_i + \frac{(b_i - a_i)}{n_{\text{dis}}}, \omega),\ \ldots,\ A(a_i + \frac{(n_{\text{dis}} - 1)}{n_{\text{dis}}}(b_i - a_i), \omega),\ A(b_i, \omega) \right)^t.$$

The mean and covariance matrix of this vector are computed using those of the Gaussian process $A(X_i, \omega)$. The random vector $V_{n_{\text{dis}}}$ is then multiplied by the matrix related to the numerical scheme used to compute the integral (rectangle or trapeze method, Simpson's formula ...). The Gaussian vector obtained from this multiplication is denoted $\tilde{V}_{n_{\text{dis}}}$. To simulate it, we use the well known simulation method based on the Cholesky factorisation of the covariance matrix (Cressie [6]). We simulate a n_{dis}-size centered and reduced Gaussian vector and multiply it by the triangular matrix from the Cholesky decomposition. Then, an evaluation of the random integral which constitutes a realization of $\tilde{S}_i$ is computed from the simulation of the vector $\tilde{V}_{n_{\text{dis}}}$. This operation is done k_{sim} times to obtain a probability distribution for $\tilde{S}_i$. It can be noted, that only one Cholesky factorization of the covariance matrix of the n_{dis}-size vector is necessary, and used for all the k_{sim} simulations of $\tilde{S}_i$. To determine if the discretization number n_{dis} and the number of simulations k_{sim} are sufficient, the convergence of the mean and variance of $\tilde{S}_i$ can be studied. Indeed, their values can be easily computed following their analytical expressions (11).

3.5 APPLICATIONS

3.5.1 Comparison of S_i and $\mu_{\tilde{S}_i}$

To compare and study the behavior of the two sensitivity indices S_i and $\mu_{\tilde{S}_i}$, we consider several test functions where the true values of Sobol indices are known. Comparisons between the two approaches are performed in terms of metamodel predictivity, i.e. relatively to the accuracy of the Gp model, constructed from a learning sample. This accuracy is represented by the predictivity coefficient

Q_2. It corresponds to the classical coefficient of determination R^2 for a test sample, i.e. for prediction residuals :

$$Q_2(Y,\hat{Y}) = 1 - \frac{\sum_{i=1}^{n_{\text{test}}}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n_{\text{test}}}\left(\bar{Y} - Y_i\right)^2},$$

where Y denotes the n_{test} true observations of the test set and $\bar{Y}$ is their empirical mean. $\hat{Y}$ represents the Gp model predicted values. To obtain different values of Q_2, we simulate different learning samples with varying size n. For each size n, a Latin Hypercube Sample of the inputs is simulated (McKay et al. [19]) to give the matrix X_s (n rows, d columns). Then, the test function is evaluated on the n data points to constitute (X_s, Y_s) and a conditional Gp model is built on each learning sample. For each Gp model built, the predictivity coefficient Q_2 is estimated on a new test sample of size 10000 and the two sensitivity indices S_i and $\mu_{\tilde{S}_i}$ are computed. For each value of the learning sample size n, all this procedure, i.e. Gp modeling and estimation of sensitivity indices, is done 100 times. Consequently, the empirical mean, 0.05-quantile and 0.95-quantile of S_i and $\mu_{\tilde{S}_i}$ are computed for same values of learning sample size n, and similar approximate values of Q_2.

3.5.2 Test on the g-function of Sobol

First, an analytical function called the g-function of Sobol is used to compare the Sobol indices S_i based on the predictor and the Sobol indices $\mu_{\tilde{S}_i}$ based on the global Gp model. The g-function of Sobol is defined for d inputs $(X_1, \ldots, X_d)$ uniformly distributed on $[0,1]^d$:

$$g_{\text{Sobol}}(X_1,\ldots,X_d) = \prod_{k=1}^{d} g_k(X_k) \text{ where } g_k(X_k) = \frac{|4X_k - 2| + a_k}{1 + a_k} \text{ and } a_k \geq 0.$$

Because of its complexity (considerable nonlinear and non-monotonic relationships) and to the availability of analytical sensitivity indices, it is a well known test example in the studies of global sensitivity analysis algorithms (Saltelli et al. [24]). The importance of each input X_k is represented by the coefficient a_k. The lower this coefficient a_k, the more significant the variable X_k. The theoretical values of first order Sobol indices are known :

$$S_i = \frac{\frac{1}{3(1+a_i)^2}}{\prod_{k=1}^{d} \frac{1}{3(1+a_k)^2}} \text{ for } i = 1, \ldots, d.$$

For our analytical test, we choose $d = 5$ and $a_k = k$ for $k = 1, \ldots, 5$..

Let us recall that we study only first order sensitivity indices. For each input X_i, the convergence of S_i and $\mu_{\tilde{S}_i}$ in function of the predictivity coefficient Q_2 is illustrated in figure 3.1. The convergence of sensitivity index estimates to their exact values in function of the metamodel predictivity is verified. In practical situations, a metamodel with a predictivity lower than 0.7 is often considered as a poor approximation of the computer code. Table 3.1 shows the connection between the learning sample size n and the predictivity coefficient Q_2. As the simulation of a learning sample and its Gp modeling are done 100 times for each value of n, the mean and the standard deviation of Q_2 are indicated. Figure 3.1 also shows how the global Gp model outperforms the predictor-only model by showing smaller confidence intervals for the five sensitivity indices.

To sum up the convergence of the indices for the different inputs, it can be useful to consider the error between the theoretical values of Sobol indices S_i^{theo} and the estimated ones in L_2 norm :

$$\text{err}_{L_2} = \sum_{i=1}^{d}(S_i^{theo} - \hat{S}_i)^2 \qquad (13)$$

where $\hat{S}_i$ denotes the indices estimated with one of the two methods ($\hat{S}_i = S_i$ or $\hat{S}_i = \mu_{\tilde{S}_i}$). Figure 3.2 illustrates this convergence in function of the learning sample size n and in function of the predictivity coefficient Q_2.

Learning sample size n	Predictivity coefficient Q_2	
	Mean	Standard deviation
25	0.67	0.21
35	0.88	0.09
45	0.96	0.02
55	0.98	0.01
65	0.98	6.10^{-3}
75	0.99	4.10^{-3}
85	0.99	3.10^{-3}
95	0.99	2.10^{-3}

TAB. 3.1 – **Connection between the learning sample size n and the predictivity coefficient Q_2 (g-Sobol function).**

From Figure 3.2, we conclude that the sensitivity indices defined using the global Gp model ($\mu_{\tilde{S}_i}$) are better in mean than the one estimated with the predictor only (S_i). This difference between the two approaches is especially significant for high values of Sobol indices like the indices related to the first input (S_1 and $\mu_{\tilde{S}_1}$). For lower indices, these two approaches give in mean the same results. Even if the two sensitivity indices seem to have the same rate of convergence in function of n or Q_2, it is important to notice that the second approach is more robust. Indeed, $\mu_{\tilde{S}_i}$ has a lower sampling deviation and variability than S_i. Besides, this higher robustness is more significant when the accuracy of the metamodel is weak ($Q_2 < 0.8$). So, taking into account the covariance structure of the Gp model appears useful to reduce the variability of the estimation of the sensitivity index.

3.5.3 Test on Ishigami function

We now consider another analytical function currently used in sensitivity studies (Saltelli et al. [24]), the Ishigami function, where each of the three input random variables (X_1, X_2, X_3) follows a uniform probability distribution on $[-\pi, +\pi]$:

$$f_{\text{Ishig}}(X_1, X_2, X_3) = \sin(X_1) + 7\sin^2(X_2) + 0.1X_3^4\sin(X_1)$$

The theoretical values of first order Sobol indices are known :

$$\begin{cases} S_1 = 0.3139 \\ S_2 = 0.4424 \\ S_3 = 0 \end{cases}$$

Like for the g-function of Sobol, the error with the theoretical values of Sobol indices in L_2 norm is computed for the two approaches for different learning sample size n and consequently for different values of Q_2. As before (Eq. (13)), the diagrams of convergence are shown in figure 3.3.

As observed for the g-function of Sobol, the indices defined with the global model are still more robust and less variable particularly for low values of Q_2. However, the difference between the mean of the two indices is not significant. For high values of the Gp model accuracy ($Q_2 > 0.8$), the two

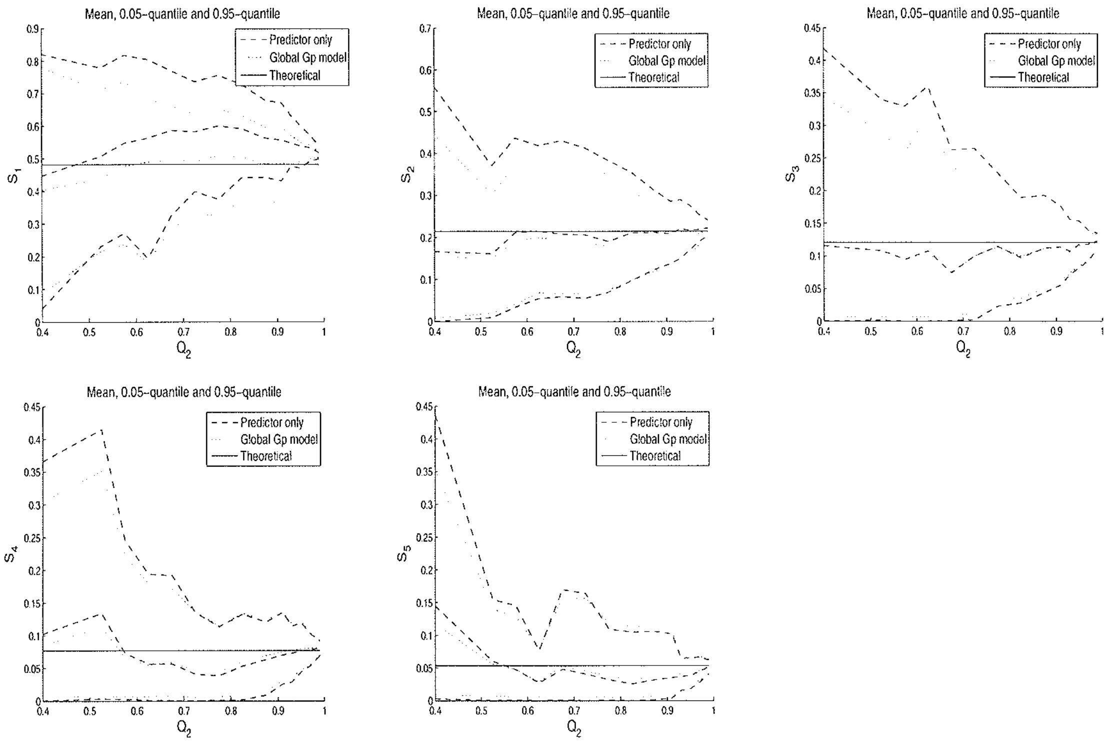

Fig. 3.1 – **Convergence of sensitivity indices in function of the predictivity coefficient Q_2 (g-Sobol function).**

approaches give the same values but the first one (with only the predictor) remains easier to compute. So, the use of the covariance structure through the index $\tilde{S}_i$ seems to have a significant interest when the Gp metamodel is inaccurate or when few data are available to avoid too much variability of the estimated indices.

3.5.4 Construction of confidence intervals for sensitivity indices

As well as being more robust in mean, the index defined with the second approach $\tilde{S}_i$ has the advantage to have a variance easy to compute. More generally, it is possible to build a confidence interval of any level for this sensitivity index, using the methodology described in section 3.4.3 to simulate its distribution. This estimation of the uncertainty on the estimation of Sobol indices is another advantage of using the global Gp model : in practical cases with small learning sample size, only one Gp model is constructed. The predictivity coefficient Q_2 can be estimated by cross-validation or leave-one-out, and if Q_2 shows a low predictivity (typically less than 0.8), we wish to have some confidence in the estimation of Sobol indices computed from the Gp model. Contrary to Gp model, other metamodels do not allow to directly estimate the Sobol indices uncertainties due to the model uncertainties.

To illustrate this, let us consider again the g-function of Sobol. Like in the previous section 3.5.2, we consider different sizes of the learning sample (from $n = 20$ to $n = 50$). For each value of n, we build a conditional Gp model and we control its accuracy estimating the Q_2 on a test sample. We simulate the distribution of $\tilde{S}_i$ to obtain the empirical 0.05 and 0.95-quantiles and consequently an empirical 90%-confidence interval. Then, we check if the theoretical values of Sobol indices belong to the empirical 90%-confidence interval. We repeat this procedure 100 times for each size n. Therefore, we are able to estimate the real level of our confidence interval and compare it to the 90% expected. The real levels obtained in mean for any size n and each input are presented in Table 3.2.

For the high values of Sobol indices (S_1 and S_2 for example), the observed levels of the 90%-

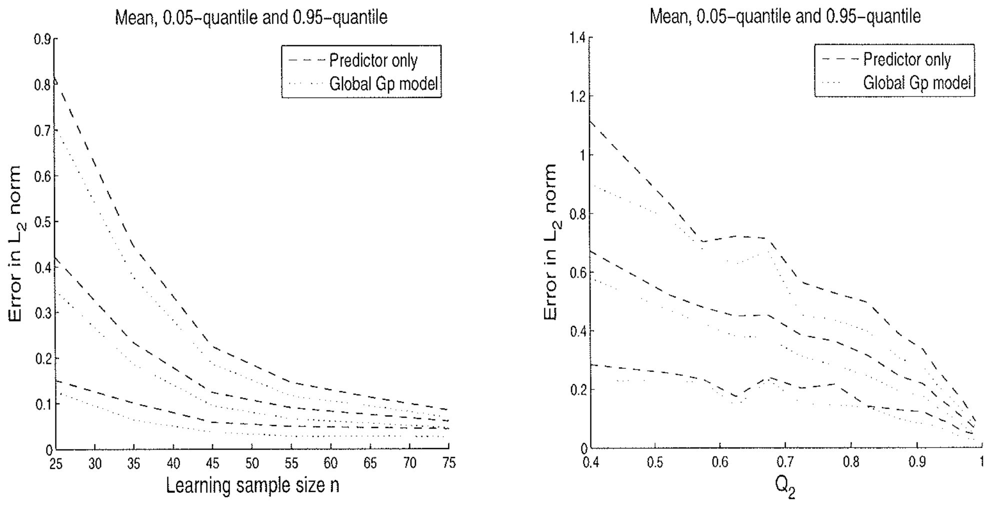

Fig. 3.2 – **Error in L_2 norm for sensitivity indices in function of n and Q_2 (g-Sobol function).**

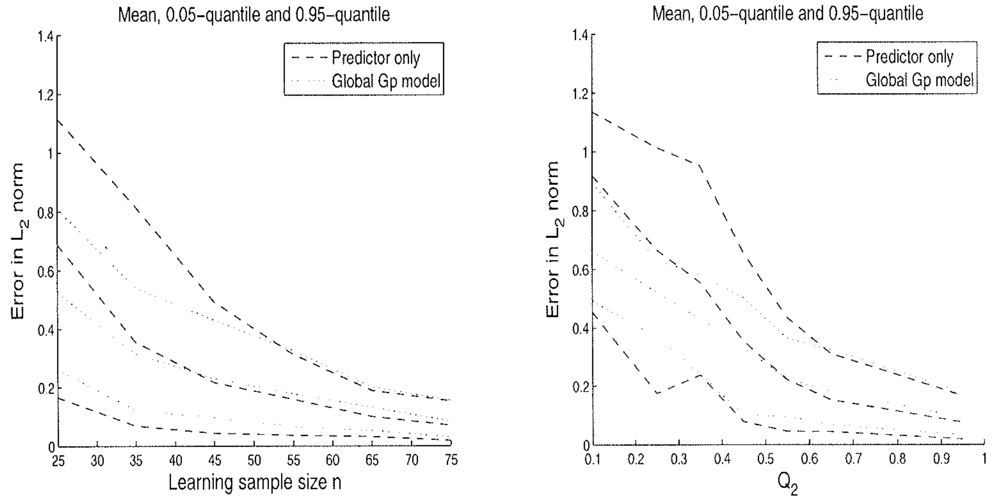

Fig. 3.3 – **Error in L_2 norm for sensitivity indices in function of n and Q_2 (Ishigami function).**

Variable	Theoretical value of Sobol index	Mean of $\mu_{\hat{S}_i}$	Observed level of the empirical confidence interval
X_1	0.7164	0.7341	0.9381
X_2	0.1791	0.1574	0.9369
X_3	0.0237	0.0242	0.5830
X_4	0.0072	0.0156	0.8886
X_5	0.0001	0.0160	0.0674

TAB. 3.2 – **Real observed level of the empirical 90%-confidence interval built with the Gp model for the Sobol index of each input parameter (g-Sobol function).**

confidence interval built from the simulation of the distribution of $\tilde{S}_i$ are really satisfactory and close to the expected level. In this case, the use of the global Gp model which gives confidence intervals for Sobol indices has a significant interest. On the other hand, for very low indices (close to zero), the Gp metamodel overestimates the Sobol indices. It explains the inaccuracy of the confidence interval. Indeed, without a procedure of inputs selection, each variable appears in the Gp metamodel and in its covariance. Taking into account the variance leads to give a minimal bound for the influence of all the variables and consequently to overestimate the lowest Sobol indices. This tendency is confirmed by the observation of the mean of $\mu_{\tilde{S}_i}$ estimated for the three last inputs in Table 3.2.

We can make the same study with the Ishigami function for $n = 30$ to $n = 130$ which induces a Q_2 varying from 0.5 to 0.95. As all the procedure (i.e. learning sample simulation, Gp modeling and sensitivity analysis) is repeated 100 times for each size n, the convergence of the observed level of the empirical 90%-confidence interval can be observed in function of n. Similarly, we can study this convergence in function of Q_2. Figure 3.4 shows all these diagrams of convergence. As previously remarked on the g-function of Sobol, the 90%-confidence intervals are efficient for the high values of Sobol indices (S_1 and S_2 for example). For these indices, the observed level of the confidence interval converges to theoretical level 0.9. We can also notice that the predictivity quality of the Gp modeling which is required to obtain accurate confidence interval corresponds approximately to $Q_2 > 0.80$. However, we judge that for $Q_2 > 0.6$, the error is not too strong and the obtained 90%-confidence interval can be considered as a reliable and useful information. On the other hand, for very low indices (close to zero), the problem of overestimating the Sobol indices still damages the accuracy of the interval confidence for any size n and any Q_2. This remark is particularly true when the index is equal to zero (for example S_3).

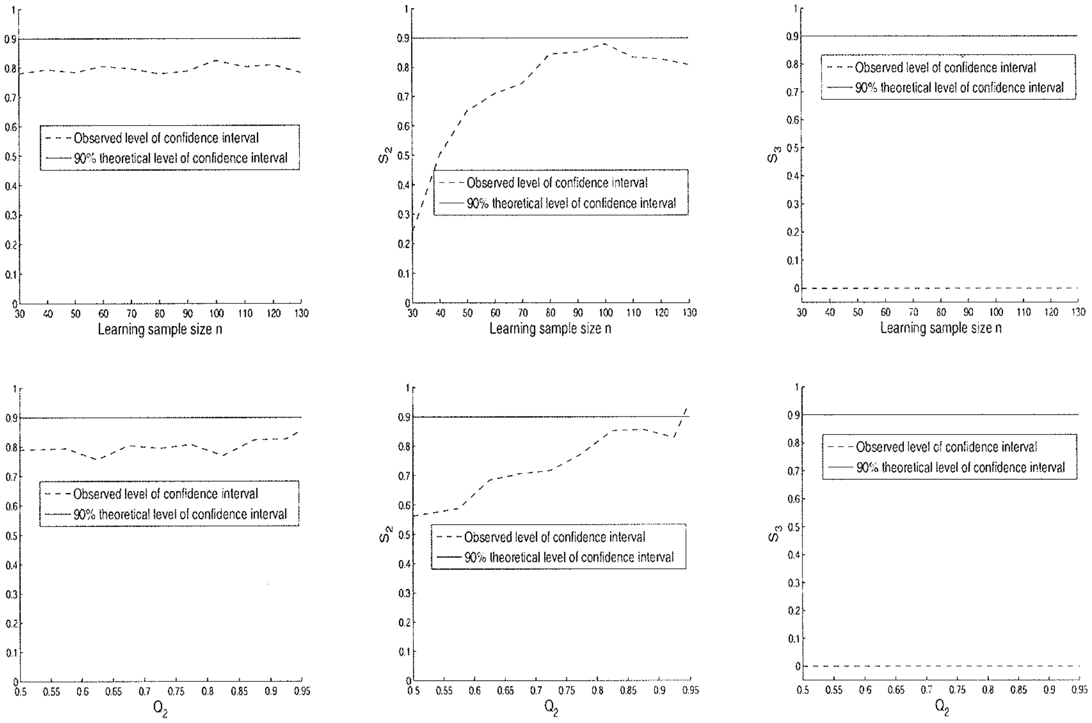

Fig. 3.4 – **Convergence of the observed level of the empirical** 90%**-confidence in function of** n **and** Q_2 **(Ishigami function).**

3.5.5 Application on an hydrogeologic transport code

The two approaches to compute the Sobol indices are now applied to the data obtained from the modeling of strontium 90 (noted ^{90}Sr) transport in saturated porous media using the MARTHE software (developed by BRGM, France). The MARTHE computer code models flow and transport equations in three-dimensional porous formations. In the context of an environmental impact study, the MARTHE computer code has been applied to the model of ^{90}Sr transport in saturated media for a radwaste temporary storage site in Russia (Volkova et al. [29]). One of the final purposes is to determine the short-term evolution of ^{90}Sr transport in soils in order to help the rehabilitation decision making. Only a partial characterization of the site has been made and, consequently, values of the model input parameters are not known precisely. One of the first goals is to identify the most influential parameters of the computer code in order to improve the characterization of the site in a judicious way. To realize this global sensitivity analysis and because of large computing time of the MARTHE code, a Gp metamodel is built on the basis of a first learning sample.

Data presentation

The 20 uncertain model parameters are permeability of different geological layers composing the simulated field (parameters 1 to 7), longitudinal dispersivity coefficients (parameters 8 to 10), transverse dispersivity coefficients (parameters 11 to 13), sorption coefficients (parameters 14 to 16), porosity (parameter 17) and meteoric water infiltration intensities (parameters 18 to 20). To study sensitivity of the MARTHE code to these parameters, 300 simulations of these 20 parameters have been made by the LHS method. For each simulated set of parameters, MARTHE code computes transport equations of ^{90}Sr and predicts the evolution of ^{90}Sr concentration for year 2010. For each simulation, the output we consider is the value of ^{90}Sr concentration, predicted for year 2010, in a piezometer located on the waste repository site.

Gp modeling and computation of Sobol indices

To model the concentration in the piezometer predicted by MARTHE code in 2010 in function of the 20 input parameters, we fit a Gp metamodel conditionally to 300 simulations of the code. The regression and correlation parameters of the Gp model are estimated by maximum likelihood and a procedure of input selection is used. The input variables introduced in the metamodel are the sorption coefficient of the upper layer (parameter 14 denoted $kd1$), an infiltration intensity (parameter 20 denoted $i3$) and the permeability of the upper layer (parameter 1 denoted $per1$). The accuracy of the Gp model is checked with the estimation of Q_2 by a cross validation on the learning sample. The predictivity coefficient estimated is : $Q_2 = 0.92$. From previous study (Marrel et al. [15]), we have found that the linear regression gives a $Q_2 = 0.69$ and the metamodel based on boosting of regression trees gives a $Q_2 = 0.83$. From laboratory measures and bibliographical information, prior distributions have been determined for the inputs $kd1$, $i3$ and $per1$ and are respectively a Weibull, a trapezoidal and a uniform distributions. The parameters of these distributions has been estimated or fixed a priori. Then, using the global Gp model, the Sobol indices defined by $\mu_{\tilde{S}_i}$ are computed (Eq. (12)) as well as the standard deviation $\sigma_{\tilde{S}_i}$ and the 90%-confidence interval associated (cf. methodology 3.4.3). The results are presented in Table 3.3, with the Sobol indices obtained with the predictor-only approach and with the boosting predictor. The use of Gp model gives a better predictivity than the boosting of regression trees (respectively $Q_2 = 0.92$ and $Q_2 = 0.83$) and consequently a more accurate estimation of Sobol indices. Besides, the Sobol indices estimated with the boosting model do not even belong to the confidence intervals given by the Gp model. Even if the sensitivity indices based on the predictor only and the ones estimated with the whole Gp model are very close, the second approach has the advantage to give confidence intervals and consequently to have a more rigorous analysis.

Without considering their interactions, the 3 inputs $kd1$, $i3$ and $per1$ explained nearly 90% of the total variance of the output and the most influent input is clearly $kd1$, followed by $i3$ and $per1$. So, $kd1$ is the most important parameter to be characterized in order to reduce the variability of the concentration predicted by MARTHE code. Using the whole Gp model, we also have an indication of the accuracy of Sobol indices. The standard deviation of the indices are small and increase the

input variable	Boosting of regression trees	Predictor only (Gp model)	Whole Gp model		
	S_i	S_i	$\mu_{\tilde{S}_i}$	$\sigma_{\tilde{S}_i}$	90%-confidence interval
per1	0.03	0.081	0.078	0.020	[0.046 ; 0.112]
kd1	0.90	0.756	0.687	0.081	[0.562 ; 0.825]
i3	0.03	0.148	0.132	0.022	[0.100 ; 0.170]

TAB. 3.3 – **Estimated Sobol indices, associated standard deviation and confidence intervals for MARTHE data.**

confidence in the value estimated ($\mu_{\tilde{S}_{kd1}}$, $\mu_{\tilde{S}_{i3}}$ and $\mu_{\tilde{S}_{per1}}$). Moreover, the very small overlap of the 90%-confidence interval of the 3 indices indicates that the order of influence of the inputs is well determined and strongly confirms the predominance of $kd1$. So, the confidence intervals and the standard deviation obtained with the whole Gp model give more confidence in the interpretation of Sobol indices.

Taking into account the variability of the Gp model via its covariance structure gives more robustness to the results and their analysis. However, this increase of precision and confidence has a numerical cost. Indeed, the number of numerical integrals being computed is of order $O(dn^2)$ where d is the number of inputs and n the number of simulations, i.e. the learning sample size. The numerical cost depends also on the numerical precision required for the approximation of the integrals. Moreover, a high precision is often essential to provide the robustness of the computation of Sobol indices, especially when the distribution of the inputs is narrow and far from the uniform distribution (like the Weibull distribution of $kd1$). In this last case, it can be judicious to adapt the numerical scheme in order to increase the precision in the region of high density.

3.6 CONCLUSION

We have studied the Gaussian process metamodel to perform sensitivity analysis, by estimating Sobol indices, of complex computer codes. This metamodel is built conditionally to a learning sample, i.e. to n simulations of the computer code. The Gp model proposes an analytical formula which can be directly used to derive analytical expressions of Sobol indices. Indeed, in the case of independent inputs and with our choice of regression and covariance functions, the formula of Gp model leads to one and two-dimensional numerical integrals, avoiding a large number of metamodel predictor evaluations in Monte Carlo methods. The use of Gp model instead of other metamodel is therefore highly efficient. Another advantage of Gp metamodel stands in using its covariance structure to compute Sobol indices and to build associated confidence intervals, by using the global stochastic model including its covariance.

On analytical functions, the behavior and convergence of the Sobol index estimates were studied in function of the learning sample size n and the predictivity of the Gp metamodel. This analysis reveals the significant interest of the global stochastic model approach when the Gp metamodel is inaccurate or when few data are available. Indeed, the use of the covariance structure gives sensitivity indices which are more robust and less variable. Moreover, all the distribution of the sensitivity index (defined as a random variable) can be simulated following an original algorithm. Confidence intervals of any level for the Sobol index can then be built. In our tests, the observed level of the interval was compared to the expected one on analytical functions. For the highest values of Sobol indices and under the hypothesis of a Gp metamodel with a predictivity coefficient larger than 60%, the confidence intervals are satisfactory. In this case, the use of the global Gp model which gives confidence intervals for Sobol indices has a significant interest. The only drawback is that the use of covariance structure has a tendency to give a minimal bound for the influence of all the variables and consequently to overestimate the lowest Sobol indices and to give inaccurate confidence intervals for very low indices (close to zero).

The use of covariance structure was also illustrated on real data, obtained from a complex hydrogeological computer code, simulating radionuclide groundwater transport. This application confirmed the interest of the second approach and the advantage of Gp metamodel which, unlike other efficient metamodels (neural networks, regression trees, polynomial chaos, ...), gives confidence intervals for the estimated sensitivity indices. The same approach based on the use of the global Gp metamodel can be used to make uncertainty propagation studies and to estimate the distribution of the computer code output in function of the uncertainties on the inputs.

3.7 ACKNOWLEDGMENTS

This work has been done with Béatrice Laurent (INSA Toulouse) and Olivier Roustant (Ecole des Mines de Saint-Etienne).

3.8 REFERENCES

[1] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear programming.* John Wiley & Sons, Inc, 1993.

[2] G.E. Box and N.R. Draper. *Empirical model building and response surfaces.* Wiley Series in Probability and Mathematical Statistics. Wiley, 1987.

[3] V.C.P. Chen, K-L. Tsui, R.R. Barton, and M. Meckesheimer. A review on design, modeling and applications of computer experiments. *IIE Transactions*, 38 :273–291, 2006.

[4] W. Chen, R. Jin, and A. Sudjianto. Analytical metamodel-based global sensitivity analysis and uncertainty propagation for robust design. *Journal of Mechanical Design*, 127 :875–886, 2005.

[5] J-P. Chilès and P. Delfiner. *Geostatistics : Modeling spatial uncertainty.* Wiley, New-York, 1999.

[6] N.A.C. Cressie. *Statistics for spatial data.* Wiley Series in Probability and Mathematical Statistics. Wiley, 1993.

[7] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416) :953–963, 1991.

[8] K-T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments.* Chapman & Hall/CRC, 2006.

[9] J.C. Helton, J.D. Johnson, C.J. Salaberry, and C.B. Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering and System Safety*, 91 :1175–1209, 2006.

[10] B. Iooss, F. Van Dorpe, and N. Devictor. Response surfaces and sensitivity analyses for an environmental model of dose calculations. *Reliability Engineering and System Safety*, 91 :1241–1251, 2006.

[11] J. Jacques. *Contributions à l'analyse de sensibilité et à l'analyse discriminante généralisée.* Thèse de l'Université Joseph Fourier, Grenoble 1, 2005.

[12] J. Kleijnen. Sensitivity analysis and related analyses : a review of some statistical techniques. *Journal of Statistical Computation and Simulation*, 57 :111–142, 1997.

[13] J. Kleijnen and R.G. Sargent. A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research*, 120 :14–29, 2000.

[14] S.N. Lophaven, H.B. Nielsen, and J. Sondergaard. DACE - A Matlab kriging toolbox, version 2.0. Technical Report IMM-TR-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark, 2002. <http ://www.immm.dtu.dk/~hbn/dace>.

[15] A. Marrel, B. Iooss, F. Van Dorpe, and E. Volkova. An efficient methodology for modeling complex computer codes with gaussian processes. *Submitted in Computational Statistics and Data Analysis*, 2007.

[16] M. Marseguerra, R. Masini, E. Zio, and G. Cojazzi. Variance decomposition-based sensitivity analysis via neural networks. *Reliability Engineering and System Safety*, 79 :229–238, 2003.

[17] J.D. Martin and T.W. Simpson. On the use of kriging models to approximate deterministic computer models. *AIAA Journal*, 43 :4 :853–863, 2005.

[18] G. Matheron. *La Théorie des Variables Régionalisées, et ses Applications.* Les Cahiers du Centre de Morphologie Mathématique de Fontainebleau, Fascicule 5. Ecole des Mines de Paris, 1970.

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

[20] J.E. Oakley and A. O'Hagan. Probabilistic sensitivity analysis of complex models : a bayesian approach. *Journal of the Royal Statistical Society, Series B*, 66 :751–769, 2004.

[21] A. O'Hagan. Bayesian analysis of computer code outputs : A tutorial. *Reliability Engineering and System Safety*, 91 :1290–1300, 2006.

[22] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning.* MIT Press, 2006.

[23] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4 :409–435, 1989.

[24] A. Saltelli, K. Chan, and E.M. Scott, editors. *Sensitivity analysis.* Wiley Series in Probability and Statistics. Wiley, 2000.

[25] T. Santner, B. Williams, and W. Notz. *The design and analysis of computer experiments.* Springer, 2003.

[26] I.M. Sobol. Sensitivity estimates for non linear mathematical models. *Mathematical Modelling and Computational Experiments*, 1 :407–414, 1993.

[27] B. Sudret. Global sensitivity analysis using polynomial chaos expansion. *To appear in Reliability Engineering and System Safety*, 2007.

[28] E. Vazquez, E. Walter, and G. Fleury. Intrinsic kriging and prior information. *Applied Stochastic Models in Business and Industry*, 21 :215–226, 2005.

[29] E. Volkova, B. Iooss, and F. Van Dorpe. Global sensitivity analysis for a numerical model of radionuclide migration from the RRC "Kurchatov Institute" radwaste disposal site. *To appear in Stochastic Environmental Research and Risk Assesment*, 2007.

[30] R. von Mises. *Mathematical Theory of Probability and Statistics.* Academic Press, 1964.

[31] 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.