Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Cotutelle internationale avec :

Présentée et soutenue par :
Ekaterina SERGIENKO

Le lundi 12 novembre 2012

Titre :

Adapted reservoir characterization for monitoring and uncertainty analysis of CO2 storage

---

ED MITT : Domaine Mathématiques : Mathématiques appliquées

Unité de recherche :
Statistique et Probabilités

Directeur(s) de Thèse :

Fabrice GAMBOA
Daniel BUSBY

Rapporteurs :

Olivier ROUSTANT
Gérard BIAU

Autre(s) membre(s) du jury :

Yann LE GALLO
Jean-Claude FORT
Thierry KLEIN
Jan Dirk JANSEN

Acknowledgements

This thesis would have not been realized without expertise and help of the people who were around me and who were supporting me during all the three years of my work.

First, I deeply appreciate the contribution of my scientific advisor Fabrice Gamboa. For me it was an honor to work with him. Moreover, it was always a pleasure to have our rare meetings and conversations that were always very fruitful and brought a lot of motivation in my research. At the same time, I am very grateful to my co-supervisor Daniel Busby who initiated this PhD project. I would like to thank him for his invaluable scientific and non-scientific guidance, for his research ideas and sparing his time for discussion throughout all the three years of my work.

Next, I would like to thank my colleagues from EDF Bertrand Iooss, Paul Lemaitre, Aurelie Arnaud, and Nicolas Bousquet. My work has benefited from your suggestions, mathematical insight and technical excellence. This collaboration work inspired me for new research directions.

I want to thank all my colleagues from IFP Energies Nouvelles for interesting discussions and collaboration. In particular, I would like to thank Ratiba, Samir, Zyed, Franck, Francois, Setayesh, Leila and Marius for the friendly atmosphere in the office from the first day of my arrival.

I also thank all my friends from all around the world. Thank you for your support and encouragement.

Finally, I express my deep gratitude to my mother for her unconditional support, constant interest and her belief in my success. In addition, I want to thank Pavel for his love, patience and support.

Abstract

Risk analysis of CO_2 geological storage involves the simulation of the dynamics of the storage process and the evaluation of the probability of the possible leakage events. The approach followed here focuses on Gaussian Process response surface modelling in order to reduce the number of calls to the expensive reservoir simulator. Three major problems related to uncertainty analysis of CO_2 storage are addressed:

1. Injection well placement
2. Reliability estimation
3. Reliability sensitivity analysis

To tackle the first problem we provide a response surface method to handle discrete parameters (well positions) and discrete functional outputs to treat responses varying trough time (reservoir pressure evolutions). In addition, we introduce a new method for modelling functional outputs based on curves characterization and involving shape invariant model. To address the reliability problem, we introduce a subset simulation algorithm linked with the Gaussian Process model. It involves adaptive experimental design refinement and the model updating. To solve the last problem we suggest a new method for reliability sensitivity analysis. It is based on a perturbation of a probability distribution of input variables in order to evaluate which one contributes the most in the variability of the failure probability.

All the proposed methods have been numerically tested on analytical and CO_2 storage examples.

Key words: meta-modelling • kriging • discrete parameters • times series • functional ouputs • adaptive sampling • reliability analysis • failure probability • reliability sensitivity analysis • CO_2 reservoir storage • well placement

Résumé

L'analyse de risques du stockage géologique de CO_2 consiste à simuler la dynamique du processus de stockage et à évaluer la probabilité de fuites. L'approche proposée dans ce travail consiste à utiliser des surfaces de réponses basées sur les processus Gaussiens, cela permet de réduire le grand nombre d'appels au simulateur de réservoir nécessaire à cette analyse. Dans cette thèse des méthodes innovantes sont etudiées pour resoudre les problèmes suivants:

1. Emplacement des puits d'injection
2. Estimation de la fiabilité
3. Analyse de sensibilité fiabiliste

Pour résoudre le premier problème nous proposons une méthode de surface de réponse pour gérer les paramètres discrets (positions des puits) et les sorties fonctionnelles discrètes (évolution de pression du réservoir). Par ailleurs, nous introduisons une nouvelle méthode pour la modélisation des réponses variées dans le temps. Pour cela, la caractérisation des courbes est effectuée en utilisant des modèles à forme invariante. Pour le problème de fiabilité, nous avons développé une approche combinant la méthode de réduction d'ensemble et le krigeage. Un échantillonnage adaptatif est construit afin d'améliorer itérativement l'estimation de la probabilité de défaillance du modèle. Pour répondre au dernier problème, nous proposons une méthode pour l'analyse de sensibilité fiabiliste. Elle est basée sur une perturbation de la distribution de probabilité des variables d'entrée afin de trouver les facteurs qui contribuent le plus à la variabilité de la probabilité de défaillance.

Toutes les méthodes proposées ont été testées numériquement sur des exemples analytiques et des cas test de stockage de CO_2.

Mots-clés: méta-modélisation • krigeage • paramétres discréts • series temporelles • sorties fonctionnelles • échantillonage adaptative • analyse de fiabilité • probabilité de défaillance • analyse de sensibilité • stockage du CO_2 • placement de puits

Contents

List of Figures

Chapter 1

Introduction

Contents

1.1 Problem description

Carbon Capture and Storage (CCS) stands for the collection of CO_2 from industrial sources and its injection into deep geological formations for a permanent storage. There are three possible options for injection: unmined coalbed formations, saline aquifers and depleted oil and gas reservoirs [Polson et al., 2009]. The last one could benefit with the enhanced oil recovery. The overview of the geological storage options and the life cycle of a typical storage project is shown in Figure 1.1 [Benson, 2006].

Nevertheless, the following principal environmental question arises: what is the probability that CO_2 will remain underground for hundreds to thousands of years after its capture and injection into a storage formation?

The primary risk of CO_2 geological storage is unintended gas leakage from the storage reservoir. There are several principal possibilities for the leakage. The first one is CO_2 leakage through a well: either through an injection well or through the nearby wells (if they were not properly sealed). The second is a leakage from the storage formation up to the atmosphere through a fault or a fracture. In this

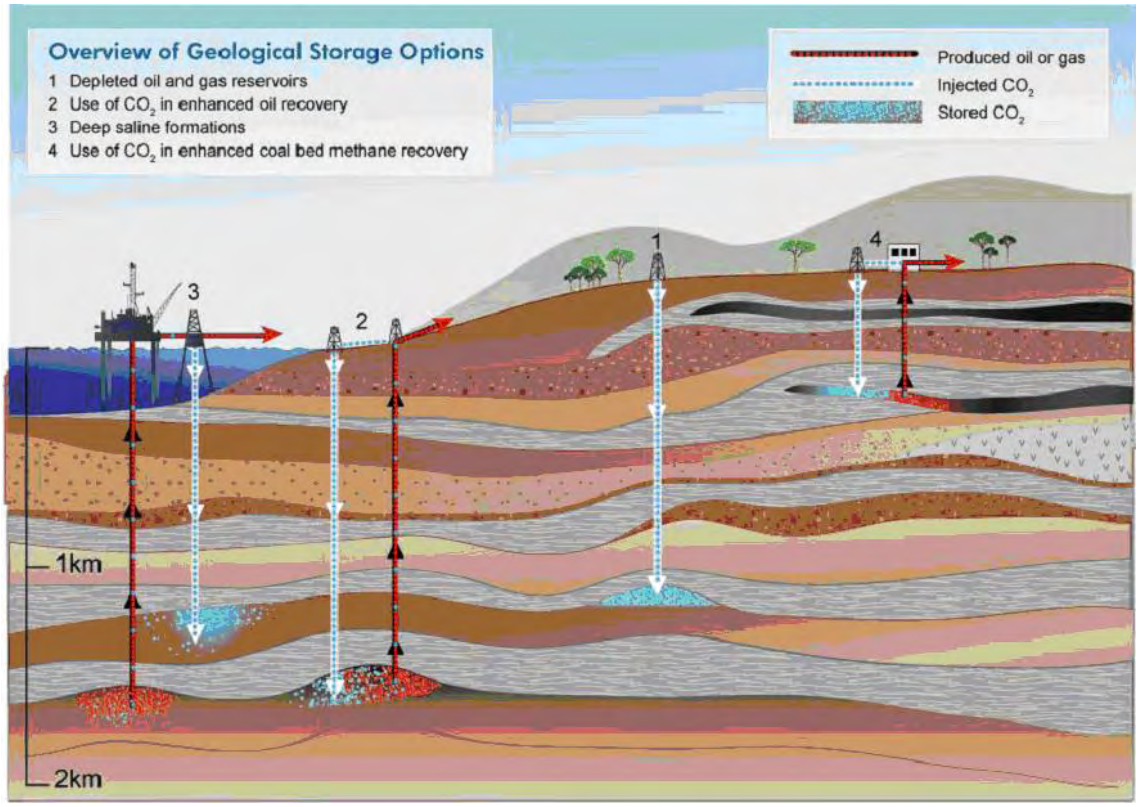


Figure 1.1: CO_2 storage option and its life cycle IPCC, 2006)

work, we focus on the second leakage possibility. These fractures may be of different origins. Some of them were not properly identified in the original geological model, others could appear by cap rock fracturation during the injection. This can happen when the reservoir pressure is higher than the caprock fracture pressure. There is also a possibility of leakage through the caprock due to the capillary effects. All the possible leakage scenarios could be a significant risk to the environment [Bowden and Rigg, 2004; Le Guen et al., 2008].

Numerical modelling and simulation is an integral component of CO_2 storage assessment and monitoring. The technology is adapted from the petroleum industry. The reservoir simulation models are constructed based on reservoir and production data. They are used to predict and analyze CO_2 plume distribution and reservoir pressure development during injection and the storage periods. Moreover, numerical simulation models can be used to forecast CO_2 storage performance and to evaluate the risks of possible leakage. In this study, we use COORES^{TM} reservoir simulator [Tillier et al., 2007; Trenty et al., 2006], which is a multi-component three-phase and 3-D fluid flow simulator in heterogeneous porous media, accounting for mineralogy

changes as well as relative permeability and capillary pressure effects.

Reservoir simulation studies involves a sequence of various activities: data collection, model building, simulation runs and analysis of the results. However, data used for the reservoir characterization are often noisy. Moreover, gathering sufficient information to build an accurate reservoir model is an expensive and usually an impossible task. The main source of uncertainty is the collected geological data, for example, the reservoir porosity and the reservoir permeability, placement of faults and fractures and the reservoir geometry. For example, when the gas is stored in a saline aquifer, there are almost no available reliable hand-on data about the future potential storage reservoir.

Uncertainty analysis has been recognized as a principal part of safety and risk assessment. Generally speaking, when the main sources of uncertainties have been identified, Uncertainty Analysis (UA) consists in quantifying the uncertainty in the model output resulting from uncertainties in the model inputs. At the same time, Sensitivity Analysis (SA) aims at identifying the contributions of each model uncertain input to the variability of the model output. It helps to determine the most influential input parameters and to analyze how these parameters affect the model output.

Uncertainty analysis is also performed to assess the reliability of the system. Reliability is measured by the failure probability. For a case of CO_2 storage, a failure probability can be, for example, the probability of exceeding the caprock fracturing pressure by reservoir pressure during injection. If we denote $\mathrm{P}_{reservoir}$ the reservoir pressure and $\mathrm{P}_{fracture}$ the caprock fracturing pressure, then we consider the following failure probability:

$$p_f = \mathrm{P}(\mathrm{P}_{reservoir} \geq \mathrm{P}_{fracture}).$$

In practice, uncertainty and sensitivity analysis require a large number of reservoir simulator runs to explore all the input parameters space. However, higher accuracy of a simulator involves a higher simulation time. One simulator run can take from few minutes up to several hours or even days. Therefore, when the simulation time becomes too high, uncertainty analysis may become unfeasible. Consequently, response surface models have recently gained attention. These models are intended to approximate the output from a simulator. The main advantage of the response surface model is that evaluation time is almost negligible compared to using the

simulator.

In this work, we focus on Gaussian Process (GP) response surface models, also known as *kriging*. The GP response surface modelling was originally introduced in the field of geostatistics by Krige in the 1950's [Krige, 1951] and formalized in 1960's by Matheron [1963]. Sacks et al. [1989] proposed the statistical approach to uncertainty analysis of complex computer codes referred to as the Design of Computer Experiments. The theory and the algorithms are formalized in Sacks et al. [1989]; Santner et al. [2003]; Welch et al. [1992]. In a nutshell, departing from some simulator runs at carefully chosen input parameters configurations, called the experimental design, we build an approximation of a reservoir simulator output. Making a prediction at an unknown input is much faster than using the simulator. Therefore, uncertainty and sensitivity analysis become affordable.

In this thesis, we make a review of the GP response surface modelling with particular attention to handling discrete parameters and addressing discrete functional outputs of a dynamic simulator. We also propose a new sequential adaptive approach for experimental design with the objective of a reliable estimation of the failure probability. In addition, the subsequent sensitivity analysis of the failure probability is presented.

1.2 Objectives of the thesis

The objective of the thesis is to provide a comprehensive methodology for risk and uncertainty analysis for the projects of CO_2 storage. This methodology is based on GP response surface modelling and consists in the following steps:

1. First, we propose a new CO_2 injection well placement technique. By incorporating discrete parameters, such as well coordinates, we can place the well under a condition of minimization of a possible leakage risk. By implementing an approach for discrete functional outputs, we can consider numerous possible well positions with a low CPU time cost.

2. Then, we propose a new adaptive refinement method for a GP response surface model in order to provide a reliable estimation of the probability of a possible leakage. By sampling adaptively close to the failure region, we can update the existing GP response surface model. The new model is designed specifically for

the estimation of the failure probability. Consequently, the relevant estimation of the risk and the reliability are computed.

3. The third step is the sensitivity analysis of the computed failure probability. This analysis indicates which input uncertain parameters contribute the most to the variability of the failure probability of the system.

This work was done with respect to CO_2 reservoir storage. However, all the developed methodologies could be applied to other areas of science and engineering applications involving complex numerical simulators for uncertainty, sensitivity and reliability analysis.

1.3 Thesis outline

Chapter 2 starts with a brief review of GP response surface modelling. A particular emphasis of this chapter is how to better handle discrete parameters in GP response surface modelling. When working with discrete parameters, the main difficulty is the design of an appropriate correlation function. The function should be valid for all the possible combinations of the discrete parameters. We present some recent attempts to handle discrete parameters and we investigate different correlation functions. In particular, we study the method proposed by Qian et al. [2008]. We have tested how the choice of the correlation function affects model predictivity on analytical test cases. We also present a reservoir case involving discrete parameters both of qualitative and quantitative nature.

Chapter 3 concentrates on GP response surface modelling for dynamic simulators. The output of such simulators consists of a sequence of outputs at different time-steps. Therefore, it represents discrete functional outputs with a finite support, or time series. The classical approach assumes distinct GP models for every time step. In this chapter, we propose two approaches to model a dynamic simulator. The first one is the extension of the method presented in Chapter 2, considering the time variable as an auxiliary discrete variable. Another approach is based on the Shape Invariant Model (SIM). For a given experimental design the simulator provides a set of discrete curves. The model assumes that we are working with a set of curves that have a common shape function. All other curves are obtained by a proper parametrical transformation. In this chapter, we provide an efficient

algorithm for estimating SIM transformation parameters. Then, we propose a prediction algorithm for the dynamic simulator. We present the numerical examples obtained for both methods.

Chapter 4 is devoted to the review of reliability methods. The objective of these methods is to estimate the failure probability of a system p_f. It can be achieved either by approximation methods or by simulation methods. We present the basic reliability methods and we provide a comparison of these methods on analytical test cases. We will describe more in detail subset simulation method. This method make possible estimating the failure probability even in high dimensions with rather low CPU time cost. Then, we introduce a new adaptive refinement technique, based on the subset simulation algorithm. We propose to incorporate the data generated at each level of the subset simulation algorithm. So that, we are sure that we are sampling from or close to the failure region. By combining the subset simulation algorithm and the proposed statistical criterion, we iteratively update the original GP response surface model in order to compute relevant estimation of the failure probability. The method was tested on an analytical and a CO_2 storage reservoir cases.

In Chapter 5 the reliability sensitivity analysis is considered. As soon as the failure probability p_f is estimated, it is important to know which parameters contribute the most to its variability. We propose a new moment independent sensitivity measure that is based on a perturbation of the original probability distribution of the input variables. By perturbing the original law by a low value δ independently for all the input variables, we can estimate the new failure probability p_δ. If the new value is considerably different from the original failure probability p_f, this means that the corresponding input variables has a high impact. Vice versa, if difference between the original and the new computed values is negligible, then the parameters has low or no impact on the failure probability. In this section, we will also introduce a new method to estimate the perturbed failure probability without additional function or simulator calls. The method is based on the importance sampling approach. We will also provide analytical and CO_2 storage examples.

To conclude, in the last Chapter 6 the overall conclusions are stated and some open points for possible future research are presented.

Chapter 2

Discrete parameters in GP response surface models

Contents

2.1 Introduction

Gaussian process (GP) based response surface modelling has been widely applied and has gained considerable attention in the literature over the recent years [Forrester et al., 2008; Sacks et al., 1989; Santner et al., 2003]. GP based response surface model is a statistical approximation of a deterministic function based on the information derived from an experimental design. GP models have been established as a core tool for response surface modelling mostly for two reasons. First, the response surface is

easy to compute and, second, it provides a full posterior probability distribution on interpolating functions [Sacks et al., 1989; Santner et al., 2003].

The objective of this chapter is to develop a predictive GP response surface model for the output of a simulator involving both discrete and continuous parameters. Discrete parameters may identify distinct classes, categories or levels. These parameters can describe factors of a qualitative or a quantitative nature. In the frame of computer experiments, one mostly works with continuous parameters, but there are sometimes parameters that are discrete by nature and could not be expressed continuously. Such parameters may play a significant role for the simulator output. In the reservoir engineering, we often deal with different discrete parameters. The quantitative discrete parameters can be, for example, well position coordinates on a reservoir grid. We can consider both horizontal or vertical wells. In such case, the quantitative discrete parameters are defined by a cell number of the well trajectory in a reservoir grid. There are also qualitative discrete parameters that may characterize reservoir uncertainties. For example, we can consider different geological porosity or permeability map realizations or different sets of relative permeability or capillary pressure curves.

Actually, numerous response surface models are available for predicting the output from a computer code when all inputs are continuous and few have been attempted to propose new methods for the cases involving both discrete and continuous parameters. The main objective of a response surface model involving discrete parameters is to answer whether a single model can adequately describe the input-output relationship for different levels of parameters. Standard approach assumes distinct GP models for modelling observations generated at different levels or combination of levels for different parameters. This approach can be time consuming and often it is not affordable. For example, let us consider a reservoir model with 2 discrete parameters such as an injection well coordinates with 11 possible levels for each coordinate. Then, the standard approach assumes different GP models for different combination of parameters. This corresponds to $11 \times 11 = 121$ different GP models (121 hyperparameters evaluations). Therefore, fitting one response surface model with discrete parameters can help in:

- Reducing computational time;

- Including discrete parameters in Sensitivity analysis (SA) and Uncertainty analysis (UA).

Different approaches have been proposed to model a simulator involving discrete parameters. McMillan et al. [1999] developed a proportionality model to analyze protein activity data. However, the correlation function proposed in their article has a limited applicability. Kennedy and O'Hagan [2000] introduced a Bayesian approach with an autoregressive model that combines the information from outputs of simulators of multi-level complexity. Qian and Wu [2008] extended this method by introducing Bayesian hierarchical Gaussian process for integrating multiple data sources of low accuracy and high accuracy. Furthermore, Qian et al. [2008] proposed a new formulation for correlation function describing qualitative and quantitative parameters.

In this work, we extend the method initiated by Qian et al. [2008]. We have worked with different correlation functions proposed therein and we have tested these models with a reservoir engineering case. In Chapter 3 we show how we can apply this method to time-series output modelling. Furthermore, we have also studied a CO_2 injection well placement application (see Chapter 3).

This chapter is organized as follows. Firstly, we briefly introduce GP response surface modelling in Section 2.2. Then, Section 2.3 discusses correlation function specifications both for continuous and discrete parameters. Section 2.4 provides analytical test cases. Section 2.5 provides a reservoir case example involving discrete quantitative and qualitative parameters.

2.2 Gaussian process response surface modelling

In this section, we recall the basics of GP modelling or *kriging*.

The idea of modelling an unknown function by a stochastic process was introduced in the field of geostatistics by Krige in the 1950's [Krige, 1951] and formalized in 1960's by Matheron [1963]. Later Sacks et al. [1989] proposed the use of kriging for prediction and design of experiments. The theory and the algorithms are formalized in Sacks et al. [1989]; Santner et al. [2003]; Welch et al. [1992].

Consider the output of a simulator as an unknown deterministic function $\mathrm{F}(\mathbf{x}) \in \mathbb{R}$, where $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ is a specified set of selected input parameters. The function F is only known in predetermined design points: $\mathbf{X}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{Y}^n = \{\mathrm{F}(\mathbf{x}_1), \ldots, \mathbf{x}_n)\}$. The objective is to predict the function $\mathrm{F}_0 = \mathrm{F}(\mathbf{x}_0)$ for some new arbitrary input

$\mathbf{x}_0$. The function is modeled as a sample path of a stochastic process of the form:

$$\widehat{\mathrm{F}}(\mathbf{x}) = \sum_{j=1}^{m} h_j(\mathbf{x}) \cdot \beta_j + Z(\mathbf{x}) = \boldsymbol{\beta}^\top \mathbf{h}(\mathbf{x}) + Z(\mathbf{x}) \tag{2.1}$$

where:

- $\boldsymbol{\beta}^\top \mathbf{h}(\mathbf{x})$ is the mean of the process and corresponds to a linear regression model with preselected given real-valued functions $\mathbf{h} = \{h_i, 1 \leq i \leq m\}$. Here, we only consider the case $\mathbf{h} = \mathbf{1}$.
- $Z(\mathbf{x})$ is a centered Gaussian stationary random process. This is defined by its covariance function: $\mathrm{C}(\mathbf{x}, \mathbf{y}) = \mathbb{E}\left[Z(\mathbf{x})Z(\mathbf{y})\right] = \sigma^2 \mathrm{R}(\mathbf{x}, \mathbf{y})$. $\mathrm{R}(\mathbf{x}, \mathbf{y})$ is the correlation function and $\sigma^2 = \mathbb{E}[Z(\mathbf{x})^2]$ denotes the process variance. Stationarity condition assumes: $\mathrm{R}(\mathbf{x}, \mathbf{y}) = \mathrm{R}(|\mathbf{x} - \mathbf{y}|)$, where $|\mathbf{x} - \mathbf{y}|$ denotes the distance between $\mathbf{x} \in \Omega$ and $\mathbf{y} \in \Omega$.

Numerous families of correlation functions have been proposed in the literature [Abrahamsen, 1997]. We use here Gaussian correlation function, the special case of the power exponential family. The power exponential correlation function is of the following form:

$$\mathrm{R}(\mathbf{x}, \mathbf{y}) = \exp\left(-\sum_{j=1}^{d} \frac{|x_j - y_j|^{p_j}}{\theta_j}\right) = \exp\left(-\sum_{j=1}^{d} \frac{d_j^{p_j}}{\theta_j}\right) \tag{2.2}$$

where $d_j = |x_j - y_j|$, $0 < p_j \leq 2$ and $\theta_j > 0$. The hyperparameters $(\theta_1, .., \theta_d)$ stands for correlation lengths which affect how far a sample point's influence extends. A high θ_i means that all points have a high correlation ($\mathrm{F}(\mathbf{x}_i)$ being similar across our sample), while a low θ_i means that there are significant difference between the $\mathrm{F}(\mathbf{x}_i)$'s [Forrester and Keane, 2009]. The parameters p_j corresponds to the smoothness parameters. These parameters determine mean-square differentiability of the random process $Z(x)$. For $p_j = 2$ the process is infinitely mean-square differentiable and the correlation function is called Gaussian correlation function. Hence, Gaussian correlation function is infinitely mean-square differentiable and it leads to a stationary and anisotropic process $Z(x)$ [Santner et al., 2003; Welch et al., 1992]. Regardless the choice of a correlation function, the estimation of hyperparameters $(\theta_1, .., \theta_d)$ is crucial for a reliable prediction. Here, we use maximum likelihood criterion to estimate hyperparameters [Santner et al., 2003].

The experimental design points are selected in order to retrieve the most information on the function at the lowest computational cost. The number of design points for a reliable response surface model depends on the number of inputs and on the complexity of the response to analyze [McKay et al., 1979; Santner et al., 2003]. Latin Hypercube Designs (LHD) provides a uniform coverage of the input domain. If we wish to generate a sample of size n, first, we partition the domain of each variable into n intervals of equal probability. Then, we randomly sample $\mathbf{x}_1$ according to the distribution of each of the n intervals. Further, for each of the n values for $\mathbf{x}_1$, we randomly select one interval to sample for $\mathbf{x}_2$, so that only one sample of $\mathbf{x}_2$ is taken in each interval. We continue the process of a random sampling without replacement until all the variables have been sampled. As a result we generate a sample where each of d inputs is sampled only once in each of n intervals. Latin hypercube designs have been applied in many computer experiments since they were proposed by [McKay et al., 1979].

In this work, we use a modified version of LHD - maximin LHD. It is based on maximizing a measure of the closeness of the points in a design $\mathbf{D}^n$:

$$\max_{\text{design } \mathbf{D}^n} \min_{\mathbf{x}_1,\mathbf{x}_2 \in \mathbf{D}^n} d(\mathbf{x}_1, \mathbf{x}_2)$$

It can guarantee that any two points in the design are not "too close". Hence, the design points are uniformly spread over the input domain.

Consequently, when we have the experimental design $\mathbf{X}^n = (\mathbf{x}_1, .., \mathbf{x}_n)$ and the observation data $\mathbf{Y}^n = (F(\mathbf{x}_1), .., F(\mathbf{x}_n))$, the multivariate distribution according to the model (2.1) for the Gaussian correlation function can be expressed as:

$$\begin{pmatrix} Y_0 \\ \mathbf{Y}^n \end{pmatrix} \sim \mathcal{N}_{1+n} \left[\begin{pmatrix} \mathbf{h}^\top(\mathbf{x}_0) \\ \mathbf{H} \end{pmatrix} \boldsymbol{\beta}, \ \boldsymbol{\sigma}^2 \begin{pmatrix} 1 & r^\top(\mathbf{x}_0) \\ r(\mathbf{x}_0) & \mathbf{R} \end{pmatrix} \right],$$

where $\mathbf{R} = (\mathrm{R}(\mathbf{x}_i, \mathbf{x}_j))_{1 \le i,j \le n} \in \mathbb{R}^{n \times n}$ is the correlation matrix among the observations; $\mathbf{r}(\mathbf{x}_0) = (\mathrm{R}(\mathbf{x}_1, \mathbf{x}_0), .., \mathrm{R}(\mathbf{x}_n, \mathbf{x}_0))^\top \in \mathbb{R}^n$ is the correlation vector between the observations and the prediction point; $\mathbf{h}^\top(\mathbf{x}_0) = (h_j(\mathbf{x}_0))_{1 \le j \le m} \in \mathbb{R}^m$ is the vector of regression function at the prediction point $\mathbf{x}_0$ and $\mathbf{H} = (h_j(\mathbf{x}_i))_{1 \le i \le n, 1 \le j \le m} \in \mathbb{R}^{n \times m}$ is the matrix of regression functions at the experimental design. The parameters $\boldsymbol{\beta}$ and $\boldsymbol{\sigma}$ are unknown.

Considering the unbiasedness constraint, the parameter $\boldsymbol{\beta}$ is replaced by the

generalized least squares estimate $\widehat{\boldsymbol{\beta}}$ in (2.1). Here, $\widehat{\boldsymbol{\beta}}$ is of the following form: $\widehat{\boldsymbol{\beta}} = \left(\mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\right)^{-1} \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{Y}^n$. Assuming that the correlation function is the Gaussian correlation function, the prediction is therefore given by:

$$\widehat{F}(\mathbf{x}_0) = \mathbf{h}^\top(\mathbf{x}_0) \cdot \hat{\boldsymbol{\beta}} + \hat{\mathbf{r}}(\mathbf{x}_0) \;\; \hat{\mathbf{R}}^{-1} \left(\mathbf{Y}^n - \mathbf{H} \cdot \hat{\boldsymbol{\beta}}\right) \tag{2.3}$$

The hyperparameters $\boldsymbol{\theta} = (\theta_1, .., \theta_m)$ and the process variance $\boldsymbol{\sigma}^2$ are estimated by Maximum Likelihood (MLE). Using the multivariate normal assumption, the MLE for $\boldsymbol{\sigma}^2$ is:

$$\hat{\boldsymbol{\sigma}}^2 = \frac{1}{n} \left(\mathbf{Y}^n - \mathbf{H}\, \hat{\boldsymbol{\beta}}\right) \;\; \mathbf{R}^{-1} \left(\mathbf{Y}^n - \mathbf{H}\, \hat{\boldsymbol{\beta}}\right) \tag{2.4}$$

Knowing the estimations for $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\sigma}}^2$, the coefficients $\boldsymbol{\theta}$ are estimated by maximizing the log likelihood:

$$l\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^2, \boldsymbol{\theta}\right) = -\frac{1}{2} \left[n \log \hat{\boldsymbol{\sigma}}^2(\boldsymbol{\theta}) + \log(det(\mathbf{R}(\boldsymbol{\theta})) + n + n \log(2\pi)\right] \tag{2.5}$$

This function (2.5) depends only on $\boldsymbol{\theta}$.

The resulting predictor $\widehat{F}(\mathbf{x}_0) \sim \mathcal{N}(\mu_{\widehat{F}(\mathbf{x}_0)}, {\sigma^2}_{\widehat{F}(\mathbf{x}_0)})$ is a Gaussian random variable. Therefore, we can provide the confidence interval of the prediction. For a given level of probability $1 - \alpha$, the confidence interval is given by:

$$Y_0 \in \left[\mu_{\widehat{F}(\mathbf{x}_0)} - \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) {\sigma^2}_{\widehat{F}(\mathbf{x}_0)}; \mu_{\widehat{F}(\mathbf{x}_0)} + \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) {\sigma^2}_{\widehat{F}(\mathbf{x}_0)}\right]$$

where Φ^{-1} is the inverse function to the standard normal cumulative distribution function. Figure 2.1 displays an example on analytical function $F(x) = x \sin(x)$. The confidence interval is calculated for $\alpha = 5\%$. The black bold line corresponds to the approximated function, the blue dotted line depicts the estimated predictor. The red dots are the experimental design for $n = 5$. It is clear that more points in design leads to a more narrow confidence interval. It is important to have more design points in the region of interest to obtain a more reliable approximation.

After having estimated all the model parameters, we now need to validate the model. In this work for estimation of prediction accuracy of the model, we use the predictivity index, Q_2, and Root Mean Squared Error, $RMSE$. The predictivity

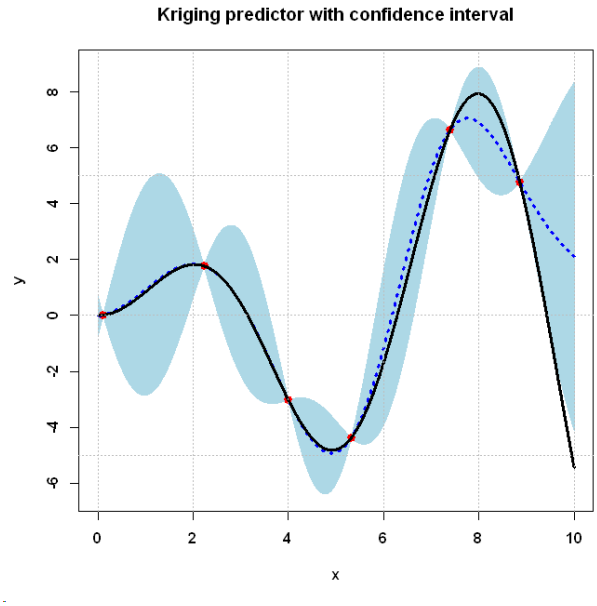


Figure 2.1: Kriging approximation with a confidence interval.

index is calculated basing on a cross validation and it has the following form:

$$RMSE := \sqrt{\sum_{i=1}^{n} \frac{(\hat{S}_{X/\mathbf{x}_i} - F(\mathbf{x}_i))^2}{n}} \qquad Q_2 := 1 - \frac{\sum_{i=1}^{n} \left(\hat{S}_{X/\mathbf{x}_i} - F(\mathbf{x}_i)\right)^2}{\sum_{i=1}^{n} \left(F(\mathbf{x}_i) - \tilde{F}\right)^2} \tag{2.6}$$

$\hat{S}_{X/\mathbf{x}_i}$ is the kriging model computed using all the design points $\mathbf{X}^n$ except $\mathbf{x}_i$ and $\tilde{F}$ is the mean of $\{F(\mathbf{x}_i), i = 1, \ldots, n\}$.

The closer Q_2 is to 1 or $RMSE$ is to 0, the higher is the model predictivity. These criteria can be also calculated on separate validation test data by performing additional simulator runs. This approach provides higher accuracy measure though it requires additional CPU time costs.

2.3 Correlation function with discrete parameters

In this section, we provide a general method to build a correlation function in the mixed case of continuous and discrete parameters.

As discussed in the previous section, the problem of handling discrete parameters in response surface modelling has been studied in previous works by different authors [Kennedy and O'Hagan, 2000; McMillan et al., 1999; Qian et al., 2008] . Among all

the proposed methods we can highlight two of them that are based on adaptation of the correlation function of stochastic process for the discrete parameters [McMillan et al., 1999; Qian et al., 2008].

Recall that a the GP response surface model is defined by the following equation:

$$\widehat{f}(\mathbf{x}) = \sum_{j=1}^{m} \beta_j h_j(\mathbf{x}) + Z(\mathbf{x}), \quad x \in \Omega \subset \mathbb{R}^d,$$

where $\mathbf{h} = (h_1, \ldots, h_m)^\top$ are user-selected regression functions, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$ is vector of unknown coefficients and the residual $Z(\mathbf{x})$ is a centered stationary Gaussian process. The stochastic process $Z(\mathbf{x})$ is characterized by its variance σ^2 and its correlation function $R(\mathbf{x}, \mathbf{y}) = R(|\mathbf{x} - \mathbf{y}|), \quad (\mathbf{x}, \mathbf{y}) \in \Omega^2$. The correlation function measures dependency as a function of distance between two points in the input domain Ω. When the input domain Ω has a discrete part, the main problem is to define a *"good"* distance in this part. Indeed, a correlation function, through all the levels of parameters, should satisfy basic relationship between the inputs. When the distance between the point $\mathbf{x}$ and the point $\mathbf{y}$ is small, this means that $R(\mathbf{x}, \mathbf{y})$ is close to one and the values of $Z(\mathbf{x})$ and $Z(\mathbf{y})$ are correlated. Reciprocally, when the distance between $\mathbf{x}$ and $\mathbf{y}$ is high, the value $R(\mathbf{x}, \mathbf{y})$ is close to zero and $Z(\mathbf{x})$ and $Z(\mathbf{y})$ are nearly independent .

McMillan et al. [1999] introduced a multiplicative formulation of the correlation function. Let G be a Gaussian correlation function:

$$G(\mathbf{u}, \mathbf{v}) = \prod_{i=1}^{d} \exp\left(-\frac{|u_i - v_i|^2}{\theta_i}\right), \quad \mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_d \end{pmatrix}, \mathbf{v} = \begin{pmatrix} v_1 \\ \vdots \\ u_d \end{pmatrix} \in \Omega \subset \mathbb{R}^d$$

Here, the unknown hyperparameters are $\boldsymbol{\theta} = (\theta_1, ..., \theta_d)$. For a discrete parameter with k levels in McMillan et al. [1999], the correlation function multiplier $\exp \frac{|u_i - v_i|^2}{\theta_i}$ is replaced by:

$$\prod_{j=1}^{k} \exp\left(-\frac{|\mathbf{I}(u_i = j) - \mathbf{I}(v_i = j)|}{\theta_{ij}}\right).$$

where $\mathbf{I}(v_i = j)$ is an indicator function (equals to 1 if the discrete parameter v_i takes level j, and 0 otherwise).

The correlation function defined in this way has a limited applicability and is

difficult to interpret. If we consider, for example, $k = 4$ then the correlation matrix is under parametrized: six different correlations are defined by four parameters. Therefore, it is impossible to estimate independently all the hyperparameters of the model. In our work, we rather use the following formulation (proposed by Qian et al. [2008]) for the correlation function.

Firstly, let us discuss more precisely our framework. Suppose we are working with a function (the simulator output): $f(\mathbf{w}) = f(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = (x_1, .., x_I)^\top$ are the continuous variables and $\mathbf{y} = (y_1, .., y_J)^\top$ are the discrete variables. Each discrete variable $y_j, \quad j = 1, \ldots, J$ has n_j possible values or levels ($y_j \in \{1, ..., n_j\}$ for $j = 1, .., J$). For this function, the response at an input $\mathbf{w}$ is modeled as follows:

$$\widehat{f}(\mathbf{w}) = \sum_{j=1}^{m} \beta_j \cdot h_j(\mathbf{w}) + Z(\mathbf{w}) \tag{2.7}$$

Now, we define the correlation structure for $Z(\mathbf{w})$ that involves both discrete and continuous variables. To keep things simple, we consider the case with only one discrete parameter y_1 with n_1 levels: $y_1 \in \{1, .., n_1\}$. So that, we want to define a correlation function for $Z(\mathbf{w})$, where $\mathbf{w} = (\mathbf{x}, y_1)^\top$.

For every fixed value of the discrete variable $y_1 = i \in \{1, \ldots, n_1\}$, the Gaussian process: $Z(\mathbf{w}) = Z(\mathbf{x}, y_1) = Z_i(\mathbf{x})$ is completely defined by its variance σ^2 and by the Gaussian correlation function. So that, if we look over all the possible values of the discrete variable $y_1 \in \{1, \ldots, n_1\}$, for every fixed value the resulting process depends only on the set of the uncertain continuous variables $\mathbf{x} \in \Omega \in \mathbb{R}^d$. We can define a mean-zero n_1-variate process: $\mathbf{Z}(\mathbf{x}) = (Z_1(\mathbf{x}), \ldots, Z_{n_1}(\mathbf{x}))$. The problem is to define cross-correlation function for $\mathbf{Z}(\mathbf{x})$.

For a given arbitrary input $\mathbf{w}_1 = (\mathbf{x}_1, y_{1_1})$ and $\mathbf{w}_2 = (\mathbf{x}_2, y_{1_2})$, the correlation function of the process $Z(\mathbf{w})$ is a cross-correlation between two process corresponding to different levels of the discrete variable y_{1_1} and y_{1_2}:

$$\operatorname{corr}\left(Z(\mathbf{w}_1), Z(\mathbf{w}_2)\right) = \operatorname{corr}\left(Z_{y_{1_1}}(\mathbf{x}_1), Z_{y_{1_2}}(\mathbf{x}_2)\right) \tag{2.8}$$

For the sake of clarity, by extracting a proper matrix of multipliers $\mathbf{A}$, we can represent the multivariate process $\mathbf{Z}(\mathbf{x}) = (Z_1(\mathbf{x}), \ldots, Z_{n_1}(\mathbf{x}))$ as a linear combination of the processes with the same variance σ^2 and the correlation function $\mathrm{R}_{\boldsymbol{\theta}}(\mathbf{x}_1, \mathbf{x}_2)$:

$$\mathbf{Z}(\mathbf{x}) = \mathbf{A} \cdot \boldsymbol{\eta}(\mathbf{x}),$$

where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_{n_1}]^t$ is $n_1 \times n_1$ nonsingular matrix such that: $\mathbf{a}_j^t \mathbf{a}_j = 1$ for $j = 1, \ldots, n_1$. Vector $\boldsymbol{\eta}(\mathbf{x}) = (\eta_1(\mathbf{x}), \ldots, \eta_{n_1}(\mathbf{x}))^t$ is a vector of independent Gaussian stochastic processes with the same variance σ^2 and the correlation function $\mathrm{R}_{\boldsymbol{\theta}}(\mathbf{x}_1, \mathbf{x}_2)$. Then, the cross-correlation for the vector η is expressed as:

$$\operatorname{corr}(\eta(\mathbf{x}_1), \eta(\mathbf{x}_2)) = \mathrm{R}_{\theta}(\mathbf{x}_1, \mathbf{x}_2) \cdot \mathbf{I}_{n_1},$$

where $\mathbf{I}_{n_1}$ is $n_1 \times n_1$ identity matrix.

Coming back to equation (2.8) for the correlation function of stochastic process $Z(\mathbf{w})$ from (2.7). With the help of selected representation, (2.8) can be expressed as:

$$\operatorname{corr}(Z(\mathbf{w}_1), Z(\mathbf{w}_2)) = \operatorname{corr}\left(\boldsymbol{a}^{\boldsymbol{t}}{}_{y_{1_1}} \eta(\mathbf{x_1}), \boldsymbol{a}^{\boldsymbol{t}}{}_{y_{1_2}} \eta(\mathbf{x_2})\right) = \boldsymbol{a}^{\boldsymbol{t}}{}_{y_{1_1}} \boldsymbol{a}_{y_{1_2}} \mathrm{R}_{\boldsymbol{\theta}}(\mathbf{x}_1, \mathbf{x}_2) \quad (2.9)$$

Here, $\boldsymbol{a}^{\boldsymbol{t}}{}_{y_{1_1}} \boldsymbol{a}_{y_{1_2}}$ corresponds to the cross-correlation between the different values of levels of discrete variable y_1. Let us denote $\boldsymbol{\tau}_{r,s} = \mathbf{a}_r^t \mathbf{a}_s$, where $r, s \in (1, \ldots, n_1)$. Then, by its original definition the matrix $\mathbf{T} = (\boldsymbol{\tau}_{r,s}) = \mathbf{A}\mathbf{A}^t$ is a $n_1 \times n_1$ positive definite matrix with unit diagonal elements. Thus, for any correlation function $\mathrm{R}(\mathbf{x}_1, \mathbf{x}_2)$ the function (2.9) is positive definite. Therefore, it is a possible correlation function for the process $Z(\mathbf{w})$ [Abrahamsen, 1997].

For a general case with J discrete factors: $\mathbf{y} = (y_1, .., y_J)$, $y_j \in (1, .., n_j)$ for $j = 1, \ldots, J$. Following the equation (2.9), a correlation function to $Z(\mathbf{w})$ is a product:

$$\operatorname{corr}(Z(\mathbf{w}_1, \mathbf{w}_2)) = \prod_{j=1}^{J} \left[\boldsymbol{\tau}_{j,y_{j_1},y_{j_2}} \mathrm{R}_{\boldsymbol{\theta}_j}(\mathbf{x}_1, \mathbf{x}_2)\right], \quad (2.10)$$

where every matrix $\mathbf{T}_j = (\boldsymbol{\tau}_{j,r,s})$ is an $n_j \times n_j$ positive definite matrix with unit diagonal elements. Consequently, equation (2.10) defines a valid correlation function as the product of J valid correlation functions Abrahamsen [1997].

In this work, we focus on Gaussian family of correlation functions:

$$\mathrm{R}_{\boldsymbol{\theta}_j}(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\sum_{i=1}^{I} \frac{|x_{1i} - x_{2i}|^2}{\theta_{j_i}}\right).$$

Therefore, if we collect the hyperparameters corresponding to the same continuous variable x_i and we denote $1/\boldsymbol{\theta}_i = \sum_{j=1}^{J} 1/\theta_{j_i}$. Then, for the case of Gaussian

family of correlation functions (2.10) becomes:

$$\operatorname{corr}\left(Z\left(\mathbf{w}_1\right), Z\left(\mathbf{w}_2\right)\right) = \prod_{j=1}^{J} \left[\boldsymbol{\tau}_{j,y_{j_1},y_{j_2}}\right] \exp\left(-\sum_{i=1}^{I} \frac{|x_{1i} - x_{2i}|^2}{\boldsymbol{\theta}_i}\right) \tag{2.11}$$

Here, the parameters $\boldsymbol{\tau}_{j,y_{j_1},y_{j_2}}$ measures the correlation between the response at any two input values $\mathbf{w}_1$ and $\mathbf{w}_2$ that differ only by the value of the discrete variable y_j: at levels y_{j_1} and y_{j_2} respectively. Whereas, parameters $\bar{\boldsymbol{\theta}} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_I)$ are the correlation lengths for continuous variables $\mathbf{x} \in \Omega \subset \mathbb{R}^d$.

Thus, we have defined a correlation function for an emulator modelling the computer code involving both continuous and discrete parameters. However, the right choice of the matrices $\mathbf{T_j}$ is not straightforward. In the following subsection, we discuss another method in case of an ordered discrete parameters space.

2.3.1 Correlation function for ordered factors

Different structures for the matrices $\mathbf{T_j}$ can be chosen. Qian et al. [2008] considered the cases of isotropic, multiplicative or group correlation functions. In this section, we focus on a correlation function when the discrete parameter space is ordered. The ordered assumption is quite natural in the experimental design paradigm. For example, we have the discretized well coordinates. The levels are already ordered by a given grid of a reservoir model.

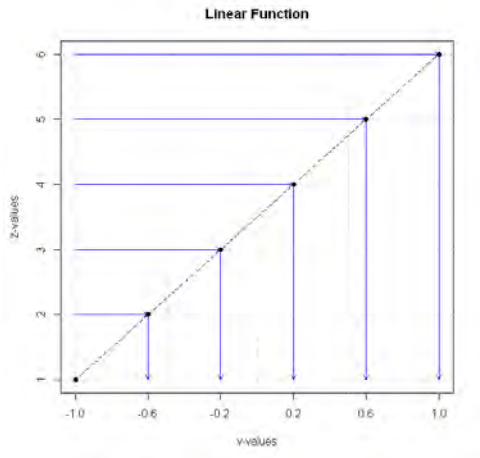
Suppose, that all the discrete parameters $\mathbf{y} = (y_1, \ldots, y_J)$ are ordered in an increasing way. This means that for each $y_j \in \{1, \ldots, n_j\}$ the following inequality holds: $1 < 2 < \cdots < n_j$. We consider the case of one discrete parameter $y \in (1, \ldots, n)$ and $1 < 2 < \cdots < n$.

The method consists in transforming the discrete parameter y to a quantitative one v. This is performed by using of a transformation function F. The function F should be continuous and strongly increasing on $[-1, 1]$. Then, we transform the level k of parameter y to level v_k of parameter v by the following equation:

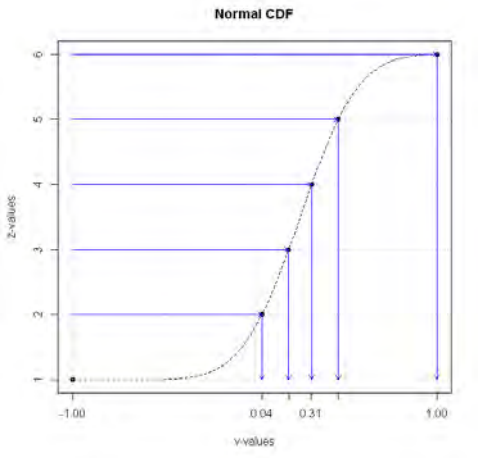
$$F\left(v_k\right) = F(-1) + (k-1) \cdot \frac{F(1) - F(-1)}{n-1}, \tag{2.12}$$

where $-1 = v_1 < \ldots < v_n = 1$. Consequently, we assign: $\tau_{r,s} = \mathrm{R}(v_r, v_s)$, where R is a correlation function for v. Having converted all the discrete factors, we obtain the Gaussian correlation function with $I + J$ number of variables $\mathbf{w} = (\mathbf{x}, \mathbf{v})$.

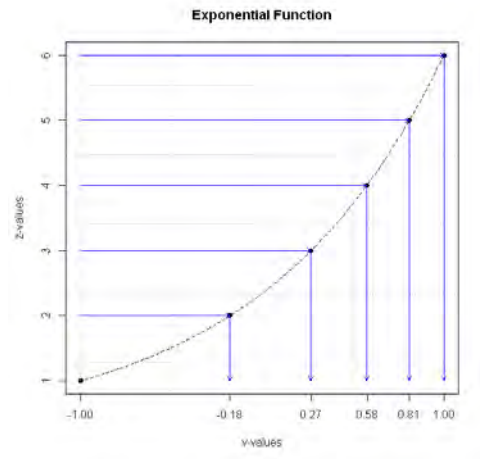
Different transformation functions F can be selected. Cumulative distribution functions (CDF) are a reasonable choice. Also, some analytical functions such as exponential or sine functions can be chosen. In Figure (2.2) three different transformation functions are displayed for one qualitative factor with 6 ordered levels. Figure (2.2(a)) displays linear function, that is also a CDF for uniform distribution on $[-1, 1]$ and Figure (2.2(b)) is a CDF of Gaussian distribution with mean $\mu = 0.25$ and variance $\sigma = 0.25$. Figure (2.2(d)) shows analytical sine function and Figure (2.2(d)) shows analytical exponential function.



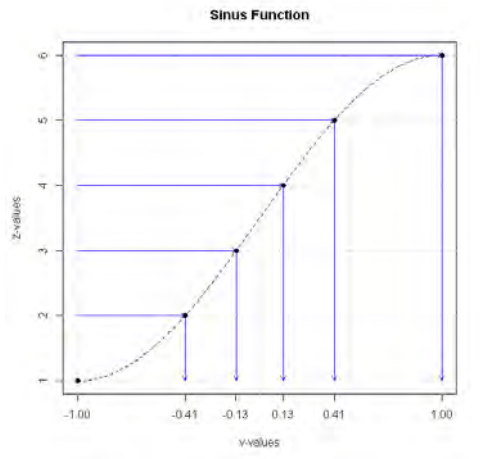
(a) Linear function



(b) Normal CDF, mean=0.25, sd=0.25



(c) Exponential function: $\exp(x)$



(d) Sine function: $\sin(\pi x/2)$

Figure 2.2: Transformation function examples.

By definition, the transformation function determines the choice of parameter v_k. It is worth to mention, that for the discrete parameters with originally unknown order, the method can be used firstly by assigning some arbitrary order of levels. The proper order can be established by trials-and-errors method basing on some test data. So that, we can select the order that provides the highest model predictivity.

In the next section, we have studied the prediction performance of different transformation functions on analytical functions. The results are discussed later.

2.4 Analytical tests

In this section, we present analytical test functions for the performance analysis of the method introduced in the previous section. In Subsection 2.4.1 we give an example of a multi-case function. Here, the level of discrete parameter defines the function behavior. We compare the classical independent analysis and the method proposed in Section 2.3. Subsection 2.4.2 provides a comparative analysis of different transformation functions applied on two analytical functions.

2.4.1 Multi-case function example

Here, we consider the multi-case function (2.13) of 2 continuous variables (x, y) and one discrete variable z. The discrete parameter, z defines the different cases of the function. The parameter z has three possible values $\{1, 2, 3\}$. The function is defined as follows:

$$F = \begin{cases} \exp(1.9x)\sin(5\pi y/2) & \text{if } z_1 = 1; \\ \exp(2.5x)\sin(5\pi y/2) & \text{if } z_1 = 2; \\ \exp(2.1x)\sin(5\pi y/2) & \text{if } z_1 = 3 \end{cases} \tag{2.13}$$

The functional cross-section at $x = y$ is depicted in Figure (2.3). We can notice that the functions have similar behavior for different levels of the discrete parameter. We compare two approaches. Independent analysis assumes distinct GP models for modelling the data collected at different values of the discrete factor z. At the same time, for integrated analysis we build a single GP model across different values of discrete variable z. We apply a linear transformation function for integrated analysis.

We have studied the method prediction performance at different sizes of experi-

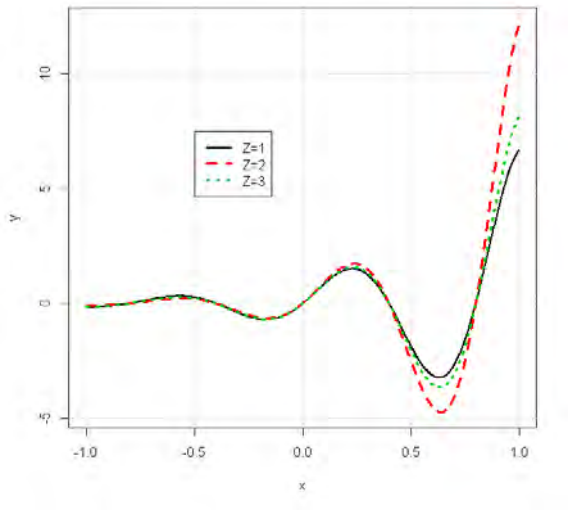


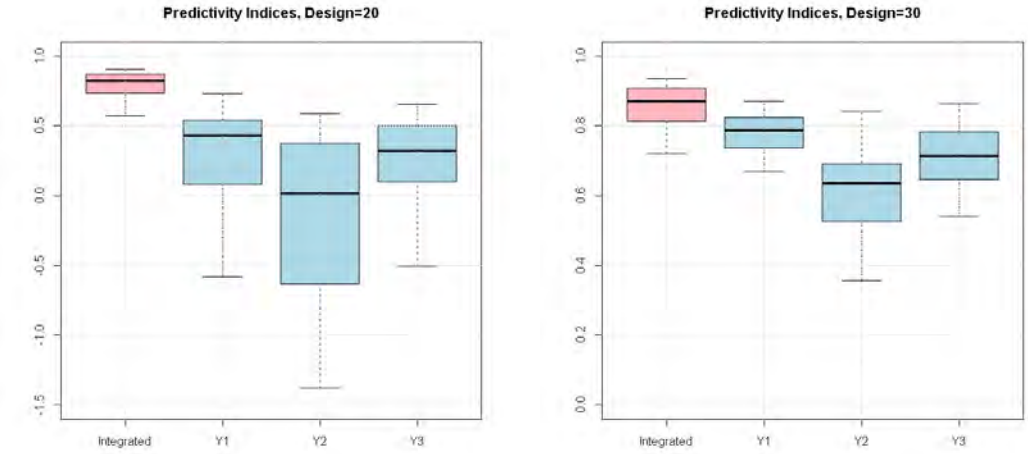
Figure 2.3: Multi-case function cross-section.

mental design N. For independent analysis, for a fixed value of z we computed three different GP models of two variables (x, y) generated by an experimental design of the same size N. For integrated approach, we work with a single GP model of three variables (x, y, z) based on an experimental design of size $3 \times N$ accordingly. The experimental design is a maximin Latin hypercube design. We have evaluated 50 different GP models computed from 50 different realizations of an experimental design.

We use a predictivity index as a criterion for comparison of different approaches. Recall that the predictivity index is defined as follows:

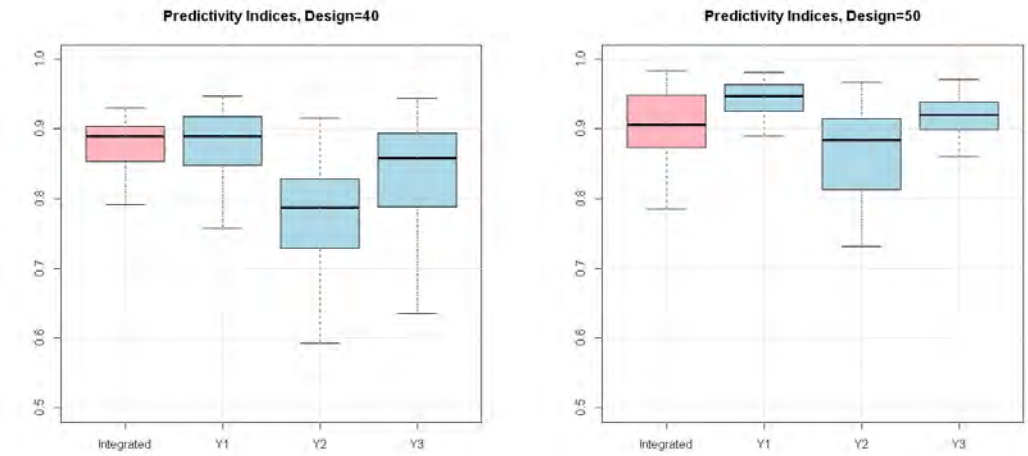
$$Q_2 := 1 - \frac{\sum_{i=1}^{N_{test}} \left(\widehat{F}(\mathbf{x}_i) - F(\mathbf{x}_i) \right)^2}{\sum_{i=1}^{N_{test}} \left(F(\mathbf{x}_i) - \tilde{\mathbf{F}} \right)^2}$$

where $\mathbf{Y}_{test} = (F(\mathbf{x}_1), \ldots, F(\mathbf{x}_{N_{test}}))$ are the test data, $\tilde{\mathbf{F}}$ is their empirical mean and $\widehat{F}(\mathbf{x}_i)$ is the GP model approximation at the prediction point $\mathbf{x}_i$. For estimation of the predictivity indices we use additional 500 function evaluations. Figure 2.4 compares predictivity indices calculated at different designs of size $N = 20,\ 30,\ 40,\ 50$. The integrated approach provides higher value of predictivity index Q_2 at lesser number N of input design points. The prediction efficiency of the integrated approach is clearly better for the case of $N = 20$ and $N = 30$. There is some starting value



(a) Predictivity indices, N=20

(b) Predictivity indices, N=30



(c) Predictivity indices, N=40

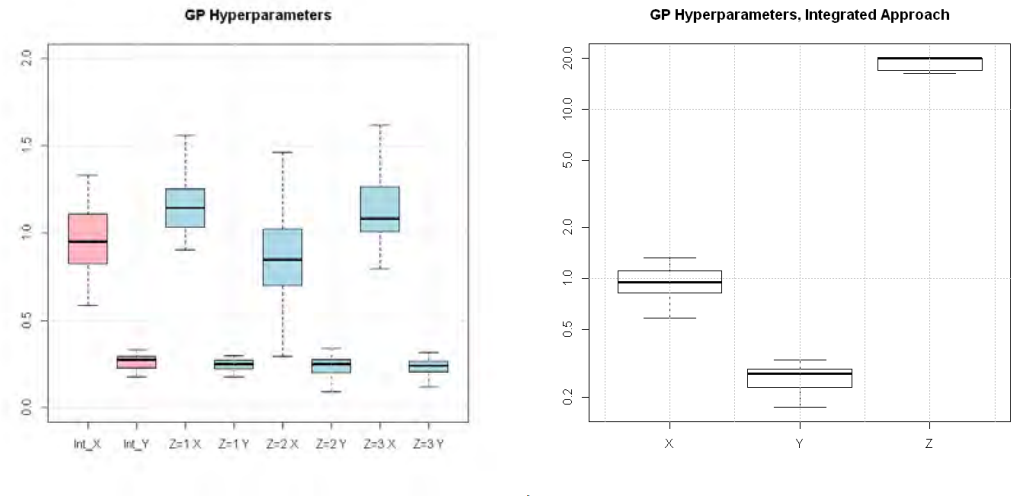
(d) Predictivity indices, N=50

Figure 2.4: Predictivity indices comparison.

of input sample size N from which both approaches demonstrate equivalent prediction accuracy. For this case, it is $N = 50$, where all the models provide equivalent prediction efficiency. This value of N generally depends on the complexity of the function and the total number of input variables including the number of discrete parameters and the associated number of levels.

In addition, we provide boxplots comparison of hyperparameters estimations by different approaches for an experimental design of size $N = 30$. Figure 2.5(a) displays comparison of the hyperparameters for continuous variables (x, y) calculated

respectively by integrated approach and by three different GP models for every fixed level of the discrete parameter $z = 1, \quad 2, \quad 3$. Figure 2.5(b) depicts the hyperparameters estimations for (x, y, z) calculated by the integrated approach.



(a) Hyperparameters comparison (b) Integrated approach

Figure 2.5: Estimated hyperparameters comparison.

If we compare estimations of the hyperparameters corresponding to continuous variables (x, y), we can see that all four models provide comparable estimations of the hyperparameters.

To summarize, the presented analytical results show that with the integrated approach it is possible to achieve higher prediction accuracy with a smaller size of the involved experimental design and therefore with much lower CPU cost.

2.4.2 Transformation function selection

In this section, we provide sensitivity analysis of prediction accuracy depending on the choice of transformation function. We consider an analytical function involving different number of discrete parameters and different number of corresponding levels.

We are approximating the following functions (2.14) with a GP response surface model. The size of the input experimental design is equal to 50 for every case. To provide comprehensive comparison, we evaluate 50 different models and the

predictivity was estimated with 250 additional confirmation function evaluations.

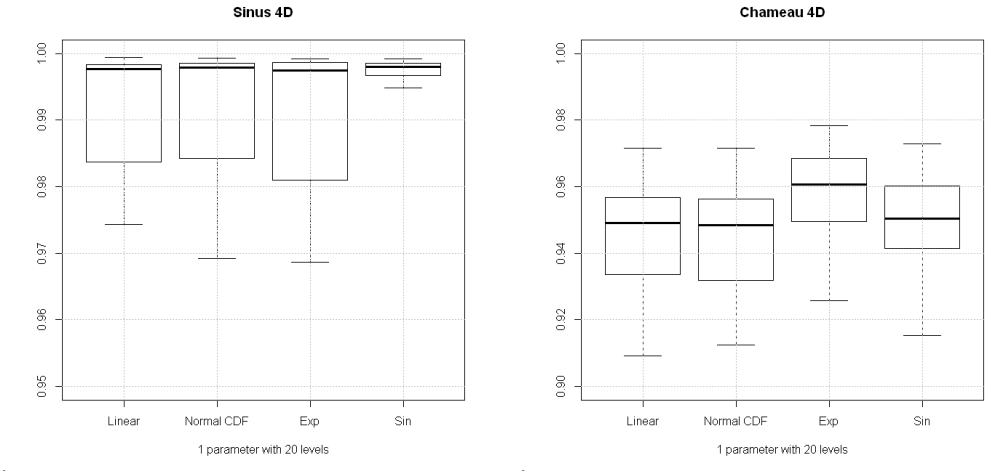
Chameau 4D
$$F(\mathbf{x}) = 4x^4 - \frac{21}{10}x^4 + x \cdot y - 4y^2 + 4y^4 + 3\sin(z) + x \cdot z + w^4 + 2w \cdot z + \frac{2}{3}w^5$$
Sinus 4D
$$F(\mathbf{x}) = \frac{\sin(5x+5)}{5x+5} - \frac{\sin(5y-3)}{5y-3} - \frac{\sin(z)}{z} - \frac{\sin(3w+3)}{3w+3} \tag{2.14}$$

The presented functions involve four variables. We consider cases where one, w, or two of them (z, w) are discrete. Transformations from discrete to continuous variables follow:

1. Linear: $F_1(x) = x$
2. Standard normal distribution function: $F_2(x) = \frac{1}{\sqrt{2 \cdot \pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) dt$
3. Exponential function: $F_3(x) = \exp(x)$
4. Sine Transformation Function: $F_4(x) = \sin(\frac{\pi \cdot x}{2})$

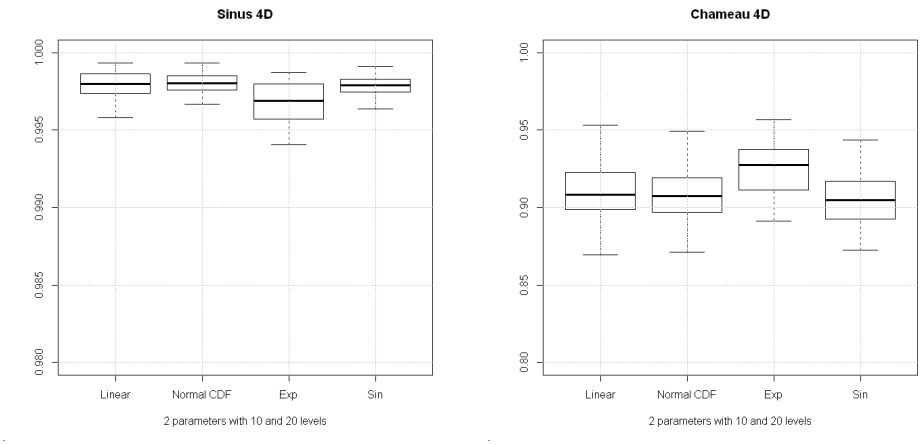
These are the transformation functions plotted in Figure 2.2.

We considered the case of one discrete parameter w with 10 levels and two discrete parameters (z, w) with 10 and 20 levels accordingly. Boxplots of estimated predictivity indices for different transformation functions are presented in Figure (2.4.2). Figures 2.6(a) and 2.6(c) display the predictivity indices estimations for the function *Sinus 4D* in the case of one and two involved discrete variables respectively. The same cases for function *Chameau 4D* are depicted in Figures 2.6(b) and 2.6(d)



(a) Predictivity indices, Sinus 4D

(b) Predictivity indices, Chameau 4D

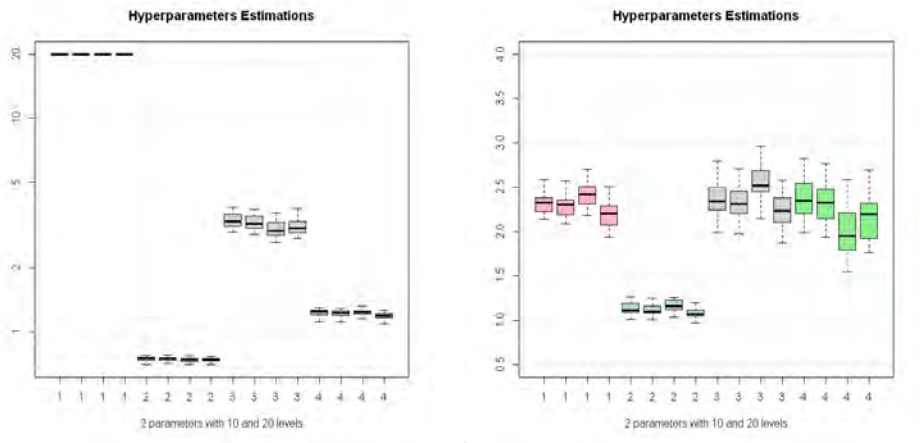


(c) Predictivity indices, Sinus 4D (d) Predictivity indices, Chameau 4D

Figure 2.6: Predictivity indices comparison.

Comparing the boxplot of the computed predictivity indices, we may conclude that for some cases specific transformation function provides a little higher prediction accuracy. However, in general if we compare the order of the differences, it is clear that the model performance differences are not significant.

In addition, we present hyperparameters estimations for the case of two discrete parameters for the function *Sinus 4D* (Figure 2.7(a)) and Chameau 4D (Figure 2.7(b)).



(a) Hyperparameters estimations, Sinus 4D (b) Hyperparameters estimations, Chameau 4D

Figure 2.7: Estimated hyperparameters comparison.

The estimated hyperparameters are low sensitive to the transformation function selection. So that, the GP response surface model performance involving both continuous and discrete parameters are insensitive to the choice of the transformation function.

In the next section, we present the method approbation with the reservoir simulation model.

2.5 Reservoir study

As discussed in the Introduction, the technology for CO_2 storage monitoring was completely adapted from petroleum industry. In this work, the PUNQ-S3 reservoir case has been used for testing the method [PUNQ-S3 [1996]].

The PUNQ-S3 test case is a 3D synthetic reservoir model derived from the real field data of the North Sea Brent reservoir. PUNQ-S3 (Production forecasting with UNcertainty Quantification) is a benchmark model to test and analyze new methods on uncertainty quantification and inverse problems of history matching.

A grid of the PUNQ-S3 reservoir field and the original pressure distribution in the reservoir are displayed in Figure 2.5. The simulation model contains $19 \times 28 \times 5$ grid blocks. The reservoir is surrounded by a strong aquifer in the North and the West, and is bounded by a fault to the East and South. A small gas cap is located in the center of this dome-shaped structure. The simulation model consists of five independent layers. The porosity distribution in each layer is modeled by geostatistical simulation. The layers 1, 3, 4 and 5 are assumed to be of good quality, while the layer 2 is of poorer quality. Initially, the field contains six production wells located around the gas-oil contact. No injection wells are required due to the strong aquifer. The reservoir pressure map and the production wells are displayed in Figure 2.5.

Table 2.1 represents the main uncertainty parameters considered in this study. There are 6 continuous parameters as **AQUI**, **MPH1**, **MPH2**, **MPV1**, **MPV2** and two discrete parameters: **P1X**, **P1Y** that corresponds to X and Y well position grid coordinates. Here, we consider an additional qualitative discrete parameter: **Z**. It represents different realizations of porosity and permeability maps. The parameter has nine different levels that correspond to nine possible geostatistical realizations.

The model simulation period is 10 years. Production data is observed every

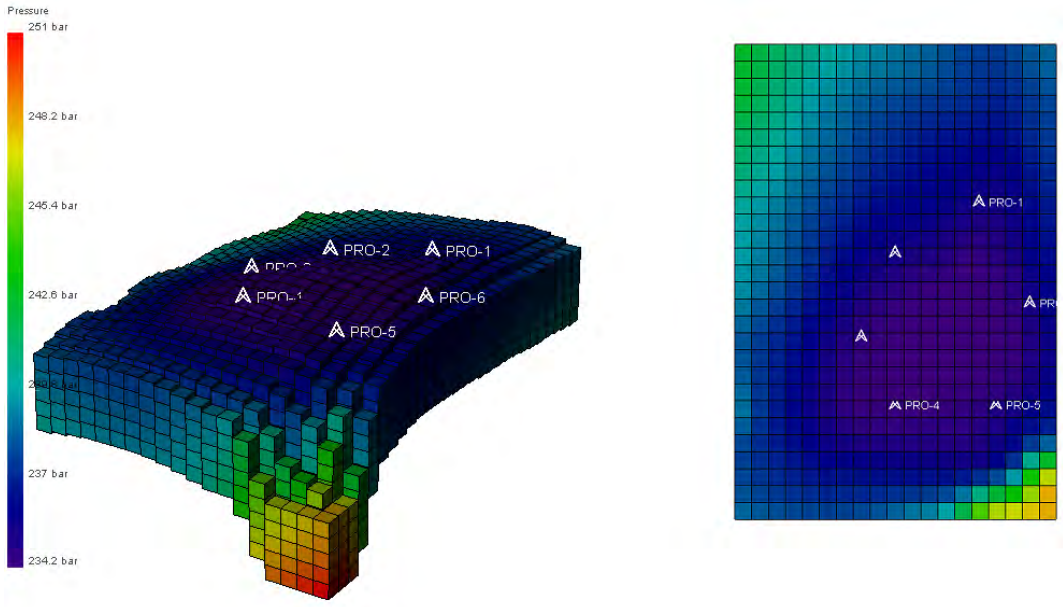


Figure 2.8: PUNQ-S3 reservoir model: pressure distribution.

Name	Description	Min	Max	Def
AQUI	Analytical porosity of the aquifer strength	0.2	0.3	0.2
MPH1	Horizontal transmissibility multiplier in good units	0.8	1.2	1.0
MPH2	Horizontal transmissibility multiplier in poor units	0.8	1.2	1.0
MPV1	Vertical transmissibility multiplier in good units	0.8	1.2	1.0
MPV2	Vertical transmissibility multiplier in poor units	0.8	1.2	1.0
SORG	Critical saturation values used in endpoint scaling	0.15	0.2	0.18
SORW	Critical saturation values used in endpoint scaling	0.15	0.2	0.18
P1X	X coordinate of well PRO-1 / *Quantitative Discrete*	6	11	8
P1Y	Y coordinate of well PRO-1 / *Quantitative Discrete*	21	23	22
Z	Permeability distribution map / *Qualitative Discrete*	1	9	1

Table 2.1: PUNQS case study parameters.

2 years. Input experimental design is maximinLHD of size 100 and additional 200 confirmation runs were performed to test the prediction accuracy. Predictivity index Q_2 was selected as a criterion of prediction accuracy. Here, we consider two simulator output for approximation: Total Field Oil Production (FOPT) and Field Water Cut (FWC). FOPT is a cumulative oil production calculated at the end of every year. FWC stands for a ratio of total volume of water produced by year to total volume of liquid produced by year. Table (2.2) summarized the results. The table provides

estimation of the hyperparameters corresponding to every variable and the computed predictivity indices Q_2.

year	1	2	3	4	5	year	1	2	3	4	5
θ_1	20	20	20	12.66	10.71	θ_1	12.19	19.92	17.50	13.82	8.51
θ_2	16.36	4.82	3.49	4.61	5.30	θ_2	15.77	8.44	7.93	7.88	6.63
θ_3	20	20	20	18.10	20	θ_3	20	20	20	20	20
θ_4	20	20	20	20	20	θ_4	17.85	20	19.27	19.15	20
θ_5	20	20	5.99	4.55	6.02	θ_5	6.27	5.54	8.08	10.10	17.65
θ_6	18.96	8.68	17.28	19.66	17.61	θ_6	15.70	17.32	20	20	20
θ_7	6.31	5.60	3.77	4.06	3.73	θ_7	16.35	19.02	10.45	7.35	6.14
θ_8	5.24	2.02	1.56	1.48	1.80	θ_8	1.48	1.72	1.95	2.39	2.79
θ_9	14.85	20	13.98	15.36	5.05	θ_9	2.56	3.40	3.87	4.52	4.90
θ_{10}	0.25	0.06	0.05	0.05	0.05	θ_{10}	0.05	0.05	0.05	0.05	0.05
$\mathbf{Q_2}$	**0.97**	**0.94**	**0.85**	**0.89**	**0.90**	$\mathbf{Q_2}$	**0.85**	**0.87**	**0.88**	**0.89**	**0.89**

Table 2.2: Predictivity indices by years for FOPT and FWC.

For both considered functions, the predictivity indices values range from 0.85 to 0.97. It corresponds to high level of prediction accuracy. If we have a look at the hyperparameters calculated over the years, the estimations vary significantly from one factor to another. Parameter Z provides the lowest value of correlation parameter: θ_{10}. At the same time, **AQUI**, **MPH2**, **MPV1**, **MPV2** and **SORG** have the least influence to the response outputs.

2.6 Conclusion

In this chapter, we introduced a GP response surface model for modelling experiments involving discrete and continuous parameters. We adapted a general family of correlation functions in case of ordered discrete parameters proposed by Qian et al. [2008]. It provides a correlation function involving continuous and discrete variables. As soon as the correlation function is available, classical GP modelling can be applied.

The proposed integrated approach was tested on analytical and reservoir simulation model case examples. The method outperforms independent approach with a proven prediction efficiency at a lower size of the input experimental design. The numerical results showed higher prediction accuracy at lesser CPU time. We have

also conducted a sensitivity analysis for the transformation function selection. The presented results showed a low sensitivity of the model predictivity to the choice of the transformation function.

The important application of the suggested approach can be found in modelling of time-series outputs considering the time variable as an auxiliary discrete input variable. Moreover, regarding engineering application, the problem of optimal well placement can be solved by adding well grid coordinates as additional quantitative discrete variables. It will be discussed more in details in the following chapter.

Chapter 3

Response Surface modeling for time-series outputs

Contents

3.1 Introduction

Simulation models are used nowadays in many industrial applications to predict and analyze the behavior of complex systems. A simulator is a complex computer code embedding the physical laws governing the physical system under investigation. The

input of such simulators can be adjustable or uncontrollable parameters which are only partially known and thus are affected by uncertainty. Uncertainty analysis of numerical experiments is used to assess the confidence of the model outcomes which are then used to make decisions [De Rocquigny et al., 2008]. Here, we focus on a particular type of dynamic simulators that are used to make predictions in the future. These simulators are based typically on finite element / finite difference codes (for instance, for the simulation of flows and transfers in porous media). Industrial applications using this type of simulators can be hydrocarbons reservoir forecasting and carbon dioxide (CO_2) underground storage [Busby and Sergienko, 2010; Sergienko and Busby, 2011]. Such applications involve very complex numerical codes with a large number of inputs and with high uncertainty derived from an incomplete knowledge of subsurface formation [Subbey et al., 2004]. The uncertainty on the simulator output is usually assumed to be mostly due to the propagation of the uncertainty on the input. Nevertheless, modeling errors can also play a significant role.

In our frame, the dynamic simulator models a multi-phase 3-D fluid flow in heterogeneous porous media, operating over fixed number of time-steps. The typical output of such simulators consists of a sequence of outputs at different time-steps. Therefore, it represents time series related, for instance, to a recovery rate for a given well or a group of wells. It can be also a spatial output such as a pressure map or a saturation map also evolving with time. Here, we focus on 1D time series output which can be typically measured in a well.

Let us consider the output of a simulator as a deterministic function $Y(t) = F(\mathbf{x}, t)$, where $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ is a configuration of preselected input parameters and $0 < t \leq T$ refers to a time step. $Y(t)$ is a time dependent output, e.g. oil production rate or reservoir pressure.

The function $F : \mathbb{R}^d \times]0, T] \rightarrow \mathbb{R}$ models the relationship between the input and the output of a numerical simulator. Methods such as Monte Carlo ones can be used to propagate uncertainty. However, each model evaluation being generally very time-consuming this approach can be unpractical in industrial applications. Moreover, the probability distribution of the input may vary as new data comes in or within a process of dynamic data assimilation. Therefore, a response surface approach can be a reasonable alternative in such applications [Busby and Feraille, 2008; Busby and Sergienko, 2010; Busby et al., 2008].

The advantage of a response surface model is that it is fast to evaluate and it is designed to approximate the complex computer code based on a limited number

of simulator evaluations. These evaluations of the simulator are taken at some well chosen input configurations also called the training set or the experimental design. Numerous experimental designs have been proposed and many of them are quite sophisticated. Here, we use Latin Hypercube designs [McKay et al., 1979] for their good space filling properties. Usually, it can be coupled with other criteria such as the maximin design (maximum minimum distance) [Sacks et al., 1989; Santner et al., 2003]. In this work, we focus on Gaussian process (GP) based response surface models also known as *kriging* [Matheron, 1963; Sacks et al., 1989; Santner et al., 2003].

The aim of this chapter is to propose a new method to address time series outputs. For such dynamic simulator a standard approach assumes a single step GP model for each time step. This basic approach can be computationally intensive for a high number of time steps especially when a single GP model evaluation is highly time consuming (e.g. the size of the training set and the number of variables are large). Therefore, this can become unpractical in some industrial applications.

The problem of response surface modelling for dynamic simulators was recently investigated by different authors. Several principal approaches can be distinguished. A first possible approach is GP modelling considering the time steps as the model additional input parameter [Busby and Sergienko, 2010; Conti and O'Hagan, 2010; Qian et al., 2008]. This approach is easy to implement, however to take into account all the information from an experimental design and every time steps, the size of new resulting experimental design is multiplied by the number of time steps. In case of large size of original experimental design or in case of high density of the steps in the time scale, it leads to matrices of high dimensions. That can lead to numerical problems because of matrix inversion. Conti et al. [2009] developed an iterative approach to build a model of dynamic computer code, assuming that the model output at a given time step depends only on the output at the previous time step (Markovian assumption). Bayarri et al. [2007] introduced wavelet decomposition to model time series outputs. Campbell et al. [2006], Higdon et al. [2008] and Lamboni et al. [2009] suggested application of principal component decomposition. Auder et al. [2011] extended this approach by a preliminary data classification.

In this work, we propose and compare two different methods to model time series outputs. Firstly, we extend the method introduced in the previous Chapter 2. This work was already presented in Sergienko and Busby [2011] and Busby and Sergienko [2010]. In Section 3.2 we provide a new specification of the algorithm for application

to time series outputs with a detailed explanation of the proposed algorithm. We also consider the possible issues and discuss how to avoid them in Section 3.2.1. We show the method efficiency with two different reservoir simulation models. One of them is PUNQ-S3 reservoir model that we used in Section 3.2.2. Another one is presented in Section 3.2.3, where we introduce a new method to address the problem of an injection well placement for CO_2 storage projects.

Further, we provide a new functional approach to model time series outputs of a dynamic simulator in Section 3.3. This approach involves a combination of Shape Invariant Model (SIM) and the Gaussian Process (GP) response surface modelling. This model assumes a common pattern shape curve and curve-specific differences of time series outputs in amplitude and timing are modeled with linear transformations. We provide a model description in Section 3.3.2 and an efficient algorithm of the model transformation parameters estimation in Section 3.3.2.1. Next, in Section 3.3.3 we discuss how to model time series outputs and provide an effective prediction algorithm. The method was tested with a CO_2 storage reservoir case using an industrial commercial simulator and compared with a standard single step approach in Section 3.3.4.

Finally, Section 3.4 summarizes the results of this chapter and highlights advantages and drawbacks of both proposed approaches.

3.2 Time as a discrete parameter

This section is inspired by the work of Qian et al. [2008]. In Chapter 2 we discussed the construction of a correlation function involving both continuous and discrete variables. Here, we extend the method applying it for time series outputs modelling. We consider time as an auxiliary discrete input variable. We modify the algorithm in order to avoid possible computational issues.

To begin with, we consider a dynamic simulator as an output of a deterministic function: $Y_t = F(\mathbf{x}, t)$, where $\mathbf{x} = (x_1, .., x_d) \in \Omega \subset \mathbb{R}^d$ is a vector of input variables and $t : \quad 0 < t \leq T$ is a time step. The number of time steps is finite and fixed $t \in \{1, \ldots, T\}$. We suppose that t is equispaced in the time interval $]0, T]$. The output of the dynamic simulator for a given input variables configuration $\mathbf{x}$ is a vector: $\{Y_1, \ldots, Y_T\}$ representing the simulator evaluations at a given configuration $\mathbf{x}$ and at every considered time step. This vector of outputs can be considered as

time series.

The standard approach to model a dynamic simulator is a single step GP modelling. It assumes, that for every fixed t we consider a single output function Y_t. For such simulator, we can build a GP model based on a given experimental design set: $\{\mathbf{X}^n, \mathbf{Y}_t^n\}$. Doing it iteratively for every fixed t, we can cover all the time domain. This approach has the following disadvantages. Firstly, the resulting GP response surface model is constructed for a particular time step t. Secondly, for the complex output function a single model evaluation can be time consuming. In such case, single step modelling approach can be unfeasible especially when there is a large number of time steps. Here, we introduce a new algorithm and compare its prediction performance with a single step GP model considered as a benchmark.

Firstly, let us present what is an experimental design for a dynamic simulator. For an experimental design $\mathbf{X}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \quad \mathbf{x}_i \in \Omega \subset \mathbb{R}^d, i = 1, \ldots, n$ we run the simulator for every experimental unit: $\mathbf{x}_i \in \mathbf{X}^n, \quad 1 \leq i \leq n$. We collect then the output at each time step $t \in \{1, \ldots, T\}$. So that, the output $\mathbf{Y}^n$ for a given experimental design $\mathbf{X}^n$ is not a single vector, but the matrix (3.1). Every row (for a fixed i): $\mathbf{Y}^n[i,] = F(\mathbf{x}_i, t), \quad 1 \leq t \leq T$, represents a discrete curve evolving over time t.

$$\mathbf{X}^n = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} \quad \mathbf{Y}^n = \begin{pmatrix} F(\mathbf{x}_1, 1) \ldots & F(\mathbf{x}_1, t) \ldots & F(\mathbf{x}_1, T) \\ \vdots & \vdots & \vdots \\ F(\mathbf{x}_n, 1) \ldots & F(\mathbf{x}_n, t) \ldots & F(\mathbf{x}_n, T) \end{pmatrix} \tag{3.1}$$

To model time series outputs we propose to apply the method introduced in Chapter 2. Indeed, we can consider the time parameter t as an additional discrete variable with the corresponding set of possible levels: $\{1, \ldots, T\}$. Moreover, this set is already ordered and we can apply the correlation function for ordered discrete variables with a linear transformation function. So that, a valid correlation function is automatically available. The input variables domain Ω is replaced then by $\Omega \times]0, T]$ and we can build a unique single output GP model based response surface model. This model can be used to make a prediction for any arbitrary input $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ at any time step $t \in \{1, \ldots, T\}$. The standard algorithm is completely similar to the original one presented in Chapter 2.

The method is straightforward and easy to implement. Nevertheless, a direct application of the method can lead to some numerical problems. When we are working

with fitting a Gaussian process based model, the computational constraints typically arise from the size n of the experimental design set $\mathbf{X}^n$. The main computational problem is the matrix inversion during the maximum likelihood estimation. In order to achieve higher prediction performance, the aim is to account all the available observations data from all the time steps. So that, we should transform the columns of observations matrix $\mathbf{Y}^n$ for a given experimental design $\mathbf{X}^n$ into a vector by joining sequentially every column $\{\mathbf{Y}^n[,1],\ldots,\mathbf{Y}^n[,T]\}$. Going over all the time steps, we obtain a new vector of observations: $\mathbf{Y}^n_t = (\mathbf{Y}^n[,1]^\top,\ldots,\mathbf{Y}^n[,T]^\top)^\top$. Now, we have to inverse a $n\mathrm{T}\times n\mathrm{T}$ matrix. In case of a large number of time steps or high dimension of input parameters space (the associated experimental design size n is large), the matrix inversion can be very computationally demanding. It can lead to misestimating of the model parameters. Hence, the direct application of the method is restricted by the size of the original experimental design n and the number of the time steps T involved in the dynamic simulator. To avoid this problem, we propose a modification of the algorithm.

In our work, the original experimental design is a maximin Latin hypercube design (LHD) introduced in Chapter 1. This provides a uniform coverage of the input variables domain $\Omega \in \mathbb{R}^d$. Working with a dynamic simulator, for a given experimental design $\mathbf{X}^n$ the information at every time step from the time interval $]0,T]$ is available. So that, we wish to keep all the available information and to include the time as an auxiliary discrete variable. The size of the new experimental design is then $nT\times(d+1)$. It corresponds to a combination of the original LHD of the input uncertain parameters $\mathbf{X}^n$ with the vector of all the time steps $t=\{1,\ldots,T\}$.

$$\mathbf{X}_t = \begin{pmatrix} & 1 \\ \mathbf{X}^n & \vdots \\ & 1 \\ \cdots & \cdots \\ & T \\ \mathbf{X}^n & \vdots \\ & T \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 & 1 \\ \mathbf{x}_i & \vdots \\ \mathbf{x}_n & 1 \\ \cdots & \cdots \\ \mathbf{x}_1 & T \\ \mathbf{x}_i & \vdots \\ \mathbf{x}_n & T \end{pmatrix} \quad \mathbf{Y}_t = \begin{pmatrix} \mathbf{Y}^n[1,1] \\ \vdots \\ \mathbf{Y}^n[n,1] \\ \cdots \\ \mathbf{Y}^n[1,T] \\ \vdots \\ \mathbf{Y}^n[n,T] \end{pmatrix} \tag{3.2}$$

The main reason of computational problems is the high number of time steps T and the large size n of original experimental design $\mathbf{X}^n$. As a modification of the algorithm, we propose to selectively take out the information about the function

behavior at the time scale by combining two experimental designs (on the input variables space $\Omega \in \mathbb{R}^d$ and the time interval $]0,T]$). The objective is to reduce the total size of the new experimental design with time considered as an auxiliary variable but keeping the same information as it is provided with (3.2). So that, we reduce the size of new experimental design from $n \times T$ to n_{new}, where n_{new} is the size of the new experimental design on $\Omega\times]0,T] \in \mathbb{R}^{d+1}$. To keep a uniform coverage of the input variables space $\Omega \in \mathbb{R}^d$, we recommend to choose n_{new} to be a multiple of the size of the original design n: $n_{new} = n \times n_{train}$ with $n_{train} < T$.

We generate a new design vector of time steps $\mathbf{t}^{n_{new}} = (t_1, \ldots, t_{n_{new}})$ as a Latin hypercube permutation at the time interval $t \in \{1, \ldots, T\}$. Then we combine $\mathbf{t}^{n_{new}}$ with $\mathbf{X}^n$. By a proper selection of n_{new} and by a proper permutation $\mathbf{t}^{n_{new}}$, we can keep the same level of information about the function at $\Omega\times]0,T]$ avoiding computational problems. For this algorithm, we do not have to conduct any additional simulator runs because we use only available information from $\mathbf{Y}^n$ and $\mathbf{X}^n$.

As soon as we have this new experimental design, we can apply a standard GP response surface modelling for a single output function: $f_t(\mathbf{x}_t) = F(\mathbf{x}, t)$, where $\mathbf{x}_t \in \Omega\times]0,T] \subset \mathbb{R}^{d+1}$ involves both discrete and continuous variables. The complete algorithm is formalized in (3.2).

Algorithm 1 Algorithm for the GP modelling with time as an auxiliary variable.

Input: Dynamic simulator: $Y(t) = F(\mathbf{x}, t), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad t \in]0,T]$

Output: Unique GP response surface model

Create a design $\mathbf{X}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ to span the domain $\Omega \in \mathbb{R}^d$

Run the simulator $\mathbf{Y}^n = F(\mathbf{X}^n, t)$ at design points for $t = 1, \ldots, T$

Choose the size of new design: n_{train} and $n_{new} = n_{train} \cdot n$

Create a design $\mathbf{t}_{n_{new}} = \{t_1, \ldots, t_{n_{new}}\}$ to cover the time interval $]0,T]$

Create a multiple design: $\mathbf{X}^{n_{new}} = \{\underbrace{\mathbf{X}^n, \ldots, \mathbf{X}^n}_{n_{train} \quad times}\}$

Combine $\mathbf{X}_t^{n_{new}} = (\mathbf{X}^{n_{new}}, \mathbf{t}_{n_{new}})$

Extract the vector of output $\mathbf{Y}^{n_{new}} = \mathbf{Y}^n[\cdot, \mathbf{t}_{n_{new}}]$

Conversion of all input discrete variables with a linear transformation modelling

Classical GP response surface model basing on new design

Return: Unique GP model formulation

With a reasonable choice of n_{new} (so of n_{train}) we can keep as much information about the function as possible. The performance depends on the choice of n_{train}. If n_{train} is too small it leads to lower prediction efficiency due to the lack of the information. If n_{train} is too high it can lead to numerical problems. We have tested how the choice of n_{new} affects the prediction efficiency of the method. We discuss it in the Section 3.2.2 on a reservoir model case example.

While working with this method in different applications, we have encountered some difficulties. We point out few of them and discuss possible solutions to avoid them.

3.2.1 Possible difficulties

3.2.1.1 Covariance structure

Let us consider the covariance between two different points in space and time $\mathbf{x}_{t1} = (\mathbf{x}_1, t_1)$ and $\mathbf{x}_{t2} = (\mathbf{x}_2, t_2)$ and the corresponding outputs $Y_{t1} = F(\mathbf{x}_1, t_1)$ and $Y_{t2} = F(\mathbf{x}_2, t_2)$. With the selected structure of the correlation function, the covariance is expressed by a product of the variance σ^2 of the stochastic process and the correlation function:

$$\mathrm{R}_t(\mathbf{x}_{t1}, \mathbf{x}_{t2}) = \exp\left(-\sum_{i=1}^{d+1} \frac{|\mathbf{x}_{t1_i} - \mathbf{x}_{t2_i}|^2}{\theta_i}\right).$$

The hyperparameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ corresponds to the correlation lengths of the input variables $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ and θ_{d+1} corresponds to correlation length of the time variable. The hyperparameters affect how far the sample point influence extends. A high θ_i means that all points have a high correlation and the corresponding outputs being similar over the sample. While a low θ_i means that there is a significant difference between the different corresponding outputs. By examining the estimations of the hyperparameters, one can decide what are the parameters that have the most influence (and perhaps eliminate non influential variables). When we include time as an auxiliary parameter, the correlation lengths of other input uncertain variables can be misestimated. The reason is the output function dependence over the time interval $]0, T]$. The difference of the function evaluations at two different points in time can be considerably different.

3.2.1.2 High density of the time domain

Another possible issue is a high density of time step points in the time interval. When we generate a new experimental design with included time variable, we collect all the data from all the time step points. Number of the time steps can be large. For example, with a case of CO_2 storage, the reservoir model simulation period can be up to hundreds or thousands of years. Another example is the method application to the sound analysis where the data are gathered every minute or even every second. In such case, for a new combined experimental design we have to include all the possible time variable values from $\{1, \ldots, T\}$. Thus, if the choice of a new design size n_{new} is not sufficient, the prediction efficiency of resulting unique GP model can be poor in view of the lack of information.

In such case, we propose to prioritize the information from different time steps and include only the time steps with the relevant information. So that, we reduce the total number of points T to a less number n_t (number of included time steps). By reducing the number of involved time steps, we avoid the possible numerical problems. It is worthwhile noting that the transformation from the discrete to continuous variables should be still performed with a transformation function considering the original number of levels $\{1, \ldots, T\}$. In the case when the information from all the time steps are of equal importance, we propose an alternative way to address this issue in the following subsection.

3.2.1.3 Time domain division

Sometimes, it is not possible to allocate all the data to account all the design points over the time interval. This can be due to computational difficulties. In other cases, the simulator output has significantly different behavior over the selected time interval $]0, T]$. As discussed before, the hyperparameters can be misestimated in such cases. Here, we propose to divide time domain into a set of non intersecting time intervals: $]T_i, T_{i+1}]$, where $\bigcup_{i=1}^{n_{intervals}}]T_i, T_{i+1}] =]0, T]$. After that, we consider different GP models on the different time subdomains. The main question is how to select the time domain partition?

Different algorithms for division can be performed. For example, in case of quite similar output function behavior over the time interval, partition into equal subintervals is a reasonable choice. For a case, when the function behavior is different over the time interval, change points analysis can be applied. So that, the time

domain is split by the points corresponding to the function behavior changes. For complicated cases, statistical classification methods may be used.

There are different possible classification algorithms. To keep things simple, we consider one of the simplest: the $k - means$ classification algorithm. First, we should choose the number of possible clusters $n_{clusters}$. Then, we consider a vector of empirical mean or empirical variance, calculated on the observation data matrices. The number of clusters could be chosen by the method of trials and errors. Starting from different possible assumptions and subsequent analysis the combined model prediction performance. From the practice, we can say that sometimes even a partition in two clusters can significantly increase the model predictivity. We have tested this method on a reservoir case example. We present results in the following section.

3.2.2 Reservoir case example

Firstly, we consider the reservoir model PUNQ-S3 for numerical approbation of the method. The model was previously introduced in Section 2.5.

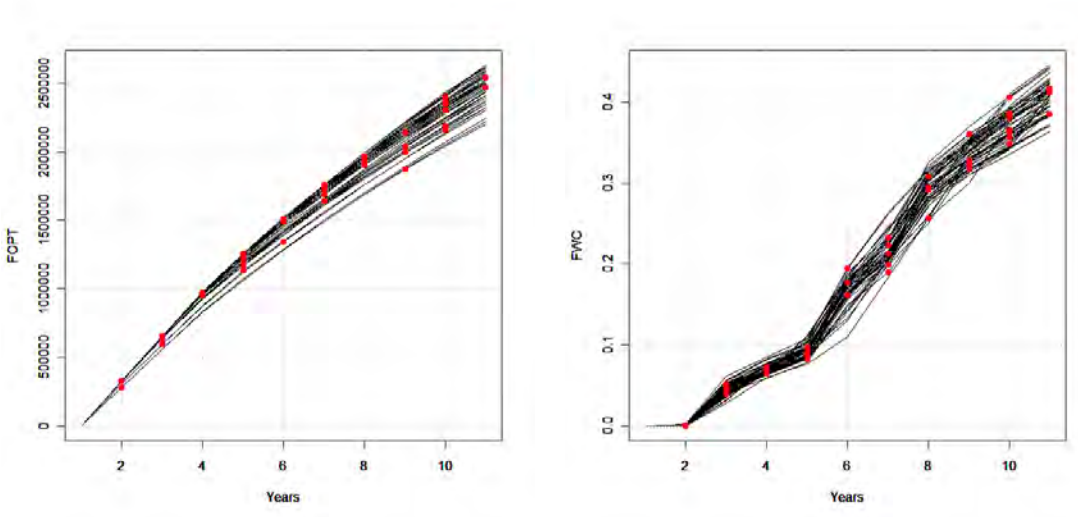
We consider 9 uncertain parameters including four discrete parameters with five possible values. Discrete parameters correspond to wells coordinates. Table (3.1) contains all the parameters used in this study. The selected uncertain parameters describes different possible fluid flows and the reservoir conditions. The reservoir model simulation period is 10 years that correspond to 10 time steps.

Name	Description	Min	Max	Def
AQUI	Analytical porosity of the aquifer strength	0.2	0.3	0.2
MPH1	Horizontal transmissibility multiplier in good units	0.8	1.2	1.0
MPH2	Horizontal transmissibility multiplier in poor units	0.8	1.2	1.0
MPV1	Vertical transmissibility multiplier in good units	0.8	1.2	1.0
MPV1	Vertical transmissibility multiplier in poor units	0.8	1.2	1.0
SORG	Critical saturation values used in endpoint scaling	0.15	0.2	0.18
SORW	Critical saturation values used in endpoint scaling	0.15	0.2	0.18
P4X	X coordinate of well PRO-4 */Discrete*	7	11	9
P4Y	Y coordinate of well PRO-4 */Discrete*	15	19	17
P12X	X coordinate of well PRO-12 */Discrete*	13	17	15
P12Y	Y coordinate of well PRO-12 */Discrete*	10	14	12

Table 3.1: PUNQS case study parameters.

For an experimental design we select a maximin LHD $\mathbf{X}^n$ of size 50. The matrix

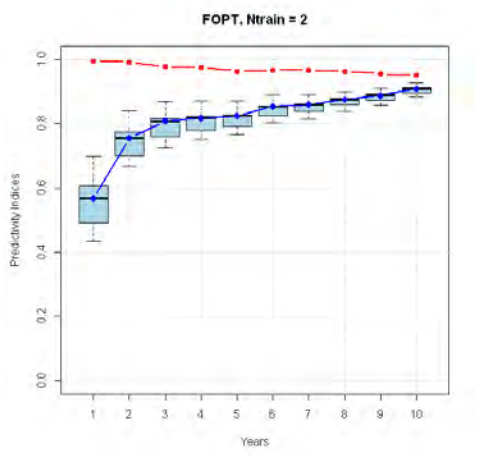
with corresponding simulator outputs is $\mathbf{Y}^n$ - $[50, 10]$ matrix. Additional 100 confirmations runs is conducted to test the prediction performance. Predictivity index Q_2 was computed for every year. We consider 10 single step GP response surface models for every year as a reference case. The selected output functions for analysis are the same as in Example 2.5: Field Oil Total Production (FOPT) and Field Water Cut (FWC). These functions are presented in Figure 3.1. To illustrate the choice of new design, we have presented it in Figure 3.1 by red dots for $n_{train} = 1$.



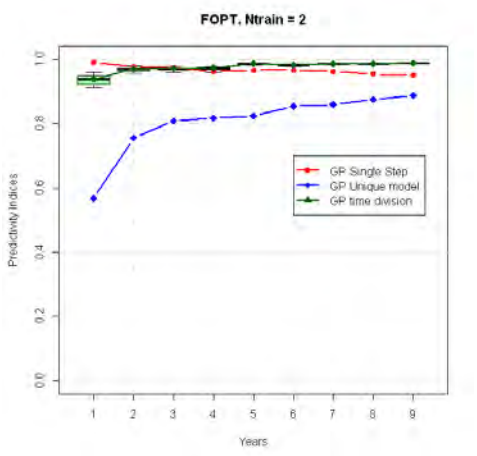
(a) FOPT (b) FWC

Figure 3.1: The functions of FOPT and FWC.

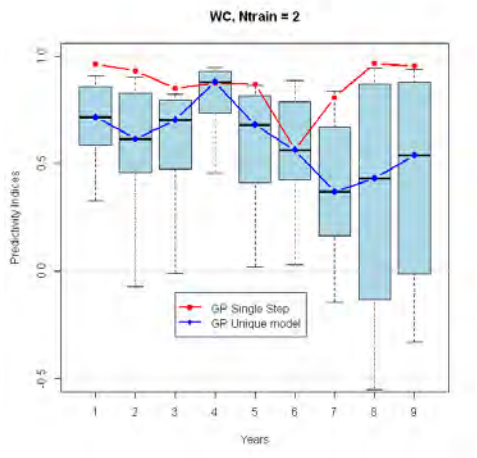
We have studied how the choice of n_{new}, or n_{train}, influence the model prediction efficiency. For a fixed $n_{train} = 1,\ \ 2,\ \ 3$ we have conducted 50 different integrated GP model evaluations. The only difference of these models is a sampling of t variable. Figures 3.2(a) depicts the boxplots of evaluated predictivity indices Q_2 for the function of FOPT for $n_{train} = 2$. Figure 3.2(b) displays the results for the same function assuming two different model on two time intervals. The red dots are the predictivity indices computed for the reference model. The green and blue dots are the empirical means of Q_2 evaluations for the boxplots with and without splitting the time domain. Figure 3.2(c) represents the same case for FWC with $n_{train} = 2$. It can be seen that the predictivity varies from one sample to another and some values corresponds to very low predictivity. Whereas, by splitting the time domain into 2 equal intervals the predictivity is significantly improved. For some time points this model even outperforms the classical single step approach. Q_2 estimations for the rest of cases could be found in Appendix A, section A.1.



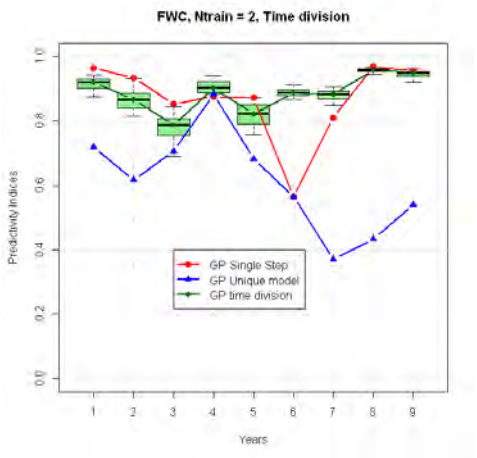
(a) FOPT, $n_{train} = 2$



(b) FOPT, $n_{train} = 2$ with time scale division



(c) FWC, $n_{train} = 2$



(d) FWC, $n_{train} = 2$ with time scale division

Figure 3.2: Predictivity indices estimations for different sizes of combined sampling.

It can be concluded that the prediction efficiency depends on the choice of n_{train}. For the presented cases the higher size of new experimental design means the higher prediction performance. Worthwhile noting, that starting from some value of n_{train}, the prediction efficiency decreases due to arising matrices of quite high dimension.

As the main conclusion, comparing crossplots and Q_2 it is clear that this new approach provides the same level of predictivity with the less time-costs. Predictivity could be improved significantly by splitting time domain for different GP models. In

this case, the predictivity is even better than the reference case. Table (3.2) represents CPU time comparison for independent single step approach and the proposed approach.

	Unique GP model	Single-step modelling
FOPT	00:01:44	00:08:40
FWC	00:01:18	00:08:52

Table 3.2: CPU time comparison

3.2.3 Injection well placement optimization

Carbon Capture and Storage technology (CCS) stands for collection of CO_2 from industrial sources and its injection underground. Carbon dioxide is stored in a deep geological formation that is sealed on a top by a low permeability cap [Busch et al., 2010]. Subsurface storage of CO_2 is always associated with an excess pressure in the reservoir and one of the primary environmental risks is a pressure-driven leakage of the gas from the storage formation [Benson, 2006; Bowden and Rigg, 2004].

The problem of an injection well placement is an important issue that can contribute significantly to the risk mitigation. In order to assess the risk of CO_2 leakage through the caprock we can simulate different potential well placement scenarios. After that, the injector is placed providing the minimum risk. However, it can be infeasible in case of complex simulation models and numerous possible placements for the reason of an excessive simulation time. At the same time, response surface modelling is intended to approximate complex and computationally demanding simulator codes basing on a limited number of simulations with lower time costs. Here, we focus on Gaussian Process (GP) model. We propose to approximate the reservoir simulator output with the response surface model by taking well positions as controllable discrete parameters and a time step as an auxiliary model variable. So that, instead of looking over all the possible combinations of the well positions, we run a unique model for the analysis of possible pressure evolution. The suggested approach leads to considerable time-savings reducing the required number of simulations.

For the problem of well placement the output of the simulator can be represented by $F(\mathbf{x}, \mathbf{y}, t)$, where the set $\mathbf{x} \in \Omega_x \subset \mathbb{R}^{d1}$ represents collectively the uncertain pa-

rameters describing the reservoir: porosity, permeability etc. The set of parameters $\mathbf{y} \in \Omega_y \subset \mathbb{R}^{d2}$ are man-controlled discrete parameters (the well position in our case) and $t \in]0, T]$ corresponds to time step. The objective is to place a well with a minimized risk of leakage. Therefore, the objective function to approximate is the reservoir overpressure increase in the upper layer of the reservoir. When the response surface model is computed, it should be validated. After that, to be more confident about the leakage risk, we aim to estimate a quantile P90 of $F(\mathbf{x}, \mathbf{y}, t)$ as a function of controllable parameters $\mathbf{y}$ basing on sampling $\mathbf{x}$. The quantile is calculcated by Monte-Carlo sampling of the uncertain parameters $\mathbf{x}$ for every fixed well position $\mathbf{y}$. After that, the optimal well position $\mathbf{y}^*$ is the one that provides a minumum possible pressure evolution. It can be represented as a solution of the following optimization problem:

$$\mathbf{y}^* = \arg \min_{\mathbf{y} \in \Omega_y} \max_{t \in]0,T]} (\widehat{F}_{90}(\mathbf{y}, t)).$$

The method was tested on a reservoir model of potential CO_2 storage site in Europe. The reservoir simulation model was constructed from incomplete data and can include significant uncertainties. The model is realized by a coarse grid resolution of 800m × 800m cells and the grid dimensions are: $N_x \times N_y \times N_z = 43 \times 52 \times 52$. Two facies are used in the model: sand and shale. Initial storage pressure was assumed to be at the hydrostatic conditions (100 bars at 1000 m). The CO_2 injection is supposed to be at a constant rate for 30 years. Table 3.3 summarizes the parameters considered for this study.

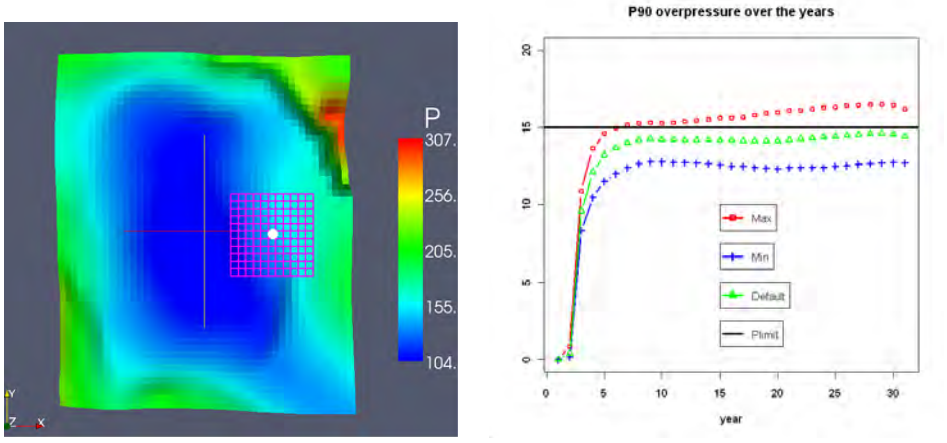
Name	Description	Min	Max	Def
PHI	Porosity multiplier	0.5	1.5	1
K	Permeability multiplier	0.8	1.2	1
Nsand	N exponent for sand facies	1	3	2
Nshale	N exponent for shale facies	2	4	3
INJ1X	X coordinate of injection well	26	36	31
IN1Y	Y coordinate on injection well	22	32	27

Table 3.3: CCS study parameters.

Figure 3.3(a) displays the original pressure distribution map. The injection well is to be placed inside the predetermined region. Each coordinate includes 11 possible levels. The uncertain parameters $\mathbf{x}$ selected for this study characterize the reservoir and the fluid properties. This implies different CO_2 flowing possibilities between the

reservoir layers. The well position cell number refers to the controllable parameters to be optimized.

Following the engineering safety criteria, the margin on the cap rock allows a pressure increase of about 15 % compared to static condition. Therefore, the maximum allowed overpressure is $\mathbf{P}_{limit} = 15$ bars. Exceeding this value can lead to a leakage. The first step is an approximation of the overpressure in the upper



(a) Original reservoir pressure distribution

(b) The best, the worst and the default positions

Figure 3.3: Injection well placement study.

layer by a GP response surface model and the model validation. After that, if the satisfactory predictivity is achieved, by fixing controllable parameters and Monte-Carlo sampling of the uncertain parameters, we can go over all the possible 121 coordinates combinations and estimate P90 of these realizations. Afterward, the optimum well position can be identified.

Figure 3.3(b) represents the possible overpressure evolutions. It compares the overpressure development over the simulation period for the best, the worst and the default well position that was used in the original simulation model. The optimal well position corresponds to $[X, Y] = [36, 32]$. Whereas, placing an injection well in $[X, Y] = [26, 32]$ can lead to a leakage.

By this section, we propose a new method for an injection well placement optimization under uncertainty, particularly CO_2 injection well placement. The method is based on a GP response surface approximation of the reservoir overpressure basing on a limited set of reservoir simulator runs.

We applied the method to the potential CO_2 storage reservoir model. The injection well position is determined under minimizing the risk of apossible leakage. To conclude, this example demonstrated a reliable level of predictivity. Moreover, including controllable parameters as additional variables allows to considerably reduce the required number of simulations. The method provides significant time savings compared to the standard approach.

Unique GP model	Single step modelling
00:00:37	00:07:57

Table 3.4: Time costs comparison.

3.3 Shape invariant model approach

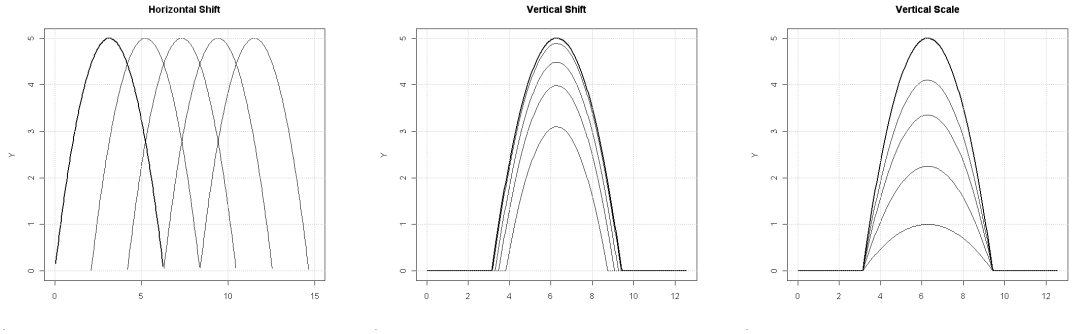
3.3.1 Introduction

In this section, we discuss the shape invariant model representation and the procedure for efficient parameters estimation.

The shape invariant model was introduced by Lawton et al. [1972]. The model assumes that we are working with a set of curves that have a common shape function that is modeled nonparametrically. The deformation of this function is modeled parametrically by choosing a proper parametrical transformation. We consider a class of linear transformations only. These parameters can be normally interpreted as a shift in timing or a scale in amplitude. For this reason, shape invariant model is widely applied to study periodic data such as temperature annual cycle [Vimond, 2010] or traffic data analysis [Gamboa et al., 2007]. Indeed, in these cases there is always some variability in time cycles or amplitude. The model has also been used to study biological data , where the departure from the pattern can be caused by a different environmental conditions [Brumback and Lindstrom, 2004; Izem and Marron, 2007]. In this work, we use the model to represent reservoir simulator output data. For example, if we consider the field water cut that is the ratio of produced water compared to the volume of total liquids produced. In this example, the shift in time is caused by different moment of water breakthrough. Whereas, for the cumulative oil production function we can clearly observe vertical scaling

transformation for different flowing conditions in the reservoir model. These flowing conditions are defined by selection of input uncertain parameters. So that, we can study the influence of the model input parameters on the overall shapes of functional for curves the selected output.

In figure (3.4) we display three possible transformations that we consider in our work: horizontal shift 3.4(a), vertical shift 3.4(b) and vertical scaling 3.4(c). The bold line represents original pattern shape.



(a) Horizontal shift (b) Vertical shift (c) Vertical scale

Figure 3.4: Parametrical transformation examples.

We are interested in the common shape as well as in the efficient estimation of transformation parameters. So that, we can reproduce any curve and align it to the pattern. Moreover, we can make a prediction for an input configuration $\mathbf{x}_0$ by modelling the transformation parameters.

As discussed, for an experimental design $\mathbf{X}^n = \{\mathbf{x}_1, .., \mathbf{x}_n\}$, we have a set of observations $\mathbf{Y}^n = \{Y_{i,j} = F(\mathbf{x}_i, t_j), \quad 1 \leq i \leq n, \quad 1 \leq j \leq J\}$, where $\mathbf{x}_i$ is the set of preselected input parameters and t_j refers to a time step. So that, $Y_{i,j}$ is the j^{th} observation on the i^{th} experimental design unit, with $1 \leq i \leq n$ and $1 \leq j \leq J$. Thereby, the idea is to find a general pattern curve that make possible subsequent transformations of any curve to this selected pattern with properly adjusted transformation parameters.

We focus here on linear transformations. Thus, the model structure may be written as:

$$Y_{i,j} = \alpha_i^* \cdot f(t_j - \theta_i^*) + v_i^* + \sigma_i^{*2} \cdot \varepsilon_{ij} \tag{3.3}$$

where t_j is observation time which assumed to be known and equispaced in the interval $]0, T]$. The vector of parameters: $(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*}) = (\alpha_1, .., \alpha_n, \theta_1, .., \theta_n, v_1, .., v_n)$

is unknown as well as the pattern function f . The errors ε_{ij} are i.i.d. with a normal distribution $\mathcal{N}(0,1)$, $(i,j) \in \{1,..,n\} \times \{1,..,J\}$. It represents the observation noise and without loss of generality we assume that $\sigma^2 = 1$. The variance does not affect exactly on the parameters estimation procedure and the method still works for a variance σ^2. The function f is assumed to be 2π-periodic [Gamboa et al., 2007; Vimond, 2010].

In this section we provide the algorithm for efficient estimation of the transformation parameters under unknown function pattern f. Since the functional pattern f is unknown, the pattern is replaced by its estimate. So that, it seems natural to study the problem (3.3) in a semi-parametric framework: the transformation shifts and scales are the parameters to be estimated, while the pattern stands for an unknown nuisance functional parameter. We use an M-estimator built on the Fourier series of the data. Under identifiability assumptions it is possible to provide a consistent algorithm to estimate $(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*})$ when f is unknown. This section is organized as following. Firstly, we describe the method of efficient estimation of transformation parameters of the shape invariant model. The method is illustrated with an example on an analytical function. Then, we present the forecast algorithm for dynamic simulators with a practical application of the algorithm on a CO_2 storage reservoir case.

3.3.2 Model description

Consider $\mathbf{Y}^n = F(\mathbf{X}^n, \boldsymbol{t}) = \{F(\mathbf{x}_i, t_j), \quad 1 \leq i \leq n, 1 \leq j \leq J\}$ is $(n \times J)$ matrix of observations. We model these observations in the following way:

$$Y_{i,j} = \alpha_i^* \cdot f(t_j - \theta_i^*) + v_i^* + \varepsilon_{ij}, \quad 1 \leq i \leq n, 1 \leq j \leq J \tag{3.4}$$

where $f : \mathbb{R} \to \mathbb{R}$ is an unknown 2π-periodic continuous function, $\boldsymbol{\theta}^* = (\theta_1^*, ..., \theta_n^*)$, $\boldsymbol{\alpha}^* = (\alpha_1^*, ..., \alpha_n^*)$, $\boldsymbol{v}^* = (v_1, ..., v_n) \in \mathbb{R}^n$ are unknown parameters, ε_{ij} is a Gaussian white noise with variance equal to 1. The time period is linearly transformed such that $]0, T] \to]0, 2\pi]$, therefore $t_j = \frac{j}{J} 2\pi, \quad j = 1, .., J$ is equispaced in $]0, 2\pi]$.

The objective is to estimate the horizontal shift $\boldsymbol{\theta}^* = (\theta_1^*, ..., \theta_n^*)$, the vertical shift $\boldsymbol{v}^* = (v_1, ..., v_n)$ and the scale parameter $\boldsymbol{\alpha}^* = (\alpha_1^*, ..., \alpha_n^*)$ without knowing the pattern f. Fourier analysis is well suited for the selected structure of the model. Indeed, this transformation is linear and shift invariant. Therefore, applying a

discrete Fourier transform to (3.4) and supposing J is odd, the model becomes:

$$d_{kl} = \begin{cases} \alpha_k^* e^{-il\theta_k^*} c_l(f) + w_{kl}, & 1 \leq k \leq n, \quad 0 < |l| \leq (J-1)/2 \\ \alpha_k^* c_0(f) + v_k^* + w_{k0}, & 1 \leq k \leq n, \quad l = 0 \end{cases} \tag{3.5}$$

where $c_l(f) = \frac{1}{J} \sum_{m=1}^{J} f(t_m) e^{-2i\pi \frac{ml}{J}}$, $\left(|l| \leq (J-1)/2\right)$ are the discrete Fourier coefficients and w_{kl}, $\left(1 \leq k \leq n, \ |l| \leq (J-1)/2\right)$ is a complex white Gaussian noise with independent real and imaginary parts.

We also notice that in order to ensure the identifiability of the model (3.5) we can consider different constraints. In this work, we are working in the following parameters space:
$\mathbf{A} = \{(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) : \alpha_1 = 1, \theta_1 = 0, v_1 = 0, \quad 0 < \theta_i < 2\pi, -\pi < v_i < \pi, 0 < \alpha_i < \alpha_{max}\}$.
This means that we fix the first curve from the experimental design. We assume it as the reference curve and all others are representation of this reference realized by horizontal shifts, vertical shifts and scaling parameters.

The objective of this section is to estimate the transformation parameters ($\boldsymbol{\alpha}^*$, $\boldsymbol{\theta}^*$, $\boldsymbol{v}^*$) without any prior knowledge on the function f. The estimation depends on the unknown functional parameter $(c_l(f))_{|l| \leq (J-1)/2}$, the Fourier coefficients of the unknown function f. So that, we consider a semi-parametrical method based on an M-estimation procedure. M-estimation theory enables to build an estimator defined as a minimizer of a well-tailored empirical criterion that is given in the following subsection.

3.3.2.1 Parameters estimation

The goal is to estimate the vector of parameters ($\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*, \boldsymbol{v}^*$) that depends on the Fourier coefficients of the unknown function f. We consider a semi-parametric method based on an M-estimation procedure [Gamboa et al., 2007].

To construct an M-estimator, we define the rephased (untranslated and rescaled) coefficients for any vector $(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{v}) \in \mathbf{A}$:

$$\tilde{c}_{kl}(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) = \begin{cases} \frac{1}{\alpha_k} e^{il\theta_k} d_{kl}, & 1 \leq k \leq n, \quad 0 < |l| \leq (J-1)/2 \\ \frac{1}{\alpha_k} \left(d_{kl} - v_k\right), & 1 \leq k \leq n, \quad l = 0 \end{cases}$$

and the mean:

$$\hat{c}_l(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) = \frac{1}{n} \sum_{k=1}^{n} \tilde{c}_{kl}(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}), \quad |l| \leq (J-1)/2$$

Therefore, for $(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*})$ we obtain:

$$\tilde{c}_{kl}(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*}) = c_l(f) + \frac{1}{\alpha_k^*} \ e^{il\theta_k^*} w_{kl} \ \ 1 \leq k \leq n$$

$$\hat{c}_l(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*}) = c_l(f) + \frac{1}{n} \sum_{k=1}^{n} \frac{e^{il\theta_k^*} \cdot w_{kl}}{\alpha_k^*}$$

.

Hence, $|\tilde{c}_{kl}(\alpha, \theta, v) - \hat{c}_l(\alpha, \theta, v)|^2$ should be small when $(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v})$ is closed to $(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*})$.

Now, consider a bounded measure μ on $]0, T]$ and set

$$\delta_l := \int_0^T \exp\left(\frac{2i\pi l}{T}\omega\right) d\mu(\omega), \quad (l \in \mathbb{Z}) \tag{3.6}$$

Obviously, $|\delta_l| > 0, l \neq 0$ and this sequence (δ_l) is bounded. Without loss of generality we assume that $\delta_0 = 0$. Assume further that $\sum_l |\delta_l|^2 |c_l(f)|^2 < +\infty$. So that $f * \mu$ is a well defined square integrable function:

$$f * \mu(x) = \int f(x-y) d\mu(y)$$

We construct the following empirical contrast function (3.7):

$$M_n(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) = \frac{1}{n} \cdot \sum_{k=1}^{n} \sum_{l=-\frac{J-1}{2}}^{\frac{J-1}{2}} |\delta_l|^2 |\tilde{c}_{kl}(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) - \hat{c}_l(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v})|^2 \tag{3.7}$$

The random function M_n is non negative. Furthermore, its minimum value is reached close to the true parameters $(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*})$. We define the estimator by:

$$(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}})_n = \arg \min_{(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) \in \mathbf{A}} M_n(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v})$$

The prof of convergence $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}})_n \xrightarrow[n \to +\infty]{\mathbf{P}} (\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*})$ and asymptotic normality of the estimators can be found in Gamboa et al. [2007]; Vimond [2010]. The weights δ_l in (3.6) are chosen to guarantee the convergence of the contrast function to a

deterministic function M_n and to provide the asymptotic normality of the estimators. Moreover, the convergence can be sped up by proper selection of weights. The analysis of convergence at different weights is presented in [Gamboa et al., 2007]. In this work, we used the weights $\delta_l = 1/|l|^\beta$ with $\beta = 1.5$.

The computation of the estimators is quick since only a Fast Fourier algorithm and a minimization algorithm of a quadratic functional are needed. The procedure is summarized in Algorithm (3.3.2.1)

Algorithm 2 Parameters estimation procedure.

Input: Input set of curves from experimental design $\mathbf{Y}^n = \{Y_{ij}, i = 1, .., n; \quad j = 1, .., J\}$

Output: Transformation parameters estimation $(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*})$

Define the identifiability condition: $\mathbf{A} = \{(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{v^*}) \in [-\pi, \pi[^{3\times n}: \alpha_1 = 1, \theta_1 = 0, v_1 = 0\}$

Compute the matrix of discrete Fourier coefficients $D = \{d_{kl}, \quad k = 1, .., n; \quad |l| \leq (J-1)/2\}$

Compute the matrix of rephased Fourier coefficient $\tilde{C} = \{\tilde{c}_{kl}, \quad k = 1, .., n; \ |l| \leq (J-1)/2\}$

Compute the vector of mean of rephased coefficients $\widehat{C} = \{\widehat{c}_l, \quad |l| \leq (J-1)/2\}$

Choose the weight sequence δ_l

Define $M_n(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) = \frac{1}{n} \cdot \sum_{k=1}^{n} \sum_{l=-\frac{J-1}{2}}^{\frac{J-1}{2}} |\delta_l|^2 \, |\tilde{c}_{kl}(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) - \hat{c}_l(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v})|^2$

Compute $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}}) = \arg\min_{(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) \in \mathbf{A}} M_n(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) \in \mathbb{R}^{3\times(n-1)}$

Return: $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}}) \in \mathbb{R}^{3\times(n-1)}$

3.3.2.2 Analytical function example

In this section the shape invariant model and the efficient parameters estimation are presented on an analytical function. The minimization algorithm used in the estimation procedure is a Newton-type algorithm .

Let us consider the following function:

$$f(x) = 20 \cdot (1 - x/(2\pi)) \cdot x/(2\pi)$$

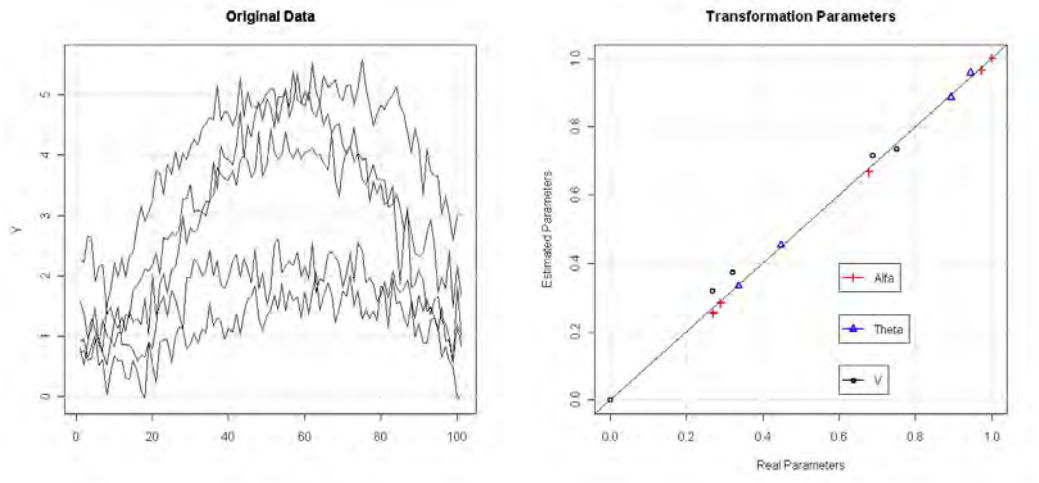
Simulated data are generated as follows:

$$Y_{ij} = \alpha_i^* \cdot f(t_j - \theta_i^*) + v_i^* + \varepsilon_{ij}, \quad 1 \leq i \leq n, 1 \leq j \leq J$$

with the following choice of parameters: $n = 5, \quad J = 101, \quad t_j = \frac{j}{J} 2\pi, \quad 1 \leq j \leq J$ are equally spaced points on $]0, 2\pi]$. Transformation parameters $(\boldsymbol{\theta^*}, \boldsymbol{\alpha^*}, \boldsymbol{v^*})$ are uniformly distributed on $[0, 1]$, where $\theta_1^* = 0, \quad v_1^* = 0, \quad \alpha_1^* = 1$; the noise $\varepsilon_{ji}, j = 1, .., J, i = 1, .., n$ are simulated with a Gaussian law with mean 0 and variance 0.5.

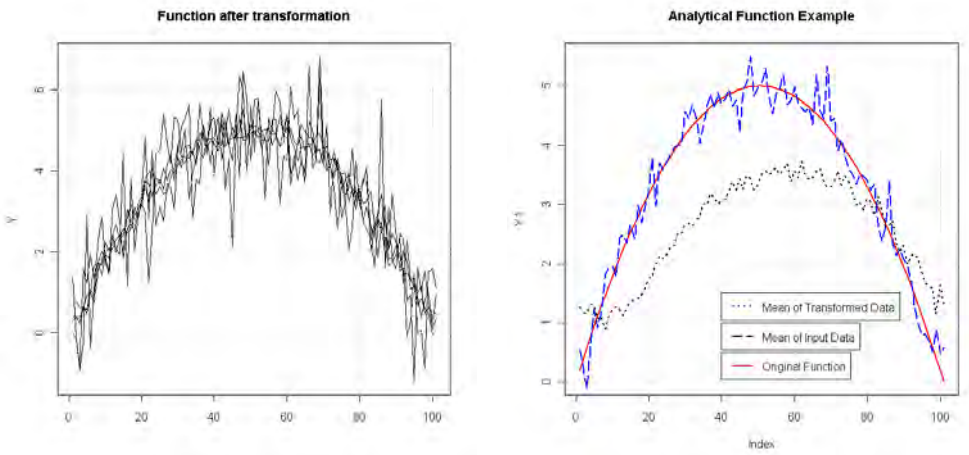
Results are displayed in Figure 3.5. The function f is plotted by a solid red line in Figure 3.5(d). Figure 3.5(a) shows the original simulated noisy data $Y_{i,j}$. The cross-sectional mean function of these data is presented in Figure 3.5(d) by black dotted line. Figure 3.5(b) plots estimated transformation parameters versus the originally simulated parameters. As it can be seen, the estimations are very close to the original parameters. The inverse transformation using the estimated parameters is displayed in Figure 3.5(c) and the mean function of restored curves is displayed in Figure 3.5(d) by blue dashed line. Figure 3.5(d) compares the cross-sectional mean of inversely transformed data and the cross-sectional mean of originally simulated data. Despite the noise, it is noticeable that the data after inverse transformation are much more closer to the original function f than the original cross-sectional mean function.

This analytical example shows that the method is effective in estimating the transformation parameters of the shape invariant model.



(a) Original data

(b) Calculated parameters



(c) Inverse transformation (d) Comparison

Figure 3.5: Analytical example.

3.3.2.3 Modified model with internal scaling

We can also consider a model including internal scaling of the variable. In this case, the observations data $\mathbf{Y}^n = \{Y_{i,j}, 1 \leq i \leq n, \quad 1 \leq j \leq J\}$ of a dynamic simulator can be represented as follows:

$$Y_{i,j} = f(a_i \cdot t_j), \quad 1 \leq i \leq n, \quad 1 \leq j \leq J \tag{3.8}$$

Figure 3.6 displays how this transformation changes the general pattern curve with an example on $\sin x$ function. The original function is depicted by a bold line. You can clearly see that with this transformation changes 2π-periodic function to a $a_i 2\pi$-periodic function. Here, we provide the estimation procedure of internal scaling parameters: $\boldsymbol{a} = (a_1, \ldots, a_n)$.

To estimate the parameters $\boldsymbol{a} = (a_1, \ldots, a_n)$ we propose to change the original variable so that: $t \to \exp u$. Then:

$$Y = f(a_i \cdot t) = f(\exp(\log(a_i \cdot t))) = f(\exp(\log a_i + \log t)) = g(u + \widehat{a}_i),$$

where $g(\cdot) = f(\exp(\cdot))$, So that, replacing the variables we reduce the problem of the internal scale parameters estimations to the previous model for the function g considering only the horizontal shifts parameters: $\widehat{\boldsymbol{a}} = (\widehat{a}_1, \ldots, \widehat{a}_n)$. However, such variable replacement should be done keeping the same boundaries $[1, T]$. The

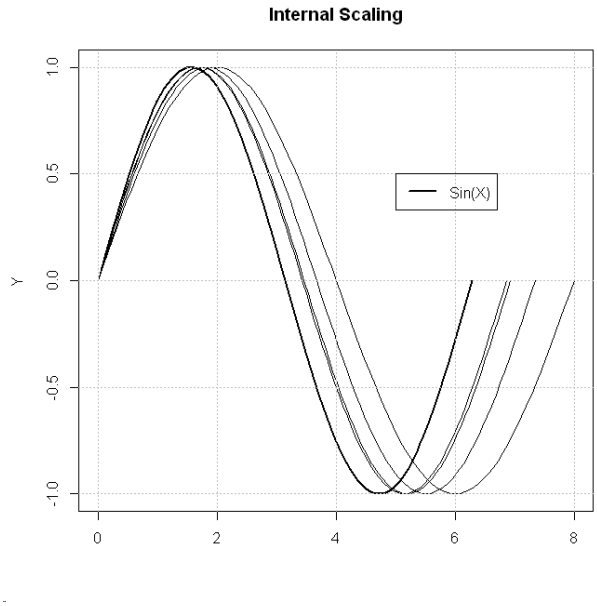


Figure 3.6: Internal scaling example.

variable t is replaced by u by the following formulation:

$$\begin{aligned} t &= \exp\left((T-1)\,\tfrac{u-1}{\log T-\log 1} + \log 1\right) = \exp\left((T-1)\,\tfrac{u-1}{\log T}\right) \\ &\text{or} \\ u &= \tfrac{(\log t-\log 1)(T-1)}{\log T-\log 1} + 1 = \tfrac{(\log t)(T-1)}{\log T} + 1 \end{aligned} \tag{3.9}$$

We should also keep in mind that the original function f in original coordinate space is 2π periodic. To estimate correctly the horizontal shifts parameters, we should keep new function g also 2π periodic.

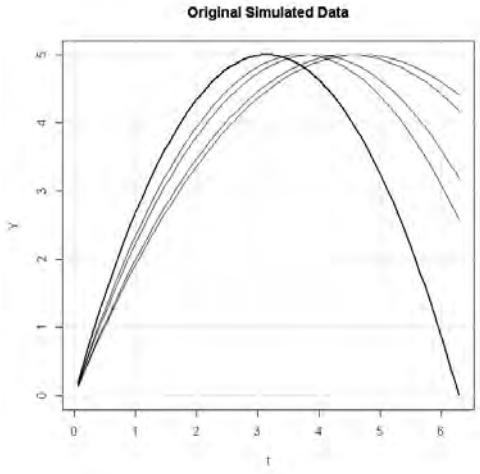
We illustrate the procedure of estimation of internal scale parameters with the same analytical function example as before:

$$f(x) = 20 \cdot (1 - x/(2\pi)) \cdot x/(2\pi)$$

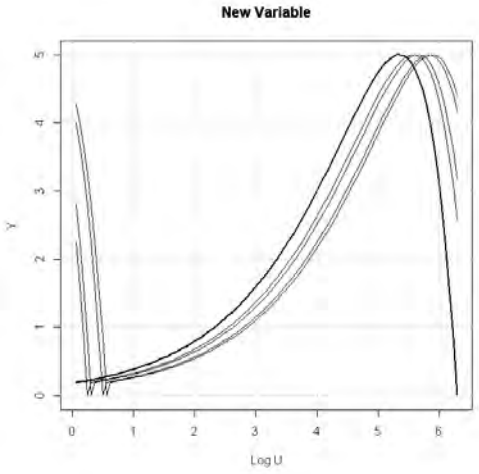
We consider the following choice of parameters: $J = 5$, $N = 101$, $t_i = \frac{i}{N}2\pi$, $1 \leq i \leq N$ are equally spaced points on $]0, 2\pi]$. Transformation parameters ($\boldsymbol{a^*}$) are uniformly distributed on $[0, 1]$, where $a_1^* = 1$. We do not consider the noise in this example to show the parameters estimation.

Results are displayed in Figure 3.7. The function f is plotted by a solid red line in Figure 3.7(a). Figure 3.7(a) shows the original simulated data $Y_{i,j}$. The data

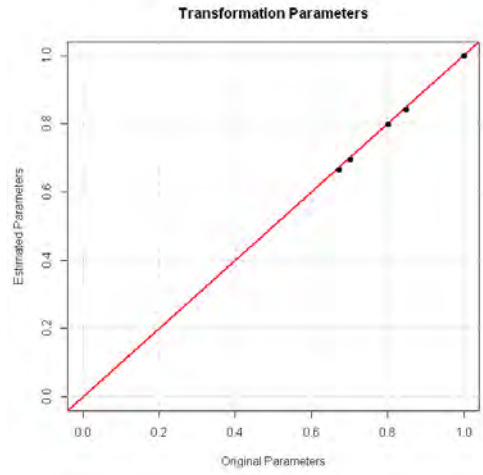
obtained after the variable replacement is presented in Figure 3.7(b). Horizontal shifts are now clearly observable with new variable. Figure 3.7(c) plots estimated internal scaling parameters versus the originally simulated parameters. As it can be seen, the estimations are very close to the original parameters. The inverse transformation using the estimated parameters is displayed in Figure 3.7(d). It is noticeable that the data after inverse transformation are much more closer to the original function f than the original cross-sectional mean function.



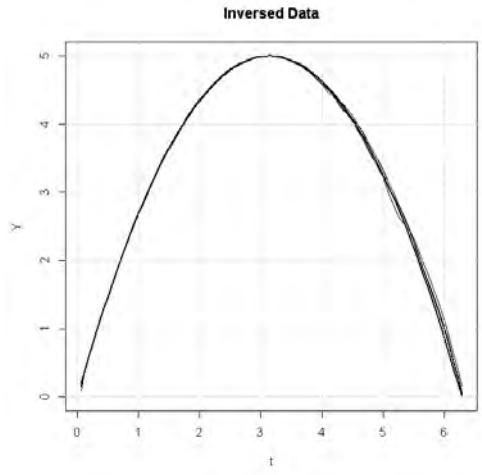
(a) Original noisy data



(b) Data with new variable



(c) Calculated parameters



(d) Inverse transformation

Figure 3.7: Analytical function example.

Applying the shape invariant model to the modelling of the output a dynamic simulator in reservoir engineering, we are mostly working with the model described in previous section. In the next section, we present the prediction algorithm based on the model (3.4). However, for some application the model (3.8) with internal scaling can be applicable. Then, the prediction algorithm is completely the same considering only the parameters of internal scaling.

In the next section, we explain how this model can be applied in reservoir engineering forecast problems.

3.3.3 Prediction algorithm

To apply the shape invariant model approach to model the dynamic simulator, firstly we have to modify the parameters estimation procedure for large number of curves. When we are working with uncertainty modelling, we always start from an experimental design $\mathbf{X}^n$ and a set of observation $\mathbf{Y}^n$. As we have mentioned, the number of design points depends on the number of inputs and on the complexity of the response. Hence, optimization problems could arise when we compute the contrast function with a large number of curves, i. e. large size of the experimental design. It can be time consuming and the results can be inaccurate. Therefore, we propose to adapt the Algorithm 3.7 to the Algorithm (3). The original observation data is split into blocks and the optimization procedure is then performed on each of the blocks. Following the identifiability condition, we are working on a compact set $\mathbf{A} = \{(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*, \boldsymbol{v}^*) \in [-\pi, \pi[^{3\times n} \colon \boldsymbol{\alpha}_1 = 1, \boldsymbol{\theta}_1 = 0, \boldsymbol{v}_1 = 0\}$. This condition should be satisfied on every block optimization by adding the first reference curve to the block.

Algorithm 3 Parameters estimation procedure for large n.

Input: Input set of curves from experimental design $\mathbf{Y}^n = \{Y_{ij}, i = 1, .., n; \quad j = 1, .., J\}$

Output: Transformation parameters estimation $(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*, \boldsymbol{v}^*)$

Split the observation data into N_b blocks of $(K + 1)$ curves

for $m = 1, .., N_b$ **do**

Define block curves $\mathbf{Y}^{K+1} = \{Y_1, Y_{(m-1)(K+1)...,Y_{mK}}\} = \{Y_{ij}, \quad i = 1, .., K+1, j = 1, .., J\}$

Perform Algorithm 3.3.2.1

Compute $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}}) = \arg \min_{\alpha,\theta,v \in \mathbf{A}} M_n(\alpha, \theta, v)$, where $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}}) \in \mathbb{R}^{3\times K}$

end for

Return: $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}}) \in \mathbb{R}^{3\times(n-1)}$

With this procedure, we can estimate the transformation parameters for an experimental design of any size. As soon as we have estimated the parameters for every curve from observation data set, we can formulate the prediction algorithm. Instead of reproducing the simulator output for a prediction point $\mathbf{x}_0$ at every time step, we model the whole output curve with apropriately estimated transformation parameter. This curve provides the approximation of the output for the selected input configuration $\mathbf{x}_0$ at each of considered time steps $\{t_j, \quad j = 1, .., J\}$. The transformation parameters for the input $\mathbf{x}_0$ are evaluated with the Gaussian process response surface modelling. The model is based on the experimental design and the set of evaluated transformation parameters calculated for the observation data curves. The prediction framework for an arbitrary input configuration $\mathbf{x}_0$ is presented by the following Algorithm 4.

With the proposed algorithm, the problem of response surface modelling for dynamic simulators is reduced to an optimization problem and GP modelling for the transformation parameters. However, it is worth to mention that before performing the algorithm it is important to analyze the curves behavior for the observation data set. Studying the curves characterization, probably we may fix for example vertical shifts $\boldsymbol{v}$ or horizontal shifts $\boldsymbol{\theta}$ at zero. Also, if the curves have significantly different behavior at different time series we can split the observation data in time as well in order to achieve higher prediction accuracy.

The next section presents an application with a dynamic reservoir simulator case.

Algorithm 4 Prediction algorithm for dynamic simulator.

Input: Dynamic simulator $Y = F(\mathbf{x}, t)$ with $t \in \{t_j, \quad j = 1, .., J\}$ and prediction point $\mathbf{x}_0$

Output: Prediction $Y^0 = F(\mathbf{x}_0, t_j)$ for all $j = 1, .., J$

Generate an experimental design $\mathbf{X}^n = (\mathbf{x}_1, .., \mathbf{x}_n)$ to span the space of interest

Evaluate $\mathbf{Y}^n = F(\mathbf{X}^n, t_j)$ at every time step $t_j, \quad 1 \leq j \leq J$

Generate a set of discrete curves $\{Y_{i,j}\}, \quad i = 1, .., n; j = 1, .., J$

Estimate the $(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{v}) \in (\mathbb{R}^n)^3$ with Algorithm 3

Construct new experimental design for the function of parameters: $(\mathbf{X}^n, \theta(\mathbf{X}^n)), \quad (\mathbf{X}^n, \alpha(\mathbf{X}^n))$ and $(\mathbf{X}^n, v(\mathbf{X}^n))$

Estimation of hyperparameters for GP models of transformation parameters

$\alpha(\mathbf{x}_0), \quad \theta(\mathbf{x}_0)$ and $v(\mathbf{x}_0)$ are approximated with corresponding GP models

Reproduce: $F(\mathbf{x}_0, t_j) = \boldsymbol{\alpha}(\mathbf{x}_0) f(t_j - \boldsymbol{\theta}(\mathbf{x}_0)) + \boldsymbol{v}(\mathbf{x}_0)$ for all $\{t_j, \quad j = 1, .., J\}$

Return: Discrete time series $Y^0 = F(\mathbf{x}_0, t)$ with $t \in \{t_j, \quad j = 1, .., J\}$

3.3.4 CO_2 storage case example

As we discussed, subsurface CO_2 storage is always associated with an excess reservoir pressure. The one of primary environmental risks is a pressure-driven leakage of CO_2 from the storage formation. In this section, we consider another CO_2 storage reservoir simulation model. We apply the prediction algorithm and compare it with a single step GP modelling as a reference case.

In order to assess the risk of CO_2 leakage through the cap rock we consider a synthetic reservoir model. The model is made up of three zones (3.8):

- a reservoir made of 10 layers
- a cap-rock made up of 1 layer
- a zone-to-surface composed of 9 layers

The XY size of the grid is set at 10 km total length. Each layer is 5m thick, including the cell above the cap-rock. The total number of cells is 13520 (26x26x20 model grid). The structure of the reservoir is reduced to its simplest expression. The zone above the cap-rock (up to the surface) is currently set to 1 layer. The salinity of the water is 35gm/l. The temperature of the reservoir is set to 60°C and the initial pressure is hydrostatic. The injection bottom rate is set to 10E+06 tons/year. The fracture pressure is estimated by geomechanical experts to 122 bars.

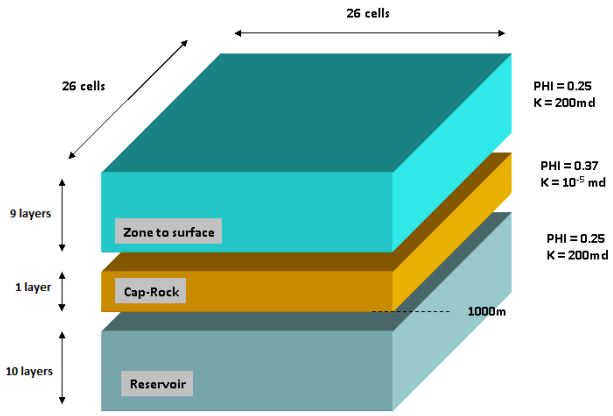


Figure 3.8: Reservoir model.

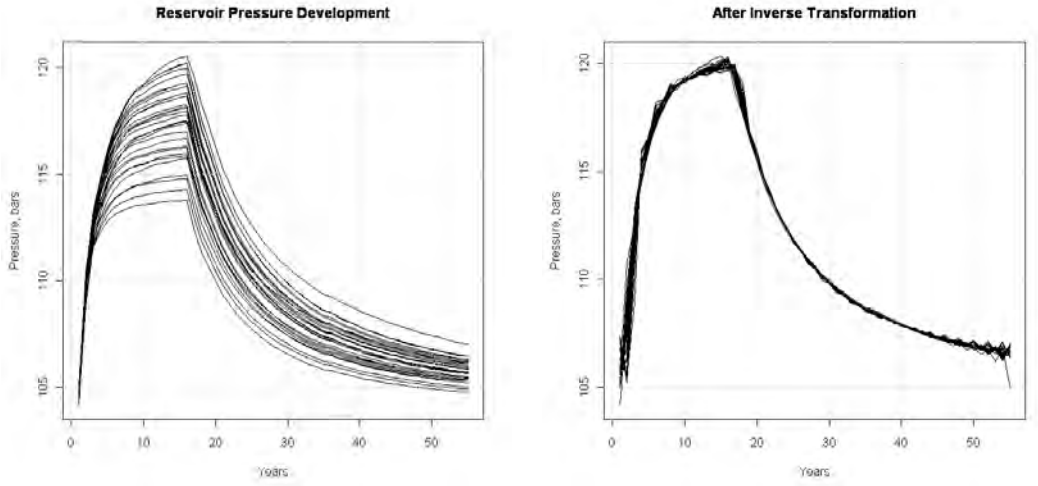
Exceeding this value of reservoir pressure can lead to a leakage. The simulation period is 55 years that include an injection period of 15 years followed by 40 years of storage. In this study we analyze the possibility of leakage through a cap rock. Therefore, we consider pressure in the storage reservoir as an objective function to be approximated.

The uncertain parameters selected for this study characterize the reservoir and the fluid properties. It implies different CO_2 flowing possibilities between the reservoir layers. The distribution low for the parameters is uniform. Table (3.5) represents the parameters description with their range of minimum and maximum values.

Name	Description	Min	Max
PORO	Reservoir Porosity	0.15	0.35
KSAND	Reservoir Permeability	10	300
KRSAND	Water relative permeability end-point	0.5	1.0

Table 3.5: Uncertain parameters.

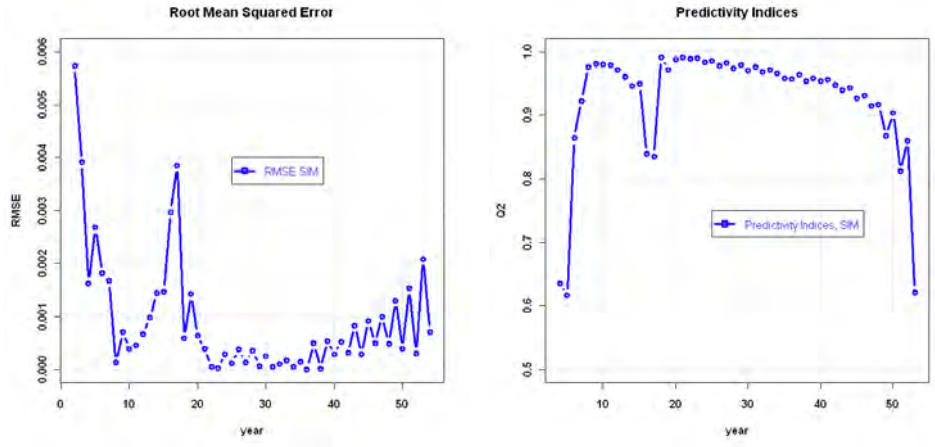
We start from the observation data $\mathbf{Y}^n$ - 30x55 matrix of simulator outputs. By means of Algorithm (3) we provide the transformations parameters estimations. Figure (3.9(a)) provides the original set of curves and Figure (3.9(b)) represents the same set after inverse transformation. The pattern curve is differentiable after the inverse transformation with the estimated parameters.



(a) Observation data (b) Restored data

Figure 3.9: Original observation data and data after inverse transformation.

As we are sure here that the model parameters are efficiently estimated, we can proceed with the next step of prediction algorithm (4). The next step is Gaussian process response surface modelling for the transformation parameters basing on the estimation set of parameters.

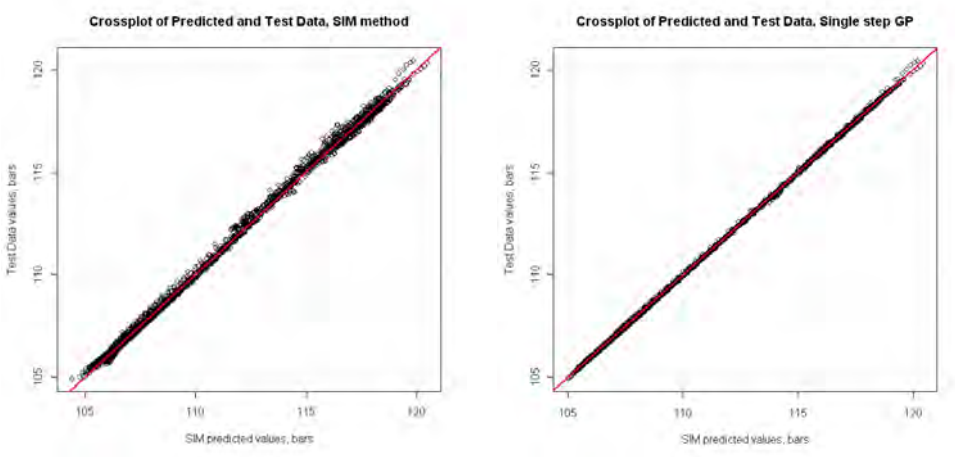


(a) RMSE (b) Predictivity indices

Figure 3.10: Predictivity Indices and RMSE.

In Figure (3.10) we display the model validation criteria: Root Mean Square Error (RMSE) and predictivity indices (Q2), calculated separately for every year. The

criteria were computed with help of additional confirmation test data. The low predictivity in the first and last years is caused by low variance of data in that period. In general, the method provides reliable level of predictivity. It is also reflected by crossplots of test and predicted data. Figure (3.11(a)) is based on SIM modelling approach and Figure (3.11(b)) corresponds to single step GP modelling. Both method provides a high level of approximation.



(a) SIM method (b) Single step GP

Figure 3.11: Crossplot comparison.

Table (3.6) compares the simulation CPU time for both methods. Single step GP model requires 5 times more CPU time, although SIM modelling method provides the same level of reliability.

	SIM method			Single step GP
	Optimization	Parameters modelling	Total	
CPU time	00:00:35	00:00:15	00:00:50	00:04:28

Table 3.6: CPU time comparison.

Worthwhile noting, that in this study we consider a simple model with only three uncertain parameters. So that, to estimate the function with GP model at every single step takes approximately 10 seconds. For more complex functions and more input uncertain parameters involved a single step model evaluation can take up to 10-20 minutes. So that, for the same simulation period of 55 years CPU time can increase to 10-20 hours. Whereas SIM approach does not depend on number of time

steps and the method always conclude only a single optimization problem and as maximum 3 GP models for the transformation parameters.

3.4 Conclusions

In this chapter, we focus on a problem of GP response surface modelling for dynamic simulators. The output of such simulator is time series at the time interval $t \in]0, T]$. We propose two different methods to address time series outputs.

The first method is an extension of the method introduced in Chapter 2 for handling discrete parameters. We consider time as an auxiliary variable and we build a new correlation function working in a new variable space $\Omega \times]0, T] \subset \mathbb{R}^d$. We have modified the algorithm in order to avoid possible numerical issues. The method was tested with a PUNQ-S3 reservoir model. The method is easy to implement. Moreover, by tuning parameters such as n_{train}, the method performance is comparable to a single step GP modelling approach. However, CPU time costs are much less. The method performance efficiency could be increased by splitting time domain into few subdomains, as we have shown in Section 3.2.2 for FOPT and FWC functions.

Here, we also propose a new method for a CO_2 injection well placement under uncertainty. We can consider injection well coordinates as controllable parameters. After that, with help of GP response surface model for a dynamic simulator we can sample all the uncertain parameters for every fixed well position. So, that the we reduce the problem to a unique GP response surface model. Whereas, standard single step approach assumes GP models for every possible well coordinates combination at every time step. The CO_2 storage reservoir example provided in Section 3.2.3 proves the reliability of the method.

Another approach to address time series outputs is a functional based approach. We adapted an efficient algorithm for estimation of transformation parameters of the shape invariant model with the specification for large sets of data curves. We suggested an application of this approach to model the time series outputs from a dynamic simulator. The proposed method reduces the problem of functional outputs modelling to one optimization problem and three GP response surface models for transformation parameters. We have tested the method with a CO_2 storage reservoir case. Numerical results show that the method provides satisfactory and comparable predictivity at lesser CPU time. The main advantage of this method is that its

performance does not depend on the number of the involved time-steps. It can be very advantageous when we are working with a model involving large number of times steps such as CO_2 storage when the reservoir model simulation period can include up to hundreds or thousands time steps. However, if the set of output curves have significantly different behavior, preliminary curves classification may be required.

To summarize, both of proposed methods proved their reliability with analytical and reservoir model examples. Including time as additional variable is easy to apply. The method performance is comparable to a single step approach especially in case of low number of time steps. Whereas, shape invariant model approach is very advantageous for large number of time steps. It is completely based on curves behavior analysis and does not depend on number of time steps.

Chapter 4

Reliability analysis

Contents

4.1 Introduction

CO_2 storage in deep geological formations is a safe and an effective measure of greenhouse gases mitigation. It is important to understand the reservoir performance, uncertainties and risks that are associated with the gas storage. Cap rock

failure assessment is one of the most important factors in preventing CO_2 leakage from the storage formation up to the surface. The cap rock integrity and the associated risks were recently studied by different authors in different aspects. Most of these studies are devoted to geomechanical analysis of the cap rock integrity and the storage formation. Rutqvist et al. [2007] analyze a maximum sustainable injection pressure using coupled analysis of fluid flow and geomechanical fault reactivation. Hawkes et al. [2005] review the geomechanical factors affecting hydraulic integrity of the storage reservoir. There are also studies involving risk and uncertainty analysis. Condor et al. [2011] provide a comprehensive summary and comparison analysis of existing risk assessment methodologies for the geological storage of carbon dioxide. Korre et al. [2007] present uncertainty modelling involved in CO_2 storage performance assessment in coalbed methane reservoirs. There is a risk of leakage not only through the cap rock fracturing but also through abandoned wells that were not properly sealed. Kopp et al. [2010] develop methods to quantify a risk of such leakage from subsurface reservoir. A response surface methodology was applied by Rohmer and Bouc [2010]. They use a linear regression model to estimate the effective stress state in the reservoir after CO_2 injection.

In this chapter, we propose to estimate the possible risk of leakage through a cap rock due to the excess the cap rock fracturing pressure. We propose an application of GP response surface modelling for uncertainty and risk analysis in CO_2 storage. By approximating the function of reservoir pressure, we can quantify the risk and estimate the reliability of the system. If we know the reservoir pressure, we can define a performance or limit state function $\mathbf{g}(\mathbf{x})$. This function determines if the system is in a failure state or in a safe state.

Underground reservoir modelling is always related to some uncertainties about input parameters. The reservoir pressure is approximated as a realization of a Gaussian Process. Therefore, we will define the risk of leakage due to overpressure as a failure probability. If $\mathrm{P}_{reservoir}(\mathbf{x})$ stands for a GP response surface model for the function of reservoir pressure and $\mathrm{P}_{fracture}$ is the fracture pressure, then we can define a limit-state function as:

$$g(\mathbf{x}) = \mathrm{P}_{fracture} - \mathrm{P}_{reservoir}(\mathbf{x}) \tag{4.1}$$

Then, the failure region is defined as: $\Omega_f = \{\mathbf{x} : g(\mathbf{x}) < 0\} \subset \Omega \subset \mathbb{R}^d$ and the

probability of failure is given by the following integral:

$$p_f = \mathrm{P}\{g(\mathbf{x}) < 0\} = \int_{\Omega_f} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{4.2}$$

where $f_{\mathbf{x}}(\mathbf{x})$ is the joint probability density function of input parameters $\mathbf{x} \in \Omega \subset \mathbb{R}^d$.

In the same way we can define the safe region $\Omega_s = \{\mathbf{x} : g(\mathbf{x}) > 0\} \subset \Omega \subset \mathbb{R}^d$ and the limit-state surface $\Omega_o = \{\mathbf{x} : g(\mathbf{x}) = 0\} \subset \Omega \subset \mathbb{R}^d$. By definition we have: $\Omega_f \cup \Omega_s \cup \Omega_o = \Omega \subset \mathbb{R}^d$.

The failure region and the failure probability are generally unknown. Identifying these quantities is one of the most important and challenging problems in reliability and risk analysis. In this work, we assume that all the input variables are independent and that their probability densities are known. The chapter is organized as follows. Firstly, we will review the reliability methods for estimating a failure probability. The methods can be grouped into approximation ones and simulation ones. We will pay more attention to subset simulation algorithm. Then, we will discuss the main drawbacks of GP response surface models for failure probability estimation. To improve the model reliability, we propose an adaptive refinement algorithm based on subset simulation. The method performance efficiency is studied on an analytical and a CO_2 storage reservoir case examples.

4.2 Reliability analysis methods

In this section, we provide a brief overview of the existing methods for estimation of a failure probability p_f. We divide the methods into the approximation ones and the simulation ones. The first ones consist in approximating the original limit-state function either by a Taylor expansion up to first or second order, or by a response surface model. Here, we will present only First and Second Order Reliability Methods (FORM and SORM) that are based on Taylor expansion approximations. The methods involving a response surface modelling will be considered separately in Section 4.3. The simulation methods is a class of methods that consist in numerous evaluations of the limit-state function. Then, basing on these simulations we can estimate directly the integral in (4.2). In addition, we can estimate the coefficient of variation and the variance of the computed estimation.

This section is organized as follows. First, we will introduce the approximation

methods such as FORM and SORM. Then, we will put in action the following simulation methods: Crude Monte-Carlo simulations, Direct Simulation, Importance Sampling and Subset Simulations. The present section is a review of the literature concerning structural reliability analysis [Bourinet, 2010; Du, 2005; Lemaire et al., 2009; Madsen et al., 2006].

4.2.1 Approximation methods

In this section, we work with the vector of independent input variables $\mathbf{x} = (x_1, \ldots, x_d) \in \Omega \subset \mathbb{R}^d$. As discussed previously, the failure probability p_f is given by the following integral:

$$p_f = \mathrm{P}\{g(\mathbf{x}) < 0\} = \int_{\Omega_f} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}, \tag{4.3}$$

where $g(\mathbf{x})$ is a performance function, $f_{\mathbf{x}}(\mathbf{x})$ is the known joint probability density function of $\mathbf{x}$ and $\Omega_f = \{\mathbf{x} \in \Omega : g(\mathbf{x}) < 0\}$ is the failure region.

The basic idea of approximation methods is to simplify the computation of the integral (4.3). There are two possible ways. It can be done either through the simplification of the integrand function, i.e. the joint density $f_{\mathbf{x}}(\mathbf{x})$, or the performance function $g(\mathbf{x})$.

A majority of reliability methods are designed for a standard normal space, or U-space. It facilitates the numerical computation by replacing the joint probability density $f_{\mathbf{x}}(\mathbf{x})$ with a standard normal density $\phi(\mathbf{x})$. Here, we will discuss the transformation of the input variables $\mathbf{x} = (x_1, \ldots, x_d)$ from its original physical space (or X-space) into $\mathbf{u} = (u_1, \ldots, u_d)$, where $\mathbf{u} \sim \mathcal{N}(0, \mathbf{I}_d)$ and $\mathcal{N}(0, \mathbf{I}_d)$ corresponds to the standard normal distribution. The failure probability estimation should be the same in the both spaces. Therefore:

$$p_f = \begin{cases} = \mathrm{P}\left(g(\mathbf{x}) \leq 0\right) = \mathrm{P}\left(G(\mathbf{u}) \leq 0\right) \\ = \int_{\Omega_f} f_{\mathbf{x}}(\mathbf{x}) \mathrm{d}\mathbf{x} = \int_{\Omega_u} \phi_d(\mathbf{u}) \mathrm{d}\mathbf{u} \end{cases} \tag{4.4}$$

The Rosenblat transformation can be applied [Rosenblatt, 1952]. If we denote F_{x_i} the cumulative distribution function for variable x_i in X-space and Φ is a standard normal cumulative distribution function, then the transformation can be expressed by:

$$\begin{aligned} \mathrm{F}_{x_i}(x_i) &= \Phi(u_i) \\ u_i &= \Phi^{-1}\left[\mathrm{F}_{x_i}(x_i)\right] \end{aligned} \tag{4.5}$$

For example, consider a Gaussian random variable $x \sim \mathcal{N}(\mu, \sigma^2)$ with cdf $\mathrm{F}_x = \Phi\left(\frac{t-\mu}{\sigma}\right)$. Therefore, (4.5) simplifies to:

$$u = \Phi^{-1}\left[\mathrm{F}_x(x)\right] = \Phi^{-1}\left[\Phi\left(\frac{x-\mu}{\sigma}\right)\right] = \frac{x-\mu}{\sigma},$$

Or

$$x = \mu + \sigma u.$$

When we apply the Rosenblatt transformation, the performance function also changes:

$$g(\mathbf{x}) = G(\mathbf{u})$$

Therefore, according to (4.4) the failure probability may be expressed as:

$$p_f = \mathrm{P}(G(\mathbf{u}) < 0) = \int_{\Omega_u} \phi_d(\mathbf{u}) d\mathbf{u},$$

where $\phi_d(\mathbf{u})$ is the joint probability of the vector $\mathbf{u} = (u_1, \ldots, u_d)$. Hence, the failure probability is given by the following integral:

$$p_f = \int_{G(\mathbf{u}<0)} \cdots \int \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u_i^2}{2}\right) du_1 \cdots du_d$$

As soon as we have simplified the integrand function, the next step is to simplify the integration boundary $G(\mathbf{u}) = 0$. First and Second Order Reliability Methods (FORM and SORM) consist in the approximation of the performance function G by a Taylor expansion. FORM uses a linear approximation, whereas SORM uses a second order Taylor expansion. In the next subsections these methods are described in details.

4.2.1.1 First order reliability method

In the First Order Reliability Method (FORM) we approximate the limit state function $G(\mathbf{u})$ by its first order Taylor expansion.

We linearize the function in the following way:

$$G(\mathbf{u}) \approx \mathrm{L}(\mathbf{u}) = G(\mathbf{u}^*) + \nabla G(\mathbf{u}^*)(\mathbf{u} - \mathbf{u}^*)^\top \tag{4.6}$$

$\mathrm{L}(\mathbf{u})$ stands for the linearized performance function, $\mathbf{u}^*$ is an approximation point and $\nabla G(\mathbf{u}^*)$ is the gradient of $G(\mathbf{u})$ evaluated at $\mathbf{u}^*$. The gradient is given by:

$$\nabla G(\mathbf{u}^*) = \left(\frac{\partial G(\mathbf{u})}{\partial u_1}, \frac{\partial G(\mathbf{u})}{\partial u_2}, \cdots, \frac{\partial G(\mathbf{u})}{\partial u_d} \right) \Bigg|_{u^*} \tag{4.7}$$

Recall that we are working in the U-space of standard normal random variables. The joint standard normal density $\phi_d(\mathbf{u})$ attains its maximum value at $\mathbf{u} = \mathbf{0}$ and then it decreases exponentially with $\|\mathbf{u}\|$ symmetrically. It is then a natural choice to approximate the performance function $G(\mathbf{u})$ at a point where $\phi_d(\mathbf{u})$ has the highest value. Consequently, the first step is to determine the Most Probable Point (MPP). It is the point on the limit-state surface $G(\mathbf{u}) = 0$ that maximizes the joint density $\phi_d(\mathbf{u})$. We can formulate an optimization problem for MPP:

$$\mathbf{u}^* = \arg\max \phi_d(\mathbf{u}) \;\; \text{subject to} \;\; G(\mathbf{u}) = 0. \tag{4.8}$$

We can rewrite the joint standard normal density definition as:

$$\phi_d(\mathbf{u}) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u_i^2}{2}\right) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \exp\left(-\frac{1}{2}\sum_{i=1}^{d} u_i^2\right)$$

Then, the maximization problem (4.8) is equivalent to the following minimization problem:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \sum_{i=1}^{d} u_i^2 = \arg\min_{\mathbf{u}} \|\mathbf{u}\|^2 \;\; \text{subject to} \;\; G(\mathbf{u}) = 0 \tag{4.9}$$

where $\|\cdot\|$ stands for the Euclidean norm.

The Most Probable Point (MPP) is the closest point from the limit-state surface to 0 in U-space. We can define the reliability index as the distance from the origin to the MPP $\mathbf{u}^*$: $\beta = \|\mathbf{u}^*\|$. A two dimension example is given in Figure 4.1.

By substituting the MPP and since $G(\mathbf{u}^*) = 0$, (4.6) reduces to

$$\mathrm{L}(\mathbf{u}) = \nabla G(\mathbf{u}^*)(\mathbf{u} - \mathbf{u}^*)^\top = \sum_{i=1}^{d} \frac{\partial G(\mathbf{u})}{\partial u_i}\Bigg|_{\mathbf{u}^*} (u_i - u_i^*) = \boldsymbol{\gamma}_0 + \sum_{i=1}^{d} \boldsymbol{\gamma}_i u_i \tag{4.10}$$

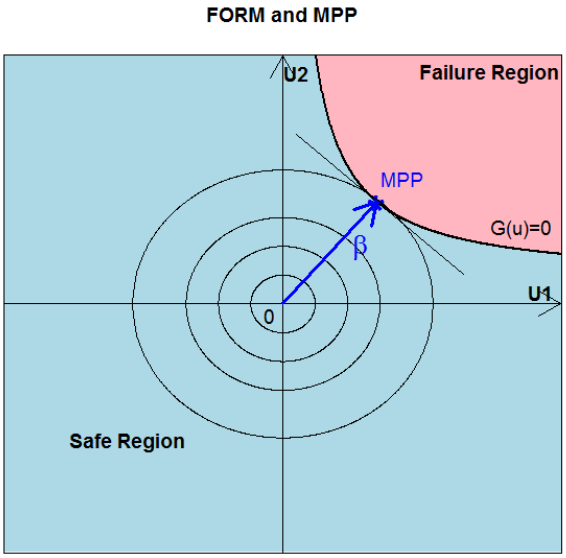


Figure 4.1: Most probable point and reliability index.

where the constants are defined as follows:

$$\gamma_0 = -\sum_{i=1}^{d} \left.\frac{\partial G(\mathbf{u})}{\partial u_i}\right|_{\mathbf{u}^*} u_i^* \quad \text{and} \quad \gamma_i = \left.\frac{\partial G(\mathbf{u})}{\partial u_i}\right|_{\mathbf{u}^*}.$$

Approximation (4.10) implies that $L(\mathbf{u})$ is a normal random variable with mean $\mu_L = \gamma_0$ and variance $\sigma_L = \sqrt{\sum_{i=1}^{d} \gamma_i^2}$. Therefore, the failure probability is given by:

$$\begin{aligned} p_f &\approx \mathrm{P}(L(\mathbf{u}) \leq 0) = \Phi(-\frac{\mu_L}{\sigma_L}) = \Phi\left(-\frac{\gamma_0}{\sqrt{\sum_{i=1}^{d} \gamma_i^2}}\right) = \\ &= \Phi\left(\frac{\sum_{i=1}^{d} \frac{\partial G(\mathbf{u})}{\partial u_i} \cdot u_i^*}{\sqrt{\sum_{i=1}^{d} \left(\left.\frac{\partial G(\mathbf{u})}{\partial u_i}\right|_{\mathbf{u}^*}\right)^2}}\right) = \Phi(\boldsymbol{\alpha}\mathbf{u}^{*\top}), \end{aligned}$$

where $\boldsymbol{\alpha} = \frac{\nabla G(\mathbf{u}^*)}{\|\nabla G(\mathbf{u}^*)\|}$ is the normalized gradient vector: $\|\boldsymbol{\alpha}\| = 1$. The vector $\boldsymbol{\alpha}$ is perpendicular to the limit-state surface $G(\mathbf{u}) = 0$ at the MPP. On the other hand, the MPP $\mathbf{u}^*$ is defined as the closest to the origin at the limit-state surface. According to (4.9), the MPP is the tangent point of the limit-state surface $G(\mathbf{u}) = 0$

and the circle with the radius of $\|\mathbf{u}^*\| = \beta$ (see Figure 4.1). Hence, the vector $\mathbf{u}^*$ is perpendicular to the surface $G(\mathbf{u}) = 0$ and it is oppositely directed to the gradient vector $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = -\frac{\mathbf{u}^*}{\boldsymbol{\beta}} \quad \text{or} \quad \mathbf{u}^* = -\boldsymbol{\beta}\boldsymbol{\alpha}.$$

Then, the failure probability is approximated by:

$$p_f \approx \mathrm{P}(L(\mathbf{u}) < 0) = \Phi(\boldsymbol{\alpha}\mathbf{u}^{*\top}) = \Phi(-\boldsymbol{\alpha}\boldsymbol{\alpha}^\top\boldsymbol{\beta}) = \Phi(-\boldsymbol{\beta}).$$

The MPP can be generally found by numerical iterative methods. The First Order Reliability Method is quick and it is widely used in the areas of structural reliability or reliability based design [Ditlevsen and Madsen, 1996; Madsen et al., 2006].

4.2.1.2 Second order reliability method

The Second Order Reliability Method (SORM) uses the second order Taylor expansion to approximate the performance function $G(\mathbf{u})$ at the Most Probable Point (MPP) $\mathbf{u}^*$. The expansion is as follows:

$$G(\mathbf{u}) \approx \mathrm{G}_2(\mathbf{u}) = G(\mathbf{u}^*) + \nabla G(\mathbf{u}^*)(\mathbf{u} - \mathbf{u}^*)^\top + \frac{1}{2}(\mathbf{u} - \mathbf{u}^*)\mathbf{H}(\mathbf{u}^*)(\mathbf{u} - \mathbf{u}^*)^\top \quad (4.11)$$

$\mathrm{G}_2(\mathbf{u})$ stands for approximation of the performance function, $\mathbf{u}^*$ is the MPP ($G(\mathbf{u}^*) = 0$), $\nabla G(\mathbf{u}^*)$ is the gradient of $G(\mathbf{u})$ evaluated at $\mathbf{u}^*$ and $\mathbf{H}(\mathbf{u}^*)$ is the Hessian evaluated at $\mathbf{u}^*$. The Hessian Matrix is the matrix of second derivatives:

$$\mathbf{H}(\mathbf{u}^*) = \begin{bmatrix} \frac{\partial^2 G}{\partial^2 u_1} & \frac{\partial^2 G}{\partial u_1 \partial u_2} \cdots & \frac{\partial^2 G}{\partial u_1 \partial u_d} \\ \frac{\partial^2 G}{\partial u_2 \partial u_1} & \frac{\partial^2 G}{\partial^2 u_2} \cdots & \frac{\partial^2 G}{\partial u_2 \partial u_d} \\ \cdots & \cdots\cdots & \cdots \\ \frac{\partial^2 G}{\partial u_d \partial u_1} & \frac{\partial^2 G}{\partial u_d \partial u_2} \cdots & \frac{\partial^2 G}{\partial^2 u_d} \end{bmatrix} (\mathbf{u}^*).$$

By performing different linear transformations, the approximation $\mathrm{G}_2(\mathbf{u})$ can be simplified and the failure probability estimation is given by [Du, 2005]:

$$p_f = \mathrm{P}(g(\mathbf{u}) < 0) = \Phi\left(\prod_{i=1}^{d} \sqrt{1 + \boldsymbol{\beta}\boldsymbol{\kappa}_i}\right), \quad (4.12)$$

where $\boldsymbol{\beta}$ stands for reliability index and $\boldsymbol{\kappa}_i$ denotes the ith main curvature of the performance function $G(\mathbf{u})$.

SORM is applied when the MPP is available from FORM analysis. FORM is easy to apply because only the first derivatives are involved. These quantities could be quickly numerically computed. Application of SORM implies second order derivatives and it requires more function calls. However, it provides more accurate approximations.

4.2.2 Simulation methods

Simulation methods are based on multiple performance function evaluations for a random sample from the original joint probability $f_{\mathbf{x}}(\mathbf{x})$. These methods are more expensive from a CPU time point of view. However, simulation methods are widely used in reliability engineering disciplines. Recently, multiple methods enhancements were proposed in order to decrease the number of function calls and CPU time respectively. In this section, we first consider crude Monte Carlo estimations and some basic modifications such as importance sampling, directional sampling and subset simulations.

4.2.2.1 Crude Monte Carlo sampling

This method is the simplest simulation method.

First, let us reformulate the definition of the failure probability as an expectation of a failure indicator function. Here, we can work both in the original X-space of input variables or in the standard normal U-space. By definition:

$$p_f = \mathrm{P}(g(\mathbf{x}) < 0) = \int_{\Omega_f} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = \int_{\Omega} \mathbf{I}(\mathbf{x}) f_{\mathbf{x}} d\mathbf{x} = \mathbb{E}_{f_{\mathbf{x}}}[\mathbf{I}(\mathbf{x})], \tag{4.13}$$

where the failure indicator function is defined as:

$$\mathbf{I}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \Omega_f, \\ 0, & \text{if } \mathbf{x} \in \Omega_s \end{cases}$$

Suppose, we have $\mathbf{X}^N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ a set of N i.i.d. samples randomly drawn from the joint density $f_{\mathbf{x}}(\mathbf{x})$. Then, the Crude Monte Carlo (CMC) estimation is as

follows:

$$\widehat{p}_{f_{CMC}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{I}(\mathbf{x}_i) = \frac{N_f}{N} \tag{4.14}$$

where N_f is the number of samples $\mathbf{x}_i$ that fall inside the failure region.

Figure 4.2 displays an example of crude Monte Carlo sampling in U-space. The red points fall inside the failure region. As a sum of independent random variables

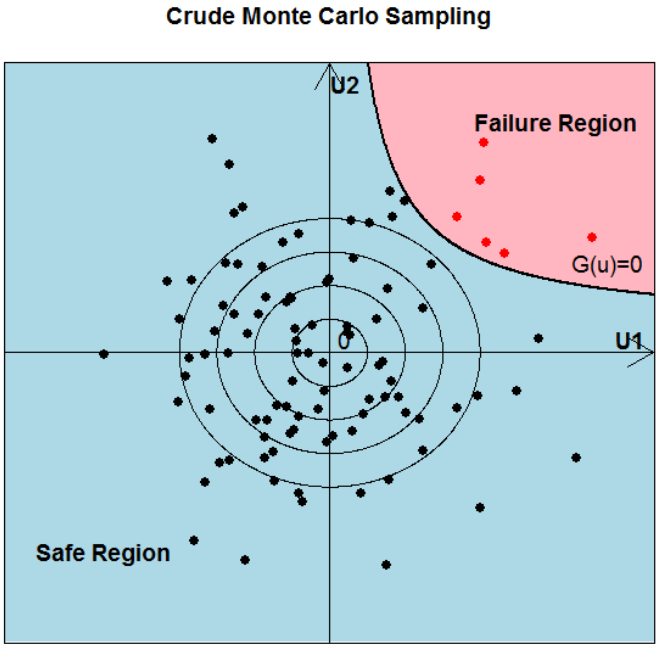


Figure 4.2: Crude Monte Carlo sampling.

with the same distribution, $\widehat{p}_{f_{CMC}}$ is asymptotically Gaussian. Moreover, it is an unbiased estimator of p_f. Indeed, we have:

$$\mathbb{E}[\widehat{p}_{f_{CMC}}] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[\mathbf{I}(\mathbf{x}_i)]] \overset{i.i.d.}{=} \frac{1}{N}\left(N\mathbb{E}[\mathbf{I}(\mathbf{x})]\right) = \mathbb{E}[\mathbf{I}(\mathbf{x})] = 0\times(1-p_f)+1\times p_f = p_f$$

The variance VAR$[\widehat{p}_{fCMC}]$ of the estimator is given by:

$$\text{VAR}[\widehat{p}_{f_{CMC}}] = \frac{1}{N}p_f(1-p_f).$$

To evaluate the accuracy of the CMC estimator, the coefficient of variation (or relative standard deviation) can be used. It is defined as:

$$\boldsymbol{\delta}_{\widehat{p}_{fCMC}} = \frac{\sqrt{\mathrm{VAR}[\widehat{p}_{fCMC}]}}{\mathbb{E}[\widehat{p}_{fCMC}]} = \sqrt{\frac{1-p_f}{Np_f}}.$$

For $p_f \ll 1$, the following approximation holds:

$$\boldsymbol{\delta}_{\widehat{p}_{f_{CMC}}} \approx \frac{1}{\sqrt{Np_f}}.$$

Therefore, for a target accuracy of 10%, i.e. $\boldsymbol{\delta}_{\widehat{p}_{f_{CMC}}} = 0.1$, the size of the sample should be $N \approx \frac{100}{p_f}$. Suppose that $p_f \approx 10^{-5}$, then the required number of function evaluations is $N \approx 10^7$. This is not always affordable.

Despite the fact that this method is computationally expensive, it remains one of the most widely used methods.

4.2.2.2 Importance sampling

Importance sampling is a well established technique for variance reduction in Monte Carlo simulation. It is described by numerous authors [Melchers, 1989; Ripley, 1987; Rubinstein and Kroese, 2008; Shinozuka, 1983]. The idea is to sample more efficiently according to a "good" distribution. This distribution is designed for the failure region Ω_f. Recall that the failure probability is defined as

$$p_f = \int_{\Omega} \mathbf{I}(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}$$

The method assumes a change of the probability density $f_{\mathbf{x}}(\mathbf{x})$. For example, let $h(\mathbf{x})$ be a density vanishing away of the failure region Ω_f. It means that:

$$h(\mathbf{x}) = 0 \quad \Rightarrow \quad \mathbf{I}(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}) = 0,$$

$$\mathbf{I}(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}) \neq 0 \quad \Rightarrow \quad h(\mathbf{x}) \neq 0.$$

The function $h(\mathbf{x})$ is called a proposal or importance sampling probability density function. If we multiply the integrand by $1 = \frac{h(\mathbf{x})}{h(\mathbf{x})}$, we get:

$$p_f = \int_\Omega \mathbf{I}(\mathbf{x}) \frac{f_\mathbf{x}(\mathbf{x})}{h(\mathbf{x})} h(\mathbf{x}) d\mathbf{x} = \mathbb{E}_h \left[\mathbf{I}(\mathbf{x}) \frac{f_\mathbf{x}(\mathbf{x})}{h(\mathbf{x})} \right].$$

The estimation of the failure probability is equivalent to the estimation of the above expectation. This can be done by Monte Carlo sampling according to the new density $h(\mathbf{x})$. Let $\mathbf{X}^N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a set of N i.i.d. samples of density $h(\mathbf{x})$. Then:

$$\widehat{p}_{f_{IS}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{I}(\mathbf{x}) \frac{f_\mathbf{x}(\mathbf{x}_i)}{h(\mathbf{x}_i)}. \tag{4.15}$$

is an unbiased estimation of $\mathbb{E}_h [\widehat{p}_{fIS}]$. The variance of this estimation is:

$$\text{VAR} [\widehat{p}_{f_{IS}}] = \frac{1}{N} \left(\mathbb{E}_h \left[\left(\mathbf{I}(\mathbf{x}) \frac{f_\mathbf{x}(\mathbf{x})}{h(\mathbf{x})} \right)^2 \right] - p_f^2 \right).$$

The optimal choice of the proposal density $h(\mathbf{x})$ is the one that minimizes the variance of the estimator $\widehat{p}_{f_{IS}}$. It is well known (see Rubinstein and Kroese [2008]), that the optimal proposal density is the following density:

$$h^*(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) f_\mathbf{x}(\mathbf{x})}{p_f}.$$

Unfortunately, the optimal density depends on the unknown value of p_f. However, it provides an idea of the best sampling density. In the field of structural reliability, different proposal densities were studied. For example, Shinozuka [1983] proposed a uniform probability density in a hypercube centered at the MPP. Later, Melchers [1989] analyzed the standard Gaussian probability density centered at the MPP. A sample drawn with this method is presented in Figure 4.3. Different choices of $h(\mathbf{x})$ have their advantages and limitations. The choice of proposal density can even be changed during simulations depending on the reliability problem. There are also different variants of importance sampling: bridge sampling [Meng and Wong, 1996], linked importance sampling [Neal, 2005], particle filtering [Chopin, 2002] etc.

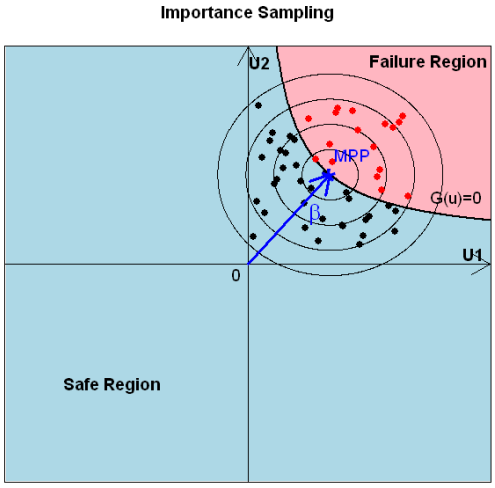


Figure 4.3: Importance sampling example.

4.2.2.3 Directional sampling

The directional sampling method uses the polar coordinate system. In these coordinates, the main variables are the radius $\mathbf{R}$ and the direction vector $\mathbf{A}$. The method was introduced by Bjerager [1988] and Ditlevsen et al. [1988]. It is a variance reduction method. Hence, the estimation of the failure probability for a given accuracy can be achieved with a smaller number of simulations.

Assume that we work in the standard normal U-space. For independent but non-Gaussian variable $\mathbf{x}$, the Rosenblatt transformation can be applied (Rosenblatt [1952], See Section 4.2.1). We represent a standard Gaussian variable $\mathbf{u}$ as: $\mathbf{u} = r\mathbf{a}$, where r^2 is a χ^2 random variable with d degree of freedom and $\mathbf{a}$ is a unit vector ($\|\mathbf{a}\| = 1$) with a uniform distribution over the d-dimensional unit hypersphere $\mathbf{S}_1^d$. Then the failure probability can be rewritten as:

$$p_f = \int_{\mathbf{S}_1^d} \mathrm{P}\left[G(r\mathbf{A}) \leq 0 | \mathbf{A} = \mathbf{a}\right] d\mathbf{a} \tag{4.16}$$

where $d(\mathbf{a})$ is a normalized Lesbegue measure on the unit hypersphere $\mathbf{S}_1^d$.

Let, $r(\mathbf{a})$ be the radius of the point on the limit-state surface $G(\mathbf{u}) = 0$ which is the closest to the origin in direction $\mathbf{a}$. We can rewrite then a part of the integrand

function as:

$$\mathrm{P}\left[G(r\mathbf{A}) \leq 0 |\, \mathbf{A} = \mathbf{a}\right] = \mathrm{P}(r > r(\mathbf{a}) | \mathbf{A} = \mathbf{a}) = \mathrm{P}(R^2 > r(\mathbf{a})^2 | \mathbf{A} = \mathbf{a}) = \\ = 1 - \mathrm{P}(R^2 \leq r(\mathbf{a})^2 | \mathbf{A} = \mathbf{a}) = 1 - \boldsymbol{\chi}_d^2\left(r(\mathbf{a})^2\right)$$

Then the failure probability may be expressed as:

$$p_f = \int_{\mathbf{S}_1^d} \left[1 - \boldsymbol{\chi}_d^2\left(r(\mathbf{a})^2\right)\right] d\mathbf{a}.$$

Now, we can provide an estimation of failure probability. Suppose, $\mathbf{A}^N = \{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$ is a set of N vectors uniformly distributed over the unit hypersphere. Then, p_f can be estimated by:

$$\widehat{p}_{fDS} = \frac{1}{N} \sum_{i=1}^{N} \left[1 - \boldsymbol{\chi}_d^2\left(r(\mathbf{a_i})^2\right)\right]$$

where $r(\mathbf{a}_i)$ is the distance between the origin and the point intersecting the limit-state surface $G(\mathbf{u}) = 0$ in the direction $\mathbf{a}_i$. In practice, these intersections are estimated iteratively. Once the closest points are determined, the estimation of the failure probability p_f is straightforward. However, the algorithm can fail to find the closest point on $G(\mathbf{u}) = 0$. Figure 4.4 displays an example of directional simulations explaining graphically the choice of the direction vectors a_i.

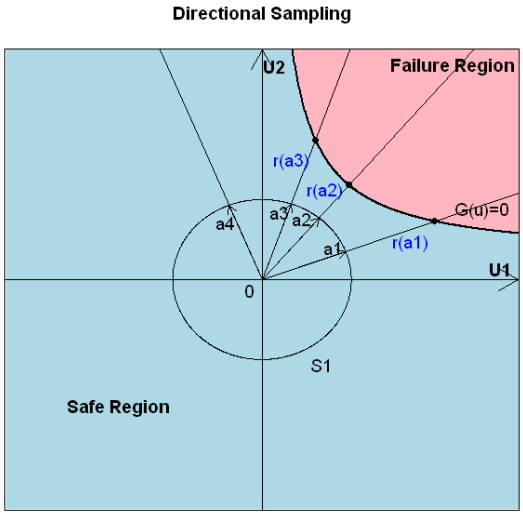


Figure 4.4: Directional sampling example.

Moarefzadeh and Melchers [1999] proposed a combination of directional simulation and importance sampling for application in a physical X-space .

4.2.2.4 Subset simulation

Subset simulation for estimation of the failure probability was introduced by Au and Beck [2001]. In particular, Au and Beck [2003] applied their method to the analysis of seismic dynamic risks. It has also been used in the context of stochastic reliability optimization in Taflanidis and Beck [2008] and Taflanidis and Beck [2009]. Later, Zuev et al. [2011] presented a Bayesian approach, where their goal is to estimate the posterior distribution of the failure probability estimator.

The basic idea of subset simulation is to represent the very low failure probability p_f as a product of larger probabilities that can be computed more easily:

$$p_f = \prod_{i=1}^{m} p_i \tag{4.17}$$

For such representation, we will consider a decreasing sequence of nested subsets in the parameters space. This sequence commences from the entire space Ω and it converges to the failure region Ω_f.

$$\Omega = F_0 \supset F_1 \supset \cdots \supset F_m = \Omega_f$$

Subsets $F_1, \ldots, F_m$ are called intermediate failure events. Since $\bigcap_{i=1}^{m} F_i = \Omega_f$, the failure probability $p_f = \mathrm{P}(\Omega_f)$ can be rewritten as a product of conditional probabilities. By definition of conditional probability we have [Au and Beck, 2001]:

$$p_f = \mathrm{P}(\Omega_f) = \mathrm{P}(\bigcap_{i=1}^{m} F_i) = \mathrm{P}(F_1) \prod_{i=2}^{m} \mathrm{P}(F_i | F_{i-1}) = \prod_{i=1}^{m} p_i$$

Therefore, the problem of failure probability estimation is reduced to the estimation of the intermediate failure probabilities p_i. By a proper choice of the intermediate failure events F_i, the corresponding intermediate failure probabilities can be sufficiently large to be efficiently estimated by Crude Monte Carlo (CMC) sampling. For example, suppose $p_i \approx 0.1, \quad i = 1, \ldots, 4$ then $p_f \approx 10^{-4}$ which is rather small to be estimated precisely with CMC simulation. However, the intermediate probabilities $p_i \approx 0.1$ can be efficiently estimated at low CPU time cost.

The original problem of p_f estimation is then replaced by a sequence of MC simulations of more frequent intermediate events in the conditional probability spaces $\mathrm{P}(F_i|F_{i-1}),\quad i=2,\ldots,m$ with the conditional densities $q(\cdot|F_{i-1})$. The conditional probabilities $p_i=\mathrm{P}(F_i|F_{i-1})$ are estimated sequentially.

$$p_i \approx \widehat{p}_i^{MC} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{I}_{F_i}\left(\mathbf{x}_k^i\right), \quad \text{where} \quad \mathbf{x}_k^i \overset{i.i.d.}{\sim} q(\cdot|F_{i-1}) \tag{4.18}$$

Working with the conditional densities $q(\cdot|F_{i-1})$ means that we need to generate a sample $\mathbf{X}^i=\{\mathbf{x}_1^i,\ldots,\mathbf{x}_N^i\}$ which follows the joint density $f_{\mathbf{x}}(\mathbf{x})$ and which lies in the intermediate failure domain F_{i-1}. Then, among this sample, we estimate p_i by evaluating the number of points that lie into F_i.

In this sequence, $p_1=\mathrm{P}(F_1|F_0)=\mathrm{P}(F_1|\Omega)=\mathrm{P}(F_1)$ can be easily estimated by a Monte Carlo according to a known joint density $q=q(\cdot|\Omega)=f_{\mathbf{x}}(\mathbf{x})$:

$$\mathrm{P}_1 \approx \widehat{P}(F_1) = \frac{1}{N}\sum_{k=1}^{N} \mathbf{I}_{F_1}(\mathbf{x}_k).$$

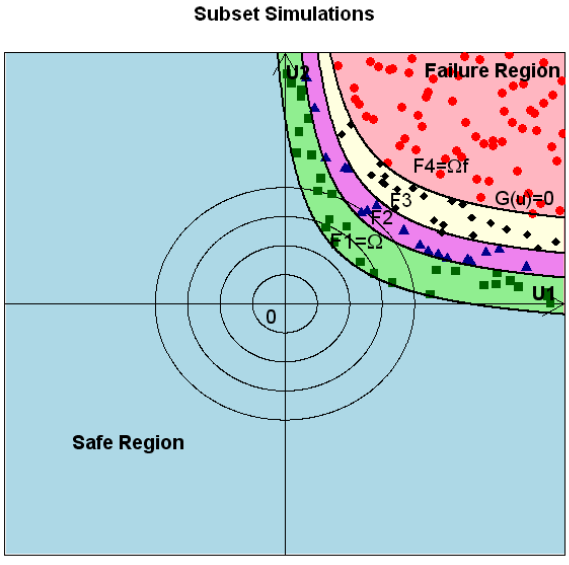


Figure 4.5: Subset simulation example.

To estimate the probabilities p_i for $i \geq 2$, we need to sample from conditional density functions $q(\cdot|F_{i-1})$. Markov Chain Monte Carlo (MCMC) is a reasonable alternative. MCMC is a Markov chain technique to sample from an arbitrary multi-dimensional probability distribution. Au and Beck [2001] introduced a modified Metropolis-Hastings algorithm for efficient sampling according to the multi-dimensional density $q(\cdot|F_{i-1})$. Details can be found in Au and Beck [2001, 2003]. Briefly, for every vector coordinate we separately generate a candidate state. Then it is accepted if a vector of the candidates lies in F_{i-1}. As soon as we have a sample of size N, we can apply (4.18) for estimation of $p_i = \mathrm{P}(F_i|F_{i-1})$.

Figure 4.5 presents an example with $m = 4$: $F_1 \supset F_2 \supset F_3 \supset F_4 = \Omega_f$. It illustrates how we can use a sequence of intermediate failure events to approach at every level the failure region Ω_f. Intermediate failure regions of different levels are displayed by different colors.

We will study in detail both the modified Metropolis-Hastings algorithm and the practical implementation of the algorithm of subset simulation in Section 4.2.4. We will also provide an estimation for the coefficient of variation δ of the computed estimator.

4.2.3 Analytical example

In this subsection, we will compare the estimations of the failure probability provided by different methods.

We will consider the following performance function depending on three variables:

$$f(\mathbf{x}) = 5 - 0.2x_1 - 0.7x_2 - x_3$$

where all $x_i \sim \mathcal{N}(0, 1)$ are independent standard normal variables with. As a sum of independent standard normal variables, function $f(\mathbf{x})$ is also normal variable with mean $\mu = 5$ and variance $\sigma^2 = 0.2^2 + 0.7^2 + 1 = 1.53$. Therefore, we can estimate precisely the failure probability:

$$\mathrm{P}(f(\mathbf{x}) \leq 0) = \Phi(-\mu/\sigma) = 2.64693 \times 10^{-5}.$$

We will compare estimations provided by the presented methods: FORM, SORM, Importance Sampling (IS), Directional Sampling (DS) and Subset simulation (SS). For IS method as the proposal density $h(\mathbf{x})$ we use the 3-dimensional standard

normal density centered at the MPP. Table 4.1 summarizes the estimations, number of function calls and CPU time required. The methods were performed using the FERUM software [Bourinet, 2010].

Method	$\widehat{p}_f$	CPU time	Number of function calls
FORM	2.6469e-5	0.0798 sec	14
SORM	2.6469e-5	0.1054 sec	23
IS	3.2000e-5	1.5871 sec	1e6
DS	2.6469e-5	12.9425 sec	4.55e6
SS	2.6710e-5	0.5055 sec	49316

Table 4.1: Reliability methods comparison.

It can be seen that for this function FORM/SORM provides the most accurate estimation with the less number of function calls. Of course, that is easily explained by the linearity of the performance function. FORM approximates the limit state surface by a hyperplane and, for the linear case, this hyperplane coincides with the original limit state surface. However, for more complicated functions the approximation is not straightforward. The error of the failure probability estimation for FORM and SORM increases with the complexity of the function. Regarding the simulation methods, the most accurate estimate is given by directional simulation. However, subset simulation provides reliable estimation of failure probability p_f with less number of function calls. Furthermore, importance sampling overestimates p_f.

The advantage of FORM/SORM method is that it needs a small number of function calls. However, the errors of estimation can be significantly larger for more complicated functions. In addition, these methods do not provide statistical properties of the computed estimator, such as the variance and the coefficient of variation. Among simulation methods, subset simulation yields a reliable estimation of the failure probability p_f at less cost. Though, it still requires a high number of function evaluations for a reliable estimation. This is not always possible with an expensive reservoir simulator. For this reason, in Section 4.3 we introduce a Gaussian process response surface modelling in estimation of the failure probability p_f. We will discuss the method to improve the reliability of the provided estimation.

In the next section, we will discuss in details the Subset simulation algorithm and its practical implementation.

4.2.4 Subset simulation algorithm

As already discussed, the objective of subset simulation is to represent the failure probability p_f as a product of larger probabilities corresponding to intermediate failure events: $\Omega = F_0 \supset F_1 \supset \cdots \supset F_m = \Omega_f$. The failure domain $\Omega_f = \{\mathbf{x} : g(\mathbf{x}) < 0\}$ is then equal to the intersection $\Omega_f = \bigcap_{i=1}^{m} F_i$. The failure probability p_f is evaluated as a product of conditional probabilities:

$$p_f = \prod_{i=1}^{m} \mathrm{P}(F_i|F_{i-1}) = \prod_{i=1}^{m} p_i, \tag{4.19}$$

where every $p_i = \mathrm{P}(F_i|F_{i-1})$ is evaluated sequentially by sampling from the conditional density $q(\cdot|F_{i-1})$. Recall that the conditional density $q(\cdot|F_{i-1})$ is designed to generate a sample that follows the original joint density $f_{\mathbf{x}}(\mathbf{x})$ and lies in the domain F_{i-1}. Thus, we replace the problem of estimating very low probability p_f by estimating higher conditional probabilities $p_i = \mathrm{P}(F_i|F_{i-1}), \quad i = 1, \ldots, m$.

To estimate p_1 crude Monte Carlo sampling can be applied since $q(\cdot|F_0) = q(\cdot|\Omega) = f_{\mathbf{x}}(\mathbf{x})$. To estimate $p_i, \quad i \geq 2$ and to sample from the conditional densities, Markov Chain Monte Carlo (MCMC) sampling is used. It is an effective technique to sample from an arbitrary distribution. In particular, the Metropolis-Hastings algorithm was originally introduced by Metropolis et al. [1953] and then modified by Hastings [1970]. By a proper application of this algorithm, the distribution of the generated sample converges to the given conditional density $q(\cdot|F_{i-1})$ if the length of the sample is sufficiently large [Au and Beck, 2003].

Suppose, that we want to sample a d-dimensional vector $\mathbf{x} = (x_1, \ldots, x_d)$ according to the conditional density $q(\cdot|F_{i-1})$. We suppose that all the vector components are independent. We can rewrite:

$$q(\mathbf{x}|F_{i-1}) = \frac{f_{\mathbf{x}}(\mathbf{x})\mathbf{I}_{F_{i-1}}(\mathbf{x})}{\mathrm{P}(F_{i-1})} = \frac{\left[\prod_{k=1}^{d} f_k(x_k)\right] \mathbf{I}_{F_{i-1}}(\mathbf{x})}{\mathrm{P}(F_{i-1})}.$$

In the case of a high dimensional vector $\mathbf{x}$ with many independent components, the original Metropolis-Hastings algorithm produces a Markov chain with highly correlated states [Au and Beck, 2001; Zuev et al., 2011]. For this reason, Au and Beck [2001] proposed a modified Metropolis algorithm that consists in generating a MCMC sample separately component by component. Sampling by components does

not change the stationary distribution of the Markov chain. It was shown that if the current sample $\mathbf{x}^j \sim q(\cdot|F_{i-1})$, then the next sample $\mathbf{x}^{j+1}$ has the same distribution (see Au and Beck [2001]).

Now, we will briefly explain the modified Metropolis algorithm. First, we have to choose a proposal density. Let $r_i(\xi_i|\alpha_i)$ be a univariable symmetric density for variable x_i. The symmetry property means that $r_i(\xi_i|\alpha_i) = r_i(\alpha_i|\xi_i)$. This density is centered at α_i. We can use the same proposal density for all the coordinates $r_i(x_i) = r(x_i),\ \ i = 1, \ldots, d$. In order to estimate p_i we need to generate a sample of size N from the conditional probability $q(\cdot|F_{i-1})$. Let $\mathbf{x}_1 \in F_{i-1}$ be the initial state of a target Markov Chain. As a proposal density, we suggest a uniform univariable density centered at $\mathbf{x}_1$ on the interval of selected width. Then, if $f_{\mathbf{x}}(\mathbf{x}) = \prod_{k=1}^{d} f_k(x_k)$ is the original joint pdf, the modified Metropolis-Hasting algorithm can be summarized as in Algorithm 5.

Algorithm 5 Modified Metropolis-Hastings algorithm.

Input:
- Initial state $\mathbf{x}_1 \in F_{i-1}$
- Joint density $f_{\mathbf{x}}(\mathbf{x}) = \prod_{k=1}^{d} f_k(x_k)$
- Size of sample N
- Proposal uniform density $r(\cdot|\alpha_i)$

for $j = 1, \ldots, N$ **do**
  for $k = 1, \ldots, d$ **do**
    Sample component $\widetilde{x}_k^{j+1} \sim r(\cdot|x_k^j)$
    Compute ratio $h = \frac{f_k(\widetilde{x}_k^{j+1})}{f_k(x_k^j)}$
    Set $\widetilde{x}_k^{j+1} = \begin{cases} \widetilde{x}_k^{j+1}, & \text{with probability } \min\{1, h\} \\ x_k^j, & \text{with probability } 1 - \min\{1, h\} \end{cases}$
  end for
  Accept/Reject $\widetilde{\mathbf{x}}^{j+1}$
  Check if $\widetilde{\mathbf{x}}^{j+1} \in F_{i-1}$
  Set $\mathbf{x}^{k+1} = \begin{cases} \widetilde{\mathbf{x}}^{j+1}, & \text{if } \widetilde{x}^{j+1} \in F_{i-1} \\ \mathbf{x}^j, & \text{if } \widetilde{x}^{j+1} \notin F_{i-1} \end{cases}$
end for

Output: $\mathbf{X}^j = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ a Markov chain of length N with stationary pdf $q(\cdot|F_{i-1})$

This algorithm ensures that the next sample is always in F_{i-1}. Choosing the uniform density centered at the previous state as a proposal, $\widetilde{\mathbf{x}}^{j+1}$ has a high probability to be in F_{i-1}. It yields a rather high acceptance ratio h that accelerates the exploration of the failure region.

When the sample $\mathbf{X}^i = \{\mathbf{x}_1^i, \ldots, \mathbf{x}_N^i\} \sim q(\cdot|F_{i-1})$ is available, we can estimate the conditional probability p_i by:

$$p_i = \mathrm{P}(F_i|F_{i-1}) \approx \widehat{p}_i = \frac{1}{N}\sum_{k=1}^{N} \mathbf{I}_{F_i}(\mathbf{x}_k^i), \tag{4.20}$$

and the final estimation is given by:

$$\widehat{p}_f = \prod_{i=1}^{m} \widehat{p}_i. \tag{4.21}$$

Using the provided algorithm, the intermediate failure probabilities $p_i, \;\; i = 1, \ldots, m$ can be estimated by sampling from conditional density $\mathbf{x} \sim q(\cdot|F_{i-1}), \;\; i = 1, \ldots, m$. However, it remains to select the sequence of the intermediate failure events $F_i, \;\; i = 1, \ldots, m$. We discuss this choice as well as some implementation issues in the following section.

4.2.4.1 Practical implementation

Subset simulation proceeds as follows. As discussed above, we estimate first $p_1 = \mathrm{P}(F_1|\Omega) \approx \widehat{p}_1$ by direct Monte Carlo sampling from the original joint density $f_{\mathbf{x}}(\mathbf{x})$. From this sample we can pick up those who are already in F_1. Clearly, this picked sample is distributed as $q(\cdot|F_1)$ and it can be used as a starting point for the next Markov chain. By sampling from $q(\cdot|F_1)$, we can estimate $p_2 = \mathrm{P}(F_2|F_1)$ according to (4.20). Then, we select from this new sample the points that are in F_2 to estimate $p_3 = \mathrm{P}(F_3|F_2)$. We repeat this sequential sampling until we reach the failure domain $F_m = \Omega_f$. Finally, the failure probability estimator is expressed as (4.21).

The choice of the intermediate failure domains directly affects the efficiency of the subset simulation procedure. It defines how fast the failure region Ω_f is approximated. If $g(\mathbf{x})$ is a performance function, then the failure region $\Omega_f = \{\mathbf{x} \in \Omega : g(\mathbf{x}) \leq 0\}$ is the last member in the sequence of intermediate failure events: $\Omega_f = \bigcap_{i=1}^{m} F_i = F_m = \{\mathbf{x} \in \Omega : g(\mathbf{x}) \leq 0\} \subset \Omega$.

Let us consider a sequence of intermediate positive thresholds $y_1 > y_2 > \ldots >$

$y_m = 0$. We can define the intermediate failure events as:

$$F_i = \{\mathbf{x} \in \Omega : g(\mathbf{x}) \leq y_i\} \text{ and } p_i = \mathrm{P}(g(\mathbf{x}) \leq y_i).$$

The choice of the sequence $(y_i)_{i=1}^m$ defines the intermediate conditional probabilities $p_i = \mathrm{P}(g(\mathbf{x}) \leq y_i)$. If the sequence $(y_i)_{i=1}^m$ decreases slowly, then the values $(p_i)_{i=1}^m$ are large and requires smaller sample size N for reliable estimation. Such sequence requires more simulation steps though. Conversely, a fast decreasing sequence leads to smaller intermediate conditional probabilities $(p_i)_{i=1}^m$ and higher sample size N [Au and Beck, 2003].

In practice, the choice of the thresholds sequence $(y_i)_{i=1}^m$ in advance is not straightforward. A reasonable alternative may be to define $(y_i)_{i=1}^m$ iteratively in such way that the corresponding estimated conditional probabilities are equal to a predetermined fixed value $p_i = \alpha_0$. Then, the sequence $(y_i)_{i=1}^m$ is evaluated for every subset simulation procedure level i as a quantile of α_0 for the conditional distribution $q(\cdot|F_{i-1})$. For a target failure probability p_f of order $10^{-4} - 10^{-6}$, the choice of $\alpha_0 = 0.1$ provides a reliable estimation. We will use the same number N of samples at every stage of subset simulation.

We estimate $(y_i)_{i=1}^m$ sequentially as the empirical quantiles of $q(\cdot|F_{i-1})$. The first level y_1 is estimated from a sample of size N according to the original joint pdf $\mathbf{X}^1 \sim f_{\mathbf{x}}(\mathbf{x})$. Then, an empirical α_0-quantile of $\mathbf{g}^1 = g(\mathbf{X}^1)$ is given as follows:

$$\widehat{y}_1 : \mathrm{P}(\mathbf{x} \in \Omega : g(\mathbf{x}) \leq y_1) \approx \sum_{l=1}^{N} \mathbf{I}(g(\mathbf{x}_l^1) \leq \widehat{y}_1)/N = \alpha_0.$$

If $y_1 \leq 0$ then we have already sampled from the failure region Ω_f. The failure probability can be estimated with the sample $\mathbf{x}^1$:

$$\widehat{p}_f = \sum_{l=1}^{N} \mathbf{I}(\mathbf{g}_l^1 \leq 0)/N.$$

If $y_1 > 0$, it means that we need to continue to sample. At every stage we estimate $\widehat{y}_i$ so that:

$$\mathrm{P}(\mathbf{x} \in \Omega : g(\mathbf{x}) \leq y_i) \approx \mathrm{P}(g(\mathbf{X}^{i-1}) \leq \widehat{y}_i) = \alpha_0.$$

The algorithm is summarized in Algorithm 6.

Algorithm 6 Subset simulation algorithm.

Input:
- Intermediate conditional probability α_0
- Sample size N
- Joint pdf $f_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{d} f_i(x_i)$

Set $j = 1$

Sample $\mathbf{X}^1 = \{\mathbf{x}_1^1, \ldots, \mathbf{x}_N^1\} \sim f_{\mathbf{x}}(\cdot)$

Evaluate $\mathbf{g}^1 = \{g_1^1, \ldots, g_N^1\} = \{g(\mathbf{x}_1^1), \ldots, g(\mathbf{x}_N^1)\}$

Estimate $y_1 : \mathrm{P}(\mathbf{x} : g(\mathbf{x}) \leq y_1) = \alpha_0$ empirically by $\widehat{y}_1 : \sum_{l=1}^{N} \mathbf{I}(g_l^1 \leq \widehat{y}_1)/N = \alpha_0$

Set $\mathbf{N_f^1} = \sum_{l=1}^{N} \mathbf{I}(g_l^1 \leq \widehat{y}_1)$

if $\widehat{y}_1 \leq 0$ **then**

Set $\widehat{p}_f = \mathbf{N_f^1}/N$

STOP

end if

while $\widehat{y}_j > 0$ **do**

Set $j = j + 1$

Sample $\mathbf{X}^j = \{\mathbf{x}_1^j, \ldots, \mathbf{x}_N^j\} \sim q(\cdot|F_{j-1})$, centered at $\mathbf{X}_F^{j-1} = \{\mathbf{x}_i^{j-1} : g(\mathbf{x}_i^{j-1}) \leq \widehat{y}_{j-1}\} \in F_{j-1}$

Evaluate $\mathbf{g}^j = \{g_1^j, \ldots, g_N^j\} = \{g(\mathbf{x}_1^j), \ldots, g(\mathbf{x}_N^j)\}$

Estimate $\widehat{y}_j : \sum_{l=1}^{N} \mathbf{I}(g_j^l \leq \widehat{y}_j)/N = \alpha_0$

Set $\mathbf{N_f^j} = \sum_{l=1}^{N} \mathbf{I}(\mathbf{g}_l^j \leq \widehat{y}_j)$

end while

if $y_j \leq 0$ **then**

Set $m = j$ and $p_m = \sum_{l=1}^{N} \mathbf{I}(g_l^j \leq 0)/N = \mathbf{N_f^m}/N$

end if

Output: Estimation of the failure probability $\widehat{p}_f = \alpha_0^{m-1} p_m$

4.2.4.2 Statistical properties of the estimator

Au and Beck [2001] and Au and Beck [2003] provide detailed discussion about the statistical properties of the previous estimation. Further, they also discuss the choice of the tuning parameters such as the sample size N and the intermediate failure probability α_0. Here, we provide a brief analysis of the estimator.

Firstly, we will discuss the statistical properties of the intermediate failure probability estimator. Particularly, we will provide estimations of the coefficient of variation for different steps of the subset simulation algorithm. For the first step, we start from the distribution density $f_{\mathbf{x}}(\mathbf{x})$. Therefore, the coefficient of variation can be computed in the same way as for crude Monte Carlo sampling:

$$\delta_1 = \sqrt{\frac{1-p_1}{Np_1}}, \quad \text{where} \quad p_1 = \mathrm{P}(F_1).$$

Intermediate conditional probability estimator

For intermediate failure events, the coefficient of variation is different. It accounts for the correlation among the states of a Markov chain at intermediate levels [Au and Beck, 2001; Zuev et al., 2011].

Assume that at level i of the subset simulation algorithm we aim to estimate the conditional failure probability $p_i = \mathrm{P}(F_i|F_{i-1})$. Since the intermediate conditional probabilities are estimated sequentially, at level i we have already estimated the probabilities $p_1, \ldots, p_{i-1}$ and the corresponding conditional samples used for estimation are $\mathbf{X}^1, \ldots, \mathbf{X}^{i-1}$. Every sample $\mathbf{X}^1, \ldots, \mathbf{X}^{i-1}$ is of size N. To evaluate the intermediate conditional failure probability $\widehat{p}_i \approx p_i = \mathrm{P}(F_i|F_{i-1})$, we generate a Markov chain distributed with $q(\cdot|F_{i-1})$. In practice, at every step the algorithm uses not a single one, but several simultaneous Markov chains starting from the points $\{\mathbf{x} \in \mathbf{X}^{i-1} : \mathbf{x} \in F_{i-1}\}$. The number of Markov chains at level i is $N_c = N/\mathbf{N}_f^{i-1}$. If α_0 is a predetermined conditional probability, then $\mathbf{N}_f^{i-1} = \alpha_0 N$ and $N_c = 1/\alpha_0$. Therefore, to calculate the coefficient of variation of the estimator at level i we have to take into account the correlation between the states of different Markov chains. The coefficient of variation for every intermediate step may be expressed as [Au and Beck, 2001]:

$$\delta_i = \sqrt{\frac{1-p_i}{p_i N}(1+\gamma_i)},$$

where

$$\gamma_i = 2 \sum_{k=1}^{N/N_c - 1} \left(1 - \frac{kN_c}{N}\right) \frac{R_i(k)}{R_i(0)}$$

is a correlation factor between the states of a Markov chain. Let us denote $x_{l,k}^i$ the k^{th} sample of the l^{th} chain at the i^{th} step of the algorithm. Define:

$$R_i(k) = \mathbb{E}\left[\mathbf{I}_{F_i}\left(x_{l,1}^{i-1}\right)\mathbf{I}_{F_i}\left(x_{l,1+k}^{i-1}\right)\right] - p_i^2,$$

which is the autocovariance of the stationary sequence:

$$\left\{\mathbf{I}(x_{l,k}^i), k = 1, \ldots, N_f^i = N/N_c\right\} \quad \text{at lag} \quad k.$$

When the generated samples are dependent, $\gamma_i > 0$ and the coefficient of variation δ increases. In the independent case, $\gamma_i = 0$ and the coefficient of variation is equal to a coefficient of variation corresponding to a Crude MC sampling.

Failure probability estimator

Intermediate estimators $\widehat{p}_i$ converge almost surely to the intermediate conditional probabilities p_i (Strong Law of Large Numbers) [Au and Beck, 2001]. Therefore, the final estimator $\widehat{p}_f$ converges to the failure probability p_f.

$$\begin{gathered}\widehat{p}_1 \overset{N\to\infty}{\longrightarrow} \mathrm{P}(F_1) \\ \widehat{p}_i \overset{N\to\infty}{\longrightarrow} \mathrm{P}(F_i|F_{i-1}), i = 2, \ldots, m \\ \widehat{p}_f \overset{N\to\infty}{\longrightarrow} p_f\end{gathered}$$

Au and Beck [2001] prove that due to the correlation between the estimators $\widehat{p}_i$, the estimator $\widehat{p}_f$ is biased for every fixed N, but it is unbiased asymptotically. The correlation comes from the fact that to sample the new Markov chain at every level of subset simulation i, we use the points from F_{i-1} (that were used to estimated $\widehat{p}_{i-1}$).

Choice of the level α_0

The optimal choice of the conditional failure probability α_0 is a trade off between the number of algorithm stages and the maximum allowable N available for reliable estimation. Au and Beck [2003] provide an estimation of the coefficient of variation of the estimator $\widehat{p}_f$. Let us assume that there are m levels in the algorithm. At every

level a Markov chain is sampled during N steps. Then the coefficient of variation may be approximated by:

$$\delta^2 \approx \frac{m(1-\alpha_0)}{N\alpha_0}(1+\overline{\gamma}), \tag{4.22}$$

where $\overline{\gamma}$ is the average value of the correlation parameters $\gamma_i, i = 1, \ldots, m$ [Au and Beck, 2001]. We denote, the total number of draw by $N_T = mN$. Also, the order of the failure probability can be estimated by: $\widehat{p}_f \approx \alpha_0^m$. So, $m = \log \widehat{p}_f / \log \alpha_0$. Therefore, we can rewrite (4.22) as:

$$\delta^2 \approx \frac{(1-\alpha_0)}{N_T\alpha_0}\frac{(\log \widehat{p}_f)^2}{(\log \alpha_0)^2}(1+\overline{\gamma}).$$

Hence, we can estimate the size of a sample required for a reliable estimation with a given coefficient of variation δ:

$$N_T = \frac{(1-\alpha_0)(\log p_f)^2(1+\overline{\gamma})}{\alpha_0(\log \widehat{p}_f)^2\delta^2}$$

We can analyze the dependency of the coefficient of variation on α_0 for a fixed γ. Figure 4.6 represents a comparative analysis for $p_f = 10^{-4}$, $N_T = 5 \times 10^3$ and $\gamma = 0, 2, 4, 6, 10$. We can see that choosing $\alpha_0 \in [0.1, 0.4]$ leads to the most efficient implementation. We select relatively low value of N_T to clearly observe the effect of dependency.

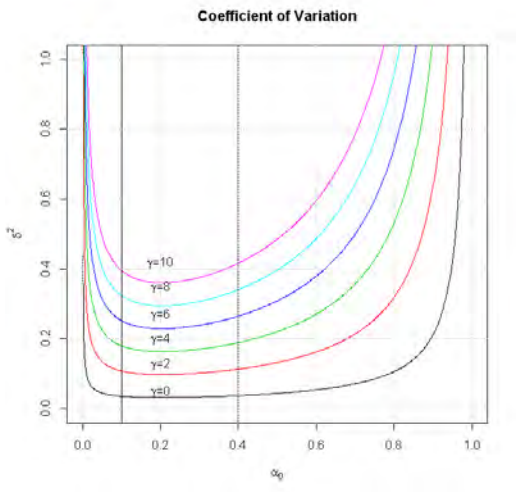


Figure 4.6: Coefficient of variation δ as a function of α_0.

Recently, Zuev et al. [2011] applied a Bayesian approach to the estimator $\widehat{p}_f$. By assuming that the intermediate conditional failure probability is a stochastic variable, it is possible to compute the posterior distribution of the estimator $\widehat{p}_f$. Firstly, we assume as a prior distribution for p_j the uniform distribution on $[0,1]$, Then, the estimator $\widehat{p}_j$ is a sum of Bernoulli variables as a sum of indicator functions on a sample $\mathbf{X}^j = \{\mathbf{x}_1^j, \dots, \mathbf{x}_N^j\}$. It leads to the beta distribution for intermediate estimators: $\widehat{p}_j \sim \mathcal{B}(n_j + 1, N - n_j + 1)$, where $n_j = \sum_{k=1}^{N} \mathbf{I}_{F_j}(\mathbf{x}_k^{j-1})$. Then, the final estimator is a product of beta random variables. Zuev et al. [2011] prove that $\widehat{p}_f \sim \mathcal{B}(\mathbf{a}, \mathbf{b})$ follows a beta distribution with the following parameters:

$$\mathbf{a} = \frac{\prod_{j=1}^{m} \frac{n_j+1}{N+2} \left(1 - \prod_{j=1}^{m} \frac{n_j+2}{N+3}\right)}{\prod_{j=1}^{m} \frac{n_j+2}{N+3} - \prod_{j=1}^{m} \frac{n_j+1}{N+2}}$$

$$\mathbf{b} = \frac{\left(1 - \prod_{j=1}^{m} \frac{n_j+1}{N+2}\right) \left(1 - \prod_{j=1}^{m} \frac{n_j+2}{N+3}\right)}{\prod_{j=1}^{m} \frac{n_j+2}{N+3} - \prod_{j=1}^{m} \frac{n_j+1}{N+2}}$$

where $n_j = \sum_{k=1}^{m} \mathbf{I}_{F_j}(x_k^{j-1})$ is the number of times when $\mathbf{X}^{j-1}$ falls into the intermediate failure region F_j. In view of the practical implementation, at level j we estimate the threshold y_j as α_0-quantile of sample $\mathbf{X}^{j-1}$ of size N. It means that:

$$n_j = \begin{cases} \alpha_0 N, & \text{if } \ j < m, \\ \mathbf{N}_f^m, & \text{if } \ j = m \end{cases}$$

where N is the size of the sample at every level and $\mathbf{N}_f^m$ is the number of points at the last level falling into the failure region $F_m = \Omega_f$.

This gives us a possibility to study the influence of the choice of the algorithm parameters such as α_0 and N to the estimation of the failure probability $\widehat{p}_f$.

4.2.4.3 Analytical example

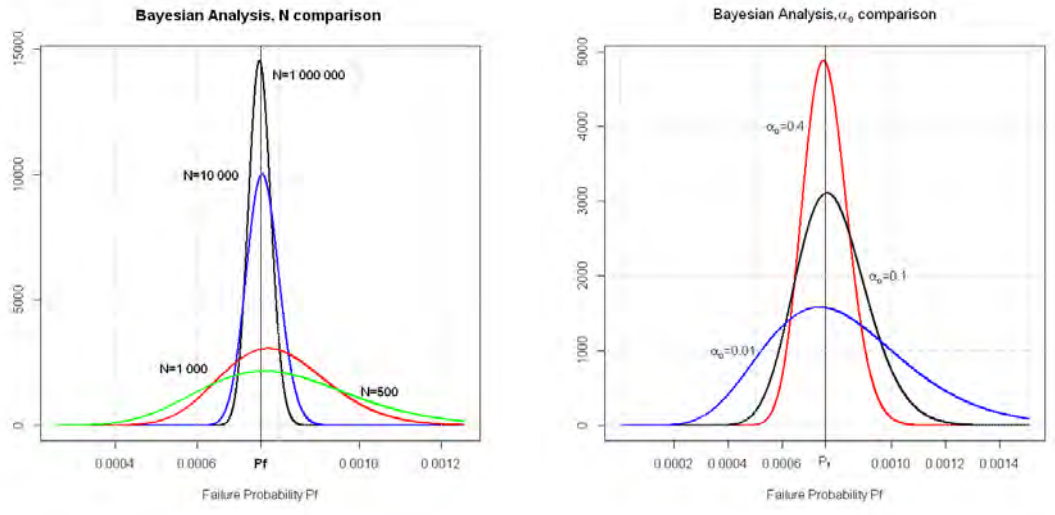
We consider an analytical example similar to the one of Section 4.2.3. We will consider as a performance function the following linear combination of Gaussian variables:

$$f(\mathbf{x}) = 4 - 0.1x_1 - 0.3x_2 - 0.7x_3 - x_4$$

where all the variables are independent with standard normal distribution: $x_i \sim \mathcal{N}(0,1)$. The function $f(\mathbf{x})$ is a Gaussian random variable as a sum of independent standard normal variables. The mean is $\mu = 4$ and the variance is $\sigma^2 = 0.1^2 + 0.3^2 + 0.7^2 + 1 = 1.59$. Therefore, we can compute precisely the failure probability:

$$\mathrm{P}(f(\mathbf{x}) \leq 0) = \Phi(-\mu/\sigma) = 7.564 \times 10^{-4}.$$

Figure 4.7(a) represents a comparison of the posterior distribution for the estimator of p_f for different values of N at the fixed value of $\alpha_0 = 0.1$. Figure 4.7(b) displays a comparison between the empirical distributions of the estimator for different values of α_0 at the fixed value $N = 10^3$.



(a) Analysis of N influence

(b) Analysis of α_0 influence

Figure 4.7: Bayesian analysis.

As it can be seen, the computed distributions are centered around the true value of p_f. The variance, and therefore the coefficient of variation, decreases when N increases. This implies that for the higher value of N the uncertainty about the computed estimator p_f is lower. On the contrary, a low value of $\alpha_0 = 0.01$ leads to fewer levels in the algorithm levels and the variance becomes higher than for the values of $\alpha_0 \in [0.1, 0.4]$. Estimated coefficient of variations are displayed in Table 4.2.

	Sample size				α_0		
	$N = 500$	$N = 1000$	$N = 10^4$	$N = 10^6$	$\alpha_0 = 0.01$	$\alpha_0 = 0.1$	$\alpha_0 = 0.4$
c.o.v. δ	0.233	0.165	0.052	0.036	0.321	0.165	0.108

Table 4.2: Coefficients of variation.

For this example, we may conclude that for $N = 10^4$ the algorithm provides reliable estimation of the failure probability with coefficient of variation of 5%.

4.3 GP model based reliability analysis

In the previous section we have discussed different methods to evaluate the failure probability p_f for a given analytical performance function $g(\mathbf{x})$. Often, dealing with dynamic reservoir simulators, the analytical expression for a performance function is not available. In this section, we introduce another approximation method. It is based on approximation of the performance function by a response surface model. Particularly, Gaussian Process (GP) response surface model with subsequent application of subset simulation algorithm will be considered. The use of GP response surface model in reliability analysis has recently gained attention by different authors [Bect et al., 2012; Bichon et al., 2011; Dubourg et al., 2011; Echard et al., 2011; Kaymaz, 2005; Picheny, 2009].

The expensive simulator code is replaced with a response surface model that is faster to compute. In the case of CO_2 reservoir storage, the failure event occurs when the reservoir pressure exceeds the cap rock fracturing pressure. Let $P(\mathbf{x})$ be a function of the reservoir pressure and $P_{fracture}$ be a geomechanically estimated cap rock fracturing pressure. Then, the failure region may be defined as:

$$\Omega_f = \{\mathbf{x} \in \Omega : P(\mathbf{x}) \geq P_{fracture}\}.$$

Therefore, the performance function to approximate is:

$$g(\mathbf{x}) = P_{fracture} - P(\mathbf{x}) \quad \text{and} \quad p_f = \mathrm{P}(g(\mathbf{x}) \leq 0)$$

The function $g(\mathbf{x})$ is replaced by a GP response surface model $\widehat{g}(\mathbf{x})$. This response surface model is built using an experimental design $\{\mathbf{X}^n, \mathbf{Y}^n = g(\mathbf{X}^n)\}$. Then, the

estimator for failure probability is:

$$\widehat{p}_f = \mathrm{P}(\widehat{g}(\mathbf{x}) \leq 0) = \int_{\Omega_{\widehat{f}}} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x},$$

where $\Omega_{\widehat{f}} = \{\mathbf{x} \in \Omega : \widehat{g}(\mathbf{x}) \leq 0\}$ is an approximated failure region and $f_{\mathbf{x}}(\mathbf{x})$ is the joint density of $\mathbf{x} \in \Omega \subset \mathbb{R}^d$.

The accuracy of the approximation of the GP response surface model is crucial for a reliable estimation of the failure probability p_f. Furthermore, the experimental design plays a major role in the model reliability. The existing design and the related response surface model can be improved by adaptive sampling in the region of interest (the failure region Ω_f in our case). In order to improve the quality of the model, we propose to update the original experimental design with additional simulations around the boundary of the failure region. Here, we propose an adaptive sampling technique that is based on the subset simulation algorithm. First, we discuss some existing techniques for reliability-oriented adaptive designs. Then, we introduce a novel adaptive sampling technique. In order to demonstrate the efficiency of the proposed technique, we will provide examples on an analytical function and on a CO_2 reservoir storage case. We will compare the estimations of the failure probability provided by the original GP model and the improved GP model.

4.3.1 Sequential adaptive design

Local refinement of a particular region of interest has recently gained attention in the field of reliability analysis. When a GP response surface model is used to estimate the failure probability, it is required that the model is sufficiently accurate in the region of failure. However, classical design sampling techniques (such as maximin LHD [Santner et al., 2003]) aim to cover uniformly the input domain: $\mathbf{x} \in \Omega \subset \mathbb{R}^d$. Therefore, in order to get more information about the function close to or inside the failure region Ω_f we propose to refine iteratively the original experimental design $\{\mathbf{X}^n, \mathbf{Y}^n = g(\mathbf{X}^n)\}$. At each iteration we will add adaptively the points close to the limit-state surface, thus reducing the uncertainty about the failure region. This method is referred to an adaptive refinement. The objective of the adaptive refinement is to improve the reliability of the failure probability estimator.

While working with dynamic reservoir simulators, the experimental design is of primary importance, since the CPU time required for a single simulator run may be large. Here, we focus on the construction of an accurate response surface model to estimate the failure probability p_f. Sequential adaptive designs aim at minimizing uncertainty close to the limit state surface $\{\mathbf{x} \in \Omega : g(\mathbf{x}) = 0\}$. This approach assumes that the original experimental design is performed for example with a Latin hypercube sampling [McKay et al., 1979]. Then subsequent runs are performed at new input configurations which are selected under some statistical criterion minimizing the uncertainty. The general algorithm of any sequential adaptive design can be described in three stages:

1. Find the best configuration $\mathbf{x}^*$ satisfying the chosen criterion
2. Run the simulator at $\mathbf{x}^*$ to obtain $g(\mathbf{x}^*)$
3. Update the GP model by including $(\mathbf{x}^*, g(\mathbf{x}^*))$ to the experimental design

The procedure continues until some stopping criterion is reached. The stopping criterion can rely for example on the prediction error or on the chosen measure of uncertainty.

Different adaptive refinement techniques were recently proposed. Bect et al. [2012] propose a stepwise uncertainty reduction algorithm. This algorithm is based on a Bayesian approach for the problem of failure probability estimation. It takes the origin from the Efficient Global Optimization algorithm (EGO) [Jones et al., 1998]. Bect et al. [2012] propose to measure an error of the estimation of the failure probability. They derive an upper bound for this measure that does not depend on the value of p_f. The next point to add is chosen by minimization of this error bound. Picheny et al. [2010] suggest another error measurement which they call the weighted integrated mean-square error. The weighting density depends only on the available original observation data. Bichon et al. [2011] developed so called the expected feasibility function which measures the expected proximity to the limit threshold value. Every next point is added by minimizing the feasibility function and approaching the vicinity of the limit state surface. Finally, Dubourg [2011] introduce a refinement method based on splice sampling coupled with a k-means classification algorithm.

The presented adaptive techniques have both advantages and drawbacks. The choice of an optimal technique is always a trade-off between accuracy and the re-

quired CPU time for every iteration. For expensive simulators, the CPU time required for every simulator run could be very restrictive. Consequently, the optimal number of points to be added at every iteration should be as small as possible, while bringing as much information from the failure region. One is always interested in approaching the limit state surface and the failure region fast with less possible number of simulator runs. At the same time, subset simulation is known as an effective estimation of the failure probability even in very high dimensions. Depending on the chosen intermediate probability α_0, the failure region is approached rather rapidly. At every level of the subset simulation algorithm, we estimate the sequence of intermediate thresholds $(y_j)_{j=1}^m$ by sampling from the conditional density $q(\cdot|F_{j-1})$. The intermediate failure events are considered as a nested sequence: $\Omega = F_0 \supset F_1 \supset \ldots \supset F_m = \Omega_f$. So that, at every level of the algorithm we are sampling closer to the failure region Ω_f. The sample of size N generated at every level j: $\mathbf{X}^j = \{\mathbf{x}_1^j, \ldots, \mathbf{x}_N^j\}$ is automatically available. We suggest to incorporate these data to the experimental design. At every level of the subset simulation algorithm we update the original GP model with the points sampled at the previous level.

Thus, we provide a new subset simulation algorithm that involves the GP model refinement at every level. The refinement takes place until a stopping criterion is reached. In the end, the last updated GP model is used to estimate the failure probability p_f. As a result, it provides a reliable failure probability estimation.

The next subsection describes the proposed refinement algorithm in details.

4.3.2 Adaptive refinement algorithm

The complete algorithm is summarized in Algorithm 7. Here, we provide a brief explanation and argumentation of the algorithm.

The proposed refinement algorithm has two principal objectives: the first one is to explore the failure region, the second one is to improve the estimation of the failure probability. It may be achieved with the minimization of the uncertainty about the performance function $g(\mathbf{x})$ near the limit state surface $\Omega_s = \{\mathbf{x} \in \Omega : g(\mathbf{x}) = 0\}$ and the failure region $\Omega_f = \{\mathbf{x} \in \Omega : g(\mathbf{x}) < 0\}$. The GP response surface model based on such design is specifically built to estimate the failure probability p_f.

As discussed above, the basic idea of the subset simulation algorithm is to represent the failure probability p_f as a product of larger conditional probabilities by introducing intermediate failure events $F_j = \{\mathbf{x} \in \Omega : g(\mathbf{x}) < y_j\}$, where

$\Omega = F_0 \supset F_1 \supset \ldots \supset F_m = \Omega_f$ and $y_1 > y_2 > \ldots > y_m = 0$. Then, the estimation of the failure probability p_f is:

$$\widehat{p}_f = \prod_{j=1}^{m} \mathrm{P}(F_i|F_{i-1}).$$

By choosing the value of the intermediate failure probability α_0 and the size of the intermediate samples N at the level k of the algorithm the intermediate threshold value y_k is estimated. This is done by sampling $\mathbf{X}^k = \{\mathbf{x}_1^k, \ldots, \mathbf{x}_N^k\} \sim q(\cdot|F_{k-1})$, where the conditional density $q(\cdot|F_{k-1})$ is realized with Markov chain Monte Carlo sampling (see Algorithm 6 for more details). We suggest to perform the subset simulation algorithm simultaneously with the GP model refinement. It means that at every level of the subset simulation algorithm the GP model is updated with the refined experimental design. It is continued until some stopping criterion is reached. As soon as the criterion is reached, to estimate the failure probability the classical subset simulation algorithm is applied to the last updated GP model $\widehat{g}_{\mathbf{f}}(\mathbf{x})$. To update the experimental design at the iteration $k+1$ we propose to use the data sampled at the level k of the subset simulation algorithm $\mathbf{X}^k = \{x_1^k, \ldots, x_N^k\} \sim q(\cdot|F_{k-1})$. The sequence $\Omega = F_0 \supset F_1 \supset \ldots \supset F_m = \Omega_f$ is a nested sequence converging to the failure region. At every refinement iteration by a correct statistical analysis of these data, we can select out the points to add in order to retrieve the most information about the approximated function in the failure region Ω_f. Below we provide the selecting and the stopping criteria of the refinement procedure.

First, we have to define the number of points to be added at every refinement iteration. We will denote it by N_{add}. Now, we want to precise that the original model $\widehat{g}(\mathbf{x})$ is built according to the original experimental design of size n: $\mathbf{X}^n$ and the simulator runs $\mathbf{Y}^n = g(\mathbf{X}^n)$. It is important to note that in order to keep the points only close to the failure region, after the first iteration of the refinement algorithm we will replace the original design $\mathbf{X}^n$ with the new selected points $\mathbf{X}_{add}^1 = \{\mathbf{x}_1^1, \ldots, \mathbf{x}_{N_{add}}^1\}$. Then we will add the points only to this new replaced design. The model based on such experimental design is specifically focused on the failure region. Let us denote by $\widehat{p}_f^{SS}$ the estimation of the failure probability provided by the subset simulation applied to the original model $\widehat{g}(\mathbf{x})$. Suppose that we are given with α_0, the value of the intermediate failure probability, and N, the size of the intermediate conditional samples. Recall that for any arbitrary input configuration $\mathbf{x}^*$ the kriging

predictor yields the approximation $\widehat{g}(\mathbf{x}^*)$ and the estimation of the variance of this approximation $\mathrm{VAR}(\widehat{g}(\mathbf{x}^*))$. Hence, the estimation of the coefficient of variation at the input $\mathbf{x}^*$ is also available. We will denote it by:

$$\delta(\mathbf{x}^*) = \sqrt{\mathrm{VAR}(\widehat{g}(\mathbf{x}^*))}/\widehat{g}(\mathbf{x}^*).$$

The coefficient of variation is a statistical measure of the dispersion around the mean. Therefore, the lower the coefficient of variation, the lower the uncertainty about the performance function $g(\cdot)$ around the input $\mathbf{x}^*$. By minimizing the coefficient of variation δ of the GP response surface model $\widehat{g}(\mathbf{x})$ close to the failure region, we reduce the uncertainty about the performance function in this region. Therefore, at the level k for a sample $\mathbf{X}^k = \{x_1^k, \dots, x_N^k\} \sim q(\cdot|F_{k-1})$ we select out the points in order to minimize the coefficient of variation $\boldsymbol{\delta^k} = \{\delta_1^k, \dots, \delta_N^k\}$.

The stopping criterion of the refinement procedure is defined by the maximum possible value for the coefficient of variation $\delta_{max} = \boldsymbol{\varepsilon}$. By selecting the threshold value $\boldsymbol{\varepsilon}$, we continue the algorithm until $\max\limits_{\mathbf{x}\in\mathbf{X}^k}\delta(\mathbf{x}) < \boldsymbol{\varepsilon}$. When the condition is met, we use the final updated GP model to estimate the failure probability.

Here, we want make a point about the selecting criterion. Let us denote the set of points added at the level k of the refinement algorithm by $\mathbf{X}_{add}^k = \{\mathbf{x}_1^k, \dots, \mathbf{x}_{N_{add}}^k\} \subset \mathbf{X}^k$. We will also use the notation $\widehat{g}^{k-1}$ for the GP response surface model evaluated at the level $k-1$. The model $\widehat{g}^{k-1}$ is built with the design $\{\mathbf{X}_{add}^1, \dots, \mathbf{X}_{add}^{k-1}\}$. Recall that at the level k of the subset simulation refinement algorithm we select out the points from the sample $\mathbf{X}^k = \{x_1^k, \dots, x_N^k\} \sim q(\cdot|F_{k-1})$ in order to minimize the corresponding value of the coefficient of variation $\boldsymbol{\delta^k} = \{\delta_1^k, \dots, \delta_N^k\}$. It means that we start refining with the input providing the maximum value $\delta_{max}^k = \max\limits_{\mathbf{x}\in\mathbf{X}^k}(\boldsymbol{\delta^k})$:

$$\mathbf{x}_1^k := \arg\max_{\mathbf{x}\in\mathbf{X}^k}(\boldsymbol{\delta^k}).$$

However, if the size of the sample $\mathbf{X}^k$ is rather large, e.g. $N = 10^4 - 10^6$, the sample points are highly concentrated. For this reason, we propose a modification of the algorithm in order to avoid sampling in the vicinity of the first point $\mathbf{x}_1^k$. Inside the refinement iteration k, once the input corresponding to δ_{max}^k has been identified, we suggest to artificially update the design $\{\mathbf{X}_{add}^1, \dots, \mathbf{X}_{add}^{k-1}\}$ with the pair $\{\mathbf{x}_1^k, \widehat{g}^{k-1}(\mathbf{x}_1^k)\}$, while keeping the hyperparameters $\boldsymbol{\theta}^{k-1} = \{\theta_1^{k-1}, \dots, \theta_d^{k-1}\}$ of the model $\widehat{g}^{k-1}$ unchanged. By adding this pair to the current experimental

design, the variance of $\widehat{g}^{k-1}$ at the point $\mathbf{x}_1^k$ is equal to zero. The variance of $\widehat{g}^{k-1}$ around the input $\mathbf{x}_1^k$ is close to zero. Therefore, now we can choose the next input $\mathbf{x}_2^k \in \mathbf{X}_{-1}^k = \mathbf{X}^k \backslash \{\mathbf{x}_1^k\}$:

$$\mathbf{x}_2^k := \arg \max_{\mathbf{x} \in \mathbf{X}_{-1}^k} \delta(\mathbf{X}_{-1}^k).$$

We continue until we sample N_{add} points. As soon as the required number of points is sampled, we evaluate the performance function $g(\cdot)$ by running the simulator at $\mathbf{X}_{add}^k = \{\mathbf{x}_1^k, \ldots, \mathbf{x}_{N_{add}}^k\}$. We can now reevaluate the hyperparameters $\boldsymbol{\theta}^k = \{\theta_1^k, \ldots, \theta_d^k\}$ and update the GP model by $\widehat{g}^k$ with $\{\mathbf{X}_{add}^1, \ldots, \mathbf{X}_{add}^k\}$.

This algorithm can be considered as a subset simulation algorithm for a Gaussian Process response surface models. At every iteration k we update the GP model $\widehat{g}^k(\mathbf{x})$ and recalculate the hyperparameters based on the experimental design from the failure region Ω_f. This algorithm provides a reliable estimation of the failure probability for the GP approximation of the performance function. The accuracy of the estimation depends on the value of the stopping criterion $\boldsymbol{\varepsilon}$. The higher the value of $\boldsymbol{\varepsilon}$, the lower the total number of refinement iterations. However, the lower the value of $\boldsymbol{\varepsilon}$, the higher the estimation accuracy. In practice, we propose to set $\boldsymbol{\varepsilon} = 0.1$. It leads to a reliable estimation with a rather low number of iterations.

The complete algorithm is presented in Algorithm 7. In the next sections we provide examples on a known analytical function and on a CO_2 reservoir storage case. We illustrate the convergence of the algorithm. In addition, we will compare the results provided by the original and updated GP models.

Algorithm 7 Adaptive refinement algorithm.

Input:
- Initial experimental design $\mathbf{X}^n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and GP model $\widehat{g}(\mathbf{x})$
- Number of samples to add at each iteration N_{add}
- Stopping criterion: $\boldsymbol{\varepsilon}$
- Subset simulation parameters: α_0 and N

Output: Updated GP model $\widehat{g}_f$ and the failure probability estimation $\widehat{p}_f^{SS}$

Set $k = 1$

The first step of SS algorithm: evaluate y_1 and sample $\mathbf{X}^1 = \{\mathbf{x}_1^1, \dots, \mathbf{x}_N^1\}$: $\mathrm{P}(\widehat{g}(\mathbf{x^1}) < y_1) = \alpha_0$. Set $\mathbf{g}^1 = \widehat{g}(\mathbf{X}^1)$.

Evaluate $\boldsymbol{\delta}^1 = \delta(\mathbf{X}^1)$ and $\boldsymbol{\delta}^1_{\max} = \max_{\mathbf{x}\in\mathbf{X}^1}(\delta(\mathbf{x}))$

if $\boldsymbol{\delta}^1_{\max} < \boldsymbol{\varepsilon}$ **then**
  stop
else
  for i **in** $1 : \mathrm{N}_{add}$ **do**
    Find $\mathbf{x}_i^1 \in \mathbf{X}^1 : \delta(\mathbf{x}_i^1) = \max_{\mathbf{x}\in\mathbf{X}^1}(\delta(\mathbf{x}))$
    Update $\mathbf{X}^n = \{\mathbf{X}^n, \mathbf{x}_i^1\}$ and $\mathbf{Y}^n = \{\mathbf{Y}^n, \widehat{g}(\mathbf{x}_i^1)\}$
    Update $\mathbf{X}^1_{add} = \{\mathbf{X}^1_{add}, \mathbf{x}_i^1\}$
    Update the model $\widehat{g}$ with the design $\{\mathbf{X}^n, \mathbf{Y}^n\}$. The updated model: $\tilde{g}(\cdot)$
    Re-evaluate the c.o.v. δ at the sample $\mathbf{X}^1$: $\boldsymbol{\delta}^1 = \delta_{\tilde{g}}(\mathbf{X}^1)$
  end for
end if

Evaluate the performance function: $\mathbf{Y}^1_{add} = g(\mathbf{X}^1_{add})$. Replace the original design: $\mathbf{X}^n := \mathbf{X}^1_{add}$ and $\mathbf{Y}^n := \mathbf{Y}^1_{add}$. Evaluate the GP model $\widehat{g}^1$ based on $\{\mathbf{X}^1_{add}, \mathbf{Y}^1_{add}\}$

while $\boldsymbol{\delta}^k_{\max} \geq \boldsymbol{\varepsilon}$ **do**
  Set $k = k + 1$ and $\mathbf{X}^k_{add} = \{\}$
  Perform step k of SS algorithm: evaluate y_k and sample $\mathbf{X}^k = \{\mathbf{x}_1^k, \dots, \mathbf{x}_N^k\}$: $\mathrm{P}(\mathbf{g}^k < y_k) = \alpha_0$, where $\mathbf{g}^k = \widehat{g}^{k-1}(\mathbf{X}^k)$
  Evaluate $\boldsymbol{\delta}^k = \delta(\mathbf{X}^k)$ and $\boldsymbol{\delta}^k_{\max} = \max_{\mathbf{x}\in\mathbf{X}^k}(\delta(\mathbf{x}))$
  for i **in** $1 : \mathrm{N}_{add}$ **do**
    Find $\mathbf{x}_i^k \in \mathbf{X}^k : \delta(\mathbf{x}_i^k) = \max_{\mathbf{x}\in\mathbf{X}^k}(\delta(\mathbf{x}))$
    Update $\mathbf{X}^k_{add} = \{\mathbf{X}^k_{add}, \mathbf{x}_i^k\}$
    Update the GP model $\widehat{g}^{k-1}$ with the design $\{\left(\mathbf{X}^n, \mathbf{X}^k_{add}\right); \left(\mathbf{Y}^n, \widehat{g}^{k-1}(\mathbf{X}^k_{add})\right)\}$. The updated GP model $\tilde{g}^k(\cdot)$.
    Re-evaluate the c.o.v. at the sample $\mathbf{X}^k$: $\boldsymbol{\delta}^k = \delta_{\tilde{g}^k}(\mathbf{X}^k)$
  end for
  Evaluate GP model $\widehat{g}^k$ basing on the refined design $\mathbf{X}^n := \{\mathbf{X}^n, \mathrm{X}^k_{add}\}$ and the simulator runs $\mathbf{Y}^n := \{\mathbf{Y}^n, \mathrm{Y}^k_{add}\}$
end while

Evaluate $\widehat{p}_f^{SS}$ with SS algorithm for the final model $\widehat{g}_f^k$

Return: Estimation of failure probability $\widehat{p}_f^{SS}$

4.4 Numerical examples

4.4.1 Analytical function example

In this section, we apply the algorithm on an analytical function. First, we compute the estimation of the failure probability by subset simulation algorithm applied to the known analytical function. We will use this estimation as a reference value p_f. Then, we will compare this reference value with the estimations given by the original GP model and the updated GP model. We also analyze a map of the variance and the coefficient of variation (c.o.v.) calculated at different levels of the refinement procedure.

The analytical function is given by the following formulation:

$$f(\mathbf{x}) = 3 - 0.3x_1^2 - 0.7x_2 \tag{4.23}$$

where $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ are independent random variables and $x_i \sim \mathcal{N}(0,1), i = 1,2$. The following table represents estimations provided by different reliability methods: FORM, SORM, Importance Sampling (IS), Directional Sampling (DS) and Subset Simulation (SS). For the subset simulation we use: $\alpha_0 = 0.1$ and $N = 10^4$. It can be seen that FORM/SORM underestimates the failure probability p_f.

	FORM	SORM	IS	DS	SS
$\widehat{p}_f$	1.645e-3	1.809e-3	3.564e-3	3.657e-3	3.662e-3
Function calls	48	53	1e6	1e6	28623

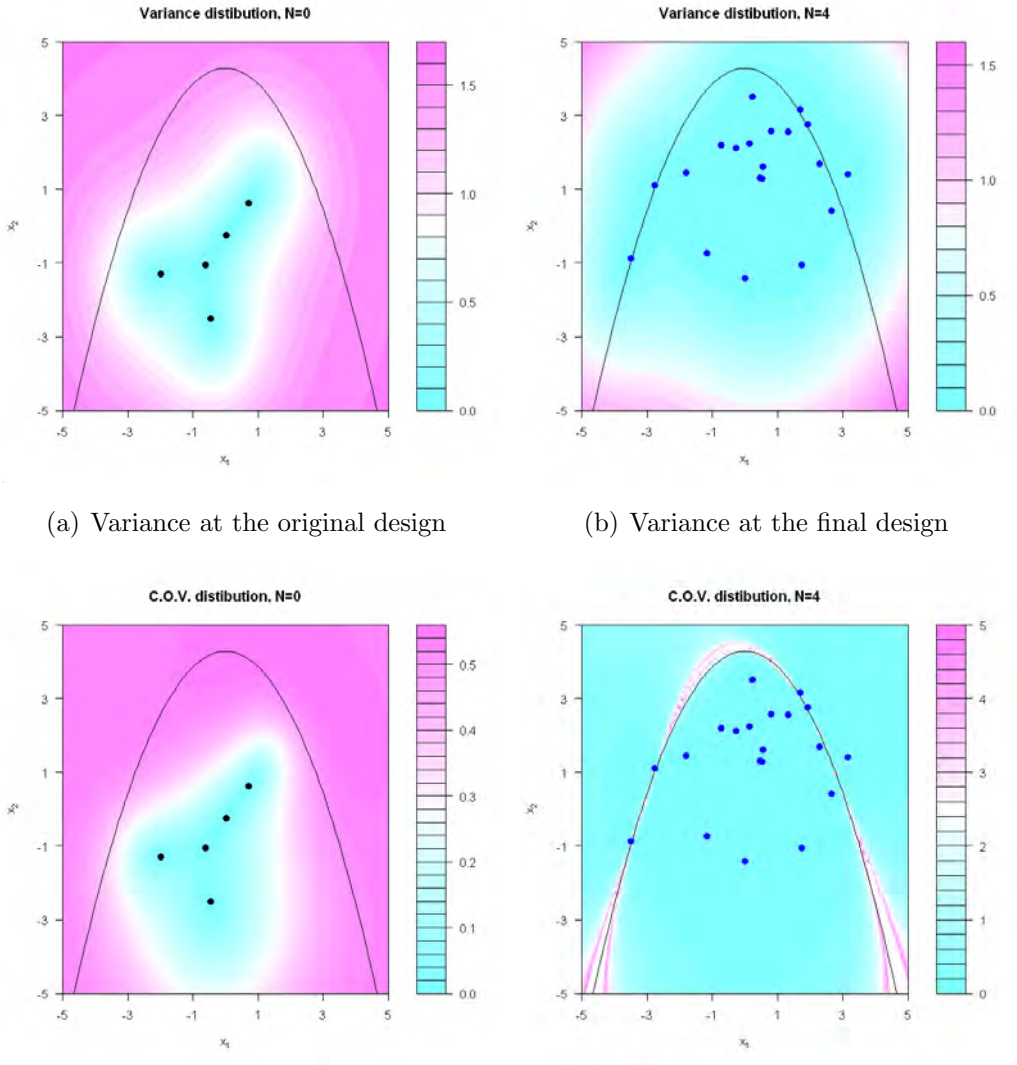
Table 4.3: p_f estimations.

The original GP response surface model is computed using the maximin Latin hypercube design $\{\mathbf{X}^n, \mathbf{Y}^n\}$ of size $n = 5$. The corresponding estimation of the failure probability is $\widehat{p}_f^o = 0$. We keep"' $\alpha_0 = 0.1$ and $N = 10^4$. We add $\mathrm{N}_{add} = 5$ points at each refinement step. The stopping threshold is $\boldsymbol{\varepsilon} = 0.1$. The algorithm stops after 4 iterations after 20 new added design points. Table 4.4 provides the evolution of the $\widehat{p}_f$ estimations and the maximum c.o.v. δ_{GP}.

	Step 1	Step 2	Step 3	Step 4
$\widehat{p}_f$	0.9866	0.006154	0.003308	0.003512
δ_{GP}	0.5396	2.3012	0.4024	0.0575

Table 4.4: $\widehat{p}_f$ and δ_{GP} estimations.

Figure 4.8 compares the distribution of the variance and the coefficient of variation (c.o.v.) between the original and the final designs. The solid line corresponds to the limit-state surface $g(x_1, x_2) = 0$. Black points are used in the original design and the blue points were added by the refinement algorithm. We easily see that the limit state surface is well approximated and that the variance is significantly smaller at the final step. This is also confirmed in the estimation of the failure probability. The estimation after 2 steps is sufficiently close to the reference value p_f.



(a) Variance at the original design

(b) Variance at the final design

(c) C.O.V. at the original design

(d) C.O.V. at the final design

Figure 4.8: Evolution of variance and coefficient of variation at different stages of the refinement algorithm.

4.4.2 CO_2 storage case example

In this section, we will consider the CO_2 reservoir storage example presented in Section 3.3.4. The original design consists of 30 simulator runs. The objective is to evaluate the risk of leakage due to the overpressure in the reservoir. Geomechanical experts established the fracturing pressure to be $\mathrm{P}_{fr} = 122$ bars. The original design used in this section is presented in Figure 4.9. This model considers three variables with uniform distribution at the given intervals. Details could be found in Table 3.5. However, we propose to use the uniform assumption for the refinement procedure if we know the boundaries of a distribution. It allows to explore the failure region more rapidly. After that, when the refined model is available, the failure probability can be recalculated under the chosen distribution assumption.

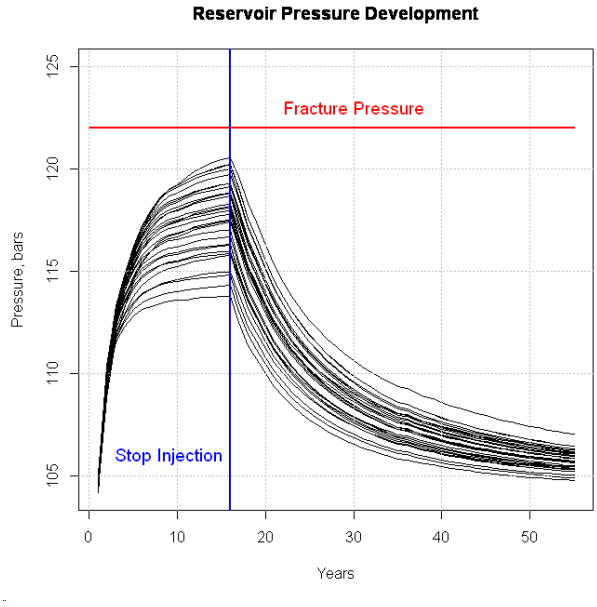


Figure 4.9: Reservoir pressure development.

Obviously, no curve of the original design overpasses the limit pressure value. In this study, we approximate the reservoir pressure in the last year of injection when there is the highest risk of leakage. We will denote by $P(\mathbf{x})$ the reservoir pressure and by $\widehat{P}(\mathbf{x})$ the corresponding GP response surface model. The objective is to estimate the following probability:

$$p_f = \mathrm{P}(P(\mathbf{x}) \geq \mathrm{P}_{fr}) \tag{4.24}$$

We use the subset simulation for the estimation of this probability. The original GP

model $\widehat{P}_0(\mathbf{x})$ is computed using the original design $\{\mathbf{X}^{30}, \mathbf{Y}^{30}\}$. The estimation of the failure probability for this model is $\widehat{p}_f^{SS} = 0$. By varying the threshold value, we can evaluate the maximum reservoir pressure $\mathrm{P}_{max} = 121.5$ bars. This value can be overpassed with the probability $p_{max} = 3.93 \times 10^{-7}$. Now, we will use the adaptive refinement algorithm in order to improve the original model and to reevaluate the failure probability. The subset simulation parameters are $\alpha_0 = 0.1$ and $N = 10^4$. The stopping criterion is $\boldsymbol{\varepsilon} = 0.1$ and the number of points to be added at each iteration is $N_{add} = 5$. The algorithm stops after 4 iterations. Table 4.5 gives the estimations computed at every step of the algorithm.

	Step 1	Step 2	Step 3	Step 4
$\widehat{p}_f$	0.008348	0.0008179	0.0006109	0.0005456
δ_{GP}	0.224	1.113	1.248	0.079

Table 4.5: $\widehat{p}_f$ and δ_{GP} estimations.

Figure 4.10 compares the reservoir pressure simulated at the last year of injection. The first 30 values correspond to the original design used in the original GP model. The pink values correspond to the values simulated for the updated design. Obviously, after each iteration the values of the reservoir pressure become higher and therefore closer to the fracture pressure (or limit-state surface). We assume that the final estimate $\widehat{p}_f = 5.4 \times 10^{-4}$ is the relevant estimation. This estimation can be subsequently improved by changing the stopping criterion, for example taking $\boldsymbol{\varepsilon} = 0.05$

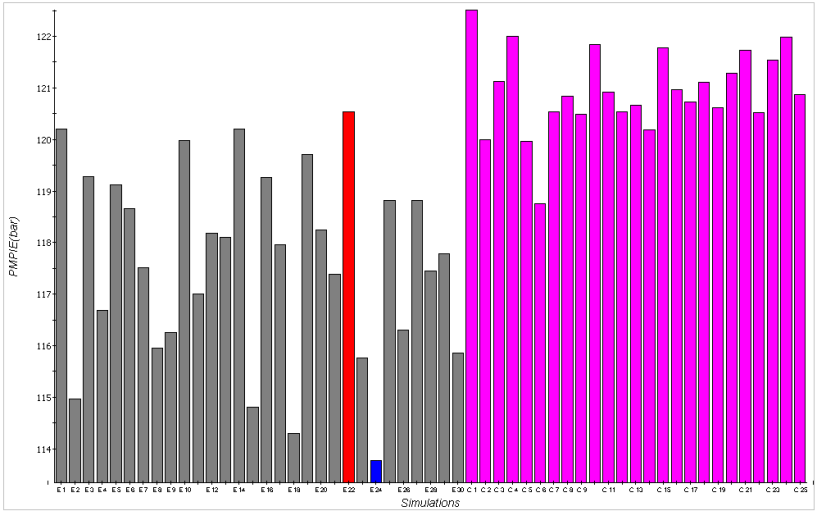


Figure 4.10: Reservoir pressure development with updated data.

4.5 Conclusions

This section covered the reliability analysis in CO_2 storage. First, we have reviewed several common methods to estimate the failure probability of the system. We have compared these methods on an analytical function example. For a linear function, FORM/SORM provide the reliable estimation at the least cost. For more complicated functions, simulation methods usually provide more relevant estimation. Another advantage of the simulation methods is the possibility to measure the error of estimation by computing the coefficient of variation.

Among the presented simulation methods, subset simulation uses the least number of function calls. For this reason, we provided the detailed algorithm and discussed the statistical properties of the estimator. The coefficient of variation of the estimation depends on both the size of the intermediate samples N and the choice of the intermediate failure probability α_0. We have also analyzed the Bayesian approach of subset simulation. Based on the analysis of the posterior distribution, we propose to choose the intermediate failure probability $\alpha_0 \in [0.1, 0.4]$ and $N = 10^3 - 10^4$.

However, subset simulation can become infeasible when the performance function is too expensive to evaluate. For example, this is the case for a dynamic reservoir simulator. One simulator run may take from few minutes to few hours or even days. The approach to reliability analysis based on GP response surface modelling can significantly decrease the required CPU time. The reservoir simulator approximation is computed using a limited number of the simulator runs. Therefore, to estimate the failure probability it is highly important that the model is accurate in the failure region and close to the limit-state surface. Classical experimental design techniques aim to cover uniformly the input domain, whereas we are interested in the information near the failure region. This leads to the introduction of an adaptive refinement technique. We propose a new adaptive refinement methodology. It combines GP response surface modelling and subset simulation algorithm. At each iteration of the subset simulation algorithm, we choose points among the generated sample and update the GP model with the selected points. The main criterion is to reduce the coefficient of variation of the updated GP model.

This method demonstrates promising results. We have tested the method with a known analytical function and a CO_2 storage reservoir case. For an analytical function, the results provided by an updated GP model are sufficiently close to the

reference estimation. The main advantage of the method is a low CPU time. For the analytical example, GP model involves only 20 additional function calls. It is comparable with FORM and SORM approximation methods. However, it provides more robust estimation. For the CO_2 storage reservoir case, the estimations of the failure probability changes from 0 to 4.2×10^{-4}. These estimations could even be improved by decreasing the stopping threshold ε. By this adaptive refinement, we can estimate the probability of leakage before and after the CO_2 injection well placement.

The stopping threshold ε for the coefficient of variation can be chosen depending on the affordable number of additional simulator runs. The lower value of ε involves more iteration steps. In practice, it was noticed that the value $\varepsilon = 0.1$ is sufficient for a reliable estimation. For lower values of ε, the number of additional simulation increases. Whereas, the difference at the failure probability estimation at every iteration is insignificant.

As soon as we know how to generate a reliable GP model for estimation of failure probability, it is important to analyze what are the parameters that mostly affect the failure probability. We will cover the sensitivity analysis of failure probability in the following chapter.

Chapter 5

Sensitivity analysis for failure probability

Contents

5.1 Introduction

For complex systems, such as CO_2 reservoir storage, it is important to evaluate the probability of failure p_f. We discussed in the previous chapter how to effectively

estimate this value for a complex dynamic simulator. In this chapter, we will discuss sensitivity analysis for the failure probability. This is an essential part of any reliability analysis. It highlights the most significant input variables that contribute mostly to the variability of the output. Sensitivity analysis can help in answering the question: which input variable is most influential to the failure probability of the system? For which random input a change in the original probability distribution can improve the system reliability?

Let us consider the vector of uncertain input parameters $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ and the output function $F(\cdot) : \mathbb{R}^d \to \mathbb{R}$. We assume that the input parameters are independent random variables with the joint density $f_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{d} f_{x_i}(x_i)$. Hence, every input variable has marginal density f_{x_i}. The objective of the sensitivity analysis is to identify the most influential parameters $\mathbf{x}$ in terms of the variability of the output $F(\mathbf{x})$. As discussed in previous chapters, this means to rank the input parameters and to identify for which of them a small variation implies a large variation of the output.

The widely used methods for sensitivity analysis are based on a variance decomposition of the output. Sobol' [1993] introduced the variability measure expressed by the so-called Sobol' sensitivity indices. Knowing the probability distribution of the input parameters and the output, we can define sensitivity indices for each of the input variables. It is expressed as a ratio of the variance caused by a given input on the total output variance. We can also measure the interaction effect of two input variables as the proportion of the total variance of the output that is due to these inputs. Sobol [2001] proposed the use of Monte Carlo sampling to estimate Sobol indices. Nowadays, numerous methods are available for efficient estimation of the sensitivity indices both for analytical deterministic functions or involving response surface modelling techniques. However, when dealing with a failure probability, the function of interest is the expectation of an indicator function. Generally, the failure probability is of very low order. For these reasons, variance decomposition may provide irrelevant indices estimation. In addition, it requires numerous function evaluations that is not always affordable.

There have been a few attempts to develop well suited sensitivity analysis methods for a failure probability. First, as complementary results of the First Order Reliability Method, sensitivity to the distribution of the input parameters can be obtained. Sensitivity is expressed as the partial derivative of the reliability index $\boldsymbol{\beta}$ [Lemaire et al., 2009]. Another approach was proposed by Morio [2011]. He uses the

variance decomposition and Sobol' sensitivity indices to study the rate of change in the failure probability due to the changes of the input distribution density parameters. Borgonovo et al. [2011] suggested moment independent importance measures in the reliability analysis. This measure does not involve the variance. For a fixed variable x_i it quantifies the effect of knowing x_i by computing the L_1 norm between the unconditional joint density $f_{\mathbf{x}}(\cdot)$ and conditional density $f_{\mathbf{x}|x_i}(\cdot)$.

In this chapter, we present a novel moment independent approach for sensitivity analysis of a failure probability. We develop a methodology for estimating the influence of the input variables by perturbing its original probability density function $f_{\mathbf{x}}(\mathbf{x})$. In particular, we estimate the effect of the perturbation on the value of the failure probability p_f. Here, we propose to distinguish classes of the distributions by their supports. For the case of a bounded support, such as the uniform or the triangular distributions, the main source of uncertainties is about the boundaries of the support. On the other hand, in the case of infinite support, such as the normal or the log-normal distributions, the main source of uncertainties comes from the distribution parameters, such as mean and variance. The main advantage of the proposed methodology is the efficiency in terms of CPU time. In order to estimate the sensitivity indices for all the input variables, we only use one sample $\mathbf{X}^N = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \overset{i.i.d.}{\sim} f_{\mathbf{x}}(\cdot)$. The performance function is evaluated only once on the given sample.

This chapter is organized as follows. First, we introduce the proposed density perturbation for the distributions with an unbounded support. Then, we precisely discuss the method for different families of distributions. Later, we introduce the technique to calculate the perturbed failure probability using the same sample $\mathbf{X}^N$. It is based on an inverse technique of the importance sampling. After that, we present the formulation for the moment independent sensitivity indices. We also study statistical properties of the presented indices. Finally, we provide analytical and CO_2 storage reservoir case examples.

5.2 Density perturbation for different probability distributions

In this chapter, we keep the same notation as before: $g(\mathbf{x})$ is the performance function of the system, $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ is a set of independent input variables with joint

density $f_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{d} f_{x_i}(x_i)$. The failure probability is expressed as:

$$p_f = \mathrm{P}(g(\mathbf{x}) \leq 0) = \mathbb{E}_{f_{\mathbf{x}}}\left[\mathbf{I}_{g(\mathbf{x})\leq 0}\right] = \int_{\Omega_f} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x},$$

where $\Omega_f = \{\mathbf{x} \in \Omega : g(\mathbf{x}) \leq 0\}$ is a failure region.

In general, the density $f_{\mathbf{x}}(\cdot)$ is built using some engineering restrictions. In this section, we study how a perturbation on the original probability density $f_{\mathbf{x}}(\cdot)$ affects the failure probability of the system p_f.

This work is inspired by Lemaitre et al. [2012]. Originally the method was applicable only for Gaussian distributions [Lemaitre and Arnaud, 2011]. Here, the method is extended to more general distributions of the input variables. The objective of this study is to estimate the influence of the input random variables from the vector $\mathbf{x}$ to the failure probability of the system p_f. The proposed approach consists in perturbing the original density for a given fixed variable x_i while keeping constant the probability density functions for all other variables $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$. Then, a new failure probability is computed. If this new value $p_{i\delta}$ differs significantly from the original one p_f, it means that this selected input variable x_i is influential. Conversely, if the new failure probability $p_{i\delta}$ is close to p_f, then the input x_i has low influence on the failure probability.

5.2.1 Density perturbation

To define the probability density perturbation, we use some ideas from the information theory. The main idea is to perform a density perturbation that can be applicable to any probability distribution. We select the Kullback-Leibler (KL) divergence as a measure of perturbations. KL divergence quantifies the "closeness" of two probability distribution P and Q. Suppose that P and Q are continuous probability distributions with densities $p(\mathbf{x})$ and $q(\mathbf{x})$. Then, the KL divergence between P and Q is given by:

$$\mathrm{D}_{\mathrm{KL}}(P, Q) = \int_{-\infty}^{\infty} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \tag{5.1}$$

Let us denote for the variable x_i the new (perturbed) density as $f_{x_{i\delta}}(\cdot)$, then $f_{x_{i\delta}}(\cdot)$ is selected in such way that:

$$\mathrm{D}_{\mathrm{KL}}(f_{x_i}, f_{x_{i\delta}}) = \delta. \tag{5.2}$$

Possible values of the perturbation δ may be restricted by inequalities for Kullback-Leibler divergence [Dragomir and Glušcević, 2000]. Let us define the function $r(\mathbf{x}) := \frac{p(\mathbf{x})}{q(\mathbf{x})}, x \in \Omega \subset \mathbb{R}^d$ and assume that $0 < r < r(\mathbf{x}) < R$ for all $\mathbf{x} \in \Omega \subset \mathbb{R}^d$. We have:

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}(p, q) &\geq 0 \\ \mathrm{D}_{\mathrm{KL}}(p, q) &\leq \frac{(R-r)^2}{4rR} = \delta_{max} \end{aligned}$$

According to these inequalities, we choose $\delta \in [0, \delta_{max}]$, where $\delta_{max} = \frac{(R-r)^2}{4rR}$ can be estimated precisely for the selected family of distributions F.

For some given $\delta > 0$ and the original probability density function $f_{x_i}(\cdot)$, we propose to restrict our choice of perturbation among the following class of densities:

$$f_{x_{i\tau}}(x) = \exp(\tau x - \psi(\tau)) f_{x_i}(x). \tag{5.3}$$

This density class minimizes the KL divergence between the original and the perturbated distributions [Lemaitre and Sergienko et. al, 2012]. Here, τ is a constant depending on δ (it is chosen under the condition (5.2)). The function $\psi(\tau)$ is a normalization function and it may be expressed as:

$$\psi(\tau) = \log \left[\int_{-\infty}^{\infty} \exp(\tau t) f_{x_i}(t) dt \right].$$

It is the cumulant generating function for the perturbed probability distribution. We set $\mathrm{M}_k := \int_{-\infty}^{\infty} (t-\mu)^k f_{x_{i\tau}}(t) dt$ the k^{th} central moment of the distribution $F_{i\tau}$, where $\mathrm{M}_1 = 0$ and $\mathrm{M}_2 = \sigma^2$ (μ and σ^2 is the distribution mean and variance). Then, the following equalities hold for $\psi(\tau)$:

$$\begin{aligned} \psi(0) &= 0 \\ \psi'(0) &= \mu \\ \psi''(0) &= \mathrm{M}_2 \end{aligned} \tag{5.4}$$

When the KL divergence between the original and the perturbed density is set to the value δ, we obtain the following expression for τ:

$$\begin{aligned} \mathrm{KL}(f_{x_{i\tau}}, f_{x_i}) &= \int_{-\infty}^{\infty} f_{x_{i\tau}}(t) \log \frac{f_{x_{i\tau}}(t)}{f_{x_i}(t)} dt = \int_{-\infty}^{\infty} f_{x_{i\tau}}(t) \left(\tau t - \psi(\tau)\right) dt \quad (5.5) \\ &= \tau\psi'(\tau) - \psi(\tau). \end{aligned}$$

Therefore, τ should satisfy the equation:

$$\tau\psi'(\tau) - \psi(\tau) = \delta. \quad (5.6)$$

Let $\tau^* = \tau(\delta)$ be a solution of (5.6). We use this parameter in order to define the perturbed density modification $f_{x_{i\tau}}(\cdot)$ defined by equation (5.3).

Let us consider the function:

$$G(\tau) = \tau\psi'(\tau) - \psi(\tau) - \delta.$$

This function has a global minimum at $\tau = 0$ and $G(0) = -\delta < 0$ for $\delta > 0$. We show in Appendix B.1.2 that $G(\tau)$ has not more than two zeros $\tau_1 < 0$ and $\tau_2 > 0$ (depending on the functional domain). Therefore, equation (5.6) has exactly two solutions τ_1 and τ_2.

For every fixed level of δ, we can study two effects of the perturbation (5.3). We denote the perturbed densities corresponding to the solutions of equation (5.6) by $f_{x_{i\tau_1}}$ and $f_{x_{i\tau_2}}$. Then the joint perturbed probability density is expressed as:

$$f_{\mathbf{x}_{i\tau_j}}(\mathbf{x}) = f_{x_{i\tau_j}} \prod_{k=1, k\neq i}^{d} f_{x_k}(x_k), \quad j = 1, 2.$$

The corresponding value of the perturbed failure probability $p_{i\delta_j}$ $(j = 1, 2)$ can be computed as the following integral:

$$p_{i\delta_j} = \mathbb{E}_{f_{\mathbf{x}_{i\tau_j}}} \left[\mathbf{I}_{g(\mathbf{x})<0}\right] = \int \mathbf{I}_{g(\mathbf{x})<0} f_{\mathbf{x}_{i\tau_j}} d\mathbf{x}, \quad j = 1, 2. \quad (5.7)$$

In the same way, we can study the interaction effect by perturbing two variables x_i and x_j at the same time by δ_1 and δ_2 respectively. Suppose that $\tau(\delta_1) = (\tau_{1\delta_1}, \tau_{2\delta_1})$ and $\tau(\delta_2) = (\tau_{1\delta_2}, \tau_{2\delta_2})$ are the solutions of equation (5.6), where δ_1 and δ_2 are the perturbations of KL divergence (5.2) for the variables x_i and x_j, respectively. The

new joint probability density function is:

$$f_{\mathbf{x}_{ij,\tau(\delta_1),\tau(\delta_2)}}(\mathbf{x}) = f_{x_{i\tau(\delta_1)}} f_{x_{j\tau(\delta_2)}} \prod_{k=1,k\neq i,j}^{d} f_{x_k}(x_k).$$

The corresponding value of the perturbed failure probability is estimated in the same way by putting in (5.7) the new joint probability density $f_{\mathbf{x}_{ij,\tau(\delta_1),\tau(\delta_2)}}(\cdot)$.

Later, we introduce a method to efficiently estimate the perturbed failure probability $p_{i\delta}$ using the same Monte Carlo sample as for the original estimation of the failure probability p_f. The method allows to compute the perturbed failure probability with low additional CPU cost. It is applicable to the estimation of the interaction effects as well. Before that, we first study the effect of the perturbation for different families of probability distributions.

5.2.1.1 Normal distribution

When $x_i \sim \mathcal{N}(\mu, \sigma^2)$, the density $f_{x_i}(x)$ is:

$$f_{x_i}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Then, the cumulant generating function is:

$$\psi(\tau) = \mu\tau + \frac{\tau^2\sigma^2}{2}, \quad \tau \in \mathbb{R}.$$

By substituting the expression for $\psi(\tau)$, the new perturbed density is given by:

$$f_{x_{i\tau}} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-(\mu+\tau\sigma^2))^2}{2\sigma^2}\right).$$

This means that the variable $x_{i\tau}$ is normally distributed with mean $\mu + \tau\sigma^2$ and variance σ^2. The expression for τ can be computed by fixing the Kullback-Leibler divergence (5.2) at the level $\delta > 0$. We have to solve the following equation for τ:

$$\mathrm{D}_{\mathrm{KL}}(f_{x_{i\tau}}, f_{x_i}) = \psi'(\tau)\tau - \psi(\tau) = \delta.$$

There are two solutions for any positive fixed value of δ:

$$\tau_1 = -\frac{\sqrt{2\delta}}{\sigma} \quad \text{and} \quad \tau_2 = \frac{\sqrt{2\delta}}{\sigma}.$$

Hence, the perturbation of the original normal distribution of the variable x_i leads to a normal distribution with mean translated by $\tau\sigma^2$. The two possible solutions for τ mean either positive or negative translation. Therefore, by conducting sensitivity analysis we can study how the mean translation affects the failure probability.

The case of the truncated Gaussian distribution is presented in Appendix B.4.

5.2.1.2 Lognormal distribution

For a lognormal distribution $\log \mathcal{N}(\mu, \sigma^2)$, the density has the following form:

$$f_{x_i}(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\log x - \mu)^2}{2\sigma^2}\right)$$

In order to solve equation (5.3) we propose to look for a density having the form:

$$f_{x_{i\tau}(x)} = \exp(\tau \log x - \psi(\tau)) f_{x_i}(x).$$

In this case, all computations are identical to the one performed for the normal distribution. We then obtain the same expression for the normalization function $\psi(\tau)$ as for the previous case of the normal distribution. The modified density is:

$$f_{x_{i\tau}} = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\log x - (\mu + \tau\sigma^2))^2}{2\sigma^2}\right).$$

The values for τ are the same as for the normal distribution:

$$\tau_1 = -\frac{\sqrt{2\delta}}{\sigma} \quad \text{and} \quad \tau_2 = \frac{\sqrt{2\delta}}{\sigma}.$$

This density perturbation corresponds to positive or negative translation of the location parameter μ of the distribution.

5.2.1.3 Exponential distribution

Assume that $x_i \sim \text{Exp}(\lambda), \quad \lambda > 0$, which is the exponential distribution with the parameter λ. The density is given by:

$$f_{x_i}(x) = \lambda \mathrm{e}^{-\lambda x}.$$

The expression for $\psi(\tau)$ can be rewritten as:

$$\psi(\tau) = \log\left(\frac{\lambda}{\lambda - \tau}\right), \quad \tau < \lambda.$$

For $\delta > 0$ we can define the equation for $\tau(\delta)$:

$$\frac{\tau}{\lambda - \tau} - \log\frac{\lambda}{\lambda - \tau} = \delta, \quad \tau < \lambda,$$

The two possible solutions of this equation are expressed with the W Lambert function. We denote by $W_0(x)$ the upper branch of the Lambert function on the interval $[-1/\mathrm{e}, 0]$ and by $W_{-1}(x)$ the lower branch on the same interval. Then the solutions for $\tau(\delta)$ have the following form:

$$\tau_1(\delta) = \frac{\lambda\left(W_{-1}(-\mathrm{e}^{-1-\delta}) + 1\right)}{W_{-1}(-\mathrm{e}^{-1-\delta})} \quad \text{and} \quad \tau_2(\delta) = \frac{\lambda\left(W_0(-\mathrm{e}^{-1-\delta}) + 1\right)}{W_0(-\mathrm{e}^{-1-\delta})}.$$

Knowing the value for $\tau^* = \tau_1, \tau_2$, the new perturbed density will take the following form:

$$f_{x_{i\tau}} = \exp(\tau^* x)\exp(-\lambda x)(\lambda - \tau^*) = (\lambda - \tau^*)\exp\left(-\left(\lambda - \tau^*\right)x\right).$$

Therefore, $x_{i\tau} \sim \text{Exp}(\lambda - \tau^*)$ has the exponential distribution with parameter $(\lambda - \tau^*)$.

5.2.1.4 Poisson distribution

Assume that variable $x_i \sim \text{Pois}(\lambda), \lambda > 0$, which is Poisson distribution with parameter λ. The density is given by:

$$f_{x_i}(k) = \frac{\lambda^k \mathrm{e}^{-\lambda}}{k!}, \quad k \in \mathbb{N}.$$

Thus, the normalization function takes the form:

$$\psi(\tau) = \lambda(\mathrm{e}^{\tau} - 1), \quad \tau \in \mathbb{R}.$$

The equation for $\tau(\delta)$ becomes:

$$\lambda \mathrm{e}^{\tau}(\tau - 1) + \lambda = \delta.$$

The two possible solutions of this equation can also be expressed with the W Lambert function. The solutions for $\tau(\delta)$ have the following form:

$$\tau_1(\delta) = W_{-1}\left(-\frac{\lambda - \delta}{\mathrm{e}\lambda}\right) + 1 \quad \text{and} \quad \tau_2(\delta) = W_0\left(-\frac{\lambda - \delta}{\mathrm{e}\lambda}\right) + 1.$$

Knowing the value for $\tau^* = \tau_1, \tau_2$, we can rewrite the perturbed density $f_{x_{i\tau}}$ as:

$$f_{x_{i\tau}} = \exp(\tau^* k - \psi(\tau^*)) \frac{\mathrm{e}^{-\lambda}\lambda^k}{k!} = \frac{\left(\lambda \mathrm{e}^{\tau^*}\right)^k}{k!} \exp(-\lambda \mathrm{e}^{\tau^*}).$$

Therefore, the new perturbed variable $x_{i\tau} \sim \mathrm{Pois}(\lambda \mathrm{e}^{\tau^*})$ follows the Poisson distribution with parameter $\lambda \mathrm{e}^{\tau^*}$.

5.2.1.5 Uniform distribution

Assume that $x_i \sim \mathrm{U}[a, b]$, the uniform distribution on the interval $[a, b]$. The density is:

$$f_{x_i}(x) = \frac{1}{b - a} \mathbf{I}_{x \in [a,b]}(x), (b > a).$$

The normalization function becomes:

$$\psi(\tau) = \log\left(\frac{\mathrm{e}^{\tau b} - \mathrm{e}^{\tau a}}{\tau(b - a)}\right), \tau \in \mathbb{R}. \tag{5.8}$$

Then the equation for $\tau(\delta)$ is:

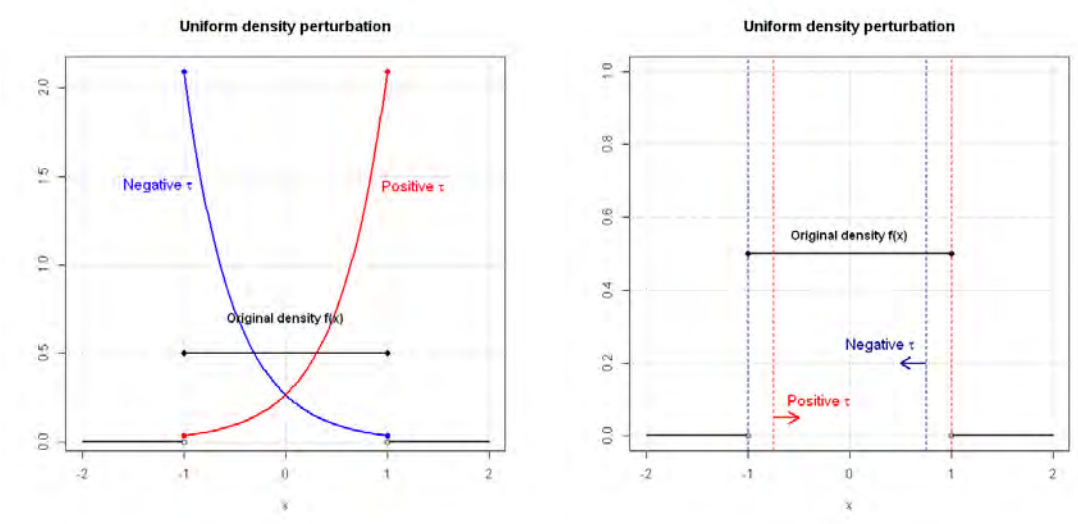
$$\frac{\tau b \mathrm{e}^{\tau b} - \tau a \mathrm{e}^{\tau a}}{\mathrm{e}^{\tau b} - \mathrm{e}^{\tau a}} - 1 - \log\left(\frac{\mathrm{e}^{\tau b} - \mathrm{e}^{\tau a}}{\tau(b - a)}\right) = \delta.$$

This equation has no explicit solutions for τ. The solutions can be found using a numerical solver. Suppose that $\tau^* = \tau(\delta)$ is a solution of equation (5.8). Then, the

perturbed density is:

$$f_{x_{i\tau}} = \frac{\tau^* \mathrm{e}^{\tau^* x}}{\mathrm{e}^{\tau^* b} - \mathrm{e}^{\tau^* a}} \mathbf{I}_{x \in [a,b]}(x).$$

Therefore, the new perturbed variable $x_{i\tau}$ is no longer uniform on $[a, b]$. This density modification for $a = -1, b = 1$ and $\delta = 0.5$ is displayed in Figure 5.1(a).



(a) Exponential density modification

(b) Boundaries perturbation

Figure 5.1: Uniform density perturbation.

While working with uncertain parameters defined on a compact support, the main source of uncertainty is on the boundaries of the support. For such distributions with a bounded support, we propose to apply another density perturbation. The idea consists in perturbing the original boundaries by $\tau = \pm\delta$. In the same way as with infinite support we consider the effect of positive or negative perturbation. For example, consider $x_i \sim \mathrm{U}[a, b]$ to be uniformly distributed on the interval $[a, b]$. Then, in order to stay inside the support the perturbed random variable $x_{i\tau}$ is uniformly distributed either on $U[a + \delta, b]$ or on $U[a, b - \delta]$. The corresponding density for perturbed uniform distribution can be expressed as:

$$f_\tau = \frac{1}{b - a - \delta} \mathbf{I}_{x \in [a+\delta, b]}(x) \quad \text{or} \quad f_\tau = \frac{1}{b - a - \delta} \mathbf{I}_{x \in [a, b-\delta]}(x).$$

The same perturbation may be applied to triangular or trapezoidal distributions. It can be also applied to the truncated Gaussian distribution if one is interested about the boundary influence on the failure probability p_f. In this case the density

function should be corrected for the new boundaries.

To summarize this section, we have covered many classical families of distributions. In the next section, we explain how to efficiently estimate a perturbed failure probability p_{f_δ}.

5.3 Importance sampling and sensitivity analysis

As discussed in Chapter 4, Monte Carlo sampling is the most popular simulation methods to estimate the failure probability. We consider the input parameters space $\Omega \in \mathbb{R}^d$. Suppose that all the input parameters are independent and that $f_{\mathbf{x}}(\mathbf{x}) = \prod_{k=1}^{d} f_{x_k}(x_k)$ is the joint density of input parameters. Let $\mathbf{X}^N = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \overset{i.i.d.}{\sim} f_{\mathbf{x}}(\cdot)$ be a sample of size N. Then, the estimation of the failure probability p_f is given by:

$$\widehat{p}_f = \frac{1}{N} \sum_{k=1}^{N} \mathbf{I}_{g(\mathbf{x}_k)\leq 0}. \tag{5.9}$$

Now, assume that $f_{x_{i\tau}}$ is a new perturbed density for the input parameter x_i. Then the new joint density is $f_{\mathbf{x}_\tau} = \prod f_{x_1} \cdots f_{x_{i-1}} f_{x_{i\tau}} f_{x_{i+1}} \cdots f_{x_d} = f_{x_{i\tau}} \prod_{k=1, k\neq i}^{d} f_{x_i}(x_i)$. The corresponding failure probability $p_{i\delta}$ is defined as an expectation of the indicator function for this new distribution:

$$p_{i\delta} = \mathbb{E}_{f_{\mathbf{x}_{i\tau}}} \left[\mathbf{I}_{g(\mathbf{x})\leq 0}\right] = \int_{\Omega} \mathbf{I}_{g(\mathbf{x})\leq 0} f_{\mathbf{x}_{i\tau}}(\mathbf{x}) d\mathbf{x}$$

Here, we propose to apply the technique used in the Importance Sampling (IS) simulation method. We multiply the integrand function by $\mathbf{1} = \frac{f_{\mathbf{x}}(\mathbf{x})}{f_{\mathbf{x}}(\mathbf{x})}$. Both density functions $f_{\mathbf{x}}(\mathbf{x})$ and $f_{x_{i\tau}}$ are the products of the density functions of the independent variables $\mathbf{x} = (x_1, \dots, x_d) \in \Omega \subset \mathbb{R}^d$ and only differ for the variable x_i. Therefore, we obtain:

$$p_{i\delta} = \int_{\Omega} \mathbf{I}_{g(\mathbf{x})\leq 0} f_{\mathbf{x}_{i\tau}}(\mathbf{x}) d\mathbf{x} = \int_{\Omega} \mathbf{I}_{g(\mathbf{x})\leq 0} \frac{f_{x_{i\tau}}(x_i)}{f_{x_i}(x_i)} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = \mathbb{E}_{f_{\mathbf{x}}} \left[\mathbf{I}_{g(\mathbf{x})\leq 0} \frac{f_{x_{i\tau}}(x_i)}{f_{x_i}(x_i)}\right].$$

By doing this, we do not need to throw a new sample according to the density function $f_{\mathbf{x}_{i\tau}}(\mathbf{x})$. We are working in the same probability space integrating the function $\left[\mathbf{I}_{g(\mathbf{x})\leq 0} \frac{f_{x_{i\tau}}(x_i)}{f_{x_i}(x_i)}\right]$. In addition, to estimate the perturbed failure probability $p_{i\delta}$ we keep the same sample points from the failure region: $\mathbf{X}_f^N = \{\mathbf{x} \in \mathbf{X}^N : g(\mathbf{x}) \leq 0\}$ that provide non-zero values of the indicator function $\mathbf{I}_{g(\mathbf{x})\leq 0}$. Consequently, we

do not need to reevaluate the function and we do not need any additional function calls. The estimation of the failure probability for perturbed density is expressed as:

$$\widehat{p}_{i\delta} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{I}_{g(\mathbf{x}_k)\leq 0} \frac{f_{x_{i\tau}}(\mathbf{x}_{ki})}{f_{x_i}(\mathbf{x}_{ki})}. \tag{5.10}$$

If we are interested in the interaction effects, we perturb the probability densities for the variables x_i and x_j simultaneously. Then, the new joint density is:

$$f_{\mathbf{x}_{ij,\tau(\delta_1),\tau(\delta_2)}}(\mathbf{x}) = f_{x_{i\tau(\delta_1)}} f_{x_{j\tau(\delta_2)}} \prod_{k=1,k\neq i,j}^{d} f_{x_k}(x_k).$$

Therefore, in this case the new failure probability can be estimated by:

$$\widehat{p}_{f_{\tau(\delta_1),\tau(\delta_2)}} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{I}_{g(\mathbf{x})\leq 0} \frac{f_{x_{i\tau(\delta_1)}}(\mathbf{x}_{ki}) f_{x_{j\tau(\delta_2)}}(\mathbf{x}_{kj})}{f_{x_i}(\mathbf{x}_{ki}) f_{x_j}(\mathbf{x}_{kj})}.$$

This new value of the failure probability describes the interaction effects of two variables x_i and x_j on the failure probability p_f.

The proposed reliability sensitivity analysis is based on the analysis of the value of the perturbed failure probability. The higher the effect of the perturbation by the same value of δ is, the more influential the variable to the failure probability is. The new failure probability itself can be a sensitivity measure because it makes possible to determine what variables has the highest impact. However, we propose another sensitivity measure. It allows to clearly differentiate the input variables by the level of their influence on the failure probability. We will consider different possible formulations of sensitivity indices in the next section.

5.4 Sensitivity indices formulations

Here, we analyze different possible sensitivity indices based on the previous density perturbation. These indices have different asymptotic statistical properties. In this section, we investigate three possible sensitivity indices. We detail and compare these suggestions on the same analytical function afterwards.

5.4.1 Basic indices

The basic index is based on the difference $p_{i\delta} - p_f$ with the original failure probability p_f. It is expressed as the ratio:

$$S_{i\delta} = \frac{p_{i\delta} - p_f}{p_f} \tag{5.11}$$

The support of these indices is $[-1, +\infty)$. A negative value of $S_{i\delta}$ means that the proposed density modification reduces the failure probability. Conversely, a positive value of this index means an increase in the failure probability. Zero value of $S_{i\delta}$ means that the variable x_i has no impact on the failure probability.

In practice, p_f is estimated by $\widehat{p}_f$ with the Monte Carlo simulation method. In the same way, according to (5.10) $p_{i\delta}$ is estimated by $\widehat{p}_{i\delta}$. The estimator of the indices $\widehat{S}_{i\delta}$ thus can be expressed as:

$$\widehat{S}_{i\delta} = \frac{\widehat{p}_{i\delta} - \widehat{p}_f}{\widehat{p}_f}.$$

This estimation provides an asymptotic unbiased estimation of $S_{i\delta}$. Moreover, according to the Central Limit Theorem (CLT):

$$\frac{1}{\sqrt{\mathrm{VAR}_{f_{\mathbf{x}}}\left[\widehat{S}_{i\delta}\right]}} \left(\widehat{S}_{i\delta} - S_{i\delta}\right) \overset{N\to\infty}{\to} \mathcal{N}(0,1)$$

The proof and an asymptotic expression of the variance can be found in Appendix B.2.2.

The above basic indices are computed separately for every fixed value of the density perturbation δ. It allows to study the effect of perturbation by varying the value of δ. It can help in reliability design optimization by adjusting the distribution parameters of the input variables $\mathbf{x} = (x_1, \ldots, x_d)$ in order to achieve the lowest failure probability p_f.

5.4.2 Symmetric indices

The previous index is bounded below by -1. Therefore, in some cases it may be difficult to use this index to rank the parameters by their influence on the failure probability. For some density perturbations the indices for some variables may have

similar behavior around the value -1. For this reason, we propose an alternative index. For the input variable x_i and a perturbation parameter $\delta > 0$ we set:

$$\tilde{S}_{i\delta} = \left[\frac{p_{i\delta}}{p_f} - 1\right] \mathbf{I}_{p_{i\delta} > p_f} + \left[1 - \frac{p_f}{p_{i\delta}}\right] \mathbf{I}_{p_{i\delta} < p_f} = \frac{p_{i\delta} - p_f}{p_f \mathbf{I}_{p_{i\delta} > p_f} + p_{i\delta} \mathbf{I}_{p_{i\delta} < p_f}}.$$

The value of this index also equals to 0 when the perturbed probability $p_{f_{i\tau}}$ does not differ from the original value p_f. The negative values of such indices occur for decreasing failure probability, whereas positive values occur for an increasing value of the failure probability.

To estimate these indices we will also use the Monte Carlo estimations $\widehat{p}_f$ and $\widehat{p}_{i\delta}$. More details on these indices can be found in Appendix C.1 and in Lemaitre and Sergienko et. al, 2012, where the statistical properties of the estimator are studied.

5.4.3 Variance based sensitivity indices

Another possible way to measure the influence of the input variables can be based on the variance decomposition.

Here, we assume that the value of the density perturbation δ is a random variable. For example, suppose that $\delta \sim \mathrm{U}[0,1]$ is uniformly distributed on $[0,1]$. Then, we can compute the variance of the perturbed failure probability $p_{i\delta}$ as a function of the random variable δ. Let us denote this value by $\mathrm{VAR}_\delta(p_{i\delta})$. This variance can be used to evaluate the impact of the input variable x_i in the variation of the failure probability. We propose to normalize this measure by the "total variance" of the perturbed failure probability denoted by $\mathrm{VAR}_\delta(p_{\mathbf{x}\delta})$. By the "total variance" we mean the variance of the perturbed failure probability computed by simultaneous perturbing all the variables densities $f_{\mathbf{x}}(\cdot) = \prod_{k=1}^{d} f_{x_i}(\cdot)$. The proposed index is then:

$$\tilde{\tilde{S}}_{\delta_i} = \frac{\mathrm{VAR}_\delta(p_{i\delta})}{\mathrm{VAR}_\delta(p_{\mathbf{x}\delta})}, \tag{5.12}$$

where $p_{i\delta}$ is the failure probability computed for the density f_{x_i} perturbed by δ. The value $p_{\mathbf{x}\delta}$ is the failure probability computed by perturbing all the input variables densities $f_{\mathbf{x}} = \prod_{k=1}^{d} f_{x_i}$ simultaneously by a vector $\delta = (\delta_1, \ldots, \delta_d) \in [0,1]^d$. The ratio (5.12) evaluates the amount of variation caused by perturbing the density f_{x_i} relatively to the total variation caused by perturbing all the variables densities. We show how to estimate the ratio (5.12) with a Monte Carlo sampling of the random

variable δ in Appendix B.5. Here, we will detail the estimation of the probability $p_{\mathbf{x}\delta}$.

To estimate the perturbed failure probability $p_{\mathbf{x}\delta}$, we perturb the vector of the densities of the input variables $\mathbf{x} = (x_1, \ldots, x_d)$ simultaneously by a vector $\delta = (\delta_1, \ldots, \delta_d)$. Assume that $\tau = (\tau_1(\delta_1), \ldots, \tau_d(\delta_d))$ is the corresponding vector of the solutions (5.6). Then, the new perturbed joint density is expressed as:

$$f_{\mathbf{x}\tau} = \prod_{k=1}^{d} f_{k\tau_k}(x_k),$$

and the related perturbed failure probability:

$$p_{\mathbf{x}\delta} = \int_{\Omega} \mathbf{I}_{g(\mathbf{x})<0} f_{\mathbf{x}\tau} d\mathbf{x}.$$

In the same way as with (5.10), the last probability can be estimated by:

$$\widehat{p}_{\mathbf{x}\delta} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{I}_{g(\mathbf{x}_k)<0} \frac{\prod_{l=1}^{d} f_{l\tau_l}(x_l)}{\prod_{l=1}^{d} f_l(x_l)}.$$

With the indices defined by (5.12), we can compare directly the influence of different input variables to the failure probability p_f. However, we should make the *a priori* assumption about the probability distribution of the variable δ. In addition, it is not straightforward to evaluate the statistical properties of this estimator.

5.4.4 Analytical function example

To investigate the previous indices, we will consider a simple linear function of three independent normally distributed variables: $x_1, x_2, x_3 \sim \mathcal{N}(0, 1)$:

$$g(x_1, x_2, x_3) = 3 - 0.1x_1 - 0.5x_2 - 1.0x_3. \tag{5.13}$$

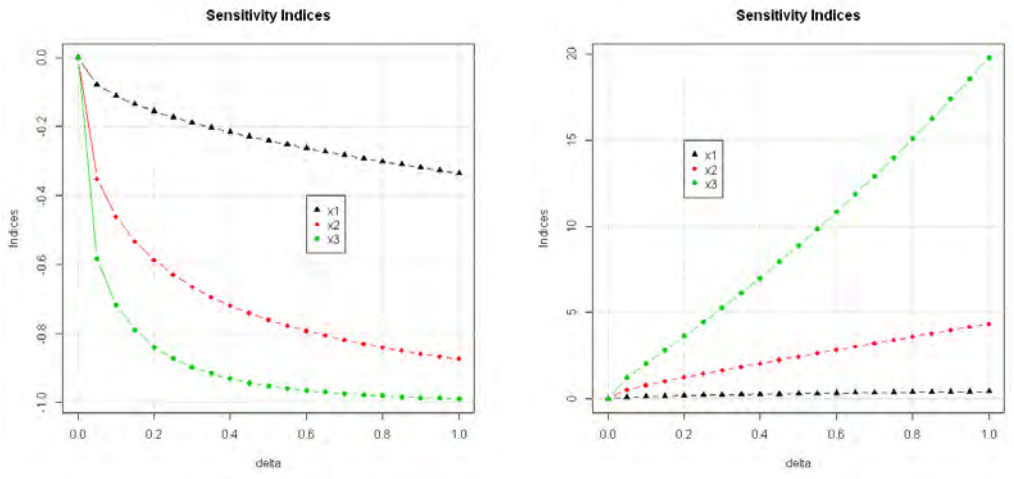
As a linear combination of independent Gaussian random variables, $g(\mathbf{x}) \sim \mathcal{N}(\mu, \sigma)$, where $\mu = 3$ and $\sigma^2 = 0.1^2 + 0.5^2 + 1.0^2 = 1.26$. The original failure probability can be explicitly computed:

$$p_f = 1 - \Phi(\frac{\mu}{\sigma}) = 3.7 \times 10^{-3}.$$

The estimation provided by a Monte Carlo sample of size $N = 10^6$ yields $\widehat{p}_f = 3.69 \times 10^{-3}$. We will use the same Monte Carlo sample to estimate the perturbed failure probabilities $p_{i\delta}$ and the sensitivity indices $S_{i\delta}$.

Basic index

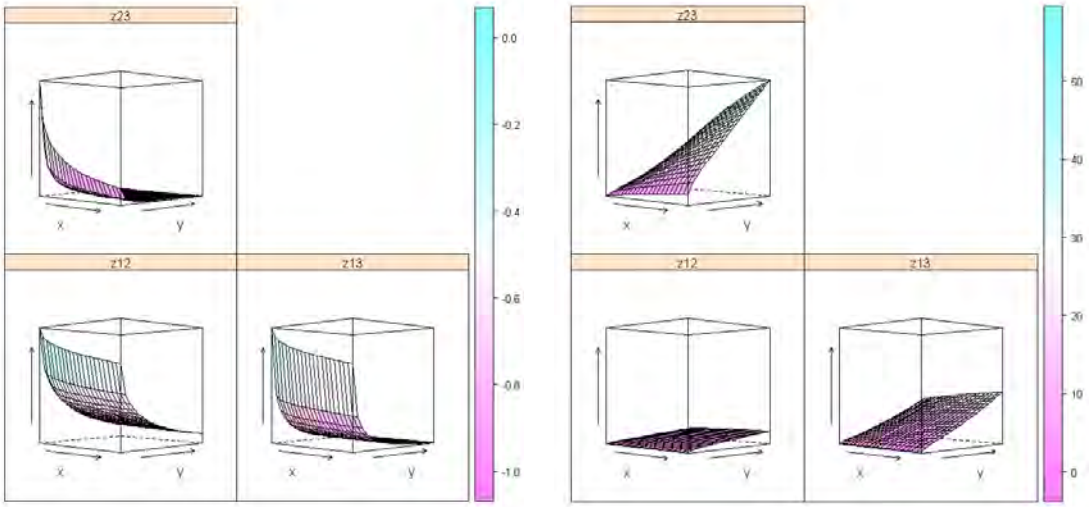
Firstly, we will present the basic sensitivity indices and the related interaction effects. Figure 5.2 depicts sensitivity indices calculated for $\delta \in [0, 1]$ with the positive (5.2(b)) and the negative (5.2(a)) mean shifting. It can be clearly observed that the highest impact on the failure probability is due to the variable x_3 and that the variable x_1 has the lowest impact for both cases of τ. Moreover, the higher the value of δ is, the higher is the influence of the parameters.



(a) Negative τ (b) Positive τ

Figure 5.2: Basic sensitivity indices example.

The interaction effects is computed the same way by perturbing two parameters simultaneously. We denote $p_{ik,\delta_1\delta_2}$ the failure probability obtained by perturbing the distribution of the two variables x_i and x_k at the same time. Therefore, here we have to analyze the surface of perturbations for all possible combinations $(\delta_1, \delta_2) \in [0, 1]^2$.



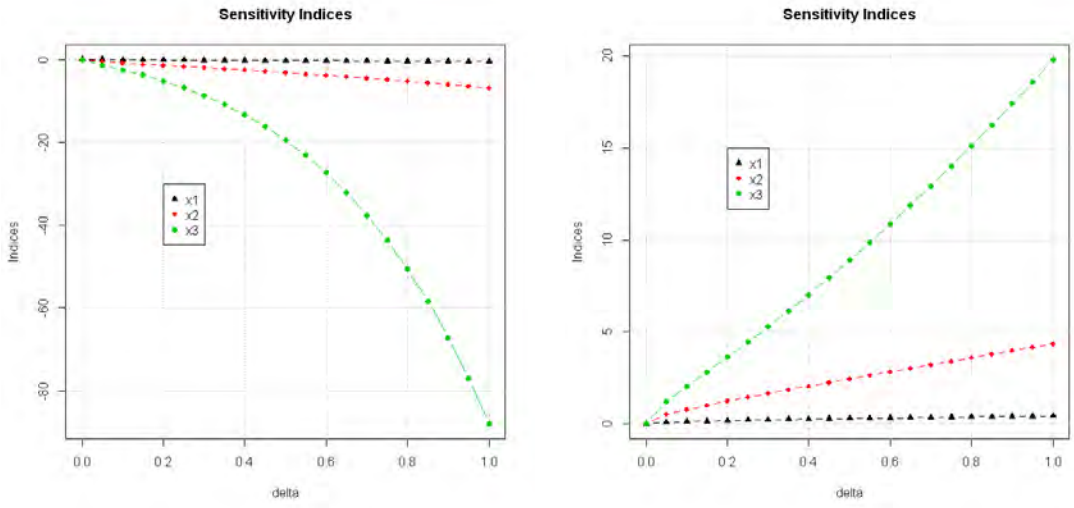
(a) Negative τ (b) Positive τ

Figure 5.3: Basic interaction effect example.

The same results are valid for the interaction effect. The higher the impact of a single variable, the higher the influence of the interaction with other variables. Here, we can see that the highest value of interaction effects corresponds to the couple of variables (x_2, x_3).

Symmetric indices formulation

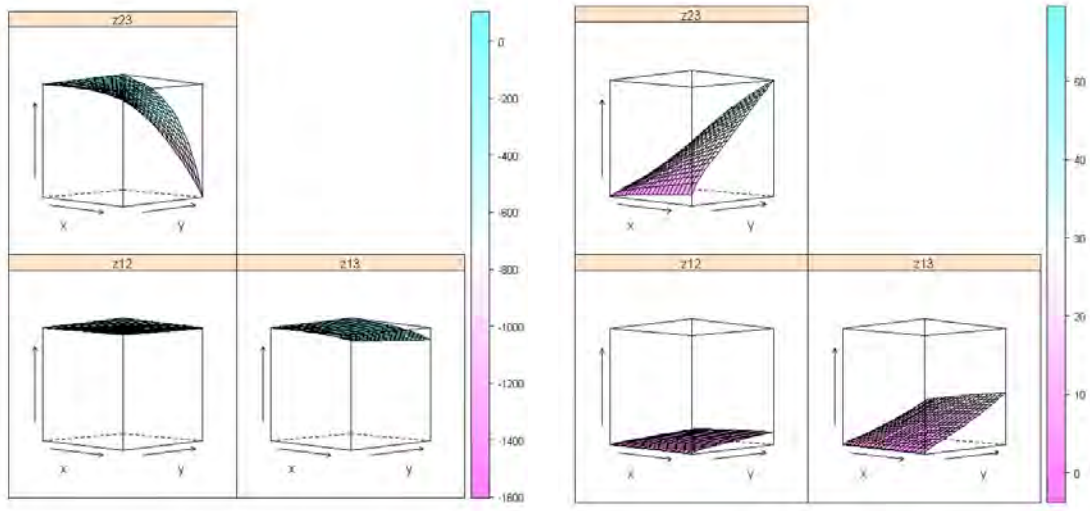
Here, we provide estimations of the symmetric indices formulation. Figure 5.4 displays the computed value of symmetric sensitivity indices for negative (Figure 5.4(a)) and positive (Figure 5.4(b)) values of τ. The results are similar to the ones obtained above. The variable x_3 has the highest influence on the failure probability. However, in this case there is no lower boundary of -1.



(a) Negative τ (b) Positive τ

Figure 5.4: Symmetric sensitivity indices example.

The same results are valid for the estimated interaction effects.



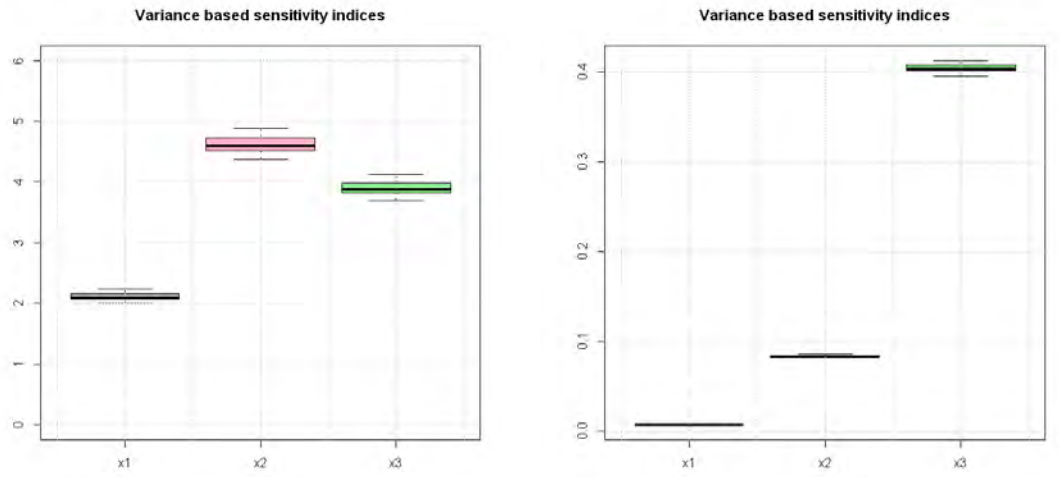
(a) Negative τ (b) Positive τ

Figure 5.5: Symmetric interaction effect example.

Variance based formulation

Here, we present the estimations for the variance based sensitivity indices $S_{i\delta}$. The indices formulated this way provide a single value estimation for a sample of pertur-

bations δ. Here, we present boxplots of indices computed for each of the variables $x_i, i = 1, 2, 3$. The boxplots were generated with 30 indices realizations. Figure 5.6 compares the simulated results for negative (Figure 5.6(a)) and positive (Figure 5.6(b)) values of τ.



(a) Negative τ (b) Positive τ

Figure 5.6: Variance based sensitivity indices

It can be observed that for positive values of τ there is almost no variation in the estimation of the indices. In this case, the variable x_3 shows the highest impact on the failure probability. Whereas for the case of negative values of τ all the variables demonstrate similar values of the indices. The reason of rather high values of indices for negative τ is very low order (-4) of the values of total and partial variances.

5.5 CO_2 storage case example

In this section, we provide an application of the proposed reliability sensitivity analysis to CO_2 storage reservoir case. We study the same reservoir case from the Section 3.3.4. The following input variables are considered in this reservoir case:

Name	Description	Min	Max
PORO	Reservoir Porosity	0.15	0.35
KSAND	Reservoir Permeability	10	300
KRSAND	Water relative permeability end-point	0.5	1.0

Table 5.1: Uncertain parameters.

For sake of clarity we transform the original intervals into $[-1, 1]$. In this section, we assume the truncated standard normal distribution for all the parameters $\mathcal{N}_{[-1,1]}(0, 1)$.

Let us recall the formulation of the performance function. Suppose that $\mathrm{P}_{fracture}$ is a fixed value of the cap rock fracturing pressure. $\mathrm{P}_{reservoir}(\mathbf{x})$ is a function of the reservoir pressure depending on the input parameters configuration $\mathbf{x} \in \Omega \subset \mathbb{R}^d$. Then, the performance function defining the event of the gas leakage can be expressed as:

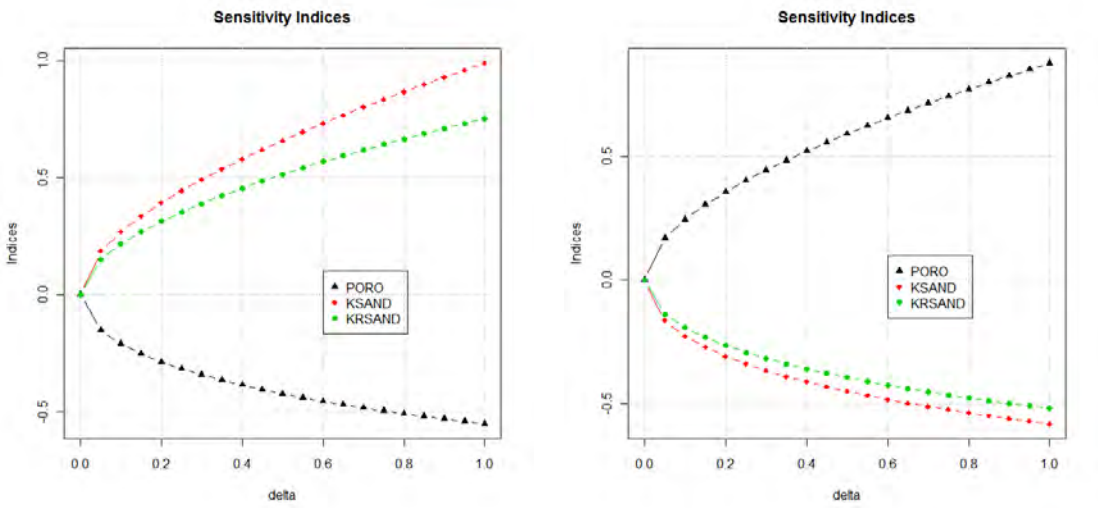
$$g(\mathbf{x}) = \mathrm{P}_{fracture} - \mathrm{P}_{reservoir}(\mathbf{x}).$$

The reservoir pressure $\mathrm{P}_{reservoir}(\mathbf{x})$ is computed with a complex dynamic reservoir simulator. For this reason, we use the GP response surface model approximation $\widehat{P}(\mathbf{x})$. We use the refined GP response surface model obtained in Section 4.4.2. This model provides a reliable estimation of the failure probability p_f.

Under the assumption about the truncated standard normal distribution, the estimation of the failure probability for the refined model is $\widehat{p}_f^{SS} = 2.25 \times 10^{-4}$. This estimation is computed with the subset simulation. The Monte Carlo sample of size $N = 10^6$ provides the estimation $\tilde{p}_f = 2.26 \times 10^{-4}$. We use this same sample to estimate the perturbed failure probability $p_{i\delta}$ and the sensitivity indices $S_{i\delta}$. Here, we use the basic formulation of the indices presented in Section 5.4.1.

There are two possible ways for sensitivity analysis of the truncated Gaussian distribution. We can perturb either the boundaries of the support or the mean value of the distribution. In this example, we compare the results for both cases. We start with the sensitivity indices calculated by the mean shifting. Figure 5.7 displays the evolution of the sensitivity indices for $\delta \in [0, 1]$ for negative (Figure 5.7(a)) and positive (Figure 5.7(b)) values of shifting. The parameters ranking depends on the sign of τ. When $\tau > 0$ (i.e. the positive mean shifting) the porosity parameter has the highest impact on the failure probability. It means that increasing the mean value of the reservoir porosity PORO leads to increasing the failure probability. On

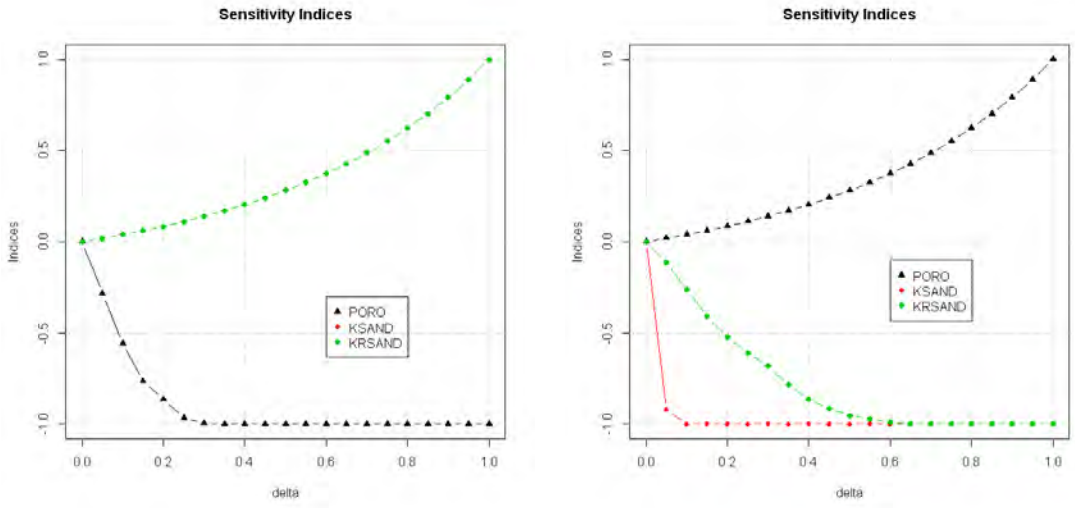
the contrary, increasing the mean value of the reservoir permeability KSAND and the water relative permeability end-point KRSAND has a negative effect on the risk of leakage. For the negative mean shifting, the parameters KSAND and KRSAND have the highest influence on the failure probability. Reducing the reservoir permeability and the end-point water relative permeability impedes the gas flow in the reservoir. It increases the risk of the leakage.



(a) Negative τ (b) Positive τ

Figure 5.7: Mean shifting.

Now, we consider the boundaries perturbation. It means that we are moving one of the the distribution boundaries by δ in positive or negative directions keeping the values of mean and variance unchanged. Figure 5.8 depicts the sensitivity indices for $\delta \in [0, 1]$ for negative (Figure 5.8(a)) and positive (Figure 5.8(b)) values of $\tau = \pm\delta$. When $\tau < 0$ (i.e. the resulting distribution is $\mathcal{N}_{[-1,1-\delta]}(0, 1)$) the porosity parameter has the highest impact on the failure probability. It means that by decreasing the maximum value of the reservoir porosity PORO the failure probability decreases. It is also proved by Figure 5.7(a). It can be also observed that for this parameter the new perturbed failure probability $p_{i\delta}$ is equal to zero when $\delta > 0.3$. Then, for $\tau > 0$ the resulting distribution is $\mathcal{N}_{[-1+\delta,1]}(0, 1)$. For this case, increasing the reservoir permeability KSAND and the water relative permeability end-point KRSAND reduces the failure probability $p_{i\delta}$.



(a) Negative δ (b) Positive δ

Figure 5.8: Boundaries shifting.

Both methods provide comprehensive and complementary results. If the main uncertainty is about the boundaries of the support, one can start with the boundaries perturbation. By moving the boundaries of the original distribution, it is possible to determine the safe intervals for the input parameters by detecting the value of δ: $p_{i\delta} = 0$ or $S_{i\delta} = -1$. After that, the effect of the mean perturbation can be studied.

5.6 Conclusions

In this chapter, we introduced a novel approach to reliability sensitivity analysis. Currently the majority of the methods for reliability analysis is based on the variance decomposition and Sobol' sensitivity indices. We present a new moment independent sensitivity measure. The method is based on a perturbation of the original probability distribution of input random variables. We can analyze the a priori assumption about the input distributions and measure the effect of possible deviations from this assumption. In particular, we select the Kullback-Leibler divergence as a measure of the perturbation.

We introduced a class of distributions from which we select the perturbed distribution. We provided the analysis of different probability distribution families and studied the effect of the perturbation of the Kullback-Leibler divergence for these families.

For the perturbed distribution, we proposed an effective method to estimate the corresponding perturbed failure probability. The method is based on a technique coming from the importance sampling simulation method. It allows to estimate the new failure probability without supplementary performance function evaluations. The new sensitivity indices formulation describes the relation between the new failure probability $p_{i\delta}$ and the original failure probability of the system p_f. By varying the value of the perturbation δ, we can study how the positive or negative probability density perturbation affect the failure probability. If the model has input controllable parameters, the method can help improving the system reliability and the design optimization.

We investigated the method on analytical and CO_2 storage reservoir cases. The method provides relevant results and can be applied in reliability sensitivity analysis. In the presented examples we considered the identically distributed input variables. However, the method can be equally applied to different families of distributions.

We also provided the analysis of the statistical properties of the proposed estimators for the perturbed failure probability $\widehat{p}_{i\delta}$ and the sensitivity indices $\widehat{S}_{i\delta}$.

Chapter 6

Conclusions and perspectives

The objective of this work has been to provide a comprehensive thorough methodology for uncertainty and risk analysis in CO_2 storage. The presented methodology gives some solutions to the problem from different aspects.

We focus in this thesis on a Gaussian Process (GP) response surface modelling. The expensive reservoir simulator is replaced with a GP response surface model that is much faster to evaluate. The presented methodology is based on a GP response surface modelling and consists in the following steps:

1. Optimal injection well placement
2. Reliability estimation
3. Reliability sensitivity analysis

We will discuss the most important contributions separately for each of the formulated problems.

Injection well placement

First of all, the problem of CO_2 injection well placement represents a problem of handling discrete parameters (such as well coordinates) in the GP response surface modelling. In Chapter 2 we have presented different formulations for a correlation function involving both discrete and continuous variables. We have tested these formulations and their performance on analytical function examples. We have successfully used the method on a CO_2 reservoir storage case.

While working with a well placement optimization another difficulty arises. It is related to addressing the discrete functional outputs. In this work, the dynamic

simulator models a multi-phase 3-D fluid flow in the storage reservoir operating over fixed number of time-steps. The output of such simulators is a sequence of outputs at different time-steps. Therefore, it is a time series, such as a overpressure development at different years of injection. The classical approach to model time-series outputs assumes distinct GP modelling for every time step. In Chapter 3 we have presented two different methods to model time series outputs.

The first one is based on considering the time variable as an auxiliary discrete variable. However, it can be computationally demanding. The GP response surface modelling requires the inversion of the design covariance matrix. Depending on the size of the matrix (i.e. the number of available experimental design data and number of involved time steps), the matrix inversion could be infeasible. In addition, the difficulties could be encountered in the case of high density of the points in the time scale when the design data are in close proximity. We propose three different techniques that helps in improving the method efficiency. We have tested proposed method on a PUNQ-S3 reservoir case. Moreover, we have provided a CO_2 reservoir storage example with a direct application to an injection well placement.

Another approach to address time series outputs is a functional based approach. This approach involves a combination of Shape Invariant Model (SIM) and the Gaussian Process (GP) response surface modelling. This model assumes a common pattern shape curve and curve-specific differences of time series outputs in amplitude and timing are modeled with linear transformations. We provided an efficient algorithm for estimation of the transformation parameters of the shape invariant model with the specification for large sets of data curves. We proposed a novel approach that reduces the problem of functional outputs modelling to one optimization problem and GP response surface modelling for transformation parameters. This method does not depend on the number of time steps and could be very advantageous for complex dynamic simulators with a large number of time steps. The prediction algorithm was tested on a CO_2 reservoir storage case. It was compared to a classical single step approach. For the presented example, new method outperforms the classical one in terms of CPU time with a factor five. However, when the design set of curves is significantly different in functional behavior, preliminary classification may be required.

Reliability estimation

Chapter 4 provides a critical review of existing reliability methods for computationally expensive reservoir simulation models. We have provided a detailed discussion of the existing methods for reliability analysis, including approximation methods (such as First and Second Order Reliability Methods (FORM/SORM)) and simulation methods (such as Monte Carlo sampling, Importance Sampling, Directional Sampling and Subset Simulation). Many of the relative advantages and drawbacks of these methods have been presented. In particular, while presented simulation methods give the robust estimation of the failure probability, applying to a reservoir simulator all of them becomes very computationally demanding. For this reason we provide another approximation method. It is based on a GP response surface model approximation of the reservoir simulator.

The reservoir simulator approximation is computed using an experimental design. The design represents a limited number of simulator runs. For this reason, in order to provide a robust estimation of the failure probability of a system, it is highly important that the model is sufficiently accurate in the failure region and close to limit-state surface. Whereas classical experimental design techniques, such as Latin hyper cube, aim at uniform coverage of the input domain. Hence, we propose a new adaptive refinement technique for an experimental design. It is based on a subset simulation algorithm. We propose an updated subset simulation that allows to compute the failure probability with a target level of accuracy. The method consist in integrating the data generated at every level of the subset simulation algorithm in experimental design refinement. It allows to get more information from the failure region. The method entails simultaneous update of the design and re-evaluating the corresponding GP model.

We have tested the method with a known analytical function and a CO_2 storage reservoir case. The failure probability estimations provided by the original and the updated GP models have demonstrated the method efficiency. From a practical experience, we propose the threshold value $\varepsilon = 0.1$ as a reasonable trade-off between the affordable number of iterations and the reliable failure probability estimation. The proposed adaptive refinement technique offers a robust estimation of the probability of leakage for a chosen CO_2 injection well placement.

Reliability sensitivity analysis

As soon as we know the failure probability of the system, it is important to analyze which input parameter contribute most in the variability of the failure probability. Chapter 5 introduced a new moment independent measure for reliability sensitivity analysis. This value is based on a perturbation on the original probability distribution of input random variables. It can help to analyze how the a priori assumption about the input distributions affect the failure probability estimation. Kullback-Leibler divergence was chosen as a measure of perturbation. We have studied the effect of the perturbation for different probability distributions..

We propose a new method to estimate the perturbed failure probability without additional simulator runs. It is based on importance sampling method. Consequently, without additional CPU cost by varying the value of perturbation we can study the impact of positive or negative probability density perturbation on the failure probability.

In order to compare numerically the influence of the input variables on the failure probability, we have provided different possible formulations for sensitivity indices. The analysis of statistical properties of the suggested estimator was also presented.

The method was tested both on an analytical function example and a CO_2 storage reservoir case. The presented results are promising. The method could be effectively applied in reliability sensitivity analysis.

Future research

Preliminary curves classification

In Chapter 3 we have presented the approach to model discrete functional outputs based on Shape Invariant Model. This approach could be improved by adding an intelligent curves classification. Starting from the experimental design set of curves we can perform a preliminary curves classification. Then, separately for every class we compute corresponding transformation parameters. We compute the GP response surface models for the transformation parameters in every class. Then, we would have to develop a clustering function. This function for a given input configuration will determine a number of cluster. When the number of cluster is

known the appropriate GP model is used for parameters estimation.

We have tested some curves classification methods. However, for our application it was not advantageous in terms of CPU time. In this case, we have to solve not a single optimization problem. The number of optimizations is equal to the number of curves clusters. However, for some applications it could outperform the classical single step approach.

Reliability analysis with a random performance function

The threshold value defining the performance function is not always a fixed value. It depends on the input parameters of the system. We consider the input parameters as random variables. Therefore, it seems to be reasonable to consider the threshold value (cap rock fracturing pressure in our case) as the random variable. This random function depends on the configuration of input parameters.

It is interesting to develop a subset simulation algorithm with a random threshold value. In this case the algorithm should be modified as the intermediate threshold values will depend on the probability distribution of the original threshold.

Reliability based design optimization

Reliability based design optimization aims to find optimum set of controllable design variables maximizing the system reliability.

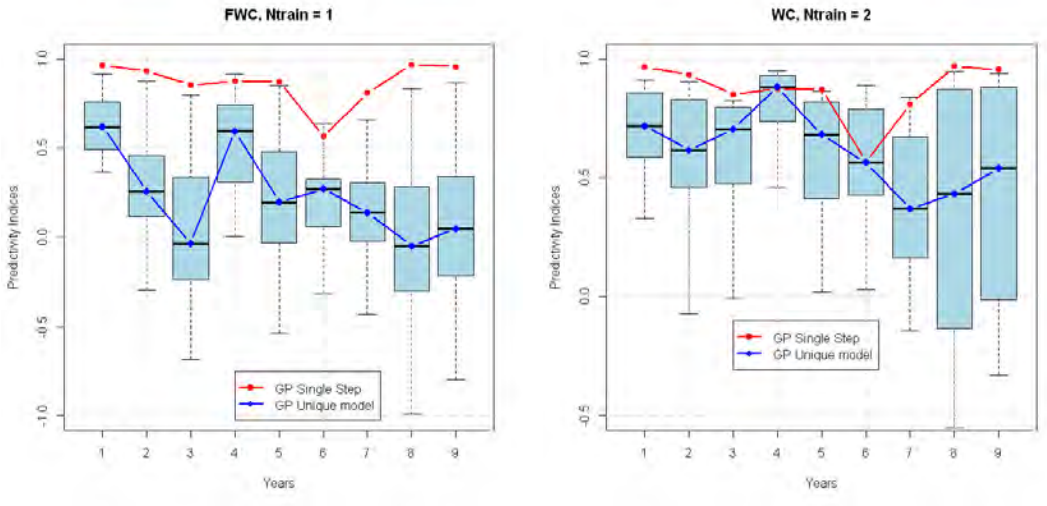
The proposed methodology for reliability sensitivity analysis could contribute to the reliability based design optimization. By perturbing the original probability distribution of the input controllable parameters we can analyze changes in the failure probability. Therefore, by varying the values of perturbation δ, we can find the optimal probability distribution parameters (such as mean and variance) for the design variables in order to achieve the lowest possible failure probability.

For the case of CO_2 storage we can study the range of optimal injection rate or the longest possible injection period. It could be coupled with the classical reliability based design optimization methods for a higher efficiency.

Appendix A

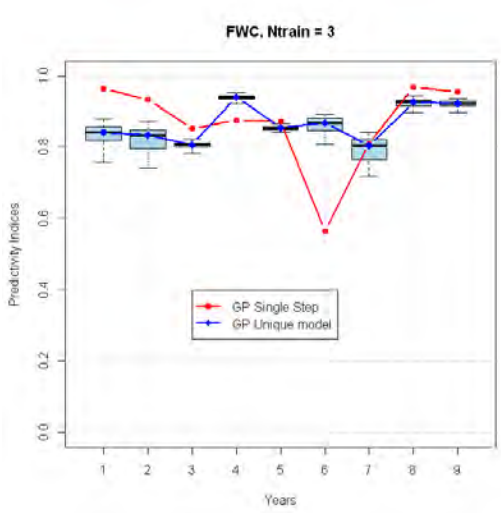
A.1 PUNQ-S3 example

A.1.1 Field watercut function



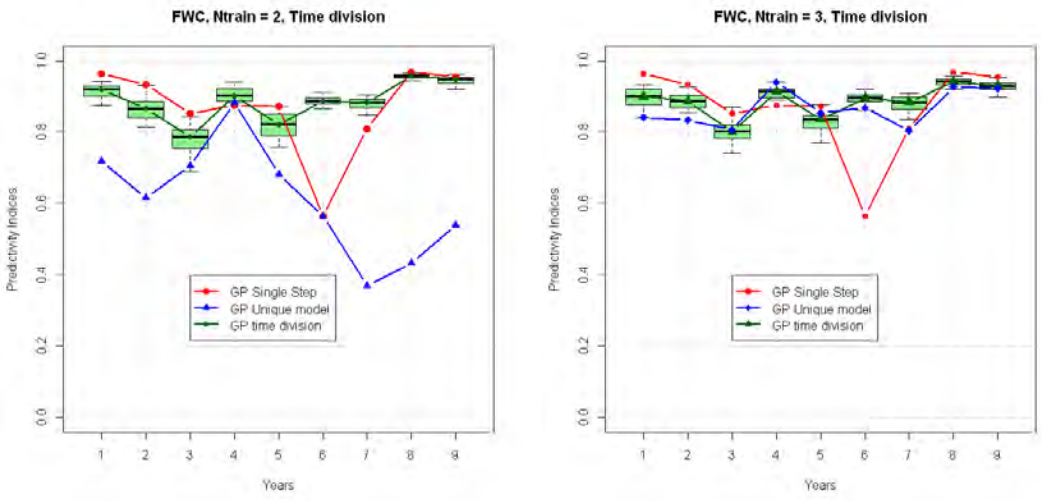
(a) Ntrain=1

(b) Ntrain=2



(c) Ntrain=3

Figure A.1: FWC

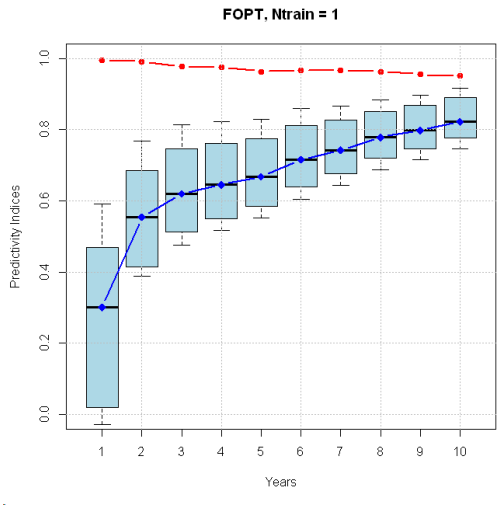


(a) Ntrain=2

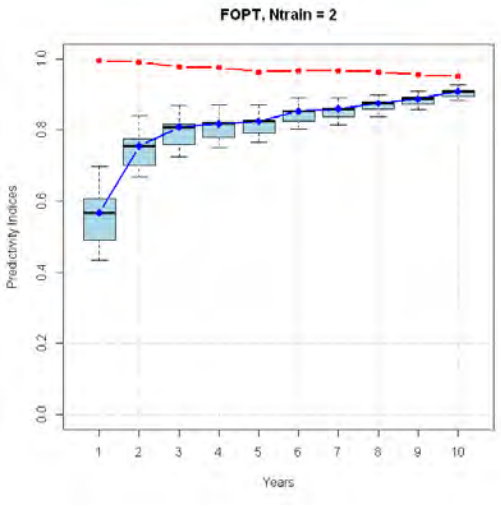
(b) Ntrain=3

Figure A.2: Time division

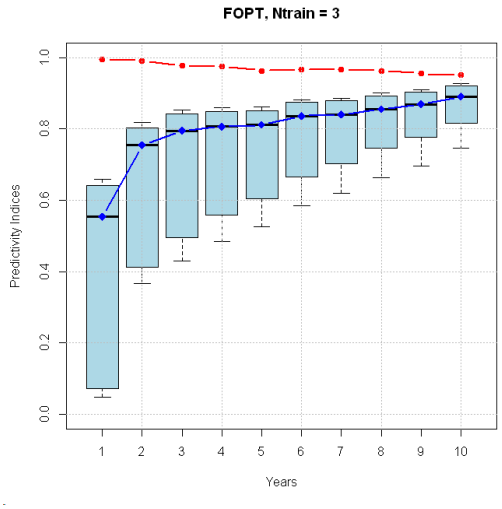
A.1.2 Cumulative field oil production function



(a) Ntrain=1

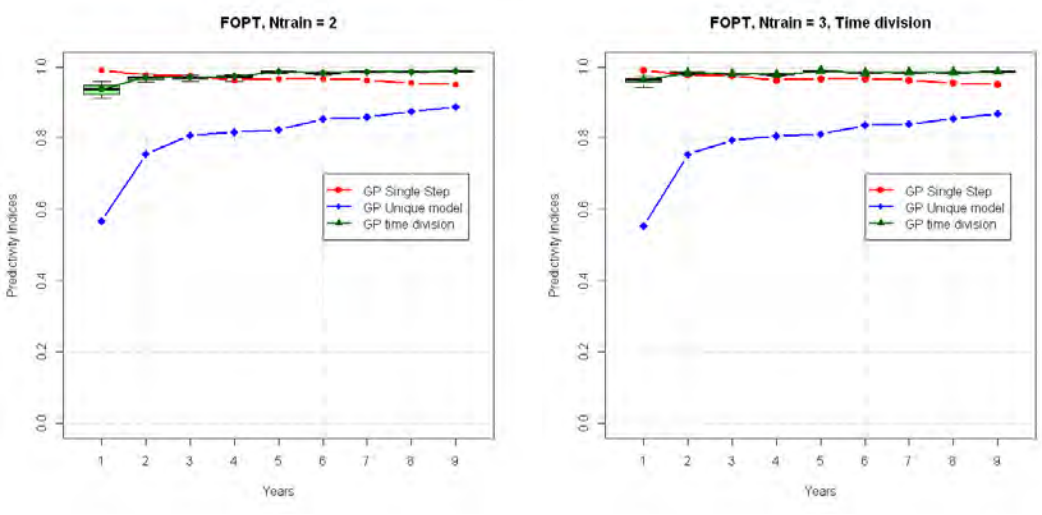


(b) Ntrain=2



(c) Ntrain=3

Figure A.3: FOPT



(a) Ntrain=2 (b) Ntrain=3

Figure A.4: Time division

Appendix B

B.1 Properties of the normalization function

For a given random variable x with probability density $f(x)$. We propose the density modification $x_\tau \sim f_\tau(\cdot)$ as follows:

$$f_\tau(x) = \exp(\tau x - \psi(\tau))f(x),$$

where $\psi(\tau)$ is a normalization function given by:

$$\psi(\tau) = \log\left[\int_{-\infty}^{\infty} \exp(\tau x) f(x) dx\right].$$

Let us define $\mathcal{D} = \{\tau \in \mathbb{R} : \psi(\tau) < +\infty\}$ and $\mathring{\mathcal{D}}$ is the interior of $\mathcal{D}$. We will also suppose, that $\exists \varepsilon : \mathcal{D} \supset]-\varepsilon, \varepsilon[$. Here, we will study the properties of this normalization function.

B.1.1 Derivatives

- $\psi'(\tau) = \mathbb{E}[x_\tau], \tau \in \mathring{\mathcal{D}}$

$$\psi'(\tau) = \frac{\frac{d}{d\tau}\left[\int_{-\infty}^{\infty} \exp(\tau x) f(x) dx\right]}{\int_{-\infty}^{\infty} \exp(\tau x) f(x) dx} = \int_{-\infty}^{\infty} x \exp(\tau x - \psi(\tau)) f(x) dx = \int_{-\infty}^{\infty} x f_\tau(x) dx = \mathbb{E}[x_\tau]$$

- $\psi''(\tau) = \mathbb{E}\left[x_\tau - \mathbb{E}(x_\tau)\right]^2, \tau \in \mathring{\mathcal{D}}$

$$\begin{aligned}
\psi''(\tau) &= \frac{d}{d\tau}\left[\int_{-\infty}^{\infty} x\exp(\tau x-\psi(\tau))f(x)dx\right] = \\
&= \int_{-\infty}^{\infty}[x(x-\psi'(\tau))\exp(\tau x-\psi(\tau))f(x)dx] = \\
&= \int_{-\infty}^{\infty} x^2 f_\tau(x)dx - \psi'(\tau)\int_{-\infty}^{\infty} x f_\tau(x)dx = \int_{-\infty}^{\infty} x^2 f_\tau(x)dx - [\mathbb{E}[x_\tau]]^2 = \\
&= \mathbb{E}\left[x_\tau - \mathbb{E}(x_\tau)\right]^2 = \mathrm{VAR}(x_\tau)
\end{aligned}$$

- $\psi'''(\tau) = \mathbb{E}\left[x_\tau - \mathbb{E}(x_\tau)\right]^3, \tau \in \mathring{\mathcal{D}}$

$$\begin{aligned}
\mathbb{E}\left[x_\tau - \mathbb{E}(x_\tau)\right]^3 &= \int_{-\infty}^{\infty}\left[x-\psi'(\tau)\right]^3 f_\tau(x)dx = \\
&= \int_{-\infty}^{\infty} x^3 f_\tau(x)dx - 3\psi'(\tau)\int_{-\infty}^{\infty} x^2 f_\tau(x)dx + 3\psi'(\tau)^2\int_{-\infty}^{\infty} x f_\tau(x)dx - \psi'(\tau)^3 \\
&= \int_{-\infty}^{\infty} x^3 f_\tau(x)dx - 3\psi'(\tau)\int_{-\infty}^{\infty} x^2 f_\tau(x)dx + 2\psi'(\tau)^3 = \psi'''(\tau)
\end{aligned}$$

B.1.2 The two solutions of the equation

Here, we will discuss about possible solutions of the equation $\psi'(\tau)\tau - \psi(\tau) = \delta, \tau \in \mathring{\mathcal{D}}$ for sufficiently small $\delta > 0$.

Let $G(\tau) := \psi'(\tau)\tau - \psi(\tau) - \delta, \tau \in \mathring{\mathcal{D}}$. We will show that this function has exactly two zeros $\tau_1 < 0$ and $\tau_2 > 0$.

- $G'(\tau) = \tau\psi''(\tau), \tau \in \mathring{\mathcal{D}}$

We do not consider a constant random variable. Hence, $\psi''(\tau) = \mathrm{VAR}(x_\tau) > 0$ and $G'(\tau)$ has the same sign as $\tau \in \mathring{\mathcal{D}}$ and $G'(\tau) = 0 \Leftrightarrow \tau = 0$.

- $G''(\tau) = \tau\psi'''(\tau) + \psi''(\tau)|_{\tau=0} = \psi''(0) = \mathrm{VAR}(x) > 0$

Therefore, $\tau = 0$ is the minimum. $G(0) = -\psi(\tau)|_{\tau=0} - \delta = -\delta < 0$ for $\delta > 0$. The derivative $G'(\tau) < 0$ for $\tau < 0$ and $G'(\tau) > 0$ for $\tau > 0$. Thus, $G(\tau)$ is strictly decreasing for $\tau < 0$ and $G(\tau)$ is strictly increasing for $\tau > 0$. Hence, for sufficiently small δ and $\tau \in \mathring{\mathcal{D}}$ the function $G(\tau)$ has less than two zeros $\tau_1 < 0$ and $\tau_2 > 0$ if both of them $\tau_1, \tau_2 \in \mathring{\mathcal{D}}$.

B.2 Statistical properties of the indices estimator

Here, we will study statistical properties of the estimator of the perturbed failure probability $\widehat{p}_{i\delta}$ and corresponding estimator of the sensitivity indices $S_{i\delta} = \frac{\widehat{p}_{i\delta} - \widehat{p}_f}{\widehat{p}_f}$. We will start with studying the properties of $\widehat{p}_{i\delta}$.

B.2.1 Estimator of the perturbed failure probability

Suppose, $f_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{d} f_{x_i}(x_i)$ is the input joint density and $f_{x_{i\tau}}$ is a perturbed probability density for the parameter x_i. Recall that for a sample of size N: $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \overset{i.i.d.}{\sim} f_{\mathbf{x}}(\mathbf{x})$, the estimation of $\widehat{p}_{i\delta}$ is computed by:

$$\widehat{p}_{i\delta} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{I}_{g(\mathbf{x}_k<0)} \frac{f_{x_{i\tau}}(\mathbf{x}_{k_i})}{f_{x_i}(\mathbf{x}_{k_i})}.$$

First, we study the expectation and the variance of this estimator.

1. $\mathbb{E}_{f_{\mathbf{x}}}[\widehat{p}_{i\delta}] = p_{i\delta}$
2. $\mathrm{VAR}_{f_{\mathbf{x}}}[\widehat{p}_{i\delta}] = \frac{1}{N}\mathrm{VAR}_{f_{\mathbf{x}}}\left[\mathbf{I}_{g(\mathbf{x})<0}\frac{f_{x_{i\tau}}(x_i)}{f_{x_i}(x_i)}\right] = \frac{1}{N}\left[\int \mathbf{I}_{g(\mathbf{x})<0}\frac{f^2_{x_{i\tau}}(x_i)}{f^2_{x_i}(x_i)} f_{\mathbf{x}}(\mathbf{x})d\mathbf{x} - p^2_{i\delta}\right]$

This variance tends to 0 when $N \to \infty$. Furthermore, by the Central Limit Theorem (CLT):

$$\frac{1}{\sqrt{\mathrm{VAR}_{f_{\mathbf{x}}}[\widehat{p}_{i\delta}]}}\left(\widehat{p}_{i\delta} - p_{i\delta}\right) \overset{N\to\infty}{\to} \mathcal{N}(0,1)$$

Note that the covariance between the estimator $\widehat{p}_f$ and $\widehat{p}_{i\delta}$ does not vanish. Indeed, we use the same sample to estimate p_f and $p_{i\delta}$. We can compute this covariance (see also C.6.2):

$$\begin{aligned}
\mathrm{COV}(\widehat{p}_f, \widehat{p}_{i\delta}) &= \mathbb{E}_{f_{\mathbf{x}}}(\widehat{p}_f\widehat{p}_{i\delta}) - \mathbb{E}(\widehat{p}_f)\mathbb{E}(\widehat{p}_{i\delta}) = \\
&= \frac{N}{N^2}\int \mathbf{I}_{g(\mathbf{x}_k<0)}\frac{f_{x_{i\tau}}(x_i)}{f_{x_i}(x_i)} f_{x_i}(x_i)d\mathbf{x} - p_f p_{i\delta} = \\
&= \frac{1}{N}p_{i\delta}(1-p_f)
\end{aligned}$$

The value of this covariance decreases when the sample size N increases.

B.2.2 Sensitivity indices estimator

Recall, that the first sensitivity index is:

$$S_{i\delta} = \frac{p_{i\delta} - p_f}{p_f} = \frac{p_{i\delta}}{p_f} - 1, i = 1, \ldots, d.$$

We estimate this value with the Monte Carlo method and the importance sampling by estimating consistently $p_{i\delta}$ and p_f. The estimator of this index is:

$$\widehat{S}_{i\delta} = \frac{\widehat{p}_{i\delta}}{\widehat{p}_f} - 1.$$

Here, we will study some proprieties of this estimator.

It is not straightforward to compute directly $\mathbb{E}_{f_{\mathbf{x}}}\left[\widehat{S}_{i\delta}\right] = \mathbb{E}_{f_{\mathbf{x}}}\left[\frac{\widehat{p}_{i\delta}}{\widehat{p}_f}\right] - 1$ and $\mathrm{VAR}_{f_{\mathbf{x}}}\left[\widehat{S}_{i\delta}\right] = \mathrm{VAR}_{f_{\mathbf{x}}}\left[\frac{\widehat{p}_{i\delta}}{\widehat{p}_f}\right]$. We propose to use the Delta Method to approximate these values [Van der Vaart, 2000].

Let us recall the Taylor expansion with integral form for the remainder. Let ϕ be a two times differentiable function on $[t_0, t]$, then:

$$\phi(t) := \phi(t_0) + \phi'(t_0)(t - t_0) + \int_{t_0}^{t} (1 - u)\phi''(u)du \tag{B.1}$$

We will define a function:

$$\phi(t) = \frac{y(t)}{x(t)} - 1,$$

where $x(t) = (1-t)p_f + t\widehat{p}_f$ and $y(t) = (1-t)p_{i\delta} + t\widehat{p}_{i\delta}$. For this function: $\phi(0) = S_{i\delta}$ and $\phi(1) = \widehat{S}_{i\delta}$. Following the Taylor expansion (B.1) we will expand $\phi(t)$ with $t = 1$ and $t_0 = 0$. First, we will compute the derivatives.

1. $x'(t) = \widehat{p}_f - p_f$
2. $y'(t) = \widehat{p}_{i\delta} - p_{i\delta}$

Then,

$$\begin{aligned} \phi'(t) &= \frac{y'(t)x(t) - x'(t)y(t)}{x^2(t)} = \frac{p_f\widehat{p}_{i\delta} - \widehat{p}_f p_{i\delta}}{\left((1-t)p_f + t\widehat{p}_f\right)^2} \\ \phi'(t)|_{t=0} &= \frac{p_f\widehat{p}_{i\delta} - \widehat{p}_f p_{i\delta}}{p_f^2} \end{aligned}$$

The second derivative is:

$$\phi''(t) = \frac{2x(t)x'(t)\left(\widehat{p}_f p_{i\delta} - p_f\widehat{p}_{i\delta}\right)}{x^4(t)} = \frac{2x'(t)\left(\widehat{p}_f p_{i\delta} - p_f\widehat{p}_{i\delta}\right)}{x^3(t)} = \frac{2(\widehat{p}_f - p_f)\left(\widehat{p}_f p_{i\delta} - p_f\widehat{p}_{i\delta}\right)}{x^3(t)}.$$

Therefore, the reminder is:

$$\int_0^1 \frac{2(1-t)(\widehat{p}_f - p_f)\left(\widehat{p}_f p_{i\delta} - p_f \widehat{p}_{i\delta}\right)}{\left((1-t)p_f + t\widehat{p}_f\right)^3} dt = \frac{(\widehat{p}_f - p_f)(p_f \widehat{p}_{i\delta} - \widehat{p}_f p_{i\delta})}{p_f^2 \widehat{p}_f}.$$

So that, by the Taylor expansion (B.1) we obtain:

$$\widehat{S}_{i\delta} = S_{i\delta} + \frac{p_f \widehat{p}_{i\delta} - \widehat{p}_f p_{i\delta}}{p_f^2} + \frac{(\widehat{p}_f - p_f)(p_f \widehat{p}_{i\delta} - \widehat{p}_f p_{i\delta})}{p_f^2 \widehat{p}_f}.$$

The last term $R = \frac{(\widehat{p}_f - p_f)(p_f \widehat{p}_{i\delta} - \widehat{p}_f p_{i\delta})}{p_f^2 \widehat{p}_f}$ is the remainder. This remainder is bounded and we can neglect it for the approximation.

$$\widehat{S}_{i\delta} \approx S_{i\delta} + \frac{p_f \widehat{p}_{i\delta} - \widehat{p}_f p_{i\delta}}{p_f^2}.$$

Now, we can approximate the mean and the variance of $\widehat{S}_{i\delta}$.

1. $\mathbb{E}_{f_{\mathbf{x}}}\left[\widehat{S}_{i\delta}\right] \sim S_{i\delta}$
2. $\mathrm{VAR}_{f_{\mathbf{x}}}\left[\widehat{S}_{i\delta}\right] \sim \mathrm{VAR}_{f_{\mathbf{x}}}\left[\frac{\widehat{p}_{i\delta}}{p_f}\right] + \mathrm{VAR}_{f_{\mathbf{x}}}\left[\frac{\widehat{p}_f p_{i\delta}}{p_f^2}\right] - \frac{2p_{i\delta}}{p_f^3}\mathrm{COV}(\widehat{p}_{i\delta}, \widehat{p}_f) = \frac{1}{p_f^2}\mathrm{VAR}_{f_{\mathbf{x}}}\left[\widehat{p}_{i\delta}\right] - \frac{p_{i\delta}^2(1-p_f)}{N p_f^3}$.

Therefore, the variance of the indices estimator tends to 0 when $N \to \infty$. With some extra computations we can show that:

$$\frac{1}{\sqrt{\mathrm{VAR}_{f_{\mathbf{x}}}\left[\widehat{S}_{i\delta}\right]}}\left(\widehat{S}_{i\delta} - S_{i\delta}\right) \overset{N\to\infty}{\to} \mathcal{N}(0,1).$$

So that, knowing the variance of the estimator, the confidence region for the indices may be computed.

B.3 Lambert W function

Lambert W function, also known as the Omega function or Product logarithm, is a set of functions (or the branches) of the inverse relation of the function $f(w) = w\exp(w)$, where $w \in \mathbb{C}$ is complex. The following equation defines $W(z)$:

$$z = W(z)\exp(W(z)), \quad z \in \mathbb{C}.$$

The Lambert W relation cannot be expressed by elementary functions. The Lambert W relation is multivalued. If we consider only real-valued branches of W, then the function is defined only for $x \geq -1/e$. This funciton is double-valued on $(-1/e, 0)$. The additional constraint $W = -1$ defines a single-valued function $W_0(x)$: $W_0(0) = 0$ and $W_0(-1/e) = -1$. Meanwhile, the lower branch is such that $W_{-1} = -1$ and it is denoted by $W_{-1}(x)$. This branch decreases from $W_{-1}(-1/e) = -1$ to $W_{-1}(0^-) = -\infty$.

More information about Lambert W function could be found in Corless et al. [1996].

B.4 Truncated Gaussian distribution

Suppose $x_i \sim \mathcal{N}(\mu, \sigma^2)_{[a,b]}$ is distributed according truncated Gaussian distribution. Suppose, that $\Phi(\cdot)$ is the standard normal cumulative distribution function. Let us make few notations to ease the density introduction:

$$\alpha = \frac{a-\mu}{\sigma}, \quad \beta = \frac{b-\mu}{\sigma}, \quad Z = \Phi(\beta) - \Phi(\alpha),$$

where $\Phi = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} \exp\left(-\frac{u^2}{2}\right) du$ is the standard normal distribution function. Then the density function of the Gaussian distribution truncated on the interval $[a, b]$ is given by:

$$f(x) = \frac{1}{Z\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \mathbf{I}_{[a,b]},$$

where $\mathbf{I}_{[a,b]}$ is an indicator function for the interval $[a, b]$. Therefore, the expression for $\psi(\tau)$ is:

$$\psi(\tau) = \log\left[\int_a^b (\exp(\tau x) f(x))\, dx\right].$$

By direct calculation we can obtain:

$$\psi(\tau) = \frac{\tau^2\sigma^2}{2} + \mu\tau + \log\left[\frac{\Phi\left(\frac{b-\tau\sigma^2-\mu}{\sigma}\right) - \Phi\left(\frac{a-\tau\sigma^2-\mu}{\sigma}\right)}{Z}\right].$$

The derivative is expressed as:

$$\psi'(\tau) = \mu + \tau\sigma^2 + \sigma\frac{\phi\left(\frac{b-\tau\sigma^2-\mu}{\sigma}\right) - \phi\left(\frac{a-\tau\sigma^2-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\tau\sigma^2-\mu}{\sigma}\right) - \Phi\left(\frac{a-\tau\sigma^2-\mu}{\sigma}\right)},$$

where $\phi(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right)$ is the standard normal density function. The solutions of the equation:

$$\psi'(\tau)\tau - \psi(\tau) = \delta \tag{B.2}$$

can be found using a numerical solver.

Suppose, τ^* is one of the solutions of (B.2). Then, by substituting directly the normalization function $\psi(\tau)$, we obtain the formulation for the perturbed density $f_\tau(x)$:

$$f_\tau(x) = \exp(\tau x - \psi(\tau)) f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{\exp\left(-\frac{\left(x-(\mu+\tau^*\sigma^2)\right)^2}{2\sigma^2}\right)}{\Phi\left(\frac{b-\tau^*\sigma^2-\mu}{\sigma}\right) - \Phi\left(\frac{a-\tau^*\sigma^2-\mu}{\sigma}\right)}.$$

This density corresponds to truncated Gaussian distribution on the interval $[a, b]$ with the mean $\mu + \tau^*\sigma^2$ and the variance σ, where τ^* is the solution of (B.2).

B.5 Variance based sensitivity indices

Here, we will show how to estimate the values of the numerator and the denominator for the sensitivity indices defined in Section 5.4.3.

Recall, that for the variable $x_i \sim f_{x_i}(\cdot)$ the sensitivity indices is defined as:

$$\tilde{\tilde{S}}_{\delta_i} = \frac{\mathrm{VAR}_\delta(p_{i\delta})}{\mathrm{VAR}_\delta(p_{\mathbf{x}\boldsymbol{\delta}})}, \tag{B.3}$$

where $p_{i\delta}$ is the failure probability computed for the δ - perturbed density $f_{x_i\tau}$ and $p_{\mathbf{x}\boldsymbol{\delta}}$ is the failure probability computed by perturbing all the input variables densities $f_{\mathbf{x}} = \prod_{k=1}^{d} f_{x_i}$ simultaneously by a vector $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_d) \in [0, 1]^d$.

We will show how to estimate the numerator and the denominator in (B.3) by a Monte Carlo sampling of the random variable δ.

- The variance $\mathrm{VAR}_\delta(p_{i\delta})$ for the variable x_i

Suppose that $\boldsymbol{\delta}^i = \{\delta_1^i, \ldots, \delta_N^i\} \sim \mathrm{U}_{[0,1]}, i = 1, \ldots, d$ is a set uniform sample of size N of the random variable δ. Then, if $N >> 2$ an unbiased estimator of the numerator in (B.3) is given by:

$$\widehat{\mathrm{VAR}}_\delta(p_{i\delta}) = \frac{1}{N-1} \sum_{k=1}^{N} \left(p_{i\delta_k^i} - \overline{p}_{i\boldsymbol{\delta}^i}\right)^2,$$

where $\overline{p}_{i\boldsymbol{\delta}^i} = \frac{1}{N}\sum_{k=1}^{N} p_{i\delta_k^i}$ is the empirical mean.

By sampling sequentially $\boldsymbol{\delta}^i, \quad i = 1, \ldots, d$, we can estimate the numerator of the index (B.3) for all the input variables $\mathbf{x} = (x_1, \ldots, x_d)$.

- The total variance $\mathrm{VAR}_\delta(p_{\mathbf{x}\boldsymbol{\delta}})$

The denominator of the index (B.3) is the value of the total variance of the perturbed failure probability. It is common for all the variables. To estimate the denominator in (B.3), we will use $N \times d$ matrix composed of d samples of size N of the random variable δ: $\Delta = (\boldsymbol{\delta}^1, \ldots, \boldsymbol{\delta}^d)^\top$, where $\boldsymbol{\delta}^j = \{\delta_1^j, \ldots, \delta_N^j\}, \quad j = 1, \ldots, d$. This matrix provides N independent vector of simultaneous perturbations. If we denote the random vector of simultaneous perturbation by $\boldsymbol{\delta}_k = (\delta_k^1, \ldots, \delta_k^d), \quad k = 1, \ldots, N$, the total variance is then estimated by:

$$\widehat{\mathrm{VAR}}_\delta(p_{\mathbf{x}\boldsymbol{\delta}}) = \frac{1}{N-1}\sum_{k=1}^{N}\left(p_{\mathbf{x}\boldsymbol{\delta}_k} - \overline{p}_{\mathbf{x}\Delta}\right)^2,$$

where $\overline{p}_{\mathbf{x}\Delta} = \frac{1}{N}\sum_{k=1}^{N} p_{\mathbf{x}\boldsymbol{\delta}_k}$ is the empirical mean.

Appendix C

In this Section, we provide an article that was submitted to the Journal of Statistical Computation and Simulation [Lemaitre and Sergienko et. al, 2012]. This article provides another approach to the presented reliability sensitivity analysis. It details the choice of the presented exponential density modification. In this work, we also consider perturbation of a variance of the original distribution. The method was tested numerically on analytical examples and a reliability engineering example.

C.1 Introduction

In the context of structural reliability, computer codes are used in order to assess the safety of industrial systems relying on complex physical phenomena. For instance, an electric operator would like to predict the height of a potential river flood in order to determine the height of a dyke preventing any disaster. In this example, the computer code (simulating the hydraulic model) has some uncertain input variables (flow rate, river length, water height, etc.), that are modelled by random variables. In this paper, the computer code is a deterministic numerical model and one of its output is considered. Due to the randomness of the model inputs, this output is a random variable more or less sensitive to the uncertainty of the input variables.

Sensitivity analysis (SA) is a tool used to explore, understand and (partially) validate computer code. It aims at explaining the outputs regarding the input uncertainties (Saltelli et al. [2000]). The definition of SA differs from fields and authors. We use the "global SA" definition given by Saltelli et al. wherein the whole variation range of the inputs is considered. The application of such an approach can be model simplification (by removing irrelevant modelling elements), input variables ranking or research prioritization. There is a wide range of SA techniques, regarding

what type of problem the experimenter is faced with (Iooss [2011]). For instance, screening methods are to be applied when there is a large number of inputs, and few models assumptions. For a quantitative point of view, the most popular techniques are variance-based methods, based upon the functional Hoeffding variance decomposition Antoniadis [1984] and the so-called Sobol' indices (Saltelli et al.).

It should be noted that most SA methods focus on real-valued continuous numerical output variables. When the quantity of interest is a binary value (e.g. when the numerical model returns "faulty system" or "safe system"), SA techniques are underdeveloped. Some basic techniques can be quoted, such as Monte-Carlo filtering (Saltelli et al.) which consists in measuring differences between a "safe" sample and a "faulty" sample via standard statistical tests. In a different scientific field, the reliability index resulting from the First or Second Order Reliability Methods (FORM/SORM, Lemaire et al. [2009]) can also be used to classify the impact of the inputs on the failure probability. More recent works give methods combining always the two objectives: estimating a failure probability and assessing the influence of the input uncertainty on the failure probability (Morio [2011]; Munoz Zuniga et al. [2011]).

In this paper, a real-valued numerical model denoted by $G : \mathbb{R}^d \to \mathbb{R}$ is considered. This model may further be called the "failure function". In practice, each run of G can be CPU time consuming. We are interested in the event $G(\mathbf{X}) < 0$ (system failure) and in the complementary event $G(\mathbf{X}) \geq 0$ (system safe mode). $\mathbf{X} = (X_1, ..., X_d)^T$ is a d-dimensional continuous random variable whose joint probability density function (pdf) is denoted f. For $i = 1, \cdots, d$, let f_i denotes the distribution of X_i (the marginal density). We make the assumption that all components of $\mathbf{X}$ are independent. The quantity of interest is the system failure probability:

$$P = \int \mathbb{1}_{\{G(\mathbf{x})<0\}} f(\mathbf{x}) d\mathbf{x}.$$

The aim of this work is the quantification of the influence of each variable X_i on this probability, by using the same set of calculations that have been used in the failure probability (P) estimation.

In most cited works, sensitivity indices for failure probabilities were defined in strong correspondence with a given method of estimation (e.g. Lemaire et al. [2009];

Munoz Zuniga et al. [2011]), and their interpretation is consequently limited, as stressed in Lemaitre and Arnaud [2011]. To answer to genericity concerns expressed by these authors, this article first aims at defining sensivity indices that have more intrinsic relevance (Section C.2). Nonetheless, they have to be estimated in practice in function of the method. For simplicity reasons, a classical Monte Carlo framework is considered to estimate P and the indices, and derive the theoretical properties of the estimators of the latter. Pursuing the same idea of offering extended tools of sensitivity analysis, Section C.3 focuses on generic strategies of input perturbation based on maximum entropy rules. The behaviour of the indices is examined in Section C.4 through numerical simulations in various complexity settings, involving toy examples and a realistic case-study. Comparisons with two reference methods (FORM indices and Sobol' indices) highlight the relevance of the new indices in most situations. The main advantages, remaining issues are finally discussed in the last section of the article, as well as avenues for future research.

C.2 Definition, estimation and properties of a sensitivity index

Given a unidimensional input variable X_i with pdf f_i and some perturbation parameter δ lying in a given subset of $\mathbb{R}$, let call $X_{i\delta} \sim f_{i\delta}$ the corresponding perturbed random input. Accordingly, the failure probability becomes

$$P_{i\delta} = \int 1_{\{G(\mathbf{x})<0\}} \frac{f_{i\delta}(x_i)}{f_i(x_i)} f(\mathbf{x}) d\mathbf{x} \tag{C.4}$$

where x_i is the $i-$th component of the vector $\mathbf{x}$. Independently of the mechanism chosen for the perturbation (see next Section for proposals), a good sensitivity index $S_{i\delta}$ should have intuitive features that make it appealing to reliability engineers and decision-makers. We believe that the following proposal can fulfil these requirements:

$$S_{i\delta} \;=\; \left[\frac{P_{i\delta}}{P} - 1\right] \mathbf{1}_{\{P_{i\delta}>P\}} + \left[1 - \frac{P}{P_{i\delta}}\right] \mathbf{1}_{\{P_{i\delta}<P\}} \;=\; \frac{P_{i\delta} - P}{P \cdot \mathbf{1}_{\{P_{i\delta}>P\}} + P_{i\delta} \cdot \mathbf{1}_{\{P_{i\delta}<P\}}} \,.$$

Firstly, $S_{i\delta} = 0$ if $P_{i\delta} = P$, as expected if X_i is a non-influential variable or if δ expresses a negligible perturbation. Secondly, the sign of $S_{i\delta}$ indicates how the perturbation impacts the failure probability qualitatively. It highlights the situations when $P_{i\delta} > P$ amounts to determining if the remaining (*epistemic*) uncertainty on the modelling $X_i \sim f_i$ can increase the failure risk and therefore should be more accurately analysed. Conversely, P can be interpreted as a conservative assessment of the failure probability, robust to perturbations on X_i, if $P_{i\delta} < P$. In such a case, deeper modelling studies on X_i appear less essential. Thirdly, given its sign the absolute value of $S_{i\delta}$ is of simple interpretation and provides a level of the conservatism or non-conservatism induced by the perturbation: a value of $\alpha > 0$ for the index means that $P_{i\delta} = (1+\alpha)P$. If $S_{i\delta} = -\alpha < 0$ then $P_{i\delta} = (1/(1+|\alpha|))P$.

The postulated ability of $S_{i\delta}$ to enlighten the sensitivity of P to input perturbations must be tested in concrete cases, when an estimator $\widehat{P}_N$ of P can be computed using an already available design of N numerical experiments. In this paper, N is assumed to be large enough such that statistical estimation stands within the framework of asymptotic theory. Besides, we assume for simplicity a standard Monte Carlo design of experiments, according to which $\widehat{P}_N = \sum_{n=1}^{N} \mathbb{1}_{\{G(\mathbf{x}^n)<0\}}/N$ where the $\mathbf{x}^1, \cdots, \mathbf{x}^N$ are independent realisations of X. The strong Law of Large Numbers (LLN) and the central limit theorem (CLT) ensure that for almost all realisations $\widehat{P}_N \xrightarrow[N\to\infty]{} P$ and

$$\sqrt{N/[P(1-P)]}(\widehat{P}_N - P) \quad \xrightarrow[N\to\infty]{\mathcal{L}} \quad \mathcal{N}(0,1).$$

An interest of the Monte Carlo framework is that $P_{i\delta}$ can be consistently estimated without new calls to G, through a "reverse" importance sampling mechanism:

$$\widehat{P}_{i\delta N} \quad = \quad \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}_{\{G(\mathbf{x}^n)<0\}} \frac{f_{i\delta}(x_i^n)}{f_i(x_i^n)}.$$

This property holds in the more general case when P is originally estimated by importance sampling rather than simple Monte Carlo, which is more appealing in contexts when G is time-consuming Beckman and McKay [1987]; Hesterberg [1996]. This generalization is discussed further in the text (Section C.5).

Lemma C.2.1: Assume the usual conditions

(i) $\text{Supp}(f_{i\delta}) \subseteq \text{Supp}(f_i)$,

(ii) $\displaystyle\int_{\text{Supp}(f_i)} \frac{f_{i\delta}^2(x)}{f_i(x)}\, dx < \infty.$

Then, $\widehat{P}_{i\delta N} \xrightarrow[N\to\infty]{} P_{i\delta}$ and $\sqrt{N}\sigma_{i\delta N}^{-1}\left(\widehat{P}_{i\delta N} - P_{i\delta}\right) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}(0,1)$. The exact expression of $\sigma_{i\delta N}^{-1}$ is given in Appendix C.6. It can be consistently estimated by

$$\widehat{\sigma}_{i\delta N}^2 = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}_{\{G(\mathbf{x}^n)<0\}} \left(\frac{f_{i\delta}(x_i^n)}{f_i(x_i^n)}\right)^2 - \widehat{P}_{i\delta N}^2.$$

The asymptotic properties of any estimator of $S_{i\delta}$ will depend on the correlation between $\widehat{P}_N$ and $\widehat{P}_{i\delta N}$. The next proposition summarizes the features of the joint asymptotic distribution of both estimators.

Proposition C.2.2: Under assumptions (i) and (ii) of Lemma C.2.1,

$$\sqrt{N}\Sigma_{i\delta}^{-1/2}\left[\begin{pmatrix} \widehat{P}_N \\ \widehat{P}_{i\delta N} \end{pmatrix} - \begin{pmatrix} P \\ P_{i\delta} \end{pmatrix}\right] \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}_2(0, I)$$

where $\Sigma_{i\delta}$ is given in Appendix C.6 and can be consistently estimated by

$$\widehat{\Sigma}_{i\delta} = \begin{pmatrix} \widehat{P}_N(1-\widehat{P}_N) & \widehat{P}_{i\delta N}(1-\widehat{P}_N) \\ \widehat{P}_{i\delta N}(1-\widehat{P}_N) & \widehat{\sigma}_{i\delta N}^2 \end{pmatrix}.$$

Given $(\widehat{P}_N, \widehat{P}_{i\delta N})$, the plugging estimator for $S_{i\delta}$ is

$$\widehat{S}_{\text{i}\delta N} = \left[\frac{\widehat{P}_{i\delta N}}{\widehat{P}_N} - 1\right]\mathbf{1}_{\left\{\widehat{P}_{i\delta N} > \widehat{P}_N\right\}} + \left[1 - \frac{\widehat{P}_N}{\widehat{P}_{i\delta N}}\right]\mathbf{1}_{\left\{\widehat{P}_{i\delta N} < \widehat{P}_N\right\}}. \tag{C.5}$$

In corollary of Proposition C.2.2, applying the continuous-mapping theorem to the continuous function $s(x,y) = \left[\frac{y}{x} - 1\right]\mathbf{1}_{y>x} + \left[1 - \frac{x}{y}\right]\mathbf{1}_{y<x}$, $\widehat{S}_{\text{i}\delta N}$ converges a.s. to $S_{i\delta}$. The following CLT results from Theorem 3.1 in Van der Vaart [2000].

Proposition C.2.3: Assume that assumptions (i) and (ii) of Lemma C.2.1 hold

and further that $P \neq P_{i\delta}$. Then,

$$\sqrt{N}\left[d^T \Sigma d\right]^{-1/2}\left[\widehat{S}_{\mathrm{i}\delta N} - S_{\mathrm{i}\delta}\right] \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}(0,1) \tag{C.6}$$

with $d = (\frac{\partial s}{\partial x}(P, P_{i\delta}), \frac{\partial s}{\partial y}(P, P_{i\delta}))^T$ for $x \neq y$, and

$$\begin{aligned}\frac{\partial s}{\partial x}(x,y) &= -y\mathbb{1}_{\{y>x\}}/x^2 - (1/y)\mathbb{1}_{\{y<x\}},\\ \frac{\partial s}{\partial y}(x,y) &= (1/x)\mathbb{1}_{\{y>x\}} + x\mathbb{1}_{\{y<x\}}/y^2.\end{aligned}$$

C.3 Methodologies of input perturbation

Our sensitivity analysis method requires the definition of a perturbation for each input. In general, and especially in preliminary reliability studies, there is no prior rule allowing to elicit a specialized perturbation for each input variable. When conducting such an analysis, it is advisable to propose one or several fair methodologies for perturbing the inputs. Roughly speaking, to compare the impact of those perturbations, each should be defined *intrinsically*, independently of each input location and scale. as relative shifts .

More precisely, we suggest to define a perturbed input density $f_{i\delta}$ as the closest distribution to the original f_i in the entropy sense and under some constraints of perturbation. Information-theoretical arguments (Cover and Thomas [2006]) led us to choose the Kullback-Leibler (KL) divergence between $f_{i\delta}$ and f_i as a measure of the discrepancy to minimize under those constraints. Recall that between two densities p and q we have

$$KL(p,q) = \int_{-\infty}^{+\infty} p(y)\log\frac{p(y)}{q(y)}dy \text{ if } \log\frac{p(y)}{q(y)} \in L^1(p(y)dy). \tag{C.7}$$

Let $i = 1, \cdots, d$, the constraints are linear as functional of the modified density f_{mod}:

$$\int g_k(x_i) f_{\mathrm{mod}}(x_i)dx_i = \delta_{k,i} \ \ (k = 1 \cdots K) \tag{C.8}$$

Here, for $k = 1, \cdots, K$, g_k are given functions and $\delta_{k,i}$ are given real. These quantities will lead to a perturbation of the original density. The modified density $f_{i\delta}$ considered in our work is:

$$f_{i\delta} = \underset{f_{\text{mod}} \text{ satisfies } (C.8)}{\text{argmin}} KL(f_{\text{mod}}, f_i) \tag{C.9}$$

and the result takes an explicit form (Csiszár [1975]) given in the following proposition.

Proposition C.3.1: Let us define, for $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_K)^T \in \mathbb{R}^K$,

$$\psi_i(\boldsymbol{\lambda}) = \log \int f_i(x) \exp\left[\sum_{k=1}^{K} \lambda_k g_k(x)\right] dx \ , \tag{C.10}$$

where the last integral can be finite or infinite (in this last case $\psi_i(\boldsymbol{\lambda}) = +\infty$). Further, set dom $\psi_i = \{\boldsymbol{\lambda} \in \mathbb{R}^K \psi_i(\boldsymbol{\lambda}) < +\infty\}$. Assume that there exists at least one density f_m satisfying (C.8) and that dom ψ_i is an open set. Then, there exists a unique $\boldsymbol{\lambda}^*$ such that the unique solution of the minimisation problem (C.9) is

$$f_{i\delta}(x_i) = f_i(x_i) \exp\left[\sum_{k=1}^{K} \lambda_k^* g_k(x_i) - \psi_i(\boldsymbol{\lambda^*})\right]$$

The theoretical technique to obtain $\boldsymbol{\lambda}$ is provided in appendix C.7. Hereby are presented two kind of perturbations used further on.

Mean twisting The first moment is often used to parametrize a distribution. Thus the first perturbation presented here is a mean shift, that is expressed with a single constraint:

$$\int x_i f_{\text{mod}}(x_i) dx = \delta_i \ . \tag{C.11}$$

In term of SA, this perturbation should be used when the user wants to understand the sensitivity of the inputs to a mean shift - that is to say "what if the real mean

of input X_i were δ_i instead of $\mathbb{E}[X_i]$".

Proposition C.3.2: Considering the constraint (C.11) Under the assumptions of Proposition C.3.1 the expression of the optimal perturbed density is

$$f_{i\delta_i}(x_i) = \exp(\lambda^* x_i - \psi_i(\lambda^*)) f_i(x_i)$$

where λ^* is such that equation (C.11) holds.

It can also be noted that equation (C.10) becomes

$$\psi_i(\lambda) = \log \int f_i(x_i) \exp(\lambda x_i) dx_i = \log\left(M_{X_i}(\lambda)\right)$$

where $M_{X_i}(u)$ is the moment generating function (mgf) of the $i-$th input. With this notation λ^* is such that

$$\int x_i \exp\left(\lambda^* x_i - \log\left(M_{X_i}(\lambda^*)\right)\right) f_i(x_i) dx_i = \delta_i \ ,$$

which leads to

$$\int x_i \exp\left(\lambda^* x_i\right) f_i(x_i) dx = \delta_i M_{X_i}(\lambda^*) \ .$$

This equation can be simplified to

$$\frac{M'_{X_i}(\lambda^*)}{M_{X_i}(\lambda^*)} = \delta_i \ .$$

This equation may be easy to solve when one has the expression of the mgf of the input and of its derivative.

Variance twisting In some cases, the mean of an input may not be the main source of uncertainty, but rather the second moment. This case may be treated considering a couple of constraints. The perturbation presented is a variance shift, therefore the set of constraints is

$$\begin{cases} \int x_i f_{\text{mod}}(x_i) dx_i = \mathbb{E}[X_i] \ , \\ \int x_i^2 f_{\text{mod}}(x_i) dx_i = V_{\text{per},i} + \mathbb{E}[X_i]^2 \ . \end{cases} \tag{C.12}$$

The perturbed distribution has the same expectation $\mathbb{E}[X_i]$ as the original one and a perturbed variance $V_{\text{per},i} = \text{Var}X_i \pm \delta_i$.

Proposition C.3.3: Under the assumptions of Proposition C.3.1, for the constraint (C.12), the expression of the optimal perturbed density is

$$f_{i\delta_i}(x_i) = \exp(\lambda_1^* x + \lambda_2^* x^2 - \psi_i(\boldsymbol{\lambda}^*)) f_i(x_i)$$

where λ_1^* and λ_2^* are so that equation (C.12) holds.

Proposition C.3.4: Considering that the original random variable X_i is distributed according to a Natural Exponential Family (NEF). Recalls that a NEF's pdf is of form :

$$f_{i,\theta}(x_i) = b(x) \exp[x_i\theta - \phi(\theta)]$$

where θ is a parameter from a parametric space Θ and

$$\phi(\theta) = \log \int b(x) \exp[x_i\theta]\, dx_i$$

is the cumulant distribution function. Considering the assumptions of Proposition C.3.1, then it is straightforward by (Csiszár [1975])that optimal densities proposed respectively in Proposition C.3.2 and in Proposition C.3.3 are also distributed according to a NEF.

C.4 Numerical experiments

In this Section, the methodology is tested on two academic cases and a more realistic industrial code. The new indices are compared to the results of two reference method, FORM indices (or Importance Factors, IF) and Sobol' indices (SI). Both are computed using the methodologies given in Lemaire et al. [2009] and Sobol' [1993], respectively. Concerning SI, the methodology used is the one given by Sobol' in (Sobol' [1993]). To assess the reproducibility of the estimation of the SI, a sample of 10^5 points is used, and 50 replications are made. Thus all the estimation of the SI are the mean of the obtained values and the coefficient of variation (CV) of the index is provided.

Variable	X_1	X_2	X_3	X_4
Importance factor	0.018	0.679	0.302	0

Table C.1: Importance factors for hyperplane function

C.4.1 Hyperplane failure surface

For the first example, $\mathbf{X}$ is set to be a $4-$dimensional vector, with 4 independent marginal distributions normally distributed with parameters 0 and 1. Therefore $f_{X_i} \sim \mathcal{N}(0,1)$ for $i = 1,..,4$. The failure function is defined as:

$$G(\mathbf{X}) = k - \sum_{i=1}^{4} a_i X_i$$

where k and $\mathbf{a} = (a_1, a_2, a_3, a_4)$ are the parameters of the model. For this numerical example, parameters are set with values $k = 16$ and $\mathbf{a} = (1, -6, 4, 0)$. An explicit expression for P can be given as the sum of the four variables behaves like a Gaussian distribution with mean 0 and standard deviation $\sqrt{\sum_{i=1}^{4} a_i^2}$. Therefore:

$$P = \phi\left(-k/\sqrt{\sum_{i=1}^{4} a_i^2}\right) = \phi\left(\frac{-16}{\sqrt{53}}\right) \simeq 0.014$$

where $\phi(.)$ is the standard normal cumulative distribution function.

It is expected that the influence of X_i on P strongly uniquely depends on $|a_i|$. The greater the absolute value of the coefficient is, the bigger the expected influence is. The aim of choosing one non-influential (dummy) variable is to assess if the SA methods can identify this variable as non-influential on the failure probability.

C.4.1.1 FORM

In this ideal hyperplane failure surface case, FORM performs well as expected Lemaire et al. [2009] by providing an estimate $\widehat{P}_{FORM} = 0.01398$. The importance factors, given in Table C.1, provide an accurate variable ranking for the failure function, given the a_i factors.

Sobol' Index	S_1	S_2	S_3	S_4	S_{T1}	S_{T2}	S_{T3}	S_{T4}
Mean	0.0017	0.2575	0.0544	$9.45 * 10^{-5}$	0.1984	0.9397	0.7256	0
CV	1.5854	0.04826	0.1336	27.4	0.012	0.0069	0.013	0

Table C.2: Sobol' indices for hyperplane function

C.4.1.2 Sobol' indices

The first-order and total indices are displayed in Table C.4.1.2. The interpretation of the results is that X_2 and X_3 concentrate most of the variance of the indicator function. At first order, 25% of its variance is explained by X_2 without any interaction. It should be noted that the total index for X_4 is null, assessing that this variable does not impact the failure probability. The total indices estimators' CV are small, meaning that this method is reproducible and that 10^5 points are enough to estimate in an efficient way the indices S_{Ti} . On the other hand, some CV values for low mean first order indices are quite high. The conclusion of this result is that the method correctly estimates high indices but estimates poorly the indices close to 0. This may not be a problem, given that what is of interest is knowing that the index is close to 0.

C.4.1.3 Density modification based reliability indices

The method presented throughout this article is applied on the hyperplane function. As explained in Section C.3, several way to perturb the input distributions exist. For this case, we choose to apply first a mean twisting, then a variance twisting with fixed mean. A simple calculus gives that the perturbed densities are Gaussian, respectively with the constraint mean and variance 1 for the mean twisting perturbation, and with mean 0 and the constraint variance for the variance twisting perturbation. Thus, the MC estimation gives $\widehat{P} = 0.01446$. For the mean twisting (see (C.11)), the variation range chosen for δ is from -1 to 1 with 40 points, reminding that $\delta = 0$ cannot be considered as a perturbation. For the variance twisting (see (C.12)), the variation range chosen for V_{per} is from 1/20 to 3 with 28 points, where $V_{\text{per}} = 1$ is not a perturbation. The estimated indices are plotted respectively in Figure C.5 for mean twisting and in Figure C.6 for variance twisting. 95% confidence interval are plotted around the indices.

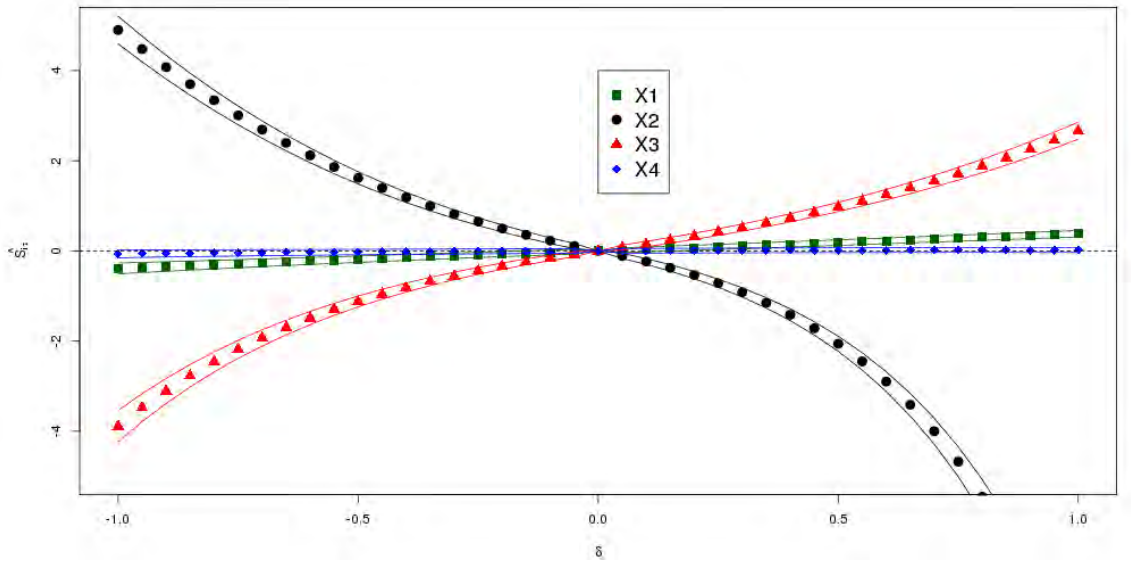


Figure C.5: Estimated indices $\widehat{S_{i\delta}}$ for hyperplane function with a mean twisting

Mean perturbation indices The indices $\widehat{S_{i\delta}}$ behave in a monotonic way given the importance of the perturbation. The slope at the origin is directly related to the value of a_i. For influential variables (X_2 and X_3), the increasing or the decreasing is faster than linear, whereas the curve seem linear for the slightly influential variable (X_1). A modification of the mean of amplitude δ positive will slightly rise the failure probability for variable X_1, highly decrease it for variable X_2 and increase it for variable X_3 (Figure C.5). The effects are reversed with same amplitude for negative δ. It can be seen that variable X_4 has no impact on the failure probability for any perturbation. Those results are consistent with the expression of the failure function.

Variance perturbation indices Increasing the variance of input X_2 and X_3 increases the failure probability, whereas it is the opposite when decreasing the variance(Figure C.6). The modification of the variance of X_1 and X_4 has no effect on the failure probability. The increasing of the indices is linear for X_2 and X_3, and the decreasing of the indices is faster than linear, especially for X_2.

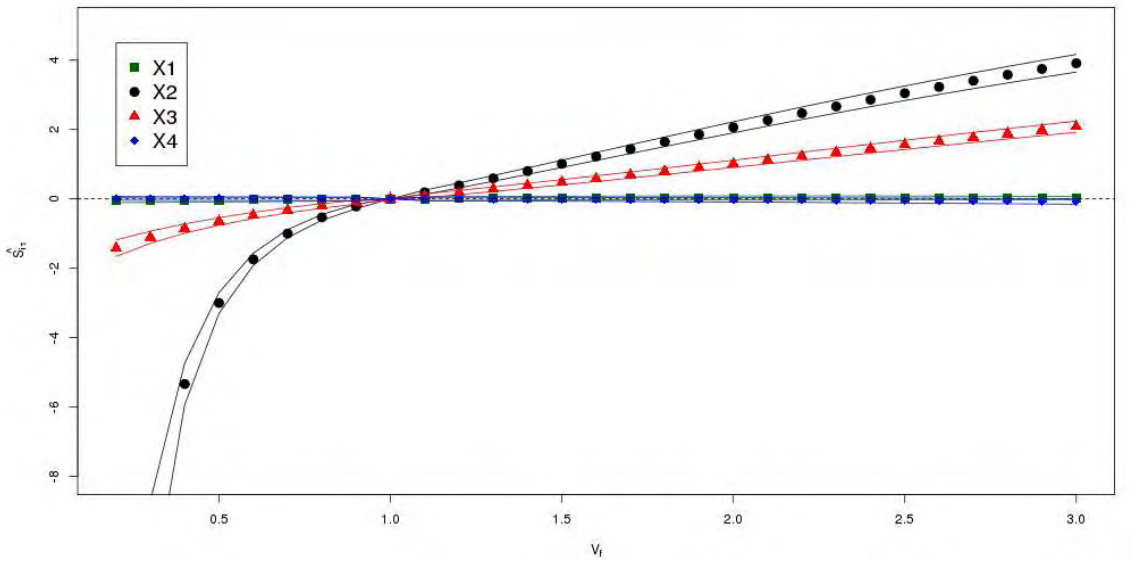


Figure C.6: Estimated indices $\widehat{S_{i,V_{\text{per}}}}$ for hyperplane function with a variance twisting

C.4.2 Thresholded Ishigami function

The Ishigami function is a common test case in SA since it has a complex expression, with interactions between the variables. A modified version of the Ishigami function will be considered in this paper. A threshold that is added to the value obtained with the regular expression is set, and this is considered as the failure function. Therefore one has:

$$G(\mathbf{X}) = \sin(X_1) + 7\sin(X_2)^2 + 0.1X_3^4 \sin(X_1) + k$$

where $k = 7$. $\mathbf{X}$ is a $3-$dimensional vector of independent marginals uniformly distributed on $[-\pi, \pi]$. In Figure C.7, the failure points (where $G(\mathbf{x}) < 0$) are plotted in a 3-d scatterplot.

There are 614 failure points on a MC sample of 10^5 points therefore the failure probability here is roughly $\widehat{P} = 6.14.10^{-3}$. The complex repartition of the failure points can be noticed. Those points lay in a zone defined by the negative values of X_1, the extremal and mean values of X_2 (around $-\pi$, 0 and π), and the extremal values of X_3 (around $-\pi$ and π).

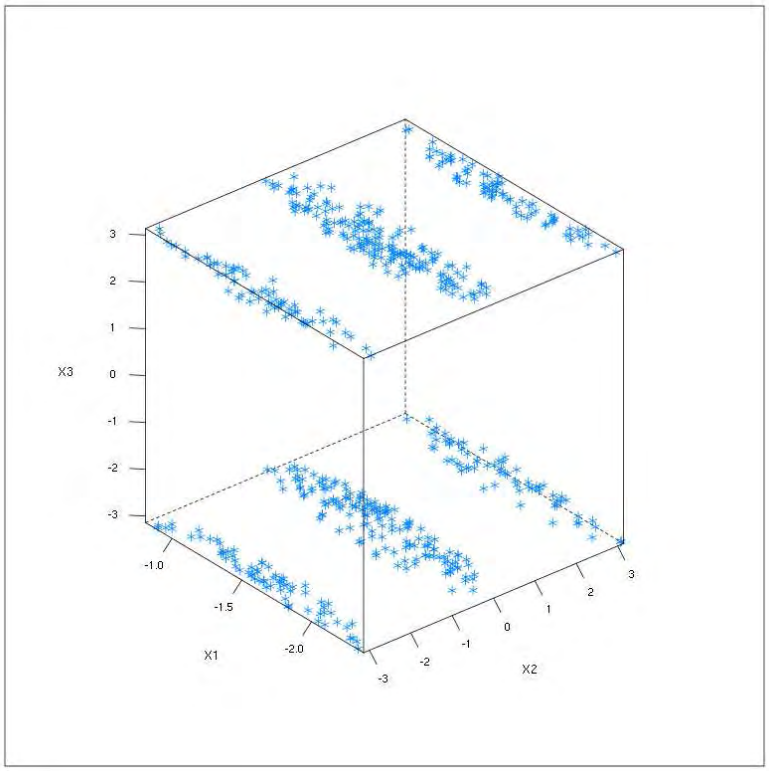


Figure C.7: Ishigami failing points from a MC sample

Variable	X_1	X_2	X_3
Importance factor	$1e^{-17}$	1	0

Table C.3: Importance factors for Ishigami function

Sobol' Index	S_1	S_2	S_3	S_{T1}	S_{T2}	S_{T3}
Mean	0.0234	0.0099	0.0667	0.8158	0.6758	0.9299
CV	0.0072	0.0051	0.0095	0.0156	0.0216	0.0094

Table C.4: Sobol' indices for Ishigami function

C.4.2.1 FORM

The algorithm FORM converges to an incoherent design point $(6.03, 0.1, 0)$ in 50 function calls, giving an approximate probability of $\widehat{P}_{FORM} = 0.54$. The importance

factors are displayed in Table C.3. The bad performance of FORM is expected given that the failure domain consists in six separate domains and that the function is highly oscillant, leading to optimization difficulties. The design point is absurd, thus the FORM results of SA are incorrect.

C.4.2.2 Sobol' indices

The first-order and total indices are displayed in Table C.4. The small values of first order indices show that no variable has impact on the variance of the indicator of failure on its own. The relatively high and similar value of the three total index states that all the variables highly interact with each other to cause system failure. The low CV shows that the method is reproducible.

C.4.2.3 Density modification based reliability indices

The method presented throughout this article is applied on the thresholded Ishigami function. As for the hyperplane test case, a mean twisting and a variance twisting are applied. When twisting an uniform distribution by shifting the mean and keeping the same support, the modified distribution (given by Proposition C.3.2) is an exponential distribution with a mode situated on one endpoint of the support. The modified density when shifting the variance and keeping the same expectation is proportional to a truncated Gaussian when decreasing the variance. When increasing the variance, the perturbed distribution is a symmetrical distribution with 2 modes close to the endpoints of the support. As previously, the same MC sample of size 10^5 (also used to produce Figure C.7) is used to estimate the indices with both perturbations. For the mean twisting (see (C.11)), the variation range chosen for δ is -3 to 3 with 60 points - numerical consideration forbidding to choose a shifted mean closer to the endpoints. For variance twisting, the variation range chosen for V_{per} is 1 to 5 with 40 points, recall that the original variance is $\text{Var}[X_i] = \pi^2/3 \simeq 3.29$. The estimated indices are plotted respectively in Figure C.8 for mean twisting and in Figure C.9 for variance twisting.

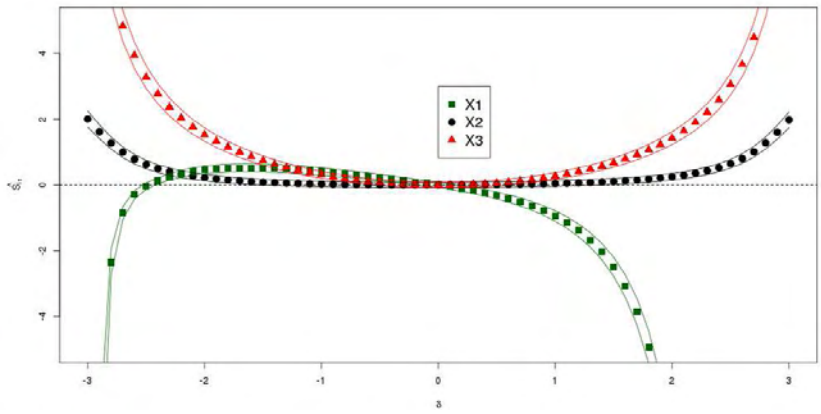


Figure C.8: Estimated indices $\widehat{S_{i\delta}}$ for thresholded Ishigami function with a mean twisting

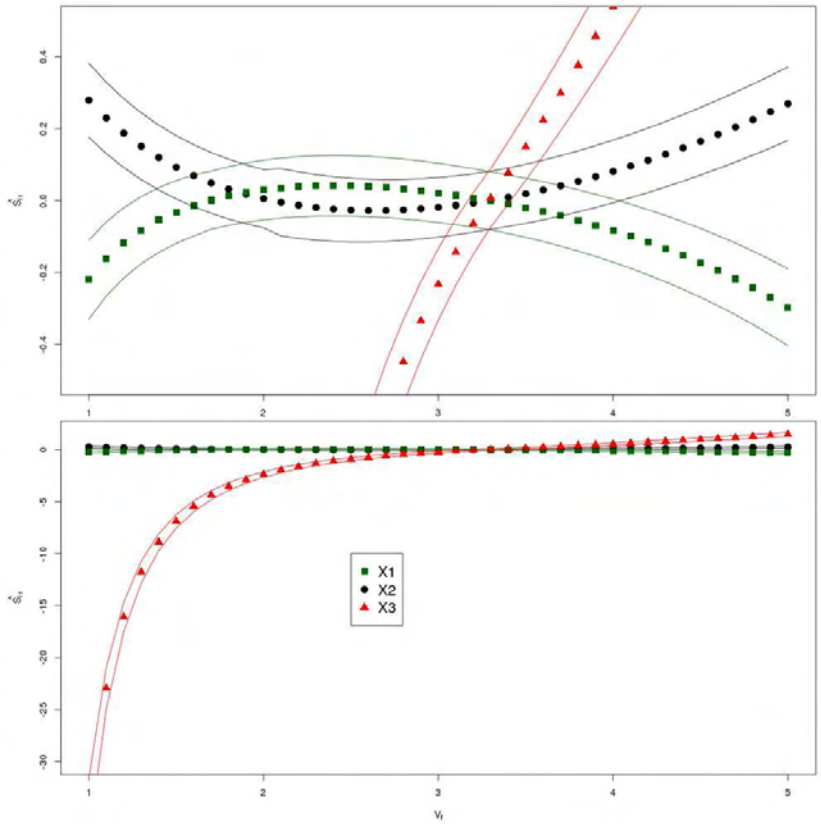


Figure C.9: Estimated indices $\widehat{S_{i,V_{\text{per}}}}$ for thresholded Ishigami function with a variance twisting

Mean perturbation indices One can see on Figure C.8 that a perturbation of the mean for X_2 and X_3 will increase the failure probability, though the impact for the same mean perturbation is stronger for X_3 ($\widehat{S_{3,-3}}$ and $\widehat{S_{3,3}}$ approximatively equal respectively 9.5 and 10). On the other hand, the indices concerning X_1 show that a mean shift between -1 and -2 increases the failure probability, whereas an augmentation of the mean or a large diminution strongly diminishes the failure probability ($\widehat{S_{1,3}}$ approximatively equals -7.10^{11}). Therefore, Figure C.8 leads to two conclusions. Firstly the failure probability can be strongly reduced when shifting the mean of the first X_1 (this is also provided by Figure C.7 wherein all failing points have a negative value of X_1). Secondly any change in the mean for v X_2 or X_3 will lead to an increase of the failure probability.

Variance perturbation indices Figure C.9 (upper) shows that a change in the variance has little effect on X_2 and X_1, though the change is of opposite effect on the failure probability. However, considering that the indices $\widehat{S_{2,V_{\text{per},i}}}$ and $\widehat{S_{1,V_{\text{per},i}}}$ lies between -0.4 and 0.4, one can conclude that the variance of theses variables are not of great influence on the failure probability. On the other hand, Figure C.9 (lower) shows that any reduction of the variance of X_3 strongly decreases the failure probability, and that an increase of the variance slightly increases the failure probability. This is relevant with the expression of the failure surface, as X_3 is fourth powered and multiplied by the sinus of X_1. A variance diminution as formulated gives a distribution concentrated around 0; thus decreasing the variance of X_3 shrinks the term containing it in $G(\mathbf{X})$, therefore reducing the failure probability.

C.4.3 Industrial case : flood case

The goal of this test case is to assess the risk of a flood over a dyke for the safety of industrial installations. This comes down to model the maximum height of a flood. Given the uncertainty upon numerous physical parameters, the uncertainty approach is used and unknown parameters are modelled by RV. From a simplification of the Saint-Venant equation, a flood risk mode is obtained. The quantity of interest is the difference between the height of the dyke and the height of water. If this quantity is negative, the installation is flooded; this is the failure event that is considered. Several quantities will be denoted as follows: Q the flow rate, L the watercourse section length studied, B the watercourse width, K_s the watercourse bed friction

Variable	Q	K_s	Z_v	Z_m
Importance factor	0.246	0.725	0.026	0.003

Table C.5: Importance factors for flood case

coefficient (also called Strickler coefficient), Z_m and Z_v respectively the upstream and downstream bottom watercourse height above sea level and H_d the dyke height measured from the bottom of the watercourse bed. The water height model is expressed as:

$$H = \left(\frac{Q}{K_s B \sqrt{\frac{Z_m - Z_v}{L}}} \right)^{\frac{3}{5}}$$

Therefore the following quantity is considered:

$$G = H_d - (Z_v + H).$$

Among the model inputs, the choice is made that the following variables are known precisely: $L = 5000$ (m), $B = 300$ (m),$H_d = 58$ (m), and the following are considered to be random. Q ($\mathrm{m}^3.\mathrm{s}^{-1}$) follows a positively truncated Gumbel distribution of parameters $a = 1013$ and $b = 558$ with a minimum value of 0. $K_s(\mathrm{m}^{1/3}\mathrm{s}^{-1})$ follows a truncated Gaussian distribution of parameters $\mu = 30$ and $\sigma = 7.5$, with a minimum value of 1. Z_v (m) follows a triangular distribution with minimum 49, mode 50 and maximum 51. Z_m (m) follows a triangular distribution with minimum 54, mode 55 and maximum 56.

C.4.3.1 FORM

The algorithm FORM converges to a design point $(1.72, -2.70, 0.55, -0.18)$ in 52 function calls, giving an approximate probability of $\widehat{P}_{FORM} = 5.8 * 10^{-4}$. The importance factors are displayed in Table C.5.

FORM assesses that K_s is of extremely high influence, followed by the Q that is of medium influence. Z_v has a very weak influence and Z_m is negligible. It can be noticed that the estimated failure probability is twice as small as the one estimated with crude MC, but remains in the same order of magnitude.

Sobol Index	S_Q	S_{K_s}	S_{Z_v}	S_{Z_m}	S_{TQ}	S_{TK_s}	S_{TZ_v}	S_{TZ_m}
Mean	0.0169	0.2402	$-7 * 10^{-5}$	$-5 * 10^{-4}$	0.7447	0.9782	0.2684	0.1062
CV	0.0122	0.0577	0.0029	0.0023	0.0553	0.0137	0.0516	0.0389

Table C.6: Sobol' indices for flood case

C.4.3.2 Sobol' indices

The first-order and total indices are displayed in Table C.6. It can be seen that the estimate of some indices is negative despite the fact that Sobol indices are theoretically positive. The estimation can indeed produce negative results for values close to 0.

Considering the first order indices, Z_v and Z_m are of null influence on their own. Q is considered to have a minimal influence (1% of the variance of the indicator function) by itself, and K_s explains 24% of the variance on its own. When considering the total indices, it can be noticed that both Z_v and Z_m have a weak impact on the failure probability. On the other hand, Q has a major influence on the failure probability. K_s total index is close to one, therefore K_s explains (with or without any interaction with other variables) almost all the variance of the failure function.

C.4.3.3 Density modification based reliability indices

The method presented throughout this article is applied on the flood case. Given some numerical difficulties to converge to an optimal solution satisfying (C.12), only the mean twisting will be applied here. One can notice that the different inputs are follow various distributions (unlike the other examples), thus the question of "equivalent" perturbation arises, and will be discussed further in Section C.5. Here the choice have been made to shift the mean relatively to the standard deviation, hence including the spread of the various inputs in their respective perturbation. So for any input, the original distribution is twisted so that the perturbed distribution's mean is the original's one plus δ times its standard deviation, δ going from -1 to 1 with 40 points. The 10^5 MC sample gives an estimation of the failure probability $\widehat{P} = 8.6 * 10^{-4}$.

Figure C.10 assesses that a increasing of the mean of the inputs increases the failure probability slightly for Z_v, strongly for Q, and diminishes it slightly for Z_m and strongly for K_s. This goes the opposite way when decreasing the mean. In term of absolute modification, K_s and Q are of same magnitude, even if K_s has a slightly

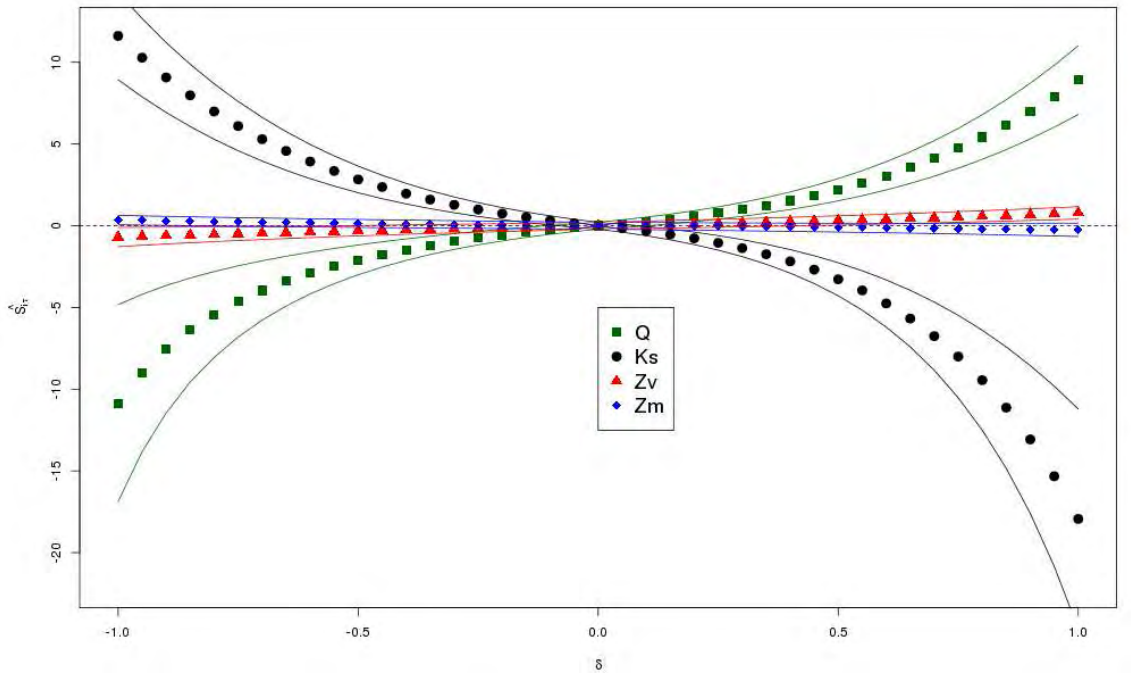


Figure C.10: Estimated indices $\widehat{S_{i\delta}}$ for flood case with a mean twisting

strongest impact. On the other hand, the effects of mean perturbation on Z_m and Z_v are practically negligible.

C.5 Discussion

C.5.1 Equivalent perturbation

The question of "equivalent" perturbation arises from cases where all inputs are not identically distributed, such as in Subsection C.4.3.3. Indeed, problems may emerge when some inputs are defined on infinite intervals and when other inputs are defined on finite intervals (such as uniform distributions). Consider a model with one Gaussian distribution and one uniform distribution as inputs. Thus, a mean shift will be a translation for the first input, whereas it will lead to a Dirac distribution in one endpoint for the other input. Hence, not any shift mean can be considered as an "equivalent" perturbation. One could think of a "relative mean shift", which seems a fairly good idea. But let one consider a model with two Gaussian inputs of equal variance 1 and of mean respectively 1 and 10000. Then, a relative mean

shift of 10% will result in Gaussian distributions with mean respectively 1.1 and 11000, and still variance 1. This counter-example shows that relative mean shift might not be an adequate perturbation in term of "equivalence". The solution presented in Subsubsection C.4.3.3 tries to take into account the spread of the inputs distributions, but would have failed on the counter-example given above.

C.5.2 Conclusion on the method

The method presented in this paper gives interesting complementary information in addition of traditional SA methods applied to a reliability problem. Additionally, it has two advantages:

- The ability for the user to set the most adapted constraints considering his/her problem.
- The MC framework allows to use previously done function calls, thus limiting the CPU cost of the SA, and allowing the user to test several perturbations.

C.5.3 Further work

Two main avenues are of interest.

- To obtain better estimator of the failure probability P, in term of variance reduction and of number of function calls. Further work will be made with importance sampling methods, and possibly subset methods:
- To find a way to perturb "equivalently" several distributions of different natures. For this, a perturbation that is not based upon a moment constraint but rather of an entropy constraint. The differential entropy of a distribution can be seen as a quantification of uncertainty Auder and Iooss [2009]. Thus an example of (non-linear) constraint on the entropy can be :

$$-\int f_{X_{i\delta}}(x)\log f_{X_{i\delta}}(x)dx = -\delta\int f_{X_i}(x)\log f_{X_i}(x)dx.$$

  Yet further computations have to be made to obtain a tractable solution of the KL minimization problem under the above constraint.

Acknoweldgements

Part of this work has been backed by French National Research Agency (ANR) through COSINUS program (project COSTA BRAVA noANR-09-COSI-015). We thank Dr. Daniel Busby (IFP EN) for several discussions. We also thank Emmanuel Remy (EDF R&D) for proofreading.

C.6 Appendix 1: Proofs

C.6.1 Proof of Lemma C.2.1

Under assumption **(i)**,

$$\int_{\mathrm{Supp}(f_{i\delta})} \mathbb{1}_{\{G(\mathbf{X})<0\}} \frac{f_{i\delta}(x_i)}{f_i(x_i)} f(x)\, dx \leq \int_{\mathrm{Supp}(f_{i\delta})} f_{i\delta}(x)\, dx \;=\; 1.$$

So that, the strong LLN may be applied to $\widehat{P}_{i\delta N}$. Defining

$$\sigma_{i\delta}^2 = \mathrm{Var}\left[\mathbb{1}_{\{G(\mathbf{X})<0\}} \frac{f_{i\delta}(X_i)}{f_i(X_i)}\right],$$

one has

$$\sigma_{i\delta}^2 \;=\; \int_{\mathrm{Supp}(f_i)} \mathbb{1}_{\{G(\mathbf{x})<0\}} \frac{f_{i\delta}^2(x_i)}{f_i(x_i)} \prod_{j\neq i}^{N} f_j(x_j)\, d\mathbf{x} - P_{i\delta}^2 \;<\; \infty \quad \text{under Condition } \textbf{(ii)}.$$

Therefore the CLT applies:

$$\sqrt{N}\sigma_{i\delta}^{-1}\left(\widehat{P}_{i\delta N} - P_{i\delta}\right) \xrightarrow{\mathcal{L}} \mathcal{N}(0,1)\,.$$

Under assumption **(ii)**, the strong LLN applies to $\widehat{\sigma}_{i\delta N}^2$. So that, the final result is straightforward using Slutsky's lemma.

C.6.2 Proof of Proposition C.2.2

First, note that

$$\begin{aligned}
\mathbb{E}\left[\widehat{P}\widehat{P_{i\delta}}\right] - PP_{i\delta} &= \mathbb{E}\left[\frac{1}{N^2}\left(\sum_{n=1}^{N}\mathbb{1}_{\{G(\mathbf{x}^n)<0\}}\right)\left(\sum_{n=1}^{N}\mathbb{1}_{\{G(\mathbf{x}^n)<0\}}\frac{f_{i\delta}(x_i^n)}{f_i(x_i^n)}\right)\right] - PP_{i\delta} \\
&= \frac{1}{N^2}\mathbb{E}\left[\sum_{n=1}^{N}\left[\mathbb{1}_{\{G(\mathbf{x}^n)<0\}}\right]^2\frac{f_{i\delta}(x_i^n)}{f_i(x_i^n)} + \sum_{n=1}^{N}\sum_{j\neq i}^{N}\mathbb{1}_{\{G(\mathbf{x}^n)<0\}}\mathbb{1}_{\{G(\mathbf{x}^j)<0\}}\frac{f_{X_{i\delta}}(x_i^j)}{f_{X_i}(x_i^j)}\right] \\
&\quad -PP_{i\delta} \\
&= \frac{1}{N^2}\left[NP_{i\delta} + N\left(N-1\right)PP_{i\delta}\right] - PP_{i\delta} \\
&= \frac{1}{N}\left(P_{i\delta} - PP_{i\delta}\right) \ .
\end{aligned}$$

Assuming the conditions under which Lemma 1 is true, the bivariate CLT follows with

$$\Sigma_{i\delta} = \begin{pmatrix} P(1-P) & P_{i\delta}(1-P) \\ P_{i\delta}(1-P) & \sigma_{i\delta}^2 \end{pmatrix} .$$

Each term of this matrix can be consistently estimated, using the results in Lemma 1 and Slutsky's lemma.

C.7 Appendix: Computation of Lagrange multipliers

Let H be the Lagrange function:

$$H(\boldsymbol{\lambda}) = \psi_i(\boldsymbol{\lambda}) - \sum_{k=1}^{K}\lambda_k\delta_k$$

Thus, using the results of Csiszár [1975]

$$\boldsymbol{\lambda}^* = \arg\min H(\boldsymbol{\lambda})$$

The expression of the gradient of H with respect to the j^{th} variable is

$$\nabla_j H(\boldsymbol{\lambda}) = \frac{\int g_j(x) f_i(x) \exp(\sum_{k=1}^{K} \lambda_k g_k(x)) dx}{\exp \psi_i(\boldsymbol{\lambda})} - \delta_j$$

In the same way, the expression of the second derivative of H with respect to the h^{th} and the j^{th} variable is

$$D_{hj} H(\boldsymbol{\lambda}) = \frac{\int g_h(x) g_j(x) f_i(x) \exp(\sum_{k=1}^{K} \lambda_k g_k(x)) dx}{\exp \psi_i(\boldsymbol{\lambda})} - \frac{\int g_j(x) f_i(x) \exp(\sum_{k=1}^{K} \lambda_k g_k(x)) dx}{\exp \psi_i(\boldsymbol{\lambda})} \frac{\int g_h(x) f_i(x) \exp(\sum_{k=1}^{K} \lambda_k g_k(x)) dx}{\exp \psi_i(\boldsymbol{\lambda})}$$

Although any gradient descent method would theoretically converge to the optimal solution $\boldsymbol{\lambda}^*$, the expressions of $\nabla_j H(\boldsymbol{\lambda})$ and $D_{hj} H(\boldsymbol{\lambda})$ may not be tractable, thus leading to computational problems.

References

P. Abrahamsen. *A review of Gaussian random fields and correlation functions*. Norsk Regnesentral/Norwegian Computing Center, 1997.

A. Antoniadis. Analysis of variance on function spaces. *Math. Operationsforsch. Statist. Ser. Statist.*, 15(1):59–71, 1984. ISSN 0323-3944. doi: 10.1080/02331888408801747. URL http://dx.doi.org/10.1080/02331888408801747.

S. K. Au and J. L. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, 2001.

S. K. Au and J. L. Beck. Subset simulation and its application to seismic risk based on dynamic analysis. *Journal of Engineering Mechanics*, 129(3):2901–2917, 2003.

B. Auder and B. Iooss. Global sensitivity analysis based on entropy. 2009.

B. Auder, A. De Crecy, B. Iooss, and M. Marques. Screening and metamodeling of computer experiments with functional outputs. application to thermal-hydraulic computations. *Reliability Engineering & System Safety*, 2011.

M. J. Bayarri, J. O. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R. J. Parthasarathy, R. Paulo, J. Sacks, and D. Walsh. Computer model validation with functional output. *The Annals of Statistics*, 35(5):1874–1906, 2007.

R.J. Beckman and M.D McKay. Monte-Carlo estimation under different distributions using the same simulation. *Technometrics*, 29(2):153–160, 1987.

J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.

S. M. Benson. Assessment of risks from storage of carbon dioxide in deep underground geological formations. Technical report, Lawrence Berkley National Laboratory, Earth Science Division, April 2006.

B. J. Bichon, J. M. McFarland, and S. Mahadevan. Efficient surrogate models for reliability analysis of systems with multiple failure modes. *Reliability Engineering and System Safety*, 96(10):1386–1395, 2011.

P. Bjerager. Probability integration by directional simulation. *Journal of Engineering Mechanics*, 114(8):1285–1302, 1988.

E. Borgonovo, W. Castaings, and S. Tarantola. Moment independent importance measures: New results and analytical test cases. *Risk Analysis*, 31(3):404–428, 2011.

J. M. Bourinet. Ferum 4.1 user's guide, July 2010. IFMA.

A. R. Bowden and A. Rigg. Assessing reservoir performance risk in CO_2 storage projects. In *Proceedings of 7th International Conference on Greenhouse Gas Control Technologies*, 2004.

L. C. Brumback and M. J. Lindstrom. Self modeling with flexible, random time transformations. *Biometrics*, 60(2):461–470, 2004.

D. Busby and M. Feraille. Adaptive design of experiments for calibration of complex simulators–an application to uncertainty quantification of a mature oil field. In *Journal of Physics: Conference Series*, volume 135, pages 12–26. IOP Publishing, 2008.

D. Busby and E. Sergienko. Combining probabilistic inversion and multi-objective optimization for production development under uncertainty. In *Proceedings of 12th European Conference in the Mathematics in Oil Recovery*, September 2010.

D. Busby, C. L. Farmer, and A. Iske. Hierarchical nonlinear approximation for experimental design and statistical data fitting. *SIAM Journal on Scientific Computing*, 29(1):49, 2008.

A. Busch, A. Amann, P. Bertier, M. Waschbusch, and B. M. Kroos. The significance of caprock sealing integrity for CO_2 storage. *SPE*, 139558, 2010.

K. Campbell, M. D. McKay, and B. J. Williams. Sensitivity analysis when model outputs are functions. *Reliability Engineering & System Safety*, 91(10):1468–1472, 2006.

N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3): 539–552, 2002.

J. Condor, D. Unatrakarn, M. Wilson, and K. Asghari. A comparative analysis of risk assessment methodologies for the geologic storage of carbon dioxide. *Energy Procedia*, 4:4036–4043, 2011.

S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of statistical planning and inference*, 140(3):640–651, 2010.

S. Conti, J. P. Gosling, J. E. Oakley, and A. O'Hagan. Gaussian process emulation of dynamic computer codes. *Biometrika*, 96(3):663–676, 2009.

R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, and D.E. Knuth. On the lambertw function. *Advances in Computational mathematics*, 5(1):329–359, 1996.

T.M. Cover and J.A. Thomas. Elements of information theory 2nd edition (Wiley series in telecommunications and signal processing). 2006.

I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, pages 146–158, 1975.

E. De Rocquigny, N. Devictor, and S. Tarantola. *Uncertainty in industrial practice*. Wiley Online Library, 2008.

O. Ditlevsen and H. O. Madsen. *Structural reliability methods*, volume 178. Wiley Chichester, 1996.

O. Ditlevsen, P. Bjerager, R. Olesen, and A. M. Hasofer. Directional simulation in gaussian processes. *Probabilistic Engineering Mechanics*, 3(4):207–217, 1988.

S. S. Dragomir and V. Glušcević. Some inequalities for the Kullback-Leibler and χ^2-distances in information theory and applications. Article 3 3(2), Victoria University of Technology, 2000. RGMIA.

X. Du. First and second order reliability methods. University of Missouri, September 2005. Chapter 7.

V. Dubourg. *Adaptive surrogate models for reliability analysis and reliability-based design optimization*. PhD thesis, Université Blaise Pascal - Clermont II, Laboratoire de Mécaniaue et Ingénieries, December 2011.

V. Dubourg, B. Sudret, and J. M. Bourinet. Reliability-based design optimization using kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44(5):1–18, 2011.

B. Echard, N. Gayton, and M. Lemaire. Ak-mcs: An active learning reliability method combining kriging and monte carlo simulation. *Structural Safety*, 33(2): 145–154, 2011.

A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, 2008.

A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, 2009.

F. Gamboa, J. M. Loubes, and E. Maza. Semi-parametric estimation of shifts. *Electronic Journal of Statistics*, 1:616–640, 2007.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

C. Hawkes, P. Mclellan, and S. Bachu. Geomechanical factors affecting geological storage of CO_2 in depleted oil and gas reservoirs. *Journal of Canadian Petroleum Technology*, 44(10), 2005.

T.C. Hesterberg. Estimates and confidence intervals for importance sampling sensitivity analysis. *Mathematical and computer modelling*, 23(8):79–85, 1996.

D. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.

B. Iooss. Revue sur l'analyse de sensibilité globale de modèles numériques. *Journal de la Société Française de Statistique*, 152(1):1–23, 2011.

R. Izem and J. S. Marron. Analysis of nonlinear modes of variation for functional data. *Electronic Journal of Statistics*, 1:641–676, 2007.

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

I. Kaymaz. Application of kriging method to structural reliability problems. *Structural Safety*, 27(2):133–151, 2005.

M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

A. Kopp, P. J. Binning, K. Johannsen, R. Helmig, and H. Class. A contribution to risk analysis for leakage through abandoned wells in geological CO_2 storage. *Advances in Water Resources*, 33(8):867–879, 2010.

A. Korre, J. Q. Shi, C. Imrie, C. Grattoni, and S. Durucan. Coalbed methane reservoir data and simulator parameter uncertainty modelling for CO_2 storage performance assessment. *International Journal of Greenhouse Gas Control*, 1(4): 492–501, 2007.

D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society*, 52: 119–139, 1951.

M. Lamboni, D. Makowski, S. Lehuger, B. Gabrielle, and H. Monod. Multivariate global sensitivity analysis for dynamic crop models. *Field Crops Research*, 113(3): 312–320, 2009.

W. H. Lawton, E. A. Sylvestre, and M. S. Maggio. Self modeling nonlinear regression. *Technometrics*, pages 513–532, 1972.

Y. Le Guen, J. Le Gouvec, and R. Chammas. CO_2 storage: managing the risk associated with well leakage over long timescales. In *Proceedings SPE Asia Pacific Oil and Gas Conference and Exhibition*, October 2008.

M. Lemaire, A. Chateauneuf, and J. C. Mitteau. *Structural reliability*. Wiley Online Library, 2009.

P. Lemaitre and A. Arnaud. Hiérarchisation des sources d'incertitudes vis à vis d'une probabilité de dépassement de seuil - une méthode basée sur la pondération des lois. In *Proceedings des 43 èmes Journées de Statistique*, Tunis, Tunisie, June 2011.

P. Lemaitre, E. Sergienko, F. Gamboa, A. Arnaud, N. Bousquet, and B. Iooss. Density modification based reliability sensitivity indices. *Submitted to Journal of Statistical Computation and Simulation*, 2012.

H. O. Madsen, S. Krenk, and N. C. Lind. *Methods of structural safety*. Dover publications Mineola, NY, 2006.

G. Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

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*, pages 239–245, 1979.

N. J. McMillan, J. Sacks, W. J. Welch, and F. Gao. Analysis of protein activity data by gaussian stochastic process models. *Journal of Biopharmaceutical Statistics*, 9 (1):145–160, 1999.

R. E. Melchers. Importance sampling in structural systems. *Structural safety*, 6(1): 3–10, 1989.

X. L. Meng and W. H. Wong. Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, 6:831–860, 1996.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

M. R. Moarefzadeh and R. E. Melchers. Directional importance sampling for ill-proportioned spaces. *Structural safety*, 21(1):1–22, 1999.

J. Morio. Influence of input pdf parameters of a model on a failure probability estimation. *Simulation Modelling Practice and Theory*, 19(10):2244–2255, 2011.

M. Munoz Zuniga, J. Garnier, E. Remy, and E. de Rocquigny. Adaptive directional stratification for controlled estimation of the probability of a rare event. *Reliability Engineering & System Safety, in Press. doi*, 10, 2011.

R. M. Neal. Estimating ratios of normalizing constants using linked importance sampling. Department of statistics and department of computer science, University of Toronto, November 2005.

V. Picheny. *Improving accuracy and compensating for uncertainty in surrogate modeling*. PhD thesis, Ecole Nationale Superieure des Mines de Saint-Etienne and University of Florida, December 2009.

V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, and N. H. Kim. Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132(7), 2010.

D. Polson, A. Curtis, C. Vivalda, and S. Saunier. Process for tracking the evolving perception of risk during CO_2 storage projects. In *Proceedings of Offshore Europe*, Septermber 2009.

PUNQ-S3. Production forecasting with uncertainty quantification. Website, 1996. URL `http://www.fault-analysis-group.ucd.ie/Projects/PUNQ.html`.

P. Z. G. Qian and C. F. J. Wu. Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50(2):192–204, 2008.

P. Z. G. Qian, H. Wu, and C. F. J. Wu. Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics*, 50(3):383–396, 2008.

B. D. Ripley. *Stochastic simulation*, volume 183. Wiley Online Library, 1987.

J. Rohmer and O. Bouc. A response surface methodology to address uncertainties in cap rock failure assessment for CO_2 geological storage in deep aquifers. *International Journal of Greenhouse Gas Control*, 4(2):198–208, 2010.

M. Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952.

R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*, volume 707. Wiley-interscience, 2008.

J. Rutqvist, J. Birkholzer, F. Cappa, and C. F. Tsang. Estimating maximum sustainable injection pressure during geological sequestration of CO_2 using coupled

fluid flow and geomechanical fault-slip analysis. *Energy Conversion and Management*, 48(6):1798–1807, 2007.

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.

A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto. Sensitivity analysis in practice: A guide to assessing scientific models. 2004. *Chichester, England: John Wiley & Sons, Ltd. Steven Batill (2) Dept. of Aerospace & Mechanical Engr. University of Notre Dame, IN*, 46556:48090–9055.

A. Saltelli, K. Chan, E.M. Scott, et al. *Sensitivity analysis*. Wiley New York, 2000.

T. J. Santner, B. J. Williams, and W. Notz. *The design and analysis of computer experiments*. Springer Verlag, 2003.

E. Sergienko and D. Busby. Optimal well placement for risk mitigation in CO_2 storage. In *Proceedings of 1st Sustainable Earth Sciences Conference*, November 2011.

M. Shinozuka. Basic analysis of structural safety. *Journal of Structural Engineering*, 109(3):721–740, 1983.

I. M. Sobol'. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling and Computational Experiments*, 1:407–414, 1993.

I. M. Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55:271–280, 2001.

S. Subbey, M. Christie, and M. Sambridge. Prediction under uncertainty in reservoir modeling. *Journal of Petroleum Science and Engineering*, 44(1):143–153, 2004.

A. A. Taflanidis and J. L. Beck. Stochastic subset optimization for optimal reliability problems. *Probabilistic Engineering Mechanics*, 23(2):324–338, 2008.

A. A. Taflanidis and J. L. Beck. Stochastic subset optimization for reliability optimization and sensitivity analysis in system design. *Computers & Structures*, 87 (5-6):318–331, 2009.

E. Tillier, A. Michel, and L. Trenty. Coupling a multiphase flow model and a reactive transport model for CO_2 storage modeling. In *Proceedings Computational Methods for Coupled Problems in Science and Engineering*, 2007.

L. Trenty, A. Michel, E. Tillier, and Y. Le Gallo. A sequential splitting strategy for CO_2 storage modelling. In *Proceedings of 10th European Conference on the Mathematics of Oil Recovery*. EAGE, 2006.

A. W. Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge University Press, 2000.

M. Vimond. Efficient estimation for a subclass of shape invariant models. *The Annals of Statistics*, 38(3):1885–1912, 2010.

W. J. Welch, Robert. J. Buck, J. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris. Screening, predicting, and computer experiments. *Technometrics*, pages 15–25, 1992.

K. M. Zuev, J. L. Beck, S. K. Au, and L. S. Katafygiotis. Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. *Computers and Structures*, 92-93:283–296, 2011.