Sequential Design of Experiments to Estimate a Probability of Failure.

Ling Li

Supélec

STiTS ECOLE DOCTORALE

N° d'ordre: 2012-06-TH

SUPÉLEC

THÈSE DE DOCTORAT

DOMAINE : STIC

SPECIALITÉ : Traitement du Signal

École Doctorale Sciences et Technologies de l'Information,
des Télécommunications et des Systèmes

Présentée par

Ling LI

Sujet :

Sequential Design of Experiments to Estimate a Probability of Failure

Soutenue le 16 Mai 2012 devant les membres du Jury:

M.	BECT	Julien	Encadrant
M.	FLEURY	Gilles	Directeur de thèse
M.	IOOSS	Bertrand	Rapporteur
M.	PRONZATO	Luc	Examinateur
M.	SUDRET	Bruno	Rapporteur
M.	VAZQUEZ	Emmanuel	Encadrant
M.	WALTER	Eric	Président

Abstract

This thesis deals with the problem of estimating the *probability of failure* of a system from computer simulations. When only an expensive-to-simulate model of the system is available, the budget for simulations is usually severely limited, which is incompatible with the use of classical Monte Carlo methods. In fact, estimating a small probability of failure with very few simulations, as required in some complex industrial problems, is a particularly difficult topic. A classical approach consists in replacing the expensive-to-simulate model with a surrogate model that will use little computer resources. Using such a surrogate model, two operations can be achieved. The first operation consists in choosing a number, as small as possible, of simulations to learn the regions in the parameter space of the system that will lead to a failure of the system. The second operation is about constructing good estimators of the probability of failure. The contributions in this thesis consist of two parts. First, we derive *SUR* (*stepwise uncertainty reduction*) strategies from a Bayesian-theoretic formulation of the problem of estimating a probability of failure. Second, we propose a new algorithm, called *Bayesian Subset Simulation*, that takes the best from the Subset Simulation algorithm and from sequential Bayesian methods based on Gaussian process modeling. The new strategies are supported by numerical results from several benchmark examples in reliability analysis. The methods proposed show good performances compared to methods of the literature.
Keywords: Computer experiments; Reliability analysis; Probability of failure; Gaussian process model; SUR strategy; Subset Simulation

Résumé

Cette thèse aborde le problème de l'estimation de la *probabilité de défaillance* d'un système à partir de simulations informatiques. Lorsqu'on dispose seulement d'un modèle du système coûteux à simuler, le budget de simulations est généralement très limité, ce qui est incompatible avec l'utilisation de méthodes Monte Carlo classiques. En fait, l'estimation d'une petite probabilité de défaillance à partir de simulations très coûteuses, comme on peut le rencontrer dans certains problèmes industriels complexes, est un sujet particulièrement difficile. Une approche classique consiste à remplacer le modèle coûteux à simuler par un modèle de substitution

nécessitant de faibles ressources informatiques. A partir d'un tel modèle de substitution, deux opérations peuvent être réalisées. La première opération consiste à choisir des simulations, en nombre aussi petit que possible, pour apprendre les régions de l'espace des paramètres du système qui conduisent à une défaillance du système. La deuxième opération consiste à construire de bons estimateurs de la probabilité de défaillance. Cette thèse propose deux contributions. Premièrement, nous proposons des stratégies de type *SUR* (*Stepwise Uncertainty Reduction*) à partir d'une formulation bayésienne du problème d'estimation d'une probabilité de défaillance. Deuxièmement, nous proposons un nouvel algorithme, appelé *Bayesian Subset Simulation*, qui prend le meilleur de l'algorithme *Subset Simulation* et des approches séquentielles bayésiennes utilisant la modélisation du système par processus gaussiens. Ces nouveaux algorithmes sont illustrés par des résultats numériques concernant plusieurs exemples de référence dans la littérature de la fiabilité. Les méthodes proposées montrent de bonnes performances par rapport aux méthodes concurrentes.

Mots-clés: Simulation; Analyse de fiabilité; Probabilité de défaillance; Processus gaussien; Stratégie SUR; Simulation de sous-ensembles

Acknowledgments

I owe tons of gratitude to my co-supervisors, Prof. Julien Bect and Prof. Emmanuel Vazquez, for their ongoing guidance and great patience. Working with you both these past three years has been an unforgettable experience for my whole life. I respect your remarkable talents. Special thanks to Mr. Gilles Fleury, the director of my thesis, for his kind encouragement and support.

I would also like to thanks the members of my committee, Mr. Bertrand IOOSS, Mr. Bruno SUDRET, Mr. Luc PRONZATO, Mr. Eric WALTER, for their valuable suggestions. I am grateful for their kindness serving on my committee, for reviewing my manuscript and for giving constructive suggestions and criticisms that help me to improve this work. Particularly, I would like to thank Mr. IOOSS and Mr. SUDRET for their reviewing as rapporteurs of my dissertation.

Many thanks to my colleagues in Signal processing and System electronics department at Supélec. Karine, Romain, Jingwen, Grégory, Régis .... – it is great to work with you guys and you have been good accompanies! Thanks also to other friends who have studied here, lived here...

Finally, I give my special thank you to my family and friends in China and David, without your support and understanding, all this couldn't have happened.

Contents

LIST OF ACRONYMS

Acronyms

MCS	Monte Carlo sampling
MPP	most probable point
MLP	most likelihood point
FORM	first order reliability method
SORM	second order reliability method
FO/SO SPA	first/second order saddle point approximation
LHS	Latin hyper-cube sampling
DS	directional sampling
IS	importance sampling
AIS	adaptive importance sampling
SIS	sequential importance sampling
IIS	iterative importance sampling
MCMC	Markov Chain Monte Carlo
SMC	sequential Monte Carlo
SVM	support vector machine
MSE	mean square error
SUR	step-wise uncertainty reduction
GP	Gaussian process
IAGO	informational approach to global optimization
EGO	efficient global optimization
tIMSE	target integrated mean square error
MLE	maximum likelihood estimator
REML	restricted maximum likelihood
BLUP	best linear unbiased predictor
BSS	Bayesian subset simulation
i.i.d.	independent identity distribution
w.r.t.	with respect to
DoE	Design of experiment

Chapter 1

Introduction

1.1 Context

1.1.1 Reliability analysis with numerical simulations

In engineering, the design of a system has to take into account the fact that some design parameters are subject to variations during manufacturing and that the conditions of operation of the system are not necessarily known in advance. The sources of uncertainty may affect the safety of the system. *Reliability analysis* aims at quantifying the chance that a system encounters a dangerous situation and the main problem addressed in this domain is that of estimating the *probability of failure* of a system.

Reliability analysis has been intensively explored in many domains, such as safety studies for building structures (Schueremans, 2001), dam safety analysis (Salmon and Hartford, 1995), seismic safety studies for nuclear power plants (Kennedy et al., 1980), automobile safety designs (Zhang and Liu, 2002), etc. As an example, consider the problem of testing car safety in case of a crash. To assess the deformation of the structure of a car, engineers use complex physical models, which are implemented under the form of resource-consuming computer programs.

Generally speaking, numerical simulations make it possible to assess the performance of a system for a particular configuration of design parameters and conditions of use of the system. One of the simplest approach to estimate the probability of failure of a system is to use a Monte Carlo approach which consists in doing many simulations for different design parameters and conditions of operation and approximating the probability of failure by the number of observed failures over the number of simulations. However, when simulations are time-consuming, the total number of simulations that can be used to assess the safety of a system can be very limited, which prevents us from using a crude Monte Carlo approach.

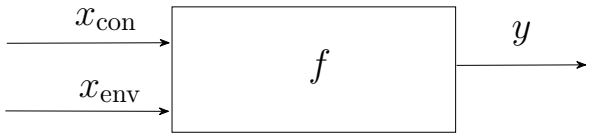


Figure 1.1: Black box view of a system.

1.1.2 Black box view of a system

In this work, we consider a system as a *black box* which has inputs and outputs. In other words, we are "blind" to the internals of a system. The only concern is the knowledge of how much the performance of the system changes when input variables vary.

In order to introduce some notations, consider Figure 1.1. The input vector of the system consists of two parts:

- the controllable factors x_{con}, also called *design variables* or *design factors*, which are the parameters that the designer of a system can adjust to optimize its performances.

- the *environmental factors* x_{env}, which cannot be controlled by the designer and correspond to the operating conditions of the system and manufacturing errors, for instance.

In what follows, we will focus on the environmental factors x_{env} and consider the controllable factors x_{con} as fixed by the designer. To simplify our notations, we denote by x the vector of environmental factors, which is assumed to take its values in a set $\mathbb{X} \subseteq \mathbb{R}^d$. The output vector y denotes the quantities of interest for the designer. For example, y can be a vector of numerical values quantifying the performances or the cost of a particular design. In this work, we will only investigate the cases where y is a scalar, and we will assume that the black box of Figure 1.1 is a deterministic function f which maps the input variables to a scalar performance or a cost.

1.2 Problem statement

1.2.1 Probability of failure

Let $f : \mathbb{X} \to \mathbb{R}$ be a function which corresponds to a performance or a cost of the system. To account for our uncertainty about the actual conditions under which a system will operate during its life, we choose a probability distribution $\mathsf{P}_{\mathbb{X}}$ over the factor space $\mathbb{X}$. Note that choosing such a distribution $\mathsf{P}_{\mathbb{X}}$ can be a difficult task in practice, but we consider that this issue is beyond the scope of our work. We will say that a failure event happens when f exceeds a prescribed value. Mathematically, the failure event is a set of points Γ written as

$$\Gamma = \{x \in \mathbb{X} : f(x) > u\}, \tag{1.1}$$

where $u \in \mathbb{R}$ is a *threshold* determined from the design requirements.

Given f, a probability distribution $\mathsf{P}_{\mathbb{X}}$, which is assumed to have a probability density function $p_{\mathbb{X}}$, and a threshold u, the probability of failure of the system can be written as

$$\begin{aligned} \alpha &= \mathsf{P}_{\mathbb{X}}(\Gamma) \\ &= \int_{\Gamma} p_{\mathbb{X}}(x)\mathrm{d}x \\ &= \int_{\mathbb{X}} \mathbf{1}_{\Gamma}(x) p_{\mathbb{X}}(x)\mathrm{d}x, \end{aligned} \tag{1.2}$$

where $\mathbf{1}_{\Gamma}$ is the indicator function of Γ which takes value one if $x \in \Gamma$ and zero if $x \notin \Gamma$.

Figure 1.2 illustrates the notion of probability of failure in a one dimensional case. The failure region is indicated in shadow areas. In higher dimension, the shape of Γ can be extremely complex.

The probability of failure in (1.2) is a multiple integral. When the input space is high-dimensional, obtaining a numerical approximation of α becomes non-trivial.

1.2.2 Designing computer experiments to estimate α

This thesis deals with the problem of obtaining good approximations of α. Any practical procedure to obtain a numerical approximation of α will be based on some evaluations of the function f, which determines Γ. We often talk about choosing a *design of experiments*, since each evaluation of f can be seen as an experiment. Moreover, since evaluating f corresponds to running a computer program, we speak

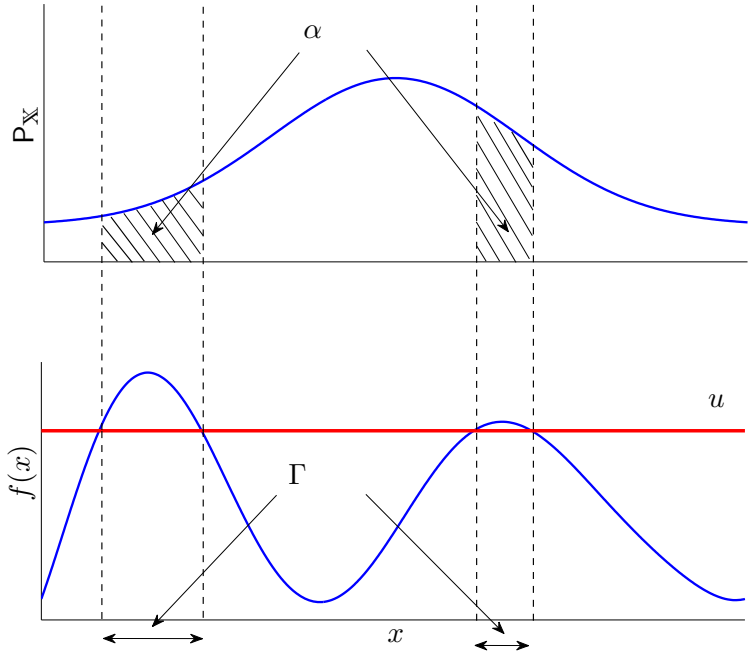


Figure 1.2: Illustration of a probability of failure in one dimension.

of *computer experiments* (see, e.g., Currin et al., 1991; Sacks et al., 1989; Santner et al., 2003a; Welch et al., 1992). When computer experiments are expensive (because they are resource-consuming), it becomes very important to choose a good design of experiments and to spend the budget of evaluations carefully in order to obtain a satisfactory approximation.

Moreover, we can say heuristically that a good approximation procedure should make evaluations in the neighborhood of Γ (otherwise, the location, the shape, and above all the volume of the set, cannot be assessed). Looking at Figure 1.3, a bad strategy would consist in choosing all evaluation points outside of Γ. If a Monte Carlo method were used in this case, the estimated probability of failure would be equal to zero.

As we see, the design of experiments plays a critical role to obtain satisfactory probability of failure estimations. There exists two major categories: non adaptive designs, and adaptive designs (also called sequential strategies). Sequential strategies try to build the design of experiment by adaptively choosing the experiments as a function of previous information. Typically, a new point is chosen to max-

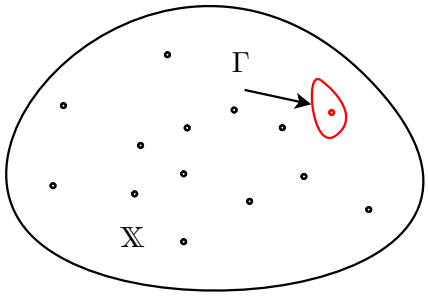


Figure 1.3: Illustration of Γ and input 𝕏.

imize/minimize some sampling criterion that represents the interest of making a new evaluation for a given estimation objective. For instance, Sacks et al. (1989) gives a sequential strategy to obtain a good approximation of a function from pointwise evaluations. Jones et al. (1998) proposes a sequential strategy called EGO for estimating the global optimum of a function.

The objective of this manuscript is twofold: 1) to provide a synthetic view about sequential strategies aiming at solving the problem of estimating a probability of failure; 2) to propose a new sampling strategy to estimate small probabilities of failure.

1.3 Outline of the dissertation

This thesis is organized as follows:

Chapter 2 gives an overview of reliability methods that are available in literature. The state of the art of reliability methods is first reviewed from the point view of probabilistic estimation. Among those approaches, variance reduction-based estimation methods have obviously gained a lot interest in the past decades, especially, importance sampling techniques have been very popular tools in reliability analysis. The emphasis of this thesis however is on methods using multilevel dividing technique, such as Subset Simulation and Sequential Monte Carlo algorithms. These techniques aim at estimating small probabilities of failure. However, the number of evaluations of the performance function remains still quite high for those methods. Global approximating methods, especially the estimation based on the Gaussian processes modeling, are then investigated for the sake of efficiency.

Chapter 3 presents a synthetic viewpoint on the sequential strategies based on a Gaussian process prior model and kriging predictor for the probability of fail-

ure estimation when the corresponding performance function under consideration is expensive-to-evaluate. This chapter starts with introducing an optimal strategy for estimating a probability of failure, which is developed from a Bayesian decision-theoretic framework aiming at minimizing a Bayesian risk. Realizing that this optimal strategy is unable to be solved by computer programming, it focuses on deriving several versions of sub-optimal strategies called *stepwise uncertainty reduction* (SUR) strategies. The idea is to substitute the exact Bayesian risk with an approximated risk which reflects the information gained from a new evaluation about the probability of failure estimation. The new strategies are then compared with other strategies in literature, such as the target IMSE criterion and criteria based on the marginal distribution. Those strategies are eventually evaluated on several numerical experiments.

In **Chapter 4**, we propose a new algorithm called Bayesian Subset Simulation, which is derived from the original Subset Simulation. The goal is to estimate a small probability of failure of a system smaller than 10^{-6}. Subset Simulation has been proved to be very efficient with a significant decreased number of evaluations to achieve the estimation of such a small probability of failure. However, due to the application of Monte Carlo sampling on the performance function, it still requires thousands of evaluations. A natural solution is to substitute the performance function with a surrogate model. In this chapter, an original idea is given to improve the efficiency of Subset Simulation algorithm, which combines the advantages of Subset Simulation and kriging prediction technique. Starting with reviewing the original Subset Simulation algorithm, it builds a Bayesian estimator and reformulates the original formula for calculating probabilities of failure. A detailed algorithm is given as well as some implementation issues. The effectiveness of this new algorithm is illustrated on three benchmark examples.

Finally, main conclusions and achievements, as well as perspectives for future research, are given in **Chapter 5**.

1.4 Publications

Chapter 3 has been published in [1]. Chapter 4 is an extension of [3].

[1] 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,* Volume 22, Number 3 (2012), 773-793, DOI: 10.1007/s11222-011-9241-4

[2] L. Li, J. Bect, and E. Vazquez. A numerical comparison of kriging-based sequential strategies for estimating a probability of failure (2011). In *11th International Conference on Applications of Statistics and Probability Civil Engineering (ICASP 11), Zurich, Switzerland, August 1-4.*

[3] Ling Li, Julien Bect, Emmanuel Vazquez (2012) Bayesian Subset Simulation : a kriging-based subset simulation algorithm for the estimation of small probabilities of failure In: Proceedings of PSAM 11 & ESREL 2012, 25-29 June 2012, Helsinki, Finland.

Chapter 2

Probabilities of failure estimation—literature review

Contents

2.1 Methods for estimating a probability of failure

Many people from different backgrounds have developed methods to estimate a probability of failure. We sort them into three major categories, represented in Figure 2.1:

- Geometric approximations of the failure region;
- Monte Carlo methods;
- Global approximations of f (surrogate models).

The purpose of this section is to provide an overview of existing methods in these categories. Moreover, because our main contribution is based on a Gaussian process (GP) approach, we focus on this category of methods in Section 2.4.4 (see Figure 2.2).

2.2 Estimation based on geometrical approximations

2.2.1 General principles

This category of methods geometrically approximate the contour of the failure region Γ. This technique has been used in many domains. For instance, it was widely used in structural safety (see, e.g., Madsen et al., 1986), and to assess the yield of manufacturing of electronic circuits (see, e.g., De Gyvez and Pradhan, 1998). The idea consists in choosing a set $\widehat{\Gamma}$ with a simple geometric shape to approximate the failure region Γ. An approximation of the probability of failure can then be written as

$$\widehat{\alpha}_{Geo} = \int_{\widehat{\Gamma}} p_{\mathbb{X}}(x)\mathrm{d}x = \int_{\mathbb{X}} \mathbf{1}_{x\in\widehat{\Gamma}} p_{\mathbb{X}}(x)\mathrm{d}x. \tag{2.1}$$

The estimator in (2.1) can then be computed using various techniques (for instance, using Monte Carlo method). Once $\widehat{\Gamma}$ is determined, there is no need to evaluate the performance function f. Therefore, the total number of evaluations of f is equal to the number of evaluations to determine $\widehat{\Gamma}$.

There are several examples of methods for region approximation in the literature. FORM/SORM (see, e.g., Breitung, 1984; Hohenbichler et al., 1987) used a hyper-plane or a paraboloid to approximate the limit state function in the standard normal space, Abdel-Malek and Hassan (1991) proposed to generate a sequence of

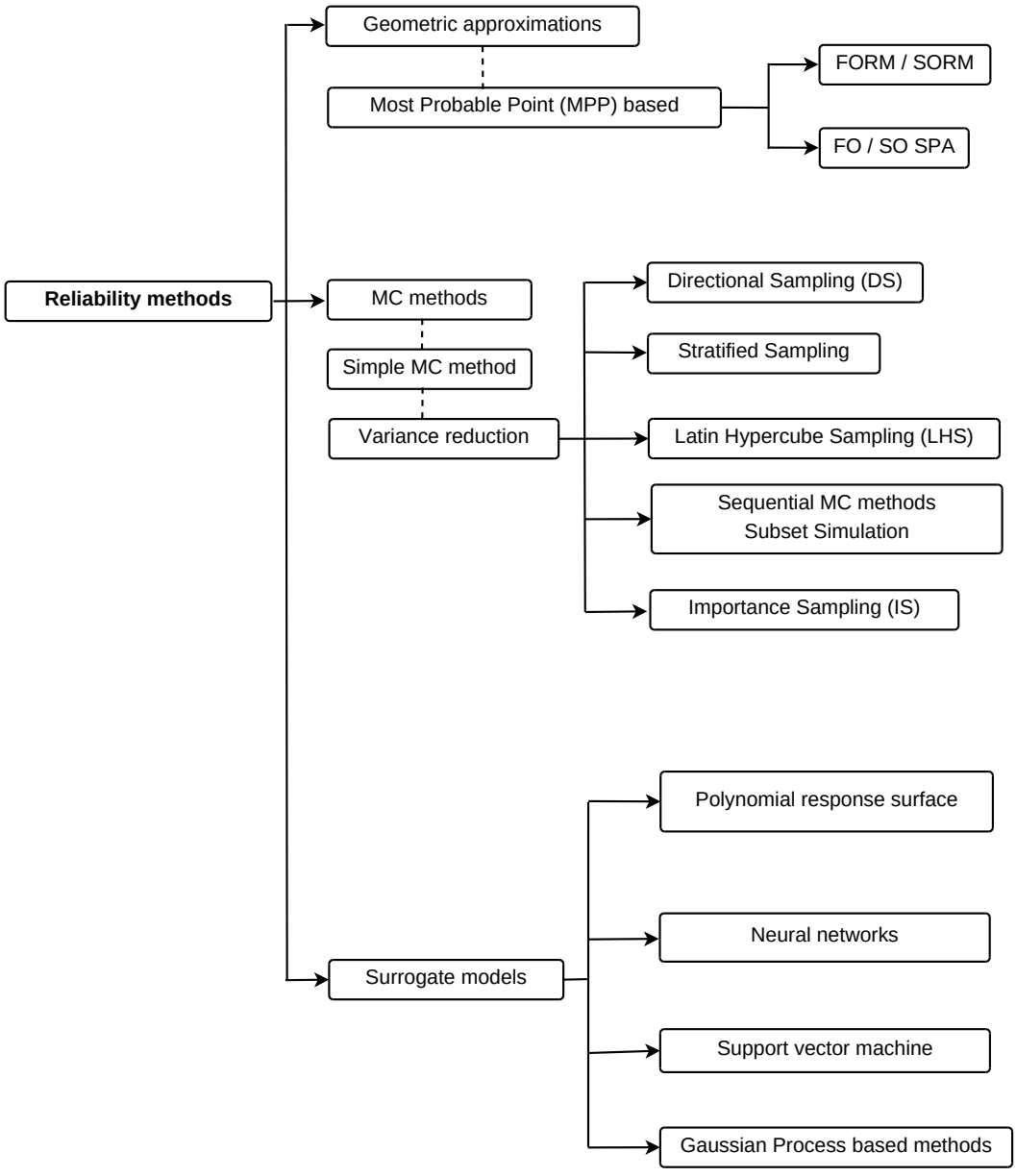


Figure 2.1: Summary of reliability methods in literature.

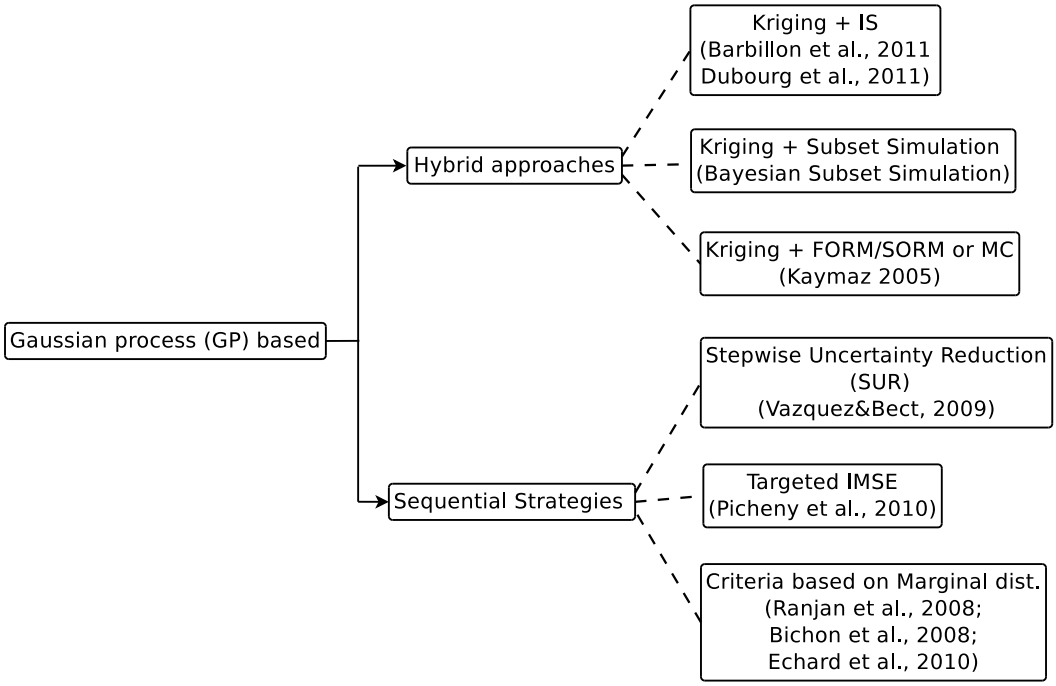


Figure 2.2: Summary of Gaussian process based methods.

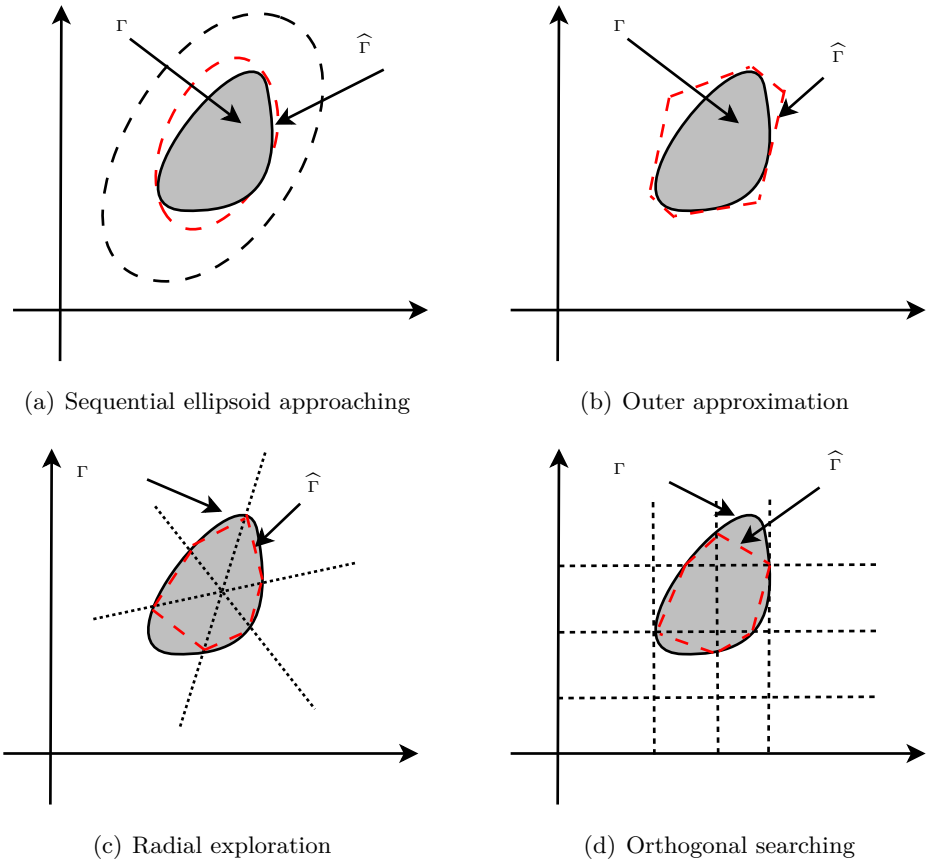
ellipsoids of decreasing volume to approach Γ, Zhang and Styblinski (1995) introduced a radial exploration approach, Ogrodzki and Styblinski (1980) used orthogonal search. Figure 2.3 shows examples of such approximation techniques. Notice that the approximations in Figure 2.3(a) and 2.3(b) will lead to an over-estimation of α, while in Figure 2.3(c) and 2.3(d) will result in an under-estimation of α.

Clearly, the success of this type of methods strongly depends on how close $\widehat{\Gamma}$ is to Γ, which also indicates that we need to have some prior information about the failure domain. If the failure region is very complex, or if Γ is not connected, the approximation will become non-trivial or even wrong. In addition, these techniques do not give information about the error of approximation.

In the next section, we will recall one of the most classical geometrical approximation methods in literature, namely the first-order and second-order reliability methods.

2.2.2 An example of geometrical approximation: FORM/SORM methods

FORM and SORM stand for *First Order Reliability Method* and *Second Order Reliability Method*.



(a) Sequential ellipsoid approaching

(b) Outer approximation

(c) Radial exploration

(d) Orthogonal searching

Figure 2.3: Examples of failure region approximation.

Definition 1. *Let f be a real-valued function defined on a factor space $\mathbb{X}$, and $u \in \mathbb{R}$ a threshold, such that a failure happens whenever $f(x) > u$. Then the function $G = u - f$ is called the limit state function for the function f w.r.t. the threshold u.*

The classical point of view is to say that the limit state surface divides the factor space into "safe" and "unsafe" (failure) regions. Consequently, the probability of failure can be written as

$$\alpha = \mathsf{P}_{\mathbb{X}}(G(x) < 0) = \int_{G(x)<0} p_{\mathbb{X}}(x)\mathrm{d}x, \tag{2.2}$$

where $p_{\mathbb{X}}$ is the probability density function as mentioned in (1.2).

FORM/SORM have four steps:

1. **Transform the input space**. Map the probability distribution of the in-

put variables $\mathsf{P}_{\mathbb{X}}$ into a standard normal distribution[1] by a transformation operator T, which is a diffeomorphism from $\mathbb{X}$ to the d-dimensional standard normal space $\mathbb{V}$ (see, e.g., Hasofer and Lind, 1973; Lebrun and Dutfoy, 2009a,b; Nataf, 1962; Rosenblatt, 1952, and the references therein).

Correspondingly, we have the transformed limit state function $\mathcal{G}$ which is defined by

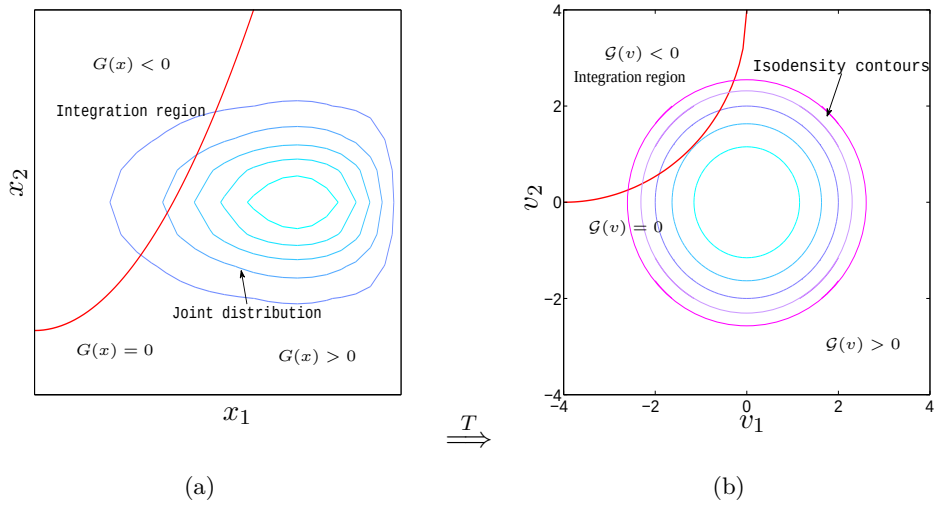
$$\mathcal{G}(v) = G(T^{-1}(v)), \quad \forall v \in \mathbb{V}. \tag{2.3}$$

Then, the probability of failure α can be rewritten as

$$\alpha = \mathsf{P}_{\mathbb{V}}(\mathcal{G}(v) < 0) = \int_{\mathbb{V}} \mathbf{1}_{\mathcal{G}(v)<0} p_{\mathbb{V}}(v) dv, \tag{2.4}$$

where $p_{\mathbb{V}}$ is the standard normal density function, which is invariant by rotation (this property indicates that $p_{\mathbb{V}}$ is only a function of $||v||$, where $||\cdot||$ denotes the euclidean distance).

Figure 2.4 shows the transformation from the original input distribution $\mathsf{P}_{\mathbb{X}}$ to the distribution $\mathsf{P}_{\mathbb{V}}$.



(a) (b)

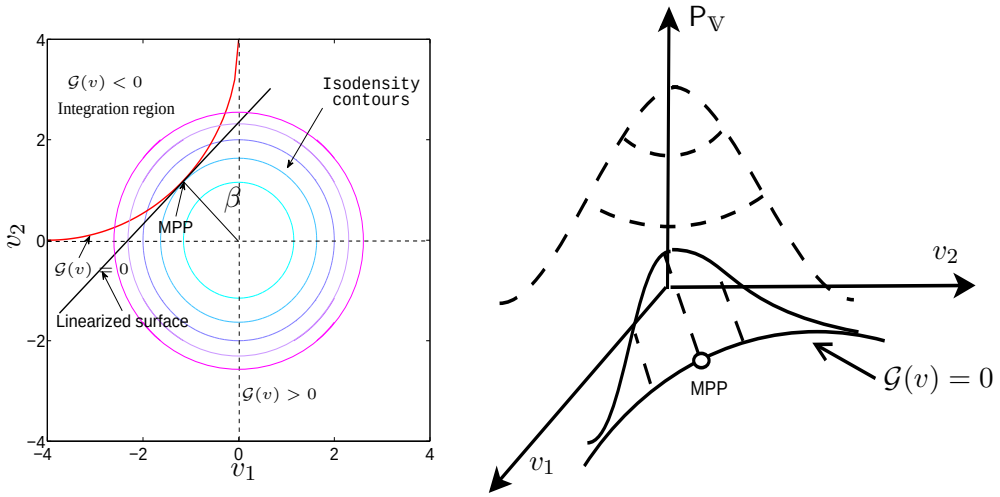
Figure 2.4: Transform the input factors from $\mathbb{X}$ into the standard normal space $\mathbb{V}$.

2. **Search for the Most Probable Point (MPP).** Let v^* be the *Most Probable Point* (MPP) on the limit state surface which is the nearest point to the origin in the standard space $\mathbb{V}$ such that $\mathcal{G}(v^*) = 0$. The distance β between the

[1] the input variables are assumed to be uncorrelated with $\mathcal{N}(0,1)$ marginal distribution

MPP and the origin of the standard space is called *reliability index* (see, e.g., Bjerager, 1990).

The tangent of the limit state surface at the MPP is orthogonal to the line going from the origin to the MPP. Notice also that the MPP has a maximum probability density on the curve $\mathcal{G}(v) = 0$. Figure 2.5(a) shows the MPP and reliability index and Figure 2.5(b) shows the maximum probability distribution density at the MPP.



(a) (b)

Figure 2.5: Left: The Most Probable Point and its corresponding reliability index; Right: the probability distribution density at MPP.

Consequently, finding the MPP v^* is a constrained optimization problem:

$$v^* = \arg\min_v ||v||, \tag{2.5}$$

subjected to the constraint

$$\mathcal{G}(v) = 0.$$

There exists several optimization algorithms to solve (2.5). Among the most used ones is a version of the gradient projection algorithm by Hasofer and Lind (1973); Rackwitz and Flessler (1978).

3. **Approximate the surface of the limit state function.** Next we need to approximate the limit state surface $\mathcal{G}(v) = 0$ with a tangent surface at the

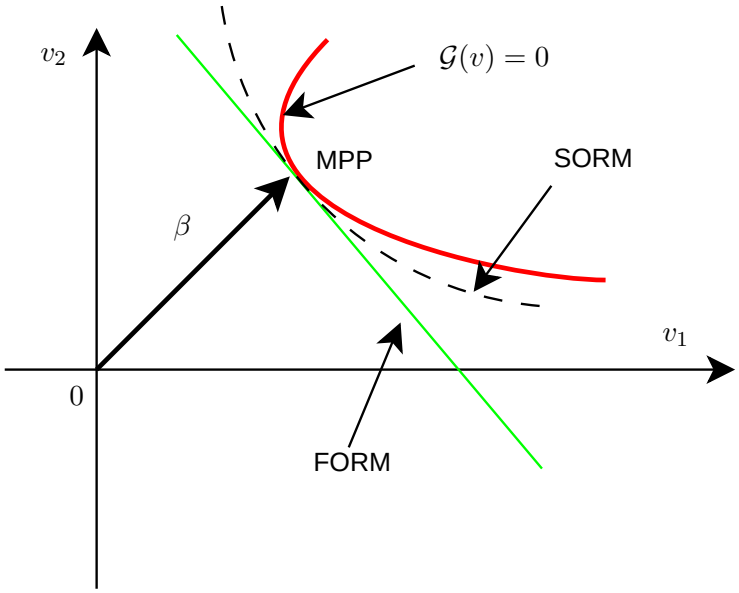


Figure 2.6: Approximation of the limit state surface to the FORM and SORM.

point v^*. In the case of FORM, the tangent surface is a hyper-plane, while in the case of SORM, a paraboloid is used (see Figure 2.6).

In general, SORM is more accurate than FORM because it has a higher order of approximation.

4. **Calculate the probability of failure.** The approximation of the probability of failure by FORM is

$$\widehat{\alpha}_{FORM} = \Phi(-\beta), \tag{2.6}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the normal distribution.

Concerning SORM, the probability of failure is approximated using functions of β and the curvature of the tangent surface at the MPP (see, e.g., Breitung, 1984; Hohenbichler et al., 1987; Tvedt, 1988).

Limit of the SORM/SORM approximations. The non-normal ($\mathbb{X}-$space) to normal transformation ($\mathbb{V}-$space) is a nonlinear transformation. In some cases, it tends to increase the non-linearity of the limit state function and may deteriorate the accuracy of approximations. Secondly, the accuracy of the estimator depends on the "quality" of the MPP. In fact, several problems can occur: a) the optimization algorithm to search for the MPP may fail to converge to the global constrained minimum of the distance function; b) FORM/SORM assumes that there is a unique

MPP on the limit state surface, which may not be the case; c) FORM/SORM assumes that the main contribution to α is concentrated in the vicinity of the MPP.

For example, the shape of the limit state surface may lead to under/over-estimation of the probability of failure. In addition, geometrical approximation methods cannot be used to quantify the approximation error. So there is no information for deciding whether the estimator is satisfactory or not. Figure 2.7 shows this phenomenon in the case of FORM. Due to our lack of information on the failure domain, it is impossible to tell whether the estimator obtained from the approximation is over (see Figure 2.7(a)) or under (see Figure 2.7(b))-estimated. Moreover, if there exists several points which have the minimum distance from the origin of the standard space, FORM/SORM can not process. Figure 2.8 shows the non-unicity problem in the simplest case of FORM.

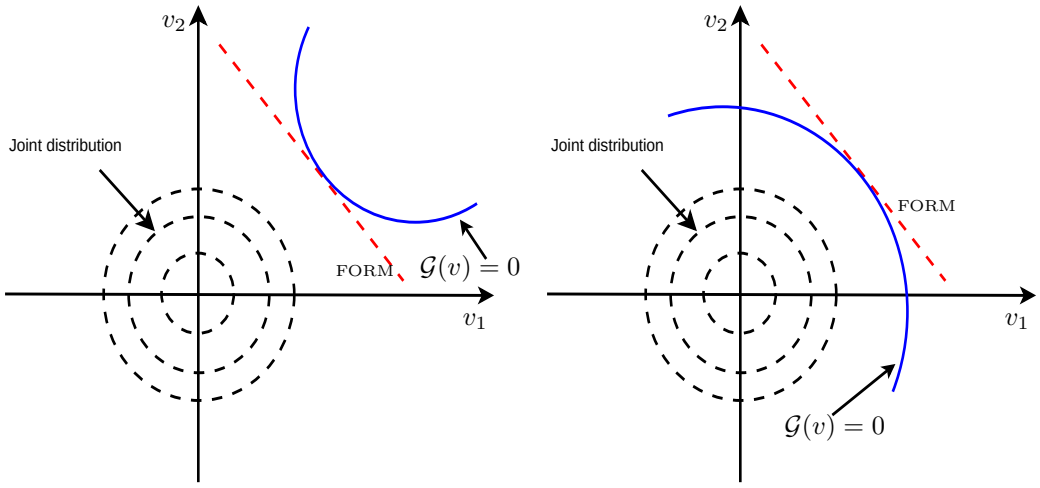


Figure 2.7: Examples of the lack of confidence on FORM approximation. (a) over-estimated; (b) under-estimated.

2.3 Monte Carlo (MC) methods

2.3.1 Simple Monte Carlo method

Monte Carlo technique is about invoking the *law of large numbers* to approximate an expectation or an integral. Here, we want to approximate the multiple integral in (1.2). The MC approach consists in drawing an independent and identically

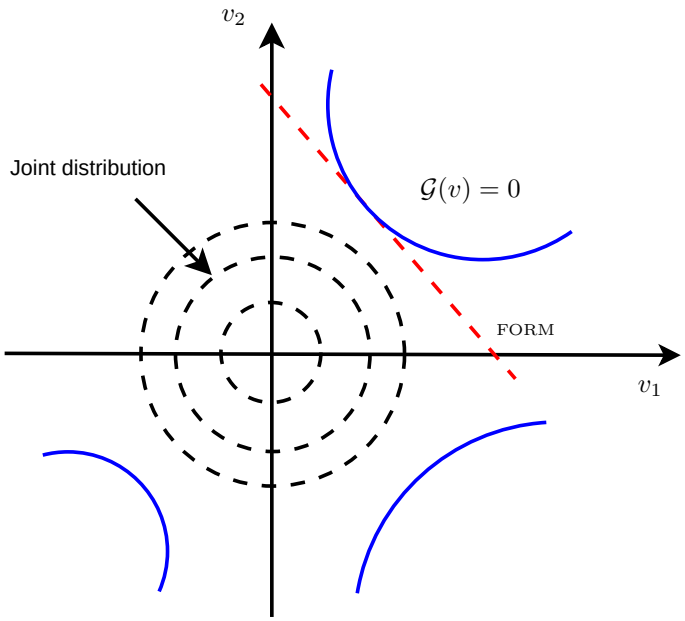


Figure 2.8: If the shape of the limit state surface has multiple failure modes, neither FORM nor SORM can provide a good approximation.

distributed (i.i.d.) sample $X_1, \ldots, X_m$ according to the distribution $\mathsf{P}_{\mathbb{X}}$. Then the empirical average

$$\widehat{\alpha}_{MC} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{X_i \in \Gamma} \tag{2.7}$$

is an estimator of the probability of failure α.

The *law of large numbers* tells us that $\widehat{\alpha}_{MC}$ converges almost surely to α. In other words, as long as the sample size m is large enough, it will be close to α. The convergence rate can be assessed by the *central limit theorem*. We have

$$\sqrt{m}(\widehat{\alpha}_{MC} - \alpha) \to \mathcal{N}(0, \sigma^2), \tag{2.8}$$

where $\sigma^2 = \mathrm{var}(\mathbf{1}_{X_i \in \Gamma})$, the variance of the indicator function. Therefore, the convergence rate of this Monte Carlo estimator is $O(1/\sqrt{m})$. Another thing to notice is that $\widehat{\alpha}_{MC}$ is an unbiased estimator.

The mean squared error (MSE) of this estimator is

$$\mathsf{E}\Big((\widehat{\alpha}_{MC} - \alpha)^2\Big) = \frac{\alpha(1-\alpha)}{m}. \tag{2.9}$$

Thus, if α is small, the standard deviation of $\widehat{\alpha}_{MC}$ is approximately $\sqrt{\alpha/m}$. To achieve a given standard deviation $\delta\alpha$, it requires approximately $1/(\delta^2\alpha)$ evaluations, which can be prohibitively high. For example, if $\alpha = 2 \times 10^{-3}$ and $\delta = 0.1$, we

need a sample size $m = 50000$. If one evaluation of f takes, say, one minute, then the entire estimation procedure will take about 35 days to complete. Thus, the simple Monte-Carlo method is not practical in the case of time-consuming simulations.

2.3.2 Some classical Monte Carlo methods from the literature of structural safety

A) **Directional Sampling** (DS). Directional sampling (see Ditlevsen and Bjerager, 1989; Ditlevsen et al., 1988; Melchers, 1994) uses lines to probe the failure domain which is quite similar to the line sampling technique (first proposed by Feldmann and Director (1993) under the name of surface integrals) as presented in Koutsourelakis (2004); Koutsourelakis et al. (2004). It is often used in literature to estimate the structural reliability for high dimensional problems (see, e.g., Schuëller et al., 2004).

The idea is to choose several appropriate directions of simulations to approximate a sphere integration. Suppose we map the input space into a standard space $\mathbb{V}$ by the same isoprobabilistic transformation as in Section 2.2.2. The probability of failure in this case can be rewritten as a sphere integration. Thus, if we choose l directions of simulations $z_1, \ldots, z_l$ (see Figure 2.9), the sphere integration can be approximated by:

$$\widehat{\alpha}_{DS} = \frac{1}{l} \sum_{i=1}^{l} p_i, \tag{2.10}$$

where p_i is the probability obtained from direction z_i, and can be computed by finding the roots for the limit state function. If several roots exist, p_i becomes a sum of all contributions.

Remark 1. *The directional sampling combined with directional simulation is very efficient for the limit stage surface close to spheres centered at the origin. It can be also used together with FORM/SORM method to approximate the probabilities of failure for an exact value. Martinez (2008) presented a directional simulation estimator in the original input space, in which case an importance sampling method is used to guide the choice of directions.*

B) **Stratified Sampling** (SS). Stratified sampling, sometimes also called *quota* sampling, broadly refers to the approach that partitions the sample space $\mathbb{X}$ into K disjoint sub-regions (or *strata*), $\mathbb{S}_1, \ldots, \mathbb{S}_K$, such that $\mathbb{S}_i \cap \mathbb{S}_j = \emptyset$ for

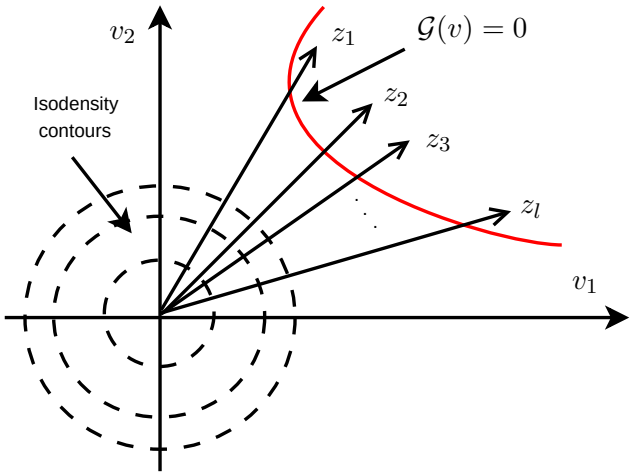


Figure 2.9: Illustration of directional simulation.

$i \neq j$, and $\mathsf{P}_{\mathbb{X}}(\cup_{i=1}^{K} \mathbb{S}_i) = 1$. Then the estimator (1.2) can be rewritten as:

$$\begin{aligned} \alpha &= \mathsf{P}_{\mathbb{X}}\{\cup_{i=1}^{K} \mathbb{S}_i \cap \Gamma\} \\ &= \sum_{i=1}^{K} \mathsf{P}_{\mathbb{X}}(\mathbb{S}_i)\mathsf{P}_{\mathbb{X}}(x \in \Gamma | x \in \mathbb{S}_i) \\ &= \sum_{i=1}^{K} p_i \mathsf{P}_{\mathbb{X}}(x \in \Gamma | x \in \mathbb{S}_i), \end{aligned} \tag{2.11}$$

where p_is are the probability that the sample lies in the strata $\mathbb{S}_i$, and

$$p_i = \int_{\mathbb{S}_i} p_{\mathbb{X}}(x) \mathrm{d}x. \tag{2.12}$$

In each strata, $i = 1, \ldots, K$, draw an i.i.d. sample $X_{ij}, j = 1, \ldots, m_i$ according to the conditional distribution $\mathsf{P}_{\mathbb{X}}(\cdot|\mathbb{S}_i)$. Apply a Monte Carlo simulation (notice that the total number of evaluations is $N = \sum_{i=1}^{K} m_i$), the stratified estimator can be written as

$$\widehat{\alpha}_{Stra} = \sum_{i=1}^{K} p_i \widehat{\alpha}_i, \tag{2.13}$$

where $\widehat{\alpha}_i = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathbf{1}_{X_{ij} \in \Gamma}$.

It is easy to prove that this estimator is unbiased, and the variance is given by Dunn and Shultis (2011, page 109).

Compared with crude Monte Carlo method, the stratified estimator $\widehat{\alpha}_{Stra}$ eliminates sampling variability across strata without affecting sampling variability within strata. To achieve this, stratified sampling tries to allocate more

points near the failure region. Figure 2.10 illustrates this sampling mechanic.

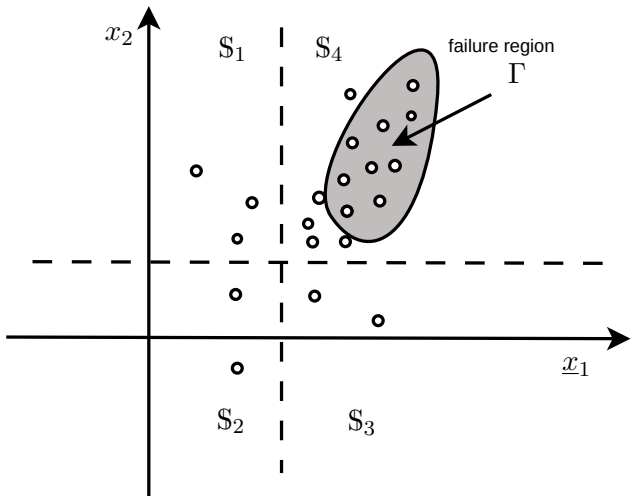


Figure 2.10: Illustration of the stratified sampling. Most points are allocated in the strata $\mathbb{S}_4$, which also covers the failure region.

C) **Latin Hyper-cube Sampling** (LHS). The method of Latin Hyper-cube sampling (LHS) was first introduced by McKay et al. (1979) and then further systematically studied by Stein (1987). It is an extension of stratified sampling but more efficient in high dimensional cases. For example, in a Stratified Sampling method, if we draw only one point from K strata in each of the d-dimension, the sample size will be K^d. LHS avoids this exponential growth of sample size with the dimension by treating all coordinates equally and drawing samples according to marginal distribution of a joint distribution for the input variables. Thus, if we want to generate a sample of size m, $X_1, \ldots, X_m$ in dimension d with a LHS method. For each coordinate $i = 1, \ldots, d$, we generate a equiprobable strata $\mathbb{S}_i^1, \ldots, \mathbb{S}_i^m$, each of which has a probability $1/m$. Then, we obtain a sample from the strata with probability $1/m$. Consequently, we have m realizations for each coordinate. Next, randomly permute the m realizations for all coordinates (precisely it makes $m!$ permutations equally likely). Finally we obtain m points which form a LHS. Notice that if we project the m points onto the ith coordinate, each strata $\mathbb{S}_i^j$ $(j = 1, \ldots, m)$ will have only one point. This property of LHS is illustrated in Figure 2.11.

Owen (1997) pointed out that for any square integrable function and any

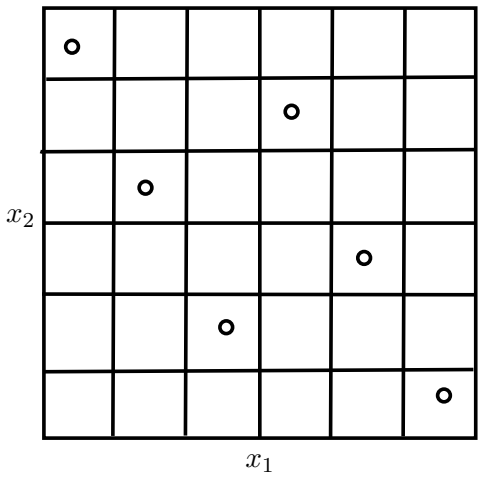


Figure 2.11: Latin hypercube sampling of size $m = 6$ in dimension $d = 2$.

sample size $m \geq 2$, the variance produced by a LHS scheme of size m is not larger than the variance produced by an i.i.d. Monte Carlo sample of size $m - 1$. This estimator is already studied by Olsson et al. (2003) in structural reliability problems. Stein (1987) explained that LHS is most effective with integrals that can be approximately separated into a sum of one-dimension functions.

Conclusions. The MC methods in this section are all based on variance reduction. The performance of directional sampling depends on the choice of the directions. A prior information or an expert may be needed as guidance in advance. In other words, this method usually works well in the cases when there is complementary information available, which can help us choosing a good simulation plan. Stratified sampling requires no knowledge of $\mathsf{P}_{\mathbb{X}}$, instead, the variances for each strata are required. It works well if good estimates of the variance in each strata are available and if the strata can be constructed to efficiently separate failure region from safe region. However in practice, this information is hardly known. So we need additional evaluations to get a rough idea of the failure region in advance. In addition, stratified sampling in high dimension is possible in principle but often unfeasible in practice. Partitioning each coordinate into K strata for input space in d dimension $\mathbb{X} \subset \mathbb{R}^d$ will produce K^d strata, which will require a sample size of at least K^d assuming each stratum is sampled once only. Thus for even moderately large d, the sampling becomes prohibitive unless K is small, in which case stratification provides little benefits. Latin Hyper-cube sampling tries to improve the

efficiency of stratified sampling. In our work however, those Monte Carlo methods are not applicable due to their requirement of large number of evaluations.

2.3.3 Importance Sampling (IS)

Let h be a *proposal* probability density function defined over $\mathbb{X}$ such that $supp(h) \supset supp(p_{\mathbb{X}})$ ($supp$ defines the support function). Then, α can be written as:

$$\alpha = \int_{\mathbb{X}} \mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}(x)\mathrm{d}x = \int_{\mathbb{X}} \mathbf{1}_{x\in\Gamma} \frac{p_{\mathbb{X}}(x)}{h(x)} \cdot h(x)\mathrm{d}x = \mathsf{E}_h\Big(\mathbf{1}_{x\in\Gamma} \frac{p_{\mathbb{X}}(x)}{h(x)}\Big), \tag{2.14}$$

where E_h denotes the expectation with respect to the density h. This gives an importance sampling Monte Carlo estimator:

$$\widehat{\alpha}_{IS} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{X_i\in\Gamma} \frac{p_{\mathbb{X}}(X_i)}{h(X_i)}, \tag{2.15}$$

where the X_is are i.i.d. from h. The estimator converges to (1.2) almost surely.

Optimal proposal density function. By minimizing the variance of $\widehat{\alpha}_{IS}$, it is easy to derive an optimal proposal density (see, e.g., Rubinstein and Kroese, 2008):

$$\begin{aligned} h^* &= \arg\min \mathrm{var}(\widehat{\alpha}_{IS}) \\ &= \frac{\mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}}{\int_{\mathbb{X}} \mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}\mathrm{d}x} \\ &= \frac{\mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}}{\alpha}. \end{aligned} \tag{2.16}$$

Notice that the optimal density h^* depends on Γ, which is exactly what we intend to identify in reliability analysis. Moreover, if there is only one sample X drawn from h, we will get $\widehat{\alpha}_{IS} = \alpha$ using the optimal proposal density in (2.16), and $h = h^*$. This is not true in real applications.

Conclusions. The success of an importance sampling technique to estimate a probability of failure depends on the selection of an appropriate importance density function: importance sampling requires a proposal distribution that approaches the optimal distribution as much as possible. This can be achieved by building h iteratively (see, e.g., the adaptive techniques which will be introduced in the next sections).

2.3.4 Adaptive Importance Sampling (AIS)

The idea of adaptive importance sampling (AIS) was first given in Kloek and Van Dijk (1978). An AIS algorithm consists in doing several intermediate Monte

Carlo simulations with proposal densities h_i, $i = 1, \ldots, S$, which are computed to resemble h^* as much as possible. During each simulation, properties of h^* are estimated based on past evaluations. Then the densities h_i are modified such that their properties match the estimated properties of h^*. In this iterative procedure, the intermediate probability densities will gradually approach the optimal density in its statistical properties.

A) **Parametric AIS.** In the parametric version of AIS, a finite vector of parameters is used to characterize the intermediate proposal densities h_i. Bucher (1988); Melchers (1990) proposed to use Gaussian density functions whose parametric form for the proposal density function is written as h_{ϑ_i}. Table 2.1 and Figure 2.12 describe the idea of the algorithm.

Table 2.1: Example of a parametric AIS method to estimate a probability of failure.

Initialization:

a) Take one value of x in the failure domain Γ as the initial mean μ_0, iteration $i = 0$.

b) Set the initial importance sampling density function $h_{\vartheta_0} = \mathcal{N}(\mu_0, 1)$.

For $i = 1, \ldots, S$:

1. Draw a Monte Carlo sample $X_1^i, \ldots, X_{N_i}^i$ according to h_{ϑ_i}, where N_i is the number of evaluations at iterative i.

2. Estimate the mean μ_i of h^*:

$$\mu_i = \sum_{j=1}^{N_i} w_j^i X_j^i, \tag{2.17}$$

and covariance k_i:

$$k_i = \sum_{j=1}^{N_i} w_j^i (X_j^i - \mu_i)(X_j^i - \mu_i)^\intercal, \tag{2.18}$$

where $w_j^i \propto \frac{p_{\mathbb{X}}(X_j^i)}{h_i(X_j^i)} \mathbf{1}_{X_j^i \in \Gamma}$, and $\sum_{j=1}^{N_i} w_j^i = 1$, update $h_{\vartheta_i} = \mathcal{N}(\mu_i, k_i)$.

3. Repeat [1] and [2] until it reaches a stop criterion.

4. Estimate the probability of failure using (2.15), with $h = h_{\vartheta_S}$.

B) **Non-Parametric AIS.** In the non-parametric case, instead of estimating

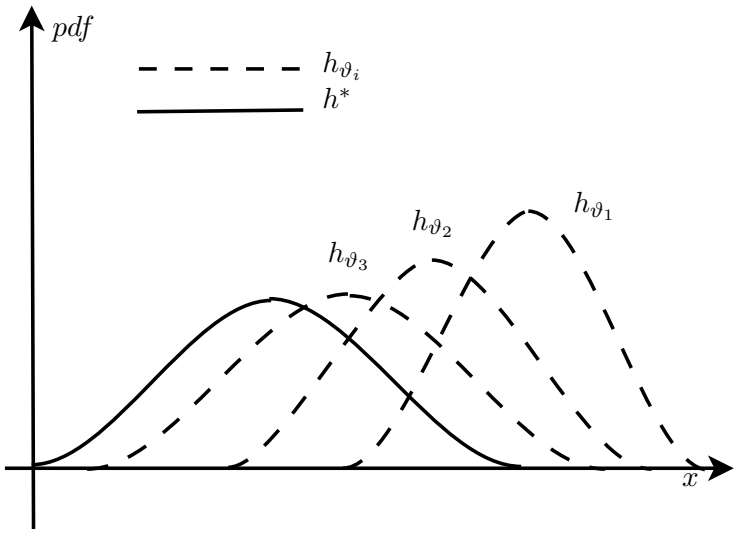


Figure 2.12: Adaptive Importance Sampling. The solid line represents the optimal density function $h^*(x)$, the dashed lines are the estimated density function with the number indicates iteration order.

some properties of the optimal density function h^*, the density function itself is estimated. Ang et al. (1990) proposed to use a simple rectangular kernel density, Au and Beck (1999) modified the it with a reduced Gaussian probability density kernel, Karamchandani et al. (1989) suggested to identify several modes representing the importance of the failure region by a preliminary simulation. The proposal density function is then constructed by replicating the target density at every mode weighted by the importance. Algorithms of this kind can be found in Melchers (1990).

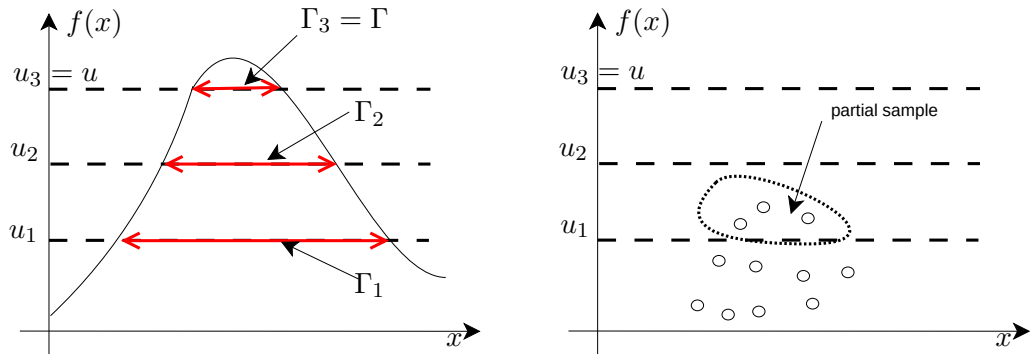
Another form of non-parametric AIS technique, the so called two-stage surrogate based importance sampling strategy, will be introduced in Section 4.2.1.

Conclusions. The category of AIS methods tries to select a "good" importance density function which iteratively approaches the optimal density in the sense of the its statistical properties or modes. The method is more efficient compared to original importance sampling. However, it is nontrivial to design a good proposal distribution in high dimensional problems. In addition, in the parametric cases, it becomes difficult to build an appropriate proposal distribution when the failure region is highly concave (Melchers, 1991).

2.3.5 Subset Simulation

When the probability of failure α is very small, Monte Carlo simulations as well as most of the variance reduction methods mentioned before do not work well any more. A natural idea is to decompose the rare event into several "not so rare" nested events, and calculate the probability of failure by multiplying the corresponding sequence of conditional probabilities.

To do this, we partition the output space into stages (the same as the *levels* used in Glasserman et al., 1999) using a sequence of intermediate thresholds $u_0 = -\infty < u_1 < u_2 < \ldots < u_T = u$. Correspondingly, we have a sequence of nested excursion sets $\Gamma_0 := \mathbb{X} \supseteq \Gamma_1 \supseteq \cdots \supseteq \Gamma_T := \Gamma$ (see Figure 2.13(a)). Thanks to this *multilevel dividing* technique, we are able to deal with the problem in each sub space (or *subset* in Au and Beck, 2001). From stage t to $t+1$, the samples that pass the intermediate threshold u_{t+1}, have more importance than the others. We reinforce and populate this *partial sample* (see Figure 2.13(b)) which is more promising to reach the failure region, and discard the others by different techniques. Methods in this section are all based on this multilevel dividing technique.



(a) Three levels dividing

(b) Partial sample

Figure 2.13: Multilevel dividing technique.

Using conditioning, the probability of failure α can be rewritten as

$$\alpha = \mathsf{P}_{\mathbb{X}}(\Gamma_T) = \mathsf{P}_{\mathbb{X}}(\Gamma_1) \prod_{t=1}^{T-1} \mathsf{P}_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t) = \prod_{t=0}^{T-1} p_t, \tag{2.19}$$

where $\mathsf{P}_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t) = p_t$.

Suppose we have a sampler which could generate an independent sample $\mathrm{X}_t = \{X_t^1, \ldots, X_t^{N_t}\}$ according to the distribution $\mathsf{P}_{\mathbb{X}}(\cdot|\Gamma_t)$ with sample size N_t and has a probability p_t to fall in Γ_{t+1}. At each iteration t, we keep the partial sample which

locate in the next excursion set Γ_{t+1}, generate an i.i.d. sample with a sampler. We can continue this process until $t = T$, and obtain a multilevel estimator:

$$\widehat{\alpha}_{MulSplit} = \prod_{t=1}^{T} \frac{\sum_{i=1}^{N_t} \mathbf{1}_{X_t^i \in \Gamma_t}}{N_t} \tag{2.20}$$

and the total number of evaluations $N = \sum_{t=1}^{T} N_t$. If the intermediate thresholds u_ts are set in advance, $\widehat{\alpha}_{MulSplit}$ is unbiased.

To obtain the estimator in (2.20), we need to generate independent samples X_t. There exists several methods based on multilevel dividing technique. This type of methods can be seen as a particular case of sequential Monte Carlo (SMC) method. To our knowledge, Au and Beck (2001, 2003) were the first to use multilevel dividing technique for reliability analysis. In the standard rare event literature, a particle system is adopted and a framework of Markov Chain Monte Carlo (MCMC) step is developed to generate samples, which aims at restricting on smaller and smaller excursion sets, with the smallest set being the failure region Γ (see, e.g., Cérou et al., 2011, and the references therein). More recently, Botev and Kroese (2011) reviewed different versions of sample population procedures for static rare event estimation.

Fixed-level algorithm. Suppose the sequence of increasing intermediate thresholds $u_0 = -\infty < u_1 < u_2 < \ldots < u_T = u$ is known in advance. At each iteration t, the algorithm chooses the partial sample which is located in the excursion set Γ_{t+1}, generating an appropriate conditional sample under the framework of MCMC. In some applications, a modified Metropolis-Hastings approach is used (see, e.g., Au and Beck, 2001). p_t at each stage can be approximated by a simple Monte Carlo estimator $\widehat{p}_t$.

The Metropolis algorithm prescribes the transition rule for a Markov chain. It is a very powerful technique to simulate samples as the states of a Markov chain such that the limiting/stationary distribution of this chain is the target distribution.

The procedure of the Metropolis algorithm is as follows: starting from an initial sample, proceeding iteratively, 1) a random *perturbation* of the current state is made according to a probability transition function $h(x'; x)$; 2) a ratio r reflecting the "gain" resulting from the perturbation is calculated; 3) a random number is generated independently; 4) accept the current state with probability $\min(1, r)$ and reject it with the remaining probability $1 - \min(1, r)$.

Given the current state t, an example of Metropolis-Hasting algorithm can be implemented by the iterations in Table 2.2.

Table 2.2: Metropolis-Hastings Algorithm.

Initialize: Choose the proposal probability distribution function $h(\tilde{x}_t|x_t)$.

Iterative: For $t = 1, 2, \ldots,$:

- Generate a candidate state $\tilde{x}_t$ according to $h(\tilde{x}_t|x_t)$.
- Draw a random number $\mathscr{C} \sim \mathcal{U}(0, 1)$, and update

$$x_{t+1} = \begin{cases} \tilde{x}_t & \text{if } \mathscr{C} \leq r(\tilde{x}_t, x_t) \\ x_t & \text{otherwise.} \end{cases}$$

Table 2.3: Fix-level SMC algorithm.

Initialize:

1. Set the intermediate threshold u_t $(t = 0, \ldots, T)$.

2. Set the Monte Carlo sample size m, and draw an i.i.d. sample $\mathrm{X}_0 \sim \mathsf{P}_{\mathbb{X}}$.

3. Choose the proposal probability distribution function $h(\tilde{x}_t|x_t)$.

Iterate: For $t = 1, 2, \ldots, T$:

1. Compute $\widehat{p}_t = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{X_{t-1}^i \in \Gamma_t}$.

2. Generate a conditional sample X_t:

2.1 Obtain a partial sample which locates in Γ_t;

2.2 Apply a Metropolis-Hasting algorithm in Table 2.2;

2.3 update sample X_t.

Estimate: The probability of failure is calculated by

$$\widehat{\alpha}_{SMC_1} = \prod_{t=0}^{T-1} \widehat{p}_t. \tag{2.21}$$

An example of fix-level SMC algorithm is presented in Table 2.3. For fixed-level algorithm, the estimator $\widehat{\alpha}_{SMC_1}$ is unbiased even though the samples are dependent. Del Moral (2004) studied the asymptotic feature of the conditional sample model in a general framework as the number of particles goes to infinity. And the relative variance given by Cérou et al. (2011) is lower bounded as

$$\sigma^2(\widehat{\alpha}_{SMC_1}) \geq \sum_{t=1}^{T} \frac{1-p_t}{p_t}, \tag{2.22}$$

where the equality holds only if the Markov chain goes to infinity. Then for a fixed probability of failure α and a number T of levels, the problem becomes minimizing (2.22) subject to the constrained $\prod_{t=1}^{T} p_t = \alpha$. Solving this optimization problem, we get $p_t = p_0$ for $t = 0, \ldots, T-1$, and $p_0 = \alpha^{1/T}$, correspondingly, we should set the intermediate thresholds which evenly divide the input space $\mathbb{X}$ in terms of conditional probabilities. In very few cases, the Markov chain may stop prematurely because the two consecutive intermediate threshold are set too far apart, which make the successful partial sample an empty set. The adaptive-level algorithm which we will introduce next will fix this problem.

Adaptive-level algorithm. This method takes advantage of the knowledge from performance function. At each level t, the algorithm sorts the values of $f(X_t^i)$ and $i = 1, \ldots, m$, and set the next level intermediate threshold u_{t+1} to the $(1-p_0)$ empirical quantile of $f(X_t^i)$. The partial sample therefore is composed by the $p_0 m$ particles with the highest $f(X_t^i)$. The iteration stops when $u_t \geq u$, correspondingly, we have T levels. The intermediate probabilities $\widehat{p}_t = p_0$ for $t = 0, \ldots, T-1$, and the probability of failure is estimated by

$$\widehat{\alpha}_{SMC_2} = \widehat{p}_{T-1} p_0^{T-1}, \tag{2.23}$$

where $\widehat{p}_{T-1}$ is a Monte Carlo estimator at level $T-1$.

There is no theoretical study for the optimal p_0 so far. Clearly, the choice of p_0 is a trade-off between the Monte Carlo sample size at each stage and the total number of stages, while the total number of evaluations is $N = mT$.

The algorithm is presented in Table 2.4. Figure 2.14 illustrates the adaptive procedure for Subset Simulation method in Au and Beck (2001). We will also introduce a similar adaptive method in Section 4.3.3.

The estimator given by the adaptive-levels algorithm is biased. However, Cérou et al. (2011) proved that the bias is positive and of order $1/m$, which is negligible compared to its standard deviation. In practice, it is always safer to have a conservative estimator in reliability assessment.

Table 2.4: Adaptive-level SMC algorithm.

Initialize: At $t = 0$,

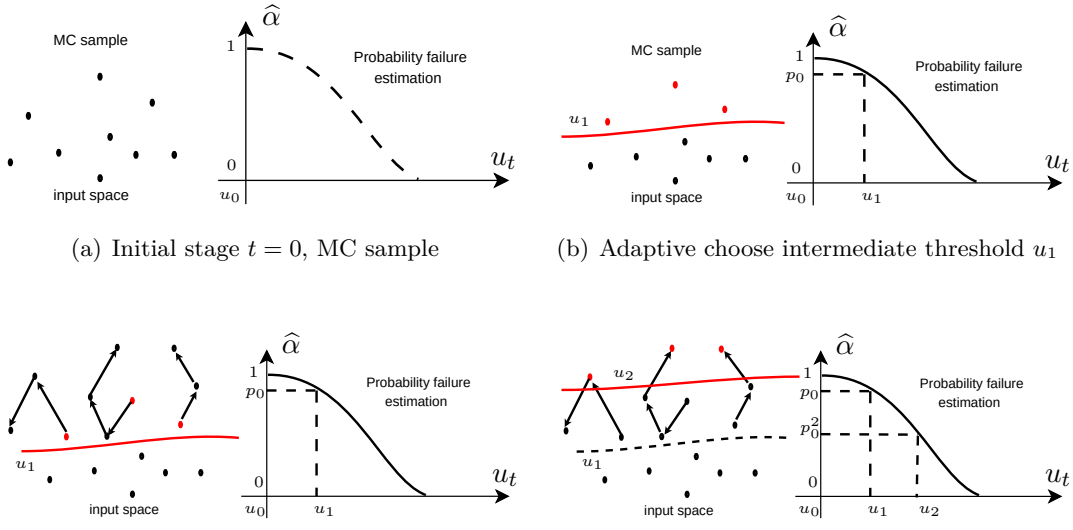
1. Set the Monte Carlo sample size m, and draw an i.i.d. sample $\mathrm{X}_0 \sim \mathsf{P}_{\mathbb{X}}$, where $\mathrm{X}_0 = \{X_0^i\}$ $(i = 1, \ldots, m)$.
2. Choose the value for p_0, set $u_0 = -\infty$.

Iterate: While $u_t < u$ $(t \leftarrow t + 1)$:

1. Set u_t to the $(1 - p_0)$th quantile of $f(\mathrm{X}_{t-1})$.
2. Start from the partial sample, generate a new sample $\mathrm{X}_t = \{X_t^i\}$ $(i = 1, \ldots, m)$ as in Table 2.3.

Estimate: Set $u_t = u$, the probability of failure is calculated by

$$\widehat{\alpha}_{SMC_2} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{X_{T-1}^i \in \Gamma} \; p_0^{T-1}. \tag{2.24}$$



(a) Initial stage $t = 0$, MC sample

(b) Adaptive choose intermediate threshold u_1

(c) Markov Chain Monte Carlo simulation

(d) Adaptive choose intermediate threshold u_2

Figure 2.14: Illustration of Subset Simulation with adaptive threshold.

Conclusions. Methods based on multilevel dividing technique are very efficient in the estimation of small probabilities of failure. Multilevel diving technique decomposes a complex problem into a sequence of easier problems. MCMC is a very powerful tool to generate samples according to an arbitrary probability distribution. The number of evaluations needed by these methods is dramatically decreased compared to MC methods. On the other hand, due to the use of MC simulation at each stage, the number of evaluations is still quite high. In the case of expensive-to-evaluate functions, Subset Simulation methods are not applicable. In Chapter 4 we will introduce our Bayesian Subset Simulation which modifies Subset Simulation methods in order to deal with expensive-to-evaluate functions.

2.3.6 Summary of the Monte Carlo methods

In this section, we introduced several Monte Carlo methods aiming at decreasing the variance of the estimator. Not like the methods based on geometric approximation, several methods, such as importance sampling, can provide some information on the confidence of the estimator.

It is well known that a sampling strategy is more efficient by integrating the information of failure region. Stratified sampling uses strata to separate failure and safe regions. More points are allocated in the strata which contains failure region. This method works well in low dimensional case and needs a good estimation of the variance in advance. LHS avoids the dimension problem by equally drawing samples from marginal distribution. However, both methods need a large number of evaluations.

Another category of methods, importance sampling and the related methods in Section 2.3.3 and 2.3.4, try to allocate more points in the regions of more "importance" by sampling according to a proposal distribution close to the input distribution. The methods are quite efficient if the proposal distribution is appropriately chosen. However, when the failure region is highly concave, it is very difficult to build a good proposal distribution for a parametric adaptive importance sampling method. The importance sampling related methods are quite successful in a lot applications.

When the probability of failure is small, methods based on multilevel dividing technique are able to break the rare event into a series of conditional events. The output space is then divided by intermediate thresholds. The probability of failure is then calculated by multiplying several conditional probabilities estimated from a

Monte Carlo simulation. MCMC technique is used to generate new samples at each stage. The essence of importance sampling is also used in a sequential Monte Carlo approach. However, due to the direct use of a Monte Carlo estimation at each stage, the number of evaluations remains still quite high. In Chapter 4 we will introduce a new algorithm which takes advantage of this dividing idea and improves Subset Simulation. Kriging prediction, which will be presented in the next section, can be used to provide a global approximation of the performance function.

2.4 Estimation based on an approximation of f

To overcome the cost of computer experiments, a natural idea is to substitute the expensive-to-evaluate computer programs with cheap surrogate models which are built by fitting a set of observations of the system responses. By definition, a surrogate model $\widehat{f}$ is an approximation of f. Once a surrogate model has been constructed, the probability of failure can be estimated from $\widehat{f}$, which is cheaper to evaluate, instead of f. The method of surrogate modeling is in fact very similar to that of geometrical approximation methods discussed in Section 2.2.

The plug-in estimator obtained from surrogate modeling can be written as

$$\begin{aligned} \widehat{\alpha} &= \mathsf{P}_{\mathbb{X}}\{\widehat{f} > u\} \\ &= \mathsf{P}_{\mathbb{X}}\{\widehat{\Gamma}\} \\ &= \int_{\widehat{\Gamma}} p_{\mathbb{X}}(x)\mathrm{d}x, \end{aligned} \tag{2.25}$$

where $\widehat{\Gamma}$ is an approximation of the excursion set on the surrogate model which is written as

$$\widehat{\Gamma} = \{x \in \mathbb{X} : \widehat{f}(x) > u\}. \tag{2.26}$$

2.4.1 Polynomial response surface

The response surface method uses a polynomial response surface to approximate the performance function. The coefficients of the response surface are generally obtained by least square regression. The probability of failure can then be calculated by applying FORM/SORM.

Response surface methods are very popular in the literature of reliability analysis. Bucher and Bourgund (1990); El-Tawil et al. (1991); Faravelli (1989); Kim and Na (1997) set up the early theoretical basis for response surface method; further developments can be found in Kaymaz and McMahon (2005); Zheng and Das (2000).

Gayton et al. (2003) proposed a method called CQ2RS (Complete Quadratic Response Surface with Re-Sampling) using a complete quadratic response surface to take into account variable interactions. The surrogate model in the standard space is then written as in Gayton et al. (2003):

$$\widehat{\mathcal{G}}(v) = a_0 + \sum_{i=1}^{d} a_i v_i + \sum_{i=1}^{d} \sum_{j \geq i}^{d} a_{i,j} v_i v_j, \tag{2.27}$$

where $v = \{v_1, \ldots, v_d\}$; $\widehat{\mathcal{G}}$ is the quadratic response surface for the limit state in standard space; a_i is the coefficient for direction v_i, and $a_{i,j}$ is the coefficient of variable interaction between v_i and v_j. The number of coefficients is then $L = \frac{(d+1)(d+2)}{2}$. This type of response surface was shown to be efficient. Gayton et al. (2003) used a statistical formulation to obtain an empirical distribution of the MPP v^* and applied a resampling step to estimate the mean value for v^*.

Blatman (2009) proposed polynomial chaos representations using spectral expansions and approximated the unknown response function in a finite dimensional basis. Their methods were applied in reliability analysis and sensitivity analysis.

Conclusions. Methods based on response surface approximation greatly decrease the number of evaluations. The surrogate model is a polynomial, and most of the time, a quadratic response surface. One problem of this kind of methods is, when the model of the system is complex, a simple quadratic function cannot fit the model very well. An inappropriate surrogate model will introduce bias into the estimator. In addition, finding a good MPP v^* becomes nontrivial when the response surface is complex.

2.4.2 Neural networks

Neural network models have found their way in the domain of reliability analysis. The idea of this method is to build a neural network as a surrogate model for f. The probability of failure can be approximated by a MC simulation on the cheap surrogate model $\widehat{f}$. This technique can be found in Hurtado and Alvarez (2000); Papadrakakis and Lagaros (2002). Since neural network is not our focus in this thesis, we will not present details concerning the training process. A general framework of this kind of methods is as follows:

1. Select several training points in $\mathbb{X}$, get their corresponding outputs.

2. Train the neural network (e.g., back-propagation networks) with the input/output pairs, obtain the parameters for the network.

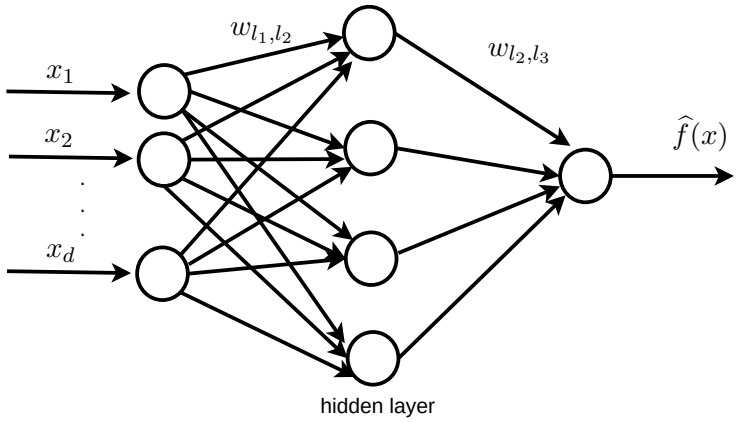


Figure 2.15: A widely used *three layers perception* neural network structure. The input $x = \{x_1, \ldots, x_d\}$; w_{l_1,l_2}s are the weights from the first layer to the second layer; w_{l_2,l_3}s are the weights from the second layer to the third layer.

3. Use the trained network as the surrogate model (see Figure 2.15), apply a simple Monte Carlo simulation. The probability of failure can be approximated by a plug-in estimator as in (2.25).

Conclusions. The neural network method is similar to the response surface method in the sense that both are aiming at using a cheap model to substitute the expensive-to-evaluate performance function in a parametric form. The difference is that the response surface method uses a least square fitting of a polynomial function (mostly in quadratic form), while neural network intends to estimate a nonlinear mapping from the input to the output of a system. It is natural that neural network provides more flexibility to the surrogate model than response surface method. However, building a neural network for a complex system, e.g., more than three hidden layers involved, requires heavy computation efforts for the training algorithms.

2.4.3 Support Vector Machine (SVM)

Support vector machine (SVM) is another very popular statistical learning technique. It is applicable both for classification and regression problems. Hurtado (2004a, 2007) brought it to the structural reliability community; the technique was then studied by Basudhar and Missoum (2008); Deheeger and Lemaire (2006); Li et al. (2006); Piera-Martinez (2008); Piera-Martinez et al. (2007).

The basic idea of support vector machine in structural reliability analysis is to

take advantage of its classification property. The method consists of two steps: 1) set a binary classifier which separates failure events from safe events in $\mathbb{X}$ using the *maximum margin principle*; 2) perform a Monte Carlo simulation and calculate the probability of failure using the classifier instead of f to decide whether a given point in $\mathbb{X}$ is in Γ or not.

In more details, consider a set of labeled training points $\{(x^1, c^1), \ldots, (x^p, c^p)\}$ where the class label c^i takes value $+1$ if $x^i \in \Gamma$ and -1 otherwise. Assume there is a transformation which operates a projection of the original input space onto a *feature space*, usually with a greater size. The classification problem becomes equivalent to find a linearly separable classifier for the failure and safe events in the transformed feature space (see Figure 2.16, also refer to the *kernel trick* in Deheeger and Lemaire, 2006). Then the approach tries to maximize the distance between the hyper-plane and the two classes. Figure 2.17 illustrates the approach.

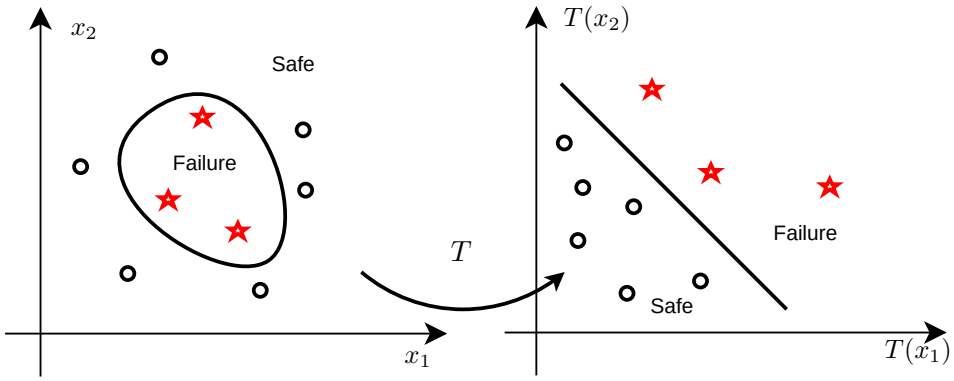


Figure 2.16: Projection onto the feature space.

This linear classifier (or the hyper-plane) in the feature space can be written in the following form:

$$\partial\widehat{\Gamma} = \{x \in \mathbb{X}; H(x) = \langle T(w), T(x)\rangle_{\mathbb{H}} + u = K(w, x) + u = 0\}, \tag{2.28}$$

where $\mathbb{H}$ is the feature space, $T(\cdot)$ is a transformation such that the inner product in the space $\mathbb{H}$ can be computed by a kernel function $K(\cdot, \cdot)$ in the original input space. There are several usual kernel functions used in SVM theory, e.g., radial basis kernels, polynomial kernels, etc. The parameter w is a weight vector that defines the projection of input vector on the hyper-plane, and u is the threshold.

The probability of failure can be estimated by performing a Monte Carlo sampling method and the classifier from the SVM method.

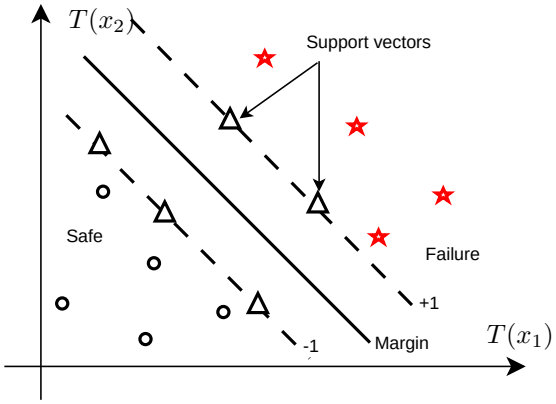


Figure 2.17: SVM principle: the optimal hyperplane maximizes the margin between two classes in the feature space.

Remark 2. *Support vector machines can also be used in regression problems to build a surrogate model for the performance function f.*

Conclusions. The probability of failure obtained from SVM method is potentially more accurate than FORM/SORM methods because the limit state function is globally approximated instead of locally approximated near a most probable point. The global approximation of the limit state function in the feature space is more satisfactory for failure probability estimation. SVM is a useful tool for reliability analysis.

2.4.4 Sequential strategies based on Gaussian processes modeling

This group of methods uses *kriging* technique, which considers a deterministic computer code f as a realization of a random process ξ (in this thesis the random process will always be a Gaussian process). This technique comes from the domain of geostatistics (Krige, 1951; Matheron, 1963) and has been used in the domain of computer experiments by Sacks et al. (1989). Kriging can be interpreted as a Bayesian approximation method (Oakley and O'Hagan, 2002). Based on an assumption of a *prior probability distribution* on f under the form of a Gaussian random process ξ, as well as the knowledge of previous evaluations $\xi(x_1), \ldots, \xi(x_n)$, the kriging predictor $\widehat{\xi}_n(x)$ is

$$\widehat{\xi}_n(x) = \mathsf{E}\Big(\xi(x) \mid \xi(x_1), \ldots, \xi(x_n)\Big), \tag{2.29}$$

and the kriging variance is

$$\sigma_n(x)^2 = \mathsf{E}\Big((\xi(x) - \widehat{\xi}_n(x))^2 \mid \xi(x_1), \ldots, \xi(x_n)\Big). \tag{2.30}$$

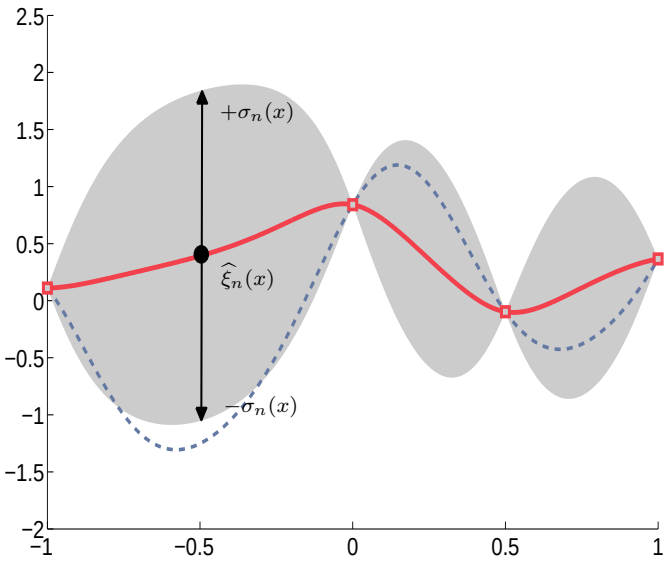


Figure 2.18: A one-dimensional illustration of kriging. Red squares indicate the location of the observed data (from the function represented in blue dashed line). The performance function is in dashed line. The kriging prediction is in solid red line. The gray area indicates the 95% confidence region.

For more information about Gaussian processes and kriging prediction, see, e.g., Chilès and Delfiner (1999).

Figure 2.18 illustrates a simple one-dimensional example of kriging prediction with four observations, where the gray region indicates confidence intervals. The kriging predictor is also called the best linear unbiased predictor (BLUP).

Kriging predictor *interpolates* the observations in the input range such as illustrated in Figure 2.19.

Figure 2.20 illustrates the fact that the kriging prediction is the average value of all possible trajactories conditioned on the observed data.

The reason we choose a Gaussian assumption for ξ is that it provides a convenient property to compute the probability of exceeding a threshold u at $x \in \mathbb{X}$ given a set of n observations, $p_n(x)$ (see more detail in Chapter 3, section 3.2.4):

$$p_n(x) = \mathsf{P}_n\{\xi(x) > u\} = \Phi\Big(\frac{\widehat{\xi}_n(x) - u}{\sigma_n(x)}\Big), \tag{2.31}$$

where Φ denotes the cumulative distribution function of the standard normal distribution.

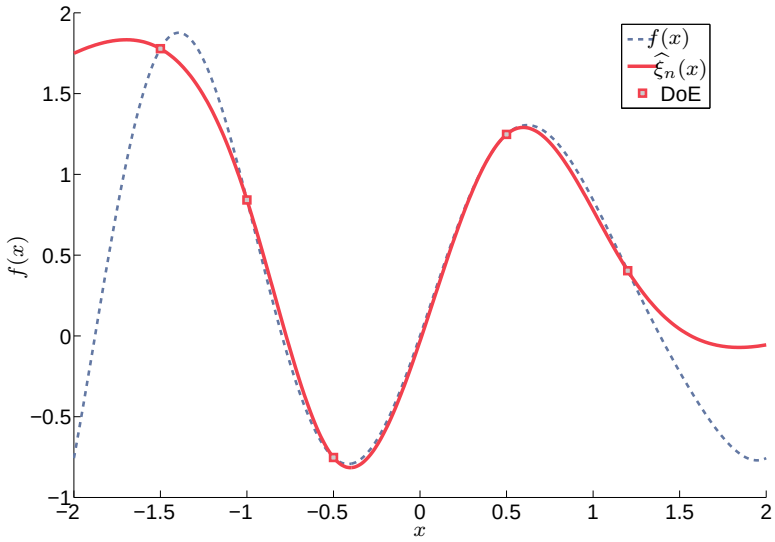


Figure 2.19: Illustration of the interpolation property of the best linear unbiased predictor in a one-dimensional function $f(x) = sin(\pi x) + sin(x^2)$. The squares indicate the $n = 5$ observations $(x_1, f(x_1), \ldots, x_n, f(x_n))$.

All strategies in Chapter 3 are built on Gaussian process priors, e.g., the target IMSE criterion in Picheny et al. (2010), and the criteria based on the marginal distributions in Bichon et al. (2008); Echard et al. (2010a,b); Ranjan et al. (2008), all of which will be reviewed in Chapter 3. The main ideas are as follows:

- Build a cheap approximation $\widehat{\xi}_n$ of f from n evaluations of f at design points $\{x_1, \ldots, x_n\}$;
- If $\widehat{\xi}_n$ is a good approximation of f, $\alpha(\widehat{\xi}_n)$ should be close to $\alpha(f)$ (see Figure 2.21);
- Estimate $\alpha(f)$ by a Monte Carlo estimator $\widehat{\alpha}_{MC}(\widehat{\xi}_n)$, with a large sample size, $\widehat{\alpha}_{MC}(\widehat{\xi}_n)$ will be close to $\alpha(\widehat{\xi}_n)$, thus, close to $\alpha(f)$.

The strategies in Chapter 3 will answer the questions of how to construct the surrogate model $\widehat{\xi}_n$ and how to choose the design points $\{x_1, \ldots, x_n\}$. In summary, the essence of the strategies is to choose a Gaussian process *prior* for f which makes it possible to:

1. Construct a surrogate model from n evaluations by kriging technique.

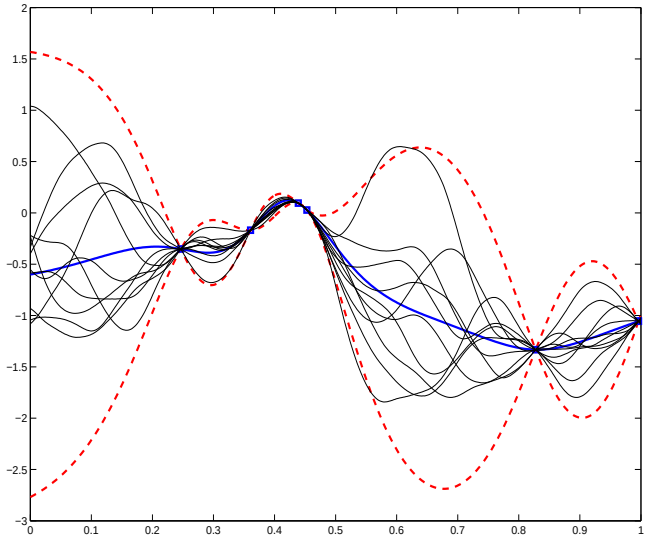


Figure 2.20: A one-dimensional illustration of averaging property of kriging, with 95% confidence. Squares indicate the location of the observed data.The kriging prediction is in solid line.The confidence intervals are in dash lines.

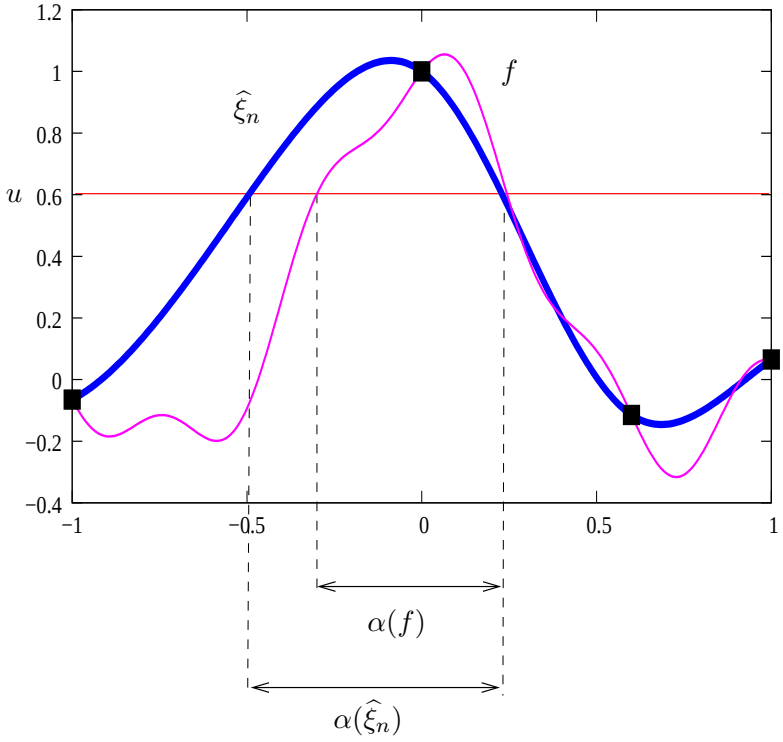


Figure 2.21: Estimation of $\alpha(f)$ by $\alpha(\widehat{\xi}_n)$.

2. Build a criterion that will refine the model by sequentially choosing evaluation points to decrease the error of the estimation of α.

3. Estimate α by a Monte Carlo simulation.

More details about these strategies will be given in Chapter 3.

Conclusions. The strategies reviewed in this section are all based on a Gaussian process model. Specifically, kriging technique as a powerful prediction tool is widely used to build a surrogate model for the performance function. Sequential step aims to refine the surrogate models focused in the failure region according to some designed criteria. This group of methods are gaining more and more interests in the reliability analysis regime. We will explain them in details in Chapter 3.

2.4.5 Summary of the methods based on global approximation

In this section we reviewed several methods based on building a surrogate model for the performance function. When the performance function is expensive-to-evaluate, such methods allow a significant reduction of the computational cost by substituting the original mathematical model with a cheap surrogate model. The algorithms that will be presented in the following Chapters are based on this idea.

2.5 Conclusions

This chapter reviewed several state-of-art reliability methods in literature.

Methods based on geometric approximation such as First Order and Second Order (FORM/SORM) and other kinds of failure region approximations generally do not require a large number of simulations, however, the results obtained may be very inaccurate and it is difficult to assess the quality of the estimators. There exist several approaches to enhance the accuracy of the estimators and improve the performance of the FORM/SORM methods, but they will use more evaluations.

Simple Monte Carlo method is easy to implement. However, it generally requires a large number of evaluations to reach an acceptable accuracy. Variance reduction methods and importance sampling are popular in the literature of reliability analysis. When the probability of failure is small (down to $10^{-6} \sim 10^{-12}$). Multilevel dividing techniques become very powerful. The idea is to gradually refine the failure domain.

When the performance function is very expensive to evaluate, most MC methods, including importance sampling and sequential Monte Carlo methods, become

impracticable. The idea of approximation methods is to substitute the original expensive-to-evaluate performance function with a much cheaper mathematical model. Then classical reliability methods can be applied on this cheap surrogate model. Such approaches may significantly reduce the number of evaluations of the performance function.

Chapter 3

Sequential design of computer experiments for estimating a probability of failure

By J. Bect, D. Ginsbourger, L. Li, V. Picheny and E. Vazquez

Published in *Statistics and Computing*

Contents

Abstract

This paper[1] deals with the problem of estimating the volume of the excursion set of a function $f : \mathbb{R}^d \to \mathbb{R}$ above a given threshold, under a probability measure on $\mathbb{R}^d$ that is assumed to be known. In the industrial world, this corresponds to the problem of estimating a probability of failure of a system. When only an expensive-to-simulate model of the system is available, the budget for simulations is usually severely limited and therefore classical Monte Carlo methods ought to be avoided. One of the main contributions of this article is to derive *SUR* (*stepwise uncertainty reduction*) strategies from a Bayesian-theoretic formulation of the problem of estimating a probability of failure. These sequential strategies use a Gaussian process model of f and aim at performing evaluations of f as efficiently as possible to infer the value of the probability of failure. We compare these strategies to other strategies also based on a Gaussian process model for estimating a probability of failure.

3.1 Introduction

The design of a system or a technological product has to take into account the fact that some design parameters are subject to unknown variations that may affect the reliability of the system. In particular, it is important to estimate the probability of the system to work under abnormal or dangerous operating conditions due to random dispersions of its characteristic parameters. The *probability of failure* of a system is usually expressed as the probability of the excursion set of a function above a fixed threshold. More precisely, let f be a measurable real function defined over a probability space $(\mathbb{X}, \mathcal{B}(\mathbb{X}), \mathsf{P}_{\mathbb{X}})$, with $\mathbb{X} \subseteq \mathbb{R}^d$, and let $u \in \mathbb{R}$ be a threshold. The problem to be considered in this paper is the estimation of the volume, under $\mathsf{P}_{\mathbb{X}}$, of the excursion set

$$\Gamma := \{x \in \mathbb{X} : f(x) > u\} \tag{3.1}$$

of the function f above the level u. In the context of robust design, the volume $\alpha := \mathsf{P}_{\mathbb{X}}(\Gamma)$ can be viewed as the probability of failure of a system: the probability $\mathsf{P}_{\mathbb{X}}$ models the uncertainty on the input vector $x \in \mathbb{X}$ of the system—the components of which are sometimes called *design variables* or *factors*—and f is some deterministic

[1] 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,* Volume 22, Number 3 (2012), 773-793, DOI: 10.1007/s11222-011-9241-4

performance function derived from the outputs of a deterministic model of the system[2]. The evaluation of the outputs of the model for a given set of input factors may involve complex and time-consuming computer simulations, which turns f into an expensive-to-evaluate function. When f is expensive to evaluate, the estimation of α must be carried out with a *restricted number of evaluations* of f, generally excluding the estimation of the probability of excursion by a Monte Carlo approach. Indeed, consider the empirical estimator

$$\alpha_m := \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{\{f(X_i)>u\}} , \tag{3.2}$$

where the X_is are independent random variables with distribution $\mathsf{P}_{\mathbb{X}}$. According to the strong law of large numbers, the estimator α_m converges to α almost surely when m increases. Moreover, it is an unbiased estimator of α, i.e. $\mathsf{E}(\alpha_m) = \alpha$. Its mean square error is

$$\mathsf{E}((\alpha_m - \alpha)^2) = \frac{1}{m}\,\alpha(1-\alpha)\,.$$

If the probability of failure α is small, then the standard deviation of α_m is approximately $\sqrt{\alpha/m}$. To achieve a given standard deviation $\delta\alpha$ thus requires approximately $1/(\delta^2\alpha)$ evaluations, which can be prohibitively high if α is small. By way of illustration, if $\alpha = 2 \times 10^{-3}$ and $\delta = 0.1$, we obtain $m = 50000$. If one evaluation of f takes, say, one minute, then the entire estimation procedure will take about 35 days to complete. Of course, a host of refined random sampling methods have been proposed to improve over the basic Monte Carlo convergence rate; for instance, methods based on importance sampling with cross-entropy (Rubinstein and Kroese, 2004), subset sampling (Au and Beck, 2001) or line sampling (Pradlwarter et al., 2007). They will not be considered here for the sake of brevity and because the required number of function evaluations is still very high.

Until recently, all the methods that do not require a large number of evaluations of f were based on the use of parametric approximations for either the function f itself or the boundary $\partial\Gamma$ of Γ. The so-called response surface method falls in the first category (see, e.g., Bucher and Bourgund, 1990; Rajashekhar and Ellingwood, 1993a, and references therein). The most popular approaches in the second category are the first- and second-order reliability method (FORM and SORM), which are based on a linear or quadratic approximation of $\partial\Gamma$ around the *most probable failure point* (see, e.g., Bjerager, 1990). In all these methods, the accuracy of

[2]Stochastic simulators are also of considerable practical interest, but raise specific modeling and computational issues that will not be considered in this paper.

the estimator depends on the actual shape of either f or $\partial\Gamma$ and its resemblance to the approximant: they do not provide statistically consistent estimators of the probability of failure.

This paper focuses on sequential sampling strategies based on Gaussian processes and kriging, which can been seen as a *non-parametric* approximation method. Several strategies of this kind have been proposed recently in the literature by Ranjan et al. (2008), Bichon et al. (2008), Picheny et al. (2010) and Echard et al. (2010a,b). The idea is that the Gaussian process model, which captures prior knowledge about the unknown function f, makes it possible to assess the uncertainty about the position of Γ given a set of evaluation results. This line of research has its roots in the field of design and analysis of computer experiments (see, e.g., Bayarri et al., 2007; Currin et al., 1991; Oakley, 2004; Oakley and O'Hagan, 2002, 2004; Sacks et al., 1989; Welch et al., 1992). More specifically, kriging-based sequential strategies for the estimation of a probability of failure are closely related to the field of Bayesian global optimization (Ginsbourger, 2009; Jones et al., 1998; Mockus, 1989; Mockus et al., 1978; Villemonteix, 2008; Villemonteix et al., 2009).

The contribution of this paper is twofold. First, we introduce a Bayesian decision-theoretic framework from which the theoretical form of an optimal strategy for the estimation of a probability of failure can be derived. One-step lookahead sub-optimal strategies are then proposed[3], which are suitable for numerical evaluation and implementation on computers. These strategies will be called SUR (stepwise uncertainty reduction) strategies in reference to the work of D. Geman and its collaborators (see, e.g. Fleuret and Geman, 1999). Second, we provide a review in a unified framework of all the kriging-based strategies proposed so far in the literature and compare them numerically with the SUR strategies proposed in this paper.

The outline of the paper is as follows. Section 3.2 introduces the Bayesian framework and recalls the basics of dynamic programming and Gaussian processes. Section 3.3 introduces SUR strategies, from the decision-theoretic underpinnings, down to the implementation level. Section 3.4 provides a review of other kriging-based strategies proposed in the literature. Section 3.5 provides some illustrations and reports an empirical comparison of these sampling criteria. Finally, Section 3.6 presents conclusions and offers perspectives for future work.

[3]Preliminary accounts of this work have been presented in Vazquez and Piera-Martinez (2007) and Vazquez and Bect (2009).

3.2 Bayesian decision-theoretic framework

3.2.1 Bayes risk and sequential strategies

Let f be a continuous function. We shall assume that f corresponds to a computer program whose output is not a closed-form expression of the inputs. Our objective is to obtain a numerical approximation of the probability of failure

$$\alpha(f) = \mathsf{P}_{\mathbb{X}}\{x \in \mathbb{X} : f(x) > u\} = \int_{\mathbb{X}} \mathbf{1}_{f>u} \, \mathrm{d}\mathsf{P}_{\mathbb{X}} \,, \tag{3.3}$$

where $\mathbf{1}_{f>u}$ stands for the characteristic function of the excursion set Γ, such that for any $x \in \mathbb{X}$, $\mathbf{1}_{f>u}(x)$ equals one if $x \in \Gamma$ and zero otherwise. The approximation of $\alpha(f)$ has to be built from a set of computer experiments, where an experiment simply consists in choosing an $x \in \mathbb{X}$ and computing the value of f at x. The result of a pointwise evaluation of f carries information about f and quantities depending on f and, in particular, about $\mathbf{1}_{f>u}$ and $\alpha(f)$. In the context of expensive computer experiments, we shall also suppose that the number of evaluations is limited. Thus, the estimation of $\alpha(f)$ must be carried out using a fixed number, say N, of evaluations of f.

A sequential non-randomized algorithm to estimate $\alpha(f)$ with a budget of N evaluations is a pair $(\underline{X}_N, \widehat{\alpha}_N)$,

$$\underline{X}_N : f \mapsto \underline{X}_N(f) = (X_1(f), X_2(f), \ldots, X_N(f)) \in \mathbb{X}^N \,, \qquad \widehat{\alpha}_N : f \mapsto \widehat{\alpha}_N(f) \in \mathbb{R}_+ \,,$$

with the following properties:

a) There exists $x_1 \in \mathbb{X}$ such that $X_1(f) = x_1$, i.e. X_1 does not depend on f.

b) Let $Z_n(f) = f(X_n(f))$, $1 \leq n \leq N$. For all $1 \leq n < N$, $X_{n+1}(f)$ depends measurably[4] on $\mathcal{I}_n(f)$, where $\mathcal{I}_n = ((X_1, Z_1), \ldots, (X_n, Z_n))$.

c) $\widehat{\alpha}_N(f)$ depends measurably on $\mathcal{I}_N(f)$.

The mapping $\underline{X}_N$ will be referred to as a strategy, or policy, or design of experiments, and $\widehat{\alpha}_N$ will be called an estimator. The algorithm $(\underline{X}_N, \widehat{\alpha}_N)$ describes a sequence of decisions, made from an increasing amount of information: $X_1(f) = x_1$ is chosen prior to any evaluation; for each $n = 1, \ldots, N-1$, the algorithm uses information $\mathcal{I}_n(f)$ to choose the next evaluation point $X_{n+1}(f)$; the estimation $\widehat{\alpha}_N(f)$ of $\alpha(f)$ is the terminal decision. In some applications, the class of sequential algorithms must be further restricted: for instance, when K computer simulations can

[4] i.e., there is a measurable map $\varphi_n : (\mathbb{X} \times \mathbb{R})^n \to \mathbb{X}$ such that $X_n = \varphi_n \circ \mathcal{I}_n$

be run in parallel, algorithms that query batches of K evaluations at a time may be preferred (see, e.g. Ginsbourger et al., 2010). In this paper no such restriction is imposed.

The choice of the estimator $\widehat{\alpha}_N$ will be addressed in Section 3.2.4: for now, we simply assume that an estimator has been chosen, and focus on the problem of finding a good strategy $\underline{X}_N$; that is, one that will produce a good final approximation $\widehat{\alpha}_N(f)$ of $\alpha(f)$. Let $\mathcal{A}_N$ be the class of all strategies $\underline{X}_N$ that query sequentially N evaluations of f. Given a loss function $L : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$, we define the error of approximation of a strategy $\underline{X}_N \in \mathcal{A}_N$ on f as $\varepsilon(\underline{X}_N, f) = L(\widehat{\alpha}_N(f), \alpha(f))$. In this paper, we shall consider the quadratic loss function, so that $\varepsilon(\underline{X}_N, f) = (\widehat{\alpha}_N(f) - \alpha(f))^2$.

We adopt a Bayesian approach to this decision problem: the unknown function f is considered as a sample path of a real-valued random process ξ defined on some probability space $(\Omega, \mathcal{B}, \mathsf{P}_0)$ with parameter in $x \in \mathbb{X}$, and a good strategy is a strategy that achieves, or gets close to, the *Bayes risk* $r_{\mathrm{B}} := \inf_{\underline{X}_N\in\mathcal{A}_N} \mathsf{E}_0\left(\varepsilon(\underline{X}_N, \xi)\right)$, where E_0 denotes the expectation with respect to P_0. From a subjective Bayesian point of view, the stochastic model ξ is a representation of our uncertain initial knowledge about f. From a more pragmatic perspective, the prior distribution can be seen as a tool to define a notion of a good strategy in an average sense. Another interesting route, not followed in this paper, would have been to consider the minimax risk $\inf_{\underline{X}_N\in\mathcal{A}_N} \max_f \mathsf{E}_0\left(\varepsilon(\underline{X}_N, \xi)\right)$ over some class of functions.

Notations. From now on, we shall consider the stochastic model ξ instead of the deterministic function f and, for abbreviation, the explicit dependence on ξ will be dropped when no there is no risk of confusion; e.g., $\widehat{\alpha}_N$ will denote the random variable $\widehat{\alpha}_N(\xi)$, X_n will denote the random variable $X_n(\xi)$, etc. We will use the notations $\mathcal{F}_n$, P_n and E_n to denote respectively the σ-algebra generated by $\mathcal{I}_n$, the conditional distribution $\mathsf{P}_0\left(\cdot \mid \mathcal{F}_n\right)$ and the conditional expectation $\mathsf{E}_0\left(\cdot \mid \mathcal{F}_n\right)$. Note that the dependence of X_{n+1} on $\mathcal{I}_n$ can be rephrased by saying that X_{n+1} is $\mathcal{F}_n$-measurable. Recall that $\mathsf{E}_n\left(Z\right)$ is $\mathcal{F}_n$-measurable, and thus can be seen as a measurable function of $\mathcal{I}_n$, for any random variable Z.

3.2.2 Optimal and k-step lookahead strategies

It is well-known (see, e.g., Berry and Fristedt, 1985; Bertsekas, 1995; Mockus, 1989) that an optimal strategy for such a finite horizon problem[5], i.e. a strategy $\underline{X}^\star_N \in \mathcal{A}_N$

[5] in other words, a sequential decision problem where the total number of steps to be performed is known from the start

such that $\mathsf{E}_0\left(\varepsilon(\underline{X}_N^\star, \xi)\right) = r_{\mathrm{B}}$, can be formally obtained by *dynamic programming*: let $R_N = \mathsf{E}_N\left(\varepsilon(\underline{X}_N, \xi)\right) = \mathsf{E}_N\big((\widehat{\alpha}_N - \alpha)^2\big)$ denote the terminal risk and define by backward induction

$$R_n = \min_{x \in \mathbb{X}} \; \mathsf{E}_n\big(R_{n+1} \mid X_{n+1} = x\big)\,, \quad n = N-1, \ldots, 0. \tag{3.4}$$

To get an insight into (3.4), notice that R_{n+1}, $n = 0, \ldots, N-1$, depends measurably on $\mathcal{I}_{n+1} = (\mathcal{I}_n, X_{n+1}, Z_{n+1})$, so that $\mathsf{E}_n\big(R_{n+1} \mid X_{n+1} = x\big)$ is in fact an expectation with respect to Z_{n+1}, and R_n is an $\mathcal{F}_n$-measurable random variable. Then, we have $R_0 = r_{\mathrm{B}}$, and the strategy $\underline{X}_N^\star$ defined by

$$X_{n+1}^\star = \arg\min_{x \in \mathbb{X}} \mathsf{E}_n\big(R_{n+1} \mid X_{n+1} = x\big)\,, \quad n = 1, \ldots, N-1, \tag{3.5}$$

is optimal[6]. It is crucial to observe here that, for this dynamic programming problem, both the space of possible actions and the space of possible outcomes at each step are continuous, and the state space $(\mathbb{X} \times \mathbb{R})^n$ at step n is of dimension $n(d+1)$. Any direct attempt at solving (3.4)–(3.5) numerically, over an horizon N of more than a few steps, will suffer from the curse of dimensionality.

Using (3.4), the optimal strategy can be expanded as

$$X_{n+1}^\star = \arg\min_{x \in \mathbb{X}} \mathsf{E}_n \left(\min_{X_{n+2}} \mathsf{E}_{n+1} \ldots \min_{X_N} \mathsf{E}_{N-1} R_N \;\middle|\; X_{n+1} = x \right).$$

A very general approach to construct sub-optimal—but hopefully good—strategies is to truncate this expansion after k terms, replacing the exact risk R_{n+k} by any available surrogate $\widetilde{R}_{n+k}$. Examples of such surrogates will be given in Sections 3.3 and 3.4. The resulting strategy,

$$X_{n+1} = \arg\min_{x \in \mathbb{X}} \mathsf{E}_n \left(\min_{X_{n+2}} \mathsf{E}_{n+1} \ldots \min_{X_{n+k}} \mathsf{E}_{n+k-1} \widetilde{R}_{n+k} \;\middle|\; X_{n+1} = x \right). \tag{3.6}$$

is called a *k-step lookahead strategy* (see, e.g., Bertsekas, 1995, Section 6.3). Note that both the optimal strategy (3.5) and the k-step lookahead strategy implicitly define a *sampling criterion* $J_n(x)$, $\mathcal{F}_n$-measurable, the minimum of which indicates the next evaluation to be performed. For instance, in the case of the k-step lookahead strategy, the sampling criterion is

$$J_n(x) = \mathsf{E}_n \left(\min_{X_{n+2}} \mathsf{E}_{n+1} \ldots \min_{X_{n+k}} \mathsf{E}_{n+k-1} \widetilde{R}_{n+k} \;\middle|\; X_{n+1} = x \right).$$

[6]Proving rigorously that, for a given P_0 and $\widehat{\alpha}_N$, equations (3.4) and (3.5) actually define a (measurable!) strategy $\underline{X}_N^\star \in \mathcal{A}_N$ is technical problem that is not of primary interest in this paper. This can be done for instance, in the case of a Gaussian process with continuous covariance function (as considered later), by proving that $x \mapsto \mathsf{E}_n\left(R_{n+1} \mid X_{n+1}(\xi) = x\right)$ is a continuous function on $\mathbb{X}$ and then using a measurable selection theorem.

In the rest of the paper, we restrict our attention to the class of one-step lookahead strategies, which is, as we shall see in Section 3.3, large enough to provide very efficient algorithms. We leave aside the interesting question of whether more complex k-step lookahead strategies (with $k \geq 2$) could provide a significant improvement over the strategies examined in this paper.

Remark 3. *In practice, the analysis of a computer code usually begins with an exploratory phase, during which the output of the code is computed on a* space-filling design *of size $n_0 < N$ (see, e.g., Santner et al., 2003b). Such an exploratory phase will be colloquially referred to as the* initial design. *Sequential strategies such as (3.5) and (3.6) are meant to be used after this initial design, at steps $n_0 + 1$, ..., N. An important (and largely open) question is the choice of the size n_0 of the initial design, for a given global budget N. As a rule of thumb, some authors recommend to start with a sample size proportional to the dimension d of the input space $\mathbb{X}$, for instance $n_0 = 10\,d$; see Loeppky et al. (2009) and the references therein.*

3.2.3 Gaussian process priors

Restricting ξ to be a Gaussian process makes it possible to deal with the conditional distributions P_n and conditional expectations E_n that appear in the strategies above. The idea of modeling an unknown function f by a Gaussian process has originally been introduced circa 1960 in time series analysis (Parzen, 1962), optimization theory (Kushner, 1964) and geostatistics (see, e.g., Chilès and Delfiner, 1999, and the references therein). Today, the Gaussian process model plays a central role in the design and analysis of computer experiments (see, e.g., Currin et al., 1991; Sacks et al., 1989; Santner et al., 2003b; Welch et al., 1992). Recall that the distribution of a Gaussian process ξ is uniquely determined by its mean function $m(x) := \mathsf{E}_0(\xi(x))$, $x \in \mathbb{X}$, and its covariance function $k(x, y) := \mathsf{E}_0\left((\xi(x) - m(x))(\xi(y) - m(y))\right)$, $x, y \in \mathbb{X}$. Hereafter, we shall use the notation $\xi \sim \mathrm{GP}\,(m,\, k)$ to say that ξ is a Gaussian process with mean function m and covariance function k.

Let $\xi \sim \mathrm{GP}\,(0,\, k)$ be a zero-mean Gaussian process. The best linear unbiased predictor (BLUP) of $\xi(x)$ from observations $\xi(x_i)$, $i = 1, \ldots, n$, also called the *kriging predictor* of $\xi(x)$, is the orthogonal projection

$$\widehat{\xi}(x; \underline{x}_n) \;:=\; \sum_{i=1}^{n} \lambda_i(x; \underline{x}_n)\, \xi(x_i) \tag{3.7}$$

of $\xi(x)$ onto $\text{span}\{\xi(x_i), i = 1, \ldots, n\}$. Here, the notation $\underline{x}_n$ stands for the set of points $\underline{x}_n = \{x_1, \ldots, x_n\}$. The weights $\lambda_i(x; \underline{x}_n)$ are the solutions of a system of linear equations

$$k(\underline{x}_n, \underline{x}_n)\lambda(x; \underline{x}_n) = k(x, \underline{x}_n) \tag{3.8}$$

where $k(\underline{x}_n, \underline{x}_n)$ stands for the $n \times n$ covariance matrix of the observation vector, $\lambda(x; \underline{x}_n) = (\lambda_1(x; \underline{x}_n), \ldots, \lambda_n(x; \underline{x}_n))^\mathsf{T}$, and $k(x, \underline{x}_n)$ is a vector with entries $k(x, x_i)$. The function $x \mapsto \widehat{\xi}(x; \underline{x}_n)$ conditioned on $\xi(x_1) = f(x_1), \ldots, \xi(x_n) = f(x_n)$, is deterministic, and provides a cheap *surrogate model* for the true function f (see, e.g., Santner et al., 2003b). The covariance function of the error of prediction, also called *kriging covariance* is given by

$$\begin{aligned} k(x, y; \underline{x}_n) &:= \text{cov}\left(\xi(x) - \widehat{\xi}(x; \underline{x}_n), \xi(y) - \widehat{\xi}(y; \underline{x}_n)\right) \\ &= k(x, y) - \sum_i \lambda_i(x; \underline{x}_n)\, k(y, x_i)\,. \end{aligned} \tag{3.9}$$

The variance of the prediction error, also called the *kriging variance*, is defined as $\sigma^2(x; \underline{x}_n) = k(x, x; \underline{x}_n)$. One fundamental property of a zero-mean Gaussian process is the following (see, e.g., Chilès and Delfiner, 1999, Chapter 3) :

Proposition 1. *If $\xi \sim \text{GP}(0, k)$, then the random process ξ conditioned on the σ-algebra $\mathcal{F}_n$ generated by $\xi(x_1), \ldots, \xi(x_n)$, which we shall denote by $\xi \mid \mathcal{F}_n$, is a Gaussian process with mean $\widehat{\xi}(\,\cdot\,; \underline{x}_n)$ given by (3.7)-(3.8) and covariance $k(\,\cdot\,, \cdot\,; \underline{x}_n)$ given by (3.9). In particular, $\widehat{\xi}(x; \underline{x}_n) = \mathsf{E}_0(\xi(x) \mid \mathcal{F}_n)$ is the best $\mathcal{F}_n$-measurable predictor of $\xi(x)$, for all $x \in \mathbb{X}$.*

In the domain of computer experiments, the mean of a Gaussian process is generally written as a linear parametric function

$$m(\,\cdot\,) = \beta^\mathsf{T} h(\,\cdot\,)\,, \tag{3.10}$$

where β is a vector of unknown parameters, and $h = (h_1, \ldots, h_l)^\mathsf{T}$ is an l-dimensional vector of functions (in practice, polynomials). The simplest case is when the mean function is assumed to be an unknown constant m, in which case we can take $\beta = m$ and $h : x \in \mathbb{X} \mapsto 1$. The covariance function is generally written as a translation-invariant function:

$$k : (x, y) \in \mathbb{X}^2 \mapsto \sigma^2\, \rho_\theta(x - y)\,,$$

where σ^2 is the variance of the (stationary) Gaussian process and ρ_θ is the correlation function, which generally depends on a parameter vector θ. When the mean

is written under the form (3.10), the kriging predictor is again a linear combination of the observations, as in (3.7), and the weights $\lambda_i(x;\underline{x}_n)$ are again solutions of a system of linear equations (see, e.g., Chilès and Delfiner, 1999), which can be written under a matrix form as

$$\begin{pmatrix} k(\underline{x}_n,\underline{x}_n) & h(\underline{x}_n)^\mathsf{T} \\ h(\underline{x}_n) & 0 \end{pmatrix} \begin{pmatrix} \lambda(x;\underline{x}_n) \\ \mu(x) \end{pmatrix} = \begin{pmatrix} k(x,\underline{x}_n) \\ h(x) \end{pmatrix}, \tag{3.11}$$

where $h(\underline{x}_n)$ is an $l \times n$ matrix with entries $h_i(x_j)$, $i = 1,\ldots,l$, $j = 1,\ldots,n$, μ is a vector of Lagrange coefficients ($k(\underline{x}_n,\underline{x}_n)$, $\lambda(x;\underline{x}_n)$, $k(x,\underline{x}_n)$ as above). The kriging covariance function is given in this case by

$$\begin{aligned} k(x,y;\underline{x}_n) &:= \operatorname{cov}\left(\xi(x) - \widehat{\xi}(x;\underline{x}_n), \xi(y) - \widehat{\xi}(y;\underline{x}_n)\right) \\ &= k(x,y) - \lambda(x;\underline{x}_n)^\mathsf{T}\, k(y,\underline{x}_n) - \mu(x)^\mathsf{T} h(y)\,. \end{aligned} \tag{3.12}$$

The following result holds (Kimeldorf and Wahba, 1970; O'Hagan, 1978):

Proposition 2. *Let k be a covariance function.*

$$\textit{If } \begin{cases} \xi \mid m \sim \mathrm{GP}(m, k) \\ m : x \mapsto \beta^\mathsf{T} h(x),\ \beta \sim \mathcal{U}_{\mathbb{R}^l} \end{cases} \textit{ then } \xi \mid \mathcal{F}_n \sim \mathrm{GP}\left(\widehat{\xi}(\cdot;\underline{x}_n),\, k(\cdot,\cdot;\underline{x}_n)\right),$$

where $\mathcal{U}_{\mathbb{R}^l}$ stands for the (improper) uniform distribution over $\mathbb{R}^l$, and where $\widehat{\xi}(\cdot;\underline{x}_n)$ and $k(\cdot,\cdot;\underline{x}_n)$ are given by (3.7), (3.11) and (3.12).

Proposition 2 justifies the use of kriging in a Bayesian framework provided that the covariance function of ξ is known. However, the covariance function is rarely assumed to be known in applications. Instead, the covariance function is generally taken in some parametric class (in this paper, we use the so-called Matérn covariance function, see Appendix A). A *fully Bayesian* approach also requires to choose a prior distribution for the unknown parameters of the covariance (see, e.g., Handcock and Stein, 1993; Kennedy and O'Hagan, 2001; Paulo, 2005). Sampling techniques (Monte Carlo Markov Chains, Sequential Monte Carlo...) are then generally used to approximate the posterior distribution of the unknown covariance parameters. Very often, the popular *empirical Bayes* approach is used instead, which consists in plugging-in the maximum likelihood (ML) estimate to approximate the posterior distribution of ξ. This approach has been used in previous papers about contour estimation or probability of failure estimation (Bichon et al., 2008; Picheny et al., 2010; Ranjan et al., 2008). In Section 3.5.2 we will adopt a plug-in approach as well.

Simplified notations. In the rest of the paper, we shall use the following simplified notations when there is no risk of confusion: $\widehat{\xi}_n(x) := \widehat{\xi}(x; \underline{X}_n)$, $\sigma_n^2(x) := \sigma^2(x; \underline{X}_n)$.

3.2.4 Estimators of the probability of failure

Given a random process ξ and a strategy $\underline{X}_N$, the optimal estimator that minimizes $\mathsf{E}_0\left((\alpha - \widehat{\alpha}_n)^2\right)$ among all $\mathcal{F}_n$-measurable estimators $\widehat{\alpha}_n$, $1 \leq n \leq N$, is

$$\widehat{\alpha}_n \;=\; \mathsf{E}_n\left(\alpha\right) \;=\; \mathsf{E}_n\left(\int_{\mathbb{X}} \mathbf{1}_{\xi>u}\, \mathrm{dP}_{\mathbb{X}}\right) \;=\; \int_{\mathbb{X}} p_n\, \mathrm{dP}_{\mathbb{X}}\,, \tag{3.13}$$

where

$$p_n : x \in \mathbb{X} \mapsto \mathsf{P}_n\left\{\xi(x) > u\right\}\,. \tag{3.14}$$

When ξ is a Gaussian process, the probability $p_n(x)$ of exceeding u at $x \in \mathbb{X}$ given $\mathcal{I}_n$ has a simple closed-form expression:

$$p_n(x) \;=\; 1 \,-\, \Phi\left(\frac{u - \widehat{\xi}_n(x)}{\sigma_n(x)}\right) \;=\; \Phi\left(\frac{\widehat{\xi}_n(x) - u}{\sigma_n(x)}\right)\,, \tag{3.15}$$

where Φ is the cumulative distribution function of the normal distribution. Thus, in the Gaussian case, the estimator (3.13) is amenable to a numerical approximation, by integrating the excess probability p_n over $\mathbb{X}$ (for instance using Monte Carlo sampling, see Section 3.3.3).

Another natural way to obtain an estimator of α given $\mathcal{I}_n$ is to approximate the excess indicator $\mathbf{1}_{\xi>u}$ by a hard classifier $\eta_n : \mathbb{X} \to \{0, 1\}$, where "hard" refers to the fact that η_n takes its values in $\{0, 1\}$. If η_n is close in some sense to $\mathbf{1}_{\xi>u}$, the estimator

$$\widehat{\alpha}_n = \int_{\mathbb{X}} \eta_n \mathrm{dP}_{\mathbb{X}} \tag{3.16}$$

should be close to α. More precisely,

$$\mathsf{E}_n\left((\widehat{\alpha}_n - \alpha)^2\right) = \mathsf{E}_n\left[\left(\int (\eta_n - \mathbf{1}_{\xi>u}) \mathrm{dP}_{\mathbb{X}}\right)^2\right] \leq \int \mathsf{E}_n\left((\eta_n - \mathbf{1}_{\xi>u})^2\right) \mathrm{dP}_{\mathbb{X}}\,. \tag{3.17}$$

Let $\tau_n(x) = \mathsf{P}_n\{\eta_n(x) \neq \mathbf{1}_{\xi(x)>u}\} = \mathsf{E}_n\left((\eta_n(x) - \mathbf{1}_{\xi(x)>u})^2\right)$ be the probability of misclassification; that is, the probability to predict a point above (resp. under) the threshold when the true value is under (resp. above) the threshold. Thus, (3.17) shows that it is desirable to use a classifier η_n such that τ_n is small for all $x \in \mathbb{X}$. For instance, the method called SMART (Deheeger and Lemaire, 2007) uses a support vector machine to build η_n. Note that

$$\tau_{n(x)} \;=\; p_n(x) \,+\, (1 - 2p_n(x))\, \eta_n(x)\,.$$

Therefore, the right-hand side of (3.17) is minimized if we set

$$\eta_n(x) = \mathbf{1}_{p_n(x)>1/2} = \mathbf{1}_{\bar{\xi}_n(x)>u}\,, \tag{3.18}$$

where $\bar{\xi}_n(x)$ denotes the posterior median of $\xi(x)$. Then, we have

$$\tau_n(x) = \min(p_n(x), 1 - p_n(x)).$$

In the case of a Gaussian process, the posterior median and the posterior mean are equal. Then, the classifier that minimizes $\tau_n(x)$ for each $x \in \mathbb{X}$ is $\eta_n = \mathbf{1}_{\widehat{\xi}_n>u}$, in which case

$$\tau_n(x) \;=\; \mathsf{P}_n\Big((\xi(x)-u)(\widehat{\xi}_n(x)-u)<0\Big) \;=\; 1 \;-\; \Phi\left(\frac{|\widehat{\xi}_n(x)-u|}{\sigma_n(x)}\right). \tag{3.19}$$

Notice that for $\eta_n = \mathbf{1}_{\widehat{\xi}_n>u}$, we have $\widehat{\alpha}_n = \alpha(\widehat{\xi}_n)$. Therefore, this approach to obtain an estimator of α can be seen as a type of plug-in estimation.

Standing assumption. It will assumed in the rest of the paper that ξ is a Gaussian process, or more generally that $\xi \mid \mathcal{F}_n \sim \mathrm{GP}(\widehat{\xi}_n, k(\cdot,\cdot;\underline{x}_n))$ for all $n \geq 1$ as in Proposition 2.

3.3 Stepwise uncertainty reduction

3.3.1 Principle

A very natural and straightforward way of building a one-step lookahead strategy is to select *greedily* each evaluation as if it were the last one. This kind of strategy, sometimes called *myopic*, has been successfully applied in the field of Bayesian global optimization (Mockus, 1989; Mockus et al., 1978), yielding the famous *expected improvement* criterion later popularized in the Efficient Global Optimization (EGO) algorithm of Jones et al. (1998).

When the Bayesian risk provides a measure of the estimation error or uncertainty (as in the present case), we call such a strategy a *stepwise uncertainty reduction* (SUR) strategy. In the field of global optimization, the Informational Approach to Global Optimization (IAGO) of Villemonteix et al. (2009) is an example of a SUR strategy, where the Shannon entropy of the minimizer is used instead of the quadratic cost. When considered in terms of utility rather than cost, such strategies have also been called *knowledge gradient policies* by Frazier et al. (2008).

Given a sequence of estimators $(\widehat{\alpha}_n)_{n\geq 1}$, a direct application of the above principle using the quadratic loss function yields the sampling criterion

$$J_n(x) = \mathsf{E}_n\left((\alpha - \widehat{\alpha}_{n+1})^2 \mid X_{n+1} = x\right). \tag{3.20}$$

Having found no closed-form expression for this criterion, and no efficient numerical procedure for its approximation, we will proceed by upper-bounding and discretizing (3.20) in order to get an expression that will lend itself to a numerically tractable approximation. By doing so, several SUR strategies will be derived, depending on the choice of estimator (the posterior mean (3.13) or the plug-in estimator (3.16) with (3.18)) and bounding technique.

3.3.2 Upper bounds of the SUR sampling criterion

Recall that $\tau_n(x) = \min(p_n(x), 1 - p_n(x))$ is the probability of misclassification at x using the optimal classifier $\mathbf{1}_{\widehat{\xi}_n(x)>u}$. Let us further denote by $\nu_n(x) := p_n(x)\,(1 - p_n(x))$ the variance of the excess indicator $\mathbf{1}_{\xi(x)\geq u}$.

Proposition 3. *Assume that either* $\widehat{\alpha}_n = \mathsf{E}_n(\alpha)$ *or* $\widehat{\alpha}_n = \int \mathbf{1}_{\widehat{\xi}_n\geq u}\mathrm{d}\mathsf{P}_{\mathbb{X}}$. *Define* $G_n := \int_{\mathbb{X}} \sqrt{\gamma_n(y)}\mathrm{d}\mathsf{P}_{\mathbb{X}}$ *for all* $n \in \{0, \ldots, N-1\}$, *with*

$$\gamma_n := \begin{cases} \nu_n = p_n(1-p_n) = \tau_n(1-\tau_n), & \text{if } \widehat{\alpha}_n = \mathsf{E}_n(\alpha), \\ \tau_n = \min(p_n, 1-p_n), & \text{if } \widehat{\alpha}_n = \int \mathbf{1}_{\widehat{\xi}_n\geq u}\mathrm{d}\mathsf{P}_{\mathbb{X}}. \end{cases}$$

Then, for all $x \in \mathbb{X}$ *and all* $n \in \{0, \ldots, N-1\}$,

$$J_n(x) \;\leq\; \widetilde{J}_n(x) \;:=\; \mathsf{E}_n\left(G_{n+1}^2 \mid X_{n+1} = x\right).$$

Note that $\gamma_n(x)$ is a function of $p_n(x)$ that vanishes at 0 and 1, and reaches its maximum at 1/2; that is, when the uncertainty on $\mathbf{1}_{\widehat{\xi}_n(x)>u}$ is maximal (see Figure 3.1).

Proof. First, observe that, for all $n \geq 0$, $\alpha - \widehat{\alpha}_n = \int U_n \,\mathrm{d}\mathsf{P}_{\mathbb{X}}$, with

$$U_n : x \in \mathbb{X} \mapsto U_n(x) \;=\; \begin{cases} \mathbf{1}_{\xi(x)>u} - p_n(x) & \text{if } \widehat{\alpha}_n = \mathsf{E}_n(\alpha), \\ \mathbf{1}_{\xi(x)>u} - \mathbf{1}_{\widehat{\xi}_n(x)>u} & \text{if } \widehat{\alpha}_n = \int \mathbf{1}_{\widehat{\xi}_n\geq u}\mathrm{d}\mathsf{P}_{\mathbb{X}}. \end{cases} \tag{3.21}$$

Moreover, note that $\gamma_n = \|U_n\|_n^2$ in both cases, where $\|\cdot\|_n : L^2(\Omega, \mathcal{B}, \mathsf{P}) \to L^2(\Omega, \mathcal{F}_n, \mathsf{P})$, $W \mapsto \mathsf{E}_n(W^2)^{1/2}$. Then, using the generalized Minkowski inequality (see, e.g., Vestrup, 2003, section 10.7) we get that

$$\left\|\int U_n \,\mathrm{d}\mathsf{P}_{\mathbb{X}}\right\|_n \;\leq\; \int \|U_n\|_n \,\mathrm{d}\mathsf{P}_{\mathbb{X}} \;=\; \int \sqrt{\gamma_n}\,\mathrm{d}\mathsf{P}_{\mathbb{X}} \;=\; G_n. \tag{3.22}$$

Finally, it follows from the tower property of conditional expectations and (3.22) that, for all $n \geq 0$,

$$\begin{aligned} J_n(x) &= \mathsf{E}_n\left(\|\alpha - \widehat{\alpha}_{n+1}\|_{n+1}^2 \mid X_{n+1} = x\right) \\ &= \mathsf{E}_n\left(\left\|\int U_{n+1}\,\mathrm{d}\mathsf{P}_{\mathbb{X}}\right\|_{n+1}^2 \,\middle|\, X_{n+1} = x\right) \\ &\leq \mathsf{E}_n\left(G_{n+1}^2 \mid X_{n+1} = x\right). \end{aligned}$$

□

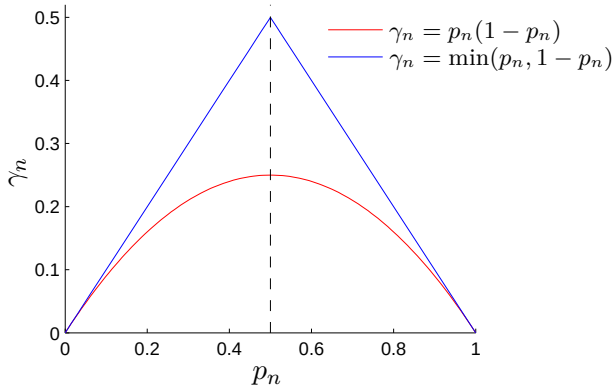


Figure 3.1: γ_n as a function of p_n (see Proposition 3). In both cases, γ_n is maximum at $p_n = 1/2$.

Note that two other upper-bounding sampling criteria readily follow from those of Proposition 3, by using the Cauchy-Schwarz inequality in $L^2\left(\mathbb{X}, \mathcal{B}(\mathbb{X}), \mathsf{P}_{\mathbb{X}}\right)$:

$$\widetilde{J}_n(x) \;\leq\; \mathsf{E}_n\left(\int \gamma_{n+1}\,\mathrm{d}\mathsf{P}_{\mathbb{X}} \,\middle|\, X_{n+1} = x\right). \tag{3.23}$$

As a result, we can write four SUR criteria, whose expressions are summarized in Table 3.1. Criterion $J_{1,n}^{\mathrm{SUR}}$ has been proposed in the PhD thesis of Piera-Martinez (2008) and in conference papers (Vazquez and Bect, 2009; Vazquez and Piera-Martinez, 2007); the other ones, to the best of our knowledge, are new. Each criterion is expressed as the conditional expectation of some (possibly squared) $\mathcal{F}_{n+1}$-measurable integral criterion, with an integrand that can be expressed as a function of the probability of misclassification τ_{n+1}. It is interesting to note that the integral in J_4^{SUR} is the integrated mean square error (IMSE)[7] for the process $\mathbf{1}_{\xi>u}$.

[7] The IMSE criterion is usually applied to the response surface ξ itself (see, e.g., Box and Draper, 1987; Sacks et al., 1989). The originality here is to consider the IMSE of the process $\mathbf{1}_{\xi>u}$ instead. Another way of adapting the IMSE criterion for the estimation of a probability of failure, proposed by Picheny et al. (2010), is recalled in Section 3.4.2.

Remark 4. *The conclusions of Proposition 3 still hold in the general case when ξ is not assumed to be a Gaussian process, provided that the posterior median $\bar{\xi}_n$ is substituted to posterior the mean $\widehat{\xi}_n$.*

Table 3.1: Expressions of four SUR-type criteria.

SUR-type sampling criterion	How it is obtained
$J_{1,n}^{\mathrm{SUR}}(x) = \mathsf{E}_n\Big(\big(\int \sqrt{\tau_{n+1}}\, \mathrm{d}\mathsf{P}_{\mathbb{X}}\big)^2 \;\Big|\; X_{n+1} = x \Big)$	Prop. 3 with $\widehat{\alpha}_n = \int \mathbf{1}_{\widehat{\xi}_n > u}\, \mathrm{d}\mathsf{P}_{\mathbb{X}}$
$J_{2,n}^{\mathrm{SUR}}(x) = \mathsf{E}_n\Big(\big(\int \sqrt{\nu_{n+1}}\, \mathrm{d}\mathsf{P}_{\mathbb{X}}\big)^2 \;\Big|\; X_{n+1} = x \Big)$	Prop. 3 with $\widehat{\alpha}_n = \mathsf{E}_n(\alpha)$
$J_{3,n}^{\mathrm{SUR}}(x) = \mathsf{E}_n\Big(\int \tau_{n+1}\, \mathrm{d}\mathsf{P}_{\mathbb{X}} \;\Big|\; X_{n+1} = x \Big)$	Eq. (3.23) with $\widehat{\alpha}_n = \int \mathbf{1}_{\widehat{\xi}_n > u}\, \mathrm{d}\mathsf{P}_{\mathbb{X}}$
$J_{4,n}^{\mathrm{SUR}}(x) = \mathsf{E}_n\Big(\int \nu_{n+1}\, \mathrm{d}\mathsf{P}_{\mathbb{X}} \;\Big|\; X_{n+1} = x \Big)$	Eq. (3.23) with $\widehat{\alpha}_n = \mathsf{E}_n(\alpha)$

3.3.3 Discretizations

In this section, we proceed with the necessary integral discretizations of the SUR criteria to make them suitable for numerical evaluation and implementation on computers. Assume that n steps of the algorithm have already been performed and consider, for instance, the criterion

$$J_{3,n}^{\mathrm{SUR}}(x) = \mathsf{E}_n\Big(\int \tau_{n+1}(y)\, \mathsf{P}_{\mathbb{X}}(\mathrm{d}y) \;\Big|\; X_{n+1} = x \Big). \tag{3.24}$$

Remember that, for each $y \in \mathbb{X}$, the probability of misclassification $\tau_{n+1}(y)$ is $\mathcal{F}_{n+1}$-measurable and, therefore, is a function of $\mathcal{I}_{n+1} = (\mathcal{I}_n, X_{n+1}, Z_{n+1})$. Since $\mathcal{I}_n$ is known at this point, we introduce the notation $v_{n+1}(y; X_{n+1}, Z_{n+1}) = \tau_{n+1}(y)$ to emphasize the fact that, when a new evaluation point must be chosen at step $(n+1)$, $\tau_{n+1}(y)$ depends on the choice of X_{n+1} and the random outcome Z_{n+1}. Let us further denote by $\mathsf{Q}_{n,x}$ the probability distribution of $\xi(x)$ under P_n. Then, (3.24) can be rewritten as

$$J_{3,n}^{\mathrm{SUR}}(x) = \iint_{\mathbb{R}\times\mathbb{X}} v_{n+1}(y; x, z)\, \mathsf{Q}_{n,x}(\mathrm{d}z)\, \mathsf{P}_{\mathbb{X}}(\mathrm{d}y)\,,$$

and the corresponding strategy is:

$$X_{n+1} \;=\; \operatorname*{arg\,min}_{x\in\mathbb{X}} \iint_{\mathbb{R}\times\mathbb{X}} v_{n+1}(y; x, z)\, \mathsf{Q}_{n,x}(\mathrm{d}z)\, \mathsf{P}_{\mathbb{X}}(\mathrm{d}y)\,. \tag{3.25}$$

Given $\mathcal{I}_n$ and a triple (x, y, z), $v_{n+1}(y; x, z)$ can be computed efficiently using the equations provided in Sections 3.2.3 and 3.2.4.

At this point, we need to address: 1) the computation of the integral on $\mathbb{X}$ with respect to $\mathsf{P}_{\mathbb{X}}$; 2) the computation of the integral on $\mathbb{R}$ with respect to $\mathsf{Q}_{n,x}$; 3) the minimization of the resulting criterion with respect to $x \in \mathbb{X}$.

To solve the first problem, we draw an i.i.d. sequence $Y_1, \ldots, Y_m \sim \mathsf{P}_{\mathbb{X}}$ and use the Monte Carlo approximation:

$$\int_{\mathbb{X}} v_{n+1}(y; x, z)\, \mathsf{P}_{\mathbb{X}}(\mathrm{d}y) \;\approx\; \frac{1}{m} \sum_{j=1}^{m} v_{n+1}(Y_j; x, z).$$

An increasing sample size $n \mapsto m_n$ should be used to build a convergent algorithm for the estimation of α (possibly with a different sequence $Y_{n,1}, \ldots, Y_{n,m_n}$ at each step). In this paper we adopt a different approach instead, which is to take a fixed sample size $m > 0$ and keep the same sample $Y_1, \ldots, Y_m$ throughout the iterations. Equivalently, it means that we choose to work from the start on a discretized version of the problem: we replace $\mathsf{P}_{\mathbb{X}}$ by the empirical distribution $\widehat{\mathsf{P}}_{\mathbb{X},n} = \frac{1}{m}\sum_{j=1}^{m} \delta_{Y_j}$, and our goal is now to *estimate the Monte Carlo estimator* $\alpha_m = \int \mathbf{1}_{\xi>u} \mathrm{d}\widehat{\mathsf{P}}_{\mathbb{X},n} = \frac{1}{m}\sum_{j=1}^{m} \mathbf{1}_{\xi(Y_j)>u}$, using either the posterior mean $\mathsf{E}_n(\alpha_m) = \frac{1}{m}\sum_j p_n(Y_j)$ or the plug-in estimate $\frac{1}{m}\sum_j \mathbf{1}_{\widehat{\xi}(Y_j;\underline{X}_n)>u}$. This kind of approach has be coined *meta-estimation* by Arnaud et al. (2010): the objective is to estimate the value of a precise Monte Carlo estimator of $\alpha(f)$ (m being large), using prior information on f to alleviate the computational burden of running m times the computer code f. This point of view also underlies the work in structural reliability of Hurtado (2004b, 2007), Deheeger and Lemaire (2007), Deheeger (2008), and more recently Echard et al. (2010a,b).

The new point of view also suggests a natural solution for the third problem, which is to replace the continuous search for a minimizer $x \in \mathbb{X}$ by a discrete search over the set $\mathbb{X}_m := \{Y_1, \ldots, Y_m\}$. This is obviously sub-optimal, even in the meta-estimation framework introduced above, since picking $x \in \mathbb{X} \setminus \mathbb{X}_m$ can sometimes bring more information about $\xi(Y_1), \ldots, \xi(Y_m)$ than the best possible choice in $\mathbb{X}_m$. Global optimization algorithms may of course be used to tackle directly the continuous search problem: for instance, Ranjan et al. (2008) use a combination of a genetic algorithm and local search technique, Bichon et al. (2008) use the DIRECT algorithm and Picheny et al. (2010) use a covariance-matrix-adaptation evolution strategy. In this paper we will stick to the discrete search approach, since it is much simpler to implement (we shall present in Section 3.3.4 a method to handle the case of large m) and provides satisfactory results (see Section 3.5).

Finally, remark that the second problem boils down to the computation of a one-dimensional integral with respect to Lebesgue's measure. Indeed, since ξ is a Gaussian process, $\mathsf{Q}_{n,x}$ is a Gaussian probability distribution with mean $\widehat{\xi}_n(x)$ and variance $\sigma_n^2(x)$ as explained in Section 3.2.3. The integral can be computed using a standard Gauss-Hermite quadrature with Q points (see, e.g., Press et al., 1992, Chapter 4) :

$$\int v_{n+1}(y;x,z)\,\mathsf{Q}_{n,x}(\mathrm{d}z) \;\approx\; \frac{1}{\sqrt{\pi}}\sum_{q=1}^{Q} w_q\, v_{n+1}(y;x,\widehat{\xi}_n(x)+\sigma_n(x)u_q\sqrt{2})\,,$$

where $u_1,\ldots,u_Q$ denote the quadrature points and $w_1,\ldots,w_Q$ the corresponding weights. Note that this is equivalent to replacing under P_n the random variable $\xi(x)$ by a quantized random variable with probability distribution $\sum_{q=1}^{Q} w'_q\delta_{z_{n+1,q}(x)}$, where $w'_q = w_q/\sqrt{\pi}$ and $z_{n+1,q}(x) = \widehat{\xi}_n(x)+\sigma_n(x)u_q\sqrt{2}$ denote quadrature points.

Taking all three discretizations into account, the proposed strategy is:

$$X_{n+1} \;=\; \operatorname*{arg\,min}_{1\le k\le m} \sum_{j=1}^{m}\sum_{q=1}^{Q} w'_q\, v_{n+1}\left(Y_j;\, Y_k, z_{n+1,q}(Y_k)\right). \tag{3.26}$$

3.3.4 Implementation

This section gives implementation guidelines for the SUR strategies described in Section 3.3. As said in Section 3.3.3, the strategy (3.26) can, in principle, be translated directly into a computer program. In practice however, we feel that there is still room for different implementations. In particular, it is important to keep the computational complexity of the strategies at a reasonable level. We shall explain in this section some simplifications we have made to achieve this goal.

A straight implementation of (3.26) for the choice of an additional evaluation point is described in Table 3.2. This procedure is meant to be called iteratively in a sequential algorithm, such as that described for instance in Table 3.3. Note that the only parameter to be specified in the SUR strategy (3.26) is Q, which tunes the precision of the approximation of the integral on $\mathbb{R}$ with respect to $\mathsf{Q}_{n,x}$. In our numerical experiments, it was observed that taking $Q = 12$ achieves a good compromise between precision and numerical complexity.

To assess the complexity of a SUR sampling strategy, recall that kriging takes $\mathrm{O}(mn^2)$ operations to predict the value of f at m locations from n evaluation results of f (we suppose that $m > n$ and no approximation is carried out). In the procedure

Table 3.2: Procedure to select a new evaluation point $X_{n+1} \in \mathbb{X}$ using a SUR strategy

---

Require computer representations of

a) a set $\mathcal{I}_n = \{(X_1, f(X_1)), \ldots, (X_n, f(X_n))\}$ of evaluation results;

b) a Gaussian process prior ξ with a (possibly unknown linear parametric) mean function and a covariance function k_θ, with parameter θ;

c) a (pseudo-)random sample $\mathbb{X}_m = \{Y_1, \ldots, Y_m\}$ of size m drawn from the distribution $\mathsf{P}_{\mathbb{X}}$;

d) quadrature points $u_1, \ldots, u_Q$ and corresponding weights $w'_1, \ldots, w'_Q$;

e) a threshold u.

1. compute the kriging approximation $\widehat{f}_n$ and kriging variance σ_n^2 on $\mathbb{X}_m$ from $\mathcal{I}_n$

2. for each candidate point Y_j, $j \in \{1, \ldots, m\}$,

  2.1 for each point Y_k, $k \in \{1, \ldots, m\}$, compute the kriging weights $\lambda_i(Y_k; \{\underline{X}_n, Y_j\})$, $i \in \{1, \ldots, (n+1)\}$, and the kriging variances $\sigma^2(Y_k; \{\underline{X}_n, Y_j\})$

  2.2 compute $z_{n+1,q}(Y_j) = \widehat{f}_n(Y_j) + \sigma_n(Y_j) u_q \sqrt{2}$, for $q = 1, \ldots, Q$

  2.3 for each $z_{n+1,q}(Y_j)$, $q \in \{1, \ldots, Q\}$,

    2.3.1 compute the kriging approximation $\tilde{f}_{n+1,j,q}$ on $\mathbb{X}_m$ from $\mathcal{I}_n \cup (Y_j, f(Y_j) = z_{n+1,q}(Y_j))$, using the weights $\lambda_i(Y_k; \{\underline{X}_n, Y_j\})$, $i = 1, \ldots, (n+1)$, $k = 1, \ldots, m$, obtained at Step 2.1.

    2.3.2 for each $k \in \{1, \ldots, m\}$, compute $v_{n+1}(Y_k;\, Y_j, z_{n+1,q}(Y_j))$, using u, $\tilde{f}_{n+1,j,q}$ obtained in 2.3.1, and $\sigma^2(Y_k; \{\underline{X}_n, Y_j\})$ obtained in 2.1

  2.4 compute $J_n(Y_j) = \sum_{k=1}^{m} \sum_{q=1}^{Q} w'_q\, v_{n+1}(Y_k;\, Y_j, z_{n+1,q}(Y_j))$.

3. find $j^\star = \arg\min_j J_n(Y_j)$ and set $X_{n+1} = Y_{j^\star}$

Table 3.3: Sequential estimation of a probability of failure

---

1. Construct an initial design of size $n_0 < N$ and evaluate f at the points of the initial design.

2. Choose a Gaussian process ξ (in practice, this amounts to choosing a parametric form for the mean of ξ and a parametric covariance function k_θ)

3. Generate a Monte Carlo sample $\mathbb{X}_m = \{Y_1, \ldots, Y_m\}$ of size m from $\mathsf{P}_{\mathbb{X}}$

4. While the evaluation budget N is not exhausted,

- **4.1** optional step: estimate the parameters of the covariance function (case of a plug-in approach);
- **4.2** select a new evaluation point, using past evaluation results, the prior ξ and $\mathbb{X}_m$;
- **4.3** perform the new evaluation.

5. Estimate the probability of failure obtained from the N evaluations of f (for instance, by using $\mathsf{E}_N(\alpha_m) = \frac{1}{m}\sum_j p_N(Y_j)$).

---

to select an evaluation, a first kriging prediction is performed at Step 1 and then, m different predictions have to performed at step 2.1. This cost becomes rapidly burdensome for large values of n and m, and we must further simplify (3.26) to be able to work on applications where m must be large. A natural idea to alleviate the computational cost of the strategy is to avoid dealing with candidate points that have a very low probability of misclassification, since they are probably far from the frontier of the domain of failure. It is also likely that those points with a low probability of misclassification will have a very small contribution in the variance of the error of estimation $\widehat{\alpha}_n - \alpha_m$.

Therefore, the idea is to rewrite the sampling strategy described by (3.26), in such a way that the first summation (over m) and the search set for the minimizer is restricted to a subset of points Y_j corresponding to the m_0 largest values of $\tau_n(Y_j)$. The corresponding algorithm is not described here for the sake of brevity but can easily be adapted from that of Table 3.2. Sections 3.5.2 and 3.5.3 will show that this *pruning* scheme has almost no consequence on the performances of the SUR strategies, even when one considers small values for m_0 (for instance $m_0 = 200$).

3.4 Other strategies proposed in the literature

3.4.1 Estimation of a probability of failure and closely related objectives

Given a real function f defined over $\mathbb{X} \subseteq \mathbb{R}^d$, and a threshold $u \in \mathbb{R}$, consider the following possible goals:

1. estimate a region $\Gamma \subset \mathbb{X}$ of the form $\Gamma = \{x \in \mathbb{X} \mid f(x) > u\}$;
2. estimate the level set $\partial\Gamma = \{x \in \mathbb{X} \mid f(x) = u\}$;
3. estimate f precisely in a neighborhood of $\partial\Gamma$;
4. estimate the probability of failure $\alpha = \mathsf{P}_{\mathbb{X}}(\Gamma)$ for a given probability measure $\mathsf{P}_{\mathbb{X}}$.

These different goals are, in fact, closely related: indeed, they all require, more or less explicitly, to select sampling points in order to get a fine knowledge of the function f in a neighborhood of the level set $\partial\Gamma$ (the location of which is unknown before the first evaluation). Any strategy proposed for one of the first

three objectives is therefore expected to perform reasonably well on the fourth one, which is the topic of this paper.

Several strategies recently introduced in the literature are presented in Sections 3.4.2 and 3.4.3, and will be compared numerically to the SUR strategy in Section 3.5. Each of these strategies has been initially proposed by its authors to address one or several of the above objectives, but they will only be discussed in this paper from the point of view of their performance on the fourth one. Of course, a comparison focused on any other objective would probably be based on different performance metrics, and thus could yield a different performance ranking of the strategies.

3.4.2 The targeted IMSE criterion

The *targeted IMSE* proposed in Picheny et al. (2010) is a modification of the IMSE (Integrated Mean Square Error) sampling criterion (Sacks et al., 1989). While the IMSE sampling criterion computes the average of the kriging variance (over a compact domain $\mathbb{X}$) in order to achieve a space-filling design, the targeted IMSE computes a weighted average of the kriging variance for a better exploration of the regions near the frontier of the domain of failure, as in Oakley (2004). The idea is to put a large weight in regions where the kriging prediction is close to the threshold u, and a small one otherwise. Given $\mathcal{I}_n$, the targeted IMSE sampling criterion, hereafter abbreviated as tIMSE, can be written as

$$J_n^{\mathrm{tIMSE}}(x) = \mathsf{E}_n\left(\int_{\mathbb{X}} \left(\xi - \widehat{\xi}_{n+1}\right)^2 W_n \, \mathrm{d}\mathsf{P}_{\mathbb{X}} \;\middle|\; X_{n+1} = x\right) \tag{3.27}$$

$$= \int_{\mathbb{X}} \sigma^2\left(y; X_1, \ldots, X_n, x\right) \, W_n(y) \, \mathsf{P}_{\mathbb{X}}(\mathrm{d}y), \tag{3.28}$$

where W_n is a weight function based on $\mathcal{I}_n$. The weight function suggested by Picheny et al. (2010) is

$$W_n(x) = \frac{1}{s_n(x)\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\widehat{\xi}_n(x) - u}{s_n(x)}\right)^2\right), \tag{3.29}$$

where $s_n^2(x) = \sigma_\varepsilon^2 + \sigma_n^2(x)$. Note that $W_n(x)$ is large when $\widehat{\xi}_n(x) \approx u$ and $\sigma_n^2(x) \approx 0$, i.e., when the function is known to be close to u.

The tIMSE criterion operates a trade-off between global uncertainty reduction (high kriging variance σ_n^2) and exploration of target regions (high weight function W_n). The weight function depends on a parameter $\sigma_\varepsilon > 0$, which allows to tune the width of the "window of interest" around the threshold. For large values

of σ_ε, J^{tIMSE} behaves approximately like the IMSE sampling criterion. The choice of an appropriate value for σ_ε, when the goal is to estimate a probability of failure, will be discussed on the basis of numerical experiments in Section 3.5.3.

The tIMSE strategy requires a computation of the expectation with respect to $\xi(x)$ in (3.27), which can be done analytically, yielding (3.28). The computation of the integral with respect to $\mathsf{P}_{\mathbb{X}}$ on $\mathbb{X}$ can be carried out with a Monte Carlo approach, as explained in Section 3.3.3. Finally, the optimization of the criterion is replaced by a discrete search in our implementation.

3.4.3 Criteria based on the marginal distributions

Other sampling criteria proposed by Ranjan et al. (2008), Bichon et al. (2008) and Echard et al. (2010a,b) are briefly reviewed in this section[8]. A common feature of these three criteria is that, unlike the SUR and tIMSE criteria discussed so far, they only depend on the *marginal posterior distribution* at the considered candidate point $x \in \mathbb{X}$, which is a Gaussian $\mathcal{N}(\widehat{\xi}_n(x), \sigma_n^2(x))$ distribution. As a consequence, they are of course much cheaper to compute than integral criteria like SUR and tIMSE.

A natural idea, in order to sequentially improve the estimation of the probability of failure, is to visit the point $x \in \mathbb{X}$ where the event $\{\xi(x) \geq u\}$ is the most uncertain. This idea, which has been explored by Echard, Gayton, and Lemaire (2010a,b), corresponds formally to the sampling criterion

$$J_n^{\mathrm{EGL}}(x) \;=\; \tau_n(x) \;=\; 1 - \Phi\left(\frac{|u - \widehat{\xi}_n(x)|}{\sigma_n(x)}\right). \tag{3.30}$$

As in the case of the tIMSE criterion and also, less explicitly, in SUR criteria, a trade-off is realized between global uncertainty reduction (choosing points with a high $\sigma_n^2(x)$) and exploration of the neighborhood of the estimated contour (where $|u - \widehat{\xi}_n(x)|$ is small).

The same leading principle motivates the criteria proposed by Ranjan et al. (2008) and Bichon et al. (2008), which can be seen as special cases of the following sampling criterion:

$$J_n^{\mathrm{RB}}(x) \;:=\; \mathsf{E}_n\left(\max\left(0, \varepsilon(x)^\delta - |u - \xi(x)|^\delta\right)\right), \tag{3.31}$$

where $\varepsilon(x) = \kappa\,\sigma_n(x)$, $\kappa, \delta > 0$. The following proposition provides some insights into this sampling criterion:

[8]Note that the paper of Ranjan et al. (2008) is the only one in this category that does not address the problem of estimating a probability of failure (i.e., Objective 4 of Section 3.4.1).

Proposition 4. *Define* $G_{\kappa,\delta} :]0,1[\to \mathbb{R}_+$ *by*

$$G_{\kappa,\delta}(p) \;:=\; \mathsf{E}\left(\max\left(0, \kappa^\delta - \left|\Phi^{-1}(p) + U\right|\right)\right),$$

where U *is a Gaussian* $\mathcal{N}(0,1)$ *random variable. Let* φ *and* Φ *denote respectively the probability density function and the cumulative distribution function of* U*.*

a) $G_{\kappa,\delta}(p) = G_{\kappa,\delta}(1-p)$ *for all* $p \in]0,1[$*.*

b) $G_{\kappa,\delta}$ *is strictly increasing on* $]0,1/2]$ *and vanishes at* 0*. Therefore,* $G_{\kappa,\delta}$ *is also strictly decreasing on* $[1/2,1[$*, vanishes at* 1*, and has a unique maximum at* $p = 1/2$*.*

c) Criterion (3.31) can be rewritten as

$$J_n^{RB}(x) = \sigma_n(x)^\delta \, G_{\kappa,\delta}(p_n(x)). \tag{3.32}$$

d) $G_{\kappa,1}$ *has the following closed-form expression:*

$$\begin{aligned} G_{\kappa,1}(p) \;=\; & \kappa\left(\Phi(t^+) - \Phi(t^-)\right) \\ & - t\left(2\Phi(t) - \Phi(t^+) - \Phi(t^-)\right) \\ & - \left(2\varphi(t) - \varphi(t^+) - \varphi(t^-)\right), \end{aligned} \tag{3.33}$$

where $t = \Phi^{-1}(1-p)$*,* $t^+ = t + \kappa$ *and* $t^- = t - \kappa$*.*

e) $G_{\kappa,2}$ *has the following closed-form expression:*

$$\begin{aligned} G_{\kappa,2}(p) \;=\; & \left(\kappa^2 - 1 - t^2\right)\left(\Phi(t^+) - \Phi(t^-)\right) \\ & - 2t\left(\varphi(t^+) - \varphi(t^-)\right) \\ & + t^+\varphi(t^+) - t^-\varphi(t^-), \end{aligned} \tag{3.34}$$

with the same notations.

It follows from a) and c) that $J_n^{\mathrm{RB}}(x)$ can also be seen as a function of the kriging variance $\sigma_n^2(x)$ and the probability of misclassification $\tau_n(x) = \min\left(p_n(x), 1 - p_n(x)\right)$. Note that, in the computation of $G_{\kappa,\delta}(p_n(x))$, the quantity denoted by t in (3.33) and (3.34) is equal to $\left(u - \widehat{\xi}_n(x)\right)/\sigma_n(x)$, i.e., equal to the normalized distance between the predicted value and the threshold.

Bichon et al.'s *expected feasibility* function corresponds to (3.32) with $\delta = 1$, and can be computed efficiently using (3.33). Similarly, Ranjan et al.'s *expected*

improvement[9] function corresponds to (3.32) with $\delta = 2$, and can be computed efficiently using (3.34). The proof of Proposition 4 is provided in Appendix B.

Remark 5. *In the case $\delta = 1$, our result coincides with the expression given by Bichon et al. (2008, Eq. (17)). In the case $\delta = 2$, we have found and corrected a mistake in the computations of Ranjan et al. (2008, Eq. (8) and Appendix B).*

3.5 Numerical experiments

3.5.1 A one-dimensional illustration of a SUR strategy

The objective of this section is to show the progress of a SUR strategy in a simple one-dimensional case. We wish to estimate $\alpha = \mathsf{P}_{\mathbb{X}}\{f > 1\}$, where $f : \mathbb{X} = \mathbb{R} \to \mathbb{R}$ is such that $\forall x \in \mathbb{R}$,

$$f(x) = (0.4x - 0.3)^2 + \exp\left(-11.534\,|x|^{1.95}\right) + \exp(-5(x - 0.8)^2)\,,$$

and where $\mathbb{X}$ is endowed with the probability distribution $\mathsf{P}_{\mathbb{X}} = \mathcal{N}(0, \sigma_{\mathbb{X}}^2)$, $\sigma_{\mathbb{X}} = 0.4$, as depicted in Figure 3.2. We know in advance that $\alpha \approx 0.2$. Thus, a Monte Carlo sample of size $m = 1500$ will give a good estimate of α.

In this illustration, ξ is a Gaussian process with constant but unknown mean and a Matérn covariance function, whose parameters are kept *fixed*, for the sake of simplicity. Figure 3.2 shows an initial design of four points and the sampling criterion $J^{\mathrm{SUR}}_{1,n=4}$. Notice that the sampling criterion is only computed at the points of the Monte Carlo sample. Figures 3.3 and 3.4 show the progress of the SUR strategy after a few iterations. Observe that the unknown function f is sampled so that the probability of excursion p_n almost equals zero or one in the region where the density of $\mathsf{P}_{\mathbb{X}}$ is high.

3.5.2 An example in structural reliability

In this section, we evaluate all criteria discussed in Section 3.3 and Section 3.4 through a classical benchmark example in structural reliability (see, e.g., Borri and Speranzini, 1997; Deheeger, 2008; Schueremans, 2001; Waarts, 2000). Echard et al. (2010a,b) used this benchmark to make a comparison among several methods proposed in Schueremans and Van Gemert (2005), some of which are based on the

[9] Despite its name and some similarity between the formulas, this criterion should not be confused with the well-known EI criterion in the field of optimization (Jones et al., 1998; Mockus et al., 1978).

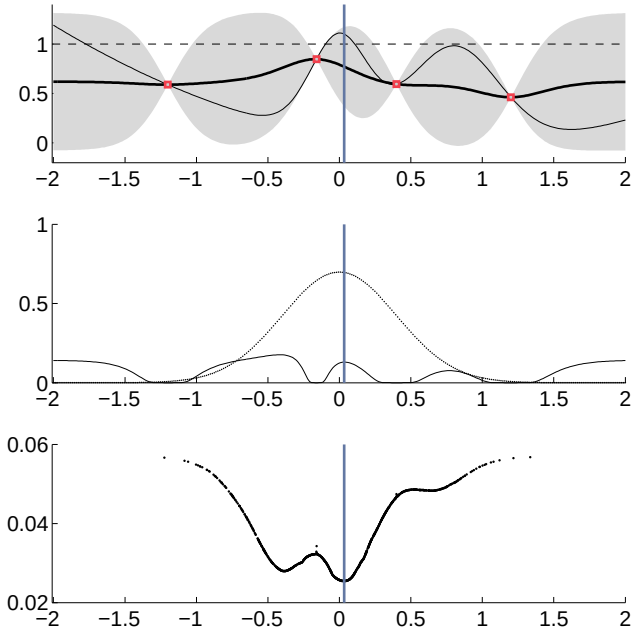


Figure 3.2: Illustration of a SUR strategy. This figure shows the initial design. Top: threshold $u = 1$ (horizontal dashed line); function f (thin line); $n = 4$ initial evaluations (squares); kriging approximation f_n (thick line); 95% confidence intervals computed from the kriging variance (shaded area). Middle: probability of excursion (solid line); probability density of $\mathsf{P}_{\mathbb{X}}$ (dotted line). Bottom: graph of $J^{\mathrm{SUR}}_{1,n=4}(Y_i)$, $i = 1, \ldots, m = 1500$, the minimum of which indicates where the next evaluation of f should be done (i.e., near the origin).

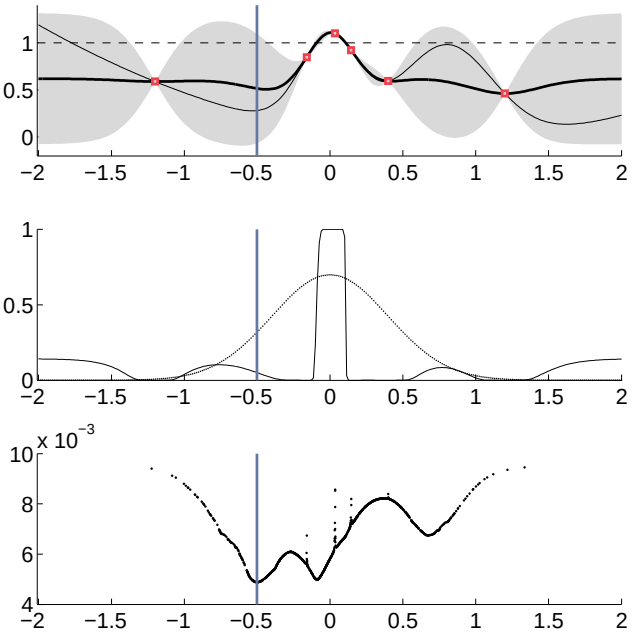


Figure 3.3: Illustration of a SUR strategy (see also Figures 3.2 and 3.4). This figure shows the progress of the SUR strategy after two iterations—a total of $n = 6$ evaluations (squares) have been performed. The next evaluation point will be approximately at $x = -0.5$

.

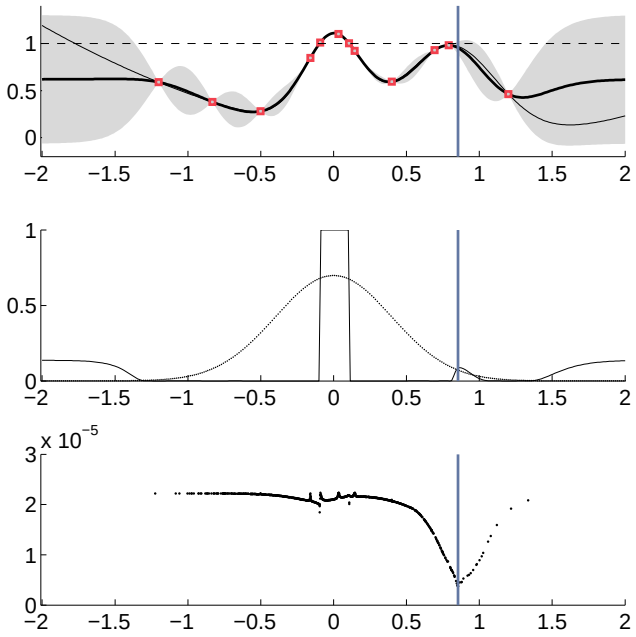


Figure 3.4: Illustration of a SUR strategy (see also Figures 3.2 and 3.3). This figure shows the progress of the SUR strategy after eight iterations—a total of $n = 12$ evaluations (squares) have been performed. At this stage, the probability of excursion p_n almost equals 0 or 1 in the region where the density of $\mathsf{P}_{\mathbb{X}}$ is high.

construction of a response surface. The objective of the benchmark is to estimate the probability of failure of a so-called *four-branch series system*. A failure happens when the system is working under the threshold $u = 0$. The performance function f for this system is defined as

$$f : (x_1, x_2) \in \mathbb{R}^2 \mapsto f(x_1, x_2) = \min \left\{ \begin{array}{l} 3 + 0.1(x_1 - x_2)^2 - (x_1 + x_2)/\sqrt{2}; \\ 3 + 0.1(x_1 - x_2)^2 + (x_1 + x_2)/\sqrt{2}; \\ (x_1 - x_2) + 6/\sqrt{2}; \\ (x_2 - x_1) + 6/\sqrt{2} \end{array} \right\}.$$

The uncertain input factors are supposed to be independent and have standard normal distribution. Figure 3.5 shows the performance function, the failure domain and the input distribution. Observe that f has a first-derivative discontinuity along four straight lines originating from the point $(0, 0)$.

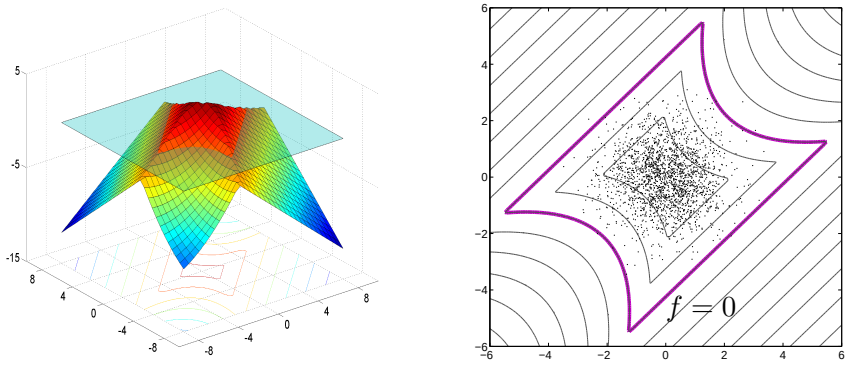


Figure 3.5: Left: mesh plot of the performance function f corresponding to the four-branch series system; a failure happens when f is below the transparent plane; Right: contour plot of f; limit state $f = 0$ (thick line); sample of size $m = 3 \times 10^3$ from $\mathsf{P}_{\mathbb{X}}$ (dots).

For each sequential method, we will follow the procedure described in Table 3.3. We generate an initial design of $n_0 = 10$ points (five times the dimension of the factor space) using a maximin LHS (Latin Hypercube Sampling)[10] on $[-6; 6] \times [-6; 6]$. We choose a Monte Carlo sample of size $m = 30000$. Since the true probability of failure is approximately $\alpha = 0.4\%$ in this example, the coefficient of variation for α_m is $1/\sqrt{m\,\alpha} \approx 9\%$. The same initial design and Monte Carlo sample

[10] More precisely, we use Matlab's `lhsdesign()` function to select the best design according to the maximin criterion among 10^4 randomly generated LHS designs.

are used for all methods.

A Gaussian process with constant unknown mean and a Matérn covariance function is used as our prior information about f. The parameters of the Matérn covariance functions are estimated on the initial design by REML (see, e.g. Stein, 1999). In this experiment, we follow the common practice of re-estimating the parameters of the covariance function during the sequential strategy, but only once every ten iterations to save some computation time.

The probability of failure is estimated by (3.13). To evaluate the rate of convergence, we compute the number n_γ of iterations that must be performed using a given strategy to observe a stabilization of the relative error of estimation within an interval of length 2γ:

$$n_\gamma = \min \left\{ n \geq 0; \forall k \geq n, \frac{|\widehat{\alpha}_{n_0+k} - \alpha_m|}{\alpha_m} < \gamma \right\} .$$

All the available sequential strategies are run 100 times, with different initial designs and Monte Carlo samples. The results for $\gamma = 0.10$, $\gamma = 0.03$ and $\gamma = 0.01$ are summarized in Table 3.4. We shall consider that $n_{0.1}$ provides a measure of the performance of the strategy in the "initial phase", where a rough estimate of α is to be found, whereas $n_{0.03}$ and $n_{0.01}$ measure the performance in the "refinement phase".

The four variants of the SUR strategy (see Table 3.1) have been run with $Q = 12$ and either $m_0 = 10$ or $m_0 = 500$. The performance are similar for all four variants and for both values of m_0. It appears, however, that the criterions J_1^{SUR} and $J_2^{\mathrm{SUR}}2$ (i.e., the criterions given directly by Proposition 3) are slightly better than J_3^{SUR} and J_4^{SUR}; this will be confirmed by the simulations of Section 3.5.3. It also seems that the SUR algorithm is slightly slower to obtain a rough estimate of the probability of failure when m_0 is very small, but performs very well in the refinement phase. (Note that $m_0 = 10$ is a drastic pruning for a sample of size $m = 30000$.)

The tIMSE strategy has been run for three different values of its tuning parameter σ_ε^2, using the pruning scheme with $m_0 = 500$. The best performance is obtained for $\sigma_\varepsilon^2 \approx 0$, and is almost as good as the performance of SUR stragies with the same value of $m0$ (a small loss of performance, of about one evaluation on average, can be noticed in the refinement phase). Note that the required accuracy was not reached after 200 iterations in 17% of the runs for $\sigma_\varepsilon^2 = 1$. In fact, the tIMSE strategy tends to behave like a space-filling strategy in this case. Figure 3.6 shows the points that have been evaluated in three cases: the evaluations are less concentrated on the boundary between the safe and the failure region when $\sigma_\varepsilon^2 = 1$.

Finally, the results obtained for J^{RB} and J^{EGL} indicate that the corresponding strategies are clearly less efficient in the "initial phase" than strategies based on J_1^{SUR} or J_2^{SUR}. For $\gamma = 0.1$, the average loss with respect to J_1^{SUR} is between approximately 0.9 evaluations for the best case (criterion J^{RB} with $\delta = 2$, $\kappa = 2$) and 3.9 evaluations for the worst case. For $\gamma = 0.03$, the loss is between 1.4 evaluations (also for (criterion J^{RB} with $\delta = 2$, $\kappa = 2$) and 3.5 evaluations. This loss of efficiency can also be observed very clearly on the 90^{th} percentile in the inital phase. Criterion J^{RB} seems to perform best with $\delta = 2$ and $\kappa = 2$ in this experiment, but this will not be confirmed by the simulations of Section 3.5.3. Tuning the parameters of this criterion for the estimation of a probability of failure does not seem to be an easy task.

Table 3.4: Comparison of the convergence to α_m in the benchmark example Section 3.5.2 for different sampling strategies. The first number (bold text) is the average value of n_γ over 100 runs. The numbers between brackets indicate the 10^{th} and 90^{th} percentile.

criterion	parameters	$\gamma = 0.10$	$\gamma = 0.03$	$\gamma = 0.01$
J_1^{SUR}	$m_0 = 500$	**16.1** [10–22]	**25.7** [17–35]	**36.0** [26–48]
	$m_0 = 10$	**19.4** [11–28]	**28.1** [19–38]	**35.4** [26–44]
J_2^{SUR}	$m_0 = 500$	**16.4** [10–24]	**25.7** [19–33]	**35.5** [25–45]
	$m_0 = 10$	**20.0** [11–30]	**28.3** [20–39]	**35.3** [26–44]
J_3^{SUR}	$m_0 = 500$	**18.2** [10–27]	**26.9** [18–37]	**35.9** [27–46]
	$m_0 = 10$	**20.1** [11–30]	**28.0** [20–36]	**35.2** [25–44]
J_4^{SUR}	$m_0 = 500$	**17.2** [10–28]	**26.5** [20–36]	**35.2** [25–45]
	$m_0 = 10$	**21.4** [13–30]	**28.9** [20–38]	**35.5** [27–44]
J^{tIMSE}	$\sigma_\varepsilon^2 = 10^{-6}$	**16.6** [10–23]	**26.5** [19–36]	**37.3** [28–49]
	$\sigma_\varepsilon^2 = 0.1$	**15.9** [10–22]	**29.1** [19–43]	**50.5** [30–79]
	$\sigma_\varepsilon^2 = 1$	**21.7** [11–31]	**52.4** [31–85]	**79.5** [42–133](*)
J^{EGL}	–	**21.0** [11–31]	**29.2** [21–39]	**36.4** [28–44]
J^{RB}	$\delta = 1$, $\kappa = 0.5$	**18.7** [10–27]	**27.5** [20–35]	**36.6** [27–44]
	$\delta = 1$, $\kappa = 2.0$	**18.9** [11–28]	**28.3** [21–35]	**37.7** [30–45]
	$\delta = 2$, $\kappa = 0.5$	**17.6** [10–24]	**27.6** [20–34]	**37.1** [29–45]
	$\delta = 2$, $\kappa = 2.0$	**17.0** [10–21]	**27.1** [20–34]	**36.8** [29–44]

(*) The required accuracy was not reached after 200 iterations in 17% of the runs

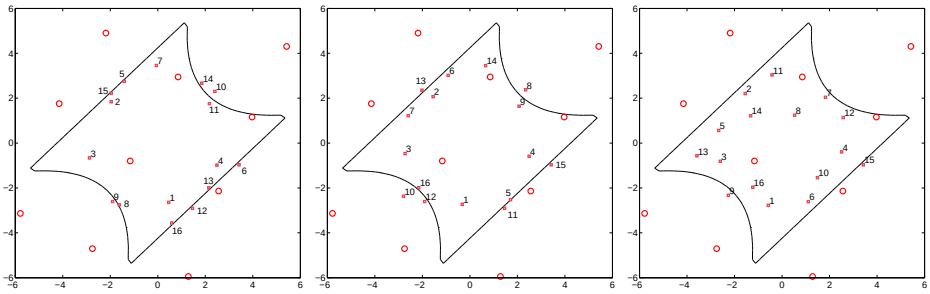


Figure 3.6: The first 16 points (squares) evaluated using sampling criterion J_1^{SUR} (left), J^{tIMSE} with $\sigma_\varepsilon^2 = 0.1$ (middle), J^{tIMSE} with $\sigma_\varepsilon^2 = 1$ (right). Numbers near squares indicate the order of evaluation. The location of the $n_0 = 10$ points of the initial design are indicated by circles.

Table 3.5: Size of the initial design and covariance parameters for the experiments of Section 3.5.3. The parametrization of the Matérn covariance function used here is defined in Appendix A.

d	n_0	σ^2	ν	ρ
1	3	1.0	2.0	0.100
2	10	1.0	2.0	0.252
3	15	1.0	2.0	0.363

3.5.3 Average performance on sample paths of a Gaussian process

This section provides a comparison of all the criteria introduced or recalled in this paper, on the basis of their average performance on the sample paths of a zero-mean Gaussian process defined on $\mathbb{X} = [0,1]^d$, for $d \in \{1,2,3\}$. In all experiments, the same covariance function is used for the generation of the sample paths and for the computation of the sampling criteria. We have considered isotropic Matérn covariance functions, whose parameters are given in Table 3.5. An initial maximin LHS design of size n_0 (also given in the table) is used: note that the value of n reported on the x-axis of Figures 3.7–3.11 is the total number of evaluations, including the initial design.

The d input variables are assumed to be independent and uniformly distributed on $[0,1]$, i.e., $\mathsf{P}_{\mathbb{X}}$ is the uniform distribution on $\mathbb{X}$. An m-sample $Y_1, \ldots, Y_m$ from $\mathsf{P}_{\mathbb{X}}$ is drawn one and for all, and used both for the approximation of integrals (in SUR and tIMSE criteria) and for the discrete search of the next sampling point (for all criteria). We take $m = 500$ and use the same MC sample for all criteria in

a given dimension d.

We adopt the meta-estimation framework as described in Section 3.3.3; in other words, our goal is to estimate the MC estimator α_m. We choose to adjust the threshold u in order to have $\alpha_m = 0.02$ for all sample paths (note that, as a consequence, there are exactly $m\alpha_m = 10$ points in the failure region) and we measure the performance of a strategy after n evaluations by its relative mean-square error (MSE) expressed in decibels (dB):

$$\text{rMSE} := 10 \log_{10} \left(\frac{1}{L} \sum_{l=1}^{L} \frac{\left(\widehat{\alpha}_{m,n}^{(l)} - \alpha_m \right)^2}{\alpha_m^2} \right),$$

where $\widehat{\alpha}_{m,n}^{(l)} = \frac{1}{m} \sum_{j=1}^{m} p_n^{(l)}(Y_j)$ is the posterior mean of the MC estimator α_m after n evaluations on the l^{th} simulated sample path ($L = 4000$).

We use a sequential maximin strategy as a reference in all of our experiments. This simple space-filling strategy is defined by $X_{n+1} = \arg\max_j \min_{1 \leq i \leq n} |Y_j - X_i|$, where the argmax runs over all indices j such that $Y_j \notin \{X_1, \ldots, X_n\}$. Note that this strategy does not depend on the choice of a Gaussian process model.

Our first experiment (Figure 3.7) provides a comparison of the four SUR strategies proposed in Section 3.3.2. It appears that all of them perform roughly the same when compared to the reference strategy. A closer look, however, reveals that the strategies J_1^{SUR} and J_2^{SUR} provided by Proposition 3 perform slightly better than the other two (noticeably so in the case $d = 3$).

The performance of the tIMSE strategy is shown on Figure 3.8 for several value of its tuning parameter σ_ε^2 (other values, not shown here, have been tried as well). It is clear that the performance of this strategy improves when σ_ε^2 goes to zero, whatever the dimension.

The performance of the strategy based on $J_{\kappa,\delta}^{\text{RB}}$ is shown on Figure 3.9 for several values of its parameters. It appears that the criterion proposed by Bichon et al. (2008), which corresponds to $\delta = 1$, performs better than the one proposed by Ranjan et al. (2008), which corresponds to $\delta = 2$, for the same value of κ. Moreover, the value $\kappa = 0.5$ has been found in our experiments to produce the best results.

Figure 3.10 illustrates that the loss of performance associated to the "pruning trick" introduced in Section 3.3.4 can be negligible if the size m_0 of the pruned MC sample is large enough (here, m_0 has been taken equal to 50). In practice, the value of m_0 should be chosen small enough to keep the overhead of the sequential strategy reasonable—in other words, large values of m_0 should only be used for very

complex computer codes.

Finally, a comparison involving the best strategy obtained in each category is presented on Figure 3.11. The best result is consistently obtained with the SUR strategy based on $J_{1,n}^{\mathrm{SUR}}$. The tIMSE strategy with $\sigma_\varepsilon^2 \approx 0$ provides results which are almost as good. Note that both strategies are one-step lookahead strategies based on the approximation of the risk by an integral criterion, which makes them rather expensive to compute. Simpler strategies based on the marginal distribution (criteria J_n^{RB} and J_n^{EGL}) provide interesting alternatives for moderately expensive computer codes: their performances, although not as good as those of one-step lookahead criterions, are still much better than that of the reference space-filling strategy.

3.6 Concluding remarks

One of the main objectives of this paper was to present a synthetic viewpoint on sequential strategies based on a Gaussian process model and kriging for the estimation of a probability of failure. The starting point of this presentation is a Bayesian decision-theoretic framework from which the theoretical form of an optimal strategy for the estimation of a probability of failure can be derived. Unfortunately, the dynamic programming problem corresponding to this strategy is not numerically tractable. It is nonetheless possible to derive from there the ingredients of a sub-optimal strategy: the idea is to focus on one-step lookahead suboptimal strategies, where the exact risk is replaced by a substitute risk that accounts for the information gain about α expected from a new evaluation. We call such a strategy a *stepwise uncertainty reduction* (SUR) strategy. Our numerical experiments show that SUR strategies perform better, on average, than the other strategies proposed in the literature. However, this comes at a higher computational cost than strategies based only on marginal distributions. The tIMSE sampling criterion, which seems to have a convergence rate comparable to that of the SUR criterions when $\sigma_\varepsilon^2 \approx 0$, also has a high computational complexity.

In which situations can we say that the sequential strategies presented in this paper are interesting alternatives to classical importance sampling methods for estimating a probability of failure, for instance the subset sampling method of Au and Beck (2001)? In our opinion, beyond the obvious role of the simulation budget N, the answer to this question depends on our capacity to elicit an appropriate prior. In the example of Section 3.5.2, as well as in many other examples of the litera-

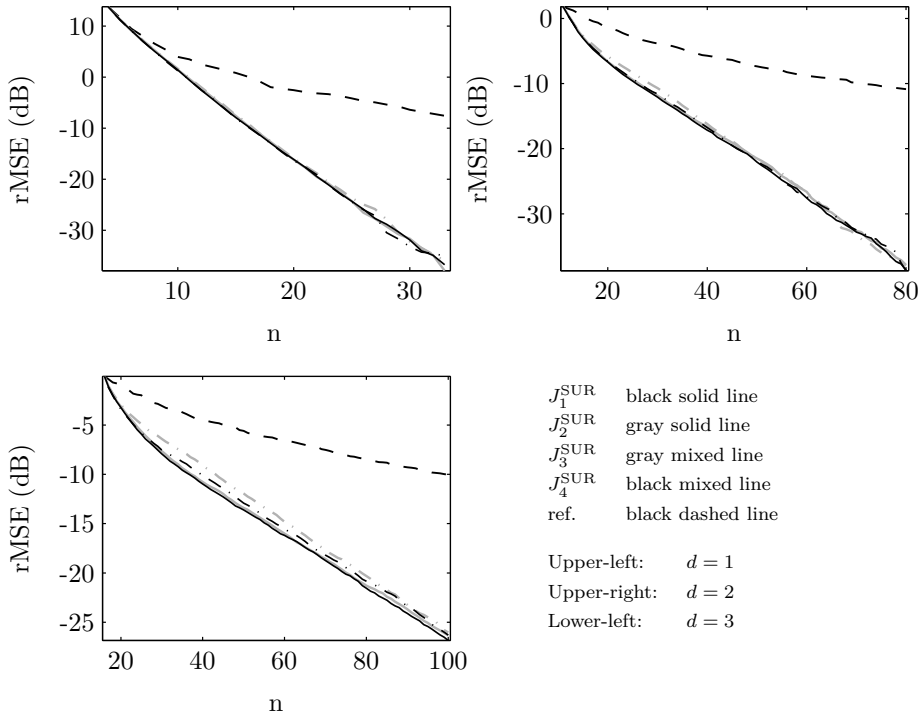


Figure 3.7: Relative MSE performance of several SUR strategies.

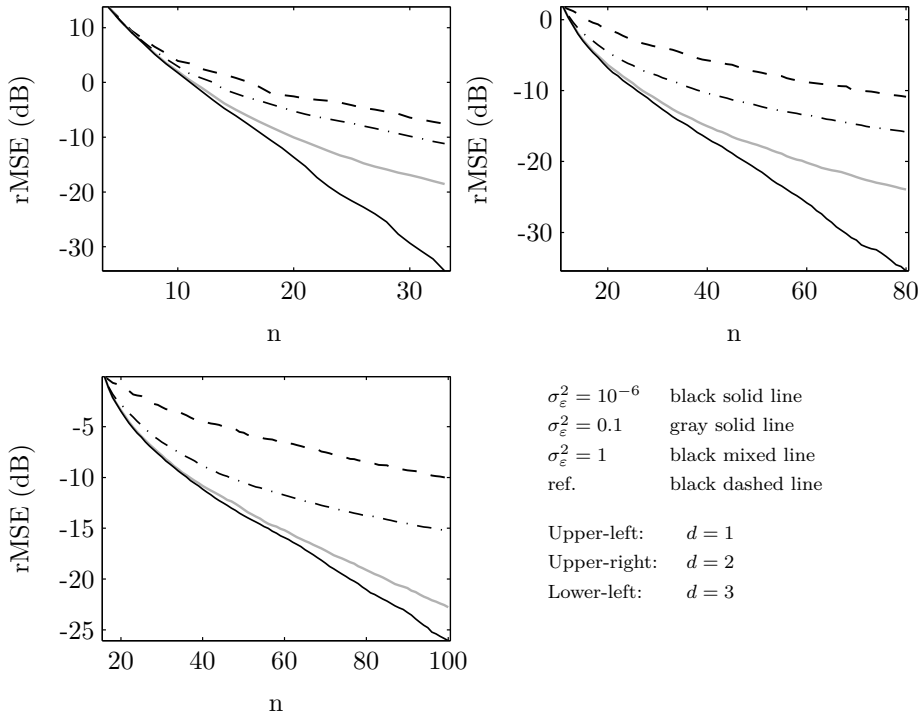


Figure 3.8: Relative MSE performance of the tIMSE strategy for several values of its parameter.

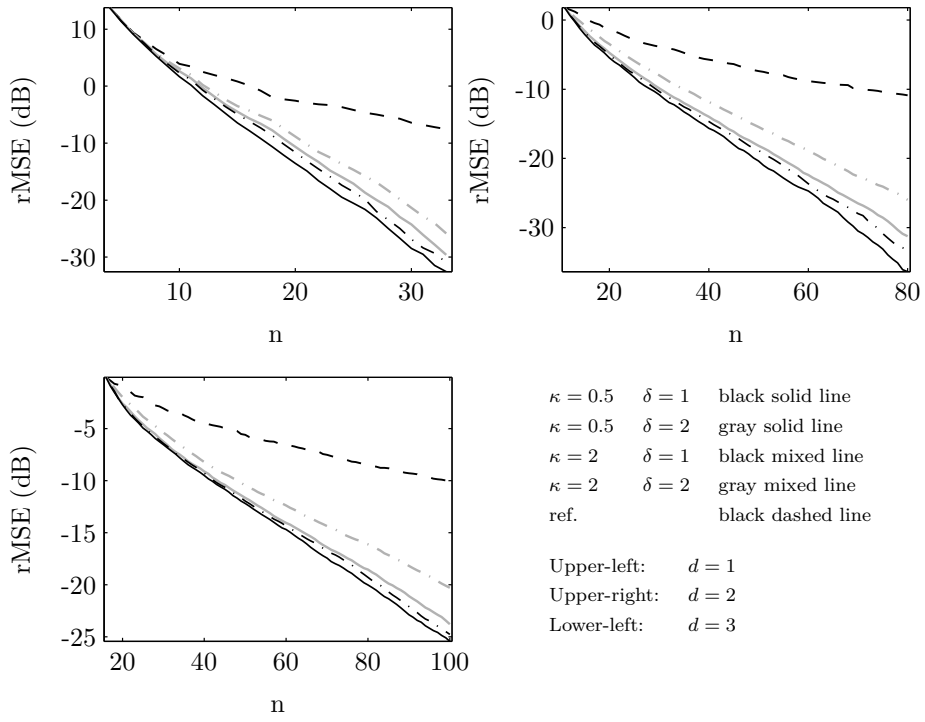


Figure 3.9: Relative MSE performance of the J^{RB} criterion, for several values of its parameters.

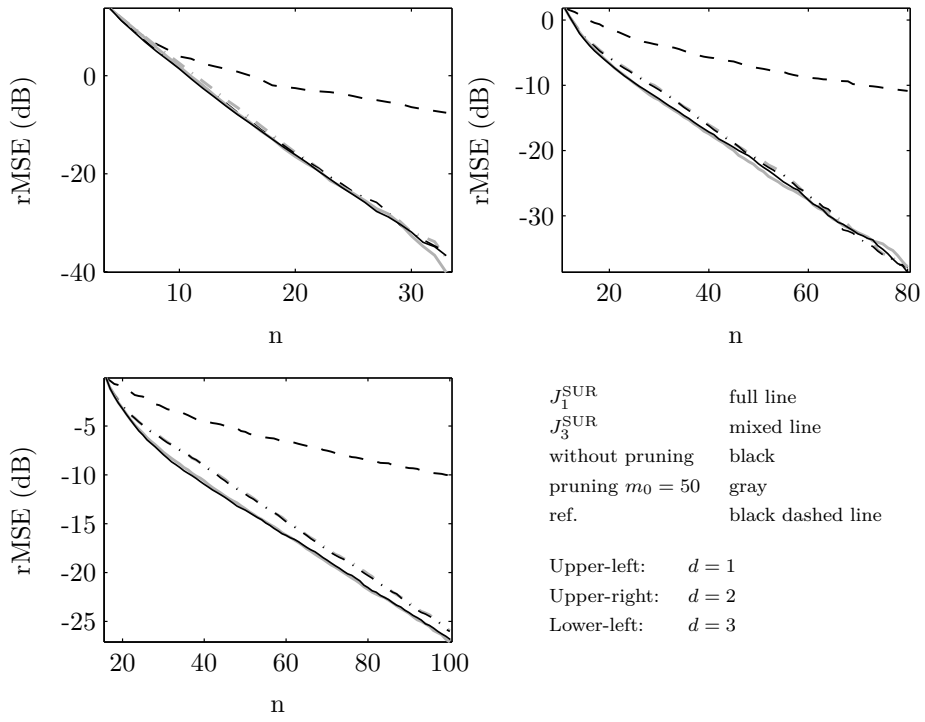


Figure 3.10: Relative MSE performance of two SUR criteria, with and without the "pruning trick" described in Section 3.3.4. The black and gray lines are almost surimposed for each of the criterions J_1^{SUR} and J_3^{SUR}.

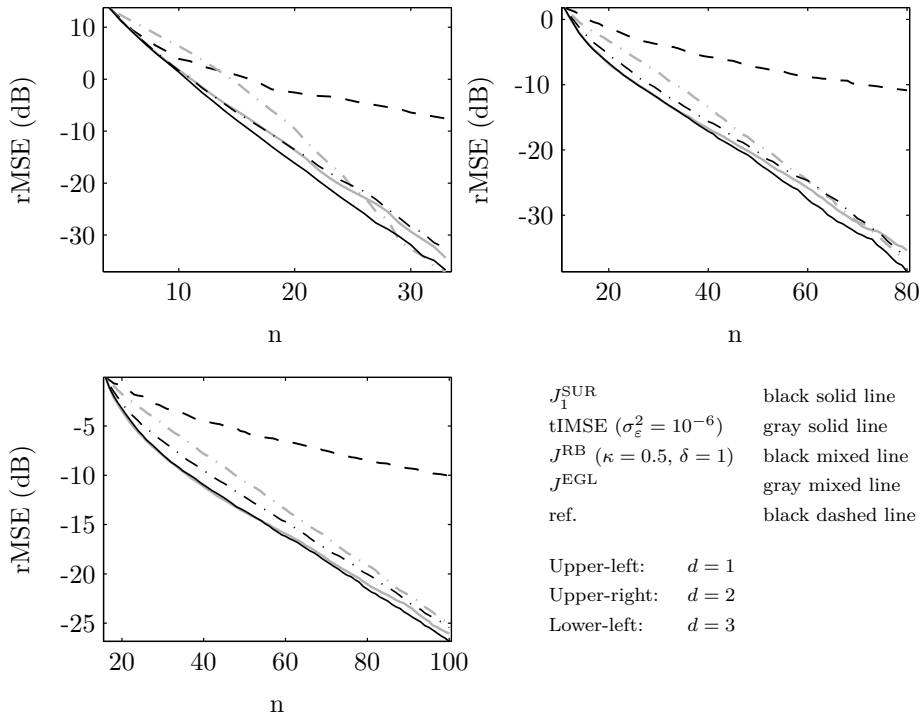


Figure 3.11: Relative MSE performance the best strategy in each category.

ture using Gaussian processes in the domain of computer experiments, the prior is easy to choose because $\mathbb{X}$ is a low-dimensional space and f tends to be smooth. Then, the plug-in approach which consists in using ML or REML to estimate the parameters of the covariance function of the Gaussian process after each new evaluation is likely to succeed. If $\mathbb{X}$ is high-dimensional and f is expensive to evaluate, difficulties arise. In particular, our sampling strategies do not take into account our uncertain knowledge of the covariance parameters, and there is no guarantee that ML estimation will do well when the points are chosen by a sampling strategy that favors some localized target region (the neighboorhood the frontier of the domain of failure in this paper, but the question is equally relevant in the field optimization, for instance). The difficult problem of deciding the size n_0 of the initial design is crucial in this connection. Fully Bayes procedures constitute a possible direction for future research, as long as they don't introduce an unacceptable computational overhead. Whatever the route, we feel that the robustness of Gaussian process-based sampling strategies with respect to the procedure of estimation of the covariance parameters should be addressed carefully in order to make these methods usable in the industrial world.

Software. We would like to draw the reader's attention on the recently published package KrigInv (Picheny and Ginsbourger, 2011) for the statistical computing environment R (see Hornik, 2010). This package provides an open source (GPLv3) implementation of all the strategies proposed in this paper. Please note that the simulation results presented in this paper were not obtained using this package, that was not available at the time of its writing.

Appendix

A. The Matérn covariance

The exponential covariance and the Matérn covariance are among the most conventionally used stationary covariances in the literature of design and analysis of computer experiments. The Matérn covariance class (Yaglom, 1986) offers the possibility to adjust the regularity of ξ with a single parameter. Stein (1999) advocates the use of the following parametrization of the Matérn function:

$$\kappa_\nu(h) = \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(2\nu^{1/2}h\right)^\nu \mathcal{K}_\nu \left(2\nu^{1/2}h\right), \quad h \in \mathbb{R} \tag{3.35}$$

where Γ is the Gamma function and $\mathcal{K}_\nu$ is the modified Bessel function of the second kind. The parameter $\nu > 0$ controls regularity at the origin of the function. To model a real-valued function f defined over $\mathbb{X} \subset \mathbb{R}^d$, with $d \geq 1$, we use the following anisotropic form of the Matérn covariance:

$$k_\theta(x, y) = \sigma^2 \kappa_\nu \left(\sqrt{\sum_{i=1}^{d} \frac{(x_{[i]} - y_{[i]})^2}{\rho_i^2}} \right), \quad x, y \in \mathbb{R}^d \tag{3.36}$$

where $x_{[i]}, y_{[i]}$ denote the i^{th} coordinate of x and y, the positive scalar σ^2 is a variance parameter (we have $k_\theta(x, x) = \sigma^2$), and the positive scalars ρ_i represent scale or *range* parameters of the covariance, *i.e.*, characteristic correlation lengths. Since $\sigma^2 > 0, \nu > 0, \rho_i > 0$, $i = 1, \ldots, d$, we can take the logarithm of these scalars, and consider the vector of parameters $\theta = \{\log \sigma^2, \log \nu, -\log \rho_1, \ldots, -\log \rho_d\} \in \mathbb{R}^{d+2}$, which is a practical parameterization when σ^2, ν, ρ_i, $i = 1, \ldots, d$, need to be estimated from data.

B. Proof of Proposition 4

a) Using the identity $\Phi^{-1}(1-p) = -\Phi^{-1}(p)$, we get

$$\left|U + \Phi^{-1}(1-p)\right| \;=\; \left|U - \Phi^{-1}(p)\right| \;\stackrel{\mathrm{d}}{=}\; \left|U + \Phi^{-1}(p)\right|,$$

where $\stackrel{\mathrm{d}}{=}$ denotes an equality in distribution. Therefore $G_{\kappa,\delta}(1-p) = G_{\kappa,\delta}(p)$.

b) Let $S_p = \max\left(0, \kappa^\delta - \left|\Phi^{-1}(p) + U\right|\right)$. Straightforward computations show that $t \mapsto \mathsf{P}\left(|t + U| \leq v\right)$ is strictly decreasing to 0 on $[0, +\infty[$, for all $v > 0$. As a consequence, $p \mapsto \mathsf{P}\left(S_p < s\right)$ is strictly increasing to 1 on $[1/2, 1[$, for all $s \in \left]0, \kappa^\delta\right[$.

Therefore, $G_{\kappa,\delta}$ is strictly decreasing on $[1/2, 1[$ and tends to zeros when $p \to 1$. The other assertions then follow from a).

c) Recall that $\xi(x) \sim \mathcal{N}(\widehat{\xi}_n(x), \sigma_n^2(x))$ under P_n. Therefore $U := (\xi(x) - \widehat{\xi}_n(x))/\sigma_n(x) \sim \mathcal{N}(0,1)$ under P_n, and the result follows by substitution in (3.31).

The closed-form expressions of Ranjan et al.'s and Bichon and al.'s criteria (assertions d) and e)) is established in the following sections.

B.1 A preliminary decomposition common to both criteria

Recall that $t = \Phi^{-1}(1-p)$, $t^+ = t + \kappa$ and $t^- = t - \kappa$. Then,

$$\begin{aligned} G_{\kappa,\delta}(p) &= G_{\kappa,\delta}(1-p) = \mathsf{E}\left(\max\left(0, \kappa^\delta - |t-U|^\delta\right)\right) \\ &= \int_{\kappa^\delta - |t-u|^\delta \geq 0} \left(\kappa^\delta - |t-u|^\delta\right) \varphi(u)\,\mathrm{d}u \\ &= \int_{t^-}^{t^+} \left(\kappa^\delta - |t-u|^\delta\right) \varphi(u)\,\mathrm{d}u \\ &= \kappa^\delta \left(\Phi(t^+) - \Phi(t^-)\right) - \underbrace{\int_{t^-}^{t^+} |t-u|^\delta \varphi(u)\,\mathrm{d}u}_{\text{Term } A}. \end{aligned} \tag{3.37}$$

The computation of the integral A will be carried separately in the next two sections for $\delta = 1$ and $\delta = 2$. For this purpose, we shall need the following elementary results:

$$\int_a^b u\varphi(u)\mathrm{d}u = \varphi(a) - \varphi(b)\,, \tag{3.38}$$

$$\int_a^b u^2\varphi(u)\mathrm{d}u = a\varphi(a) - b\varphi(b) + \Phi(b) - \Phi(a)\,. \tag{3.39}$$

B.2 Case $\delta = 1$

Let us compute the value A_1 of the integral A for $\delta = 1$:

$$\begin{aligned} A_1 &= \int_{t^-}^{t^+} |t-u|\,\varphi(u)\mathrm{d}u = \int_{t^-}^{t} (t-u)\varphi(u)\mathrm{d}u + \int_{t}^{t^+} (u-t)\varphi(u)\mathrm{d}u \\ &= t\left(\int_{t^-}^{t} \varphi(u)\,\mathrm{d}u - \int_{t}^{t^+} \varphi(u)\,\mathrm{d}u\right) - \int_{t^-}^{t} u\varphi(u)\,\mathrm{d}u + \int_{t}^{t^+} u\varphi(u)\,\mathrm{d}u \\ &= t\left(2\Phi(t) - \Phi(t^-) - \Phi(t^+)\right) + 2\varphi(t) - \varphi(t^-) - \varphi(t^+)\,, \end{aligned} \tag{3.40}$$

where (3.38) has been used to get the final result. Plugging (3.40) into (3.37) yields (3.33).

B.3 Case $\delta = 2$

Let us compute the value A_2 of the integral A for $\delta = 2$:

$$\begin{aligned} A_2 &= \int_{t^-}^{t^+} (t-u)^2 \varphi(u)\,\mathrm{d}u \\ &= t^2 \int_{t^-}^{t^+} \varphi(u)\,\mathrm{d}u - 2t \int_{t^-}^{t^+} u\varphi(u)\,\mathrm{d}u + \int_{t^-}^{t^+} u^2 \varphi(u)\,\mathrm{d}u \\ &= t^2 \Big(\Phi(t^+) - \Phi(t^-)\Big) \;-\; 2t \Big(\varphi(t^-) - \varphi(t^+)\Big) \\ &\quad + \; t^- \varphi(t^-) - t^+ \varphi(t^+) \;+\; \Phi(t^+) - \Phi(t^-)\,, \end{aligned} \tag{3.41}$$

where (3.38) and (3.39) have been used to get the final result. Plugging (3.40) into (3.37) yields (3.34).

Chapter 4

Bayesian Subset Simulation: a kriging-based subset simulation algorithm for the estimation of small probabilities of failure

---

By L. Li, J. Bect and E. Vazquez

Extended version of a paper to appear in
the Proceedings of PSAM 11 & ESREL 2012,
25-29 June 2012, Helsinki, Finland.

Contents

Abstract

The estimation of small probabilities of failure from computer simulations is a classical problem in engineering, and the Subset Simulation algorithm proposed by Au & Beck (Prob. Eng. Mech., 2001) has become one of the most popular method to solve it. Subset simulation has been shown to provide significant savings in the number of simulations to achieve a given accuracy of estimation, with respect to many other Monte Carlo approaches. The number of simulations remains still quite high however, and this method can be impractical for applications where an expensive-to-evaluate computer model is involved.

We propose a new algorithm, called Bayesian Subset Simulation, that takes the best from the Subset Simulation algorithm and from sequential Bayesian methods based on kriging (also known as Gaussian process modeling). The performance of this new algorithm is illustrated using three test cases from the literature. We are able to report promising results. In addition, we provide a numerical study of the statistical properties of the estimator.

Keywords: Computer experiments, Sequential design, Subset Simulation, Probability of failure, Stepwise Uncertainty Reduction

4.1 Introduction

In this paper[1], we propose an algorithm called *Bayesian Subset Simulation* (BSS), that combines the Bayesian decision-theoretic framework from our previous studies (see Vazquez and Bect, 2009, and Chapter 3 of this thesis) with the Subset Simulation algorithm (Au and Beck, 2001).

Let $\Gamma = \{x \in \mathbb{X} : f(x) > u\}$ denote the excursion set of a function $f : \mathbb{X} \to \mathbb{R}$ above a threshold $u \in \mathbb{R}$, where $\mathbb{X} \subseteq \mathbb{R}^d$ denotes the input space. $\mathsf{P}_{\mathbb{X}}$ stands for the uncertainty on the input vector $x \in \mathbb{X}$. We are interested in estimating the probability $\alpha(u) := \mathsf{P}_{\mathbb{X}}(\Gamma)$, which corresponds to the probability of failure of a system for which f is a cost function (see Chapter 3). If the probability $\alpha(u)$ is small, estimating it using the Monte Carlo estimator $\widehat{\alpha}_m^{\mathrm{MC}} = 1/m \sum_{i=1}^m \mathbf{1}_{f(X_i)>u}$, $X_i \overset{\text{i.i.d}}{\sim} \mathsf{P}_{\mathbb{X}}$, requires a large number of evaluations of f. If the cost function f is expensive to evaluate, this leads to use a large amount of computational resources, and in

[1] Ling Li, Julien Bect, Emmanuel Vazquez (2012) Bayesian Subset Simulation : a kriging-based subset simulation algorithm for the estimation of small probabilities of failure. In: Proceedings of PSAM 11 & ESREL 2012, 25-29 June 2012, Helsinki, Finland

some cases, it may be even impossible to proceed in reasonable time. Estimating small probabilities of failure with moderate computational resources is a challenging problem.

When $\alpha(u)$ is small, the main problem with the estimator $\widehat{\alpha}_m^{\mathrm{MC}}$ is that the sample size m must be large in order to get a reasonably high probability of observing at least a few samples in Γ. In the literature, importance sampling methods have been considered to generate more samples in the failure region Γ. However, the success of this kind of methods relies greatly on prior knowledge about the failure region Γ and on a relevant choice for the proposal sampling distribution.

The idea of Subset Simulation is to decompose the difficult problem of simulating a sample into a series of easier problems, by introducing intermediate failure events. Let $u_0 = -\infty < u_1 < u_2 < \ldots < u_T = u$ be a sequence of increasing thresholds and define a corresponding sequence of decreasing excursion sets $\Gamma_0 := \mathbb{X} \supseteq \Gamma_1 \supseteq \cdots \supseteq \Gamma_T := \Gamma$, where $\Gamma_t := \{x \in \mathbb{X} : f(x) > u_t\}$, $t = 1, \ldots, T$. Notice that $\Gamma_t = \bigcap_{i=1}^{t} \Gamma_i$. Then, using the properties

$$\begin{cases} \alpha(u_0) \;=\; 1\,, \\ \alpha(u_{t+1}) \;=\; \alpha(u_t)\, \mathsf{P}_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t)\,, \quad t \geq 0\,, \end{cases} \tag{4.1}$$

$\alpha(u)$ can be rewritten as a product of conditional probabilities:

$$\alpha(u) \;=\; \mathsf{P}_{\mathbb{X}}(\Gamma_T) \;=\; \prod_{t=0}^{T-1} \mathsf{P}_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t)\,. \tag{4.2}$$

Thus, the idea of Subset Simulation is to replace the problem of estimating the small probability $\alpha(u)$ by that of estimating the conditional probabilities $\mathsf{P}_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t)$, $0 \leq t < T$, which are larger and therefore easier to estimate.

In Au and Beck (2001), a standard Monte Carlo simulation method is used to estimate $\mathsf{P}_{\mathbb{X}}(\Gamma_1) = \mathsf{P}_{\mathbb{X}}(\Gamma_1|\Gamma_0)$. For the other conditional probabilities, a Markov Chain Monte Carlo method is used to simulate samples in Γ_t according to $\mathsf{P}_{\mathbb{X}}(\cdot|\Gamma_t)$, and then $\mathsf{P}_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t)$ is estimated using a Monte Carlo method. Due to the direct use of a Monte Carlo approximation at each stage, the number of evaluations needed still remains quite high. For many practical applications where the performance function corresponds to an expensive-to-evaluate computer model, this is not applicable. Note that the Subset Simulation algorithm has recently caught the attention of the "rare event" community: using standard tools from the sequential Monte Carlo (SMC) literature, Cérou et al. (2011) derived several theoretical results about some algorithms which are very similar to the Subset Sampling algorithm.

In this work, we propose an algorithm that takes advantage of a Gaussian process prior about f in order to decrease the number of evaluations needed to estimate the conditional probabilities $\mathsf{P}_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t)$. The Gaussian process model makes it possible to assess the uncertainty about the position of the intermediate excursion sets Γ_t, given a set of past evaluation results. The idea has its roots in the field of design and analysis of computer experiments (see, e.g., Santner et al., 2003a, and references therein). More specifically, kriging-based sequential strategies for the estimation of a probability of failure (see Chapter 3 for a review of such strategies) are closely related to the field of Bayesian global optimization (see, e.g., Ginsbourger, 2009; Villemonteix et al., 2009, and the references therein).

The paper is organized as follows. In Section 4.2, we review two recent techniques for estimating a small probability of failure using importance sampling or Subset Simulation in conjunction with non-parametric meta-models. In Section 4.3, we give a detailed presentation of our new Bayesian Subset Simulation algorithm. In Section 4.4, we apply the algorithm on three examples from the literature, and we perform numerical simulations to investigate the performance of the proposed algorithm. A comparison with Subset Simulation and the ^{2}SMART algorithm of Deheeger and Lemaire (2007) is provided. Finally, we conclude in Section 4.5.

4.2 Literature review

In this section, we will review two kinds of hybrid techniques for probabilities of failure estimation that involve importance sampling or Subset Simulation and the use of meta-models.

4.2.1 Two-stage surrogate model based Importance Sampling (IS)

An algorithm involving kriging-based adaptive sampling and Importance Sampling (IS) has been recently proposed by V. Dubourg and co-authors (Dubourg et al., 2011a) to address the problem of surrogate model based IS strategy for rare events probability of failure estimation. Auffray et al. (2011) also proposed a similar two stage procedure, which uses a different importance density function.

Their methods consist in two stages: 1) build a kriging surrogate of the expensive-to-evaluate cost function, with a number of evaluations N_1; 2) use the probabilistic prediction provided by the kriging model to propose an importance sampling density, according to which draw an i.i.d. sample with sample size N_2.

The final probability of failure is then computed as the product of two terms, one is the augmented failure probability (a rough estimator) estimated from the surrogate model, and a correction factor (a refined term near the failure region) computed from the original performance function.

The probability of failure can be written as:

$$\alpha = \mathsf{P}_{\mathbb{X}}(\Gamma) = \int_{\mathbb{X}} \mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}(x)\mathrm{d}x, \tag{4.3}$$

where $p_{\mathbb{X}}$ is the input probability density function, and $\mathbf{1}_{x\in\Gamma}$ is the indicator function taking value one if $x \in \Gamma$ and zero otherwise. As discussed in section 2.3.3, the optimal proposal density function for the estimation of α by importance sampling is:

$$q^*(x) = \frac{\mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}}{\int_{\mathbb{X}} \mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}\mathrm{d}x}. \tag{4.4}$$

However, in practice, this optimal importance distribution is not implementable due to its requirement of the knowledge of the true probability of failure α. There are several ways to approximate the optimal proposal density function (4.4) in literature though. We can write the proposal probability density function as:

$$q(x) = \frac{g(x)p_{\mathbb{X}}}{\int_{\mathbb{X}} g(x)p_{\mathbb{X}}\mathrm{d}x} \propto g(x)p_{\mathbb{X}}, \tag{4.5}$$

with

$$g(x) = \begin{cases} \mathbf{1}_{x\in\widehat{\Gamma}_1} & \text{in Auffray et al. (2011),} \\ \mathsf{P}_{N_1}(\xi(x) > u) & \text{in Dubourg et al. (2011a),} \end{cases} \tag{4.6}$$

where $\widehat{\Gamma}_1$ is defined as

$$\widehat{\Gamma}_1 = \{x \in \mathbb{X} : \widehat{\xi}_{N_1}(x) > u - \kappa\sigma_{N_1}(x)\}, \tag{4.7}$$

and κ is a constant such that $\Gamma \subset \widehat{\Gamma}_1$ has a good confidence level. In other word, we want all the failure points to be located in the approximated surrogate excursion set.

Using the basic formula of importance sampling, (4.3) can be rewritten as:

$$\begin{aligned} \alpha &= \int_{\mathbb{X}} \mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}\mathrm{d}x \\ &= \int_{\mathbb{X}} \frac{\mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}}{q(x)} q(x)\mathrm{d}x \\ &= \int_{\mathbb{X}} \frac{\mathbf{1}_{x\in\Gamma} p_{\mathbb{X}}}{g(x)p_{\mathbb{X}}} q(x)\mathrm{d}x \int_{\mathbb{X}} g(x)p_{\mathbb{X}}\mathrm{d}x \\ &= \underbrace{\int_{\mathbb{X}} g(x)p_{\mathbb{X}}\mathrm{d}x}_{\alpha_1} \underbrace{\int_{\mathbb{X}} \frac{\mathbf{1}_{x\in\Gamma}}{g(x)} q(x)\mathrm{d}x}_{\alpha_2}, \end{aligned} \tag{4.8}$$

where α_1 denotes the first stage estimator, also called the "rough estimator" in Dubourg et al. (2011a), and α_2 denotes the second stage estimator, or the correction factor.

The first stage estimator α_1 can be approximated by a Monte Carlo estimator:

$$\widehat{\alpha}_1 = \frac{1}{m}\sum_{i=1}^{m} g(X_1^i), \tag{4.9}$$

where $X_1^1, \ldots, X_1^m$ are i.i.d. samples drawn according to $p_{\mathbb{X}}$.

Similarly, we can approximate α_2 with a Monte Carlo estimator:

$$\widehat{\alpha}_2 = \frac{1}{N_2}\sum_{i=1}^{N_2} \frac{\mathbf{1}_{X_2^i \in \Gamma}}{g(X_2^i)}, \tag{4.10}$$

where $X_2^1, \ldots, X_2^m$ are samples according to $g(x)p_{\mathbb{X}} \propto q(x)$. In practice, Auffray et al. (2011) used an accept-reject algorithm that randomly generates samples according to $p_{\mathbb{X}}$, and keep the points located in $\widehat{\Gamma}_1$, however, this is very inefficient when $\mathsf{P}_{\mathbb{X}}(\widehat{\Gamma}_1)$ is small; in Dubourg et al. (2011a), it proposed to use a Markov Chain Monte Carlo simulation (MCMC) which makes use of the slice sampling technique (see, e.g., Neal, 2003).

Finally, the estimator of the probability of failure is the production of the two Monte Carlo estimators:

$$\widehat{\alpha}_{metaIS} = \widehat{\alpha}_1\widehat{\alpha}_2. \tag{4.11}$$

Notice that the Monte Carlo simulation for approximating the augmented failure probability α_1 is applied on the cheap surrogate model which is quite easy to perform, thus m can be a very large number. The total number of evaluations N thus consists of the number of evaluations needed for building the kriging surrogate model N_1 and the number of evaluations needed to approximate the correction factor α_2, and $N = N_1 + N_2$.

Dubourg et al. (2011a) adopted a sequential strategy to adaptively refine the kriging model during the first stage. In Auffray et al. (2011), a maximin design of a preset number of evaluations is used to build the kriging predictor.

Although two-stage strategies take advantage of Gaussian process prior modeling technique (the same as our proposed algorithm) and importance sampling and have shown some promising results in numerical examples, their capacities of estimating small probabilities of failure are still quite limited. The major disadvantage of both two-stage strategies is that the kriging model at the first stage is built targeting a very high threshold, thus the prediction could be very inaccurate due to

the lack of information near the threshold. In the case of dealing with a probability as low as 10^{-12} or for high dimension problems, the two-stage strategies become invalid.

Remark 6. *In the recent work of Dubourg (2011); Dubourg et al. (2011b), a sequential procedure is proposed to update the proposal distribution function $q(x)$ using MCMC and clustering techniques, which alleviate the problem mentioned above. However, it is still targeting a very high threshold at the beginning. Due to the lack of time, we could not compare their method in this work.*

4.2.2 Combining Subset Simulation with SVM (^{2}SMART)

Bourinet et al. (2011); Deheeger and Lemaire (2007) combined support vector machine (SVM) with Subset Simulation to propose a method called ^{2}SMART, and used it to assess small probabilities of failure. The approach aims at considering the Subset Simulation technique from the viewpoint of support vector machine (SVM) classification. The method was developed from the pioneering work of Hurtado (2004a), which was the first to introduce SVM to reliability analysis.

Here, we assume that all the random variables are mapped to the standard space by the transformation introduced in Chapter 2, Section 2.2.2. SVM classifiers are constructed in this standard space.

The key idea of the ^{2}SMART algorithm is to build a SVM classifier at each intermediate threshold u_t defined as in Subset Simulation (see Chapter 2), which is much easier to tackle than the target threshold u. At each stage, a "coarse to fine" sequential design strategy is used to refine the SVM classifier. The intermediate probabilities $\mathsf{P}_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t)$ (for $t = 0, \ldots, T-1$, and T is the total number of stages) are then calculated from the evaluations on the cheap SVM surrogate models. Table 4.1 presents the general structure of the ^{2}SMART algorithm.

The p_0 in Table 4.1 denotes the intermediate probability at each stage, which is usually set to $p_0 = 0.1$. T denotes the number of stages needed for the Subset Simulation procedure. $\widehat{\Gamma}_t$ is an approximation of Γ_t at stage t, provided by the updated SVM classifier at intermediate threshold u_t.

Evaluation resource needed for the ^{2}SMART algorithm consists in the points to setup and refine the SVM classifiers at each stage, which is a summary of the initial training data N_{init} and the sequential design N_{iter} (see Table 4.1). Thus, the total number of evaluations is $N = T(N_{init} + N_{iter})$.

Table 4.1: ^{2}SMART algorithm.

Initialize: $(t = 0)$

- Set initial threshold $u_0 = -\infty$, and p_0;
- Generate a MC sample $\mathbb{Y}_0 = \{Y_0^1, \ldots, Y_0^m\}$, drawn according to the distribution $\mathsf{P}_{\mathbb{X}}$.

Iteration: For $t = 1, 2, \ldots, T$ (Subset Simulation loop):

- Choose N_{init} initial training points.
  - $\rightarrow$ Decide u_t intermediate threshold;
  - $\rightarrow$ Get initial SVM classifier at u_t.
- For $n = 1, \ldots, N_{iter}$ (Update SVM classifier loop):
  - $\rightarrow$ Fine a new point according to the coarse to fine adjustment strategy;
  - $\rightarrow$ Update learning database;
  - $\rightarrow$ Train and update SVM classifier at u_t.
- Evaluate intermediate probability:

$$\mathsf{P}(\widehat{\Gamma}_{t+1}|\widehat{\Gamma}_t) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{Y_t^i \in \widehat{\Gamma}_{t+1}}, \tag{4.12}$$

  where $\widehat{\Gamma}_t$ is obtained from the final SVM classifier at u_t.
- Update MC sample $\mathbb{Y}_t$ using MCMC.

Evaluate: Final probability of failure is:

$$\widehat{\alpha}_{2SMART} = \prod_{i=0}^{T-1} \mathsf{P}(\widehat{\Gamma}_{t+1}|\widehat{\Gamma}_t). \tag{4.13}$$

The ^{2}SMART algorithm shares a few common ideas with Bayesian Subset Simulation algorithm that will be proposed in the next section, however, the sequential strategies used to build the surrogate models are quite different. Indeed, instead of using SVM classifiers, a Gaussian process prior model and kriging predictor is adopted, which provides us the information of confidence intervals and probabilities of mis-classification on the prediction, thus our strategy is expected to gain further efficiency by adaptively choosing the number of points needed at each stage. The problem of applying the ^{2}SMART algorithm is that it still requires thousands of evaluations and the number of evaluations is fixed in advance, which makes it difficult to use in expensive-to-evaluate cost functions. We will compare the ^{2}SMART algorithm with both the original Subset Simulation and our proposed Bayesian Subset Simulation algorithm in Section 4.4.

4.3 Bayesian Subset Simulation algorithm

4.3.1 Algorithm

Our objective is to build an estimator of $\alpha(u_T)$ from the evaluation results of f at a number of points $X_1, X_2, \ldots, X_N \in \mathbb{X}$. Let ξ be a random process modeling our prior knowledge about f, and for each $n \geq 0$, denote by $\mathcal{F}_n$ the σ-algebra generated by $X_1, \xi(X_1), \ldots, X_n, \xi(X_n)$. A natural Bayesian estimator of $\alpha(u_t)$ using n_t evaluations is the posterior mean

$$\widehat{\alpha}_t \;=\; \mathsf{E}_{n_t}\left(\alpha(u_t)\right) \;=\; \mathsf{E}_{n_t}\left(\int_{\mathbb{X}} \mathbf{1}_{\xi > u_t}\,\mathrm{d}\mathsf{P}_{\mathbb{X}}\right) \;=\; \int_{\mathbb{X}} g_t\,\mathrm{d}\mathsf{P}_{\mathbb{X}}, \tag{4.14}$$

where $g_t : x \in \mathbb{X} \mapsto \mathsf{P}_{n_t}\big(\xi(x) > u_t\big)$, n_t is the number of evaluations used until stage t, and E_n (resp. P_n) denotes the conditional expectation (resp. conditional probability) with respect to $\mathcal{F}_n$. Note that, $g_t(x)$ can be readily computed for any x using kriging (see Chapter 3).

Assume now that $\mathsf{P}_{\mathbb{X}}$ has a probability density function $p_{\mathbb{X}}$ and consider the sequence of probability density functions q_t, $0 \leq t \leq T$, defined by

$$q_t(x) \;=\; \frac{1}{\widehat{\alpha}_t}\,p_{\mathbb{X}}(x)\,g_t(x). \tag{4.15}$$

We can write a recurrence relation similar to (4.1) for the sequence of Bayesian estimators $\widehat{\alpha}_t$:

$$\widehat{\alpha}_t \;=\; \int g_t(x)\,p_{\mathbb{X}}(x)\,\mathrm{d}x \;=\; \widehat{\alpha}_{t-1}\int \frac{g_t(x)}{g_{t-1}(x)}\,q_{t-1}(x)\,\mathrm{d}x. \tag{4.16}$$

The idea of our new algorithm, which we call Bayesian Subset Simulation, is to construct recursively a Monte Carlo approximation $\widehat{\widehat{\alpha}}_T$ of the Bayesian estimator $\widehat{\alpha}_t$, using (4.16) and sequential Monte Carlo (SMC) simulation (see, e.g., Del Moral et al., 2006) for the evaluation of the integral with respect to q_{t-1} on the right-hand side. More precisely, denoting by m the size of the Monte Carlo sample, we will use the recurrence relation

$$\widehat{\widehat{\alpha}}_t = \widehat{\widehat{\alpha}}_{t-1} \times \frac{1}{m} \sum_{i=1}^{m} \frac{g_t(Y_{t-1}^i)}{g_{t-1}(Y_{t-1}^i)}, \quad 1 \leq t \leq T, \tag{4.17}$$

where variables $Y_{t-1}^1, \ldots, Y_{t-1}^m$ are distributed according to[2] the density q_{t-1}, which leads to

$$\widehat{\widehat{\alpha}}_T = \prod_{t=0}^{T-1} \frac{1}{m} \sum_{i=1}^{m} \frac{g_{t+1}(Y_t^i)}{g_t(Y_t^i)}. \tag{4.18}$$

The connection between the proposed algorithm and the original Subset Simulation algorithm is clear from the similarity between the recurrence relations (4.1) and (4.16), and the use of SMC simulation in both algorithms to construct recursively a "product-type" estimator of the probability of failure (see also Del Moral et al., 2006, Section 3.2.1, where this type of estimator is mentioned in a very general SMC framework).

Our choice for the sequence of densities $q_1, \ldots, q_T$ also relates to the original Subset Simulation algorithm. Indeed, note that $q_t(x) \propto \mathsf{E}_{n_t}(\mathbf{1}_{\xi > u_t}\, p_{\mathbb{X}})$, and recall that $\tilde{q}_t \propto \mathbf{1}_{\xi > u_t}\, p_{\mathbb{X}}$ is the distribution used in the Subset Simulation algorithm at stage t. (This choice of instrumental density is also used by Dubourg et al. (2011a) in the context of a two-stage kriging-based adaptive importance sampling algorithm, as explained in Section 2. This is indeed a quite natural choice, since $\tilde{q}_T \propto \mathbf{1}_{\xi > u}\, p_{\mathbb{X}}$ is the optimal instrumental density for the estimation of $\alpha(u)$ by importance sampling.)

4.3.2 Implementation

This section gives implementation details for our Bayesian Subset Simulation algorithm. The pseudo-code for the algorithm is presented in Table 4.2.

The initial Monte Carlo sample $\mathbb{Y}_0 = \{Y_0^1, \ldots, Y_0^m\}$ is a set of independent random variables drawn from the density $q_0 = p_{\mathbb{X}}$—in other words, we start with a

[2] By "distributed according to", it is not meant that $Y_{t-1}^1, \ldots, Y_{t-1}^m$ are independent and identically distributed. This is never the case in sequential Monte-Carlo techniques. What we mean is that the sample $Y_{t-1}^1, \ldots, Y_{t-1}^m$ is *targetting* the density q_{t-1} (in the sense of, e.g., Douc and Moulines, 2008).

classical Monte Carlo simulation step. At each subsequent stage $t \geq 1$, a new sample $\mathbb{Y}_t$ is produced from the previous one using the basic reweight/resample/move steps of SMC simulation (see Del Moral et al., 2006, and the references therein). In this article, resampling is carried out using a multinomial sampling scheme, and the move step relies on a Metropolis algorithm with a Gaussian-random-walk proposal distribution. We denote the corresponding Markov transition kernel by K.

A number N_t of evaluations of the performance function is done at each stage of the algorithm. This number is meant to be much smaller than the size m of the Monte Carlo sample, which would be the number of evaluations in the classical Subset Sampling algorithm, and hopefully smaller than the number of evaluations used by ^{2}SMART algorithm. For the initialization stage ($t = 0$), we choose a space filling set of points $\mathbb{Y}_0$ as usual in the design of computer experiments (Santner et al., 2003a). At each subsequent stage t, we use N_t iterations of a SUR sampling strategy (see Chapter 3) targeting a preliminary estimate $\tilde{u}_t$ of the next threshold u_t to select the evaluation points (see Section 4.3.3). Adaptive techniques to choose the sequence of thresholds u_ts and the number of points per stage are presented in the following sections.

Remark 7. *The resampling step could most certainly benefit from more elaborate schemes, such as the residual resampling scheme (Bolic et al., 2003; Douc and Cappé, 2005; Hol et al., 2006). The comparison of resampling schemes is left for future work.*

4.3.3 Adaptive choice of the thresholds u_t

It can be proved that, for an idealized[3] Subset Simulation algorithm with fixed thresholds $u_1 < \cdots < u_T = u$, it is optimal to choose the thresholds to make all conditional probabilities $\mathsf{P}_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t)$ equal (see Cérou et al., 2011, Section 2.4). This leads to the idea of choosing the thresholds adaptively in such a way that, in the product estimate

$$\widehat{\alpha}_T^{\text{SubSamp}} = \prod_{t=1}^{T} \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{\Gamma_t}(Y_{t-1}^i),$$

each term but the last is equal to some predefined constant p_0. In other words, u_t is chosen as the $(1 - p_0)$-quantile of $\mathbb{Y}_{t-1}$. This idea was first suggested by Au and Beck (2001, Section 5.2), on the heuristic ground that the algorithm should

[3] assuming that $Y_t^1, \ldots, Y_t^m$ are independent and identically distributed according to q_t.

Table 4.2: Algorithm of Bayesian Subset Simulation

a) Initialize (Stage $t = 0$):

1. Generate a MC sample $\mathbb{Y}_0 = \{Y_0^1, \ldots, Y_0^m\}$, drawn according to the distribution $\mathsf{P}_{\mathbb{X}}$, set the initial number of evaluations $n = n_0$, and $u_0 = -\infty$.

2. Initial space-filling DoE $\mathcal{I}_{n_0} = \Big((X_1, f(X_1)), \ldots, (X_{n_0}, f(X_{n_0}))\Big)$.

3. Choose an appropriate kriging model, estimate covariance parameters.

b) While $u_t < u$ $(t \leftarrow t + 1)$:

1. Compute the kriging predictor $\widehat{\xi}_n$, and choose a preliminary threshold $\tilde{u}_t$.

2. While the stopping criterion is not met $(n \leftarrow n + 1)$:

2.1 add a new point x_{new} using a SUR sampling criterion w.r.t. $\tilde{u}_t$.

2.2 update $\mathcal{I}_n = \Big(\mathcal{I}_{n-1}, (x_{new}, f(x_{new}))\Big)$, and re-estimate covariance parameters.

3. Calculate the intermediate threshold u_t. If $u_t > u$, go to step c).

4. Generate a new sample $\mathbb{Y}_t$:

4.1 reweight: calculate weights: $w_i^t \propto g_t(Y_{t-1}^i)/g_{t-1}(Y_{t-1}^i)$,

4.2 resample: generate a sample $\widetilde{Y}_{t-1,i}$ according to weights,

4.3 move: for each $i \leq m$, $Y_t^i \backsim K\big(\widetilde{Y}_{t-1,i}, \bullet\big)$.

c) The final estimation of the probability of failure is calculated by

$$\widehat{\widehat{\alpha}}_T = \prod_{t=0}^{T-1} \Big(\frac{1}{m} \sum_{i=1}^{m} \frac{g_{t+1}(Y_t^i)}{g_t(Y_t^i)}\Big).$$

perform well when the conditional probabilities are neither too small (otherwise they are hard to estimate) nor too large (otherwise a large number of stages is required). The asymptotic behaviour of the resulting algorithm, when m is large, has been anayzed by Cérou et al. (2011, Section 3).

In Bayesian Subset Simulation, we propose to choose the thresholds adaptively using a similar approach. More precisely, considering the product form of the estimator (4.18), we suggest to choose u_t in such a way that

$$\frac{1}{m}\sum_{i=1}^{m}\frac{g_t(Y_{t-1}^i)}{g_{t-1}(Y_{t-1}^i)} \;=\; p_0. \tag{4.19}$$

Equation (4.19) can be easily solved since the left-hand side is a strictly decreasing function of u_t.

As we have seen in the implementation in Table 4.2, we need to target a preliminary threshold $\tilde{u}_t$ to apply a SUR strategy. In practice, we calculate $\tilde{u}_t$ using an equation similar to (4.19):

$$\frac{1}{m}\sum_{i=1}^{m}\frac{\mathsf{P}_{n_{t-1}}(\xi(Y_{t-1}^i) > \tilde{u}_t)}{g_{t-1}(Y_{t-1}^i)} \;=\; p_0. \tag{4.20}$$

Remark 8. *Note that Cérou et al. (2011) proved that choosing adaptive levels in Subset Simulation introduces a positive bias of order $1/m$, which is negligible compared to the standard deviation.*

4.3.4 Adaptive choice of the number N_t of evaluation at each stage

In this section, we propose a technique to choose adaptively the number N_t of evaluations of the performance function that must be done at each stage of the algorithm.

Let us assume that $t \geq 1$ is the current stage number; at the beginning of the stage, n_{t-1} evaluations of the performance function are available from previous stages. After several additional evaluations, the number of available observations of f is $n \geq n_{t-1}$. Then, for each $i \in \{1, \ldots, m\}$, the probability of misclassification[4] of $x \in \mathbb{X}$ with respect to the threshold u_t is

$$\tau_{t,n}(x) \;=\; \min\Big(p_n(x,u_t),\, 1 - p_n(x,u_t)\Big),$$

where $p_n(x,u) = \mathsf{E}_n\left(\mathbf{1}_{\xi(x)>u}\right)$. We shall decide to stop adding new evaluations at stage t when

$$\frac{1}{m}\sum_{i=1}^{m}\tau_{t,n}\left(Y_{t-1}^i\right) \leq \eta,$$

[4] See Section 3.4 for more information

for some prescribed $\eta > 0$.

4.4 Numerical results

In this section, we apply the proposed algorithm on three examples from the structural reliability literature. In each example, the number of evaluations needed at each stage are provided. The results obtained from Bayesian Subset Simulation are then compared with a reference value from a brute force Monte Carlo simulation, the classical Subset Simulation algorithm, as well as the ^{2}SMART algorithm.

For all examples, a Gaussian process with constant unknown mean and a Matérn covariance function is used as our prior information about f. The parameters of the Matérn covariance functions are estimated on the initial design by REML (see, e.g., Stein, 1999). In our experiments, we follow the common practice of re-estimating the parameters of the covariance function during the sequential strategy, and update the covariance function after SUR strategy. The target conditional probability between successive thesholds is set to $p_0 = 0.1$.

4.4.1 Example 1: Four-branch series system

Our first example is the four-branch series system studied by Borri and Speranzini (1997); Deheeger (2008); Schueremans (2001); Waarts (2000). Echard et al. (2010a,b) used it recently to compare several methods proposed in Schueremans and Van Gemert (2005), some of which are based on the construction of a response surface. We also used it to evaluate the performance of SUR strategies (see, e.g., Bect et al., 2010; Li et al., 2011). In this work, however, we consider a higher threshold for the failure event (hence, a smaller probability of failure).

The performance function f is defined as follows:

$$f : (x_1, x_2) \in \mathbb{R}^2 \mapsto f(x_1, x_2) = \min \left\{ \begin{array}{l} 3 + 0.1(x_1 - x_2)^2 - (x_1 + x_2)/\sqrt{2}, \\ 3 + 0.1(x_1 - x_2)^2 + (x_1 + x_2)/\sqrt{2}, \\ (x_1 - x_2) + 6/\sqrt{2}, \\ (x_2 - x_1) + 6/\sqrt{2} \end{array} \right\}. \tag{4.21}$$

The uncertain input factors are supposed to be independent and have standard normal distribution. A failure happens when the system is working under the threshold u. Figure 4.1 shows the failure domain and the input distribution. Observe that f has a first-derivative discontinuity along four straight lines originating

from the point $(0, 0)$.

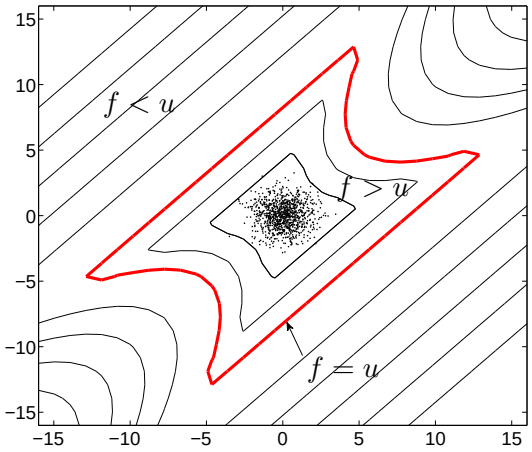


Figure 4.1: Example 1: contour plot of f; limit state $f = u$ (thick red line); sample of size $m = 10^3$ from $\mathsf{P}_{\mathbb{X}}$ (dots).

We use $u = -4$ as the threshold for the definition of the failure event. The probability of failure, obtained using $\widehat{\alpha}_m^{\mathrm{MC}}$ with $m = 10^{12}$, is approximately 5.49×10^{-9} (with a coefficient of variation of about $1/\sqrt{m\,\alpha} \approx 1.4\%$).

The Bayesian Subset Simulation algorithm is initialized with a maximin LHS (Latin Hypercube Sampling)[5] design of size $N_0 = 10$ on $[-15; 15] \times [-15; 15]$. At each stage, we choose a Monte Carlo sample of size $m = 10^3$ (or $m = 10^4$). The proposal distribution q_t in the random walk step is a normal distribution $\mathcal{N}(0, 1)$ for each coordinate. The stopping criterion for the adaptive SUR strategy is set to $\eta_t = 5 \times 10^{-8}$ (for $t = 1, \ldots, T-1$) and $\eta_T = 10^{-9}$.

Table 4.3 lists the total number of evaluations needed at each stage averaged over 50 independent runs. A total of $N = \sum_{t=0}^{T} N_t = 352$ evaluations are needed for our proposed algorithm with $m = 1000$, while for Subset Simulation, the number is $1000 + 900 \times 8 = 8200$.

To evaluate the statistical properties of the estimator, we consider the *absolute relative bias*

$$\kappa = \left| \frac{\mathsf{E}(\widehat{\alpha}) - \alpha}{\alpha} \right|, \tag{4.22}$$

[5] More precisely, we use Matlab's `lhsdesign()` function to select the best design according to the maximin criterion among 10^4 randomly generated LHS designs.

Table 4.3: Example 1: average number of evaluations at each stage with $m = 1000$.

t	1	2	3	4	5	6	7	8	9
Sub-Sim	1000	900	900	900	900	900	900	900	900
Bayesian Sub-Sim	46	42	44	40	37	33	31	30	39

Table 4.4: Example 1: comparison with Subset Simulation and ^{2}SMART algorithm (MCS estimator is a reference obtained from only one simulation).

Method	m	N	$\mathsf{E}(\widehat{\alpha})$ (10^{-9})	$\delta(\widehat{\alpha})$ (10^{-9})	κ	cov
MCS	10^{12}	10^{12}	5.49	0.074	0	1.4%
Sub-Sim	1000	8200	5.23	2.865	4.8%	54.8%
^{2}SMART	–	$[2753, 4279]$	5.42	0.872	1.2%	15.9%
Bayesian Sub-Sim	1000	$[325, 386]$	5.39	1.438	1.9%	26.2%
	10^4	$[706, 773]$	5.65	0.441	2.9%	8.0%

and the *coefficient of variation*

$$\text{cov} = \frac{\delta(\widehat{\alpha})}{\alpha}, \tag{4.23}$$

where $\mathsf{E}(\widehat{\alpha})$ is the average and $\delta(\widehat{\alpha})$ is the standard deviation of the estimator $\widehat{\alpha}$.

Table 4.4 shows the results of the comparison of our proposed Bayesian Subset Sampling algorithm with the Subset Simulation algorithm (see Au and Beck, 2001) and the ^{2}SMART algorithm (see Bourinet et al., 2011; Deheeger and Lemaire, 2007). The results for ^{2}SMART are directly obtained from software package FERUM (see the user's guide in Bourinet, 2010). Crude Monte Carlo sampling is used as the reference probability of failure. We run our Bayesian Subset Simulation algorithm for both $m = 10^3$ and 10^4. For ^{2}SMART algorithm, the parameters are set to be the default as in FERUM. In addition, as there are several procedures of Monte Carlo simulation in ^{2}SMART, the single sample size m is not valid in this case. 50 independent runs are performed to evaluate the average of both methods. For Bayesian Subset Simulation method and ^{2}SMART algorithm, as N is different for each run, we show the minimal and maximal of N from 50 runs.

Figure 4.2 shows the evaluation points selected at stages $t = 1, 2, 9$. It is observed that the points that are chosen by the SUR strategy at each stage are close to the frontier of the intermediate threshold u_t. We also compare in Figure 4.3 – 4.5 the predicted values versus the true values of the function f at stages all stages, both

Table 4.5: Example 2: random input factors.

Variable	Distribution	Mean m	Standard deviation σ
x_1	$\mathcal{N}$	0.001	0.0002
x_2	$\mathcal{N}$	250	37.5

before (left column) and after (right column) the addition of new samples to the design. It appears clearly that the new evaluations allow to reduce efficiently the prediction error in the neighborhood of the current threshold.

4.4.2 Example 2: deviation of a cantilever beam

In this section, we apply the proposed algorithm on another 2D test case from the structural reliability literature. The problem under consideration is the deviation of a cantilever beam, with a rectangular cross-section, and subjected to a uniform load (Gayton et al., 2003; Rajashekhar and Ellingwood, 1993b). The cost function is:

$$f(x_1, x_2) = 18.46154 - 7.476923 \times 10^{10} \, \frac{x_1}{{x_2}^3} \, . \tag{4.24}$$

The uncertain factors are x_1 and x_2, which are supposed to be independent and have normal distribution, as specified in Table 4.5. We use $u = 17.8$ as the threshold for the definition of the failure event, and a failure happens when the cost is larger than threshold. The probability of failure, which will be used as reference estimator, obtained using $\widehat{\alpha}_m^{\mathrm{MC}}$ with $m = 10^8$, is approximately 3.85×10^{-5} (with a coefficient of variation of about $1/\sqrt{m\,\alpha} \approx 1.6\%$). Figure 4.6 shows the distribution of the input factors along with a contour plot of f. Notice that the failure region is quite far from the center region of the input distribution.

In the Bayesian Subset Simulation algorithm, we set an initial design of size $N_0 = 10$ which is equal to five times the dimension d of the input space (In literature, very little is known about the problem of choosing N_0, however some authors recommend to start with a sample size proportional to the dimension d, see Loeppky et al. (2009)). Concerning the choice of N_0, we decide to apply a greedy MAXMIN algorithm and sequentially choose the points which will maximize the minimal Euclidean distance between any two points in the initial Monte Carlo sample $\mathbb{Y}_0$. At each stage, the Monte Carlo sample size keeps the same. The intermediate threshold u_t is chosen by the criterion (4.19). The proposal distribution q_t for a random walk is a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, where σ is

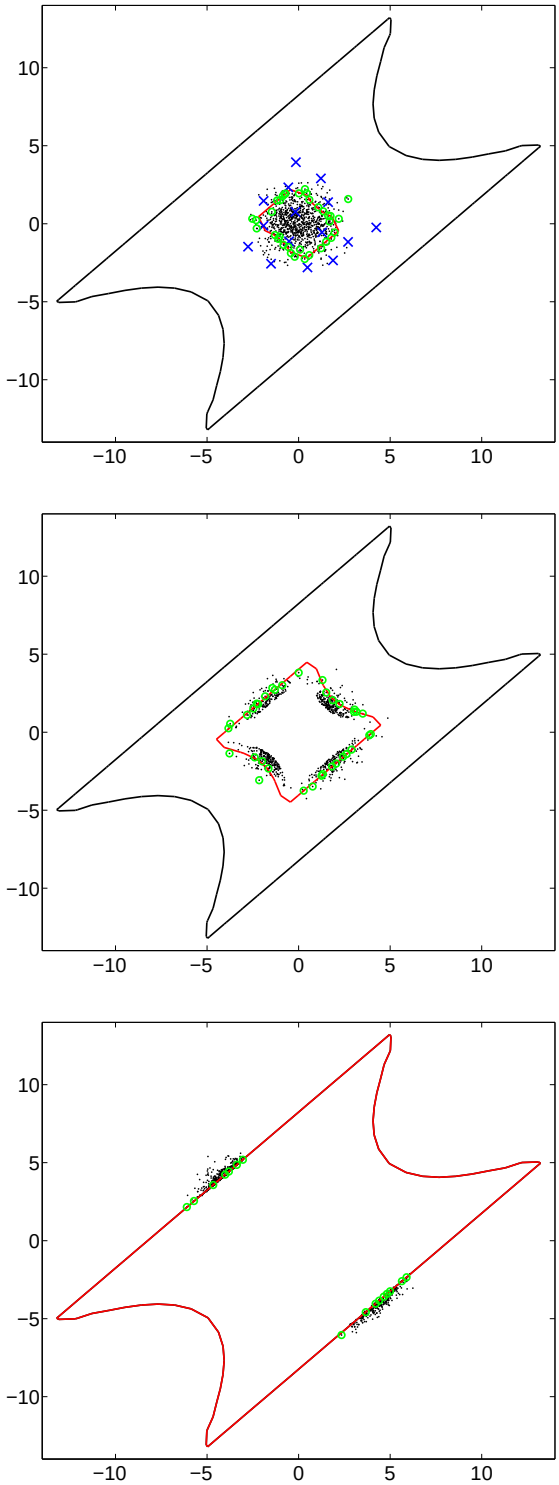


Figure 4.2: Example 1: the Design of Experiment (DoE) at stage $1, 3$ and last stage. The black line is the real threshold contour; the red line is the intermediate threshold at stage $t = 1, 3, 9$; black spots are the Monte Carlo sample at stage t, and green circles the evaluated points chosen by SUR strategy; blue crosses at the stage 1 are the initial design n_0.

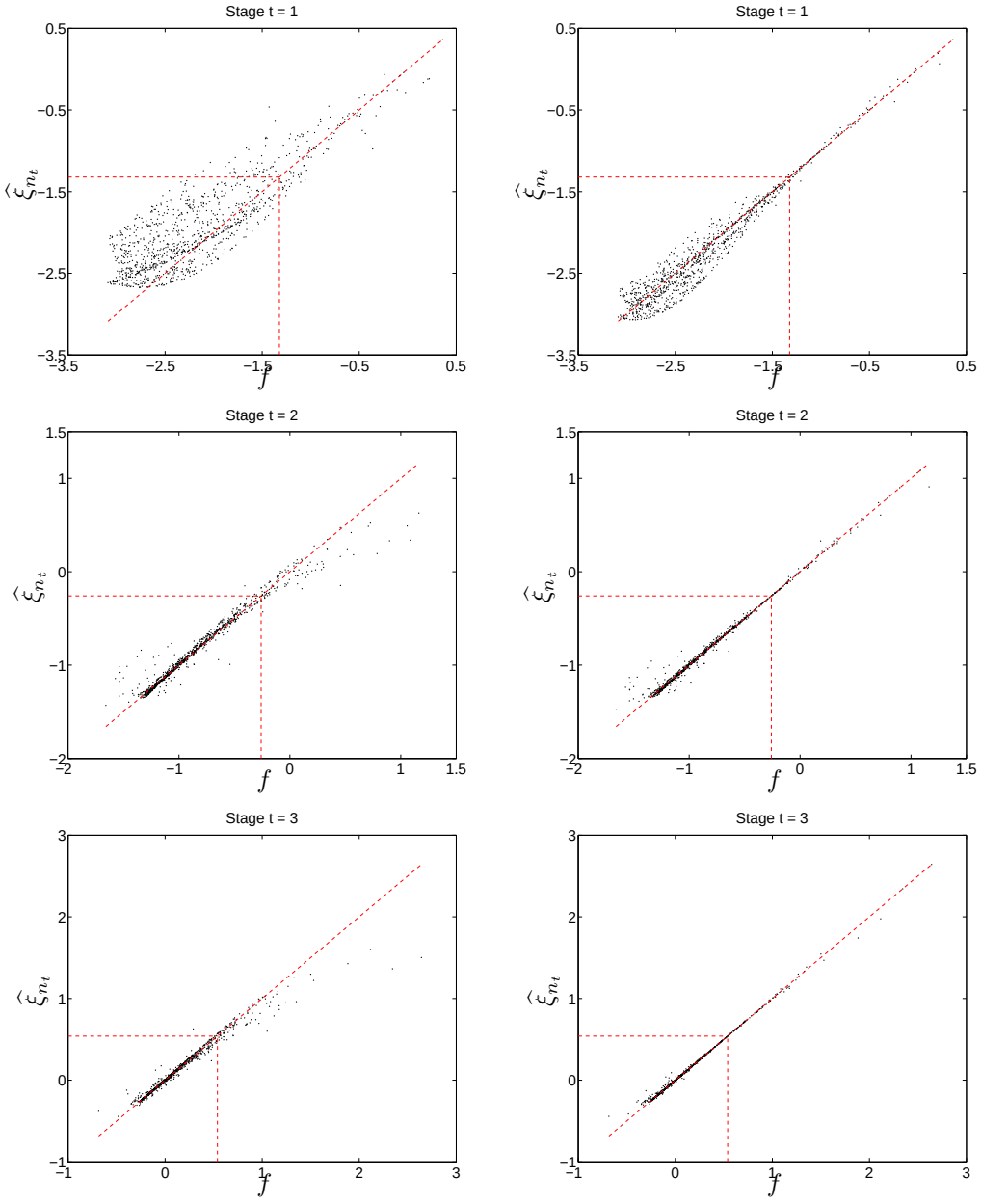


Figure 4.3: Example 1: predicted vs true value of the performance function, before (left column) and after (right column) the addition of new evaluations, for stages $t = 1, 2, 3$ (from top to bottom).

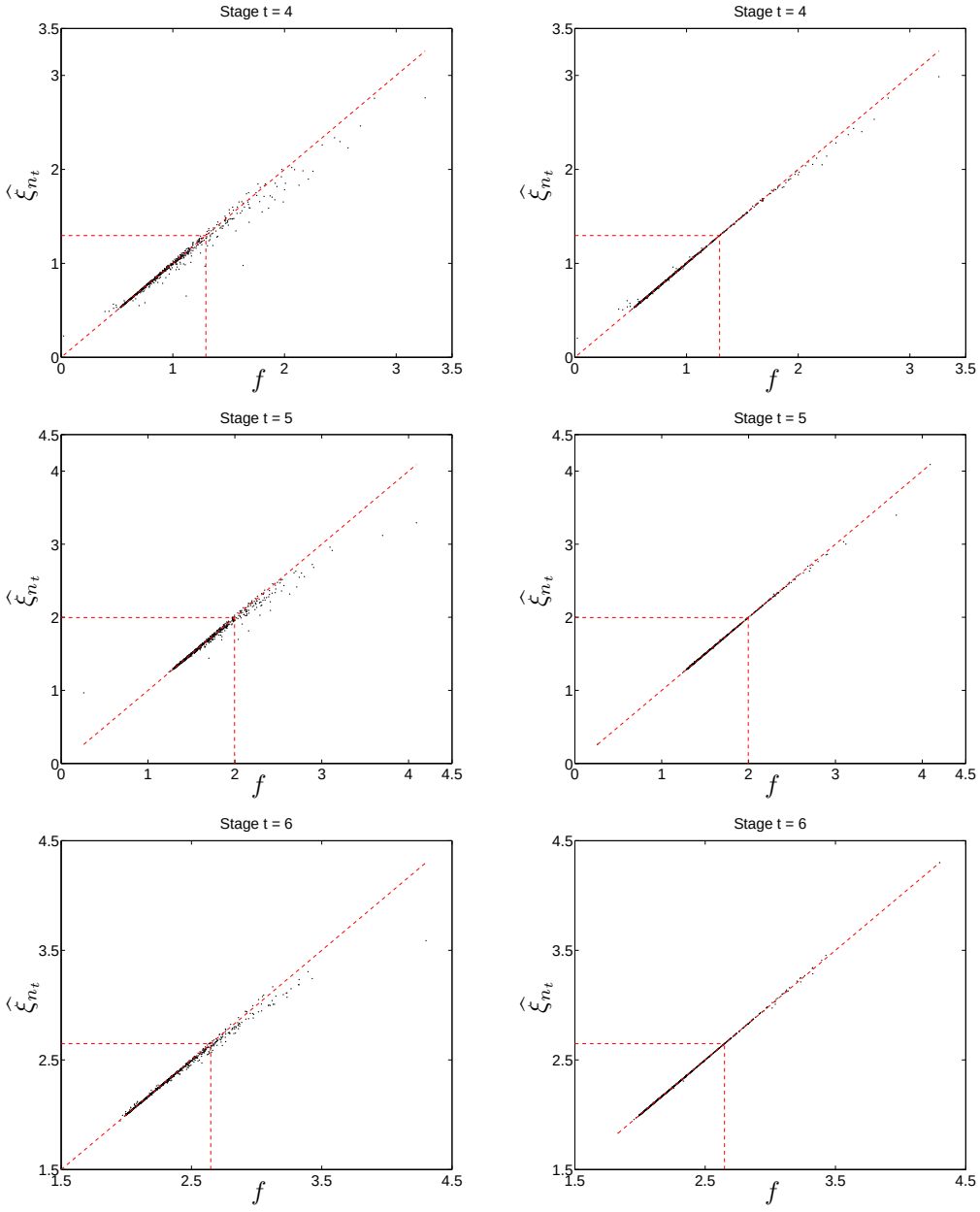


Figure 4.4: Example 1: continuation of Figure 4.3. Predicted vs true value of the performance function, before (left column) and after (right column) the addition of new evaluations, for stages $t = 4, 5, 6$ (from top to bottom).

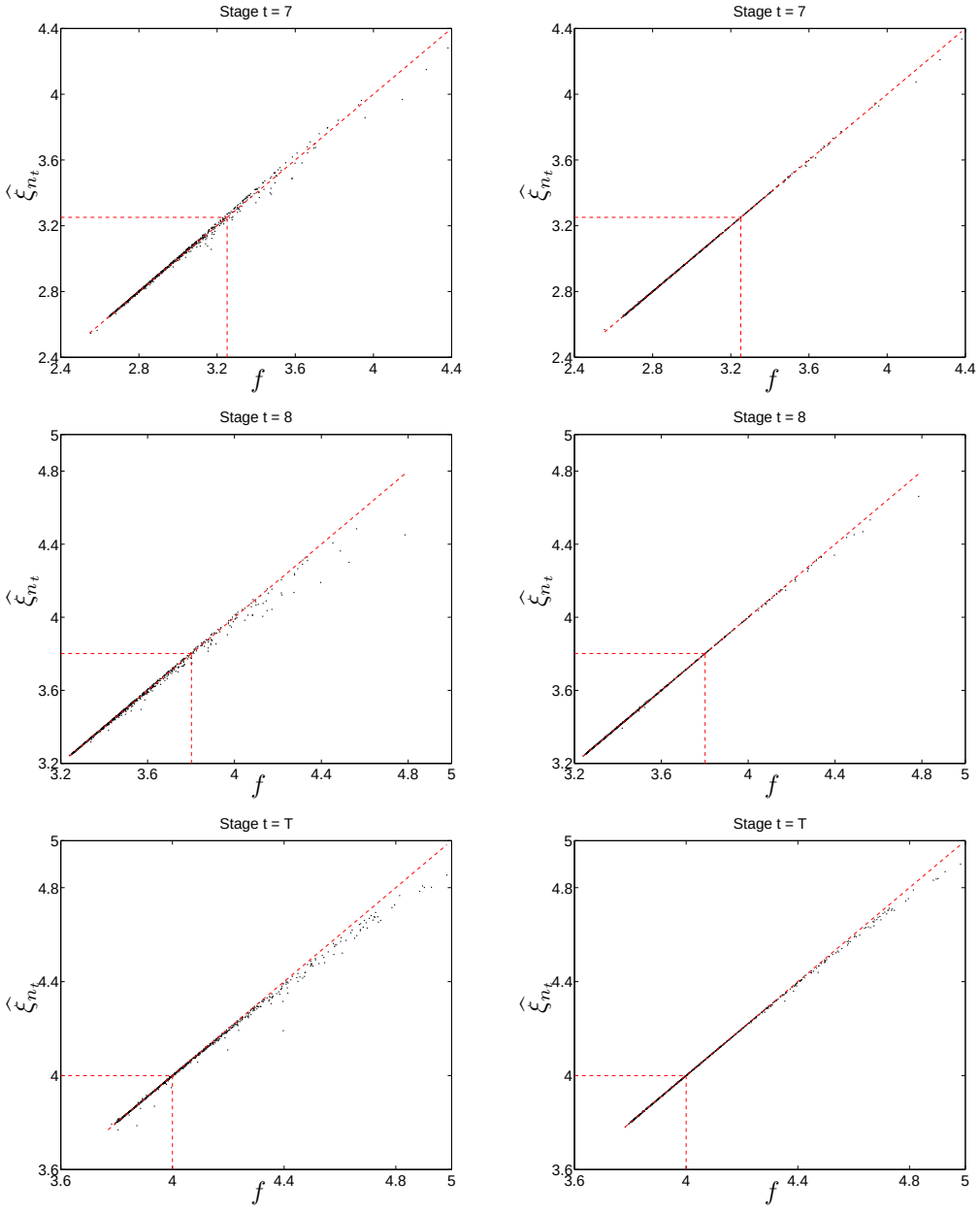


Figure 4.5: Example 1: continuation of Figure 4.4. Predicted vs true value of the performance function, before (left column) and after (right column) the addition of new evaluations, for stages $t = 7, 8, 9$ (from top to bottom).

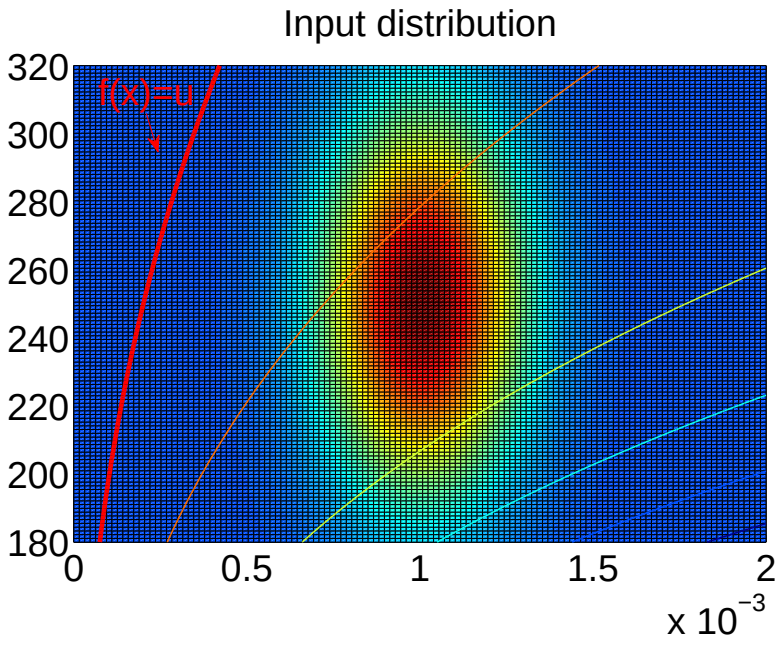


Figure 4.6: Example 2: input distribution and contour plot of the performance function.

Table 4.6: Example 2: average number of evaluations at each stage with $m = 1000$.

t	1	2	3	4	5
Sub-Sim	1000	900	900	900	900
Bayesian Sub-Sim	14	17	17	18	28

specified in Table 4.5. The stopping criterion for the adaptive SUR strategy is set to $\eta_t = 5 \times 10^{-3}$ (for $t = 1, \ldots, T-1$) and $\eta_T = 10^{-3}$.

Figure 4.7 shows the Design of Experiment (DoE) selected by the algorithm at stage $t = 1, 2, 3$ and the last stage for one run. Table 4.6 lists the average number of evaluations (rounded to integer) at each stage over 50 runs. We can see that an average total of evaluations $N = \sum_{i=0}^{T} N_t = 104$ are needed for our proposed Bayesian Subset Simulation, while for Subset Simulation, the number is $1000 + 900 \times 4 = 4600$.

Table 4.7 shows the results of the comparison of our proposed Bayesian Subset Sampling algorithm with the Subset Simulation algorithm in (Au and Beck, 2001). Crude Monte Carlo sampling is used as the reference probability of failure. For the same reason as in Example 1, we set the same intermediate probability $p_0 = 0.1$, and sample size $m = 1000$ and 10^4 for Bayesian Subset Simulation. 50 independent

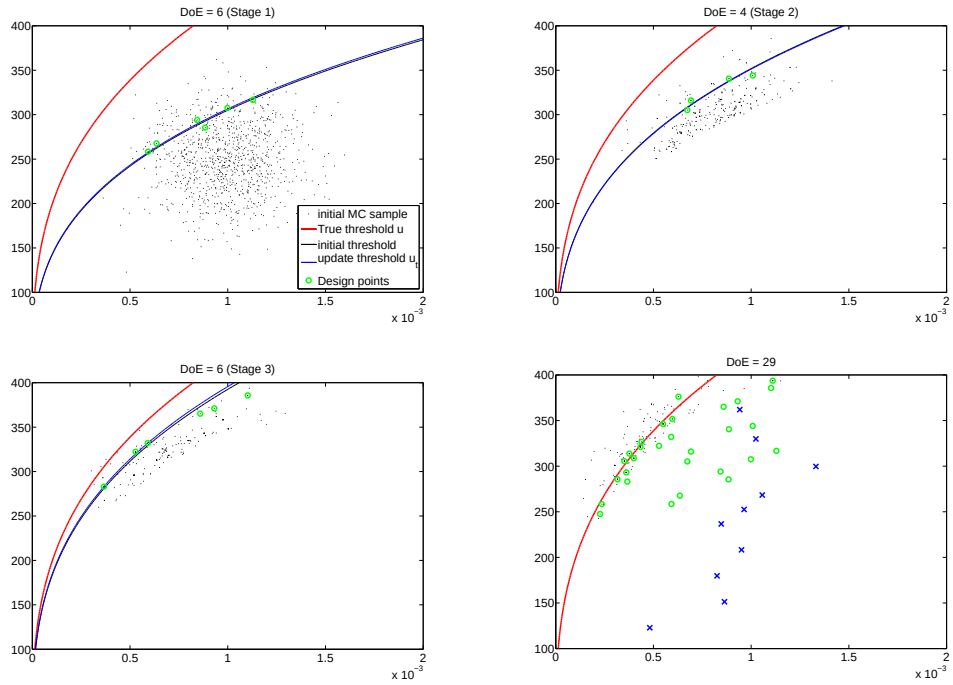


Figure 4.7: Example 2: evaluations selected by the Bayesian Subset Simulation algorithm at stages $t = 1, 2, 3$ and the final Design of Experiment (DoE); blue crosses indicate the initial design n_0.

runs are performed to evaluate the average statistical properties of the estimators.

Table 4.7: Example 2: comparison with Subset Simulation and ^{2}SMART algorithm (MCS estimator is a reference obtained from only one simulation).

Method	m	N	$\mathsf{E}(\widehat{\alpha})$ (10^{-5})	$\delta(\widehat{\alpha})$ (10^{-5})	κ	cov
MCS	10^8	10^8	3.85	0.062	0	1.6%
Sub-Sim	1000	4600	3.90	2.470	1.5%	63.2%
^{2}SMART	–	$[1332, 1722]$	3.74	0.097	2.6%	2.5%
Bayesian Sub-Sim	1000	$[94, 109]$	3.70	0.618	3.8%	16.7%
	10^4	$[325, 422]$	3.77	0.223	2%	5.8%

Influence of the sample size m

In order to study the influence of the choice of the Monte Carlo sample size m on the statistic properties of the estimator, we run 50 independent simulations with sample size $m = 50, 100, 1000$ and 10^4. The results are presented in Table 4.8.

As expected, the coefficient of variation of the estimator decreases as the sample size increases. However, increasing m brings out a price: for $m = 10^4$, the cost of running the Bayesian Subset Sampling algorithm is very high. Thus, the choice of m is a trade-off between an acceptable coefficient of variation and the cost of computational resources. From our experiments, we suggest that taking $m = 1000$ is a good choice for intermediate probabilities $p_0 = 0.1$.

Table 4.8: Example 2: study the influence of different Monte Carlo sample size m on the estimator.

Monte Carlo sample m	50	100	1000	10^4
κ	37.8%	31.2%	3.8%	2.6%
cov	113.4%	107.0%	16.7%	4.4%
Time	$12s$	$40s$	$218s$	$\sim 3h$

4.4.3 Example 3: response of a nonlinear oscillator

In this section, we consider an example about the dynamic response of a nonlinear oscillator, which was taken from the literature of structural reliability (see, e.g., Bucher and Bourgund, 1990; Gayton et al., 2003; Rajashekhar and Ellingwood, 1993b). The problem deals with a nonlinear undamped single-degree-of-freedom system as depicted in Figure 4.8.

In this example, the input variable is in dimension $d = 6$, and the cost function $f : \mathbb{R}^6 \to \mathbb{R}$ is:

$$f : x = (x_1, x_2, x_3, x_4, x_5, x_6) \mapsto 3x_4 - \left| \frac{2x_5}{x_1 {w_0}^2} \sin\left(\frac{w_0 x_6}{2}\right) \right| \tag{4.25}$$

where $w_0 = \sqrt{\frac{x_2+x_3}{x_1}}$. The distributions of the factors are specified in Table 4.9.

A failure happens when the cost function is larger than the threshold $u = 1.5$. The reference probability of failure, obtained using $\widehat{\alpha}_m^{\mathrm{MC}}$ with $m = 10^8$, is approximately 2.28×10^{-4} (with a coefficient of variation of about $1/\sqrt{m\,\alpha} \approx 0.7\%$).

The Monte Carlo sample size is $m = 1000$ for the Subset Simulation at each stage, and $m = 1000$ or 10^4 for the proposed Bayesian Subset Simulation algorithm. The initial design is set to $N_0 = 20$. The intermediate threshold u_t is chosen by the criterion (4.19). The proposal distribution q_t for a random walk is a Gaussian distributed $\mathcal{N}(0, \sigma^2)$, where σ^2 is specified in Table 4.9. The stopping criterion for the adaptive SUR strategy is set to $\eta_t = 10^{-5}$ (for $t = 1, \ldots, T-1$) and $\eta_T = 10^{-6}$.

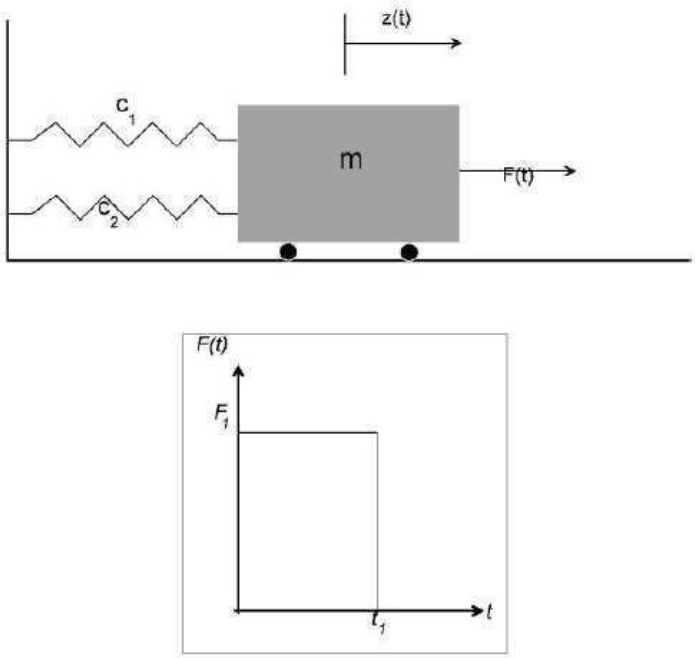


Figure 4.8: Example 3: non-linear oscillator.

Table 4.9: Example 3: random variables.

Variables	Distribution	Mean m	Standard deviation σ
x_1	$\mathcal{N}$ormal	1	0.05
x_2	$\mathcal{N}$	1	0.10
x_3	$\mathcal{N}$	0.1	0.01
x_4	$\mathcal{N}$	0.5	0.05
x_5	$\mathcal{N}$	1	0.20
x_6	$\mathcal{N}$	1	0.20

Table 4.10: Example 3: average number of evaluations at each stage with $m = 1000$.

t	1	2	3	4
Sub-Sim	1000	900	900	900
Bayesian Sub-Sim	13	16	17	35

Table 4.10 lists the average number of evaluations (rounded to integer) at each stage over 50 runs. We can see that an average total of evaluations $N = \sum_{i=0}^{T} N_t = 101$ are needed for our proposed Bayesian Subset Simulation, while for Subset Simulation, the number is $1000 + 900 \times 3 = 3700$.

We compare our proposed algorithm with Subset Simulation and ^{2}SMART algorithm in Table 4.4.3. Still, Monte Carlo method provides a reference probability of failure. The total number of evaluations N in Bayesian Subset Simulation algorithm and ^{2}SMART algorithm is a range of the minimal and maximal number of evaluations from 50 runs.

Table 4.11: Example 3: comparison with Subset Simulation and ^{2}SMART algorithm (MCS estimator is a reference obtained from only one simulation).

Method	m	N	$\mathsf{E}(\widehat{\alpha})$ (10^{-4})	$\delta(\widehat{\alpha})$ (10^{-4})	κ	cov
MCS	10^8	10^8	2.280	0.015	0	0.7%
Sub-Sim	1000	3700	2.341	0.746	2.7%	31.9%
^{2}SMART	–	$[2028, 2535]$	2.309	0.052	1.2%	2.2%
Bayesian Sub-Sim	1000	$[84, 117]$	2.286	0.384	0.3%	16.8%
	10^4	$[91, 118]$	2.275	0.103	0.2%	4.5%

At last, we provide the predictions versus the true function values as well as the probability of excursion versus the true values at the end of SUR strategy at different stages in Figure 4.9 – Figure 4.12.

4.5 Conclusions

In this paper, we propose a new algorithm called Bayesian Subset Simulation for estimating small probabilities of failure in a context of very expensive simulations. This algorithm combines the main ideas of the Subset Simulation algorithm and the SUR strategies developed in our recent work (Bect et al., 2010).

Our results show that the number of evaluations is dramatically decreased com-

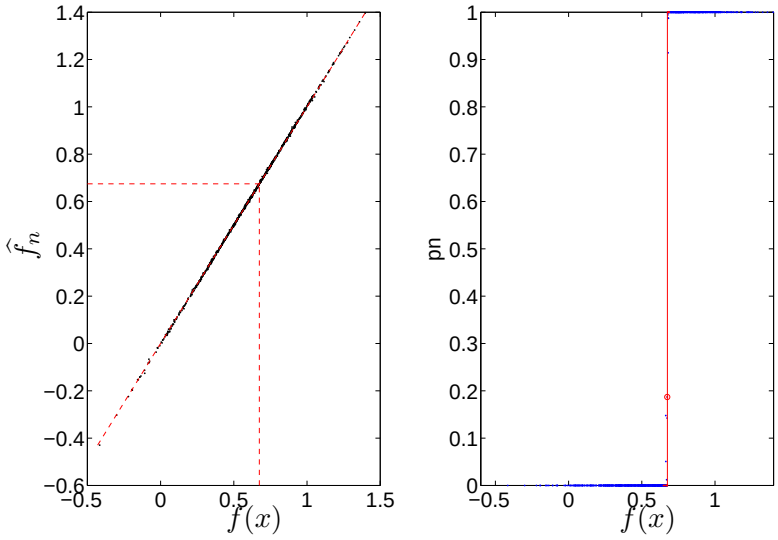


Figure 4.9: Example 3: the true value v.s. prediction and the probability of excursion. Stage: $t = 1$.

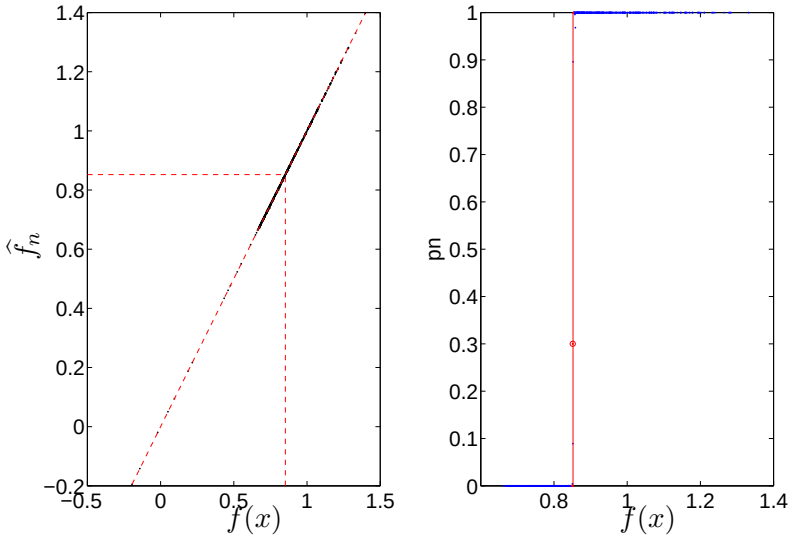


Figure 4.10: Example 3: the true value v.s. prediction and the probability of excursion. Stage: $t = 2$.

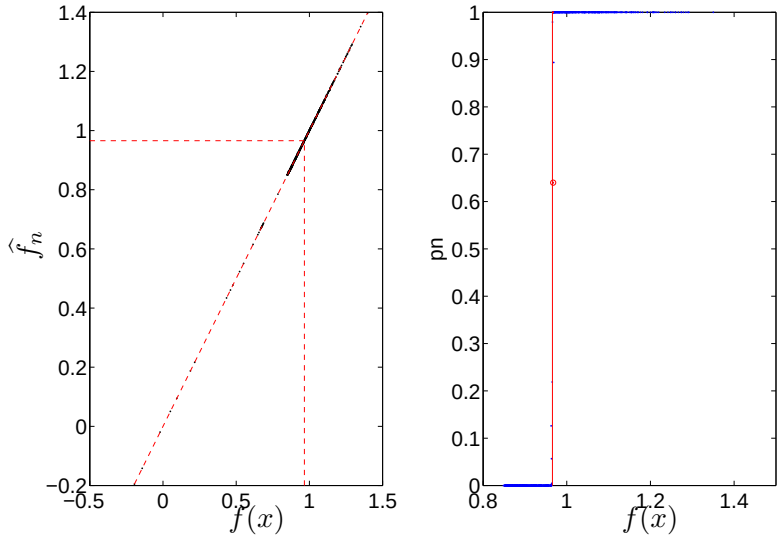


Figure 4.11: Example 3: the true value v.s. prediction and the probability of excursion. Stage: $t = 3$.

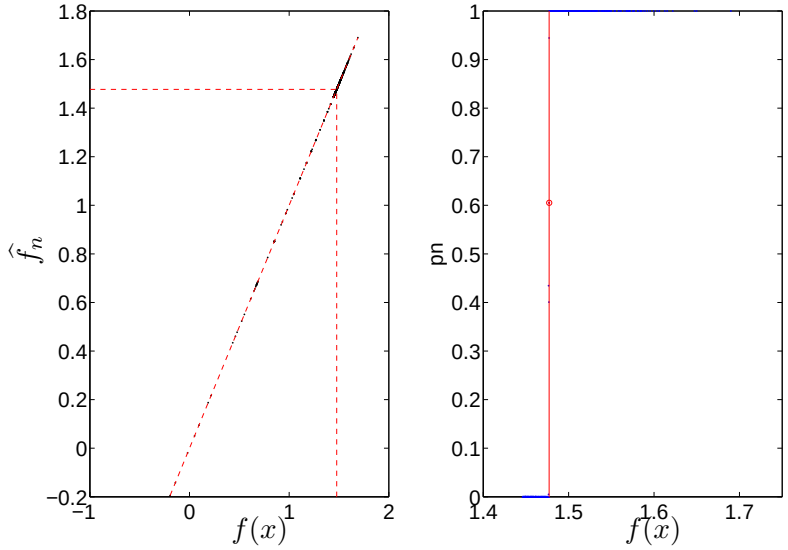


Figure 4.12: Example 3: the true value v.s. prediction and the probability of excursion. Stage: $t = 4$.

pared to the original Subset Simulation algorithm, while keeping a small bias and coefficient of variation. Although ^{2}SMART from FERUM provided an estimator with smaller bias and coefficient of variation than our proposed algorithm, it still requires thousands of evaluations. If we increase the Monte Carlo sample size for our proposed algorithm, the coefficient of variation of our estimator from Bayesian Subset Simulation decreases while keeps a smaller number of evaluations compared to ^{2}SMART algorithm.

Our future work will try to improve further the properties of our algorithm regarding the bias and the variance of the estimator by getting a better understanding of the influence of the parameters m, p_0 and η. Another concern is to run our proposed Bayesian Subset Simulation algorithm with a larger Monte Carlo sample size (e.g., $m = 10^4$ or 10^5) in a reasonable time, as it has been shown that increasing m will decrease the coefficient of variation for the estimator. Chevalier et al. (2011) proposed fast implementations for the SUR strategies which are able to significantly speed up the algorithm. We shall also test and validate the approach on more challenging examples.

Chapter 5

Conclusions and Perspectives

Summary and main contributions

The use of Gaussian processes prior modeling (or kriging prediction) is very efficient and computationally attractive in the machine learning, design of engineering systems, reliability analysis and optimization. The problem of reliability assessment using a surrogate model is a challenging topic and also crucial to the success of the design strategies.

The work presented in this thesis aimed at solving the problem of estimating a probability of failure for a complex system, the model of which corresponds to an expensive-to-evaluate function. There are two primary objectives of this thesis:

1) Propose a Bayesian decision-theoretic framework and use a cheap surrogate model to substitute the expensive-to-evaluate function; develop several sequential criteria to adaptively choose the design of experiments within a limited budget;

2) Explore a new algorithm which combines Bayesian theory and Subset Simulation for the estimation of small probabilities of failure.

First, it gave a quite exhaustive review of the reliability methods in literature in Chapter 2. A summary of reliability methods was first presented. Among all the different categories of methods, FORM/SORM are widely used in reliability analysis but they do not provide information on the estimator; Monte Carlo methods try to reduce the variance of the estimator, however, they need a large number of evaluations in order to obtain an acceptable estimator. During the past decades, surrogate modeling has been gaining more and more attentions. The Gaussian processes prior modeling was revealed as a useful surrogate modeling technique for substituting the expensive-to-evaluate functions. Besides its calculation efficiency, it could provide a local measurement of its estimation accuracy in a closed form in

a probabilistic framework, based on which we can derive a determined confidence intervals on the predictions. The ease of building the kriging predictor also allowed sequentially refine of the surrogate model according to different design criteria, such as reducing the Bayesian risk of the estimator in SUR strategy, or reducing the target region IMSE.

Sequential design of computer experiments

Then, it followed one of the major contribution in this thesis: it investigated a synthetic framework for the sequential design of computer experiments to estimate a probability of failure in Chapter 3. The optimal solution under a Bayesian risk framework is impractical. In this thesis, four different versions of *stepwise uncertainty reduction* (SUR) strategies are derived from a sub-optimal view. Compared to other strategies, SUR strategies have higher complexity in implementation and demand more computation resources. In the meantime, SUR strategies perform better concerning the number of evaluations in our experiments. The choice of an appropriate strategy depends on the specific requirement of the problem.

Bayesian Subset Simulation

Last but not the least, when the probability of failure is very small, it is natural to resort to sequential Monte Carlo methods or Subset Simulation algorithm. However, in the cases when only expensive-to-evaluate functions are available, the required number of evaluations remains still too high. The need to increase simulation efficiency leads to another contribution of this thesis, and a new algorithm called Bayesian Subset Simulation was developed in Chapter 4. The Bayesian theory provides a guidance to generate samples in the region of more interests at each stage by a posterior distribution. Such samples were then generated by the Markov Chain Monte Carlo sampling approach. Although the adaptive threshold algorithm was known to introduce bias in the estimator, numerical experiments have shown that this bias is negligible. Thanks to the use of surrogate modeling technique, the new algorithm outperforms original Subset Simulation algorithm and ^{2}SMART in the section of numerical results.

Perspectives

The use of kriging predictor has found great applications in reliability analysis, especially in the case when the performance function is expensive-to-evaluate. Even though this thesis has investigated the specific problem of reliability assessments and provided a solution for rare probability failure estimation, there are some aspects left to be cleared and future studies to be continued.

Issues related to Gaussian processes prior modeling

First of all, it has been shown that under a prescribed budget N, the success of sequential strategies depends on our capacity to choose an appropriate prior for the performance function. Although in all of our work, Gaussian process priors are used which assumed that the input variable is in low dimension and the performance function f tends to be smooth, it might be inappropriate in a lot engineering applications. When f is discontinued or smooth in general but very rough in some regions, we may need to resort to more complex priors. At this point, we are unable to give an instruction on which kind of prior is the best for any kind of applications. A possible solution is to choose non-stationary Gaussian processes, while the number of parameters will be increased. Gramacy and Le Digabel (2011) proposed an adaptive algorithm with treed Gaussian process surrogate models to locally manage non-smooth optimization problems under constraints.

Difficulties related to high-dimensional problems

Secondly, in this thesis, the input space $\mathbb{X}$ is assumed to be not too high-dimensional, the approaches used such as maximum likelihood (ML) or restricted maximum likelihood (REML) are likely to give correct results. However, as dimension size increases, difficulties will arise. In such cases, there is no guarantee that the likelihood related methods could work well any more. In particular, most strategies try to select points concentrated in some local region (near the contour of the failure region in our applications). The difficulties imposed by such behavior are two sides: 1) the correlation matrix constructed is ill-conditioned and will bring out mathematical inaccuracies in the inversion procedure in the computer calculation; 2) the likelihood function becomes inappropriate and the maximum likelihood algorithm will become not applicable.

The special issue that concerns is the choice of the initial design n_0. As we discussed before that the suggested number of initial design is five times of the input dimension $n_0 = 5d$. When d is very large, is this criterion still a good choice? In order to obtain a good estimation of the parameters, a large number of evaluations are needed for ML or REML algorithms, which will violate our purpose of budget saving. Another difficulty in high dimension problems is the heavy computational expenses. Lizotte et al. (2011) improved the performance of maximum likelihood estimation using maximum a posteriori (MAP) when initial design n_0 is small.

One natural idea is to remove redundant variables by applying a sensitivity analysis before the reliability analysis, however, a traditional sensitivity analysis needs a large number of evaluations. It is possible to apply both sensitivity and reliability analysis at the same time and remove those variables which have few contribution during the simulation. Another possible solution might resort to substitute the maximum likelihood algorithm with a fully Bayesian optimization for parameters estimation. A prior should be set in advance. The parameters will be then calculated from multiple integration of a posterior probability distribution. Markov Chain Monte Carlo method is needed to approximate the multiple integration (see, e.g., Benassi et al., 2012). Due to the limit of time, we could not realize fully Bayesian approach in this thesis, the work will be left for our future research.

Applications to a broader scope of problems

Thirdly, the problems studied in this work exclusively aimed at the design of computer experiments to estimate a probability of failure. We worked directly on the computer codes f which strongly simplified real physical systems. There are more aspects to be considered in the reliability analysis area though. Further investigations are required in order to apply our proposed algorithms to the real world applications.

It is well known that there is a strong link between the problem of global optimization and probability failure estimation. It is in the author's belief that sequential strategies in this thesis can be easily modified and applied to both constrained and unconstrained global optimization problems. Indeed, the famous expected improvement (EI) criterion can be interpreted as looking for the location of the minimum value of a probability density function, such as the famous

efficient global optimization (EGO) algorithm. Instead of looking for a set of failure region Γ, an optimization problem tries to identify the maximum value of the function. The strategy of targeting a sequence of intermediate thresholds to approach a small probability of failure can be easily transformed to sequentially direct the exploration for a global optimization point, such as the fully Bayesian expected improvement criterion in Benassi et al. (2011). Sequential Monte Carlo method is needed to update particles. Rubinstein and Kroese (2004) proposed a cross-entropy method for optimization, Monte Carlo simulation and machine learning. It is also worth to mention that the so called quantile estimation shares a lot in common with probability failure estimation.

Bibliography

H. L. Abdel-Malek and A. K. S. O. Hassan. The ellipsoidal technique for design centering and region approximation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 10(8):1006–1014, 1991. (Cited on page 11.)

G. L. Ang, A. H. S. Ang, and W. H. Tang. Kernel method in importance sampling density estimation. In *Structural Safety and Reliability*, pages 1193–1200. ASCE, 1990. (Cited on page 26.)

A. Arnaud, J. Bect, M. Couplet, A. Pasanisi, and E. Vazquez. Évaluation d'un risque d'inondation fluviale par planification séquentielle d'expériences. In *42èmes Journées de Statistique*, 2010. (Cited on page 59.)

S. K. Au and J. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probab. Engrg. Mechan.*, 16(4):263–277, 2001. (Cited on pages 27, 28, 30, 46, 76, 87, 88, 96, 101 and 107.)

S. K. Au and J. L. Beck. A new adaptive importance sampling scheme for reliability calculations. *Structural Safety*, 21(2):135–158, 1999. (Cited on page 26.)

S. K. Au and J. L. Beck. Subset simulation and its application to seismic risk based on dynamic analysis. *Journal of Engineering Mechanics*, 129:901, 2003. (Cited on page 28.)

Y. Auffray, P. Barbillon, and J. M. Marin. Estimation of rare events probabilities in computer experiments. *Arxiv preprint arXiv:1105.0871*, 2011. (Cited on pages 89, 90 and 91.)

A. Basudhar and S. Missoum. Adaptive explicit decision functions for probabilistic design and optimization using support vector machines. *Computers & Structures*, 86(19-20):1904–1917, 2008. (Cited on page 35.)

M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu. A framework for validation of computer models. *Technometrics*, 49(2):138–154, 2007. (Cited on page 47.)

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*, pages 1–21, 2010. (Cited on pages 99 and 111.)

R. Benassi, J. Bect, and E. Vazquez. Robust gaussian process-based global optimization using a fully bayesian expected improvement criterion. *Learning and Intelligent Optimization*, pages 176–190, 2011. (Cited on page 119.)

R. Benassi, J. Bect, and E. Vazquez. Bayesian optimization using sequential monte carlo. In *6th International Conference on Learning and Intelligent Optimization (LION6), Paris, France*, 2012. (Cited on page 118.)

D. A. Berry and B. Fristedt. *Bandit problems: sequential allocation of experiments*. Chapman & Hall, 1985. (Cited on page 49.)

D. P. Bertsekas. *Dynamic programming and optimal control vol. 1*. Athena Scientific, 1995. (Cited on pages 49 and 50.)

B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland. Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal*, 46(10):2459–2468, 2008. (Cited on pages 39, 47, 53, 59, 65, 66, 67 and 75.)

P. Bjerager. On computational methods for structural reliability analysis. *Structural Safety*, 9:76–96, 1990. (Cited on pages 16 and 46.)

G. Blatman. *Adaptive sparse polynomial chaos expansions for uncertainty propogation and sensitivity analysis*. PhD thesis, Université BLAISE PASCAL - Clermont II, 2009. (Cited on page 34.)

M. Bolic, P. M. Djurić, and S. Hong. New resampling algorithms for particle filters. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, volume 2, pages 589–592, 2003. (Cited on page 96.)

A. Borri and E. Speranzini. Structural reliability analysis using a standard deterministic finite element code. *Structural Safety*, 19(4):361–382, 1997. (Cited on pages 67 and 99.)

Z. I. Botev and D. P. Kroese. Efficient monte carlo simulation via the generalized splitting method. *Statistics and Computing*, pages 1–16, 2011. (Cited on page 28.)

J.-M. Bourinet. Ferum 4.1 user's guide, 2010. URL `http://www.ifma.fr/webdav/site/.../FERUM/FERUM4.1_Users_Guide.pdf`. (Cited on page 101.)

J. M. Bourinet, F. Deheeger, and M. Lemaire. Assessing small failure probabilities by combined subset simulation and support vector machines. *Structural Safety*, 2011. (Cited on pages 92 and 101.)

G. E. P. Box and N. R. Draper. *Empirical Model-Building and Response Surfaces*. Wiley, 1987. (Cited on page 57.)

K. Breitung. Asymptotic approximations for multinormal integrals. *Journal of Engineering Mechanics*, 110:357, 1984. (Cited on pages 11 and 17.)

C. G. Bucher. Adaptive sampling–an iterative fast monte carlo procedure. *Structural Safety*, 5(2):119–126, 1988. (Cited on page 25.)

C. G. Bucher and U. Bourgund. A fast and efficient response surface approach for structural reliability problems. *Structural Safety*, 7(1):57–66, 1990. (Cited on pages 33, 46 and 109.)

F. Cérou, P. Del Moral, T. Furon, and A. Guyader. Sequential monte carlo for rare event estimation. *Statistics and Computing*, pages 1–14, 2011. (Cited on pages 28, 30, 88, 96 and 98.)

C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, Y. Richet, et al. Fast kriging-based stepwise uncertainty reduction with application to the identification of an excursion set, 2011. URL `HAL:hal-00641108,version1`. (Cited on page 114.)

J. P. Chilès and P. Delfiner. *Geostatistics: Modeling Spatial Uncertainty*. Wiley, New York, 1999. (Cited on pages 38, 51, 52 and 53.)

C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Amer. Statist. Assoc.*, 86(416):953–963, 1991. (Cited on pages 4, 47 and 51.)

J. P. De Gyvez and D. K. Pradhan. *Integrated circuit manufacturability: the art of process and design integration*. Wiley-IEEE Press, 1998. (Cited on page 11.)

F. Deheeger. *Couplage mécano-fiabiliste : ^{2}SMART – méthodologie d'apprentissage stochastique en fiabilité*. PhD thesis, Université Blaise Pascal – Clermont II, 2008. (Cited on pages 59, 67 and 99.)

F. Deheeger and M. Lemaire. Reliability analysis using support vector machine classification. In *Proceedings of the 3rd International ASRANet Colloquium, Glasgow, UK, July 10-12*, 2006. (Cited on pages 35 and 36.)

F. Deheeger and M. Lemaire. Support vector machine for efficient subset simulations: ^{2}SMART method. In J. Kanda, T. Takada, and H. Furuta, editors, *10th International Conference on Application of Statistics and Probability in Civil Engineering*, Proceedings and Monographs in Engineering, Water and Earth Sciences, pages 259–260. Taylor & Francis, 2007. (Cited on pages 54, 59, 89, 92 and 101.)

P. Del Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer Verlag, 2004. (Cited on page 30.)

P. Del Moral, A. Doucet, and A. Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006. (Cited on pages 95 and 96.)

O. Ditlevsen and P. Bjerager. Plastic reliability analysis by directional simulation. *Journal of Engineering Mechanics*, 115:1347, 1989. (Cited on page 20.)

O. Ditlevsen, P. Bjerager, R. Olesen, and A. M. Hasofer. Directional simulation in gaussian processes. *Probabilistic Engineering Mechanics*, 3(4):207–217, 1988. (Cited on page 20.)

R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 64–69, 2005. (Cited on page 96.)

R. Douc and É. Moulines. Limit theorems for weighted samples with applications to sequential monte carlo methods. *The Annals of Statistics*, 36(5):2344–2376, 2008. (Cited on page 95.)

V. Dubourg. *Adaptive surrogate models for reliability analysis and reliability-based design optimization*. PhD thesis, Université Blaise Pascal, Clermont II, 2011. (Cited on page 92.)

V. Dubourg, F. Deheeger, and B. Sudret. Metamodel-based importance sampling for the simulation of rare events. In *11th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP 11)*, 2011a. (Cited on pages 89, 90, 91 and 95.)

V. Dubourg, F. Deheeger, and B. Sudret. Metamodel-based importance sampling for structural reliability analysis. *Preprint submitted to Probabilistic Engineering Mechanics*, 2011b. (Cited on page 92.)

W. L. Dunn and J. K. Shultis. *Exploring Monte Carlo Methods*. Elsevier Science Ltd, 2011. (Cited on page 21.)

B. Echard, N. Gayton, and M. Lemaire. Kriging-based Monte Carlo simulation to compute the probability of failure efficiently: AK-MCS method. In *6èmes Journées Nationales de Fiabilité, 24–26 mars, Toulouse, France*, 2010a. (Cited on pages 39, 47, 59, 65, 67 and 99.)

B. Echard, N. Gayton, and M. Lemaire. Structural reliability assessment using kriging metamodel and active learning. In *IFIP WG 7.5 Working Conference on Reliability and Optimization of Structural Systems*, 2010b. (Cited on pages 39, 47, 59, 65, 67 and 99.)

K. El-Tawil, M. Lemaire, and J. P. Muzeau. Reliability method to solve mechanical problems with implicit limit state. *Rackwitz R, Thoft-Christensen P. Reliability and optimization of structural systems*, 91:181–190, 1991. (Cited on page 33.)

L. Faravelli. Response-surface approach for reliability analysis. *Journal of Engineering Mechanics*, 115:2763, 1989. (Cited on page 33.)

P. Feldmann and S. W. Director. Integrated circuit quality optimization using surface integrals. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 12(12):1868–1879, 1993. (Cited on page 20.)

F. Fleuret and D. Geman. Graded learning for object detection. In *Proceedings of the workshop on Statistical and Computational Theories of Vision of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR/SCTV)*, 1999. (Cited on page 47.)

P. I. Frazier, W. B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47 (5):2410–2439, 2008. (Cited on page 55.)

N. Gayton, J. M. Bourinet, and M. Lemaire. Cq2rs: a new statistical approach to the response surface method for reliability analysis. *Structural Safety*, 25(1): 99–121, 2003. (Cited on pages 34, 102 and 109.)

D. Ginsbourger. *Métamodèles multiples pour l'approximation et l'optimisation de fonctions numériques multivariables*. PhD thesis, Ecole nationale supérieure des Mines de Saint-Etienne, 2009. (Cited on pages 47 and 89.)

D. Ginsbourger, R. Le Riche, and L. Carraro. Kriging is well-suited to parallelize optimization. In L. M. Hiot, Y. S. Ong, Y. Tenne, and C. K. Goh, editors, *Computational Intelligence in Expensive Optimization Problems*, volume 2 of *Adaptation Learning and Optimization*, pages 131–162. Springer, 2010. (Cited on page 49.)

P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. Multilevel splitting for estimating rare event probabilities. *Operations Research*, pages 585–600, 1999. (Cited on page 27.)

R. B. Gramacy and S. Le Digabel. The mesh adaptive direct search algorithm with treed gaussian process surrogates, 2011. URL `http://www.gerad.ca/fichiers/cahiers/G-2011-37.pdf`. (Cited on page 117.)

M. S. Handcock and M. L. Stein. A bayesian analysis of kriging. *Technometrics*, 35(4):403–410, 1993. (Cited on page 53.)

A. M. Hasofer and N. C. Lind. *An exact and invariant first-order reliability format*. University of Waterloo, Solid Mechanics Division, 1973. (Cited on pages 15 and 16.)

M. Hohenbichler, S. Gollwitzer, W. Kruse, and R. Rackwitz. New light on first-and second-order reliability methods. *Structural Safety*, 4(4):267–284, 1987. (Cited on pages 11 and 17.)

J. D. Hol, T. B. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *IEEE Workshop on Nonlinear Statistical Signal Processing Workshop*, pages 79–82, 2006. (Cited on page 96.)

K. Hornik. The R FAQ, 2010. URL `http://CRAN.R-project.org/doc/FAQ/R-FAQ.html`. ISBN 3-900051-08-9. (Cited on page 80.)

J. E. Hurtado. *Structural reliability: statistical learning perspectives*, volume 17. Springer Verlag, 2004a. (Cited on pages 35 and 92.)

J. E. Hurtado. An examination of methods for approximating implicit limit state functions from the viewpoint of statistical learning theory. *Structural Safety*, 26 (3):271–293, 2004b. (Cited on page 59.)

J. E. Hurtado. Filtered importance sampling with support vector margin: A powerful method for structural reliability analysis. *Structural Safety*, 29(1):2–15, 2007. (Cited on pages 35 and 59.)

J. E. Hurtado and D. A. Alvarez. Reliability assessment of structural systems using neural networks. In *Proc. European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS*, 2000. (Cited on page 34.)

D. R. Jones, M. Schonlau, and J. William. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998. (Cited on pages 5, 47, 55 and 67.)

A. Karamchandani, P. Bjerager, and C. A. Cornell. Adaptive importance sampling. In *Structural Safety and Reliability*, pages 855–862. ASCE, 1989. (Cited on page 26.)

I. Kaymaz and C. A. McMahon. A response surface method based on weighted regression for structural reliability analysis. *Probabilistic Engineering Mechanics*, 20(1):11–17, 2005. (Cited on page 33.)

M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(3):425–464, 2001. (Cited on page 53.)

R. P. Kennedy, C. A. Cornell, R. D. Campbell, S. Kaplan, and H. F. Perla. Probabilistic seismic safety study of an existing nuclear power plant. *Nuclear Engineering and Design*, 59(2):315–338, 1980. (Cited on page 1.)

S. H. Kim and S. W. Na. Response surface method using vector projected sampling points. *Structural Safety*, 19(1):3–19, 1997. (Cited on page 33.)

G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Ann. Math. Statist.*, 41(2): 495–502, 1970. ISSN 0003-4851. (Cited on page 53.)

T. Kloek and H. K. Van Dijk. Efficient estimation of income distribution parameters. *Journal of Econometrics*, 8(1):61–74, 1978. (Cited on page 24.)

P. S. Koutsourelakis. Reliability of structures in high dimensions, part ii: Theoretical validation. *Probabilistic Engineering Mechanics*, 19(4):419–423, 2004. (Cited on page 20.)

P. S. Koutsourelakis, H. J. Pradlwarter, and G. I. Schuëller. Reliability of structures in high dimensions, part i: algorithms and applications. *Probabilistic Engineering Mechanics*, 19(4):409–417, 2004. (Cited on page 20.)

D. G. Krige. A statistical approach to some mine valuations and allied problems at the witwatersrand. Master's thesis, University of Witwatersrand, 1951. Unpublished. (Cited on page 37.)

H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *J. Basic Engineering*, 86:97–106, 1964. (Cited on page 51.)

R. Lebrun and A. Dutfoy. A generalization of the nataf transformation to distributions with elliptical copula. *Probabilistic Engineering Mechanics*, 24(2):172–178, 2009a. (Cited on page 15.)

R. Lebrun and A. Dutfoy. Do rosenblatt and nataf isoprobabilistic transformations really differ? *Probabilistic Engineering Mechanics*, 24(4):577–584, 2009b. (Cited on page 15.)

H. Li, Z. Lü, and Z. Yue. Support vector machine for structural reliability analysis. *Applied Mathematics and Mechanics*, 27(10):1295–1303, 2006. (Cited on page 35.)

L. Li, J. Bect, and E. Vazquez. A numerical comparison of kriging-based sequential strategies for estimating a probability of failure. In *11th International Conference on Applications of Statistics and Probability Civil Engineering (ICASP 11), Zurich, Switzerland, August 1-4.*, 2011. (Cited on page 99.)

D. J. Lizotte, R. Greiner, and D. Schuurmans. An experimental methodology for response surface optimization methods. *Journal of Global Optimization*, pages 1–38, 2011. (Cited on page 118.)

J. L. Loeppky, J. Sacks, and W. J. Welch. Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51(4):366–376, 2009. (Cited on pages 51 and 102.)

H. O. Madsen, S. Krenk, and N. C. Lind. Methods of structural safety, 1986. (Cited on page 11.)

M. P. Martinez. *Modélisation des comportements extrêmes en ingénierie*. PhD thesis, Université Paris Sud Orsay, 2008. (Cited on page 20.)

G. Matheron. Principles of geostatistics. *Economic Geology*, 58:1246–1266, 1963. (Cited on page 37.)

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. (Cited on page 22.)

R. E. Melchers. Search-based importance sampling. *Structural Safety*, 9(2):117–128, 1990. (Cited on pages 25 and 26.)

R. E. Melchers. Simulation in time-invariant and time-variant reliability problems. *Reliability and Optimization of Structural Systems*, 91:39–82, 1991. (Cited on page 26.)

R. E. Melchers. Structural system reliability assessment using directional simulation. *Structural Safety*, 16(1-2):23–37, 1994. (Cited on page 20.)

J. Mockus. *Bayesian Approach to Global Optimization. Theory and Applications*. Kluwer Academic Publisher, Dordrecht, 1989. (Cited on pages 47, 49 and 55.)

J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. In L. Dixon and E. G. Szego, editors, *Towards Global Optimization*, volume 2, pages 117–129. Elsevier, 1978. (Cited on pages 47, 55 and 67.)

A. Nataf. Dtermination des distribution dont les marges sont donnees. *Comptes rendus de l'academie des sciences*, 225:42–43, 1962. (Cited on page 15.)

R. M. Neal. Slice sampling. *Annals of Statistics*, pages 705–741, 2003. (Cited on page 91.)

J. Oakley. Estimating percentiles of uncertain computer code outputs. *J. Roy. Statist. Soc. Ser. C*, 53(1):83–93, 2004. (Cited on pages 47 and 64.)

J. Oakley and A. O'Hagan. Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, 89(4), 2002. (Cited on pages 37 and 47.)

J. E. Oakley and A. O'Hagan. Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, 2004. (Cited on page 47.)

J. Ogrodzki and M. A. Styblinski. Optimal tolerancing, centering and yield optimization by one-dimensional orthogonal search (odos) technique. In *Proceeding of Europe Conference on Circuits Theory and Design*, volume 2, pages 480–485, 1980. (Cited on page 13.)

A. O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42, 1978. (Cited on page 53.)

A. Olsson, G. Sandberg, and D. Dahlblom. On latin hypercube sampling for structural reliability analysis. *Structural Safety*, 25(1):47–68, 2003. (Cited on page 23.)

A. B. Owen. Monte carlo variance of scrambled net quadrature. *SIAM Journal on Numerical Analysis*, pages 1884–1910, 1997. (Cited on page 22.)

M. Papadrakakis and N. D. Lagaros. Reliability-based structural optimization using neural networks and monte carlo simulation. *Computer Methods in Applied Mechanics and Engineering*, 191(32):3491–3507, 2002. (Cited on page 34.)

E. Parzen. An approach to time series analysis. *Ann. Math. Stat.*, 32:951–989, 1962. (Cited on page 51.)

R. Paulo. Default priors for gaussian processes. *Annals of Statistics*, 33(2):556–582, 2005. (Cited on page 53.)

V. Picheny and D. Ginsbourger. KrigInv : Kriging-based inversion for deterministic and noisy computer experiments, version 1.1, 2011. URL `http://cran.r-project.org/web/packages/KrigInv`. (Cited on page 80.)

V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, and N.-H. Kim. Adaptive designs of experiments for accurate approximation of target regions, 2010. (Cited on pages 39, 47, 53, 57, 59 and 64.)

M. Piera-Martinez. *Modélisation des comportements extrêmes en ingénierie*. PhD thesis, Université Paris Sud - Paris XI, 09 2008. (Cited on pages 35 and 57.)

M. Piera-Martinez, E. Vazquez, E. Walter, G. Fleury, et al. Rkhs classification for multivariate extreme-value analysis. In *7th IASC - Statistics for Data Mining, Learning and Knowledge Extraction*, 2007. (Cited on page 35.)

H. J. Pradlwarter, G. I. Schuëller, P. S. Koutsourelakis, and D. C. Charmpis. Application of line sampling simulation method to reliability benchmark problems. *Structural Safety*, 29(3):208 – 221, 2007. A Benchmark Study on Reliability in High Dimensions. (Cited on page 46.)

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C. The Art of Scientific Computing (Second Edition)*. Cambridge University Press, 1992. (Cited on page 60.)

R. Rackwitz and B. Flessler. Structural reliability under combined random load sequences. *Compters & Structures*, 9(5):489–494, 1978. (Cited on page 16.)

M. R. Rajashekhar and B. R. Ellingwood. A new look at the response surface approach for reliability analysis. *Structural Safety*, 12(3):205–220, 1993a. (Cited on page 46.)

M. R. Rajashekhar and B. R. Ellingwood. A new look at the response surface approach for reliability analysis. *Structural Safety*, 12(3):205–220, 1993b. (Cited on pages 102 and 109.)

P. Ranjan, D. Bingham, and G. Michailidis. Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4):527–541, 2008. (Cited on pages 39, 47, 53, 59, 65, 66, 67 and 75.)

M. Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952. (Cited on page 15.)

R. Y. Rubinstein and D. P. Kroese. *The Cross-Entropy Method*. Springer, 2004. (Cited on pages 46 and 119.)

R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*, volume 707. Wiley-interscience, 2008. (Cited on page 24.)

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989. (Cited on pages 4, 5, 37, 47, 51, 57 and 64.)

G. M. Salmon and D. N. D. Hartford. Risk analysis for dam safety. In *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, volume 32, pages 284A–284A. Elsevier, 1995. (Cited on page 1.)

T. J. Santner, B. J. Williams, and W. Notz. *The design and analysis of computer experiments*. Springer Verlag, New York, 2003a. (Cited on pages 4, 89 and 96.)

T. J. Santner, B. J. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer Verlag, 2003b. (Cited on pages 51 and 52.)

G. I. Schuëller, H. J. Pradlwarter, and P. S. Koutsourelakis. A critical appraisal of reliability estimation procedures for high dimensions. *Probabilistic Engineering Mechanics*, 19(4):463–474, 2004. (Cited on page 20.)

L. Schueremans. *Probabilistic evaluation of structural unreinforced masonry*. PhD thesis, Catholic University of Leuven, 2001. (Cited on pages 1, 67 and 99.)

L. Schueremans and D. Van Gemert. Benefit of splines and neural networks in simulation based structural reliability analysis. *Structural safety*, 27(3):246–261, 2005. (Cited on pages 67 and 99.)

M. Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, pages 143–151, 1987. (Cited on pages 22 and 23.)

M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New York, 1999. (Cited on pages 72, 81 and 99.)

L. Tvedt. Second-order reliability by an exact integral. In *2nd IFIP Working Conference on Reliability and Optimization on Structural Systems. Springer-Verlag, Germany*, 1988. (Cited on page 17.)

E. Vazquez and J. Bect. A sequential Bayesian algorithm to estimate a probability of failure. In *Proceedings of the 15th IFAC Symposium on System Identification, SYSID 2009 15th IFAC Symposium on System Identification, SYSID 2009*, Saint-Malo France, 2009. (Cited on pages 47, 57 and 87.)

E. Vazquez and M. Piera-Martinez. Estimation du volume des ensembles d'excursion d'un processus gaussien par krigeage intrinsèque. In *39ème Journées de Statistiques Conférence Journée de Statistiques*, Angers France, 2007. (Cited on pages 47 and 57.)

E. M. Vestrup. *The Theory of Measures and Integration*. Wiley, 2003. (Cited on page 56.)

J. Villemonteix. *Optimisation de fonctions coûteuses*. PhD thesis, Université Paris-Sud XI, Faculté des Sciences d'Orsay, 2008. (Cited on page 47.)

J. Villemonteix, E. Vazquez, and E. Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009. (Cited on pages 47, 55 and 89.)

P. H. Waarts. *Structural reliability using finite element methods: an appraisal of DARS*. PhD thesis, Delft University of Technology, 2000. (Cited on pages 67 and 99.)

W. J. Welch, J. Buck, R. J.and Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris. Screening, predicting and computer experiments. *Technometrics*, 34:15–25, 1992. (Cited on pages 4, 47 and 51.)

A. M. Yaglom. *Correlation Theory of Stationary and Related Random Functions I: Basic Results*. Springer Series in Statistics. Springer-Verlag, New york, 1986. (Cited on page 81.)

J. C. Zhang and M. A. Styblinski. *Yield and variability optimization of integrated circuits*. Kluwer Academic Pub, 1995. (Cited on page 13.)

Y. Zhang and Q. Liu. Reliability-based design of automobile components. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 216(6):455–471, 2002. (Cited on page 1.)

Y. Zheng and P. K. Das. Improved response surface method and its application to stiffened plate reliability analysis. *Engineering structures*, 22(5):544–551, 2000. (Cited on page 33.)