



HAL
open science

Bayesian subset simulation

Julien Bect, Ling Li, Emmanuel Vazquez

► **To cite this version:**

| Julien Bect, Ling Li, Emmanuel Vazquez. Bayesian subset simulation. 2016. hal-01253706v1

HAL Id: hal-01253706

<https://hal.science/hal-01253706v1>

Preprint submitted on 11 Jan 2016 (v1), last revised 23 Aug 2017 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

BAYESIAN SUBSET SIMULATION

BY JULIEN BECT, LING LI AND EMMANUEL VAZQUEZ

*Laboratoire des Signaux et Systèmes
CentraleSupélec / CNRS / Université Paris-Sud
Gif-sur-Yvette, France*

We consider the problem of estimating the volume α of the excursion set of a function $f : \mathbb{X} \subseteq \mathbb{R}^d \rightarrow \mathbb{R}$ above a given threshold, under a given probability measure on \mathbb{X} . In this article, we combine the popular subset simulation algorithm (Au and Beck, Probab. Eng. Mech. 2001) and our sequential Bayesian approach for the estimation of a probability of failure (Bect, Ginsbourger, Li, Picheny and Vazquez, Stat. Comput. 2012). This makes it possible to estimate α when the number of evaluations of f is very limited and α is very small. The resulting algorithm is called Bayesian subset simulation (BSS). A key idea, as in the subset simulation algorithm, is to estimate the probabilities of a sequence of excursion sets of f above intermediate thresholds, using a sequential Monte Carlo (SMC) approach. A Gaussian process prior on f is used to define the sequence of densities targeted by the SMC algorithm, and drive the selection of evaluation points of f to estimate the intermediate probabilities. Adaptive procedures are proposed to determine the intermediate thresholds and the number of evaluations to be carried out at each stage of the algorithm. Numerical experiments illustrate that BSS achieves significant savings in the number of function evaluations with respect to other Monte Carlo approaches.

1. Introduction. Probabilistic reliability analysis has become over the last thirty years an essential part of the engineer's toolbox (see, e.g., Melchers, 1999; De Rocquigny, Devictor and Tarantola, 2008a; O'Connor and Kleyner, 2012). One of the central problems in probabilistic reliability analysis is the computation of the probability of failure

$$(1) \quad \alpha = \int_{\mathbb{X}} \mathbb{1}_{f \leq 0} d\mathbb{P}_{\mathbb{X}}$$

of a system (or a component in a multicomponent system; see, e.g., Rausand and Hoyland, 2004), where $\mathbb{P}_{\mathbb{X}}$ is a probability measure over some measurable space $(\mathbb{X}, \mathcal{B})$ representing all possible sources of uncertainty acting on the

Keywords and phrases: Probability of failure, Computer experiments, Sequential design, Gaussian process, Stepwise uncertainty reduction, Sequential Monte Carlo

system—both epistemic and aleatory—and $f : \mathbb{X} \rightarrow \mathbb{R}$ is the so-called *limit-state function*, such that f takes positive values when the system behaves reliably, and negative values when the system behaves unreliably, or fails. It is assumed in this article that \mathbb{X} is a subset of \mathbb{R}^d —in other words, we consider *static* reliability problems, where all uncertain factors can be described as a d -dimensional random vector. Numerous examples of applications that fall into this category can be found in the literature (see, for instance, [Waarts, 2000](#); [Jonkman, Kok and Vrijling, 2008](#); [De Rocquigny, Devictor and Tarantola, 2008b,c](#); [Bayarri et al., 2009](#); [Zio and Pedroni, 2009](#); [Garcia, 2010](#)).

Two major difficulties usually preclude a brute-force Monte Carlo (MC) approach, that is, using the estimator

$$\hat{\alpha}^{\text{MC}} = \frac{1}{m} \sum_{i=1}^m \mathbb{1}_{f(X_i) \leq 0}, \quad X_i \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}_{\mathbb{X}},$$

which requires m evaluations of f . First, the evaluation of f for a given $x \in \mathbb{X}$ often relies on one or several complex computer programs (e.g., partial differential equation solvers) that take a long time to run. Second, in many applications, the failure region $\Gamma = \{x \in \mathbb{X} \mid f(x) \leq 0\}$ is a *rare event* under the probability $\mathbb{P}_{\mathbb{X}}$; that is, the probability of failure $\alpha = \mathbb{P}_{\mathbb{X}}(\Gamma)$ is small. When α is small, the standard deviation of $\hat{\alpha}^{\text{MC}}$ is approximately $\sqrt{\alpha/m}$. To estimate α by MC with a standard deviation of 0.1α thus requires approximately $100/\alpha$ evaluations of f . As an example, with $\alpha = 10^{-3}$ and 10 minutes per evaluation, this means almost two years of computation time.

The first issue—designing efficient algorithms to estimate α in the case of an expensive-to-evaluate limit-state function—can be seen as a problem of *design and analysis of computer experiments* (see, e.g., [Santner, Williams and Notz, 2003](#)), bearing some similarities to the problem of global optimization (see [Villemonteix, Vazquez and Walter, 2009](#), and references therein). Sequential kriging-based design strategies for the estimation of α have been proposed, and spectacular evaluation savings have been demonstrated on various examples with moderately small α (typically, 10^{-2} or 10^{-3}); see [Bect et al. \(2012\)](#) for a review of fully sequential strategies and [Dubourg, Deheeger and Sudret \(2011\)](#) and [Auffray, Barbillon and Marin \(2014\)](#) for examples of two-stage strategies. The closely related problem of quantile estimation has also been investigated along similar lines ([Oakley, 2004](#); [Cannamela, Garnier and Iooss, 2008](#); [Arnaud et al., 2010](#)).

A key idea to address the second issue—i.e., to estimate a small probability of failure—is to consider a decreasing sequence of events $\Gamma_1 \supset \Gamma_2 \supset \dots \supset \Gamma_T = \Gamma$ such that the conditional probabilities $\mathbb{P}_{\mathbb{X}}(\Gamma_t \mid \Gamma_{t-1})$ are reasonably large, and therefore easier to estimate than α itself. Then, sequential Monte

Carlo simulations (Del Moral, Doucet and Jasra, 2006) can be used to produce estimates \hat{p}_t of the conditional probabilities $P_{\mathbb{X}}(\Gamma_t | \Gamma_{t-1})$, leading to a product-form estimate $\prod_{t=1}^T \hat{p}_t$ for α . This idea was first proposed, for the simulation of static rare events in reliability analysis¹, in a seminal article of Au and Beck (2001). Their algorithm, called *subset simulation*, can be seen as variation on the *importance* (or *multilevel*) *splitting* technique used for the simulation of rare events in Markovian models (see, e.g., Glasserman et al., 1999). It has since then become one of the most popular techniques for the computation of small probabilities of failure, and the theoretical properties of several (most of the times idealized) variants of the algorithm have recently been investigated by several authors (see, e.g., Guyader, Hengartner and Matzner-Løber, 2011; Cérou et al., 2012; Caron et al., 2014). However, because of the direct use of a Monte Carlo estimator for \hat{p}_t at each stage t , the subset simulation algorithm is not applicable when f is expensive to evaluate.

In this article we propose a new algorithm, called *Bayesian subset simulation* (BSS), which tackles both issues at once using ideas from the sequential design of computer experiments and from the literature on sequential Monte Carlo methods. Section 2 reviews the subset simulation algorithm from the point of view of sequential Monte Carlo (SMC) techniques to prepare the ground for the introduction of our new algorithm. Section 3 describes the algorithm itself and Section 4 presents numerical results. Finally, Section 5 concludes the article with a discussion.

2. Subset simulation: a sequential Monte Carlo algorithm. This section recalls the main ideas of the classical subset simulation algorithm (Au and Beck, 2001), which, although not originally presented as such, can be seen as a sequential Monte Carlo sampler (Del Moral, Doucet and Jasra, 2006; Cérou et al., 2012).

2.1. *Idealized subset simulation (with fixed levels and IID sampling).* We consider the problem of estimating the probability α of a rare event Γ of the form $\Gamma = \{x \in \mathbb{X} : f(x) > u\}$, where $u \in \mathbb{R}$ and $f : \mathbb{X} \rightarrow \mathbb{R}$, using pointwise evaluations of f . Note that the limit-state function (see Section 1) can be defined as $x \mapsto u - f(x)$ with our notations. Assuming, for the sake of simplicity, that $P_{\mathbb{X}}$ has a probability density function $\pi_{\mathbb{X}}$ with respect to

¹A very similar algorithm had in fact been proposed earlier by Diaconis and Holmes (1995) for a quite different purpose (estimating the probability of a rare event under the bootstrap distribution).

Lebesgue's measure, we have

$$\alpha = \int_{\mathbb{X}} \mathbf{1}_{f(x) > u} \pi_{\mathbb{X}}(x) dx.$$

The key idea of the subset simulation algorithm is to introduce an increasing (finite) sequence of thresholds $-\infty = u_0 < u_1 < u_2 \cdots < u_T = u$, which determine a corresponding decreasing sequence of subsets:

$$\mathbb{X} = \Gamma_0 \supset \Gamma_1 \supset \cdots \supset \Gamma_T = \Gamma, \quad \Gamma_t := \{x \in \mathbb{X} : f(x) > u_t\},$$

of the input space \mathbb{X} . Let $\alpha_t = \mathbb{P}_{\mathbb{X}}(\Gamma_t)$. The decreasing sequence $(\alpha_t)_{0 \leq t \leq T}$ obeys the recurrence formula

$$(2) \quad \alpha_{t+1} = \alpha_t \mathbb{P}_{\mathbb{X}}(\Gamma_{t+1} | \Gamma_t) = \alpha_t \int \mathbf{1}_{\Gamma_{t+1}}(x) q_t(x) dx,$$

where q_t stands for the truncated density

$$q_t(x) = \frac{\mathbf{1}_{\Gamma_t}(x) \pi_{\mathbb{X}}(x)}{\int \mathbf{1}_{\Gamma_t}(y) \pi_{\mathbb{X}}(y) dy}.$$

The small probability $\alpha = \alpha_T$ can thus be rewritten as a product of conditional probabilities, which are larger (and therefore easier to estimate) than α :

$$\alpha = \prod_{t=1}^T p_t, \quad p_t := \mathbb{P}_{\mathbb{X}}(\Gamma_t | \Gamma_{t-1}).$$

Assume that, for each $t \in \{0, 1, \dots, T-1\}$, a sample $(Y_t^j)_{1 \leq j \leq m}$ of independent and identically distributed (IID) random variables from the truncated density q_t is available. Then, each conditional probability p_t can be estimated by the corresponding Monte-Carlo estimator $\hat{p}_t = \frac{1}{m} \sum_{j=1}^m \mathbf{1}_{\Gamma_t}(Y_{t-1}^j)$, and α can be estimated by the product-form estimator $\hat{\alpha}^{\text{SS}} = \prod_{t=1}^T \hat{p}_t$. By choosing the thresholds u_t in such a way that the conditional probabilities p_t are high, α can be estimated using fewer evaluations of f than what would have been necessary using a simple Monte Carlo approach (see Section 2.4 for a quantitative example).

2.2. Sequential Monte-Carlo simulation techniques. Generating exact IID draws from the densities q_t is usually not possible, at least not efficiently, even if a method to generate IID samples from $q_0 = \pi_{\mathbb{X}}$ is available. Indeed, although the accept-reject algorithm (see, e.g., [Robert and Casella](#),

2004, Section 2.3) could be used in principle, it would be extremely inefficient when t is close to T , that is, when $\mathbb{P}_{\mathbb{X}}\{\Gamma_t\}$ becomes small. This is where sequential Monte-Carlo (SMC) simulation techniques are useful.

Given a sequence $(q_t)_{0 \leq t < T}$ of probability density functions over \mathbb{X} , SMC samplers sequentially generate, for each target density q_t , a weighted sample $\mathbb{Y}_t = ((w_t^j, Y_t^j))_{1 \leq j \leq m}$, where $w_t^j \geq 0$, $\sum_j w_t^j = 1$ and $Y_t^j \in \mathbb{X}$. The random vectors Y_t^j are usually called *particles* in the SMC literature, and the weighted sample \mathbb{Y}_t is said to *target* the distribution q_t . They are, in general, neither independent nor distributed according to q_t , but when the sample size m goes to infinity, their empirical distribution $\mu_t^{(m)} = \sum_{j=1}^m w_t^j \delta_{Y_t^j}$ converges to the target distribution—that is, to the distribution with probability density function q_t —in the sense that

$$\int_{\mathbb{X}} h(x) d\mu_t^{(m)}(x) = \sum_{j=1}^m w_t^j h(Y_t^j) \rightarrow \int_{\mathbb{X}} h(x) q_t(x) dx,$$

for a certain class of integrable functions h .

In practice, each weighted sample \mathbb{Y}_t is generated from the previous one, \mathbb{Y}_{t-1} , using transformations; SMC algorithms are thus expected to be efficient when each density q_t is, in some sense, close to its predecessor density q_{t-1} . The specific transformations that are used in the subset simulation algorithm are described next. The reader is referred to [Del Moral, Doucet and Jasra \(2006\)](#), [Liu \(2008\)](#) and references therein for a broader view of SMC sampling techniques, and to [Douc and Moulines \(2008\)](#) for some theoretical results on the convergence (law of large numbers, central limit theorems) of SMC algorithms.

2.3. Reweight/resample/move. We now describe the reweight/resample/move scheme that is used in the subset simulation algorithm to turn a weighted sample \mathbb{Y}_{t-1} targeting $q_{t-1} \propto \mathbb{1}_{\Gamma_{t-1}} \pi_{\mathbb{X}}$ into a weighted sample \mathbb{Y}_t targeting $q_t \propto \mathbb{1}_{\Gamma_t} \pi_{\mathbb{X}}$. This scheme, used for instance in [Chopin \(2002\)](#), can be seen as a special case of the more general SMC sampler of [Del Moral, Doucet and Jasra \(2006\)](#)².

Assume a weighted sample $\mathbb{Y}_{t-1} = ((w_{t-1}^j, Y_{t-1}^j))_{1 \leq j \leq m}$ targeting q_{t-1} has been obtained at stage $t-1$. The *reweight* step produces a new weighted sample $\mathbb{Y}_{t,0} = ((w_{t,0}^j, Y_{t-1}^j))_{1 \leq j \leq m}$ that targets q_t , by changing only the weights in \mathbb{Y}_{t-1} :

$$w_{t,0}^j \propto \frac{q_t(Y_{t-1}^j)}{q_{t-1}(Y_{t-1}^j)} w_{t-1}^j.$$

²See in particular Section 3.1.1, Remark 1, and Section 3.3.2.3.

The *resample* and *move* steps follow the reweighting step. These steps aim at avoiding the degeneracy of the sequence of weighted samples—i.e., the accumulation of most of the probability mass on a small number of particles with large weights.

The simplest variant of resampling is the *multinomial resampling* scheme. It produces a new weighted sample $\mathbb{Y}_{t,1} = ((w_t^j, Y_{t,1}^j))_{1 \leq j \leq m}$, where the new particles $Y_{t,1}^j$ have equal weights $w_t^j = \frac{1}{m}$, and are independent and identically distributed according to the empirical distribution $\sum_{j=1}^m w_{t,0}^j \delta_{Y_{t-1}^j}$. In this work, we use the slightly more elaborate *residual resampling* scheme (see, e.g., [Liu, 2008](#)), which is known to outperform multinomial resampling ([Douc and Cappé, 2005](#), Section 3.2). As in multinomial resampling, the residual resampling scheme produces a weighted sample with equal weights $w_t^j = \frac{1}{m}$.

The resampling step alone does not prevent degeneracy, since the resulting sample contains copies of the same particles. The move step restores some diversity by moving the particles according to a Markov transition kernel K_t that leaves q_t invariant:

$$\int q_t(x) K_t(x, dx') = q_t(x') dx';$$

for instance, a random-walk Metropolis-Hastings (MH) kernel (see, e.g. [Robert and Casella, 2004](#)).

REMARK 2.1. In the special case of the subset simulation algorithm, all weights are actually equal *before* the reweighting step and, considering the inclusion $\Gamma_t \subset \Gamma_{t-1}$, the reweighting formula takes the form

$$w_{t,0}^j \propto \mathbb{1}_{\Gamma_t}(Y_{t-1}^j).$$

In other words, the particles that are outside the new subset Γ_t are given a zero weight, and the other weights are simply normalized to sum to one. Note also that the resampling step discards particles outside of Γ_t (those with zero weight at the reweighting step).

REMARK 2.2. In the general version of the reweight/resample/move procedure, which will be used in Section 3.3, the resampling step is carried out only when some degeneracy criterion—such as the expected sample size (ESS)—falls below a threshold (see, e.g., [Del Moral, Doucet and Jasra, 2006](#)). Indeed, consecutive densities in the sequence (q_t) sometimes happen to be very similar, in which case the weights do not degenerate as quickly as in the subset simulation algorithm.

2.4. *Practical subset simulation: adaptive thresholds.* It is easy to prove that the subset simulation estimator $\hat{\alpha}^{\text{SS}} = \prod_{t=1}^T \hat{p}_t$ is unbiased. Moreover, C erou et al. (2012, Proposition 3) have proved that it is asymptotically normal in the large-sample-size limit:

$$(3) \quad \sqrt{m} \frac{\hat{\alpha}^{\text{SS}} - \alpha}{\alpha} \xrightarrow[m \rightarrow \infty]{\mathcal{D}} \mathcal{N}(0; \sigma^2)$$

where $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution and

$$(4) \quad \sigma^2 \approx \sum_{t=1}^T \frac{1 - p_t}{p_t},$$

when the MCMC kernel has good mixing properties (see C erou et al.’s article for the exact expression of σ^2). For a given number T of stages, the right-hand side of (4) is minimal when all conditional probabilities are equal; that is, when $p_t = \alpha^{1/T}$.

In practice however, the value of α is of course unknown, and it is not possible to choose the sequence of threshold beforehand in order to make all the conditional probabilities equal. Instead, a value p_0 is chosen—say, $p_0 = 10\%$ —and the thresholds are tuned in such a way that, at each stage t , $\hat{p}_t = p_0$. A summary of the resulting algorithm is provided in Table 1.

Equations (3) and (4) can be used to quantify the number of evaluations of f required to reach a given coefficient of variation with the subset simulation estimator $\hat{\alpha}^{\text{SS}}$. Indeed, in the case where all conditional probabilities are equal, we have

$$(5) \quad \text{var}(\hat{\alpha}^{\text{SS}}/\alpha) \approx \frac{T}{m} \frac{1 - p_0}{p_0}.$$

with $T = \log(\alpha)/\log(p_0)$. For example, take $\alpha = 10^{-6}$. With the simple Monte Carlo estimator, the number of evaluations of f is equal to the sample size m : approximately $n = \delta^{-2} \alpha^{-1} = 10^8$ evaluations are required to achieve a coefficient of variation $\delta = \text{std}(\hat{\alpha}^{\text{MC}})/\alpha = 10\%$. In contrast, with $p_0 = 10\%$, the subset simulation algorithm will complete in $T = \log(\alpha)/\log(p_0) = 6$ stages, thus achieving a coefficient of variation $\delta = \text{std}(\hat{\alpha}^{\text{SS}})/\alpha = 10\%$ with $m = \delta^{-2} T (1 - p_0)/p_0 = 5400$ particles. Assuming that the move step uses only one evaluation of f per particle, the corresponding number of evaluations would be $n = m + (T - 1)(1 - p_0)m = 29700 \ll 10^8$.

TABLE 1
Subset simulation algorithm with adaptive thresholds

Prescribe $m_0 < m$ a fixed number of “succeeding particles”. Set $p_0 = \frac{m_0}{m}$.

1. **Initialization** (stage 0)
 - (a) Generate an m -sample $Y_0^j \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}_{\mathbb{X}}$, $1 \leq j \leq m$, and evaluate $f(Y_0^j)$ for all j .
 - (b) Set $u_0 = -\infty$ and $t = 1$.
2. **Repeat** (stage t)
 - (a) **Threshold adaptation**
 - Compute the $(m - m_0)$ -th order statistic of $(f(Y_{t-1}^j))_{1 \leq j \leq m}$ and call it u_t^0 .
 - If $u_t^0 > u$, set $u_t = u$, $T = t$ and go to the estimation step.
 - Otherwise, set $u_t = u_t^0$, $\Gamma_t = \{x \in \mathbb{X}; f(x) > u_t\}$ and $I_t = \{j: Y_{t-1}^j \in \Gamma_t\}$.
 - (b) **Reweight and resample** – For each $j \in \{1, \dots, m\}$,
 - set $Y_{t,1}^j = Y_{t-1}^j$ if $j \in I_t$,
 - set $Y_{t,1}^j = Y_{t-1}^l$, with l randomly drawn in I_t , otherwise.
 - (c) **Move** – For each $j \notin I_t$,
 - generate $Y_t^j \in \Gamma_t$ according to $Y_t^j \sim K_t(Y_{t,1}^j, \cdot)$, where K_t is a transition kernel that leaves $q_t \propto \mathbb{1}_{\Gamma_t} \pi_{\mathbb{X}}$ invariant. (NB: here, f is evaluated.)
 - (d) Increment t .
3. **Estimation** – Let m_u be the number of particles such that $f(Y_{T-1}^j) > u$. Set

$$\hat{\alpha}^{\text{SS}} = \frac{m_u}{m} p_0^{T-1}.$$

3. Bayesian subset simulation.

3.1. *Bayesian estimation and sequential design of experiment.* Our objective is to build an estimator of α from the evaluations results of f at some points $X_1, X_2, \dots, X_N \in \mathbb{X}$, where N is the total budget of evaluations available for the estimation. In order to design an efficient estimation procedure, by which we mean both the design of experiments and the estimator itself, we adopt a Bayesian approach: from now on, we shall denote by ξ the random process that models our incomplete knowledge about f . For the sake of tractability, we assume as usual that, under the prior probability that we denote as \mathbb{P}_0 , ξ is a Gaussian process (possibly with a linearly parameterized mean, whose parameters are then endowed with a uniform improper prior; see [Bect et al., 2012](#), Section 2.3, for details).

Denote by \mathbb{E}_n (resp. \mathbb{P}_n) the conditional expectation (resp. conditional probability) with respect to $X_1, \xi(X_1), \dots, X_n, \xi(X_n)$, for any $n \leq N$ and assume, as in Section 2, that $\mathbb{P}_{\mathbb{X}}$ has a probability density function $\pi_{\mathbb{X}}$ with respect to Lebesgue's measure. Then, a natural (mean-square optimal) Bayesian estimator of $\alpha = \mathbb{P}_{\mathbb{X}}(\Gamma)$ using n evaluations is the posterior mean

$$(6) \quad \mathbb{E}_n(\alpha) = \mathbb{E}_n\left(\int_{\mathbb{X}} \mathbb{1}_{\xi(x) > u} \pi_{\mathbb{X}}(x) dx\right) = \int_{\mathbb{X}} \tilde{g}_{n,u}(x) \pi_{\mathbb{X}}(x) dx,$$

where $\tilde{g}_{n,u}(x) := \mathbb{E}_n(\mathbb{1}_{\xi(x) > u}) = \mathbb{P}_n(\xi(x) > u)$ is the coverage function of the random set Γ (see, e.g., [Chevalier et al. \(2013a\)](#)). Note that, since ξ is Gaussian, $\tilde{g}_{n,u}(x)$ can be readily computed for any x using kriging (see, e.g., [Bect et al., 2012](#), Section 2.4).

Observe that $\tilde{g}_{n,u} \approx \mathbb{1}_{\Gamma}$ when the available evaluation results are informative enough to classify most input points correctly (with high probability) with respect to u . This suggests that the computation of the right-hand side of (6) should not be carried out using a brute force Monte Carlo approximation, and would benefit from an SMC approach similar to the subset simulation algorithm described in Section 2. Moreover, an SMC approach is also beneficial for the problem of choosing (sequentially) the sampling points X_1, \dots, X_N . Indeed, consider the following *stepwise uncertainty reduction* (SUR) strategy proposed by [Bect et al. \(2012\)](#):

$$(7) \quad X_{n+1} = \underset{x_{n+1} \in \mathbb{X}}{\operatorname{argmin}} J_n(x_{n+1}),$$

$$(8) \quad \begin{aligned} J_n(x_{n+1}) &:= \mathbb{E}_n\left(\mathbb{P}_{\mathbb{X}}\left(\Gamma \Delta \hat{\Gamma}_{n+1,u}\right) \mid X_{n+1} = x_{n+1}\right) \\ &= \int_{\mathbb{X}} \mathbb{E}_n(\tau_{n+1,u}(x) \mid X_{n+1} = x_{n+1}) \pi_{\mathbb{X}}(x) dx. \end{aligned}$$

where $\widehat{\Gamma}_{n,u} = \{x \in \mathbb{X} \mid \tilde{g}_{n,u}(x) > 1/2\}$ and

$$(9) \quad \tau_{n,u}(x) := \mathbf{P}_n(x \in \Gamma \triangle \widehat{\Gamma}_{n,u}) = \min(\tilde{g}_{n,u}(x), 1 - \tilde{g}_{n,u}(x)).$$

For moderately small values of α , it is possible to use a sample from $\mathbf{P}_{\mathbb{X}}$ both for the approximation of the integral in the right-hand side of (8) and for an approximate minimization of J_n (by exhaustive search in the set of sample points). However, this simple Monte Carlo approach would require a very large sample size to be applicable for very small values of α ; a subset-simulation-like SMC approach will now be proposed as a replacement.

3.2. A sequential Monte Carlo approach. Assume that α is small and consider a decreasing sequence of subsets $\mathbb{X} = \Gamma_0 \supset \Gamma_1 \supset \dots \supset \Gamma_T = \Gamma$, where $\Gamma_t = \{x \in \mathbb{X} : f(x) > u_t\}$, as in Section 2. For each $t \leq T$, denote by $\widehat{\alpha}_t^{\mathbf{B}}$ the Bayesian estimator of $\alpha_t = \mathbf{P}_{\mathbb{X}}(\Gamma_t)$ obtained from n_t observations of ξ at points X_1, \dots, X_{n_t} :

$$(10) \quad \widehat{\alpha}_t^{\mathbf{B}} := \mathbf{E}_{n_t}(\alpha_t) = \int_{\mathbb{X}} g_t \, d\mathbf{P}_{\mathbb{X}},$$

where $g_t(x) := \tilde{g}_{n_t, u_t}(x) = \mathbf{P}_{n_t}(\xi(x) > u_t)$.

The main idea of our new algorithm is to use an SMC approach to construct a sequence of approximations $\widehat{\alpha}_t^{\text{BSS}}$ of the Bayesian estimators $\widehat{\alpha}_t^{\mathbf{B}}$, $1 \leq t \leq T$ (as explained earlier, the particles of these SMC approximations will also provide suitable candidate points for the optimization of a sequential design criterion). To this end, consider the sequence of probability density functions q_t defined by

$$(11) \quad q_t(x) := \frac{\pi_{\mathbb{X}}(x) g_t(x)}{\int \pi_{\mathbb{X}}(y) g_t(y) \, dy} = \frac{1}{\widehat{\alpha}_t^{\mathbf{B}}} \pi_{\mathbb{X}}(x) g_t(x).$$

We can write a recurrence equation for the sequence of Bayesian estimators $\widehat{\alpha}_t^{\mathbf{B}}$, similar to that used for the probabilities α_t in (2):

$$(12) \quad \widehat{\alpha}_{t+1}^{\mathbf{B}} = \int g_{t+1}(x) \pi_{\mathbb{X}}(x) \, dx = \widehat{\alpha}_t^{\mathbf{B}} \int \frac{g_{t+1}(x)}{g_t(x)} q_t(x) \, dx.$$

This suggests to construct recursively a sequence of estimators $(\widehat{\alpha}_t^{\text{BSS}})$ using the following relation:

$$(13) \quad \widehat{\alpha}_{t+1}^{\text{BSS}} = \widehat{\alpha}_t^{\text{BSS}} \sum_{j=1}^m w_t^j \frac{g_{t+1}(Y_t^j)}{g_t(Y_t^j)}, \quad 0 \leq t < T,$$

where $(w_t^j, Y_t^j)_{1 \leq j \leq m}$ is a weighted sample of size m targeting q_t (as in Section 2.2) and $\widehat{\alpha}_0^{\text{BSS}} = 1$. The final estimator can be written as:

$$(14) \quad \widehat{\alpha}_T^{\text{BSS}} = \prod_{t=0}^{T-1} \frac{\widehat{\alpha}_{t+1}^{\text{BSS}}}{\widehat{\alpha}_t^{\text{BSS}}} = \prod_{t=0}^{T-1} \sum_{j=1}^m w_t^j \frac{g_{t+1}(Y_t^j)}{g_t(Y_t^j)}.$$

REMARK 3.1. The connection between the proposed algorithm and the original subset simulation algorithm is clear from the similarity between the recurrence relations (2) and (12), and from the use of SMC simulation in both algorithms to construct recursively a product-type estimator of the probability of failure (see also Del Moral, Doucet and Jasra, 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, \dots, q_T also relates to the original subset simulation algorithm. Indeed, note that $q_t(x) \propto \mathbf{E}_{n_t}(\mathbf{1}_{\xi > u_t} \pi_{\mathbb{X}})$, and recall that $\tilde{q}_t \propto \mathbf{1}_{\xi > u_t} \pi_{\mathbb{X}}$ is the distribution used in the subset simulation algorithm at stage t . (This choice of instrumental density is also used by Dubourg, Deheeger and Sudret (2013, 2011) in the context of a two-stage kriging-based adaptive importance sampling algorithm. This is indeed a quite natural choice, since $\tilde{q}_t \propto \mathbf{1}_{\xi > u_t} \pi_{\mathbb{X}}$ is the optimal instrumental density for the estimation of α_t by importance sampling.)

3.3. *The Bayesian subset simulation (BSS) algorithm.* The algorithm consists of a sequence of stages (or iterations). For the sake of clarity, assume first that the sequence of thresholds (u_t) is given. Then, each stage $t \in \mathbb{N}$ of the algorithm is associated to a threshold u_t and the corresponding excursion set $\Gamma_t = \{f > u_t\}$.

The initialization stage ($t = 0$) starts with the construction of a *space filling* set of points $\{X_1, \dots, X_{n_0}\}$ in \mathbb{X} (see Section 4.2.1 for more information on the specific technique used in this article), and an initial Monte Carlo sample $\mathbb{Y}_0 = \{Y_0^1, \dots, Y_0^m\}$, consisting of a set of independent random variables drawn from the density $q_0 = \pi_{\mathbb{X}}$.

After initialization, each subsequent stage $t \geq 1$ of BSS involves two phases: an *estimation phase*, where the estimation of Γ_t is carried out, and a *sampling phase*, where a sample \mathbb{Y}_t targeting the density q_t associated to u_t is produced from the previous sample \mathbb{Y}_{t-1} using the reweight/resample/move SMC scheme described in Section 2.3.

In more details, the estimation phase consists in making $N_t \geq 0$ new evaluations of f to refine the estimation of Γ_t . The number of evaluations 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 simulation algorithm.

The total number of evaluations at the end of the estimation phase at stage t is denoted by $n_t = n_{t-1} + N_t$. The total number of evaluations used by BSS is thus $n_T = n_0 + \sum_{t=1}^T N_t$. New evaluation points $X_{n_{t-1}+1}, X_{n_{t-1}+2}, \dots, X_{n_t}$ are determined using a SUR sampling strategy targeting the threshold u_t , as in Section 3.1 (see Appendix A for details about the numerical procedure).

In practice, the sequence of thresholds is not fixed beforehand and adaptive techniques are used to choose the thresholds (see Section 3.4) and the number of points per stage (see Section 3.5).

The BSS algorithm is presented in pseudo-code form in Table 2.

REMARK 3.2. An algorithm involving kriging-based adaptive sampling and subset simulation has been proposed by Dubourg and co-authors (Dubourg, Sudret and Bourinet, 2011; Dubourg, 2011) to address the problem of reliability-based design optimization. Their approach is quite different from this article's, which addresses the problem of reliability analysis. In particular, there is no direct interaction, in their algorithm, between the selection of evaluation points (adaptive sampling) and subset simulation (which is used, in its original form, to evaluate probabilities of failure).

3.4. *Adaptive choice of the thresholds u_t .* As discussed in Section 2.4, it can be proved that, for an idealized version of the subset simulation algorithm with fixed thresholds $u_0 < u_1 < \dots < u_T = u$, it is optimal to choose the thresholds to make all conditional probabilities $P_{\mathbb{X}}(\Gamma_{t+1}|\Gamma_t)$ equal. This leads to the idea of choosing the thresholds adaptively in such a way that, in the product estimate

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

each term but the last is equal to some prescribed 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 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).

Consider now an idealized BSS algorithm, where a) the initial design of experiment is independent of \mathbb{Y}_0 , b) the SUR criterion is computed exactly, or using a discretization scheme that does not use the \mathbb{Y}_t 's; c) the minimization of the SUR criterion is carried out independently of the \mathbb{Y}_t 's and d) the particles Y_t^j are independent and identically distributed according

TABLE 2
Bayesian subset simulation algorithm

-
1. **Initialization** (stage 0)
 - (a) Evaluate f on a set of points $\{X_1, \dots, X_{n_0}\}$, called the *initial design* (see Section 4.2.1 for details)
 - (b) Generate an IID sample $\mathbf{Y}_t = \{Y_0^1, \dots, Y_0^m\}$ from $\mathbb{P}_{\mathbf{X}}$.
 - (c) Choose a prior \mathbb{P}_0 (see Sections 3.1 and 4.2.1 for details).
 - (d) Set $u_0 = -\infty$, $g_0 = \tilde{g}_{0, -\infty} = \mathbb{1}_{\mathbf{X}}$, $n = n_0$ and $t = 1$.

 2. **Repeat** (stage t)
 - (a) **Estimation**
 - Set $k = 0$ and repeat
 - Select a threshold $\tilde{u}_{t,k}$ by solving Equation (18) for u_t (with $n_t = n$)
 - Stop if the condition (19) is met, with $n_t = n$ and $u_t = \tilde{u}_{t,k}$.
 - Select X_{n+1} using the SUR strategy (7)–(9) with respect to $\tilde{u}_{t,k}$.
 - Evaluate f at X_{n+1} . Increment n and k .
 - Set $N_t = k$, $n_t = n$, $u_t = \tilde{u}_{t,k}$ and $g_t = \tilde{g}_{n_t, u_t} = \mathbb{P}_{n_t}(\xi(\cdot)) > u_t$.
 - (b) **Sampling**
 - *Reweight*: calculate weights $w_{i,0}^t \propto g_t(Y_{t-1}^i)/g_{t-1}(Y_{t-1}^i)$.
 - *Resample*: generate a sample $\tilde{Y}_{t,i}$ from the distribution $\sum_{i=1}^m w_{i,0}^t \delta_{Y_{t-1}^i}$.
 - *Move*: for each $i \leq m$, $Y_t^i \sim K(\tilde{Y}_{t,i}, \cdot)$.
 - (c) Increment t .

 3. **Estimation** – The final probability of failure is estimated by

$$\hat{\alpha}_T^{\text{BSS}} = \prod_{t=0}^{T-1} \left(\frac{1}{m} \sum_{i=1}^m \frac{g_{t+1}(Y_t^i)}{g_t(Y_t^i)} \right).$$

to q_t . Assumptions a)–c) ensure that the sequence of densities $(q_t)_{1 \leq t \leq T}$ is deterministic given ξ . Then (see Appendix C),

$$(15) \quad \text{var} \left(\frac{\widehat{\alpha}_T^{\text{BSS}}}{\widehat{\alpha}_T^{\text{B}}} \middle| \xi \right) = \frac{1}{m} \sum_{t=1}^T \kappa_t + O \left(\frac{1}{m^2} \right),$$

where

$$(16) \quad \kappa_t := \frac{\int_{\mathbb{X}} g_t^2 / g_{t-1} \pi_{\mathbb{X}}}{(\widehat{\alpha}_t^{\text{B}})^2 / \widehat{\alpha}_{t-1}^{\text{B}}} - 1.$$

Minimizing the leading term $\frac{1}{m} \sum_{t=1}^T \kappa_t$ in (15) by an appropriate choice of thresholds is not as straightforward as in the case of the subset simulation algorithm. Assuming that $g_{t-1} \approx 1$ wherever g_t is not negligible, we get

$$\int_{\mathbb{X}} g_t^2 / g_{t-1} \pi_{\mathbb{X}} \approx \int_{\mathbb{X}} g_t^2 \pi_{\mathbb{X}} \leq \int_{\mathbb{X}} g_t \pi_{\mathbb{X}} = \widehat{\alpha}_t^{\text{B}},$$

and therefore the variance (15) is approximately upper-bounded by

$$(17) \quad \frac{1}{m} \sum_{t=1}^T (1 - \widehat{p}_t^{\text{B}}) / \widehat{p}_t^{\text{B}},$$

where $\widehat{p}_t^{\text{B}} := \widehat{\alpha}_t^{\text{B}} / \widehat{\alpha}_{t-1}^{\text{B}}$. Minimizing the approximate upper-bound (17) under the constraint

$$\prod_{t=1}^T \widehat{p}_t^{\text{B}} = \widehat{\alpha}_T^{\text{B}}$$

leads to choosing the thresholds u_t in such a way that \widehat{p}_t^{B} is the same for all stages t —that is $\widehat{p}_t^{\text{B}} = (\widehat{\alpha}_T^{\text{B}})^{1/T}$. As a consequence, we propose to choose the thresholds adaptively using the condition that, at each stage (but the last), the natural estimator $\widehat{\alpha}_t^{\text{BSS}} / \widehat{\alpha}_{t-1}^{\text{BSS}}$ of \widehat{p}_t^{B} is equal to some prescribed probability p_0 . Owing to (13), this amounts to choosing u_t in such a way that

$$(18) \quad \frac{1}{m} \sum_{i=1}^m \frac{g_t(Y_{t-1}^i)}{g_{t-1}(Y_{t-1}^i)} = p_0.$$

should be satisfied.

Equation (18) is easy to solve, since the left-hand side is a strictly decreasing and continuous function of u_t (to be precise, continuity holds under the assumption that the posterior variance of ξ does not vanish on one of the particles). In practice, we solve (18) each time a new evaluation is made, which yields a sequence of intermediate thresholds (denoted by $\tilde{u}_{t,0}, \tilde{u}_{t,1}, \dots$ in Table 2) at each stage $t \geq 1$. The actual value of u_t at stage t is only known after the last evaluation of stage t .

3.5. *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 f that must be done at each stage of the algorithm.

Assume that $t \geq 1$ is the current stage number; at the beginning of the stage, n_{t-1} evaluations are available from previous stages. After several additional evaluations, the number of available observations of f is $n \geq n_{t-1}$. We propose to stop adding new evaluations when the expected error of estimation of the set Γ_t , measured by $\mathbb{E}_n \left(\mathbb{P}_{\mathbb{X}} \left(\Gamma_t \triangle \hat{\Gamma}_{n,u_t} \right) \right)$, becomes smaller than some prescribed fraction η_t of its expected volume $\mathbb{E}_n \left(\mathbb{P}_{\mathbb{X}} \left(\Gamma_t \right) \right)$ under $\mathbb{P}_{\mathbb{X}}$. Writing these two quantities as

$$\begin{aligned} \mathbb{E}_n \left(\mathbb{P}_{\mathbb{X}} \left(\Gamma_t \right) \right) &= \int_{\mathbb{X}} \tilde{g}_{n,u_t}(x) \pi_{\mathbb{X}}(x) \, dx = \hat{\alpha}_{t-1}^{\mathbb{B}} \int_{\mathbb{X}} \frac{\tilde{g}_{n,u_t}(x)}{g_{t-1}(x)} q_{t-1}(x) \, dx, \\ \mathbb{E}_n \left(\mathbb{P}_{\mathbb{X}} \left(\Gamma_t \triangle \hat{\Gamma}_{n,u_t} \right) \right) &= \int_{\mathbb{X}} \tau_{n,u_t}(x) \pi_{\mathbb{X}}(x) \, dx = \hat{\alpha}_{t-1}^{\mathbb{B}} \int_{\mathbb{X}} \frac{\tau_{n,u_t}(x)}{g_{t-1}(x)} q_{t-1}(x) \, dx, \end{aligned}$$

where \tilde{g}_{n,u_t} and τ_{n,u_t} have been defined in Section 3.1, and estimating the integrals on the right-hand side using the SMC sample \mathbb{Y}_{t-1} , we end up with the stopping condition

$$\frac{1}{m} \sum_{i=1}^m \frac{\tau_{n,u_t} \left(Y_{t-1}^i \right)}{g_{t-1} \left(Y_{t-1}^i \right)} \leq \eta_t \cdot \frac{1}{m} \sum_{i=1}^m \frac{\tilde{g}_{n,u_t} \left(Y_{t-1}^i \right)}{g_{t-1} \left(Y_{t-1}^i \right)}.$$

which, if u_t is re-adjusted after each evaluation using Equation (18), boils down to

$$(19) \quad \sum_{i=1}^m \frac{\tau_{n,u_t} \left(Y_{t-1}^i \right)}{g_{t-1} \left(Y_{t-1}^i \right)} \leq \eta_t m p_0.$$

REMARK 3.3. In the case where several evaluations of the function can be carried out in parallel, it is possible to select evaluation points in batches during the sequential design phase of the algorithm. A batch-sequential version of the SUR strategy (7)–(9) has been proposed by [Chevalier et al. \(2013b\)](#).

REMARK 3.4. The stopping criterion (19) is slightly different from the one proposed earlier by [Li, Bect and Vazquez \(2012\)](#): $\sum_{i=1}^m \tau_{n,u_t} \left(Y_{t-1}^i \right) \leq \eta' m$. If we set $\eta' = \eta_t p_0$ and assume (quite reasonably) that $g_{t-1} \left(Y_{t-1}^i \right) \approx 1$ for the particles where $\tau_{n,u_t} \left(Y_{t-1}^i \right)$ is not negligible, then it becomes clear that the two criteria are essentially equivalent. As a consequence, the left-hand side of (19) can also be interpreted, approximately, as the expected number of misclassified particles (where the expectation is taken with respect to ξ , conditionally on the particles).

TABLE 3
Summary of test cases

Example	Name	d	α_{ref}
4.1.1	Four-branch series system	2	$5.596 \cdot 10^{-9}$
4.1.2	Deviation of a cantilever beam	2	$3.937 \cdot 10^{-6}$
4.1.3	Response of a nonlinear oscillator	6	$1.514 \cdot 10^{-8}$

4. Numerical experiments. In this section, we illustrate the proposed algorithm on three classical examples from the structural reliability literature and compare our results with those from the classical subset simulation algorithm and the ²SMART algorithm (Deheeger, 2008; Bourinet, Deheeger and Lemaire, 2011).

4.1. *Test cases.* For each of the following test cases, the reference value for the probability α has been obtained from one hundred independent runs of the subset simulation algorithm with sample size $m = 10^7$ (see Table 3).

4.1.1. *Four-branch series system.* Our first example is a variation on a classical structural reliability test case (see, e.g., Echard, Gayton and Lemaire, 2011, Example 1 with $k = 6$), where the threshold u is modified to make α smaller. The objective is to estimate the probability $\alpha = \mathbb{P}_{\mathbf{X}}(f(X) < u)$, where

$$(20) \quad f(x_1, x_2) = \min \left\{ \begin{array}{l} 3 + 0.1(x_1 - x_2)^2 - (x_1 + x_2)/\sqrt{2}, \\ 3 + 0.1(x_1 - x_2)^2 + (x_1 + x_2)/\sqrt{2}, \\ (x_1 - x_2) + 6/\sqrt{2}, \\ (x_2 - x_1) + 6/\sqrt{2} \end{array} \right\}$$

and $X_1, X_2 \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1)$. Taking $u = -4$, the probability of failure is approximately $5.596 \cdot 10^{-9}$, with a coefficient of variation of about 0.04%. Figure 1 (left panel) shows the failure domain and a sample from the input distribution $\mathbb{P}_{\mathbf{X}}$.

4.1.2. *Deviation of a cantilever beam.* Consider a cantilever beam, with a rectangular cross-section, subjected to a uniform load. The deflection of the tip of the beam can be written as

$$(21) \quad f(x_1, x_2) = \frac{3L^3}{2E} \frac{x_1}{x_2^3},$$

where x_1 is the load per unit area, x_2 the thickness of the beam, $L = 6$ m and $E = 2.6 \cdot 10^4$ MPa. The input variable X_1 and X_2 are assumed independent,

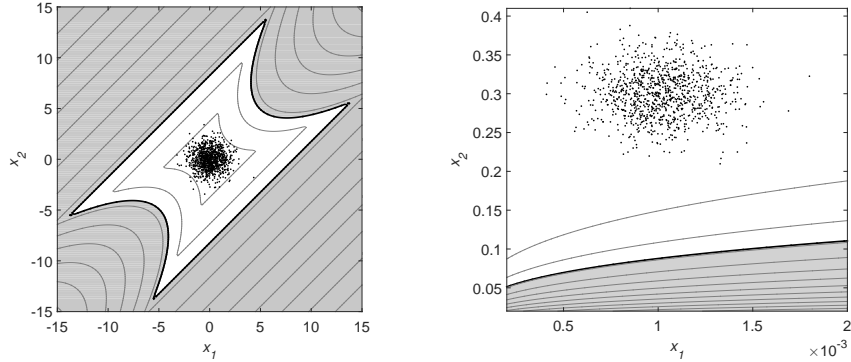


Fig 1: Contour plots of f in Example 4.1.1 (left) and Example 4.1.2 (right), along with a sample of size $m = 10^3$ from $P_{\mathbf{x}}$ (dots). A failure happens when x is in the gray area.

TABLE 4

Example 4.1.3: Means and standard deviations of the input variables.

Variable	x_1	x_2	x_3	x_4	x_5	x_6
μ_i	1	1	0.1	0.5	0.45	1
σ_i	0.05	0.1	0.01	0.05	0.075	0.2

with $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$, $\mu_1 = 10^{-3}$ MPa, $\sigma_1 = 0.2\mu_1$, and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$, $\mu_2 = 0.3$ m, $\sigma_2 = 0.1\mu_2$. A failure occurs when f is larger than $u = L/325$. The probability of failure is approximately $3.937 \cdot 10^{-6}$, with a coefficient of variation of about 0.03%. Note that the distribution of X_2 has been modified, with respect to the usual formulation (see, e.g., Gayton, Bourinet and Lemaire, 2003), to make α smaller. Figure 1 (right panel) shows a contour plot of f , along with a sample of the input distribution.

4.1.3. *Response of a nonlinear oscillator.* In this example (see, e.g., Echard et al., 2013), the input variable is six-dimensional and the cost function is:

$$(22) \quad f(x_1, x_2, x_3, x_4, x_5, x_6) = 3x_4 - \left| \frac{2x_5}{x_1 w_0^2} \sin\left(\frac{w_0 x_6}{2}\right) \right|$$

where $w_0 = \sqrt{\frac{x_2 + x_3}{x_1}}$. The input variables are assumed independent, normal, with mean and variance parameters given in Table 4. A failure happens when the cost function is larger than the threshold $u = 1.5$. The probability of failure is approximately $1.514 \cdot 10^{-8}$, with a coefficient of variation of about 0.04%. This variant of the problem corresponds exactly to the harder case in Echard et al.'s article.

4.2. Experimental settings.

4.2.1. BSS algorithm.

Initial design of experiments. We start with a initial design of size $n_0 = 5d$ (see [Loeppky, Sacks and Welch, 2009](#), for a discussion on the size of the initial design in computer experiments), generated as follows. First, a compact subset $\mathbb{X}_0 \subset \mathbb{X}$ is constructed³:

$$\mathbb{X}_0 = \prod_{j=1}^d [q_\varepsilon^j; q_{1-\varepsilon}^j]$$

where q_ε^j and $q_{1-\varepsilon}^j$ are the quantiles of order ε and $1 - \varepsilon$ of the j^{th} input variable. Then, a “good” LHS design on \mathbb{X}_0 is obtained as the best design, according to the maximin criterion ([Johnson, Moore and Ylvisaker, 1990](#); [Morris and Mitchell, 1995](#)), in set of Q random LHS designs on \mathbb{X}_0 . The values $\varepsilon = 10^{-5}$ and $Q = 10^4$ have been used in all our experiments.

Stochastic process prior. A Gaussian process prior with an unknown constant mean and a stationary anisotropic Matérn covariance function with regularity $5/2$ is used as our prior information about f (see [Appendix D](#) for more details). The unknown mean is integrated out as usual, using an improper uniform prior on \mathbb{R} ; as a consequence, the posterior mean coincides with the so-called “ordinary kriging” predictor. The remaining hyper-parameters (variance and range parameters of the covariance function) are estimated, following the empirical Bayes philosophy, by maximization of the marginal likelihood⁴. The hyper-parameters are estimated first on the data from the initial design, and then re-estimated after each new evaluation.

SMC parameters. Several values of the sample size m will be tested to study the relation between the variance of the estimator and the number of evaluations: $m \in \{500, 1000, 2000, \dots\}$. Several iterations of an adaptive anisotropic Gaussian Random Walk Metropolis-Hastings (RWMH) algorithm, fully described in [Appendix B](#), are used for the move step of the algorithm.

Stopping criterion for the SUR strategy. The number of evaluations selected using the SUR strategy is determined adaptively, using the stopping criterion [\(19\)](#) from [Section 3.5](#), with $\eta_t = 0.5$ for all $t < T$ (i.e., for all intermediate

³A similar technique is used by [Dubourg and co-authors](#) in a context of reliability-based design optimization ([Dubourg, Sudret and Bourinet, 2011](#); [Dubourg, 2011](#)).

⁴Used in combination with a uniform prior for the mean, for this specific model, the MML method is equivalent to the Restricted Maximum Likelihood (ReML) method advocated by [Stein \(1999, Section 6.4\)](#).

stages) and $\eta_T = 0.1 \widehat{\delta}_{m,T}$ where $\widehat{\delta}_{m,T}$ is the estimated coefficient of variation for the SMC estimator $\widehat{\alpha}_T^{\text{BSS}}$ of $\widehat{\alpha}_T^{\text{B}}$ (see Appendix C). Furthermore, we require for robustness a minimal number N_{\min} of evaluations at each stage, with $N_{\min} = 2$ in all our simulations.

Adaptive choice of the thresholds. The successive thresholds u_t are chosen using the adaptive rule proposed in Section 3.4, Equation (18), with $p_0 = 0.1$.

4.2.2. *Subset simulation algorithm.* The parameters used for the subset simulation algorithm are exactly the same, in all our simulations, as those used in the ‘‘SMC part’’ of the BSS algorithm (see Section 4.2.1). In particular, the number m_0 of surviving particles at each stage is determined according to the rule $p_0 = \frac{m_0}{m} = 0.1$ (see Table 1), and the adaptive MCMC algorithm described in Appendix B is used to move the particles. The number of evaluations made by the subset simulation algorithm is considered to be $m + (T - 1)(1 - p_0)m$, as explained in Section 2.4—in other words, in order to make the comparison as fair as possible, the additional evaluations required by the adaptive MCMC procedure are not taken into account.

4.2.3. *²SMART algorithm.* ²SMART (Bourinet, Deheeger and Lemaire, 2011) is a recently proposed algorithm for the estimation of small probabilities, based on the combination of Subset Simulation with Support Vector Machines (SVM). We will present results obtained using the implementation of ²SMART proposed in the software package FERUM 4.1 (Bourinet, 2010), with all parameters set to their default values (which are equal to the values given in Bourinet, Deheeger and Lemaire’s article).

4.3. Results.

4.3.1. *Illustration.* We first illustrate how BSS works using one run of the algorithm on Example 4.1.1 with sample size $m = 1000$. Snapshots of the algorithm at stages $t = 1$, $t = 5$ and $t = T = 9$ are presented on Figure 2. Observe that the additional evaluation points selected at each stage using the SUR criterion (represented by black triangles) are located in the vicinity of the current level set. The actual number of points selected at each stage, determined by the adaptive stopping rule, is reported in Table 5. Observe also that the set of particles (black dots in the right column) is able to effectively capture the bimodal target distribution. Finally, observe that a significant portion of the evaluation budget is spent on the final stage—this is again a consequence of our adaptive stopping rule, which refines the estimation of the final level set until the bias of the estimate is (on average under the posterior distribution) small compared to its standard deviation.

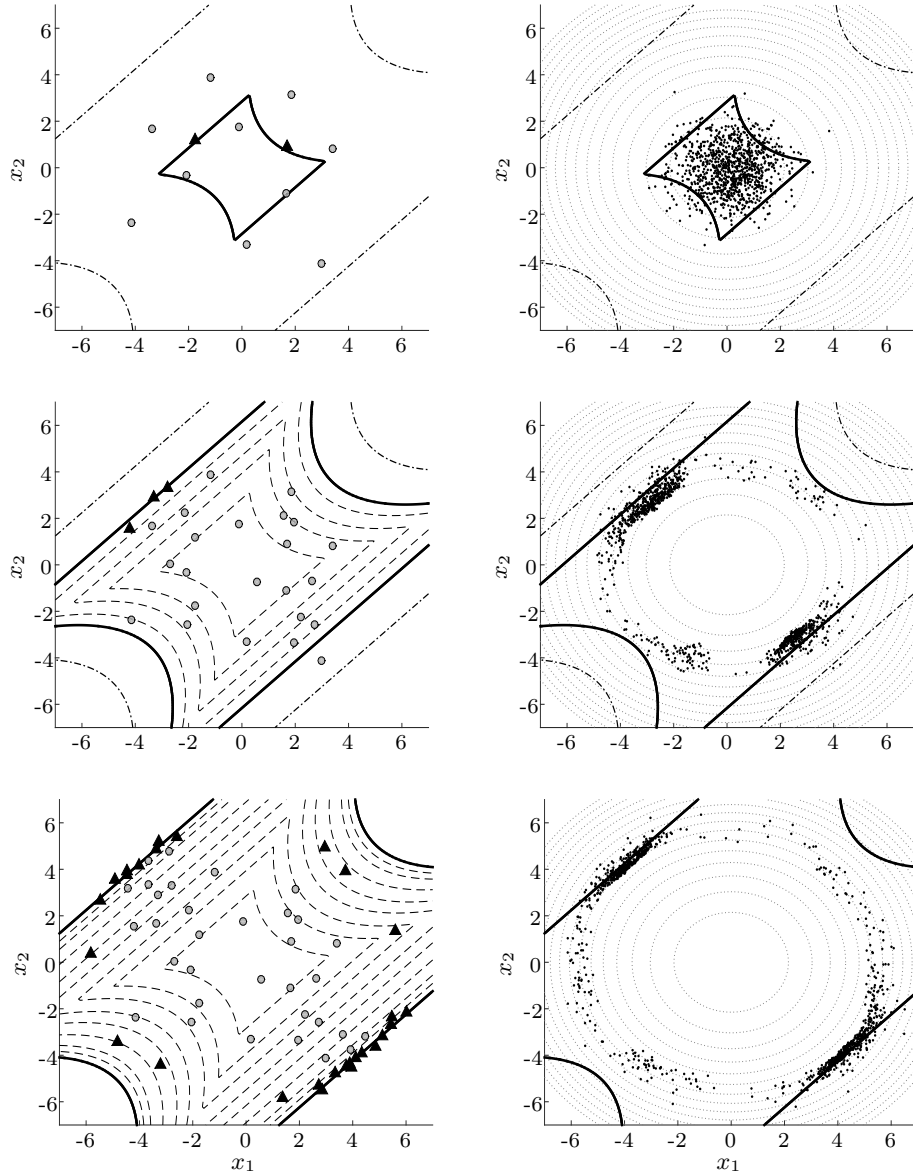


Fig 2: Snapshots of the BSS algorithm running on Example 4.1.1 (four branch series system) with sample size $m = 1000$. The first, second and third row correspond respectively to the end of the first stage ($t = 1$), the fifth stage ($t = 5$) and the last stage ($t = T = 9$). The true level set corresponding current target level u_t is represented by a thick line and, in the left column, true level sets corresponding to previous levels (u_s , $s < t$) are recalled using dashed contours. Evaluation points from previous stages are represented by gray disks (in particular, the initial design of experiment of size $n_0 = 10$ is visible on the top-left panel) and new evaluations performed at the current levels are marked by black triangles. In the right column, the sample points Y_{t-1}^j , $1 \leq j \leq m$ and the level sets of the input density $\pi_{\mathbb{X}}$ (corresponding to probabilities $1 - 10^{-k}$, $k = 1, 2, \dots$) are represented respectively by black dots and dotted lines.

TABLE 5

Number of evaluations per stage on Example 4.1.1 (four-branch series system). For the BSS algorithm, recall that the number of evaluations at each stage is chosen adaptively (see Section 3.5) and is therefore random: the numbers shown here correspond to the run with $m = 1000$ that is shown on Figure 2. For the subset simulation algorithm, the number of evaluations is directly related to the m , $q_0 = 1 - p_0$ and T (see Section 2.4).

stage number t	0	1	2	3	4	5
BSS ($m = 1000$)	$2d = 10$	2	6	3	2	3
subset simulation	m	q_0m	q_0m	q_0m	q_0m	q_0m

stage number t	6	7	8	9	total
BSS ($m = 1000$)	2	3	2	28	61
subset simulation	q_0m	q_0m	q_0m	0	$m + (T - 1)q_0m$

4.3.2. *Average results.* This section presents average results over one hundred independent runs for subset simulation, BSS and ²SMART.

Figure 3 represents the average number of evaluations used by the BSS algorithm as a function of the sample size m . The number of evaluations spent on the initial design is constant, since it only depends on the dimension d of the input space. The average number of evaluations spent on the intermediate stages ($t < T$) is also very stable⁵ and independent of the sample size m . Only the average number of evaluations spent on the final stage—i.e., to learn the level set of interest—is growing with m . This growth is necessary if one wants the estimation error to decrease when m increases: indeed, the variance of $\hat{\alpha}_T^{\text{BSS}}$ automatically goes to zero at the rate $\frac{1}{m}$, but the bias $\hat{\alpha}_T^{\text{B}} - \alpha$ does not unless additional evaluations are added at the final level to refine the estimation of Γ .

Figure 4 represents the relative Root-Mean-Square Error (RMSE) of all three algorithms, as a function of the average number of evaluations. For the subset simulation algorithm, the number of evaluations is directly proportional to m and the RMSE decreases as expected like $\frac{1}{m}$ (with a constant much smaller than that of plain Monte Carlo simulation). ²SMART clearly outperforms subset simulation, but offers no simple way of tuning the accuracy of the final estimate (which is why only one result is presented, using the default settings of the algorithm). Finally, BSS clearly and consistently outperforms both ²SMART and subset simulation on these three examples: the relative RMSE goes to zero at a rate much faster than subset simulation’s (a feature that is made possible by the smoothness of the limit-state function, which is leveraged by the Gaussian process model), and the sample-size m is the only tuning parameters that needs to be acted upon in order change

⁵Actually, for Examples 4.1.2 and 4.1.3, it is equal to $T N_{\min}$ for all runs; in other words, the adaptive stopping rule only came into play at intermediate stages for Example 4.1.1.

the accuracy of the final estimate. Figure 5 provides more insight into the error of the BSS estimate by confirming that, as intended by design of the adaptive stopping rule, variance is the main component of the RMSE (in other words, the bias is negligible in these examples).

5. Discussion. We propose an algorithm called Bayesian subset simulation for the estimation of small probabilities of failure—or more generally the estimation of the volume of excursion of a function above a threshold—when the limit-state function is expensive to evaluate. This new algorithm is built upon two key techniques: the SMC method known as subset simulation or adaptive multilevel splitting on the one hand, and the Bayesian (Gaussian process based) SUR sampling strategy on the other hand. SMC simulation provides the means for evaluating the Bayesian estimate of the probability of failure, and to evaluate and optimize the SUR sampling criterion. In turn, the SUR sampling strategy makes it possible to estimate the level sets of the (smooth) limit-state function using a restricted number of evaluations, and thus to build a good sequence of target density for SMC simulation. Our numerical experiments show that this combination achieves significant savings in evaluations on three classical examples from the structural reliability literature.

An adaptive stopping rule is used in the BSS algorithm to choose the number of evaluation added by the SUR sampling strategy at each stage. Evaluations at intermediate stages are not directly useful to refine the final probability estimate, but their importance must not be overlooked: they make it possible to learn in a robust way the level sets of the limit-state function, and therefore to build a sequence of densities that converges to the boundary of the failure region. Achieving a better understanding of the connection between the number of evaluations spent on intermediate level sets and the robustness of the algorithm is an important perspective for future work. In practice, if the budget of evaluations permits, we recommend running several passes of the BSS algorithm, with decreasing tolerances for the adaptive stopping rule, to make sure that no failure mode has been missed.

The adaptive stopping rule also makes it possible to refine the estimation of the final level set to make sure that the posterior model is good enough with respect to the SMC sample size. Other settings of the stopping rule could of course be considered. For instance, BSS could stop when the bias is expected to be of the same order than the standard deviation. Future work will focus on fully automated variants on the BSS algorithm, where the number of evaluations and the SMC sample size would be controlled in

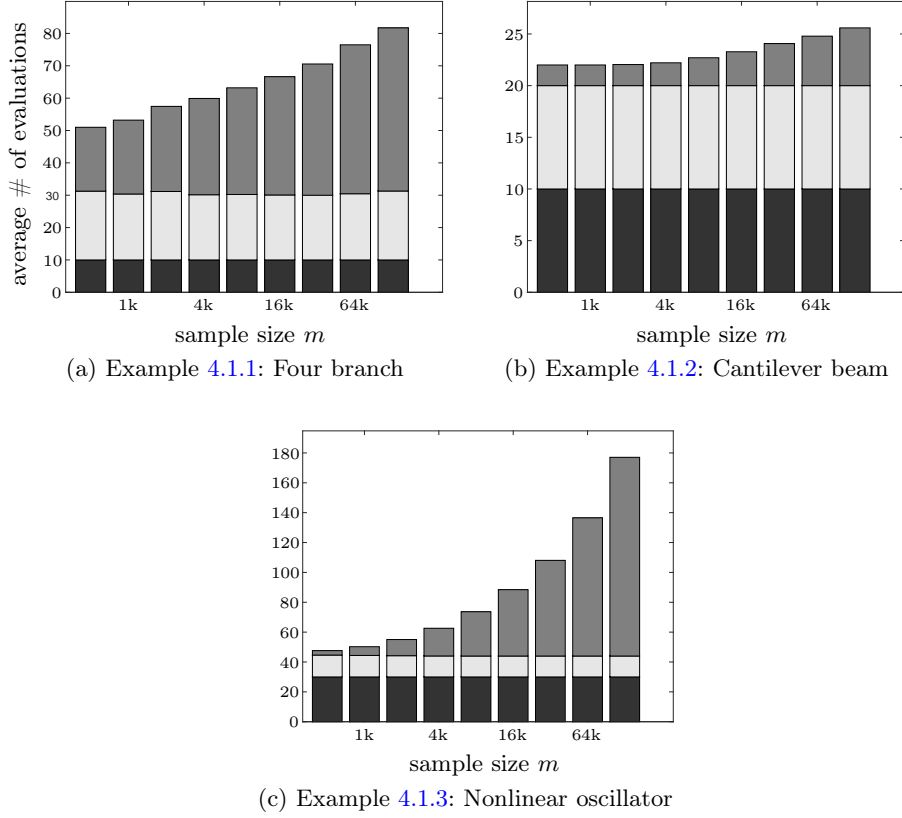


Fig 3: Average number of evaluations used by the BSS algorithm, over 100 independent runs, as a function of the sample size m on Examples 4.1.1–4.1.3. The total number of evaluations is split in three parts: the size n_0 of the initial design (dark gray), the number $\sum_{t=1}^{T-1} N_t$ of evaluations in intermediate stages (light gray) and the number of evaluations N_T in the final stage (middle gray).

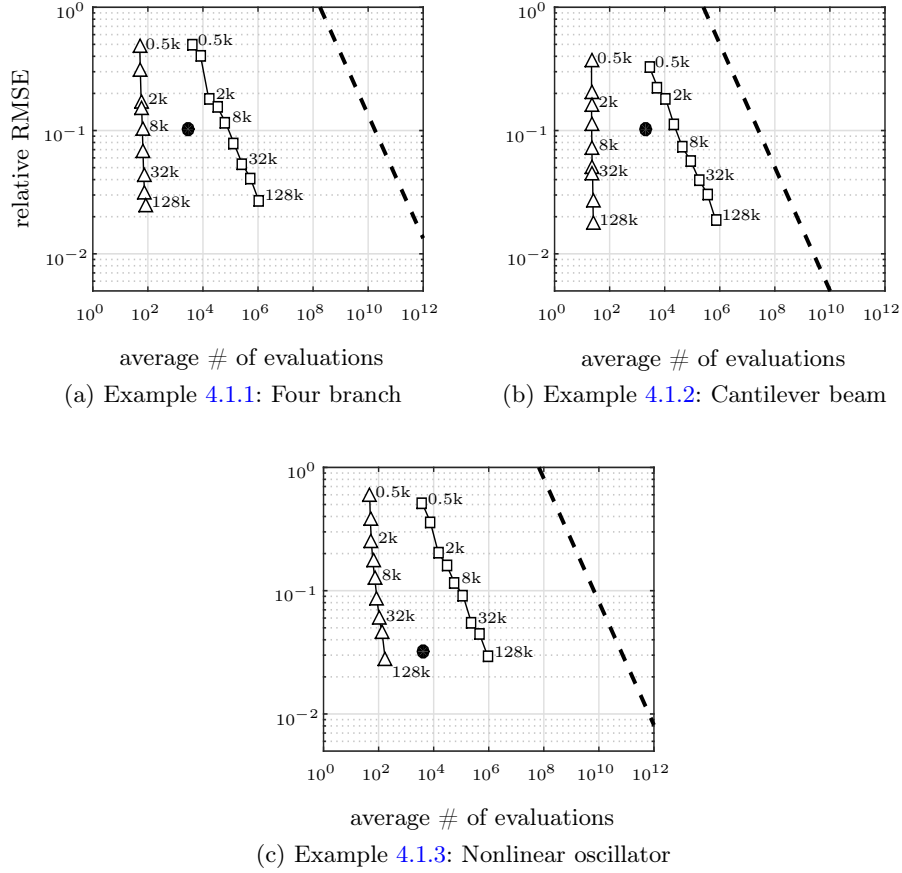


Fig 4: Relative root-mean-square error (RMSE) as a function of the average number of evaluations, over 100 independent runs, on Examples 4.1.1–4.1.3. For the subset simulation algorithm (squares) and for the BSS algorithm (triangles), the results are provided for several values of the sample size ($m \in \{500, 1000, 2000, \dots\}$). For the 2 SMART algorithm (filled circles), only one result is presented, corresponding to the default settings of the algorithm. The expected performance of plain Monte Carlo sampling is represented by a dashed line.

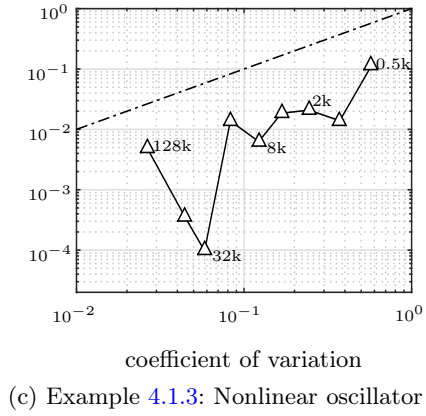
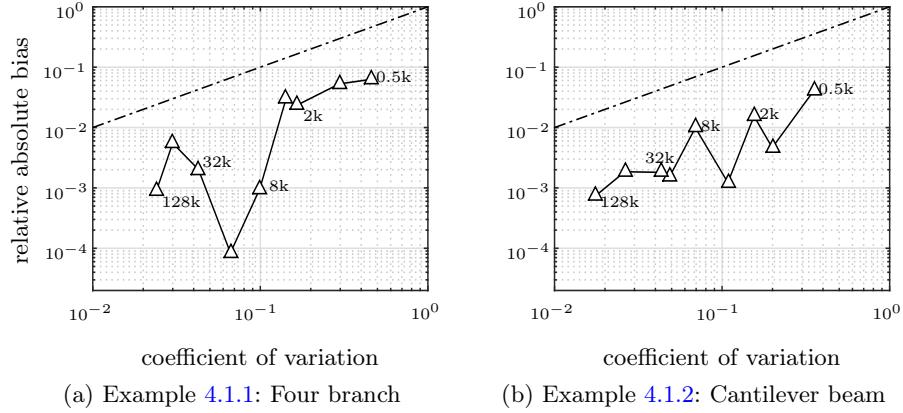


Fig 5: Relative absolute bias of the BSS estimator as a function of its coefficient of variation, estimated using 100 independent runs. The relative absolute bias is estimated using, for each test case, the reference value α_{ref} provided in Table 3.

order to achieve a prescribed error level.

Funding acknowledgment. This research was partially funded by the French *Fond Unique Interministériel* (FUI 7) in the context of the CSDL (Complex Systems Design Lab) project.

APPENDIX A: APPROXIMATION AND OPTIMIZATION OF THE SUR CRITERION

This section discusses the numerical procedure that we use for the approximation and optimization of the SUR criterion used at each stage of the BSS algorithm (see Sections 3.1 and 3.3):

$$J_n(x_{n+1}) = \int_{\mathbb{X}} \mathbf{E}_{n,x_{n+1}}(\tau_{n+1,u_t}(x)) \pi_{\mathbb{X}}(x) dx, \quad n_{t-1} \leq n \leq n_t - 1,$$

where we have introduced the simplified notation $\mathbf{E}_{n,x_{n+1}} := \mathbf{E}_n(\cdot | X_{n+1} = x_{n+1})$. The numerical approach proposed here is essentially the same as that used by [Bect et al. \(2012\)](#), with a more accurate way of computing the integrand, following ideas of [Chevalier et al. \(2013b\)](#).

Observing that

$$J_n(x_{n+1}) = \hat{\alpha}_{t-1}^{\mathbb{B}} \int_{\mathbb{X}} \frac{\mathbf{E}_{n,x_{n+1}}(\tau_{n+1,u_t}(x))}{g_{t-1}(x)} q_{t-1}(x) dx,$$

the integral over \mathbb{X} can be approximated, up to a constant, using the weighted sample \mathbb{Y}_{t-1} :

$$\begin{aligned} J_n(x_{n+1}) &\propto \int_{\mathbb{X}} \frac{\mathbf{E}_{n,x_{n+1}}(\tau_{n+1,u_t}(x))}{g_{t-1}(x)} q_{t-1}(x) dx \\ (23) \quad &\approx \sum_{j=1}^m w_{t-1}^j \frac{\mathbf{E}_{n,x_{n+1}}(\tau_{n+1,u_t}(x))|_{x=Y_{t-1}^j}}{g_{t-1}(Y_{t-1}^j)}. \end{aligned}$$

Then, simple computations using well-known properties of Gaussian processes under conditioning allow to obtain an explicit representation of the integrand, in the spirit of [Chevalier et al. \(2013b\)](#), as a function of the Gaussian process posterior mean $\hat{\xi}_n$ and posterior covariance k_n :

$$\begin{aligned} \mathbf{E}_{n,x_{n+1}}(\tau_{n+1,u_t}(x)) &= \Phi\left(\frac{u - \hat{\xi}_n(x)}{\sigma_n(x)}\right) + \Phi\left(\frac{u - \hat{\xi}_n(x)}{s_n(x, x_{n+1})}\right) \\ (24) \quad &- 2\Phi_2\left(\begin{pmatrix} u \\ u \end{pmatrix}; \begin{pmatrix} \hat{\xi}_n(x) \\ \hat{\xi}_n(x) \end{pmatrix}, \begin{pmatrix} s_n^2(x, x_{n+1}) & s_n^2(x, x_{n+1}) \\ s_n^2(x, x_{n+1}) & \sigma_n(x)^2 \end{pmatrix}\right), \end{aligned}$$

where Φ is the cumulative distribution function of the normal distribution, Φ_2 the cumulative distribution function of the bivariate normal distribution, $\sigma_n^2(x) = k_n(x, x)$ and $s_n^2(x, x_{n+1}) = k_n(x, x_{n+1})^2 / \sigma_n(x_{n+1})^2$.

The main computational bottleneck, in a direct implementation of the approximation (23) combined with the representation (24), is in our experience the computation of the bivariate cumulative distribution function Φ_2 . Indeed, assume that the optimization of the approximate criterion is carried out by means of an exhaustive discrete search on $\{Y_{t-1}^j, 1 \leq j \leq m\}$. Then m^2 evaluations of Φ_2 are required in order to select X_{n+1} . To mitigate this problem, we implemented the pruning idea proposed by [Bect et al. \(2012, Section 3.4\)](#): only a subset of size $m_0 \leq m$ of the set of particles is actually used, both for the approximation of the integral and for the optimization of the criterion. In this article, the size m_0 is determined automatically as follows: first, for each particle Y_{t-1}^j , the current weighted probability of misclassification

$$\tilde{\tau}_n^j := w_{t-1}^j \tau_{n, u_t} \left(Y_{t-1}^j \right) / g_{t-1} \left(Y_{t-1}^j \right)$$

is computed. Then, the particles are sorted according to the value of $\tilde{\tau}_n^j$, in decreasing order: $\tilde{\tau}_n^{\varphi(1)} \geq \tilde{\tau}_n^{\varphi(2)} \geq \dots \geq \tilde{\tau}_n^{\varphi(m)}$, and m_0 is set to $\min(m_0^{\max}, m_0(\tilde{\tau}_n))$, where $m_0(\tilde{\tau}_n)$ is the smallest integer such that

$$\sum_{j=1}^{m_0} \tilde{\tau}_n^{\varphi(j)} \geq \rho \sum_{j=1}^m \tilde{\tau}_n^j.$$

The values $m_0^{\max} = 1000$ and $\rho = 0.99$ have been used in all our simulations.

APPENDIX B: ADAPTIVE METROPOLIS-HASTINGS ALGORITHM FOR THE MOVE STEP

A fixed number S of iterations of an anisotropic Gaussian Random Walk Metropolis-Hastings (RWMH) kernel is used for the move step, with adaptation of the standard deviations of the increments. More precisely, for $s = 1, 2, \dots, S$, starting with the set of particles $\left(Y_{t,1}^{(j)} \right)_{1 \leq j \leq m}$ produced by the resampling step, we first produce perturbed particles:

$$\tilde{Y}_{t,s}^{(j)} = Y_{t,s}^{(j)} + \Sigma_{\text{RW},t,s} U_{t,s}, \quad 1 \leq j \leq m,$$

where $\Sigma_{\text{RW},t,s}$ is a diagonal matrix and $U_{t,s}$ a d -dimensional standard normal vector. The perturbed particle is accepted as $Y_{t,s+1}^{(j)}$ with probability

$$a_{t,s}^{(j)} = 1 \wedge \frac{q_t \left(\tilde{Y}_{t,s}^{(j)} \right)}{q_t \left(Y_{t,s}^{(j)} \right)},$$

and $Y_{t,s}^{(j)}$ is kept otherwise. The k^{th} diagonal element $\sigma_{\text{RW},t,s}^{(k)}$ of $\Sigma_{\text{RW},t,s}$ is initialized with

$$(25) \quad \sigma_{\text{RW},t,1}^{(k)} = C_{\sigma,\text{init}} \sigma_{\mathbb{X}}^{(k)},$$

where $\sigma_{\mathbb{X}}^{(k)}$ is the standard deviation of the k^{th} marginal of $\mathbb{P}_{\mathbb{X}}$, and then updated using

$$(26) \quad \log \sigma_{\text{RW},t,s+1}^{(k)} = \begin{cases} \log \sigma_{\text{RW},t,s}^{(k)} + \Delta_{\sigma}/s & \text{if } \bar{a}_{t,s} > a_{\text{target}} \\ \log \sigma_{\text{RW},t,s}^{(k)} - \Delta_{\sigma}/s & \text{otherwise,} \end{cases}$$

where $\bar{a}_{t,s} = \frac{1}{m} \sum_{j=1}^m a_{t,s}^{(j)}$ is the average acceptance probability and a_{target} some prescribed target value.

The following parameter values have been used for the numerical simulations presented in this article: $S = 10$, $C_{\sigma,\text{init}} = 2/\sqrt{d}$, $\Delta_{\sigma} = \log(2)$ and $a_{\text{target}} = 30\%$.

REMARK B.1. The $1/\sqrt{d}$ scaling for the constant $C_{\sigma,\text{init}}$ in (25) is motivated by the well-known theorem of [Roberts, Gelman and Gilks \(1997\)](#), which provides the optimal covariance matrix $2.38^2 \Sigma/d$ for a Gaussian target with covariance matrix Σ in high dimension.

REMARK B.2. The reader is referred to [Vihola \(2010\)](#) and references therein for more information on adaptive MCMC algorithms, including adaptive scaling algorithms such as (26). Note, however, that SMC algorithms do not rely on the ergodic properties of such adaptive schemes.

APPENDIX C: COMPUTATION OF THE VARIANCE IN THE IDEALIZED SETTING

This section provides a derivation of Equations (15) and (16), together with an explicit expression of the estimated coefficient of variation $\hat{\delta}_{m,T}$ used in Section 4.2.1. Both are obtained in the setting of the idealized BSS algorithm described in Section 3.4, where the samples $\mathbb{Y}_t = \{Y_t^1, \dots, Y_t^m\}$ are assumed IID (with density q_t) and mutually independent.

Recall from Equation (27) that the BSS estimator can be written as

$$(27) \quad \hat{\alpha}_T^{\text{BSS}} = \prod_{t=1}^T \frac{\hat{\alpha}_t^{\text{BSS}}}{\hat{\alpha}_{t-1}^{\text{BSS}}} = \prod_{t=1}^T \left(\frac{1}{m} \sum_{j=1}^m \frac{g_t(Y_{t-1}^j)}{g_{t-1}(Y_{t-1}^j)} \right) = \prod_{t=1}^T \hat{p}_t^{\text{BSS}},$$

where we have set, for all $t \in \{1, \dots, T\}$,

$$\widehat{p}_t^{\text{BSS}} = \frac{1}{m} \sum_{j=1}^m \frac{g_t(Y_{t-1}^j)}{g_{t-1}(Y_{t-1}^j)}.$$

Observe that the random variables $\widehat{p}_t^{\text{BSS}}$ are independent, with mean

$$\mathbb{E} \left(\widehat{p}_t^{\text{BSS}} \right) = \int_{\mathbb{X}} \frac{g_t}{g_{t-1}} q_{t-1} = \int_{\mathbb{X}} \frac{g_t}{g_{t-1}} \frac{g_{t-1} \pi_{\mathbb{X}}}{\widehat{\alpha}_{t-1}^{\text{B}}} = \frac{\widehat{\alpha}_t^{\text{B}}}{\widehat{\alpha}_{t-1}^{\text{B}}} = \widehat{p}_t^{\text{B}}$$

and variance

$$\begin{aligned} \text{var} \left(\widehat{p}_t^{\text{BSS}} \right) &= \frac{1}{m} \text{var} \left(\frac{g_t(Y_{t-1}^1)}{g_{t-1}(Y_{t-1}^1)} \right) \\ &= \frac{1}{m} \left[\int_{\mathbb{X}} \frac{g_t^2}{g_{t-1}^2} \frac{g_{t-1} \pi_{\mathbb{X}}}{\widehat{\alpha}_{t-1}^{\text{B}}} - \left(\frac{\widehat{\alpha}_t^{\text{B}}}{\widehat{\alpha}_{t-1}^{\text{B}}} \right)^2 \right] \\ &= \frac{1}{m} \left[\frac{1}{\widehat{\alpha}_{t-1}^{\text{B}}} \int_{\mathbb{X}} \frac{g_t^2}{g_{t-1}} \pi_{\mathbb{X}} - \left(\frac{\widehat{\alpha}_t^{\text{B}}}{\widehat{\alpha}_{t-1}^{\text{B}}} \right)^2 \right] \\ &= \frac{1}{m} (\widehat{p}_t^{\text{B}})^2 \kappa_t, \end{aligned}$$

where κ_t is defined by (16). Therefore, the coefficients of variation $\delta_{m,t}$ of the sequence of estimators $\widehat{\alpha}_t^{\text{BSS}}$ obey the recurrence relation $\delta_{m,t}^2 = \frac{1}{m} \kappa_t + \left(1 + \frac{1}{m} \kappa_t\right) \delta_{m,t-1}^2$, and we conclude that

$$\begin{aligned} \delta_{m,T}^2 &= \frac{1}{m} \kappa_T + \left(1 + \frac{1}{m} \kappa_T\right) \frac{1}{m} \kappa_{T-1} \\ &\quad + \left(1 + \frac{1}{m} \kappa_T\right) \left(1 + \frac{1}{m} \kappa_{T-1}\right) \frac{1}{m} \kappa_{T-2} + \dots \\ &= \frac{1}{m} \sum_{t=1}^T \kappa_t + O\left(\frac{1}{m^2}\right), \end{aligned}$$

which proves Equations (15)–(16). We construct an estimator of the coefficient of variation recursively, using the relation

$$\widehat{\delta}_{m,t}^2 = \frac{1}{m} \widehat{\kappa}_t + \left(1 + \frac{1}{m} \widehat{\kappa}_t\right) \widehat{\delta}_{m,t-1}^2,$$

with

$$\widehat{\kappa}_t = \left(\widehat{p}_t^{\text{BSS}} \right)^{-2} \cdot \frac{1}{m} \sum_{j=1}^m \left(\frac{g_t(Y_{t-1}^j)}{g_{t-1}(Y_{t-1}^j)} - \widehat{p}_t^{\text{BSS}} \right)^2.$$

APPENDIX D: STOCHASTIC PROCESS PRIOR

The stochastic process prior used for the numerical experiments in this article is a rather standard Gaussian process model. We describe it here in full detail for the sake of completeness. First, ξ is written as

$$\xi(x) = \mu + \xi^0(x),$$

where $\mu \in \mathbb{R}$ is an unknown constant mean and ξ^0 a zero-mean stationary Gaussian process with anisotropic covariance function

$$(28) \quad k(x, y) = \sigma^2 \kappa_\nu \left(\sqrt{\sum_{i=1}^d \frac{(x_{[i]} - y_{[i]})^2}{\rho_i^2}} \right), \quad x, y \in \mathbb{R}^d,$$

where $x_{[i]}, y_{[i]}$ denote the i^{th} coordinate of x and y , and κ_ν the Matérn correlation function of regularity ν (see [Stein, 1999](#), Section 2.10). The scale parameters ρ_1, \dots, ρ_d (characteristic correlation lengths) are usually called the *range* parameters of the covariance function.

In this article, the regularity parameter ν is set to the fixed value $\nu = 5/2$, leading to the following analytical expression for the Matérn correlation function:

$$(29) \quad \kappa_\nu(h) = \left(1 + \tilde{h} + \frac{1}{3}\tilde{h}^2 \right) \exp(-\tilde{h}), \quad \text{with } \tilde{h} = \sqrt{10} |h|.$$

As a consequence, ξ is twice differentiable in the mean-square sense, with sample paths almost surely in the Sobolev space $W^{s,2}$ for all $s < 5/2$ ([Scheuerer, 2010](#), Theorem 3).

REMARK D.1. The parametrization used in Equation (29) is the one advocated by [Stein \(1999, Section 2.10\)](#). Other parametrizations are sometimes used in the literature; e.g., [Rasmussen and Williams \(2006, Section 4.2.1\)](#) use $\tilde{h} = \sqrt{5} |h|$.

REFERENCES

- ARNAUD, A., BECT, J., COUPLET, M., PASANISI, A. and VAZQUEZ, E. (2010). Évaluation d'un risque d'inondation fluviale par planification séquentielle d'expériences. In *42èmes Journées de Statistique (JdS 2010)*, Marseille, France, May 24–28.
- AU, S. K. and BECK, J. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics* **16** 263–277.
- AUFFRAY, Y., BARBILLON, P. and MARIN, J. M. (2014). Bounding rare event probabilities in computer experiments. *Computational Statistics & Data Analysis* **80** 153–166.

- BAYARRI, M. J., BERGER, J. O., CALDER, E. S., DALBEY, K., LUNAGOMEZ, S., PATRA, A. K., PITMAN, E. B., SPILLER, E. T. and WOLPERT, R. L. (2009). Using statistical and computer models to quantify volcanic hazards. *Technometrics* **51** 402–413.
- BECT, J., GINSBOURGER, D., LI, L., PICHENY, V. and VAZQUEZ, E. (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing* **22** 773–793.
- BOURINET, J. M. (2010). FERUM 4.1 User’s Guide. <http://www.ifma.fr/FERUM>.
- BOURINET, J. M., DEHEEGER, F. and LEMAIRE, M. (2011). Assessing small failure probabilities by combined subset simulation and support vector machines. *Structural Safety* **33** 343–353.
- CANNAMELA, C., GARNIER, J. and IOOSS, B. (2008). Controlled stratification for quantile estimation. *Annals of Applied Statistics* **2** 1554–1580.
- CARON, V., GUYADER, A., MUNOZ ZUNIGA, M. and TUFFIN, B. (2014). Some recent results in rare event estimation. In *ESAIM: Proceedings* **44** 239–259.
- CHEVALIER, C., BECT, J., GINSBOURGER, D. and MOLCHANOV, I. (2013a). Estimating and quantifying uncertainties on level sets using the Vorob’ev expectation and deviation with Gaussian process models. In *mODa 10 — Advances in Model-Oriented Design and Analysis. Contributions to Statistics* 35–43. Springer.
- CHEVALIER, C., BECT, J., GINSBOURGER, D., VAZQUEZ, E., PICHENY, V. and RICHEL, Y. (2013b). Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics* **56** 455–465.
- CHOPIN, N. (2002). A sequential particle filter method for static models. *Biometrika* **89** 539–552.
- CÉROU, F., DEL MORAL, P., FURON, T. and GUYADER, A. (2012). Sequential Monte Carlo for rare event estimation. *Statistics and Computing* **22** 795–808.
- DE ROCQUIGNY, E., DEVICTOR, N. and TARANTOLA, S., eds. (2008a). *Uncertainty in industrial practice*. Wiley.
- DE ROCQUIGNY, E., DEVICTOR, N. and TARANTOLA, S., eds. (2008b). *Determination of the risk due to personal electronic devices (PEDs) carried out on radio-navigation systems aboard aircraft* 5, 65–80. Wiley.
- DE ROCQUIGNY, E., DEVICTOR, N. and TARANTOLA, S., eds. (2008c). *Uncertainty and reliability study of a creep law to assess the fuel cladding behaviour of PWR spent fuel assemblies during interim dry storage* 8, 111–122. Wiley.
- DEHEEGER, F. (2008). Couplage mécano-fiabiliste: ²SMART – Méthodologie d’apprentissage stochastique en fiabilité PhD thesis, Université Blaise Pascal (Clermont-Ferrand II).
- DEL MORAL, P., DOUCET, A. and JASRA, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68** 411–436.
- DIACONIS, P. and HOLMES, S. (1995). *Three examples of Monte-Carlo Markov chains: at the interface between statistical computing, computer science, and statistical mechanics*. In *Discrete probability and algorithms. IMA volumes in Mathematics and its Applications* **72** 43–56.
- DOUC, R. and CAPPÉ, O. (2005). Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA)* 64–69.
- DOUC, R. and MOULINES, E. (2008). Limit theorems for weighted samples with applications to sequential Monte Carlo methods. *The Annals of Statistics* **36** 2344–2376.
- DUBOURG, V. (2011). Adaptive surrogate models for reliability analysis and reliability-based design optimization PhD thesis, Université Blaise Pascal – Clermont II.

- DUBOURG, V., DEHEEGER, F. and SUDRET, B. (2011). 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)*.
- DUBOURG, V., DEHEEGER, F. and SUDRET, B. (2013). Metamodel-based importance sampling for structural reliability analysis. *Probabilistic Engineering Mechanics* **33** 47–57.
- DUBOURG, V., SUDRET, B. and BOURINET, J. M. (2011). Reliability-based design optimization using kriging surrogates and subset simulation. *Structural Multidisciplinary Optimization* **44** 673–690.
- ECHARD, B., GAYTON, N. and LEMAIRE, M. (2011). AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation. *Structural Safety* **33** 145–154.
- ECHARD, B., GAYTON, N., LEMAIRE, M. and RELUN, N. (2013). A combined Importance Sampling and Kriging reliability method for small failure probabilities with time-demanding numerical models. *Reliability Engineering & System Safety* **111** 232–240.
- GARCIA, E. (2010). Electromagnetic Compatibility Uncertainty, Risk, and Margin Management. *IEEE Transactions on Electromagnetic Compatibility* **52** 3–10.
- GAYTON, N., BOURINET, J. M. and LEMAIRE, M. (2003). CQ2RS: a new statistical approach to the response surface method for reliability analysis. *Structural Safety* **25** 99–121.
- GLASSERMAN, P., HEIDELBERGER, P., SHAHABUDDIN, P. and ZAJIC, T. (1999). Multi-level splitting for estimating rare event probabilities. *Operations Research* **47** 585–600.
- GUYADER, A., HENGARTNER, N. and MATZNER-LØBER, E. (2011). Simulation and Estimation of Extreme Quantiles and Extreme Probabilities. *Applied Mathematics & Optimization* **64** 171–196.
- JOHNSON, M. E., MOORE, L. M. and YLVIKAKER, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* **26** 131–148.
- JONKMAN, S. N., KOK, M. and VRIJLING, J. K. (2008). Flood risk assessment in the Netherlands: A case study for dike ring South Holland. *Risk Analysis* **28** 1357–1374.
- LI, L., BECT, J. and VAZQUEZ, E. (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*.
- LIU, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer.
- LOEPPKY, J. L., SACKS, J. and WELCH, W. J. (2009). Choosing the Sample Size of a Computer Experiment: A Practical Guide. *Technometrics* **51** 366–376.
- MELCHERS, R. E. (1999). *Structural Reliability: Analysis and Prediction. Second Edition*. Wiley.
- MORRIS, M. D. and MITCHELL, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference* **43** 381–402.
- OAKLEY, J. (2004). Estimating percentiles of uncertain computer code outputs. *J. Roy. Statist. Soc. Ser. C* **53** 83–93.
- O’CONNOR, P. P. and KLEYNER, A. (2012). *Practical Reliability Engineering*. Wiley.
- RASMUSSEN, C. E. and WILLIAMS, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.
- RAUSAND, M. and HOYLAND, A. (2004). *System reliability theory: models and statistical methods (second edition)*. Wiley.
- ROBERT, C. P. and CASELLA, G. (2004). *Monte Carlo statistical methods, 2nd edition*. Springer Verlag.
- ROBERTS, G. . O., GELMAN, A. and GILKS, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability* **7** 110–

120.

- SANTNER, T. J., WILLIAMS, B. J. and NOTZ, W. (2003). *The Design and Analysis of Computer Experiments*. Springer Verlag.
- SCHEUERER, M. (2010). Regularity of the sample paths of a general second order random field. *Stochastic Processes and their Applications* **120** 1879–1897.
- STEIN, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New York.
- VIHOLA, M. (2010). On the convergence of unconstrained adaptive Markov chain Monte Carlo algorithms PhD thesis.
- VILLEMONTAIX, J., VAZQUEZ, E. and WALTER, E. (2009). An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization* **44** 509–534.
- WAARTS, P. H. (2000). Structural reliability using finite element methods PhD thesis, Delft University of Technology.
- ZIO, E. and PEDRONI, N. (2009). Estimation of the functional failure probability of a thermal-hydraulic passive system by Subset Simulation. *Nuclear Engineering and Design* **239** 580–599.