Subset Simulation Method for Rare Event Estimation: An Introduction

Synonyms

Engineering reliability; Failure probability; Markov chain Monte Carlo; Monte Carlo simulation; Rare events; Subset Simulation

Introduction

This entry provides a detailed introductory description of Subset Simulation, an advanced stochastic simulation method for estimation of small probabilities of rare failure events. A simple and intuitive derivation of the method is given along with the discussion on its implementation. The method is illustrated with several easy-to-understand examples. The reader is assumed to be familiar only with elementary probability theory and statistics.

Subset Simulation (SS) is an efficient and elegant method for simulating rare events and estimating the corresponding small tail probabilities. The method was originally developed by Siu-Kui Au and James Beck in the already classical paper (Au and Beck 2001a) for estimation of structural reliability of complex civil engineering systems such as tall buildings and bridges at risk from earthquakes. The method turned out to be so powerful and general that over the last decade, SS has been successfully applied to reliability problems in geotechnical, aerospace, fire, and nuclear engineering. Moreover, the idea of SS proved to be useful not only in reliability analysis but also in other problems associated with general engineering systems, such as sensitivity analysis, design optimization, and uncertainty quantification. As of October 2014, according to the Web of Science (ISI) database and Google Scholar, the original SS paper (Au and Beck 2001a) received 315 and 572 citations respectively, that indicates the high impact of the Subset Simulation method on the engineering research community.

Subset Simulation is essentially based on two different ideas: conceptual and technical. The conceptual idea is to decompose the rare event F into a sequence of progressively "less-rare" nested events,

$$F = F_m \subset F_{m-1} \subset \ldots \subset F_1, \tag{1}$$

where F_1 is a relatively frequent event. For example, suppose that F represents the event of getting exactly m heads when flipping a fair coin m times. If m is large, then F is a rare event. To decompose F into a sequence (Eq. 1), let us define F_k to be the event of getting exactly k heads in the first k flips, where k = 1,…,m. The smaller the k, the less rare the corresponding event F_k, and F_1 - getting heads in the first flip - is relatively frequent.

Given a sequence of subsets (Eq. 1), the small probability $\mathbb{P}(F)$ of the rare event F can then be represented as a product of larger probabilities as follows:

$$\begin{aligned}\mathbb{P}(F) = \mathbb{P}(F_m) &= \mathbb{P}(F_1)\,\frac{\mathbb{P}(F_2)}{\mathbb{P}(F_1)}\,\frac{\mathbb{P}(F_3)}{\mathbb{P}(F_2)}\cdots\frac{\mathbb{P}(F_{m-1})}{\mathbb{P}(F_{m-2})}\,\frac{\mathbb{P}(F_m)}{\mathbb{P}(F_{m-1})} \\ &= \mathbb{P}(F_1)\cdot\mathbb{P}\left(F_2\middle|F_1\right)\cdot\cdots\cdot\mathbb{P}\left(F_m\middle|F_{m-1}\right),\end{aligned} \tag{2}$$

where $\mathbb{P}(F_k|F_{k-1}) = \mathbb{P}(F_k)/\mathbb{P}(F_{k-1})$ denotes the conditional probability of event F_k given the occurrence of event F_{k-1}, for k = 2,…,m. In the coin example, $\mathbb{P}(F_1) = 1/2$, all conditional probabilities $\mathbb{P}(F_k|F_{k-1}) = 1/2$, and the probability of the rare event $\mathbb{P}(F) = 1/2^m$.

Unlike the coin example, in real applications, it is often not obvious how to decompose the rare event into a sequence in Eq. 1 and how to compute all conditional probabilities in Eq. 2. In Subset Simulation, the "sequencing" of the rare event is done adaptively as the algorithm proceeds. This is achieved by employing Markov chain Monte Carlo, an advanced simulation technique, which constitutes the second - technical - idea behind SS. Finally, all conditional probabilities are automatically obtained as a by-product of the adaptive sequencing.

The main goals of this entry are (a) to provide a detailed exposition of Subset Simulation at an introductory level, (b) to give a simple derivation of the method and discuss its implementation, and (c) to illustrate SS with intuitive examples. Although the scope of SS is much wider, in this entry the method is described in the context of engineering reliability estimation; the problem SS was originally developed for in Au and Beck (2001a).

The rest of the entry is organized as follows: section "Engineering Reliability Problem" describes the engineering reliability problem and explains why this problem is computationally challenging. Section "The Direct Monte Carlo Method" discusses how the Direct Monte Carlo method can be used for engineering reliability estimation and why it is often inefficient. In section "Preprocessing: Transformation of Input Variables," a necessary preprocessing step which is often used by many reliability methods is briefly discussed. Section "The Subset Simulation Method" is the core of the entry, where the SS method is explained. Illustrative examples are considered in section "Illustrative Examples." For demonstration purposes, the MATLAB code for the considered examples is provided in section "MATLAB code." Section "Summary" concludes the entry with a brief summary.

Engineering Reliability Problem

One of the most important and computationally challenging problems in reliability engineering is to estimate the probability of failure for a system, that is, the probability of unacceptable system performance. The behavior of the system can be described by a response variable y, which may represent, for example, the roof displacement or the largest interstory drift. The response variable depends on input variables $x = (x_1,\ldots,x_d)$, also called basic variables, which may represent geometry, material properties, and loads,

$$y = g(x_1, \ldots, x_d), \tag{3}$$

where g(x) is called the performance function. The performance of the system is measured by comparison of the response y with a specified critical value y^*: if $y \le y^*$, then the system is safe; if $y > y^*$, then the system has failed. This failure criterion allows to define the failure domain F in the input x-space as follows:

$$F = \{x : g(x) > y^*\}. \tag{4}$$

In other words, the failure domain is a set of values of input variables that lead to unacceptance system performance, namely, to the exceedance of some prescribed critical threshold y^*, which may represent the maximum permissible roof displacement, maximum permissible interstory drift, etc.

Engineering systems are complex systems, where complexity, in particular, means that the information about the system (its geometric and material properties) and its environment (loads) is never complete. Therefore, there are always uncertainties in the values of input variables x. To account for these uncertainties, the input variables are modeled as random variables whose marginal distributions are usually obtained from test data, from expert opinion, or from literature. Let π(x) denote the joint probability density function (PDF) for x. The uncertainty in the input variables is propagated through Eq. 3 into the response variable y, which makes the failure event $\{x \in F\} = \{y > y^*\}$ also uncertain. The engineering reliability problem is then to compute the probability of failure pF, given by the following expression:

$$p_F = \mathbb{P}(x \in F) = \int_F \pi(x)dx. \tag{5}$$

The behavior of complex systems, such as tall buildings and bridges, is represented by a complex model (3). In this context, complexity means that the performance function g(x), which defines the integration region F in Eq. 5, is not explicitly known. The evaluation of g(x) for any x is often time-consuming and usually done by the finite element method (FEM), one of the most important numerical tools for computation of the response of engineering systems. Thus, it is usually impossible to evaluate the integral in Eq. 5 analytically because the integration region, the failure domain F, is not known explicitly.

Moreover, traditional numerical integration is also generally not applicable. In this approach, the d-dimensional input x-space is partitioned into a union of disjoint hypercubes, $\square_1,\ldots,\square_N$. For each hypercube $\square_i$, a "representative" point $x^{(i)}$

is chosen inside that hypercube, $x^{(i)} \in \square_i$. The integral in Eq. 5 is then approximated by the following sum:

$$p_F \approx \sum_{x^{(i)} \in F} \pi\left(x^{(i)}\right) \mathrm{vol}\left(\square_i\right), \tag{6}$$

where $\mathrm{vol}(\square_i)$ denotes the volume of $\square_i$ and summation is taken over all failure points $x^{(i)}$. Since it is not known in advance whether a given point is a failure point or not (the failure domain F is not known explicitly), to compute the sum in Eq. 6, the failure criterion in Eq. 4 must be checked for all x(i). Therefore, the approximation in Eq. 6 becomes

$$p_F \approx \sum_{i=1}^{N} I_F\left(x^{(i)}\right) \pi\left(x^{(i)}\right) \mathrm{vol}\left(\square_i\right), \tag{7}$$

where $I_F(x)$ stands for the indicator function, i.e.,

$$I_F(x) = \begin{cases} 1, & \text{if } x \in F, \\ 0, & \text{if } x \notin F. \end{cases} \tag{8}$$

If n denotes the number of intervals each dimension of the input space is partitioned into, then the total number of terms in Eq. 7 is $N = n^d$. Therefore, the computational effort of numerical integration grows exponentially with the number of dimensions d. In engineering reliability problems, the dimension of the input space is typically very large (e.g., when the stochastic load time history is discretized in time). For example, $d \sim 10^3$ is not unusual in the reliability literature. This makes numerical integration computationally infeasible.
Over the past few decades, many different methods for solving the engineering reliability problem (5) have been developed. In general, the proposed reliability methods can be classified into three categories, namely:

1. Analytic methods
   are based on the Taylor-series expansion of the performance function, e.g., the first-order reliability method (FORM) and the second-order reliability method (SORM) (Ditlevsen and Madsen 1996; Madsen et al. 2006; Melchers 1999).
2. Surrogate methods
   are based on a functional surrogate of the performance function, e.g., the response surface method (RSM) (Faravelli 1989; Schuëller et al. 1989; Bucher 1990), Neural Networks (Papadrakakis et al. 1996), support vector machines (Hurtado and Alvarez 2003), and other methods (Hurtado 2004).
3. Monte Carlo simulation methods
   , among which are Importance Sampling (Engelund and Rackwitz 1993), Importance Sampling using Elementary Events (Au and Beck 2001b), Radial-based Importance Sampling (Grooteman 2008), Adaptive Linked Importance Sampling (Katafygiotis and Zuev 2007), Directional Simulation (Ditlevsen and Madsen 1996), Line Sampling (Koutsourelakis et al. 2004), Auxiliary Domain Method (Katafygiotis et al. 2007), Horseracing Simulation (Zuev and Katafygiotis 2011), and Subset Simulation (Au and Beck 2001a).

Subset Simulation is thus a reliability method which is based on (advanced) Monte Carlo simulation.

The Direct Monte Carlo Method

The Monte Carlo method, referred in this entry as Direct Monte Carlo (DMC), is a statistical sampling technique that has been originally developed by Stan Ulam, John von Neumann, Nick Metropolis (who actually suggested the name "Monte Carlo" (Metropolis 1987)), and their collaborators for solving the problem of neutron diffusion and other problems in mathematical physics (Metropolis and Ulam 1949). From a mathematical point of view, DMC allows to estimate the expected value of a quantity of interest. More specifically, suppose the goal is to evaluate $\mathbb{E}_\pi[h(x)]$, that is, an expectation of a function $h : \chi \to \mathbb{R}$ with respect to the PDF $\pi(x)$,

$$\mathbb{E}_\pi\left[h(x)\right] = \int_\chi h(x)\pi(x)dx. \tag{9}$$

The idea behind DMC is a straightforward application of the law of large numbers that states that if $x^{(1)}$, $x^{(2)}$,… are i.i.d. (independent and identically distributed) from the PDF π(x), then the empirical average $\frac{1}{N}\sum_{i=1}^{N} h\left(x^{(i)}\right)$ converges to the true value $\mathbb{E}_\pi[h(x)]$ as N goes to + ∞. Therefore, if the number of samples N is large enough, then $\mathbb{E}_\pi[h(x)]$ can be accurately estimated by the corresponding empirical average:

$$\mathbb{E}_\pi\left[h(x)\right] \approx \frac{1}{N}\sum_{i=1}^{N} h\left(x^{(i)}\right). \tag{10}$$

The relevance of DMC to the reliability problem (5) follows from a simple observation that the failure probability p_F can be written as an expectation of the indicator function (8), namely,

$$p_F = \int_F \pi(x)dx = \int_\chi I_F(x)\pi(x)dx = \mathbb{E}_\pi\left[I_F(x)\right], \tag{11}$$

where χ denotes the entire input x-space. Therefore, the failure probability can be estimated using the DMC method (10) as follows:

$$p_F \approx \widehat{p}_F^{\,\mathrm{DMC}} = \frac{1}{N}\sum_{i=1}^{N} I_F\left(x^{(i)}\right), \tag{12}$$

where $x^{(1)}$,…0$x^{(N)}$ are i.i.d. samples from π(x).
The DMC estimate of p_F is thus just the ratio of the total number of failure samples $\sum_{i=1}^{N} I_F(x^{(i)})$, i.e., samples that produce system failure according to the system model, to the total number of samples, N. Note that $\widehat{p}_F^{\,\mathrm{DMC}}$ is an unbiased random estimate of the failure probability, that is, on average, $\widehat{p}_F^{\,\mathrm{DMC}}$ equals to p_F. Mathematically, this means that $\mathbb{E}\left[\widehat{p}_F^{\,\mathrm{DMC}}\right] = p_F$. Indeed, using the fact that $x^{(i)} \sim \pi(x)$ and in Eq. 11,

$$\mathbb{E}\left[\widehat{p}_F^{\,\mathrm{DMC}}\right] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} I_F\left(x^{(i)}\right)\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[I_F\left(x^{(i)}\right)\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_\pi\left[I_F(x)\right] =$$

The main advantage of DMC over numerical integration is that its accuracy does not depend on the dimension d of the input space. In reliability analysis, the standard measure of accuracy of an unbiased estimate $\widehat{p}_F$ of the failure probability is its coefficient of variation (c.o.v.) $\delta(\widehat{p}_F)$, which is defined as the ratio of the standard deviation to the expected value of $\widehat{p}_F$, i.e., $\delta(\widehat{p}_F) = \sqrt{\mathbb{V}[\widehat{p}_F]}/\mathbb{E}[\widehat{p}_F]$, where $\mathbb{V}$ denotes the variance. The smaller the c.o.v. $\delta(\widehat{p}_F)$, the more accurate the estimate $\widehat{p}_F$ is. It is straightforward to calculate the variance of the DMC estimate:

$$\mathbb{V}\left[\widehat{p}_F^{\mathrm{DMC}}\right] = \mathbb{V}\left[\frac{1}{N}\sum_{i=1}^{N} I_F\left(x^{(i)}\right)\right] = \frac{1}{N^2}\sum_{i=1}^{N}\mathbb{V}\left[I_F\left(x^{(i)}\right)\right]$$
$$= \frac{1}{N^2}\sum_{i=1}^{N}\left(\mathbb{E}\left[I_F\left(x^{(i)}\right)^2\right] - \mathbb{E}\left[I_F\left(x^{(i)}\right)\right]^2\right) = \frac{1}{N^2}\sum_{i=1}^{N}\left(p_F - p_F^2\right) = \frac{p_F}{}$$

Here, the identity $I_F(x)^2 = I_F(x)$ was used. Using Eqs. 13 and 14, the c.o.v. of the DMC estimate can be calculated:

$$\delta\left(\widehat{p}_F^{\mathrm{DMC}}\right) = \frac{\sqrt{\mathbb{V}\left[\widehat{p}_F^{\mathrm{DMC}}\right]}}{\mathbb{E}\left[\widehat{p}_F^{\mathrm{DMC}}\right]} = \sqrt{\frac{1-p_F}{Np_F}}. \tag{15}$$

This result shows that $\delta\left(\widehat{p}_F^{\mathrm{DMC}}\right)$ depends only on the failure probability p_F and the total number of samples N and does not depend on the dimension d of the input space. Therefore, unlike numerical integration, the DMC method does not suffer from the "curse of dimensionality," i.e., from an exponential increase in volume associated with adding extra dimensions, and is able to handle problems of high dimension.

Nevertheless, DMC has a serious drawback: it is inefficient in estimating small failure probabilities. For typical engineering reliability problems, the failure probability p_F is very small, $p_F \ll 1$. In other words, the system is usually assumed to be designed properly, so that its failure is a rare event. In the reliability literature, $p_F \sim 10^{-2} - 10^{-9}$ have been considered. If p_F is very small, then it follows from Eq. 15 that

$$\delta\left(\widehat{p}_F^{\mathrm{DMC}}\right) \approx \frac{1}{\sqrt{Np_F}}. \tag{16}$$

This means that the number of samples N needed to achieve an acceptable level of accuracy is inverse proportional to p_F, and therefore very large, $N \propto 1/p_F \gg 1$. For example, if $p_F = 10^{-4}$ and the c.o.v. of 10 % is desirable, then $N = 10^6$ samples are required. Note, however, that each evaluation of $I_F(x^{(i)})$, i = 1,…, N, in Eq. 12 requires a system analysis to be performed to check whether the sample $x^{(i)}$ is a failure sample. As it has been already mentioned in section "Engineering Reliability Problem," the computation effort for the system analysis, i.e., computation of the performance function g(x), is significant (usually involves the FEM method). As a result, the DMC method becomes excessively costly and practically inapplicable for reliability analysis. This deficiency of DMC has motivated research to develop more advanced simulation algorithms for efficient estimation of small failure probabilities in high dimensions.

Remark 1

It is important to highlight, however, that even though DMC cannot be routinely used for reliability problems (too expensive), it is a very robust method, and it is often used as a check on other reliability methods.

Preprocessing: Transformation of Input Variables

Many reliability methods, including Subset Simulation, assume that the input variables x are independent. This assumption, however, is not a limitation, since in simulation one always starts from independent variables to generate the dependent input variables. Furthermore, for convenience, it is often assumed that x are i.i.d. Gaussian. If this is not the case, a "preprocessing" step that transforms x to i.i.d. Gaussian variables z must be undertaken. The transformation from

x to z can be performed in several ways depending on the available information about the input variables. In the simplest case, when x are independent Gaussians, $x_k \sim \mathcal{N}\left(\cdot \middle| \mu_k, \sigma_k^2\right)$, where μ_k and σ_k^2 are, respectively, the mean and variance of x_k , the necessary transformation is standardization:

$$z_k = \frac{x_k - \mu_k}{\sigma_k}. \tag{17}$$

In other cases, more general techniques should be used, such as the Rosenblatt transformation (Rosenblatt 1952) and the Nataf transformation (Nataf 1962). To avoid introduction of additional notation, hereinafter, it is assumed without loss of generality that the vector x has been already transformed and it follows the standard multivariate Gaussian distribution,

$$\pi(x_1, \ldots, x_d) = \prod_{k=1}^{d} \phi(x_k), \tag{18}$$

where ϕ(·) denotes the standard Gaussian PDF,

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}. \tag{19}$$

The Subset Simulation Method

Unlike Direct Monte Carlo, where all computational resources are directly spent on sampling the input space $x^{(1)}, \ldots, x^{(N)} \sim \pi(\cdot)$ and computing the values of the performance function $g(x^{(1)}), \ldots, g(x^{(N)})$, Subset Simulation first "probes" the input space χ by generating a relatively small number of i.i.d samples $x_0^{(1)}, \ldots, x_0^{(n)} \sim \pi(x)$, $n < N$, and computing the corresponding system responses $y_0^{(1)} = g(x_0^{(1)}), \ldots, y_0^{(n)} = g(x_0^{(n)})$. Here, the subscript 0 indicates the 0^{th} stage of the algorithm. Since F is a rare event and n is relatively small, it is very likely that none of the samples $x_0^{(1)}, \ldots, x_0^{(n)}$ belongs to F, that is, $y_0^{(i)} < y^*$ for all $i = 1, \ldots, n$. Nevertheless, these Monte Carlo samples contain some useful information about the failure domain that can be utilized. To keep the notation simple, assume that $y_0^{(1)}, \ldots, y_0^{(n)}$ are arranged in the decreasing order, i.e., $y_0^{(1)} \geq \ldots \geq y_0^{(n)}$ (it is always possible to achieve this by renumbering $x_0^{(1)}, \ldots, x_0^{(n)}$ if needed). Then, $x_0^{(1)}$ and $x_0^{(n)}$ are, respectively, the closest to failure and the safest samples among $x_0^{(1)}, \ldots, x_0^{(n)}$, since $y_0^{(1)}$ and $y_0^{(n)}$ are the largest and the smallest responses. In general, the smaller the i, the closer to failure the sample $x_0^{(i)}$ is. This is shown schematically in Fig. 1.

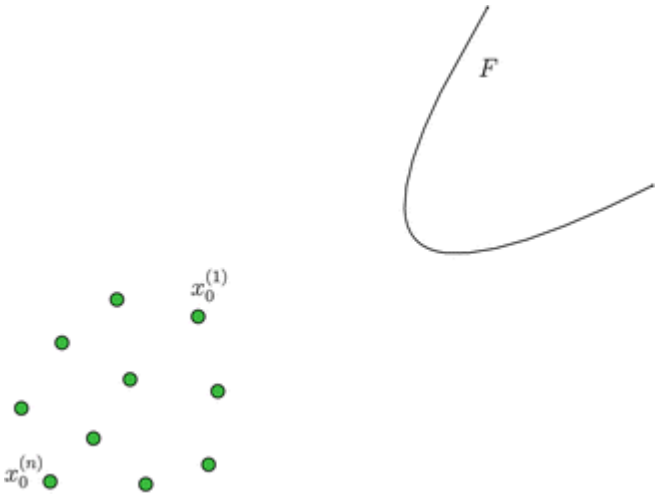

Fig. 1
Monte Carlo samples $x_0^{(1)}, \ldots, x_0^{(n)}$ and the failure domain F. $x_0^{(1)}$ and $x_0^{(n)}$ are, respectively, the closest to failure and the safest samples among $x_0^{(1)}, \ldots, x_0^{(n)}$

Let $p \in (0, 1)$ be any number such that np is integer. By analogy with Eq. 4, define the first intermediate failure domain F_1 as follows:

$$F_1 = \{x : g(x) > y_1^*\}, \tag{20}$$

where

$$y_1^* = \frac{y_0^{(np)} + y_0^{(np+1)}}{2}. \tag{21}$$

In other words, F_1 is the set of inputs that lead to the exceedance of the relaxed threshold $y_1^* < y^*$. Note that by construction, samples $x_0^{(1)}, \ldots, x_0^{(np)}$ belong to F_1, while $x_0^{(np+1)}, \ldots, x_0^{(n)}$ do not. As a consequence, the Direct Monte Carlo estimate for the probability of F_1 which is based on samples $x_0^{(1)}, \ldots, x_0^{(n)}$ is automatically equal to p,

$$\mathbb{P}(F_1) \approx \frac{1}{n}\sum_{i=1}^{n} I_{F_1}\left(x_0^{(i)}\right) = p. \tag{22}$$

The value $p = 0.1$ is often used in the literature, which makes F_1 a relatively frequent event. Figure 2 illustrates the definition of F_1.

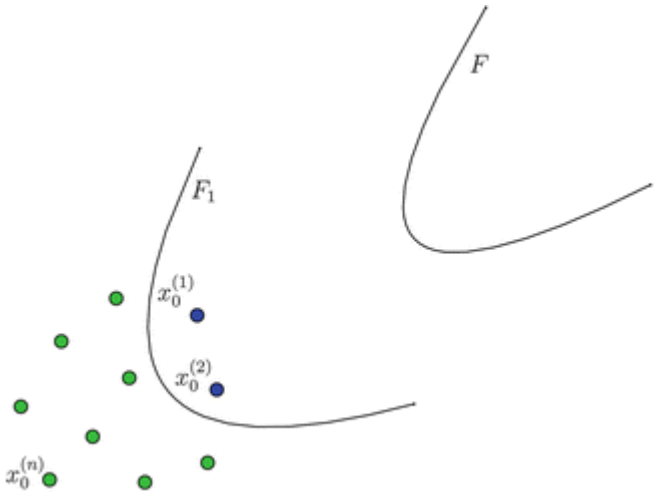

Fig. 2
The first intermediate failure domain F_1. In this schematic illustration, n = 10, p = 0:2, so that there are exactly np = 2 Monte Carlo samples in F1, $x_0^{(1)}, x_0^{(2)} \in F_1$

The first intermediate failure domain F_1 can be viewed as a (very rough) conservative approximation to the target failure domain F. Since $F \subset F_1$, the failure probability p_F can be written as a product:

$$p_F = \mathbb{P}(F_1)\mathbb{P}\left(F\middle|F_1\right), \tag{23}$$

where $\mathbb{P}(F|F_1)$ is the conditional probability of F given F_1. Therefore, in view of Eq. 22, the problem of estimating p_F is reduced to estimating the conditional probability $\mathbb{P}(F|F_1)$.

In the next stage, instead of generating samples in the whole input space (like in DMC), the SS algorithm aims to populate F_1. Specifically, the goal is to generate samples $x_1^{(1)}, \ldots, x_1^{(n)}$ from the conditional distribution

$$\pi\left(x\middle|F_1\right) = \frac{\pi(x)I_{F_1}(x)}{\mathbb{P}(F_1)} = \frac{I_{F_1}(x)}{\mathbb{P}(F_1)}\prod_{k=1}^{d}\phi(x_k). \tag{24}$$

First of all, note that samples $x_0^{(1)}, \ldots, x_0^{(np)}$ not only belong to F_1 but are also distributed according to $\pi(\cdot|F_1)$. To generate the remaining (n − np) samples from $\pi(\cdot|F_1)$, which, in general, is not a trivial task, Subset Simulation uses the so-called Modified Metropolis algorithm (MMA). MMA belongs to the class of Markov chain Monte Carlo (MCMC) algorithms (Liu 2001; Robert and Casella 2004), which are techniques for sampling from complex probability distributions that cannot be sampled directly, at least not efficiently. MMA is based on the original Metropolis algorithm (Metropolis et al. 1953) and specifically tailored for sampling from the conditional distributions of the form (24).

Modified Metropolis Algorithm

Let $x \sim \pi(\cdot|F_1)$ be a sample from the conditional distribution $\pi(\cdot|F_1)$. The Modified Metropolis algorithm generates another sample $\tilde{x}$ from $\pi(\cdot|F_1)$ as follows:

1. Generate a "candidate" sample ξ: For each coordinate k = 1,…, d,
    1. Sample $\eta_k \sim q_k(\cdot|x_k)$, where $q_k(\cdot|x_k)$, called the proposal distribution, is a univariate PDF for η_k centered at x_k with the symmetry property $q_k(\eta_k|x_k) = q_k(x_k|\eta_k)$. For example, the proposal

distribution can be a Gaussian PDF with mean x_k and variance σ_k^2,

$$q_k\left(\eta_k\middle|x_k\right) = \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{(\eta_k - x_k)^2}{2\sigma_k^2}\right), \tag{25}$$

or it can be a uniform distribution over $[x_k - \alpha, x_k + \alpha]$, for some $\alpha \geq 0$.

2. Compute the acceptance ratio

$$r_k = \frac{\phi(\eta_k)}{\phi(x_k)}. \tag{26}$$

3. Define the kth coordinate of the candidate sample by accepting or rejecting η_k,

$$\xi_k = \begin{cases} \eta_k, & \text{with probability } \min\{1, r_k\}, \\ x_k, & \text{with probability } 1 - \min\{1, r_k\}. \end{cases} \tag{27}$$

1. Accept or reject the candidate sample ξ by setting

$$\tilde{x} = \begin{cases} \xi, & \text{if } \xi \in F_1, \\ x, & \text{if } \xi \notin F_1. \end{cases} \tag{28}$$

The Modified Metropolis algorithm is schematically illustrated in Fig. 3.

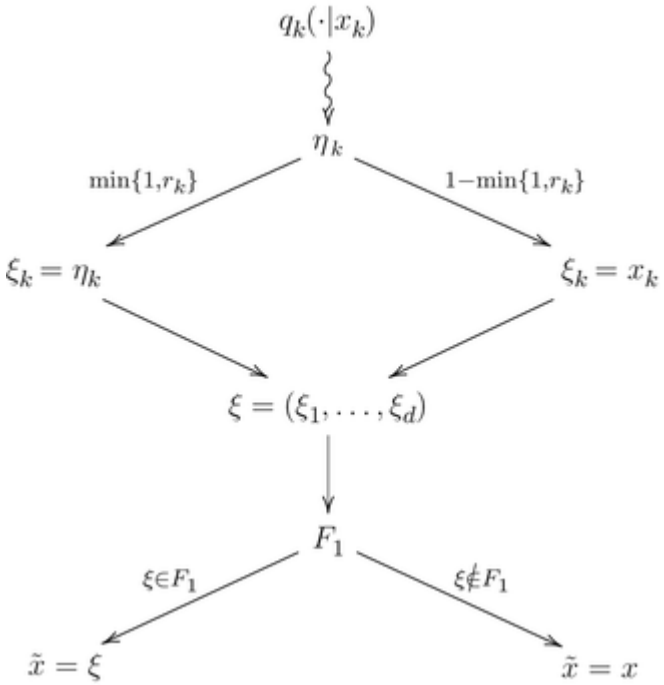

Fig. 3
Modified Metropolis algorithm

It can be shown that the sample $\tilde{x}$ generated by MMA is indeed distributed according to $\pi(\cdot|F_1)$. If the candidate sample ξ is rejected in Eq. 28, then $\tilde{x} = x \sim \pi\left(\cdot\middle|F_1\right)$ and there is nothing to prove. Suppose now that ξ is

accepted, $\tilde{x} = \xi$, so that the move from x to $\tilde{x}$ is a proper transition between two distinct points in F_1. Let f(·) denote the PDF of $\tilde{x}$ (the goal is to show that $f(\tilde{x}) = \pi\left(\tilde{x}\middle|F_1\right)$. Then

$$f\left(\tilde{x}\right) = \int_{F_1} \pi\left(x\middle|F_1\right) t\left(\tilde{x}\middle|x\right) dx, \tag{29}$$

where $t\left(\tilde{x}\middle|x\right)$ is the transition PDF from x to $\tilde{x} \neq x$. According to the first step of MMA, coordinates of $\tilde{x} = \xi$ are generated independently, and therefore $t\left(\tilde{x}\middle|x\right)$ can be expressed as a product,

$$t\left(\tilde{x}\middle|x\right) = \prod_{k=1}^{d} t_k\left(\tilde{x}_k\middle|x_k\right), \tag{30}$$

where $t_k\left(\tilde{x}_k\middle|x_k\right)$ is the transition PDF for the k [th] coordinate $\tilde{x}_k$. Combining Eqs. 24, 29, and 30 gives

$$f\left(\tilde{x}\right) = \int_{F_1} \frac{I_{F_1}(x)}{\mathbb{P}\left(F_1\right)} \prod_{k=1}^{d} \phi\left(x_k\right) \prod_{k=1}^{d} t_k\left(\tilde{x}_k\middle|x_k\right) dx = \frac{1}{\mathbb{P}\left(F_1\right)} \int_{F_1} \prod_{k=1}^{d} \phi\left(x_k\right) t_k\left(\tilde{x}_k\middle|x\right.$$

The key to the proof of $f(\tilde{x}) = \pi\left(\tilde{x}\middle|F_1\right)$ is to demonstrate that $\phi(x_k)$ and $t_k\left(\tilde{x}_k\middle|x_k\right)$ satisfy the so-called detailed balance equation,

$$\phi(x_k) t_k\left(\tilde{x}_k\middle|x_k\right) = \phi(\tilde{x}_k) t_k\left(x_k\middle|\tilde{x}_k\right). \tag{32}$$

If $\tilde{x}_k = x_k$, then Eq. 32 is trivial. Suppose that $\tilde{x}_k \neq x_k$, that is, $\tilde{x}_k = \xi_k = \eta_k$ in Eq. 27. The actual transition PDF $t_k\left(\tilde{x}_k\middle|x_k\right)$ from x_k to $\tilde{x}_k \neq x_k$ differs from the proposal PDF $q_k\left(\tilde{x}_k\middle|x_k\right)$ because the acceptance-rejection step in Eq. 27 is involved. To actually make the move from x_k to $\tilde{x}_k$, one needs not only to generate $\tilde{x}_k \sim q_k\left(\cdot\middle|x_k\right)$ but also to accept it with probability $\min\left\{1, \frac{\phi(\tilde{x}_k)}{\phi(x_k)}\right\}$. Therefore,

$$t_k\left(\tilde{x}_k\middle|x_k\right) = q_k\left(\tilde{x}_k\middle|x_k\right) \min\left\{1, \frac{\phi\left(\tilde{x}_k\right)}{\phi\left(x_k\right)}\right\}, \quad \tilde{x}_k \neq x_k. \tag{33}$$

Using Eq. 33, the symmetry property of the proposal PDF, $q_k\left(\tilde{x}_k\middle|x_k\right) = q_k\left(x_k\middle|\tilde{x}_k\right)$, and the identity $a\min\{1, \frac{b}{a}\} = b\min\{1, \frac{a}{b}\}$ for any a, b > 0,

$$\begin{aligned}\phi(x_k)\, t_k\left(\tilde{x}_k \middle| x_k\right) &= q_k\left(\tilde{x}_k \middle| x_k\right)\phi(x_k)\min\left\{1, \tfrac{\phi(\tilde{x}_k)}{\phi(x_k)}\right\} \\ &= q_k\left(x_k \middle| \tilde{x}_k\right)\phi(\tilde{x}_k)\min\left\{1, \tfrac{\phi(x_k)}{\phi(\tilde{x}_k)}\right\} = \phi(\tilde{x}_k)\, t_k\left(x_k \middle| \tilde{x}_k\right),\end{aligned} \tag{34}$$

and the detailed balance in Eq. 32 is thus established. The rest is a straightforward calculation:

$$f(\tilde{x}) = \frac{1}{\mathbb{P}(F_1)}\int_{F_1}\prod_{k=1}^{d}\phi(\tilde{x}_k)\, t_k\left(x_k \middle| \tilde{x}_k\right)dx = \frac{1}{\mathbb{P}(F_1)}\prod_{k=1}^{d}\phi(\tilde{x}_k)\int_{F_1} t\left(x \middle| \tilde{x}\right)dx =$$

since the transition PDF $t\left(x \middle| \tilde{x}\right)$ integrates to 1, and $I_{F_1}(\tilde{x}) = 1$.

Remark 2

A mathematically more rigorous proof of the Modified Metropolis algorithm is given in (Zuev et al. 2012).

Remark 3

It is worth mentioning that although the independence of input variables is crucial for the applicability of MMA, and thus for Subset Simulation, they need not be identically distributed. In other words, instead of Eq. 18, the joint PDF π(·) can have a more general form, $\pi(x) = \prod_{k=1}^{d}\pi_k(x_k)$, where $\pi_k(\cdot)$ is the marginal distributions of x_k which is not necessarily Gaussian. In this case, the expression for the acceptance ratio in Eq. 26 must be replaced by $r_k = \frac{\pi_k(\eta_k)}{\pi_k(x_k)}$.

Subset Simulation at Higher Conditional Levels

Given $x_0^{(1)}, \ldots, x_0^{(np)} \sim \pi(\cdot|F_1)$, it is clear now how to generate the remaining (n − np) samples from $\pi(\cdot|F_1)$. Namely, starting from each $x_0^{(i)}$, i = 1,…,np, the SS algorithm generates a sequence of $\left(\frac{1}{p}, -, 1\right)$ new MCMC samples $x_0^{(i)} = x_{0,0}^{(i)} \mapsto x_{0,1}^{(i)} \mapsto \ldots \mapsto x_{0,\frac{1}{p}-1}^{(i)}$ using the Modified Metropolis transition rule described above. Note that when $x_{0,j}^{(i)}$ is generated, the previous sample $x_{0,j-1}^{(i)}$ is used as an input for the transition rule. The sequence $x_{0,0}^{(i)}, x_{0,1}^{(i)}, \ldots, x_{0,\frac{1}{p}-1}^{(i)}$ is called a Markov chain with the stationary distribution $\pi(\cdot|F_1)$, and $x_{0,0}^{(i)} = x_0^{(i)}$ is often referred to as the "seed" of the Markov chain.

To simplify the notation, denote samples $\left\{x_{0,j}^{(i)}\right\}_{j=0,\ldots,\frac{1}{p}-1}^{i=1,\ldots,np}$ by $\{x_1^{(1)}, \ldots, x_1^{(n)}\}$. The subscript 1 indicates that the MCMC samples $x_1^{(1)}, \ldots, x_1^{(n)} \sim \pi(\cdot|F_1)$ are generated at the first conditional level of the SS algorithm. These conditional samples are schematically shown in Fig. 4. Also assume that the corresponding system responses $y_1^{(1)} = g(x_1^{(1)}), \ldots, y_1^{(n)} = g(x_1^{(n)})$ are arranged in the decreasing order, i.e., $y_1^{(1)} \geq \ldots \geq y_1^{(n)}$. If the failure event F is rare enough, that is, if p_F is sufficiently small, then it is very likely that none of the samples $x_1^{(1)}, \ldots, x_1^{(n)}$ belongs to F, i.e., $y_1^{(i)} < y^*$ for all i = 1,…, n. Nevertheless, these MCMC samples can be used in the similar way the Monte Carlo samples $x_0^{(1)}, \ldots, x_0^{(n)}$ were used.

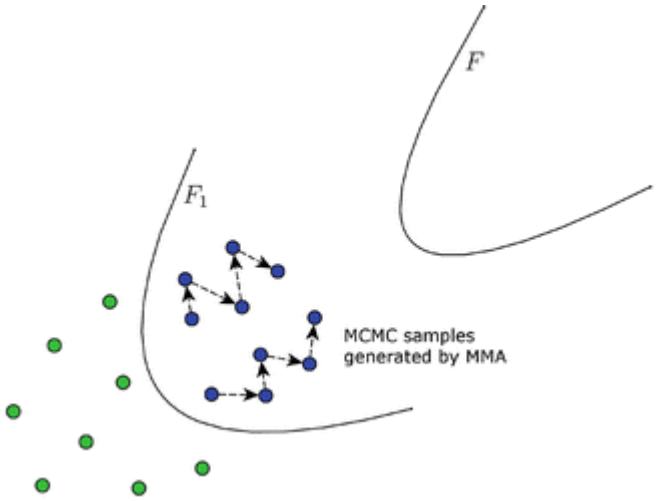

Fig. 4
MCMC samples generated by the Modified Metropolis algorithm at the first conditional level of Subset Simulation

By analogy with Eq. 20, define the second intermediate failure domain F_2 as follows:

$$F_2 = \{x : g(x) > y_2^*\}, \tag{36}$$

where

$$y_2^* = \frac{y_1^{(np)} + y_1^{(np+1)}}{2}. \tag{37}$$

Note that $y_2^* > y_1^*$ since $y_1^{(i)} > y_1^*$ for all i = 1,…,n. This means that $F \subset F_2 \subset F_1$, and therefore, F_2 can be viewed as a conservative approximation to F which is still rough, yet more accurate than F_1. Figure 5 illustrates the definition of F_2. By construction, samples $x_1^{(1)}, \ldots, x_1^{(np)}$ belong to F_2, while $x_1^{(np+1)}, \ldots, x_1^{(n)}$ do not. As a result, the estimate for the conditional probability of F_2 given F_1 which is based on samples $x_1^{(1)}, \ldots, x_1^{(n)} \sim \pi(\cdot|F_1)$ is automatically equal to p,

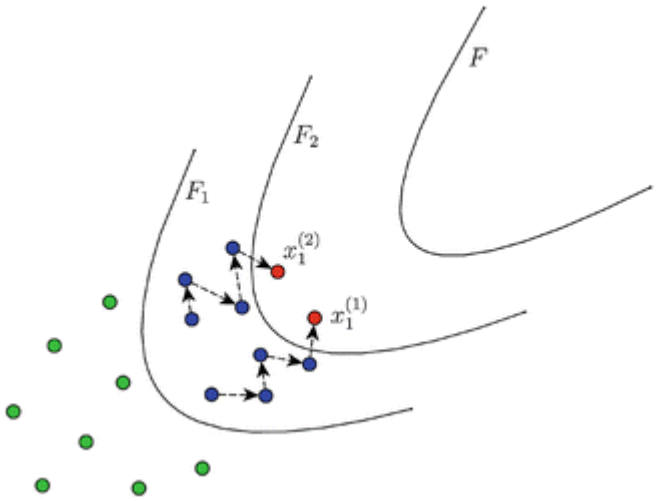

Fig. 5
The second intermediate failure domain F2. In this schematic illustration, n = 10, p = 0.2, so that there are exactly np = 2 MCMC samples in F_2, $x_1^{(1)}, x_1^{(2)} \in F_2$

$$\mathbb{P}\left(F_2\middle|F_1\right) \approx \frac{1}{n}\sum_{i=1}^{n} I_{F_2}\left(x_1^{(i)}\right) = p. \tag{38}$$

Since $F \subset F_2 \subset F_1$, the conditional probability $\mathbb{P}(F|F_1)$ that appears in Eq. 23 can be expressed as a product:

$$\mathbb{P}\left(F\middle|F_1\right) = \mathbb{P}\left(F_2\middle|F_1\right)\mathbb{P}\left(F\middle|F_2\right). \tag{39}$$

Combining Eqs. 23 and 39 gives the following expression for the failure probability:

$$p_F = \mathbb{P}(F_1)\mathbb{P}\left(F_2\middle|F_1\right)\mathbb{P}\left(F\middle|F_2\right). \tag{40}$$

Thus, in view of Eqs. 22 and 38, the problem of estimating p_F is now reduced to estimating the conditional probability $\mathbb{P}(F|F_2)$.

In the next step, as one may have already guessed, the Subset Simulation algorithm populates F_2 by generating MCMC samples $x_2^{(1)}, \ldots, x_2^{(n)}$ from $\pi(\cdot|F_2)$ using the Modified Metropolis algorithm, defines the third intermediate failure domain $F_3 \subset F_2$ such that $\mathbb{P}\left(F_3\middle|F_2\right) \approx \frac{1}{n}\sum_{i=1}^{n} I_{F_3}\left(x_2^{(i)}\right) = p$, and reduces the original problem of estimating the failure probability p_F to estimating the conditional probability $\mathbb{P}(F|F_3)$ by representing $p_F = \mathbb{P}(F_1)\mathbb{P}(F_2|F_1)\mathbb{P}(F_3|F_2)\mathbb{P}(F|F_3)$. The algorithm proceeds in this way until the target failure domain F has been sufficiently sampled so that the conditional probability $\mathbb{P}(F|F_L)$ can be accurately estimated by $\frac{1}{n}\sum_{i=1}^{n} I_F\left(x_L^{(i)}\right)$, where F_L is the Lth intermediate failure domain and $x_L^{(1)}, \ldots, x_L^{(n)} \sim \pi(\cdot|F_L)$ are the MCMC samples generated at the Lth conditional level. Subset Simulation can thus be viewed as a method that decomposes the rare failure event F into a sequence of progressively "less-rare" nested events, $F \subset F_L \subset \ldots \subset F_1$, where all intermediate failure events $F_1, \ldots, F_L$ are constructed adaptively by appropriately relaxing the value of the critical threshold $y_1^* < \ldots < y_L^* < y^*$.

Stopping Criterion

In what follows, the stopping criterion for Subset Simulation is described in detail. Let $n_F(l)$ denote the number of failure samples at the l^{th} level, that is,

$$n_F(l) = \sum_{i=1}^{n} I_F\left(x_l^{(i)}\right), \tag{41}$$

where $x_l^{(1)}, \ldots, x_l^{(n)} \sim \pi(\cdot|F_l)$. Since F is a rare event, it is very likely that $n_F(l) = 0$ for the first few conditional levels. As l gets larger, however, $n_F(l)$ starts increasing since F_l, which approximates F "from above," shrinks closer to F. In general, $n_F(l) \geq n_F(l-1)$, since $F \subset F_l \subset F_{l-1}$ and the np closest to F samples among $x_{l-1}^{(1)}, \ldots, x_{l-1}^{(n)}$ are present among $x_l^{(1)}, \ldots, x_l^{(n)}$. At conditional level l, the failure probability p_F is expressed as a product,

$$p_F = \mathbb{P}(F_1)\mathbb{P}\left(F_2\middle|F_1\right)\ldots\mathbb{P}\left(F_l\middle|F_{l-1}\right)\mathbb{P}\left(F\middle|F_l\right). \tag{42}$$

Furthermore, the adaptive choice of intermediate critical thresholds $y_1^*, \ldots, y_l^*$ guarantees that the first l factors in Eq.

42 approximately equal to p, and, thus,

$$p_F \approx p^l \cdot \mathbb{P}\left(F \middle| F_l\right). \tag{43}$$

Since there are exactly $n_F(l)$ failure samples at the l^{th} level, the estimate of the last conditional probability in Eq. 42 which is based on samples $x_l^{(1)}, \ldots, x_l^{(n)} \sim \pi(\cdot|F_l)$ is given by

$$\mathbb{P}\left(F \middle| F_l\right) \approx \frac{1}{n}\sum_{i=1}^{n} I_F\left(x_l^{(i)}\right) = \frac{n_F(l)}{n}. \tag{44}$$

If $n_F(l)$ is sufficiently large, i.e., the conditional event $(F|F_l)$ is not rare, then the estimate in Eq. 44 is fairly accurate. This leads to the following stopping criterion:

- If $\frac{n_F(l)}{n} \geq p$, i.e., there are at least np failure samples among $x_l^{(1)}, \ldots, x_l^{(n)}$, then Subset Simulation stops: the current conditional level l becomes the last level, L = l, and the failure probability estimate derived from Eqs. 43 and 44 is

$$p_F \approx \widehat{p}_F^{SS} = p^L \frac{n_F(L)}{n}. \tag{45}$$

- If $\frac{n_F(l)}{n} < p$, i.e., there are less than np failure samples among $x_l^{(1)}, \ldots, x_l^{(n)}$, then the algorithm proceeds by defining the next intermediate failure domain $F_{l+1} = \{x : g(x) > y_{l+1}^*\}$, where $y_{l+1}^* = (y_l^{(np)} + y_l^{(np+1)})/2$, and expressing $\mathbb{P}(F|F_l)$ as a product $\mathbb{P}(F|F_l) = \mathbb{P}(F_{l+1}|F_l)\mathbb{P}(F|F_{l+1}) \approx p \cdot \mathbb{P}(F|F_{l+1})$.

The described stopping criterion guarantees that the estimated values of all factors in the factorization $p_F = \mathbb{P}(F_1)\mathbb{P}(F_2|F_1)\ldots\mathbb{P}(F_L|F_{L-1})\mathbb{P}(F|FL)$ are not smaller than p. If p is relatively large (p = 0.1 is often used in applications), then it is likely that the estimates $\mathbb{P}(F_1) \approx p$, $\mathbb{P}(F_2|F_1) \approx p, \ldots, \mathbb{P}(F_L|F_{L-1}) \approx p$, and $\mathbb{P}\left(F \middle| F_L\right) \approx \frac{n_F(L)}{n} (\leq p)$ are accurate even when the sample size n is relatively small. As a result, the SS estimate in Eq. 45 is also accurate in this case. This provides an intuitive explanation as to why Subset Simulation is efficient in estimating small probabilities of rare events. For a detailed discussion of error estimation for the SS method, the reader is referred to Au and Wang (2014).

Implementation Details

In the rest of this section, the implementation details of Subset Simulation are discussed. The SS algorithm has two essential components that affect its efficiency: the parameter p and the set of univariate proposal PDFs $\{q_k\}$, k = 1,…,d.

Level Probability

The parameter p, called the level probability in Au and Wang (2014) and the conditional failure probability in Zuev et al. (2012), governs how many intermediate failure domains F_l are needed to reach the target failure domain F. As it follows from Eq. 45, a small value of p leads to a fewer total number of conditional levels L. But at the same time, it results in a large number of samples n needed at each conditional level l for accurate determination of F_l (i.e., determination of y_l^*) that satisfies $\frac{1}{n}\sum_{i=1}^{n} I_{F_l}\left(x_{l-1}^{(i)}\right) = p$. In the extreme case when $p \leq p_F$, no levels are needed, L = 0, and Subset Simulation reduces to the Direct Monte Carlo method. On the other hand, increasing the value of p will mean that fewer samples are needed at each conditional level, but it will increase the total number of levels L. The choice of the

level probability p is thus a trade-off between the total number of level L and the number of samples n at each level. In the original paper (Au and Beck 2001a), it has been found that the value p = 0.1 yields good efficiency. The latter studies (Au and Wang 2014; Zuev et al. 2012), where the c.o.v. of the SS estimate $\widehat{p}_F^{SS}$ has been analyzed, confirmed that p = 0.1 is a nearly optimal value of the level probability.

Proposal Distributions

The efficiency and accuracy of Subset Simulation also depends on the set of univariate proposal PDFs $\{q_k\}$, k = 1,…,d, that are used within the Modified Metropolis algorithm for sampling from the conditional distributions $\pi(\cdot|F_l)$. To see this, note that in contract to the Monte Carlo samples $x_0^{(1)}, \ldots, x_0^{(n)} \sim \pi(\cdot)$ which are i.i.d., the MCMC samples $x_l^{(1)}, \ldots, x_l^{(n)} \sim \pi(\cdot|F_l)$ are not independent for l ≥ 1, since the MMA transition rule uses $x_l^{(i)} \sim \pi(\cdot|F_l)$ to generate $x_l^{(i+1)} \sim \pi(\cdot|F_l)$. This means that although these MCMC samples can be used for statistical averaging as if they were i.i.d., the efficiency of the averaging is reduced if compared with the i.i.d. case (Doob 1953). Namely, the more correlated $x_l^{(1)}, \ldots, x_l^{(n)}$ are, the slower is the convergence of the estimate $P\left(F_{l+1}\middle|F_l\right) \approx \frac{1}{n}\sum_{i=1}^{n} I_{F_{l+1}}\left(x_l^{(i)}\right)$, and, therefore, the less efficient it is. The correlation between samples $x_l^{(1)}, \ldots, x_l^{(n)}$ is due to proposal PDFs $\{q_k\}$, which govern the generation of the next sample $x_l^{(i+1)}$ from the current one $x_l^{(i)}$. Hence, the choice of $\{q_k\}$ is very important.

It was observed in Au and Beck (2001a) that the efficiency of MMA is not sensitive to the type of the proposal PDFs (Gaussian, uniform, etc.); however, it strongly depends on their spread (variance). Both small and large spreads tend to increase the correlation between successive samples. Large spreads may reduce the acceptance rate in Eq. 28, increasing the number of repeated MCMC samples. Small spreads, on the contrary, may lead to a reasonably high acceptance rate, but still produce very correlated samples due to their close proximity. As a rule of thumb, the spread of q_k, k = 1,…,d, can be taken of the same order as the spread of the corresponding marginal PDF π_k (Au and Wang 2014). For example, if π is given by Eq. 18, so that all marginal PDFs are standard Gaussian, $\pi_k(x) = \phi(x)$, then all proposal PDFs can also be Gaussian with unit variance, $q_k(x|x_k) = \phi(x - x_k)$. This choice is found to give a balance between efficiency and robustness.

The spread of proposal PDFs can also be chosen adaptively. In Zuev et al. (2012), where the problem of optimal scaling for the Modified Metropolis algorithm was studied in more detail, the following nearly optimal scaling strategy was proposed: at each conditional level, select the spread such that the corresponding acceptance rate in Eq. 28 is between 30 % and 50 %. In general, finding the optimal spread of proposal distributions is problem specific and a highly nontrivial task not only for MMA but also for almost all MCMC algorithms.

Illustrative Examples

To illustrate Subset Simulation and to demonstrate its efficiency in estimating small probabilities of rare failure events, two examples are considered in this section. As it has been discussed in section "Engineering Reliability Problem," in reliability problems, the dimension d of the input space χ is usually very large. In spite of this, for visualization and educational purposes, a linear reliability problem in two dimensions (d = 2) is first considered in section "Subset Simulation in 2D." A more realistic high-dimensional example ($d = 10^3$) is considered in the subsequent section "Subset Simulation in High Dimensions."

Subset Simulation in 2D

Suppose that d = 2, i.e., the response variable y depends only on two input variables x_1 and x_2. Consider a linear performance function

$$g(x_1, x_2) = x_1 + x_2, \tag{46}$$

where x_1 and x_2 are independent standard Gaussian, $x_i \sim N(0, 1)$, $i = 1, 2$. The failure domain F is then a half-plane defined by

$$F = \{(x_1, x_2) : x_1 + x_2 > y^*\}. \tag{47}$$

In this example, the failure probability p_F can be calculated analytically. Indeed, since $x_1 + x_2 \sim N(0, 2)$ and, therefore, $\frac{x_1+x_2}{\sqrt{2}} \sim \mathcal{N}(0,1)$,

$$p_F = \mathbb{P}(x_1 + x_2 > y^*) = \mathbb{P}\left(\frac{x_1 + x_2}{\sqrt{2}} > \frac{y^*}{\sqrt{2}}\right) = 1 - \Phi\left(\frac{y^*}{\sqrt{2}}\right), \tag{48}$$

where Φ is the standard Gaussian CDF. This expression for the failure probability can be used as a check on the SS estimate. Moreover, expressing y^* in terms of p_F ,

$$y^* = \sqrt{2}\Phi^{-1}(1 - p_F), \tag{49}$$

allows to solve the inverse problem, namely, to formulate a linear reliability problem with a given value of the failure probability. Suppose that $p_F = 10^{-10}$ is the target value. Then the corresponding value of the critical threshold is $y^* \approx 9$.

Subset Simulation was used to estimate the failure probability of the rare event in Eq. 47 with $y^* = 9$. The parameters of the algorithm were chosen as follows: the level probability $p = 0.1$, the proposal PDFs $q_k(x|x_k) = \phi(x - x_k)$, and the sample size $n = 10^3$ per each level. This implementation of SS led to $L = 9$ conditional levels, making the total number of generated samples $N = n + L(n - np) = 9.1 \times 10^3$. The obtained SS estimate is $\widehat{p}_F^{SS} = 1.58 \times 10^{-10}$ which is quite close to the true value $p_F = 10^{-10}$. Note that, in this example, it is hopeless to obtain an accurate estimate by the Direct Monte Carlo method since the DMC estimate in Eq. 12 based on $N = 9.1 \times 10^3$ samples is effectively zero: the rare event F is too rare.

Figure 6 shows the samples generated by the SS method. The dashed lines represent the boundaries of intermediate failure domains F_l, $l = 1,\ldots,L = 9$. The solid line is the boundary of the target failure domain F. This illustrates how Subset Simulation pushes Monte Carlo samples (red) toward the failure region.

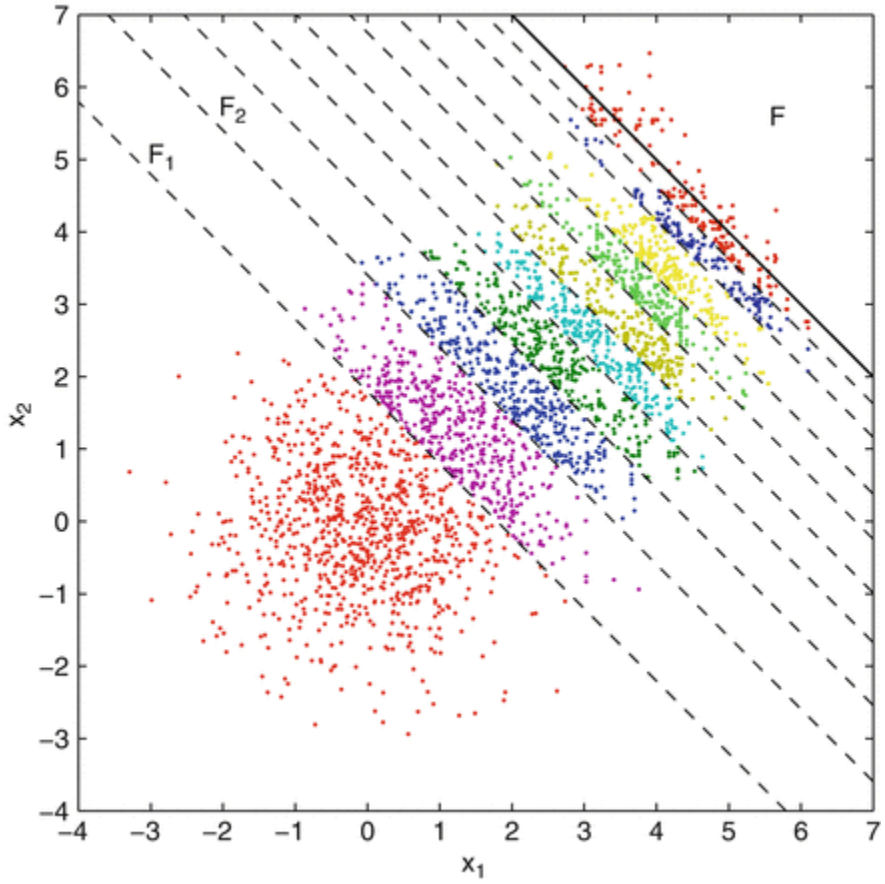

Fig. 6
Samples generated by Subset Simulation: red samples are Monte Carlo samples generated at the 0th unconditional level, purple samples are MCMC sample generated at the 1st conditional level, etc. The dashed lines represent the boundaries of intermediate failure domains F_l, $l = 1,\ldots,L = 9$. The solid line is the boundary of the target failure domain F [Example 6.1]

Subset Simulation in High Dimensions

It is straightforward to generalize the low-dimensional example considered in the previous section to high dimensions. Consider a linear performance function

$$g(x) = \sum_{i=1}^{d} x_i, \tag{50}$$

where $x_1,\ldots,x_d$ are i.i.d. standard Gaussian. The failure domain is then a half-space defined by

$$F = \left\{ x : \sum_{i=1}^{d} x_i > y^* \right\}. \tag{51}$$

In this example, $d = 10^3$ is considered; hence the input space $\chi = \mathbb{R}^d$ is indeed high dimensional. As before, the failure probability can be calculated analytically:

$$p_F = \mathbb{P}\left(\sum_{i=1}^{d} x_i > y^*\right) = \mathbb{P}\left(\frac{\sum_{i=1}^{d} x_i}{\sqrt{d}} > \frac{y^*}{\sqrt{d}}\right) = 1 - \Phi\left(\frac{y^*}{\sqrt{d}}\right). \quad (52)$$

This expression will be used as a check on the SS estimate.

First, consider the following range of values for the critical threshold, $y^* \in [0, 200]$. Figure 7 plots p_F versus y^*. The solid red curve corresponds to the sample mean of the SS estimates $\widehat{p}_F^{\,\mathrm{SS}}$ which is based on 100 independent runs of Subset Simulation. The two dashed red curves correspond to the sample mean ± one sample standard deviation. The SS parameters were set as follows: the level probability p = 0.1, the proposal PDFs $q_k(x|x_k) = \phi(x-x_k)$, and the sample size $n = 3 \times 10^3$ per each level. The solid blue curve (which almost coincides with the solid red curve) corresponds to the true values of p_F computed from Eq. 52. The dark green curves correspond to Direct Monte Carlo: the solid curve is the sample mean (based on 100 independent runs) of the DMC estimates $\widehat{p}_F^{\,\mathrm{DMC}}$ in Eq. 12, and the two dashed curves are the sample mean ± one sample standard deviation. The total number of samples N used in DMC equals to the average (based on 100 runs) total number of samples used in SS. Finally, the dashed light green curves show the theoretical performance of Direct Monte Carlo, namely, they correspond to the true value of p_F (52) ± one theoretical standard deviation obtained from Eq. 14. The bottom panel of Fig. 7 shows the zoomed-in region that corresponds to the values $y^* \in [100, 160]$ of the critical threshold. Note that for relatively large values of the failure probability, p_F <</i> 10^{-3}, both DMC and SS produce accurate estimates of p_F. For smaller values however, pF <</i> 10^{-5}, the DMC estimate starts to degenerate, while SS still accurately estimates p_F. This can be seen especially well in the bottom panel of the figure.

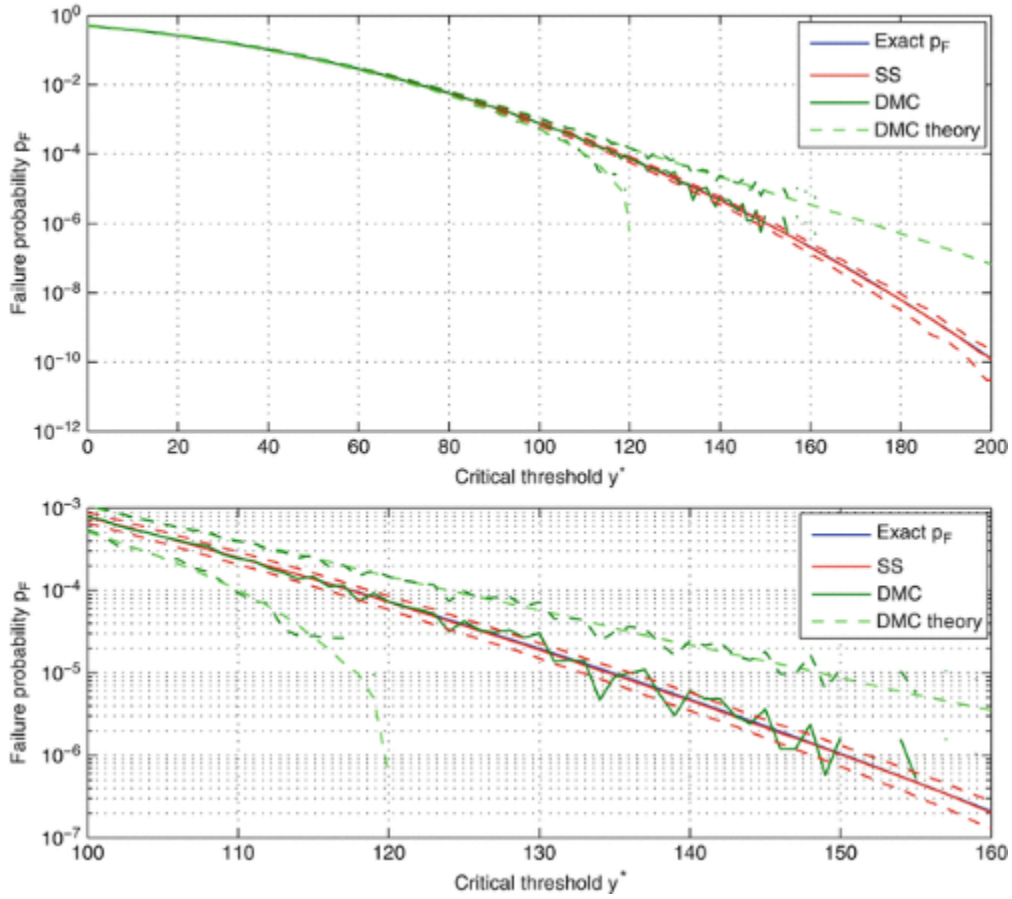

Fig. 7

Failure probability p_F versus the critical threshold y^* [Example 6.2]

The performances of Subset Simulation and Direct Monte Carlo can be also compared in terms of the coefficient of variation of the estimates $\widehat{p}_F^{\mathrm{SS}}$ and $\widehat{p}_F^{\mathrm{DMC}}$. This comparison is shown in Fig. 8. The red and dark green curves represent the sample c.o.v. for SS and DMC, respectively. The light green curve is the theoretical c.o.v. of $\widehat{p}_F^{\mathrm{DMC}}$ given by Eq. 15. When the critical threshold is relatively small y^* <</i> 60, the performances of SS and DMC are comparable. As y^* gets large, the c.o.v. of $\widehat{p}_F^{\mathrm{DMC}}$ starts to grow much faster than that of $\widehat{p}_F^{\mathrm{SS}}$. In other words, SS starts to outperform DMC, and the larger the y^*, i.e., the more rare the failure event, the more significant the outperformance is.

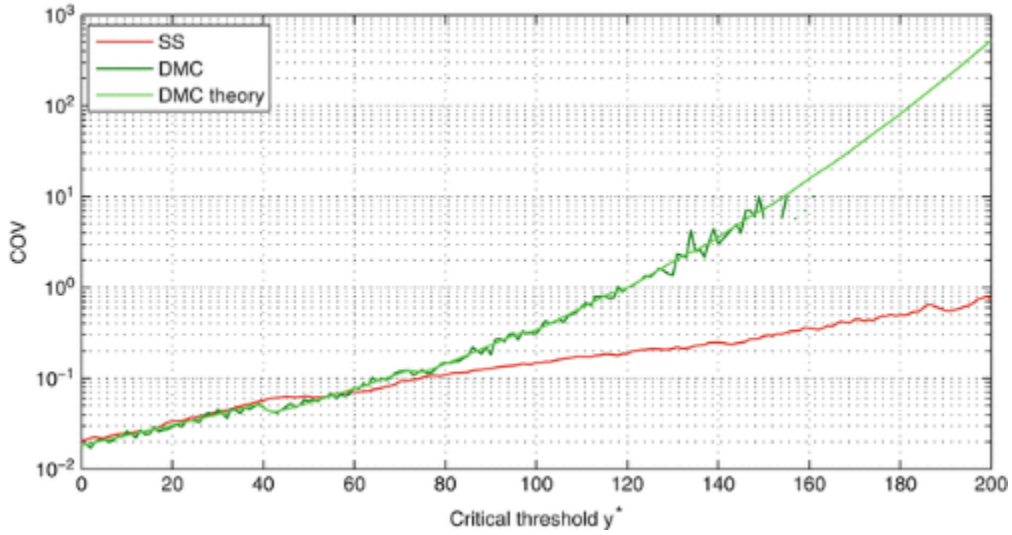

Fig. 8
C.o.v versus the critical threshold [Example 6.2]

The average total number of samples used in Subset Simulation versus the corresponding values of failure probability is shown in the top panel of Fig. 9. The staircase nature of the plot is due to the fact that every time p_F crosses the value p^k by decreasing from p^k+ to p^k-, an additional conditional level is required. In this example, $p = 0.1$ is used, that is why the jumps occur at $p_F = 10^{-k}$, $k = 1, 2,....$ The jumps are more pronounced for larger values of p_F, where the SS estimate is more accurate. For smaller values of p_F, where the SS estimate is less accurate, the jumps are more smoothed out by averaging over independent runs.

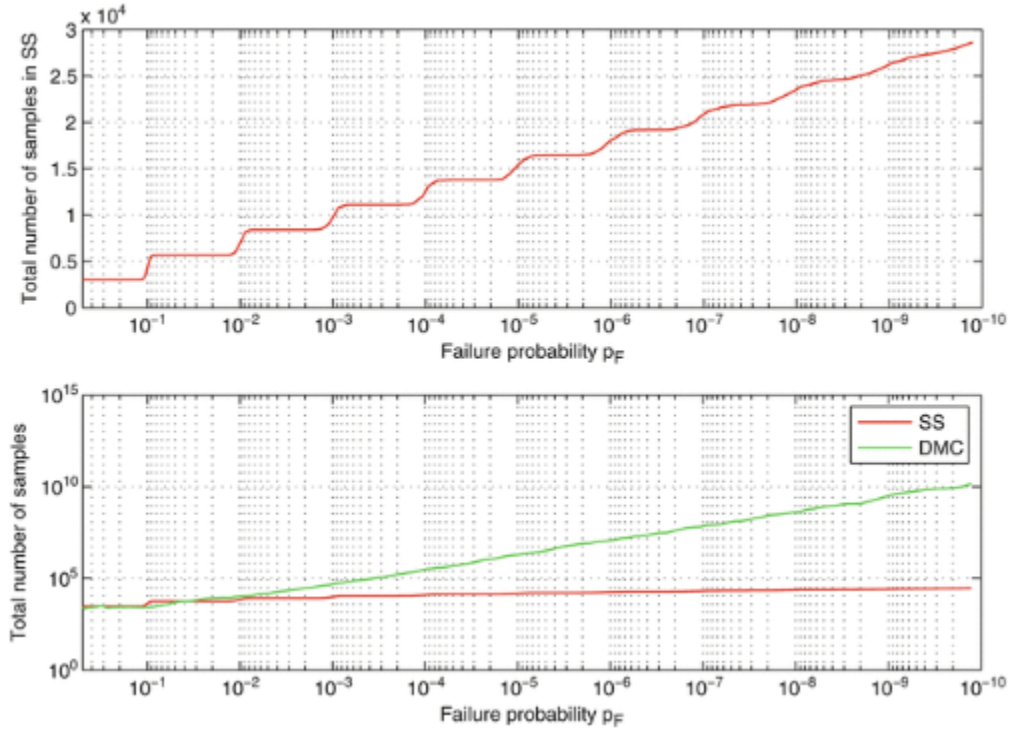

Fig. 9
Total number of samples versus the failure probability [Example 6.2]

In Fig. 8, where the c.o.v's of SS and DMC are compared, the total numbers of samples (computational efforts) used in the two methods are the same. The natural question is then the following: by how much should the total number of

samples N used in DMC be increased to achieve the same c.o.v as in SS (so that the green curve in Fig. 8 coincides with the red curve)? The answer is given in the bottom panel of Fig. 9. For example, if $p_F = 10^{-10}$, then N = 1010, while the computational effort of SS is less than 105 samples.

Simulation results presented in Figs. 7, 8, and 9 clearly indicate that (a) Subset Simulation produces a relatively accurate estimate of the failure probability and (b) Subset Simulation drastically outperforms Direct Monte Carlo when estimating probabilities of rare events.

Let us now focus on a specific value of the critical threshold, $y^* = 200$, which corresponds to a very rare failure event in Eq. 51 with probability $p_F = 1.27 \times 10^{-10}$. Figure 10 demonstrates the performance of Subset Simulation for 100 independent runs. The top panel shows the obtained SS estimate $\widehat{p}_F^{SS}$ for each run. Although $\widehat{p}_F^{SS}$ varies significantly (its c.o.v. is $\delta(\widehat{p}_F^{SS}) = 0.74$), its mean value $\overline{\widehat{p}_F^{SS}} = 1.18 \times 10^{-10}$ (dashed red line) is close to the true value of the failure probability (dashed blue line). The bottom panel shows the total number of samples used in SS in each run. It is needless to say that the DMC estimate based on $N \sim 3 \times 10^4$ samples would almost certainly be zero.

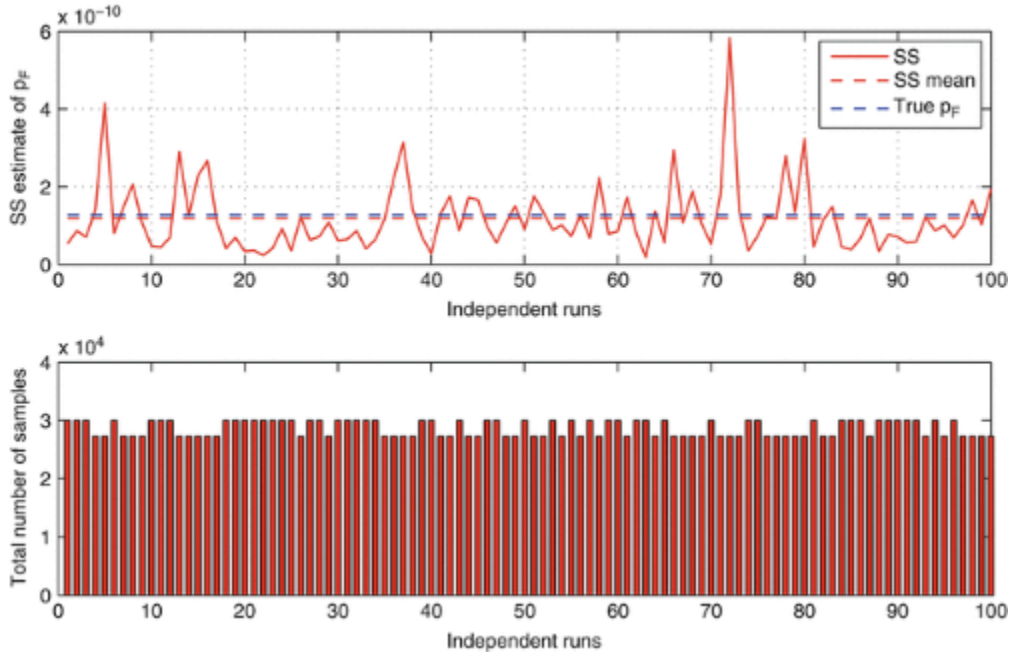

Fig. 10

Performance of Subset Simulation for 100 independent runs. The critical threshold is $y^* = 200$, and the corresponding true value of the failure probability is $p_F = 1.27 \times 10^{-10}$ [Example 6.2]

Figure 11 shows the system responses $y_l^{(1)} \geq \ldots \geq y_l^{(n)}$, $n = 3 \times 10^{-10}$ for all levels, l = 0,…,L = 9, for a fixed simulation run. As expected, for the first few levels (six levels in this case), the number of failure samples $n_F(l)$, i.e., samples $x_l^{(i)}$ with $y_l^{(i)} = g(x_l^{(i)}) > y^*$, is zero. As Subset Simulation starts pushing the samples toward the failure domain, $n_F(l)$ starts increasing with $n_F(6) = 3$, $n_F(7) = 6$, $n_F(8) = 59$, and, finally, $n_F(9) = 582$, after which the algorithm stopped since $n_F(9)/n = 0.194$ which is large than p = 0.1. Finally, Fig. 12 plots the intermediate (relaxed) critical thresholds y_1^*, …, y_L^* at different levels obtained in a fixed simulation run.

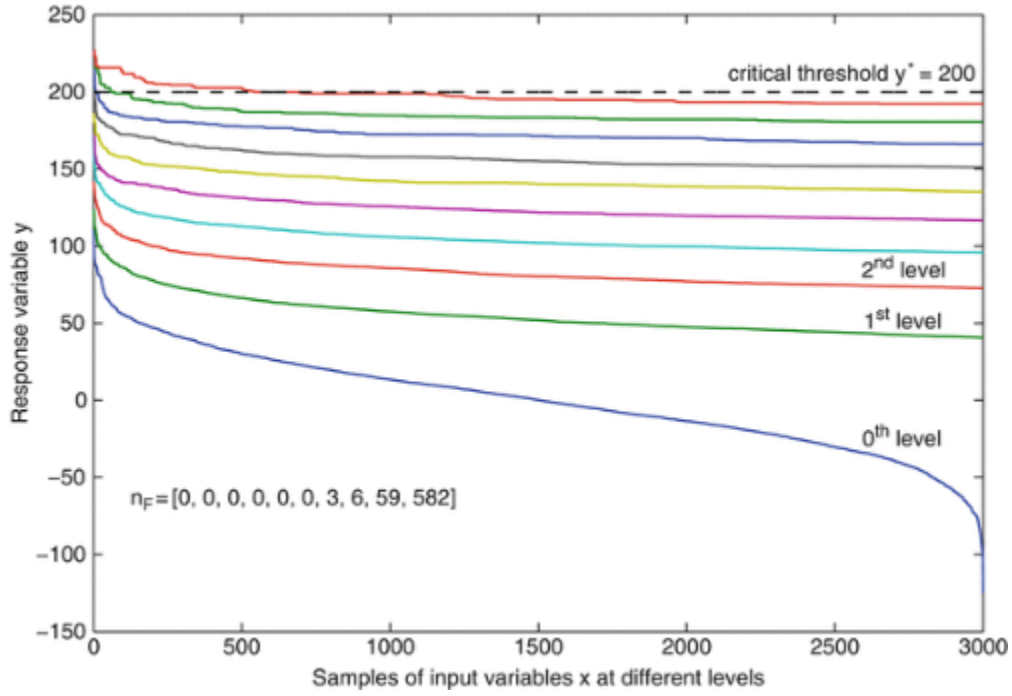

Fig. 11

System responses $y_1^{(1)} \geq \ldots \geq y_1^{(n)}$, $n = 3 \times 10^3$, for all levels, $l = 0,\ldots, L = 9$, for a fixed simulation run [Example 6.2]

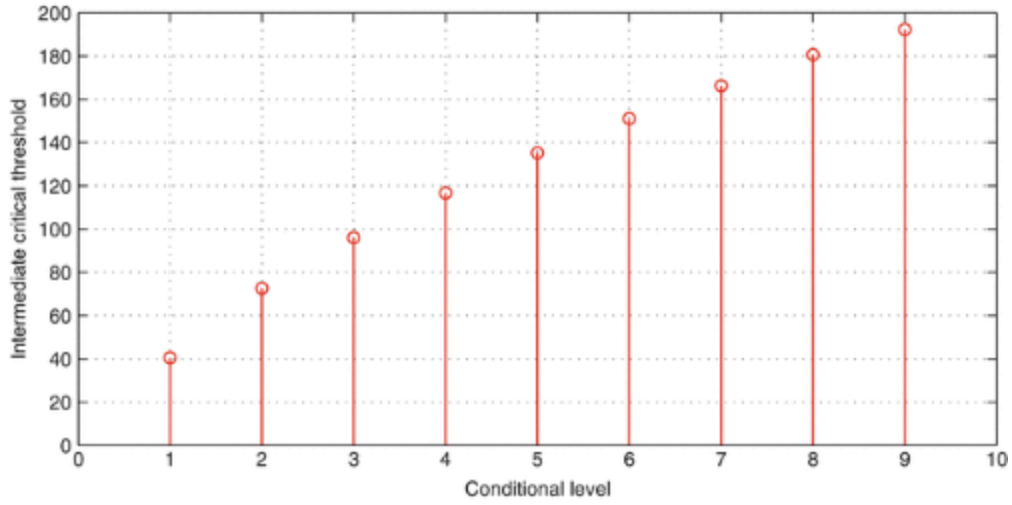

Fig. 12

Intermediate critical thresholds $y_1^{(*)}, \ldots, y_L^*$, $L = 9$, at different conditional levels in a fixed simulation run [Example 6.2]

MATLAB Code

This section contains the MATLAB code for the examples considered in section "Illustrative Examples." For educational purposes, the code was written as readable as possible with numerous comments. As a result of this approach, the efficiency of the code was unavoidably scarified.

```
% Subset Simulation for Liner Reliability Problem
% Performance function: g(x)=x1+…+xd
% Input variables x1,…,xd are i.i.d. N(0,1)
% Written by K.M. Zuev, Institute of Risk & Uncertainty, Uni of Liverpool
```

clear;	
d=1000;	% dimension of the input space
YF=200;	% critical threshold (failure <</i> = > g(x) > YF)
pF=1 − normcdf(YF/sqrt(d));	% true value of the failure probability
n=3000;	% number of samples per level
p=0.1;	% level probability
nc=n*p;	% number of Markov chains
ns=(1 − p)/p;	% number of states in each chain
L=0;	% current (unconditional) level
x=randn(d,n);	% Monte Carlo samples
nF=0;	% number of failure samples
for i=1:n	
y(i)=sum(x(:,i));	% system response y=g(x)
if y(i) > YF	% y(i) > YF <</i> = > x(:,i) is a failure sample
nF=nF+1;	
end	
end	
while nF(L+1)/n <</i> p	% stopping criterion
L=L+1;	% next conditional lelvel is needed
[y(L,:),ind]=sort(y(L,:),'descend');	% renumbered responses
x(:,:,L)=x(:,ind(:),L);	% renumbered samples
Y(L)=(y(L,nc)+y(L,nc+1))/2;	% Lˆth intermediate threshold
z(:,:,1)=x(:,1:nc,L);	% Markov chain "seeds"

```
% Modified Metropolis algorithm for sampling from pi(x | F L)
for j=1:nc
for m=1:ns
% Step 1:
for k=1:d
a=z(k,j,m)+randn; % Step 1(a)
r=min(1,normpdf(a)/normpdf(z(k,j,m))); % Step 1(b)
% Step 1(c):
if rand <</i> r
q(k)=a;
else
q(k)=z(k,j,m);
end
end
% Step 2:
if sum(q) > Y(L) % q belongs to F L
z(:,j,m+1)=q;
else
z(:,j,m+1)=z(:,j,m);
end
end
end
for j=1:nc
for m=1:ns+1
x(:,(j − 1)*(ns+1)+m,L+1)=z(:,j,m); % samples from pi(x | F_L)
end
end
clear z;
```

nF(L+1)=0;	
for i=1:n	
y(L+1,i)=sum(x(:,i,L+1));	% system response y=g(x)
if y(L+1,i) > YF	% then x(:,i,L+1) is a failure sample
nF(L+1)=nF(L+1)+1;	% number of failure samples at level L+1
end	
end	
end	
pF SS=pˆ(L)*nF(L+1)/n;	% SS estimate
N=n+n*(1 − p)*(L);	% total number of samples

Summary

In this entry, a detailed exposition of Subset Simulation, an advanced stochastic simulation method for estimation of small probabilities of rare events, is provided at an introductory level. A simple step-by-step derivation of Subset Simulation is given, and important implementation details are discussed. The method is illustrated with a few intuitive examples.

After the original paper (Au and Beck 2001a) was published, various modifications of SS were proposed: SS with splitting (Ching et al. 2005a), hybrid SS (Ching et al. 2005b), and two-stage SS (Katafygiotis and Cheung 2005), to name but a few. It is important to highlight, however, that none of these modifications offers a drastic improvement over the original algorithm. A Bayesian analog of SS was developed in Zuev et al. (2012). For further reading on Subset Simulation and its applications, a fundamental and very accessible monograph (Au and Wang 2014) is strongly recommended, where the method is presented from the CCDF (complementary cumulative distribution function) perspective and where the error estimation is discussed in detail.

Also, it is important to emphasize that Subset Simulation provides an efficient solution for general reliability problems

without using any specific information about the dynamic system other than an input-output model. This independence of a system's inherent properties makes Subset Simulation potentially useful for applications in different areas of science and engineering.
As a final remark, it is a pleasure to thank Professor Siu-Kui Au whose comments on the first draft of the entry were very helpful; Professor James Beck, who generously shared his knowledge of and experience with Subset Simulation and made important comments on the pre-final draft of the entry; and Professor Francis Bonahon for his general support and for creating a nice atmosphere at the Department of Mathematics of the University of Southern California, where the author started this work.

Cross-References

A Primer on Seismic Risk Assessment
Offshore Structures Under Stochastic Loading, Reliability of
Probabilistic Seismic Hazard Assessment: An Overview
Reliability Analysis of Nonlinear Vibrating Systems, Spectral Approach
Reliability Estimation and Analysis
Response Variability and Reliability of Structures
Seismic Reliability Assessment, Alternative Methods for
Structural Reliability Estimation for Seismic Loading
Structural Seismic Reliability Analysis

References

- Au SK, Beck JL (2001a) Estimation of small failure probabilities in high dimensions by subset simulation. Probabilist Eng Mech 16(4):263-277
- Au SK, Beck JL (2001b) First-excursion probabilities for linear systems by very efficient importance sampling. Probabilist Eng Mech 16(3):193-207
- Au SK, Wang Y (2014) Engineering risk assessment and design with subset simulation. Wiley, Singapore, To appear
- Bucher C (1990) A fast and efficient response surface approach for structural reliability problem. Struct Saf 7:57-66
- Ching J, Au SK, Beck JL (2005a) Reliability estimation of dynamical systems subject to stochastic excitation using subset simulation with splitting. Comput Method Appl Mech Eng 194(12-16):1557-1579
- Ching J, Beck JL, Au SK (2005b) Hybrid subset simulation method for reliability estimation of dynamical systems subject to stochastic excitation. Probabilist Eng Mech 20(3):199-214
- Ditlevsen O, Madsen HO (1996) Structural reliability methods. Wiley, Chichester
- Doob JL (1953) Stochastic processes. Wiley, New York
- Engelund S, Rackwitz R (1993) A benchmark study on importance sampling techniques in structural reliability. Struct Saf 12(4):255-276
- Faravelli L (1989) Response-surface approach for reliability analysis. J Eng Mech 115:2763-2781
- Grooteman F (2008) Adaptive radial-based importance sampling method for structural reliability. Struct Saf 30(6):533-542
- Hurtado JE (2004) Structural reliability. Statistical learning perspectives. Springer, Heidelberg
- Hurtado JE, Alvarez DA (2003) A classification approach for reliability analysis with stochastic finite element modeling. J Struct Eng 129(8):1141-1149
- Katafygiotis LS, Cheung SH (2005) A two-stage subset simulation-based approach for calculating the reliability of inelastic structural systems subjected to Gaussian random excitations. Comput Method Appl Mech Eng 194(12-16):1581-1595
- Katafygiotis LS, Zuev KM (2007) Estimation of small failure probabilities in high dimensions by adaptive linked importance sampling. In: M. Papadrakakis, D.C. Charmpis, N.D. Lagaros, Y. Tsompanakis (Eds.), ECCOMAS thematic conference on computational methods in structural dynamics and earthquake engineering (COMPDYN), Rethymno, Crete, Greece, June 2007

- Katafygiotis LS, Moan T, Cheung SH (2007) Auxiliary domain method for solving multi-objective dynamic reliability problems for nonlinear structures. Struct Eng Mech 25(3):347-363
- Koutsourelakis PS, Pradlwarter HJ, Schuëller GI (2004) Reliability of structures in high dimensions, part I: algorithms and applications. Probabilist Eng Mech 19(4):409-417
- Liu JS (2001) Monte Carlo strategies is scientific computing. Springer, New York
- Madsen HO, Krenk S, Lind NC (2006) Methods of structural safety. Dover, Mineola
- Melchers R (1999) Structural reliability analysis and prediction. Wiley, Chichester
- Metropolis N (1987) The beginning of the Monte Carlo method. Los Alamos Science 15:125-130
- Metropolis N, Ulam S (1949) The Monte Carlo method. J Am Stat Assoc 44:335-341
- Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21(6):1087-1092
- Nataf A (1962) Détermination des distributions de probabilité dont les marges sont donées. C R Acad Sci 225:42-43
- Papadrakakis M, Papadopoulos V, Lagaros ND (1996) Structural reliability analysis of elastic-plastic structures using neural networks and Monte Carlo simulation. Comput Method Appl Mech Eng 136:145-163
- Robert CP, Casella G (2004) Monte Carlo statistical methods. Springer, New York
- Rosenblatt M (1952) Remarks on a multivariate transformation. Ann Math Stat 23:470-472
- Schuëller GI, Bucher CG, Bourgund U, Ouypornprasert W (1989) On efficient computational schemes to calculate structural failure probabilities. Probabilist Eng Mech 4(1):10-18
- Zuev KM, Katafygiotis LS (2011) Horseracing simulation algorithm for evaluation of small failure probabilities. Probabilist Eng Mech 26(2):157-164
- Zuev KM, Beck JL, Au SK, Katafygiotis LS (2012) Bayesian postprocessor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. Comput Struct 92-93:283-296

Subset Simulation Method for Rare Event Estimation: An Introduction

Dr. Konstantin M. Zuev	Institute for Risk and Uncertainty, University of Liverpool, Liverpool, UK
DOI:	10.1007/SpringerReference_369348
URL:	http://www.springerreference.com/index/chapterdbid/369348
Part of:	Encyclopedia of Earthquake Engineering
Editors:	Dr. Michael Beer, Dr. Edoardo Patelli, Prof. Ioannis A. Kougioumtzoglou and Prof. Ivan Siu-Kui Au
PDF created on:	October, 31, 2014 04:17