

Reliability Estimation for Multiple Failure Region Problems using Importance Sampling and Approximate Metamodels

Ramon C. Kuczera and Zissimos P. Mourelatos

Oakland University

Copyright © 2008 SAE International

ABSTRACT

An efficient reliability estimation method is presented for engineering systems with multiple failure regions and potentially multiple most probable points. The method can handle implicit, nonlinear limit-state functions, with correlated or non-correlated random variables, which can be described by any probabilistic distribution. It uses a combination of approximate or "accurate-on-demand," global and local metamodels which serve as indicators to determine the failure and safe regions. Samples close to limit states define transition regions between safe and failure domains. A clustering technique identifies all transition regions which can be in general disjoint, and local metamodels of the actual limit states are generated for each transition region. Importance sampling is used to generate samples only in the identified transition and failure regions, thus allowing the method to focus on the areas near the failure region and not expend computational effort on the samples in the safe domain. A robust maximin "space-filling" sampling technique is used to construct the metamodels. Two numerical examples highlight the accuracy and efficiency of the method.

1. INTRODUCTION

The design of any engineering system requires the assurance of its reliability and quality. Variations and uncertainties in the inputs and properties of an engineering system cause variations in its performance. Uncertainties in the system characteristics prevent such assurances from being given with absolute certainty. One common approach is to quantify the reliability of performance, or risk of failure, in probabilistic terms.

For large-scale systems, the reliability prediction is usually based on efficient computational methods. Both analytical and simulation-based methods are available. The analytical methods are based on the most probable point (MPP) concept. They include the well known first-order reliability method (FORM) which has been widely used [1, 2], second-order reliability methods (SORM) [3-

5], and multi-point approximation methods [6, 7]. Among the simulation-based methods, the Monte Carlo (MC) method is very simple and accurate. However, its computational cost is prohibitively high. For this reason, more efficient simulation-based techniques have been proposed [8-11]. Among them, the adaptive importance sampling (AIS) techniques have been popular [8, 9]. Also, multi-modal importance sampling methods have been proposed where the sampling density emphasizes all important sample points in the failure domain, each in proportion to the true probability density [10, 11]. A combination of analytical and simulation-based methods has also been used [12]. The analytical methods are generally simple and efficient, but for complex problems, their accuracy cannot be guaranteed. In simulation-based methods, the accuracy can be controlled but the efficiency is generally not satisfactory.

Various math-based methods can be used to estimate the probability of failure using defined limit state functions, which separate the failure and the safe regions. In the probabilistic analysis of complex engineering systems, the limit state functions are in general, implicit and nonlinear. Furthermore, each function evaluation is usually computationally expensive. For these reasons, it is important to choose a method that minimizes the number of function evaluations in estimating the probability of failure without sacrificing accuracy.

In this paper, an accurate and efficient simulation-based method is presented for reliability assessment of engineering systems. It provides an improvement over our previous work [13, 14]. The method addresses problems with multiple MPPs with disjoint failure domains. It can also be used for system reliability assessment of problems with multiple limit states. It can easily handle implicit, nonlinear limit-state functions, with correlated or non-correlated random variables, which are described by any probabilistic distribution. It is based on "accurate-on-demand" global and local metamodels (response surfaces) of the limit states which serve as indicators to determine the "failure" and "safe" regions. High accuracy of these metamodels is not guaranteed.

from the limit states because they are simply used as “failure” indicators. However, improved accuracy is necessary close to the limit states where sample points have a “low” absolute value. In this work, points with “low” limit-state absolute value, define a transition region between the safe and failure regions. All samples in the transition region are identified, and more accurate *local* metamodels of each actual limit state are generated. The local metamodels are used to determine if the transition region points are in the safe or failure domains. The probability of failure is calculated using importance sampling.

If a limit state has multiple failure regions which may be even disjoint, or if multiple limit states exist due to a system reliability assessment, the transition region points may be grouped in clusters. A clustering technique is therefore, used to identify these groupings so that we can build local metamodels for each group. This is essential for handling multiple MPP problems which often exist in a variety of engineering applications such as vibratory problems, for example. Recently, some work has been reported for multiple MPP problems [15-17].

Although different metamodeling techniques can be successfully used in this work, we have chosen the Cross-Validated Moving Least Squares (CVMLS) [18] method. To construct a metamodel, a so-called “space-filling” sampling algorithm is needed. However, the commonly used algorithms such as Latin Hypercube (LH) and Optimal Symmetric LH (OSLH) sampling [19] among others, place most of the samples in the interior of the domain. To avoid this problem, we use a maximin “space-filling” sampling technique (see section 3). Two numerical examples highlight the accuracy and efficiency of the proposed method.

The proposed method is the first step towards developing a gradient-free, simulation-based (not analytical) reliability-based design optimization (RBDO) algorithm which can handle “noisy” limit state problems with multiple failure regions

2. OVERVIEW OF THE PROPOSED METHOD

It has been mentioned that in simulation-based reliability methods, we need the sign of the limit state function value, and not its actual function value. This is achieved in this paper by constructing approximate metamodels which can efficiently identify the safe (high value) and failure (low value) regions. A similar idea has been reported in [15] using an indicator response surface and in [16] using discriminative sampling.

In the proposed method, the entire sampling region is divided into a safe region, a failure region and a transition region between the safe and failure regions. The safe and failure regions are easily identified using an approximate global metamodel $\tilde{g}_G(\mathbf{x})$ which is also

used to identify the transition region. The transition region includes all sample points $\mathbf{x}$, such that $\tilde{g}_G(\mathbf{x}) \in (h_l, h_u)$ where h_u and h_l are the upper and lower bound values respectively, estimated using the global metamodel.

All sample points in the transition region are used in a clustering algorithm (see section 4) which determines if there are distinct transition region domains which are associated with either disjoint or different limit states. Importance sampling is then used to generate additional Monte Carlo samples within each cluster. The sampling PDF is the truncated original PDF over the transition region in order to reduce the variance of the probability of failure estimation [20, 21].

An efficient “space-filling” sampling is subsequently used based on a maximin distance algorithm, which efficiently selects a few sample points with “space-filling” properties. The maximin sampling method selects points from the additional MC samples and the selected points are used to construct a local metamodel $\tilde{g}_L(\mathbf{x})$ for each cluster of transition region points. Good accuracy is ensured by a series of refinements using a “space-filling” algorithm which places samples very close to the limit state within a transition region domain and by constructing a metamodel of a small, local region.

Once the local metamodels (one for each cluster) are created and refined they can be used to predict the system probability of failure using importance sampling (see sections 5 and 6).

The probability of failure p_f is calculated (see Eq. 9 of section 5) as,

$$P_f \approx p_1 \sum_{i=1}^n I[G(\mathbf{X}_i) \leq 0] \cdot f_s(\mathbf{x}_i) \quad (1)$$

where $\mathbf{x}_j, j=1, \dots, n$ is a MC sample according to the sampling distribution $f_s(\mathbf{x})$, n is the number of sample points, and $I(\mathbf{x})$ is an indicator function. If $\mathbf{x}$ is in the safe region ($\tilde{g}_G(\mathbf{x}) > h_u$) or in the transition region ($\tilde{g}_L(\mathbf{x}) \in (h_l, h_u)$ and $\tilde{g}_L(\mathbf{x}) > 0$), $I(\mathbf{x})=0$. Also, if $\mathbf{x}$ is in the failure region ($\tilde{g}_G(\mathbf{x}) < h_l$) or in the transition region ($\tilde{g}_L(\mathbf{x}) \in (h_l, h_u)$ and $\tilde{g}_L(\mathbf{x}) < 0$), $I(\mathbf{x})=1$.

3. A MAXIMIN SAMPLING METHOD

An efficient and accurate construction of a metamodel requires a so-called “space-filling” sampling algorithm. A variety of such algorithms have been reported in the literature including Latin Hypercube (LH) sampling and its variations such as randomized orthogonal arrays, symmetric LH sampling, optimal LH sampling and Optimal Symmetric LH (OSLH) sampling. OSLH has been extensively used [19] due to its ability to fill the

projection properties. However, OSLH has the tendency of placing most of the samples in the interior of the domain. This is especially true for moderate to high-dimensional problems with a relatively small number of available samples. In such cases, the metamodel is forced to extrapolate in order to predict the response of a boundary point. Such a prediction is usually inaccurate. In this paper, we avoid this problem by using a sampling algorithm which places samples both on the boundary and the interior of the domain. Although this requires a larger number of samples, it is necessary from the accuracy point of view.

We use an easy to implement maximin method in which the addition of new samples preserves by default, the “space-filling” properties. This is a very important property which allows us to construct “converged” metamodels with a small number of samples. Convergence is practically achieved if the reconstructed metamodel after a few more samples are added, gives similar predictions with the previous version. Similar predictions are judged based only on the sign, and not the estimated actual value, of the limit state.

Consider a design $D_n = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ which is composed of a collection of k -dimensional samples $\mathbf{x}_i \in \mathbb{R}^k, i=1, \dots, n$. A few judiciously selected samples from D_n define the maximin distance design D where, $D \subseteq D_n$. Let $d(\mathbf{u}, \mathbf{v})$ be the Euclidian distance between samples $\mathbf{u}$ and $\mathbf{v}$, where $\mathbf{u}, \mathbf{v} \in D$. Design D simply maximizes the minimum inter-site distance $\min_{\mathbf{u}, \mathbf{v} \in D} d(\mathbf{u}, \mathbf{v})$, i.e.

$$\max_D \min_{\mathbf{u}, \mathbf{v} \in D} d(\mathbf{u}, \mathbf{v}). \quad (2)$$

More information on maximin and minimax distance criteria is provided in [22, 23]. Ref. [24] discusses their use in metamodel construction.

In this paper, the maximin sampling technique of Eq. (2) is implemented in a straightforward way. Assume that a design D_n of n , k -dimensional samples is available, consisting of n MC samples. Each of the k components of the n MC samples is normalized between zero and one in order to account for potentially different units. If x denotes the k^{th} component, its normalized value is $x^* = (x - x^L)/(x^H - x^L)$, where x^L and x^H are the low and high values respectively, among all n samples.

Point $\mathbf{x}_1^* = \{\bar{x}_1^* \dots \bar{x}_i^* \dots \bar{x}_k^*\}$, where $\bar{x}_i^*, i=1, \dots, k$ is the mean x_i^* coordinate of all n normalized points, is first selected as a “seed” point. Then the distance of all n normalized points from $\mathbf{x}_1^*$ is calculated and the point $\mathbf{x}_2^*$ with the largest distance is selected. Subsequently, the distances of the remaining $(n-2)$ samples from both $\mathbf{x}_1^*$

and $\mathbf{x}_2^*$ are calculated and the point $\mathbf{x}_3^*$ with the maximum minimum distance is selected. The process is repeated m times in order to create the maximin distance design D with m sample points. It should be noted that the method first places sample points on the boundary of the domain. As more points are created, they are placed in the interior of the domain providing therefore, a “space-filling” design. A very useful property of the method is that the addition of new sample points preserves the “space-filling” properties of the new design. This is essential in order to keep the number of function evaluations low.

4. DATA CLUSTERING

Data clustering is a technique that is used to determine similarities in data and to segregate the data into groups based on those similarities. In this paper, the data is sorted using a Euclidian distance measure. Points that are in the same “neighborhood” are grouped together. For problems with multiple disjoint failure regions, clustering collects points into groups allowing us therefore, to create separate local metamodels for each failure region.

Given a set of N points to be clustered, an N by N distance matrix is formed with elements $d(\mathbf{x}^i, \mathbf{x}^j) = \sqrt{(\mathbf{x}^i - \mathbf{x}^j)^2}$ representing the Euclidian distance between points $\mathbf{x}^i$ and $\mathbf{x}^j$. Subsequently, a hierarchical cluster tree is created using the following steps [25].

1. Assign each point to a cluster, so that N clusters are formed with one point each. The distance between the clusters is therefore, the same with the distance between the points they contain.
2. Find the closest pair of clusters and merge them into a single cluster, so that we have one cluster less.
3. Compute the distances between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all points are potentially clustered into a single cluster of size N . There is obviously no point in having all N points grouped in a single cluster since we are only obtaining the dataset we started with. However, after we have a complete hierarchical tree, k clusters can be formed by simply cutting the $k - 1$ longest links.

Step 3 can be performed in different ways, which is what distinguishes single-linkage from complete-linkage and average-linkage clustering. In single-linkage clustering, the distance between two clusters is equal to the shortest distance from any point of one cluster to any point of the other cluster. In this work, the single-linkage clustering is used. The described hierarchical clustering is also called agglomerative because it merges clusters iteratively.

5. IMPORTANCE SAMPLING

Many methods exist for estimating system reliability; the most commonly used is Monte Carlo simulation because of its many advantages. It can easily handle any statistical distribution, correlated or uncorrelated random variables as well as explicit and implicit limit states. However, Monte Carlo simulation is computationally intensive. For systems with high reliability, most of sample points are in the safe domain. Each of these safe points requires a function evaluation which may be computationally or physically expensive (i.e. running a finite element code or a physical test). For this reason, it is desirable to generate sample points only in the area of interest, where the system fails or is close to failing. This is achieved using importance sampling which requires sampling from parts of a distribution (e.g. left tail, right tail or middle section). The next subsections give a brief overview of how to sample from parts of a distribution and a brief overview of the importance sampling technique.

SAMPLING FROM THE LEFT TAIL

Sampling from the left tail of a distribution is often required when a small value of the random variable can contribute to the system failure. Fig. 1 illustrates the concept where the shaded region represents the desired sampling region.

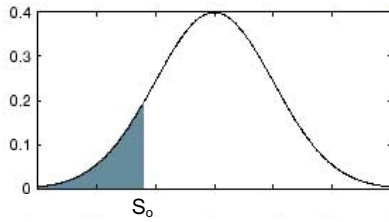


Figure 1. Left Tail Sampling

If sample points s are needed such that $s < s_0$, the sampling PDF $f_s(s)$ is defined as $f_s(s) = \frac{f_o(s)}{p_1}$ where

$f_o(s)$ is the original PDF and $p_1 = \int_{-\infty}^{s_0} f_o(s) ds = F_o(s_0)$.

Therefore, the sampling CDF is

$$F_s(s) = \int_{-\infty}^s f_s(s) ds = \frac{1}{p_1} \int_{-\infty}^s f_o(s) ds \quad \text{or}$$

$$F_s(s) = \frac{F_o(s)}{p_1} \quad (3)$$

SAMPLING FROM THE RIGHT TAIL

Similarly we can sample from the right tail of a distribution as shown in Fig. 2. In this case, we will generate sample points s with values greater or equal to s_0 .

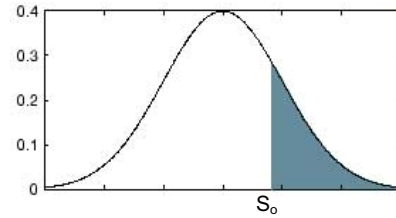


Figure 2. Right Tail Sampling

The sampling PDF is defined as $f_s(s) = \frac{f_o(s)}{1-p_1}$ for $s > s_0$

where $p_1 = F_o(s_0)$. The sampling CDF is equal to

$$F_s(s) = \int_{s_0}^s \frac{f_o(s)}{1-p_1} ds = \frac{F_o(s) - F_o(s_0)}{1-p_1} = \frac{F_o(s) - p_1}{1-p_1} \quad (4)$$

SAMPLING FROM A MIDDLE REGION

In this case we want to generate sample points s such that $s_0 < s < s_1$ (see Fig. 3).

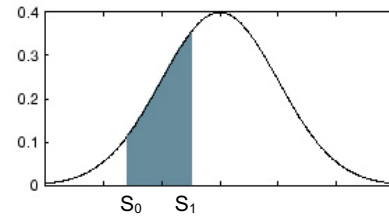


Figure 3. Middle Region Sampling

The sampling PDF is defined as $f_s(s) = \frac{f_o(s)}{p_1 - p_0}$ where

$p_1 = F_o(s_1)$ and $p_0 = F_o(s_0)$. Therefore, the sampling CDF is

$$F_s(s) = \int_{s_0}^s \frac{f_o(s)}{p_1 - p_0} ds = \frac{F_o(s) - F_o(s_0)}{p_1 - p_0} = \frac{F_o(s) - p_0}{p_1 - p_0} \quad (5)$$

IMPORTANCE SAMPLING

The probability of failure is calculated from

$$P_f = \int \dots \int I[G(\mathbf{X}) \leq 0] f_s(\mathbf{x}) d\mathbf{x} \quad (6)$$

where $G(\mathbf{X}) = 0$ is the limit state function and $I(\)$ is an indicator function (see Eq. 1).

If $f_s(\mathbf{x})$ is a sampling distribution, the above expression can be written as