

# RARE EVENT ESTIMATION USING STOCHASTIC SPECTRAL EMBEDDING

P.-R. Wagner, S. Marelli, I. Papaioannou, D. Straub and B. Sudret



## Data Sheet

---

**Journal:** Structural Safety

**Report Ref.:** RSUQ-2021-003B

**Arxiv Ref.:** <https://arxiv.org/abs/2106.05824> [stat.CO, stat.ML, stat.ME]

**DOI:** <https://doi.org/10.1016/j.strusafe.2021.102179>

**Date submitted:** June 10, 2021

**Date accepted:** December 21, 2021

---

# Rare event estimation using stochastic spectral embedding

P.-R. Wagner<sup>1</sup>, S. Marelli<sup>1</sup>, I. Papaioannou<sup>2</sup>, D. Straub<sup>2</sup>, and B. Sudret<sup>1</sup>

<sup>1</sup>*Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Stefano-Francini-Platz 5, 8093 Zurich, Switzerland*

<sup>2</sup>*Engineering Risk Analysis Group, Technical University Munich, Arcisstraße 21, 80290 Munich, Germany*

22.12.2021

## Abstract

Estimating the probability of rare failure events is an essential step in the reliability assessment of engineering systems. Computing this failure probability for complex non-linear systems is challenging, and has recently spurred the development of active-learning reliability methods. These methods approximate the limit-state function (LSF) using surrogate models trained with a sequentially enriched set of model evaluations. A recently proposed method called stochastic spectral embedding (SSE) aims to improve the local approximation accuracy of global, spectral surrogate modelling techniques by sequentially embedding local residual expansions in subdomains of the input space. In this work we apply SSE to the LSF, giving rise to a stochastic spectral embedding-based reliability (SSER) method. The resulting partition of the input space decomposes the failure probability into a set of easy-to-compute conditional failure probabilities. We propose a set of modifications that tailor the algorithm to efficiently solve rare event estimation problems. These modifications include specialized refinement domain selection, partitioning and enrichment strategies. We showcase the algorithm performance on four benchmark problems of various dimensionality and complexity in the LSF.

**Keywords:** reliability analysis – uncertainty quantification – surrogate modelling – stochastic spectral embedding – active learning – rare event estimation – sparse polynomial chaos expansions

## 1 Introduction

Ensuring the reliability of structures and systems is a core task in many engineering disciplines. This includes the reliability of machines (Bertsche, 2008), medical devices (Fries,

2017), electronic systems (LaCombe, 1999) and civil engineering systems (Haldar, 2006). With the increasing availability of computer simulations, and continuous developments in numerical methods, model-based reliability analysis has become the state-of-the-art in reliability assessment. In this context, the task of reliability analysis lies in estimating the probability that an engineering system fails or performs undesirably. The computation of this so-called *failure probability* is the objective of quantitative model-based reliability methods.

In reliability problems, the system under consideration is typically parameterized by a vector of random input parameters  $\mathbf{X} : \Omega \rightarrow \mathcal{D}_{\mathbf{X}} \subseteq \mathbb{R}^M$  following a probability distribution  $\mathbf{X} \sim f_{\mathbf{X}}$ . To identify the safe/failure state of a system as a function of the input parameters, a so-called *limit-state function*  $g : \mathcal{D}_{\mathbf{X}} \rightarrow \mathbb{R}$  is introduced. It encodes the system performance w.r.t. the limit states that apply to the system under consideration (e.g., ultimate stresses or maximum displacements at critical locations). Importantly, the function  $g$  depends on one or several, often computationally demanding, engineering models. By convention, if  $g(\mathbf{x}) \leq 0$ , then  $\mathbf{x}$  corresponds to a failed configuration, otherwise it corresponds to a safe configuration. The interface between the resulting *safe* and *failure* domains is known as the *limit-state surface*.

One can then define the probability of failure  $P_f$  as:

$$P_f \stackrel{\text{def}}{=} \mathbb{P}[g(\mathbf{X}) \leq 0] = \int_{\mathcal{D}_{\mathbf{X}}} \mathbf{1}_{\mathcal{D}_f}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \quad (1)$$

where the indicator function  $\mathbf{1}_{\mathcal{D}_f}(\mathbf{x})$  takes the value 1 in the failure domain  $\mathcal{D}_f$  and 0 everywhere else:

$$\mathbf{1}_{\mathcal{D}_f}(\mathbf{x}) = \begin{cases} 1, & \text{if } g(\mathbf{x}) \leq 0, \\ 0, & \text{if } g(\mathbf{x}) > 0. \end{cases} \quad (2)$$

Analytical computation of the failure probability is rarely possible in practice and direct numerical integration (e.g., via quadrature) is often hindered by the inherently small scale of the failure probability and the potentially high input dimensionality  $M$ . For these reasons, initial efforts in the reliability literature focused on developing methods that approximate the limit-state surface (Basler, 1960; Hasofer and Lind, 1974; Rackwitz and Fiessler, 1978; Zhang and Der Kiureghian, 1995; Hohenbichler et al., 1987; Breitung, 1994; Tvedt, 1990; Cai and Elishakoff, 1994). These approximation methods remain competitive for a certain class of reliability problems even today, but there exist well known examples where the shape of the limit-state surface leads to gross errors in the estimated failure probability (e.g., high-dimensional, non-linear problems Valdebenito et al. (2010)). In those cases the direct estimation of the failure probability by means of probabilistic simulation methods is the method of choice. The most widely used methods for this are *Monte Carlo simulation* (MCS, Fishman (2011); Rubinstein and Kroese (2016)), approximation methods in conjunction with *importance sampling* (IS, Hohenbichler and Rackwitz (1988); Melchers (1999)), its *adaptive variants* (sequential IS, Papaioannou et al. (2016) and cross entropy-based IS,

Kurtz and Song (2013); Papaioannou et al. (2019)) and *subset simulation* (SuS, Au and Beck (2001); Zuev et al. (2012)). Other noteworthy methods that can be used to directly estimate the failure probability are *radial-based importance sampling* (RBIS, Harbitz (1986)), *directional simulation* (DS, Bjerager (1988)) and *line sampling* (LS, Koutsourelakis et al. (2004); Papaioannou and Straub (2021)).

Computationally expensive models remain challenging to analyse, even with the most efficient simulation methods. In this setting, cheap to evaluate surrogate models can lead to significant computational savings, especially with so-called *active learning reliability* methods. These methods approximate the limit-state function  $g$  using a sequence of increasingly accurate surrogate models. The first method in this family was *efficient global reliability analysis* (EGRA, Bichon et al. (2008)). Arguably the best known method today is *adaptive Kriging* MCS (AK-MCS, Echard et al. (2011)) and its variants (Echard et al., 2013; Huang et al., 2016). A recent review of the field is given in Moustapha et al. (2021).

In this paper we propose a novel active learning reliability method that utilizes the recently proposed *stochastic spectral embedding* method (SSE, Marelli et al. (2021)) and more specifically the active learning sequential partitioning approach developed for Bayesian inverse problems in Wagner et al. (2021). The proposed approach, termed *stochastic spectral embedding-based reliability* (SSER), benefits from a handful of modifications to the original SSE, namely new *refinement domain selection*, *partitioning* and *sample enrichment* schemes, which we present in Section 2. SSER constructs a set of spectral representations of  $g$  with higher accuracy in proximity of the limit-state surface. The proposed adaptive partitioning strategy results in a set of subdomains in the vicinity of the limit-state surface. Provided that  $g$  is sufficiently well-behaved in these subdomains, our approach is capable of efficiently addressing problems of varying complexity (i.e., non-linear computational models, high dimensional problems, small failure probabilities). We showcase this property by comparing SSER to other state-of-the-art methods on four applications in Section 3.

## 2 Stochastic spectral embedding-based reliability

### 2.1 Stochastic spectral embedding

Stochastic spectral embedding (SSE, Marelli et al. (2021)) is a function approximation technique that works by sequentially expanding residuals  $\widehat{\mathcal{R}}_S^k(\mathbf{X})$  in subdomains of the parameter space  $\mathcal{D}_{\mathbf{X}}^k \subseteq \mathcal{D}_{\mathbf{X}}$ . At each level in the sequence, SSE constructs local spectral expansions of the residual of the current approximation at a set of subdomains, as illustrated in Figure 1. As for any other metamodeling techniques, this approach can be used in reliability analyses to directly approximate the limit-state function:

$$g(\mathbf{X}) \approx g_{\text{SSE}}(\mathbf{X}) = \sum_{k \in \mathcal{K}} \mathbf{1}_{\mathcal{D}_{\mathbf{X}}^k}(\mathbf{X}) \widehat{\mathcal{R}}_S^k(\mathbf{X}). \quad (3)$$

In this expression,  $\mathcal{K} \subseteq \mathbb{N}^2$  is a set of index pairs with elements  $k = (\ell, p)$  used to identify the residual expansions in the SSE representation. The first element of the index pair  $\ell \in \{0, \dots, L\}$  denotes the level at which the residual expansion resides, where  $L$  is the total number of levels. The second element  $p \in \{1, \dots, P_\ell\}$  denotes the index of the residual expansion at a given level, with  $P_\ell$  denoting the total number of expansions at the  $\ell$ -th level.  $\mathbf{1}_{\mathcal{D}_{\mathbf{X}}^k}$  is the indicator function that is equal to 1 in the  $k$ -th subdomain and 0 everywhere else. For further derivations, it is beneficial to distinguish between standard and *terminal domains* among the  $\text{card}(\mathcal{K})$  domains used in the SSE representation. A terminal domain is any domain that has not been split. It can be located at any level of the SSE representation and the set of all terminal domains with indices gathered in  $\mathcal{T}$  constitutes a complete partition of the input domain, i.e.,  $\mathcal{D}_{\mathbf{X}} = \bigcup_{k \in \mathcal{T}} \mathcal{D}_{\mathbf{X}}^k$  (top domains highlighted in orange in Figure (a)). With this, Eq. (3) is expanded to

$$g_{\text{SSE}}(\mathbf{X}) = \sum_{k \in \mathcal{K} \setminus \mathcal{T}} \mathbf{1}_{\mathcal{D}_{\mathbf{X}}^k}(\mathbf{X}) \widehat{\mathcal{R}}_S^k(\mathbf{X}) + \sum_{k \in \mathcal{T}} \mathbf{1}_{\mathcal{D}_{\mathbf{X}}^k}(\mathbf{X}) \widehat{\mathcal{R}}_S^k(\mathbf{X}), \quad (4)$$

In principle, any spectral expansion technique could be used to expand the residuals (Marelli et al., 2021). In the present contribution, we consider only *polynomial chaos expansions* (PCEs) such that

$$\widehat{\mathcal{R}}_S^k(\mathbf{X}) \stackrel{\text{def}}{=} \sum_{\alpha \in \mathcal{A}^k} a_\alpha^k \Psi_\alpha^k(\mathbf{X}), \quad (5)$$

where  $\mathcal{A}^k$  is a multi-index set describing the dimension-wise polynomial degrees,  $\{\Psi_\alpha^k\}_{\alpha \in \mathcal{A}^k}$  are mutually orthogonal polynomial basis functions and  $a_\alpha^k$  denote the corresponding coefficients (Xiu and Karniadakis, 2002; Blatman and Sudret, 2011; Lüthen et al., 2021).

For computational and bookkeeping ease, we manage the SSE construction in what we call the *quantile space*, i.e., the  $M$ -dimensional unit hypercube denoted by  $\mathcal{D}_{\mathcal{U}}$ . As discussed in Marelli et al. (2021), for every original or *real input space*  $\mathcal{D}_{\mathbf{X}}$  there exists an isoprobabilistic mapping to  $\mathcal{D}_{\mathcal{U}}$  (Figure (a)), such that the partitioning of the input space can be conducted in the quantile space without any loss of generality.

The SSE representation shown in Figure 1 ultimately partitions the input space into  $\text{card}(\mathcal{T})$  disjoint subdomains such that  $\bigcup_{k \in \mathcal{T}} \mathcal{D}_{\mathbf{X}}^k = \mathcal{D}_{\mathbf{X}}$ . Using this representation, the failure probability from Eq. (1) can be rewritten as a sum of *conditional failure probabilities*

$$P_f = \sum_{k \in \mathcal{T}} \mathcal{V}^k P_f^k, \quad \text{with} \quad \mathcal{V}^k = \int_{\mathcal{D}_{\mathbf{X}}^k} f_{\mathbf{X}}(\mathbf{x}) \, d\mathbf{x}, \quad (6)$$

where  $\mathcal{V}^k$  is the *domain-wise probability mass* that is readily available from the construction in the quantile space, and  $P_f^k$  is the  $k$ -th conditional failure probability given by

$$P_f^k \stackrel{\text{def}}{=} \mathbb{P} [g(\mathbf{X}) \leq 0 | \mathbf{X} \in \mathcal{D}_{\mathbf{X}}^k] \approx \mathbb{P} [g_{\text{SSE}}(\mathbf{X}) \leq 0 | \mathbf{X} \in \mathcal{D}_{\mathbf{X}}^k]. \quad (7)$$

This conditional failure probability is estimated here with suitable probabilistic simulation methods utilizing the SSE representation of the limit-state function (more details are provided in Section 2.3). Assuming the simulation method is accurate enough, the accuracy of the failure probability estimate depends only on that of the SSE representation itself.

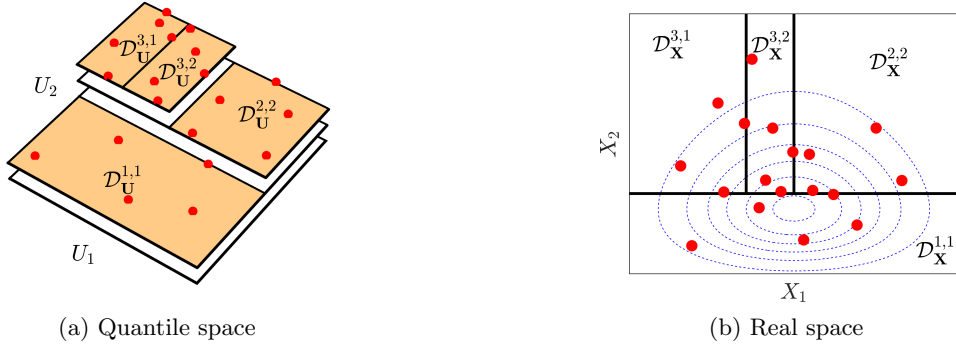


Figure 1: *Stochastic spectral embedding*: Illustration of SSE representation in Eq. (3) in the quantile and real space. Red points denote the experimental design and orange areas denote terminal or unsplit domains, that partition the input space into a set of *disjoint* subdomains such that  $\mathcal{D}_{\mathbf{X}} = \bigcup_{k \in \mathcal{T}} \mathcal{D}_{\mathbf{X}}^k$ .

To devise an adaptive algorithm, it is necessary to devise a suitable error measure on each conditional failure probability estimate. As shown in Marelli et al. (2021), the SSE representation provides *domain-wise* prediction error estimators, e.g., by means of cross-validation (e.g., leave-one-out error). Such estimators can in principle be used as a proxy to assess the accuracy of  $P_f^k$ . However, these estimators are not robust in the presence of very localized behaviour that is typical for limit-state functions. An improved estimator that takes into account the *point-wise* prediction accuracy can be derived by means of the local error estimation of bootstrap PCE (bPCE), originally introduced in a reliability context in Marelli and Sudret (2018).

### 2.1.1 Bootstrap SSE

The idea of bPCE is to use bootstrap resampling (Efron, 1979) on the original experimental design to create a set of  $B$  experimental designs and construct  $B$  sparse PCEs with them. The individual bootstrap PCE realizations can then be used to estimate point-wise statistics of the PCE prediction.

This idea can be applied to the residual expansions within the SSE approach. Due to the sequential embedding of the residual expansions, the prediction error of the full SSE approximation can be assessed solely based on the prediction errors in the terminal domains (Marelli et al., 2021). In every terminal domain, we therefore construct  $B$  residual expansions to obtain  $B$  bootstrap predictions

$$g_{\text{SSE}}^{(b)}(\mathbf{X}) = \sum_{k \in \mathcal{K} \setminus \mathcal{T}} \mathbf{1}_{\mathcal{D}_{\mathbf{X}}^k}(\mathbf{X}) \widehat{\mathcal{R}}_S^k(\mathbf{X}) + \sum_{k \in \mathcal{T}} \mathbf{1}_{\mathcal{D}_{\mathbf{X}}^k}(\mathbf{X}) \widehat{\mathcal{R}}_S^{k,(b)}(\mathbf{X}), \quad \text{for } b \in \{1, \dots, B\}. \quad (8)$$

By analogy with Marelli and Sudret (2018), we can define an estimate of the point-wise

misclassification probability as:

$$\hat{P}_m(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{B} \sum_{b=1}^B \left| \mathbf{1}_{\mathcal{D}_{f,\text{SSE}}}(\mathbf{x}) - \mathbf{1}_{\mathcal{D}_{f,\text{SSE}}^{(b)}}(\mathbf{x}) \right|, \quad (9)$$

where  $\mathbf{1}_{\mathcal{D}_{f,\text{SSE}}}$  and  $\mathbf{1}_{\mathcal{D}_{f,\text{SSE}}^{(b)}}$  are evaluated with the mean and bootstrap predictors of SSE respectively (see also Eq. (2)).

The bootstrap replications can also be used to directly obtain statistics of the conditional failure probability estimates, such as variance or confidence bounds. As an example, the conditional failure probability variance is given by

$$\text{Var} \left[ \hat{P}_f^k \right] = \text{Var} \left[ \left\{ \hat{P}_f^{k,(1)}, \dots, \hat{P}_f^{k,(B)} \right\} \right], \quad (10)$$

with the failure probability of the  $b$ -th replication given by

$$\hat{P}_f^{k,(b)} \stackrel{\text{def}}{=} \mathbb{P} \left[ g_{\text{SSE}}^{(b)}(\mathbf{X}) \leq 0 \mid \mathbf{X} \in \mathcal{D}_{\mathbf{X}}^k \right]. \quad (11)$$

By means of Eq. (6) this conditional failure probability variance can be used to write an expression for the total failure probability variance as

$$\text{Var} \left[ \hat{P}_f \right] = \sum_{k \in \mathcal{T}} (\mathcal{V}^k)^2 \text{Var} \left[ \hat{P}_f^k \right]. \quad (12)$$

This expression assumes independence between the conditional failure probability estimates. Due to the ancestry relation between subdomains, we cannot in general rule out dependence between those quantities, however, we assume that it is negligible because all expansions in terminal domains are constructed with independently enriched samples.

Similarly, the bootstrap replications can be used to define *confidence bounds* on the total failure probability. Let  $\hat{P}_f^{(b)} \stackrel{\text{def}}{=} \sum_{k \in \mathcal{T}} \mathcal{V}^k \hat{P}_f^{k,(b)}$  be the total failure probability estimated with the  $b$ -th replication from all terminal domains. The equal tail confidence bounds  $\left\{ \overline{\hat{P}_f^k}, \underline{\hat{P}_f^k} \right\}$  are then defined by

$$\begin{aligned} \mathbb{P} \left[ \hat{P}_f^k \leq \left\{ \hat{P}_f^{(1)}, \dots, \hat{P}_f^{(B)} \right\} \right] &\approx \alpha, \\ \mathbb{P} \left[ \left\{ \hat{P}_f^{(1)}, \dots, \hat{P}_f^{(B)} \right\} \leq \overline{\hat{P}_f^k} \right] &\approx 1 - \alpha \end{aligned} \quad (13)$$

with  $\alpha \in [0, 0.5]$  and  $\gamma \stackrel{\text{def}}{=} 1 - 2\alpha$  is called the *symmetrical confidence level*. It is common practice to take  $\gamma = 95\%$  and  $\alpha = 2.5\%$ .

### 2.1.2 Dependent input parameters

Generally, the input random vector  $\mathbf{X}$  may have mutually dependent marginals that we treat with copula theory in this work (Nelsen, 2006; Joe, 2015). A typical approach to address dependence in reliability problems is to use isoprobabilistic mappings to an independent space, e.g., the standard normal space. However, sparse spectral techniques acquire their strength from the *sparsity of effects principle* (Montgomery, 2019), which states that in many engineering models the majority of the output is attributable to low-interaction order



terms. A mapping to the standard normal space might therefore reduce the efficiency of those techniques.

Nonetheless, dependence is challenging for the construction of the spectral basis functions  $\Psi_{\alpha}^k$  in Eq. (5). However, as extensively studied in Torre et al. (2019a), polynomial chaos expansions seem to be most accurate for predictive purposes when dependence is simply ignored and the basis is derived by tensor product of univariate bases orthogonal to the marginals instead. Applying this to SSE suggests it is best to construct the domain-wise polynomial basis by assuming the domain-wise marginals to be independent.

As mentioned before, the partitioning into orthogonal subdomains is defined in the independent quantile space (Marelli et al., 2021). Therefore, the transformation of those domains to the dependent physical space results in non-orthogonal domains (see Figure 2). Constructing the univariate polynomial bases requires knowledge of the marginal bounds in the physical space, which cannot be computed in a straightforward fashion.

To approximate these bounds, we therefore (1) randomly sample the  $(M - 1)$ -dimensional boundary of the hypercube in the quantile space, (2) transform those points to the real space using a suitable isoprobabilistic transform (Rosenblatt, 1952; Torre et al., 2019b) and proceed to (3) compute the basis with an orthogonal *enveloping hypercube* in the physical space around those transformed points (see Figure (b)). It is interesting to note that for prediction purposes, the resulting multivariate basis will only be evaluated inside the *subdomain boundary* and not the *subdomain boundary envelope*, as the region outside the subdomain boundary belongs to neighbouring subdomains.

## 2.2 An updated sequential partitioning algorithm

Failure domains typically occupy only a small fraction of the input space. The original *equal probability* partitioning strategy presented in Marelli et al. (2021); Wagner et al. (2021) is therefore expected to converge only slowly to those domains. We therefore propose modifications that make the algorithm more efficient for reliability problems. The structure of the proposed algorithm follows the sequential partitioning algorithm used in the adaptive SSLE approach presented in Wagner et al. (2021). It is repeated here in an abbreviated form.

### 1. Initialization

- (a) Sample from input distribution
- (b) Calculate initial expansion

### 2. Select refinement domain:

- (a) Partition domain
- (b) **For each subdomain:**
  - i. Enrich sample
  - ii. Create residual expansion
  - iii. Update domain-wise error

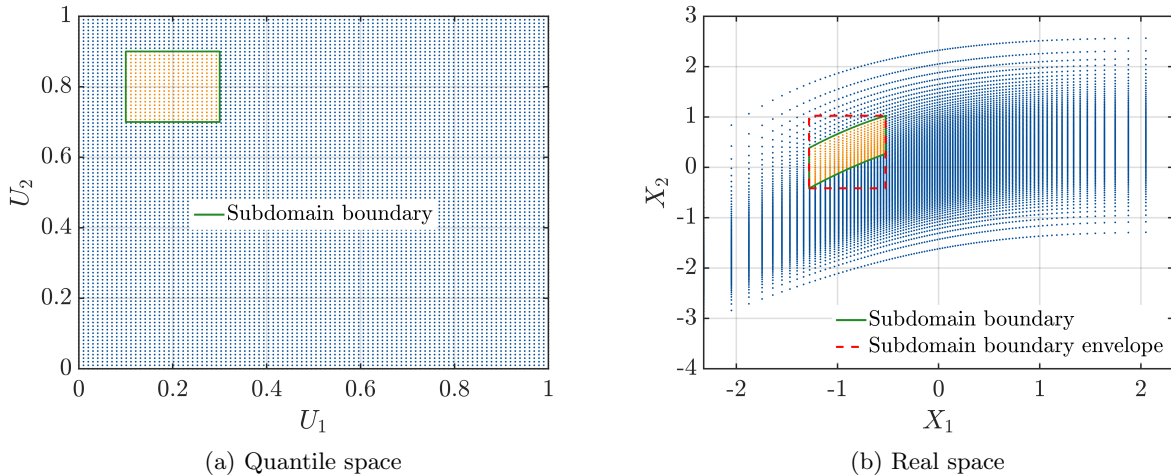


Figure 2: *Stochastic spectral embedding*: Illustration of how dependence is handled in the algorithm. For two mutually dependent input parameters  $X_1$  and  $X_2$ , the plots show a set of sample points in the real and quantile space with a subset of points inside an exemplary subdomain. The parameter dependence leads to a loss of orthogonality of the subdomains in the real space, which necessitates the introduction of an *enveloping hypercube* in the real space to construct the spectral basis functions  $\Psi_{\alpha}^k$ .

(c) Go to Step 2

The main modifications to the algorithm that are specific to reliability problems pertain to the *refinement domain selection* (Step 2), *partitioning* (Step 2a) and *sample enrichment* (Step 2(b)i) strategies, which utilize the point-wise prediction error estimators available from bPCE. These modifications are detailed in the following sections. The full SSE-based reliability (SSER) algorithm is sketched in Section 2.2.4.

### 2.2.1 Refinement domain selection

At every step, the adaptive sequential partitioning algorithm chooses a *refinement domain* from the set of terminal domains  $\mathcal{T}$  to be split and enriched with sample points. To choose a domain, the terminal domains are ranked based on their importance to the overall approximation. In the present setting, the goal of refinement is the reduction of uncertainty associated with the failure probability. Viewing Eq. (12) as a variance decomposition of the estimator variance, it is natural to refine the terminal domain that has the largest contribution to this variance estimator. More formally, out of the terminal domains we refine the domain with the largest

$$\mathcal{E}^k \stackrel{\text{def}}{=} (\mathcal{V}^k)^2 \text{Var} \left[ \hat{P}_f^k \right], \quad (14)$$

This equation depends on the domain-wise probability mass  $\mathcal{V}^k$  and the variance of the conditional failure probability estimator  $\text{Var} \left[ \hat{P}_f^k \right]$ . In practice,  $\text{Var} \left[ \hat{P}_f^k \right]$  is estimated with

Eq. (10), which in turn depends on the  $B$  estimates of  $\hat{P}_f^{k,(b)}$ . These estimates are computed with accurate simulation methods using the bootstrap SSE approximations (see Section 2.3).

If the SSE approximation in the current subdomain is not sufficiently accurate, it can happen that all bootstrap replications miss existing failure regions, ultimately resulting in a gross underestimation of  $\mathcal{E}^k$ . To avoid overlooking such domains as possible refinement domains, the algorithm additionally prioritizes them based on the probability mass  $\mathcal{V}^k$ . The refinement domain is ultimately chosen as

$$k_{\text{refine},i} = \arg \max_{k \in \mathcal{T}} \begin{cases} \mathcal{V}^k, & \text{if Eq. (16) is met,} \\ \mathcal{E}^k, & \text{otherwise,} \end{cases} \quad (15)$$

which depends on the following *intermediate re-prioritization criterion*

$$\frac{\text{Var} \left[ \{ \hat{P}_f^{(i-2)}, \hat{P}_f^{(i-1)}, \hat{P}_f^{(i)} \} \right]}{\left( \hat{P}_f^{(i)} \right)^2} < \varepsilon_{\hat{P}_f}, \quad (16)$$

where  $\hat{P}_f^{(i)}$  is the total failure probability estimator at the  $i$ -th iteration of the algorithm and the threshold is heuristically chosen to  $\varepsilon_{\hat{P}_f} = 0.1\%$ . This criterion is triggered when the failure probability estimator does not change significantly in three successive iterations.

### 2.2.2 Partitioning strategy

Once a refinement domain  $\mathcal{D}_{\mathbf{X}}^{k_{\text{refine}}}$  is selected, the partitioning strategy determines how it is split. Refinement domains are split into two subdomains with unequal probability mass. For notational simplicity, we omit the superscript  $k_{\text{refine}}$  denoting the refinement domain in this section and emphasize instead that the following equations are valid for all  $k \in \mathcal{T}$ . For illustrative purposes, the partitioning strategy is exemplified for a simple problem in Figure 3.

To maximize enrichment efficiency, we base our splitting strategy on separating regions that predict failure/safety correctly from those that do so incorrectly. A measure of prediction accuracy in this respect is given by the misclassification probability  $P_m$  (Echard et al., 2011) defined for bootstrap SSE in Eq. (9). To this end, we define two auxiliary conditional random vectors in the quantile space  $\mathcal{D}_{\mathbf{U}}$  as

$$\mathbf{Z}^0 \stackrel{\text{def}}{=} \mathbf{U} | \hat{P}_m = 0, \quad \text{and} \quad \mathbf{Z}^+ \stackrel{\text{def}}{=} \mathbf{U} | \hat{P}_m > 0. \quad (17)$$

These vectors identify regions of zero ( $\mathbf{Z}^0$ ) and non-zero ( $\mathbf{Z}^+$ ) misclassification probability in the current refinement domain.

Based on these random vectors, we compute the location of the split. We choose a set of *splitting locations* in each dimension  $v_i \in \mathcal{D}_{U_i}, i = 1, \dots, M$  that ideally completely separate the support of  $\mathbf{Z}^+$  from  $\mathbf{Z}^0$  (see Figure 3). Due to the likely disjoint support of each auxiliary random vector, the split should instead confine a maximum of  $\mathbf{Z}^+$ 's probability mass to one side of the split, while confining a maximum of  $\mathbf{Z}^0$ 's probability mass to the other side of

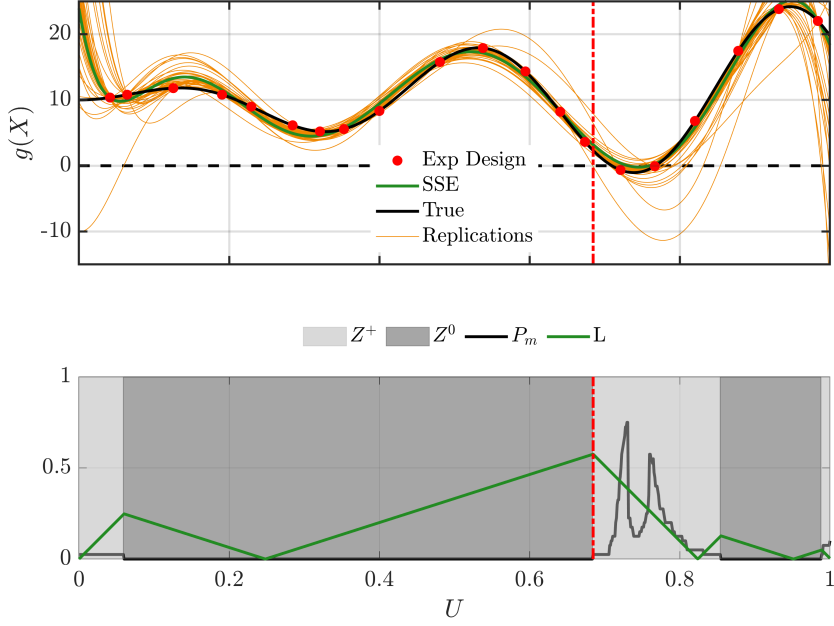


Figure 3: Partitioning strategy exemplified on a one dimensional function showing the bootstrap replications and the resulting support of  $Z^+$  and  $Z^0$  as well as the objective function  $L$ . The splitting location  $v$  is highlighted as a vertical, dashed red line. The plots are shown in the quantile space  $\mathcal{D}_U$ .

the split. Denoting by  $Z_i^+$  and  $Z_i^0$  the marginals of the auxiliary random vectors in the  $i$ -th dimension, we pick the splitting location

$$\hat{v}_i = \arg \max_{v_i \in \mathcal{D}_{X_i}} L_i(v_i), \quad \text{with} \quad L_i(v_i) \stackrel{\text{def}}{=} -1 + \max \begin{cases} \mathbb{P}[Z_i^+ \leq v_i] + \mathbb{P}[Z_i^0 > v_i], \\ \mathbb{P}[Z_i^+ > v_i] + \mathbb{P}[Z_i^0 \leq v_i], \end{cases} \quad (18)$$

where the objective function  $L_i$  characterizes the split properties by returning the maximum of the respective auxiliary probability masses in the resulting subdomains. The splitting location  $\hat{v}_i$  that maximizes  $L_i$  thus splits the initial domain  $\mathcal{D}_{\mathbf{X}}$  into two subdomains  $\mathcal{D}_{X_i}^{(1)}$  and  $\mathcal{D}_{X_i}^{(2)}$  that fulfil the initially stated goal of optimally separating regions that correctly predict failure/safety from regions that do so incorrectly. The objective function  $L_i$  is bounded between  $[0, 1]$ , where a value of 1 indicates a *perfect split* that creates one subdomain with only non-zero and one subdomain with all zero misclassification probability regions.

To ultimately choose a *splitting direction*  $d \in \{1, \dots, M\}$ , we compare the values of the objective functions  $L_i(\hat{v}_i)$  and split along the dimension that achieves the best split, i.e.,

$$d = \arg \max_{i \in \{1, \dots, M\}} L_i(\hat{v}_i). \quad (19)$$

In practice, a sufficiently large sample of the auxiliary random vectors  $\mathbf{Z}^0$  and  $\mathbf{Z}^+$  is used to conduct the computations of this section. This sample is readily available, as the SSE surrogate model can be evaluated at a negligible computational cost.

### No misclassified sample points

It might occur that the the algorithm does not detect misclassified sample points at the time of partitioning. When this happens, the misclassification probability  $P_m$  (Eq. (9)) is zero everywhere and consequently the auxiliary random vector  $\mathbf{Z}^+$  is not defined. Because of the preceding refinement selection criterion in Eq. (15) this can only happen at the first step of the algorithm or after the intermediate re-prioritization criterion (Eq. (16)) has been triggered.

To let the algorithm proceed in this case, the definition of the auxiliary random vectors is updated to discriminate based on the probability to predict values within a predefined empirical quantile around the limit-state surface. More formally,  $P_m$  in Eq. (17) is then replaced with

$$P_t(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{B} \sum_{b=1}^B \mathbf{1}_{\mathcal{D}_t}^{(b)}(\mathbf{x}), \quad (20)$$

where  $\mathbf{1}_{\mathcal{D}_t}^{(b)}$  is defined with an empirical quantile  $t$  such that  $\mathbb{P}[|g_{\text{SSE}}(\mathbf{X})| \leq t] = \varepsilon_t$  as

$$\mathbf{1}_{\mathcal{D}_t}^{(b)}(\mathbf{x}) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } |g_{\text{SSE}}^{(b)}(\mathbf{x})| \leq t, \\ 0, & \text{if } |g_{\text{SSE}}^{(b)}(\mathbf{x})| > t. \end{cases} \quad (21)$$

The value  $\varepsilon_t = 0.01$  has proven to be a good choice in practice.

### 2.2.3 Sample enrichment

After partitioning a selected refinement domain, in the original SSE algorithm points are added such that every subdomain contains exactly a total of  $N_{\text{ref}}$  points. Because the partitioning strategy now allows the creation of domains with considerably different probability mass, the original sample enrichment approach is problematic. As an example, splitting in the tails of the input distribution will result in one large domain with  $N_{\text{ref}}$  points from the previous step and one smaller domain with 0 points. In this case, no new points would be added to the large domain. The subsequently constructed residual expansion would be based on (almost) the same information as the parent domain expansion and be prone to over-fitting. We therefore employ a different strategy here, which always adds  $N_{\text{ref}}$  points to every domain, independent of the existing points inside that domain. We refer to those points as the *sample budget* here.

The second change pertains to the placement of the sample budget. The original algorithm used random sampling to place points uniformly in the refinement subdomain. To exploit the fact that the SSE for reliability applications needs to be more accurate near the limit-state surface, we propose a slightly modified strategy: we split the sample budget and randomly place

$$N_{\text{uni}} = \max \begin{cases} \frac{N_{\text{ref}}}{2} - \frac{N_{\text{curr}}}{2} \\ 0 \end{cases} \quad (22)$$

points uniformly in the refinement domain quantile space, where  $N_{\text{curr}}$  is the number of existing points in the refinement domain. The remainder of the sample budget is used to randomly place sample points in the subset of the refinement domain with non-zero misclassification probability  $P_m$ . This corresponds to sampling the auxiliary random vector  $\mathbf{Z}^+$  defined in Eq. (17). We do this to ensure stability of the residual expansions that are known to deteriorate in accuracy when the experimental design clusters in a subspace.

In the extreme case of no existing points (i.e.,  $N_{\text{curr}} = 0$ ), this places half of the sample budget uniformly and the other half according to  $\mathbf{Z}^+$ . In the other extreme case of  $N_{\text{curr}} \geq N_{\text{ref}}$ , the whole sample budget is devoted to sampling  $\mathbf{Z}^+$ .

At the first iteration, to construct the initial expansion, the enrichment occurs randomly in the entire input quantile space. We observed from repeated tests that the algorithm is less stable if the initial expansion is less accurate. To account for this, we start out with an initial experimental design of size  $2N_{\text{ref}}$  and switch to the experimental design size of  $N_{\text{ref}}$  in all subsequent iterations.

Finally, we note that  $N_{\text{ref}}$  is a tuning parameter of the algorithm. Larger values of  $N_{\text{ref}}$  lead to more accurate local expansions, resulting in increased stability of the algorithm at the cost of slower convergence rates. Vice-versa, smaller values of  $N_{\text{ref}}$  can significantly increase the convergence rates, but can potentially produce inaccurate limit-state function approximations that lead to biased failure probability estimates.

#### 2.2.4 Stochastic spectral embedding-based reliability algorithm

Taking all presented ingredients, the full stochastic spectral embedding-based reliability (SSER) algorithm is written as follows:

**Input:**

- Input parameters  $\mathbf{X}$ , limit-state function  $g$
- Expansion parameters  $N_{\text{ref}}, p_{\text{max}}, B$
- Algorithm parameters  $N_{\text{ED}}, \varepsilon_{\hat{\beta}}, \varepsilon_{\hat{p}_f}, \varepsilon_t$

**1. Initialization:**

- (a)  $\mathcal{D}_{\mathbf{X}}^{0,1} = \mathcal{D}_{\mathbf{X}}$
- (b) Sample from the input distribution an initial experimental design  $\mathcal{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(2N_{\text{ref}})}\}$
- (c) Calculate the truncated expansion  $\widehat{\mathcal{R}}_S^{0,1}(\mathbf{X})$  of  $g(\mathbf{X})$  in the full domain  $\mathcal{D}_{\mathbf{X}}^{0,1}$ , retrieve its approximation error  $\mathcal{E}^{0,1}$  and initialize  $\mathcal{T} = \{(0, 1)\}$
- (d)  $\mathcal{R}^1(\mathbf{X}) = g(\mathbf{X}) - \widehat{\mathcal{R}}_S^{0,1}(\mathbf{X})$

**2. For**  $(\ell, p) = k_{\text{refine}}$  **from** Eq. (15)

- (a) Split the current subdomain  $\mathcal{D}_{\mathbf{X}}^{\ell,p}$  in 2 sub-parts  $\mathcal{D}_{\mathbf{X}}^{\ell+1, \{s_1, s_2\}}$  according to Eq. (19), where  $s_1$  and  $s_2$  are unique indices for domains on the  $\ell$ -th level, and update  $\mathcal{T}$
- (b) **For each split**  $s \in \{s_1, s_2\}$

- i. Enrich sample  $\mathcal{X}$  with  $\min\{N_{\text{ref}}, N_{\text{ED}} - |\mathcal{X}|\}$  new points inside  $\mathcal{D}_{\mathbf{X}}^{\ell+1,s}$
- ii. **If**  $N_{\text{ref}}$  new points inside  $\mathcal{X}$ 
  - A. Create the truncated expansion  $\widehat{\mathcal{R}}_S^{\ell+1,s}(\mathbf{X})$  of the residual  $\mathcal{R}^{\ell+1}(\mathbf{X})$  in the current subdomain using a subset of  $\mathcal{X}$  inside  $\mathcal{D}_{\mathbf{X}}^{\ell+1,s}$
  - B. Update the residual on the next level in the current subdomain  $\mathcal{R}^{\ell+2}(\mathbf{X}) = \mathcal{R}^{\ell+1}(\mathbf{X}) - \widehat{\mathcal{R}}_S^{\ell+1,s}(\mathbf{X})$
- iii. Retrieve  $\mathcal{E}^{\ell+1,s}$  from Eq. (14)
- (c) Retrieve  $\hat{P}_f$  from Eq. (6) and its variance from Eq. (12)
- (d) **If** stopping criterion in Eq. (23) is met or less than two new expansions were created, terminate the algorithm, otherwise go back to 2

### 3. Termination

- (a) Return  $\hat{P}_f$  and its variance

A graphical sketch of the algorithm for a two-dimensional toy problem is shown in Figure 4. The figure shows the first two iterations of the SSE construction in the quantile and real space, highlighting the terminal domains  $\mathcal{T}$  at each step in orange. The initialization step in Figure (a) shows the initial experimental design and global domain  $\mathcal{D}_{\mathbf{X}}^{0,1}$  and their counterpart in the quantile space  $\mathcal{D}_{\mathbf{U}}^{0,1}$ . At the first iteration, the algorithm chooses the only available refinement domain  $k_{\text{refine}} = (0, 1)$ , partitions it into two subdomains, enriches the experimental design and constructs two additional residual expansions. At the second iteration in Figure (c), the algorithm refines  $\mathcal{D}_{\mathbf{X}}^{(1,2)}$  out of two terminal domains  $\mathcal{T} = \{(1, 1), (1, 2)\}$ , because it has the largest contribution to the total failure probability variance. Refinement then continues until the stopping criterion is met.

#### 2.2.5 Stopping criterion

The algorithm can be terminated based on any stopping criterion from the active learning literature (Moustapha et al., 2021). In the present implementation, we opt for a criterion based on the stability of the reliability index bounds, also known as *beta bounds*. The generalized reliability index can be computed from the failure probability as  $\beta = -\Phi^{-1}(P_f)$ , where  $\Phi^{-1}$  is the inverse normal CDF Ditlevsen (1979). Using the 95% confidence intervals on  $\hat{\beta}$  from the bootstrap replications as defined for the failure probability in Eq. (13), the stopping criterion is given by

$$\frac{\bar{\hat{\beta}} - \hat{\beta}}{\hat{\beta}} < \varepsilon_{\hat{\beta}}, \quad (23)$$

where the failure threshold is heuristically set to  $\varepsilon_{\hat{\beta}} = 3\%$ . For stability, we abort the algorithm only if this criterion is fulfilled in three consecutive iterations.

Additionally, the algorithm is terminated if the computational budget  $N_{\text{ED}}$ , corresponding to the total number of admissible limit-state evaluations, has been exhausted.

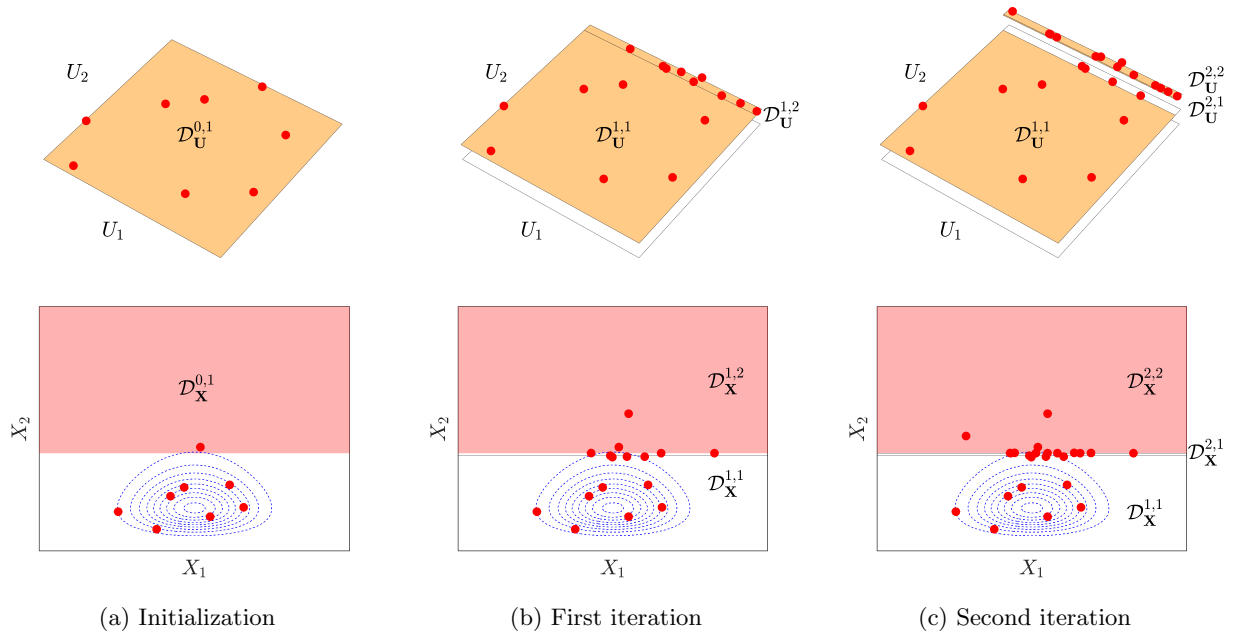


Figure 4: Graphical representation of the first steps of the active learning algorithm described in Section 2.2.4 for a two-dimensional toy problem with independent inputs. Upper row: partitioning in the quantile space; lower row: partitioning in the unbounded real space with  $f_{\mathbf{X}}$  contour lines in dashed blue. Red dots show the adaptive experimental design with  $N_{\text{ref}} = 4$ . The terminal domains  $\mathcal{T}$  are highlighted in orange. The red area denotes the failure domain. In this illustrative example the failure probability is  $\hat{P}_f \approx 4.2 \cdot 10^{-2}$ .



### 2.3 Conditional failure probability estimation

In the SSE approach, the total failure probability is decomposed into a sum of weighted, conditional failure probabilities (see Eq. (6)). As a side effect of the proposed partitioning strategy (Section 2.2.2), these conditional failure probabilities typically end up being quite large or negligibly small.

Because of the SSE surrogate, simulation methods can be used at a low computational cost. The case of large conditional failure probabilities can be treated easily with simple Monte Carlo simulation. The case of small conditional failure probabilities, however, is more challenging even with surrogate models. Small failure probabilities should not be underestimated by the algorithm in large domains, because they can contribute significantly to the total failure probability. In cases where the Monte Carlo simulation does not detect failure domains with the provided computational budget, we therefore switch to subset simulation (Au and Beck, 2001) that is known to effectively estimate very small failure probabilities (Moustapha et al., 2021).

## 3 Applications

In this section, the proposed SSER algorithm is applied to four problems of varying complexity. These applications were selected to benchmark the algorithm performance in different types of reliability problems: the first two applications have analytical limit-state functions that are challenging for many standard reliability methods. They are the well-known *four-branch function* with two input parameters and four distinct failure regions, and a two-dimensional *piecewise linear* reliability problem that is known to be challenging for subset simulation. The third and fourth applications are engineering models that involve finite element simulations. The third application is an engineering model of a *five-story frame* with  $M = 21$  mutually dependent input parameters. Lastly, the fourth application is a continuum mechanical problem of a *plate with a hole*, where the Young's modulus is parametrized by a high dimensional ( $M = 869$ ) random field.

All computations were done in MATLAB with the UQLab uncertainty quantification framework Marelli and Sudret (2014). Specifically, we used the input module Lataniotis et al. (2021) for modeling the random inputs, the PCE module Marelli et al. (2021) for the domain-wise spectral expansions with bootstrap replications and the reliability module Marelli et al. (2021) for computing the conditional failure probabilities.

A summary of the results and comparisons to other state-of-the-art reliability algorithms is given in Table 1, and a detailed rundown of each is provided in the following.

	<b>Four-branch</b>	<b>Piecewise linear</b>	<b>Five-storey frame</b>	<b>Plate with a hole</b>	
	Section 3.1	Section 3.2	Section 3.3	Section 3.4	
Reference	Method	MCS	MCS	MCS	SuS Uribe et al. (2020)
	$\hat{P}_f$	$4.46 \cdot 10^{-3}$	$3.2 \cdot 10^{-5}$	$1.49 \cdot 10^{-6}$	$3.75 \cdot 10^{-6}$
	$[\underline{\hat{P}}_f; \overline{\hat{P}}_f]$	$[4.42; 4.51] \cdot 10^{-3}$	$[3.1; 3.3] \cdot 10^{-5}$	$[1.1; 1.8] \cdot 10^{-6}$	$[3.59; 3.91] \cdot 10^{-6a}$
	$\hat{\beta}$	2.62	4.00	4.67	4.48
	$[\underline{\hat{\beta}}; \overline{\hat{\beta}}]$	[2.61; 2.62]	[3.99; 4.00]	[4.64; 4.73]	[4.47; 4.48]
	$N$	$10^7$	$10^8$	$10^8$	$1.62 \cdot 10^6$
Competing algorithms	Method	AK-MCS+U Echard et al. (2011)	–	sPCE+MCS Blatman and Sudret (2010) <sup>b</sup>	iCEred Uribe et al. (2020)
	$\hat{P}_f$	$4.42 \cdot 10^{-3}$	–	$3.9 \cdot 10^{-7}$	$3.57 \cdot 10^{-6}$
	$[\underline{\hat{P}}_f; \overline{\hat{P}}_f]$	–	–	–	$[3.21; 3.93] \cdot 10^{-6}$
	$\hat{\beta}$	2.62	–	4.94	4.49
	$[\underline{\hat{\beta}}; \overline{\hat{\beta}}]$	–	–	–	[4.47; 4.51]
	$N$	126	–	450	2,396 Uribe et al. (2020) <sup>c</sup>
Proposed algorithm	Method	SSER	SSER	SSER	SSER
	$\hat{P}_f$	$4.44 \cdot 10^{-3}$	$3.16 \cdot 10^{-5}$	$1.49 \cdot 10^{-6}$	$3.27 \cdot 10^{-6}$
	$[\underline{\hat{P}}_f; \overline{\hat{P}}_f]$	$[4.43; 4.47] \cdot 10^{-3}$	$[3.16; 3.16] \cdot 10^{-5}$	$[1.20; 1.64] \cdot 10^{-6}$	$[3.26; 5.47] \cdot 10^{-6}$
	$\hat{\beta}$	2.62	4	4.67	4.50
	$[\underline{\hat{\beta}}; \overline{\hat{\beta}}]$	[2.60; 2.63]	[4; 4]	[4.65; 4.71]	[4.40; 4.51]
	$N$	270	480	480	1,400
	$[\underline{N}; \overline{N}]^d$	[180; 330]	[320; 720]	[320; 720]	[1,200; 2,000]

Table 1: *Applications summary*: if not given with an external reference, the values were computed with UQLab Marelli and Sudret (2014); Marelli et al. (2021). Values in square brackets correspond to the 95% confidence intervals.  $N$  denotes the number of limit-state evaluations needed to compute the reported values. For SSER confidence intervals are also given for the number of limit-state evaluations until convergence from a set of independent runs. The used methods are Monte Carlo simulation (MCS), subset simulation (SuS), adaptive kriging Monte Carlo simulation with  $U$  learning function (AK-MCS+U), sparse polynomial chaos expansions with Monte Carlo simulation (sPCE+MCS), improved CE method with failure-informed dimension reduction (iCEred).

<sup>a</sup>Estimated from 100 independent SuS runs with reported  $CV_{\hat{P}_f} = 0.215$  as  $\hat{P}_f \cdot (1 \pm 2CV_{\hat{P}_f}/\sqrt{100})$ .

<sup>b</sup>Reported value  $\hat{P}_f = 3.9 \cdot 10^{-7}$  does not lie within reference confidence intervals.

<sup>c</sup>Additional 675 gradient evaluations.

<sup>d</sup>From independent algorithm runs to reach convergence.

### 3.1 Four-branch function

The four-branch function is a common benchmark in the active learning literature Schuermans and Gemert (2005); Echard et al. (2011). It simulates a series system with four distinct limit-state surfaces. The input random parameter is distributed according to a bivariate standard normal distribution, i.e.,  $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{1}_2)$ . The limit-state function is given analytically as

$$g(\mathbf{X}) = \min \begin{cases} 3 + 0.1(X_1 - X_2)^2 - \frac{X_1 + X_2}{\sqrt{2}} \\ 3 + 0.1(X_1 - X_2)^2 + \frac{X_1 + X_2}{\sqrt{2}} \\ X_1 - X_2 + \frac{6}{\sqrt{2}} \\ X_2 - X_1 + \frac{6}{\sqrt{2}}. \end{cases} \quad (24)$$

The reference failure probability of this problem from Monte Carlo simulation is  $P_f = 4.46 \cdot 10^{-3}$  ( $\beta = 2.62$ ) obtained with  $N = 10^7$  limit-state evaluations.

We perform 50 independent runs of SSER on this problem, with different random seeds and initial designs. The number of refinement samples was set to  $N_{\text{ref}} = 15$  and the maximum polynomial degree for the residual expansions to  $p_{\text{max}} = 2$ . The domain-wise accuracy was estimated with  $B = 500$  bootstrap replications.

The convergence to the reference reliability index for a selected run of SSER and all 50 independent runs is shown in Figure 5. The median number of steps required to reach the stopping criterion is 7, corresponding to  $\underbrace{2N_{\text{ref}}}_{\text{initial}} + (6 \cdot 2)N_{\text{ref}} = 210$  limit-state evaluations.

A more in-depth look at the SSER performance is shown in Figure 6, where four selected steps of the algorithm run from Figure (a) are displayed. It is important to note that due to the Gaussian probability measure, the importance of regions shrinks exponentially with the distance from the domain centre (i.e.,  $(0, 0)$ ). The initial approximation is a paraboloid that underestimates the failure probability by misclassifying regions near the domain centre. At the second step, SSER partitions the initial domain along  $d = 2$  at  $\hat{X}_2 \approx 2$  corresponding to the 98th percentile (i.e.,  $\hat{u}_2 = 0.98$ ). By Step 6 the overall approximation is extremely accurate near the centre of the domain, while it remains poor near the unimportant four corners of the safe domain. Three steps later, at Step 9, the stopping criterion is met and the algorithm is terminated.

### 3.2 Piecewise linear function

The *piecewise linear function* was devised in Breitung (2019) to break the subset simulation algorithm Au and Beck (2001). This function has two limit-state surfaces, one with a large and the second with a negligible failure probability. By providing a steep limit-state function in the direction of the second limit-state surface, the subset simulation algorithm overlooks the first limit-state surface.

The input random parameter is again distributed according to a bivariate standard nor-

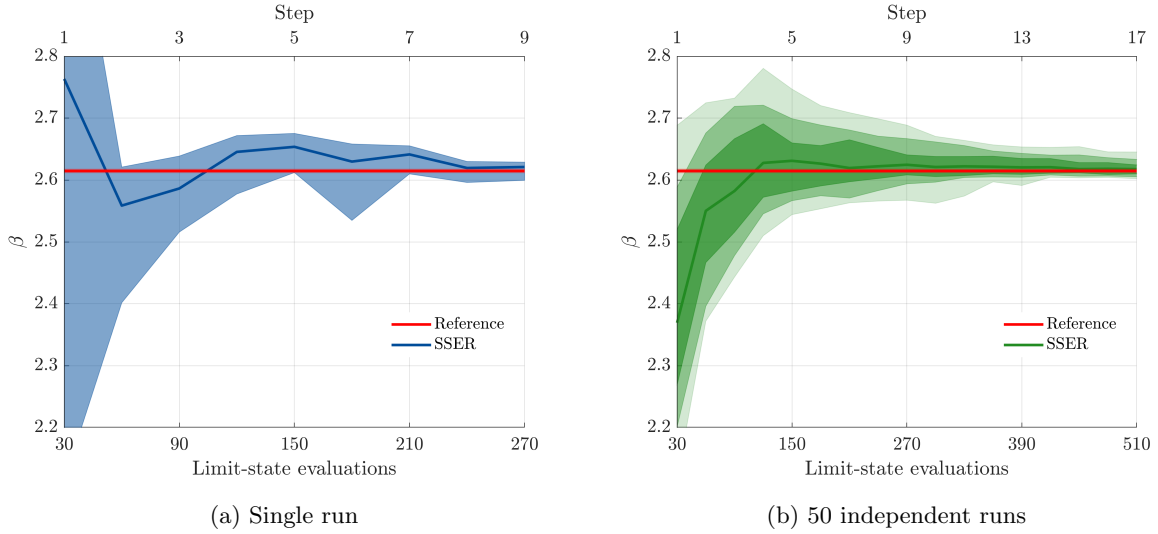


Figure 5: *Four-branch function*: Convergence to the reference reliability index  $\beta$  as a function of limit-state evaluations and algorithm steps. The 95% confidence intervals in Figure (a) are based on the bootstrap replications. This figure shows the convergence until the stopping criterion in Eq. (23) is met at Step 9 of SSER. The bounds in Figure (b) are the 90%, 75% and 50% confidence intervals from the mean predictions of the independent SSER runs.

mal distribution, i.e.,  $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{1}_2)$ , and the limit-state function is given analytically by

$$g(\mathbf{X}) = \min \begin{cases} g_1(\mathbf{X}) = \begin{cases} 4 - X_1, & \text{if } X_1 > 3.5; \\ 0.85 - 0.1 \cdot X_1, & \text{if } X_1 \leq 3.5; \end{cases} \\ g_2(\mathbf{X}) = \begin{cases} 0.5 - 0.1 \cdot X_2, & \text{if } X_2 > 2; \\ 2.3 - X_2, & \text{if } X_2 \leq 2. \end{cases} \end{cases} \quad (25)$$

We obtain a reference failure probability of  $P_f = 3.2 \cdot 10^{-5}$  ( $\beta = 4$ ) for this problem from Monte Carlo simulation with  $N = 10^8$  samples.

We perform 50 independent runs of SSER, with different random seeds. The number of refinement samples was set to  $N_{\text{ref}} = 40$  and the maximum polynomial degree for the residual expansions to  $p_{\text{max}} = 6$ . The domain-wise accuracy was estimated with  $B = 500$  bootstrap replications.

The convergence of SSER to the reference reliability index is shown in Figure 7. As intended by this limit-state function, most runs of the algorithm initially overestimate the reliability index. Due to its refinement domain selection strategy, however, SSER later returns to the high failure probability domains and accurately estimates the reliability index. The median number of steps required to reach the stopping criterion is 5, corresponding to  $\underbrace{2N_{\text{ref}}}_{\text{initial}} + (4 \cdot 2)N_{\text{ref}} = 400$  limit-state evaluations.

Figure 8 reveals why SSER initially overestimates the reliability index and underestimates

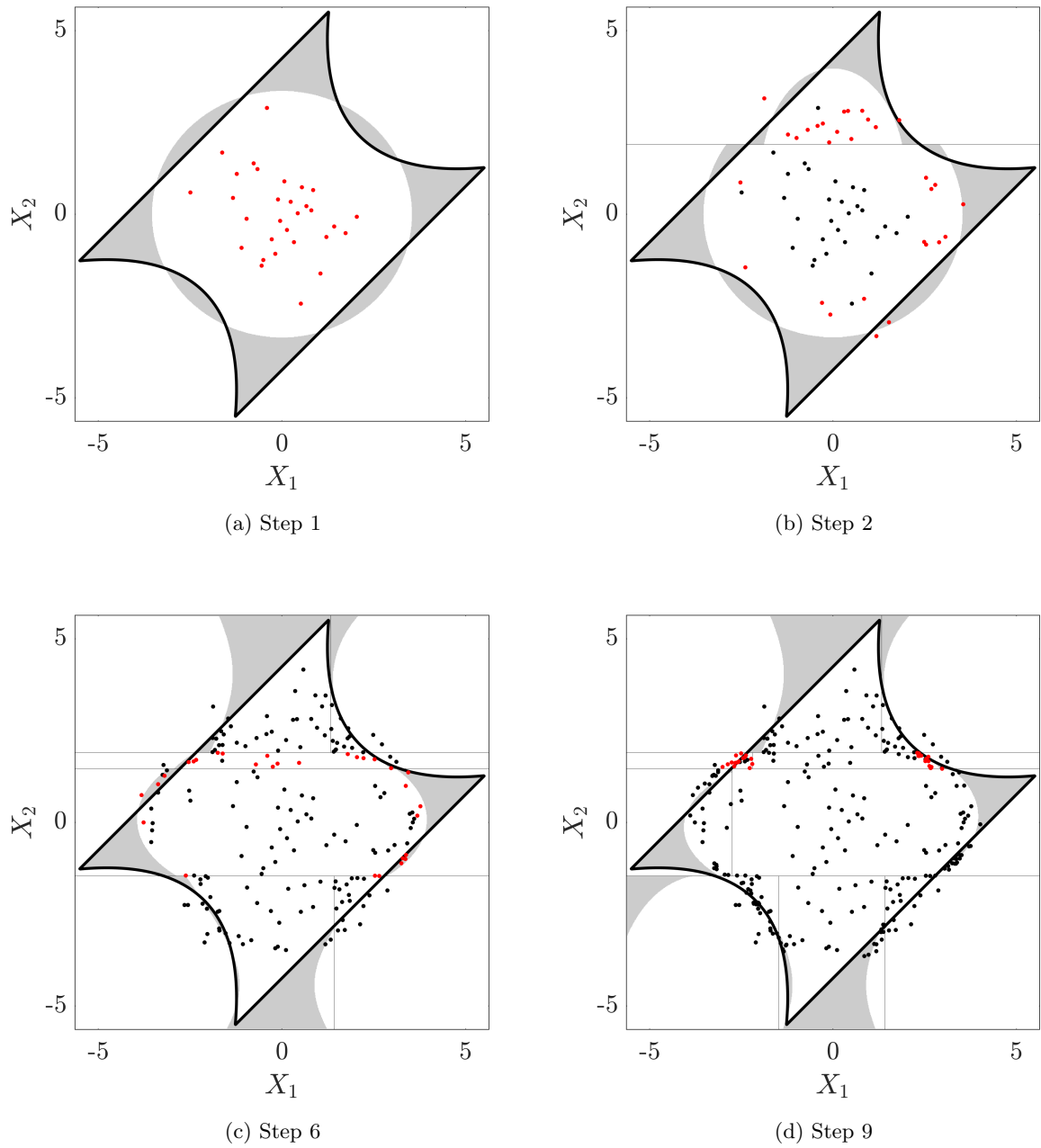


Figure 6: *Four-branch function*: Misclassification of failure/safe points for selected steps of SSER. The white areas were classified correctly, while the gray areas were misclassified. The thick black line indicates the limit-state surface. The domain bounds identified by the SSER are shown as thin black lines. The black points are the used experimental design and the red points are the points added at the shown step of the algorithm. The convergence of the reliability index estimator is shown in Figure (a).

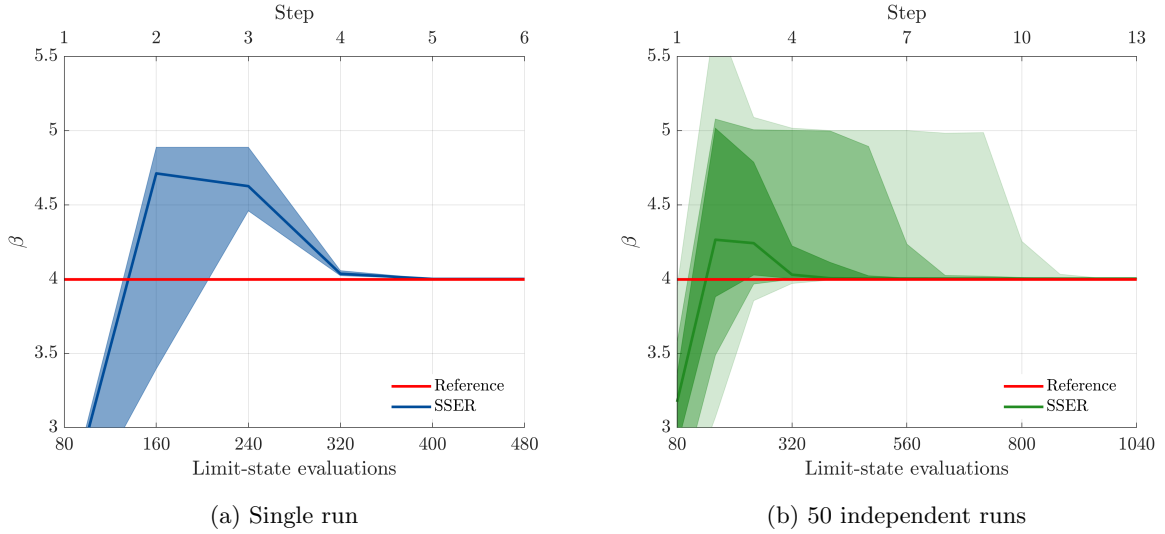


Figure 7: *Piecewise linear function*: Convergence to the reference reliability index  $\beta$  as a function of limit-state evaluations and algorithm steps. The 95% confidence intervals in Figure (a) are based on the bootstrap replications. This figure shows the convergence until the stopping criterion in Eq. (23) is met at Step 6 of SSER. The bounds in Figure (b) are the 90%, 75% and 50% confidence intervals from the mean predictions of the independent SSER runs.

the failure probability. After the initial approximation, at Step 2, the algorithm partitions along  $d = 2$  to isolate the region where it suspects the highest misclassification probability. In the following steps, it focuses again on the big region that incorporates the larger failure domain and accurately approximates it at Step 4. Two steps later, at Step 6, the stopping criterion is fulfilled and SSER is terminated.

### 3.3 Five-story frame

In this example we consider the five-story structural frame described in Figure 9. This example has been investigated before in Liu and Der Kiureghian (1991); Wei and Rahman (2007); Blatman and Sudret (2010). It consists of eight different structural elements with uncertain dimensions and material properties. The structure is subject to uncertain, horizontal loads  $P_1$ ,  $P_2$  and  $P_3$ .

The input parameters are collected in a random vector  $\mathbf{X} = \{P_1, \dots, A_{21}\}$  and sorted according to Figure (c). These parameters are not mutually independent. Instead, we impose a *Gaussian copula* (Nelsen, 2006) such that the joint input distribution function can be written as

$$F_{\mathbf{X}}(\mathbf{x}) = \Phi_{21}(\Phi^{-1}(F_{X_1}(x_1)), \dots, \Phi^{-1}(F_{X_{21}}(x_{21})); \mathbf{R}). \quad (26)$$

In this expression  $F_{X_i}$  are the *cumulative distribution functions* (CDF) of the parameters as listed in Figure (c),  $\Phi_{21}(\mathbf{U}; \mathbf{R})$  is the CDF of a 21-variate Gaussian distribution with

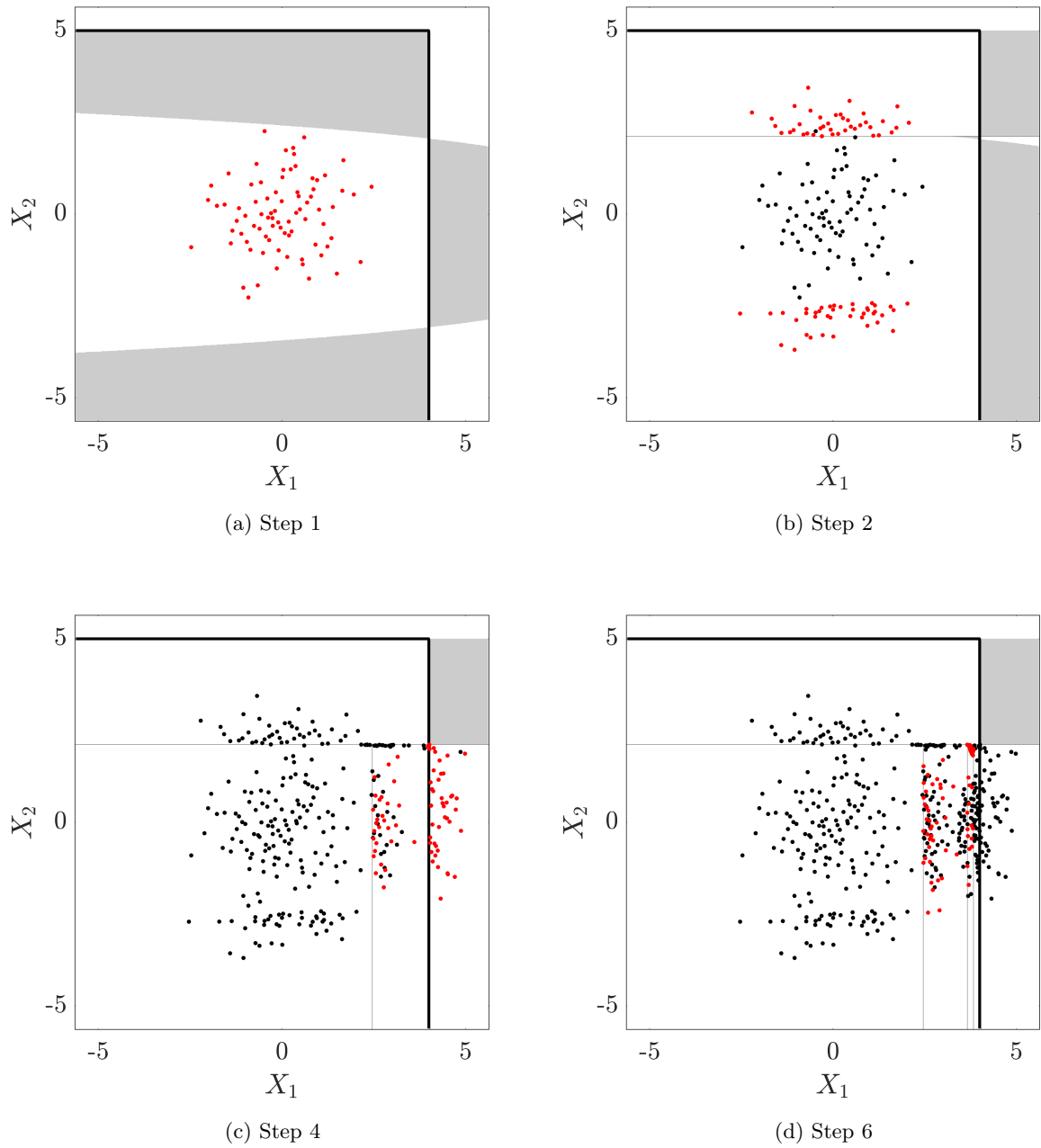


Figure 8: *Piecewise linear function*: Misclassification of failure/safe points for selected steps of SSER. The white areas were classified correctly, while the gray areas were misclassified. The thick black line indicates the limit-state surface. The domain bounds identified by SSER are shown as thin black lines. The black points are the used experimental design and the red points are the points added at the shown step of the algorithm. The convergence of the reliability index estimator is shown in Figure (a).

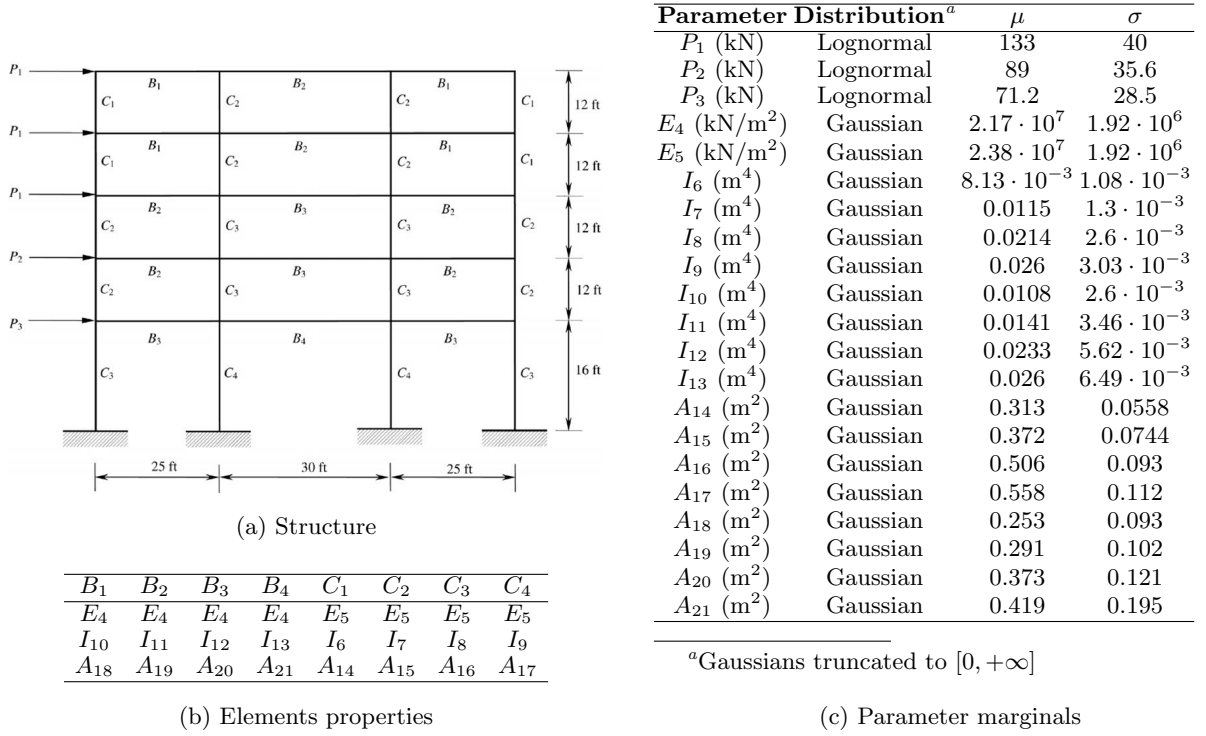


Figure 9: *Five-story frame*: Problem setup with (a) static system, (b) a table of element properties and (c) a table of the parameter marginal distributions. The element properties are listed as eight sets of Young’s moduli  $E$ , second moment of area  $I$  and area  $A$  for each element type  $B_i$  and  $C_i$  with  $i \in \{1, \dots, 4\}$ . The parameter marginals are given per input parameter where  $\mu$  and  $\sigma$  denote the mean and standard deviation respectively. For the truncated Gaussians those moments apply to their untruncated variants.

mean  $\mathbf{0}$  and correlation matrix  $\mathbf{R} \in \mathbb{R}^{21 \times 21}$  and  $\Phi^{-1}$  is the inverse CDF of the standard Gaussian distribution. Correlation is assumed only between some parameters, as reflected by the entries of the correlation matrix  $\mathbf{R}$  detailed below

**Geometrical properties** The geometrical properties are assumed to be correlated with

$$\rho_{A_i, I_j} = \rho_{I_i, I_j} = \rho_{A_i, A_j} = 0.13. \text{ For geometrical properties belonging to the same element type, a stronger correlation of } \rho_{A_i, I_{i+8}} = 0.95, i = 6 \dots, 13 \text{ is assumed.}$$

**Material properties** The two Young’s moduli are correlated with  $\rho_{E_4, E_5} = 0.9$ .

The quantity of interest is the top-floor displacement  $u(\mathbf{X})$ . This displacement can be computed as the solution of an elastic structural mechanics problem that is approximately solved with the linear finite element method. We consider the problem of determining the probability that this top-floor displacement exceeds a threshold of  $u_{\text{th}} = 9$  cm. This yields the limit-state function

$$g(\mathbf{X}) = u_{\text{th}} - u(\mathbf{X}). \quad (27)$$

This leads to a reference failure probability of  $P_f = 1.49 \cdot 10^{-6}$  ( $\beta = 4.67$ ) computed with



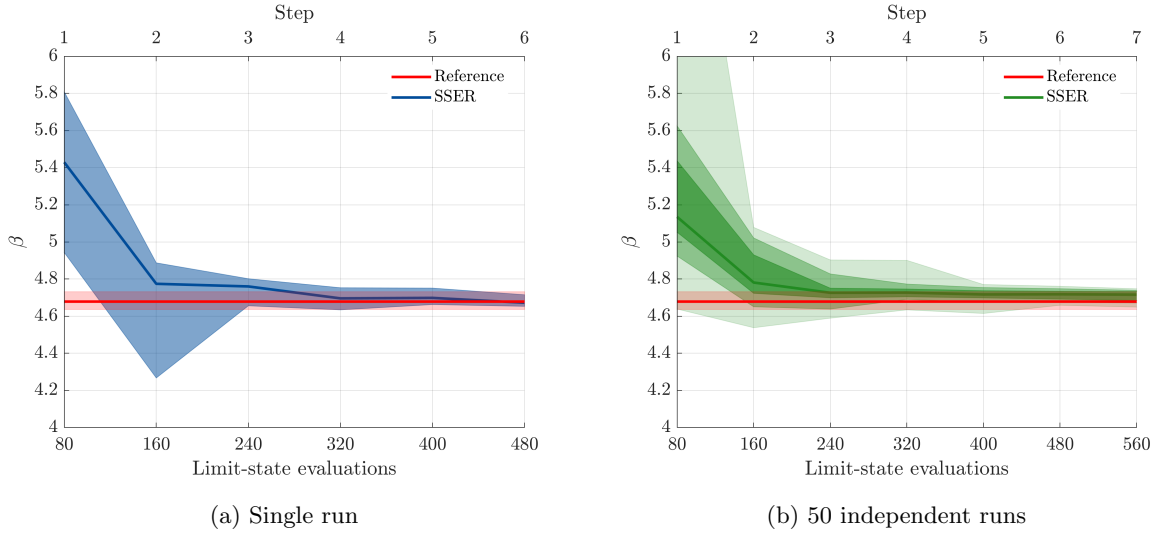


Figure 10: *Five-story frame*: Convergence to the reference reliability index  $\beta$  as a function of limit-state evaluations and algorithm steps. The 95% confidence intervals in Figure (a) are based on the bootstrap replications. This figure shows the convergence until the stopping criterion in Eq. (23) is met at Step 6 of SSER. The bounds in Figure (b) are the 90%, 75% and 50% confidence intervals from the mean predictions of the independent SSER runs.

Monte Carlo simulation and  $N = 10^8$  evaluations of the limit-state function.

We perform 50 independent runs of SSER on this problem, with different random seeds. The number of refinement samples is set to  $N_{\text{ref}} = 40$  and the maximum polynomial degree for the residual expansions to  $p_{\text{max}} = 4$ . The domain-wise accuracy is estimated with  $B = 500$  bootstrap replications.

The convergence to the reference reliability index for a single run of SSER and all 50 runs is shown in Figure 10. The median number of steps required to reach the stopping criterion is 3, corresponding to  $\underbrace{2N_{\text{ref}}}_{\text{initial}} + (2 \cdot 2)N_{\text{ref}} = 240$  limit-state evaluations. The stopping criterion for the shown run in Figure (a) is met at Step 6.

### 3.4 Plate with a hole

This last example was previously analysed in Uribe et al. (2020) and its description here follows closely the description there. Consider a two-dimensional, square steel plate under plane stress conditions with a hole in the middle (see Figure 11). The plate is defined on a square domain  $\mathcal{D}_{\boldsymbol{\xi}}$  with side lengths 0.32 m, thickness 0.01 m and a centre hole of radius 0.02 m. The coordinates in the domain are parametrized by  $\boldsymbol{\xi} = [\xi_1, \xi_2] \in \mathcal{D}_{\boldsymbol{\xi}}$ . The plate is assumed to deform elastically with an isotropic Young's modulus  $E(\boldsymbol{\xi})$ . Neglecting body-

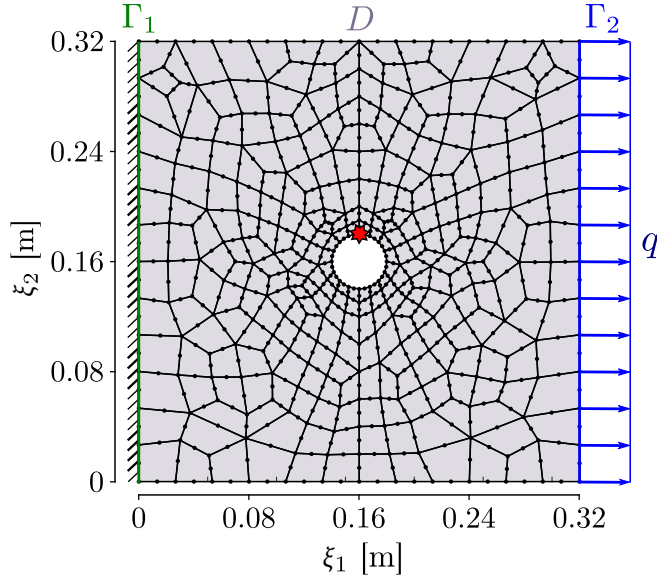


Figure 11: *Plate with a hole*: Finite element discretization with 282 eight-node serendipity quadrilateral elements (Uribe et al., 2020). The red star  $*$  marks the location of the control node for the stress exceedance criterion in Eq. (31).

forces results in the following system of partial differential equations (PDEs)

$$\frac{E(\boldsymbol{\xi})}{2(1+\nu)} \nabla^2 \mathbf{u}(\boldsymbol{\xi}) + \frac{E(\boldsymbol{\xi})}{2(1-\nu)} \nabla(\nabla \mathbf{u}(\boldsymbol{\xi}))^\top = 0, \quad (28)$$

where  $\mathbf{u}(\boldsymbol{\xi})$  is the 2D displacement field and  $\nu = 0.29$  is the steel Poisson's ratio. As sketched in Figure 11 a Dirichlet boundary condition is applied on the left domain border such that  $\mathbf{u}(\boldsymbol{\xi}) = 0$  for  $\boldsymbol{\xi} \in \Gamma_1$ . Additionally, a Neumann boundary condition is applied on the right domain border such that  $\sigma_{11}(\boldsymbol{\xi}) = q$  for  $\boldsymbol{\xi} \in \Gamma_2$ , where  $\sigma_{11}$  is the stress component in the first direction.

The Young's modulus  $E(\boldsymbol{\xi})$  is modelled as lognormal random field with mean  $\mu_E = 2 \cdot 10^5$  MPa and standard deviation  $\sigma_E = 3 \cdot 10^4$  MPa. We use an isotropic exponential kernel  $k(\boldsymbol{\xi}, \boldsymbol{\xi}') = \exp(-\ell^{-1} \|\boldsymbol{\xi} - \boldsymbol{\xi}'\|_2)$  as the autocorrelation function of the underlying Gaussian random field with correlation length  $\ell = 0.04$  m. To parametrize this random field, we resort to the Karhunen-Loève expansion Ghanem and Spanos (1991) and write

$$E(\boldsymbol{\xi}) \approx \widehat{E}(\boldsymbol{\xi}, \boldsymbol{\theta}) \stackrel{\text{def}}{=} \exp \left( \mu_{E'} + \sum_{k=1}^K \sqrt{\lambda_k} \varphi_k(\boldsymbol{\xi}) \theta_k \right), \quad (29)$$

where  $\lambda_k$  and  $\varphi_k(\boldsymbol{\xi})$  are the ordered (i.e.,  $\lambda_k \geq \lambda_{k+1}$ ) eigenvalues and eigenfunctions of the covariance operator  $C(\boldsymbol{\xi}, \boldsymbol{\xi}') \stackrel{\text{def}}{=} \sigma_{E'}^2 k(\boldsymbol{\xi}, \boldsymbol{\xi}')$ . The mean and variance of the underlying Gaussian field are  $\mu_{E'}$  and  $\sigma_{E'}^2$ , respectively and  $\boldsymbol{\theta} \in \mathbb{R}^k$  denotes the standard Gaussian vector of random coefficients. The expansion is truncated at  $K = 868$  terms, which account for 92.5% of the spatial average of the variance of the Gaussian random field. The eigenpairs are

estimated with the Nyström method based on 100 Gauss-Legendre points in each direction, disregarding points inside the hole.

Additionally, we model the applied external load as a normally distributed random variable with mean  $\mu_q = 60$  MPa and standard deviation  $\sigma_q = 12$  MPa such that  $q \sim \mathcal{N}(\mu_q, \sigma_q)$ . Assuming further independence between the standard normal vector  $\boldsymbol{\theta}$  and  $q$ , the probabilistic input vector of this problem is gathered in  $\mathbf{X} = (\boldsymbol{\theta}, q)$ .

The quantity of interest is the *principal stress* at a *control point* marked by a red star in Figure 11 defined as

$$\sigma_* \stackrel{\text{def}}{=} (\sigma_{11} + \sigma_{22})/2 + \sqrt{((\sigma_{11} - \sigma_{22})/2)^2 + \tau_{12}^2}. \quad (30)$$

It depends on the stress field  $\boldsymbol{\sigma}(\boldsymbol{\xi})$  at the control point that is obtained from the elastic constitutive equations after solving Eq. (28) with the finite element method for the displacement field  $\mathbf{u}(\boldsymbol{\xi})$ . Failure in this problem is defined as this principle stress exceeding  $\sigma_{\text{th}} = 320$  MPa, which can be formalized in the following limit-state function:

$$g(\mathbf{X}) = \sigma_{\text{th}} - \sigma_*(\mathbf{X}). \quad (31)$$

Due to the complexity of this problem, it is not possible to obtain a reference solution with Monte Carlo simulation in a reasonable time. We use instead directly the reference solution from Uribe et al. (2020) that was computed with 100 independent subset simulation (Au and Beck, 2001) runs with  $N = 3,000$  samples per level and 6 levels, leading to a reference failure probability of  $P_f = 3.75 \cdot 10^{-6}$  ( $\beta = 4.48$ ).

On this problem, we perform 10 independent runs of SSER with different random seeds. The number of refinement samples was set to  $N_{\text{ref}} = 100$  and the maximum polynomial degree for the residual expansions to  $p_{\text{max}} = 4$ . To allow PCEs to be constructed efficiently in such high dimensions, it was necessary to restrict the maximum interactions of polynomial basis functions to 1 (Marelli et al., 2021). The domain-wise accuracy was estimated with  $B = 100$  bootstrap replications.

The convergence of SSER to the reference reliability index is shown in Figure 12. The median number of steps required to reach the stopping criterion is 6, corresponding to  $\underbrace{2N_{\text{ref}}}_{\text{initial}} + (5 \cdot 2)N_{\text{ref}} = 1,200$  limit-state evaluations.

## 4 Conclusions

Leveraging the flexibility of the recently proposed stochastic spectral embedding formalism (Marelli et al., 2021), we show that the adaptive sequential partitioning approach introduced in (Wagner et al., 2021) can be efficiently modified to an active learning reliability method. The introduced modifications pertain to the refinement domain selection, partitioning and sample enrichment strategies.

The SSER algorithm is shown to accurately estimate failure probabilities spanning multiple orders of magnitude at a fraction of the cost of plain Monte Carlo simulation. The

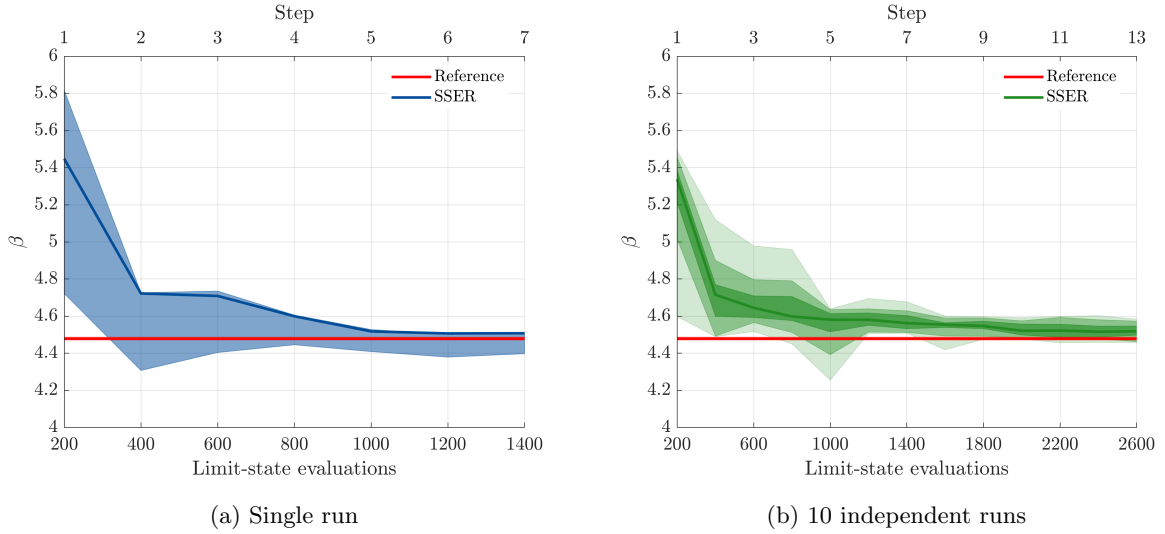


Figure 12: *Plate with a hole*: Convergence to the reference reliability index  $\beta$  as a function of limit-state evaluations and algorithm steps. The 95% confidence intervals in Figure (a) are based on the bootstrap replications. This figure shows the convergence until the stopping criterion in Eq. (23) is met at Step 7 of SSER. The bounds in Figure (b) are the 90%, 75% and 50% confidence intervals from the mean predictions of the independent SSER runs.

method is suitable for identifying multiple failure regions (*four-branch function*, Section 3.1) and is able to deal with problems that involve non-smooth limit-state functions (*piecewise linear function*, Section 3.2). It also performs reasonably well in moderate-dimensions with dependent input vectors (*five-story frame*, Section 3.3) and is competitive to a state-of-the-art reliability method iCRed (Uribe et al., 2020) on a high-dimensional example (*plate with a hole*, Section 3.4).

In general, the efficiency of SSER depends on two factors: (1) whether the algorithm identifies all failure modes and (2) whether the chosen spectral expansion technique can efficiently approximate  $g$  at the terminal domains that lie in proximity of the limit-state surface. Regarding the first factor, the refinement domain selection criterion in Eq. (15) should guarantee that SSER eventually identifies all modes of failure (e.g., *four-branch function* and *piecewise linear function*). However, in problems with a large number of disjoint limit-states, this sequential process might require a considerable number iterations and consequently model evaluations. Also, if the stopping criterion is triggered before all limit-states have been identified, the total failure probability might be underestimated. The second factor depends on  $g$ 's complexity at the terminal domains in proximity of the limit-state surface. Generally, spectral expansion techniques are known to converge slowly in highly non-linear and high-dimensional problems. Hence, we expect a decrease in efficiency of the algorithm in such problems. As an example, SSER is expected to perform poorly in first excursion problems that are both high-dimensional and highly non-linear in the limit state-function in

proximity of the limit-state surface.

In some examples (most notably in Section 3.4) there persists a small bias towards a higher reliability index compared to the reference solution. This means that the method systematically misses small parts of failure regions. A possible explanation for this is the partitioning strategy that ‘unknowingly’ cuts away undiscovered parts of failure regions and incorrectly classifies them as safe. This process is facilitated by the boundary discontinuities and will be explored in future works.

An interesting side result of the proposed method is the domain partition that hints at the location of failure domains. This information, in conjunction with the available conditional failure probabilities, could be directly used for design purposes.

The modularity of the sequential partitioning algorithm for constructing SSEs makes it well suited to for future modifications. Possible improvements could focus on different spectral expansion or even surrogate modelling techniques, as well as simultaneous partitioning in multiple dimensions. A particularly interesting idea is the introduction of rotated domain partitions by means of partial least squares Papaioannou et al. (2019). These partitions could be more flexibly tuned to fit the limit-state surface and thereby increase the efficiency of SSER.

## Acknowledgements

The PhD thesis of the first author is supported by ETH grant #44 17-1.

## References

- Au, S. K. and J. L. Beck (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics* 16(4), 263–277.
- Basler, E. (1960). *Untersuchungen über den Sicherheitsbegriff von Bauwerken*. PhD thesis, ETH Zürich.
- Bertsche, B. (2008). *Reliability in automotive and mechanical engineering*. Springer Berlin Heidelberg.
- Bichon, B. J., M. S. Eldred, L. Swiler, S. Mahadevan, and J. McFarland (2008, oct). Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal* 46(10), 2459–2468.
- Bjerager, P. (1988, August). Probability integration by directional simulation. *Journal of Engineering Mechanics* 114(8), 1285–1302.
- Blatman, G. and B. Sudret (2010, apr). An adaptive algorithm to build up sparse poly-

- nomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics* 25(2), 183–197.
- Blatman, G. and B. Sudret (2011, mar). Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics* 230(6), 2345–2367.
- Breitung, K. (1994). *Asymptotic approximations for probability integrals*. Berlin New York: Springer-Verlag.
- Breitung, K. (2019, February). The geometry of limit state function graphs and subset simulation: Counterexamples. *Reliability Engineering & System Safety* 182, 98–106.
- Cai, G. Q. and I. Elishakoff (1994, jul). Refined second-order reliability analysis. *Structural Safety* 14(4), 267–276.
- Ditlevsen, O. (1979, jan). Generalized second moment reliability index. *Journal of Structural Mechanics* 7(4), 435–451.
- Echard, B., N. Gayton, and M. Lemaire (2011, mar). AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Structural Safety* 33(2), 145–154.
- Echard, B., N. Gayton, M. Lemaire, and N. Relun (2013, March). A combined importance sampling and Kriging reliability method for small failure probabilities with time-demanding numerical models. *Reliability Engineering & System Safety* 111, 232–240.
- Efron, B. (1979). Bootstrap methods: Another look at the Jackknife. *Annals of Statistics* 7, 1–26.
- Fishman, G. (2011, May). *Monte Carlo: Concepts, algorithms, and applications*. Springer New York.
- Fries, R. C. (2017, March). *Reliable design of medical devices*. Taylor & Francis Ltd.
- Ghanem, R. and P. D. Spanos (1991). *Stochastic finite elements: A spectral approach*. Springer: Verlag.
- Haldar, A. (2006). *Recent developments in reliability-based civil engineering*. Singapore Hackensack, N.J: World Scientific.
- Harbitz, A. (1986, January). An efficient sampling method for probability of failure calculation. *Structural Safety* 3(2), 109–115.
- Hasofer, A.-M. and N.-C. Lind (1974). Exact and invariant second moment code format. *Journal of Engineering Mechanics* 100(1), 111–121.
- Hohenbichler, M., S. Gollwitzer, W. Kruse, and R. Rackwitz (1987). New light on first- and second order reliability methods. *Structural Safety* 4, 267–284.

- Hohenbichler, M. and R. Rackwitz (1988, December). Improvement of second-order reliability estimates by importance sampling. *Journal of Engineering Mechanics* 114(12), 2195–2199.
- Huang, X., J. Chen, and H. Zhu (2016, March). Assessing small failure probabilities by AK–SS: an active learning method combining Kriging and subset simulation. *Structural Safety* 59, 86–95.
- Joe, H. (2015). *Dependence modeling with copulas*. CRC Press.
- Koutsourelakis, P. S., H. J. Pradlwarter, and G. I. Schuëller (2004, oct). Reliability of structures in high dimensions, part I: Algorithms and applications. *Probabilistic Engineering Mechanics* 19(4), 409–417.
- Kurtz, N. and J. Song (2013, may). Cross-entropy-based adaptive importance sampling using Gaussian mixture. *Structural Safety* 42, 35–44.
- LaCombe, D. (1999). *Reliability control for electronic systems*. New York: Marcel Dekker, Inc.
- Lataniotis, C., E. Torre, S. Marelli, and B. Sudret (2021). UQLab user manual – The Input module. Technical report. Report # UQLab-V1.4-102.
- Liu, P.-L. and A. Der Kiureghian (1991, feb). Optimization algorithms for structural reliability. *Structural Safety* 9(3), 161–177.
- Lüthen, N., S. Marelli, and B. Sudret (2021, jan). Sparse polynomial chaos expansions: Literature survey and benchmark. *SIAM/ASA Journal on Uncertainty Quantification* 9(2), 593–649.
- Marelli, S., N. Lüthen, and B. Sudret (2021). UQLab user manual – Polynomial chaos expansions. Technical report. Report # UQLab-V1.4-104.
- Marelli, S., R. Schöbi, and B. Sudret (2021). UQLab user manual – Structural reliability (Rare event estimation). Technical report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland. Report # UQLab-V1.4-107.
- Marelli, S. and B. Sudret (2014). UQLab: A framework for uncertainty quantification in Matlab. In *Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, pp. 2554–2563.
- Marelli, S. and B. Sudret (2018). An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis. *Structural Safety* 75, 67–74.

- Marelli, S., P.-R. Wagner, C. Lataniotis, and B. Sudret (2021). Stochastic spectral embedding. *International Journal for Uncertainty Quantification* 11(2), 25–47.
- Melchers, R.-E. (1999). *Structural reliability analysis and prediction*. John Wiley & Sons.
- Montgomery, D. (2019). *Design and analysis of experiments*. Hoboken, NJ: Wiley.
- Moustapha, M., S. Marelli, and B. Sudret (2021). A generalized framework for active learning reliability: survey and benchmark. *Structural Safety*. Submitted.
- Nelsen, R. B. (2006). *An introduction to copulas*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Papadimitriou, I., M. Ehre, and D. Straub (2019, June). PLS-based adaptation for efficient PCE representation in high dimensions. *Journal of Computational Physics* 387, 186–204.
- Papadimitriou, I., S. Geyer, and D. Straub (2019, nov). Improved cross entropy-based importance sampling with a flexible mixture model. *Reliability Engineering & System Safety* 191, 106564.
- Papadimitriou, I., C. Papadimitriou, and D. Straub (2016, sep). Sequential importance sampling for structural reliability analysis. *Structural Safety* 62, 66–75.
- Papadimitriou, I. and D. Straub (2021, jan). Combination line sampling for structural reliability analysis. *Structural Safety* 88, 102025.
- Rackwitz, R. and B. Fiessler (1978, nov). Structural reliability under combined random load sequences. *Computers & Structures* 9(5), 489–494.
- Rosenblatt, M. (1952). Remarks on a multivariate transformation. *Annals of Mathematical Statistics* 23(3), 470–472.
- Rubinstein, R. Y. and D. P. Kroese (2016, November). *Simulation and the Monte Carlo method*. John Wiley & Sons, Inc.
- Schueremans, L. and D. V. Gemert (2005, July). Benefit of splines and neural networks in simulation based structural reliability analysis. *Structural Safety* 27(3), 246–261.
- Torre, E., S. Marelli, P. Embrechts, and B. Sudret (2019a, July). Data-driven polynomial chaos expansion for machine learning regression. *Journal of Computational Physics* 388, 601–623.
- Torre, E., S. Marelli, P. Embrechts, and B. Sudret (2019b, jan). A general framework for data-driven uncertainty quantification under complex input dependencies using vine copulas. *Probabilistic Engineering Mechanics* 55, 1–16.



- Tvedt, L. (1990). Distribution of quadratic forms in normal space – Applications to structural reliability. *Journal of Engineering Mechanics* 116(6), 1183–1197.
- Uribe, F., I. Papaioannou, Y. M. Marzouk, and D. Straub (2020, June). Cross-entropy-based importance sampling with failure-informed dimension reduction for rare event simulation. *arXiv e-prints*.
- Valdebenito, M. A., H. J. Pradlwarter, and G. I. Schuëller (2010, March). The role of the design point for calculating failure probabilities in view of dimensionality and structural nonlinearities. *Structural Safety* 32(2), 101–111.
- Wagner, P.-R., S. Marelli, and B. Sudret (2021, July). Bayesian model inversion using stochastic spectral embedding. *Journal of Computational Physics* 436, 110141.
- Wei, D. and S. Rahman (2007, jan). Structural reliability analysis by univariate decomposition and numerical integration. *Probabilistic Engineering Mechanics* 22(1), 27–38.
- Xiu, D. and G. E. Karniadakis (2002, jan). The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing* 24(2), 619–644.
- Zhang, Y. and A. Der Kiureghian (1995). Two improved algorithms for reliability analysis. In R. Rackwitz, G. Augusti, and A. Bori (Eds.), *Proc. 6th IFIP WG7.5 on Reliability and Optimization of Structural systems, Assisi, Italy*, pp. 297–304. Chapman & Hall, London.
- Zuev, K. M., J. L. Beck, S.-K. Au, and L. S. Katafygiotis (2012, February). Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. *Computers & Structures* 92-93, 283–296.