Estimating failure probabilities using an adaptive variant of stochastic spectral embedding

Paul-Remo Wagner, Iason Papaioannou, Stefano Marelli, Daniel Straub, Bruno Sudret

ETH zürich

Estimating failure probabilities using an adaptive variant of stochastic spectral embedding

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

Chair of Risk, Safety and Uncertainty Quantification
Stefano-Franscini-Platz 5
CH-8093 Zürich

Risk, Safety & Uncertainty Quantification

Data Sheet

Journal:	Proc. 13th International Conference on Structural Safety & Reliability, Tongji University, Shanghai (China), September 13-17
Report Ref.:	RSUQ-2022-010
Arxiv Ref.:	
DOI:	-
Date submitted:	December 16, 2021
Date accepted:	September 13, 2022

Estimating failure probabilities using an adaptive variant of stochastic spectral embedding

P.-R. Wagner [1], I. Papaioannou [2], S. Marelli [1], D. Straub [2], and B. Sudret [1]

[1] *Chair of Risk, Safety and Uncertainty Quantification, Stefano-Franscini Platz 5, 8049 Zurich, Switzerland,*
E-mail: {wagner, marelli, sudret}@ibk.baug.ethz.ch
[2] *Engineering Risk Analysis Group, Theresienstr. 90, 80333 Munich, Germany,*
E-mail: {iason.papaioannou, straub}@tum.de

September 26, 2022

Abstract

Reliability analysis aims to assess the probability of structural failure. The main difficulties in computing this quantity lie in its inherently low value, which causes most simulation methods to require a large number of expensive model evaluations. To alleviate the associated computational burden, practitioners today increasingly resort to active learning methods to train a surrogate model that is then used in lieu of the original model for computing the failure probability. In this contribution, we apply an adaptive variant of the recently proposed stochastic spectral embedding (SSE) surrogate modelling technique to solve reliability analysis problems. SSE creates a sequence of polynomial chaos expansions by splitting and refining subdomains of the input space. We propose here modified refinement and splitting criteria that can generate an efficient surrogate model with increased accuracy near the limit state surface. The performance of the algorithm is showcased on two reliability problems from the literature.

1 Introduction

The assessment of structural reliability is one of the central tasks of civil engineers. Assessment of structural performance involves the simulation of structures by means of advanced computational methods. Independent of the simulation complexity, simulation models take M input parameters $\mathbf{X} \in \mathcal{D}_{\mathbf{X}} \subseteq \mathbb{R}^M$ and produce a set of quantities of interest (e.g. maximum stress, deformations, temperatures). The latter are subsequently used to construct a *limit-state function* g that assumes positive values if the structural requirements are met, and non-positive values otherwise.

By treating the input parameters as a random vector, with an associated joint probability density function $f_{\mathbf{X}}$, it is possible to use the limit-state function to define the *probability of failure* as

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

where $\mathbf{1}_{\mathcal{D}_f}$ is the indicator function that assumes the value 1 inside the failure domain $\mathcal{D}_f$ and 0 in the safe domain:

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

As structural systems are designed to be safe, this failure probability is typically quite small, while the computer simulations required to evaluate g can be extremely expensive. This poses challenges for computational methods developed to estimate the integral in Equation 1.

A powerful strategy to estimate the failure probability is provided by so-called *active learning reliability methods* (e.g. AK-MCS in **(author?)** [1] or bootstrap PCE in **(author?)** [2]). These methods adaptively refine approximations of the limit-state function to accurately distinguish between failure and safe domains.

In this contribution we present a novel active learning reliability method based on the recently proposed *stochastic spectral embedding* (SSE, **(author?)** [3, 4]) surrogate modelling technique. This method constructs an approximation of the target function by means of local residual expansions in subdomains of the input space. By modifying its training algorithm, this technique can be efficiently applied to approximating limit-state functions of reliability problems. By smartly choosing the partitions, the total failure probability is thereby split into a set of easier-to-estimate failure probabilities that can be computed efficiently with standard reliability algorithms.

2 Stochastic spectral embedding

Spectral expansion techniques are powerful tools for functional representation. Arguably the most widely used spectral technique is polynomial chaos expansions (PCE, [5, 6]). Assuming that the limit-state function has finite variance (i.e. $\mathbb{E}\left[g(X) < +\infty\right]$), this approach expresses g in an orthogonal polynomial basis $\{\Psi_{\boldsymbol{\alpha}}\}_{\alpha \in \mathbb{N}^M}$ such that

$$g(\mathbf{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\mathbf{X}) + \mathcal{R}(\mathbf{X}), \tag{3}$$

where $\mathcal{A} \subset \mathbb{N}^M$ is a multi-index set used to denote the polynomial degree, $a_{\boldsymbol{\alpha}}$ denotes the polynomial coefficients and $\mathcal{R}$ is the residual of the expansion.

Given an experimental design $\mathcal{X} \overset{\text{def}}{=} \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}\}$ and corresponding evaluations of the limit state function $\mathcal{Y} \overset{\text{def}}{=} g(\mathcal{X})$, there exist numerous algorithms to efficiently compute the polynomial coefficients as well as $\mathcal{A}$ [6]. The idea of SSE is to extract additional information from the residual $\mathcal{R}$ by expanding it onto additional partitions of the input space (Fig. 1).

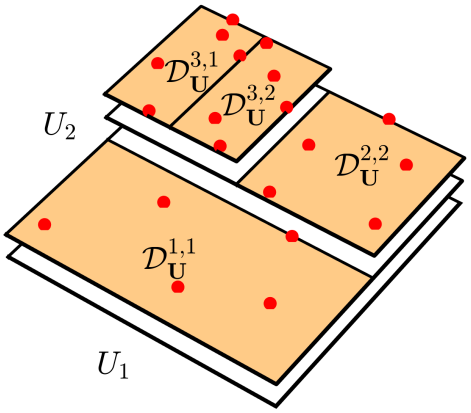

Figure 1: SSE partitions in the quantile space $\mathcal{D}_{\mathbf{U}}$. Red dots denote the experimental design $\mathcal{X}$ and orange domains are the terminal domains $\mathcal{T} \subset \mathcal{K}$.

After sequentially expanding the residuals in subdomains of decreasing size indexed by elements of $\mathcal{K}$, the SSE representation of the limit-state function can be more compactly written as

$$g(\mathbf{X}) \approx \sum_{k \in \mathcal{K}} \mathcal{R}_{\mathrm{S}}^{k}(\mathbf{X}), \tag{4}$$

where the residual expansion is given by $\mathcal{R}_{\mathrm{S}}^{k}(\mathbf{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}^k} a_{\boldsymbol{\alpha}}^{k} \Psi_{\boldsymbol{\alpha}}^{k}(\mathbf{X})$. **(author?)** [3] also derived a measure of domain-wise accuracy for the SSE representation by means of the SSE *generalization error*. In the k-th terminal domain it is defined by

$$E_{\mathrm{GEN}}^{k} \stackrel{\mathrm{def}}{=} \mathbb{E}\left[\left(\mathcal{R}_{\mathrm{S}}^{k}(\mathbf{X}^k) - \mathcal{R}^{k}(\mathbf{X}^k)\right)^2\right], \tag{5}$$

where $\mathbf{X}^k$ denotes the input vector conditional on the k-th subdomain.

Using the available PCE leave-one-out cross validation error E_{LOO}^{k}, the generalization error can be estimated as

$$E_{\mathrm{LOO}}^{k} \approx E_{\mathrm{GEN}}^{k}. \tag{6}$$

The SSE representation in Equation 4 allows a decomposition of the failure probability integral in Equation 1 into card($\mathcal{T}$) independent reliability problems such that

$$P_f = \sum_{k \in \mathcal{T}} \mathcal{V}^k P_f^k, \text{ with } \mathcal{V}^k = \int_{\mathcal{D}_{\mathbf{X}}^{k}} f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{7}$$

These *local* failure probabilities P_f^k can be efficiently estimated with any standard reliability method (Monte Carlo simulation, subset simulation [7], importance sampling [8] etc.) since (1) by construction P_f^k is either close to 1 or negligibly small and (2) the local limit-state function is a cheap-to-evaluate polynomial. Furthermore, the SSE error measure allows the estimation of bounds on the local failure probability. To this end, we assume a subdomain-wise normally

distributed proxy to the approximation error based on the SSE error measure. With this, we can estimate the bounds on the local failure probability as

$$\overline{P_f^k} \approx \mathbb{P}\left[g_{SSE}(\mathbf{X}^k) - 2\sigma^k \leq 0\right], \tag{8}$$

$$\underline{P_f^k} \approx \mathbb{P}\left[g_{SSE}(\mathbf{X}^k) + 2\sigma^k \leq 0\right], \tag{9}$$

where $\sigma^k \overset{\text{def}}{=} \sqrt{E^k_{\text{LOO}}}$ with the leave-one-out error defined in Equation 6.

By combining these bounds with Equation 7, bounds on the total failure probability can be derived as well. For the upper bound this amounts to

$$\overline{P}_f = \sum_{k \in \mathcal{T}} \mathcal{V}^k \overline{P}_f^k. \tag{10}$$

The bounds to the total failure probability derived this way are rather conservative because they assume the worst/best case for all terminal domains at the same time.

3 Active learning modifications

The adaptive sequential partitioning SSE algorithm, performs a sequence of refinement steps that consist of (1) selecting a refinement domain, (2) partitioning this domain, (3) enriching the experimental design and (4) expanding the residual in the created partitions. These steps are described in detail in [4] in the context of Bayesian model calibration.

The modular nature of SSE makes it possible to adapt the algorithm to various function approximation problems. In this contribution we modify steps (1) and (2), to derive a tailored SSE algorithm for reliability problems. These modifications are explained in this section with the proposed algorithm given in Section 3.3.

3.1 Refinement domain selection

At every refinement step, the algorithm picks a *refinement domain* from the terminal domains, i.e. unsplit domains (Fig. 1), and focuses only on this domain for the remainder of this step.

The refinement domain should be chosen in order to reduce the uncertainty on the total failure probability estimator. Using the bounds on the local failure probability from Equations 8 and 9 and the decomposition of the total failure probability in Equation 7, the domain with the largest contribution to the failure probability bounds is

$$k_{\text{refine}} = \arg\max_{k \in \mathcal{T}} \mathcal{V}^k \left(\overline{P_f^k} - \underline{P_f^k}\right). \tag{11}$$

3.2 Partitioning strategy

The partitioning strategy determines how the refinement domain is partitioned prior to constructing the residual expansions. It can be used to focus on regions of interest in the input space. The regions of interest in active learning reliability problems are those of high *misclassification probability*, i.e. regions that are possibly not classified correctly as failed or safe. These are typically located in the tails of the input distribution.

The default *equal-splitting* strategy of the original algorithm, which splits the space along a certain Cartesian coordinate into equal probability regions, would require a large number of steps until reaching those regions of interest. Therefore, the proposed partitioning strategy introduces splits with the intent to isolate regions of high misclassification probability from regions with a low misclassification probability. To this end, we define a *misclassification index*, using the standard deviation of the SSE predictor as

$$I_m(\mathbf{x}) = 1 - \mathbf{1}_{\overline{\mathcal{D}_f}}(\mathbf{x}) \cdot \mathbf{1}_{\underline{\mathcal{D}_f}}(\mathbf{x}), \tag{12}$$

where

$$\mathbf{1}_{\overline{\mathcal{D}_f}}(\mathbf{x}) = \begin{cases} 1, & \text{if} \quad g_{\text{SSE}}(\mathbf{x}) - 2\sigma^k \leq 0, \\ 0, & \text{if} \quad g_{\text{SSE}}(\mathbf{x}) - 2\sigma^k > 0, \end{cases} \tag{13}$$

$$\mathbf{1}_{\underline{\mathcal{D}_f}}(\mathbf{x}) = \begin{cases} 1, & \text{if} \quad g_{\text{SSE}}(\mathbf{x}) + 2\sigma^k \leq 0, \\ 0, & \text{if} \quad g_{\text{SSE}}(\mathbf{x}) + 2\sigma^k > 0. \end{cases} \tag{14}$$

The misclassification index in a given point can be used to define two auxiliary random vectors as

$$\mathbf{Z}^0 \overset{\text{def}}{=} \mathbf{X} | I_m = 0 \tag{15}$$

$$\mathbf{Z}^1 \overset{\text{def}}{=} \mathbf{X} | I_m = 1, \tag{16}$$

These conditional random vectors have disjoint supports and distinguish regions with $I_m = 0$ and $I_m = 1$ respectively (Fig. 2).

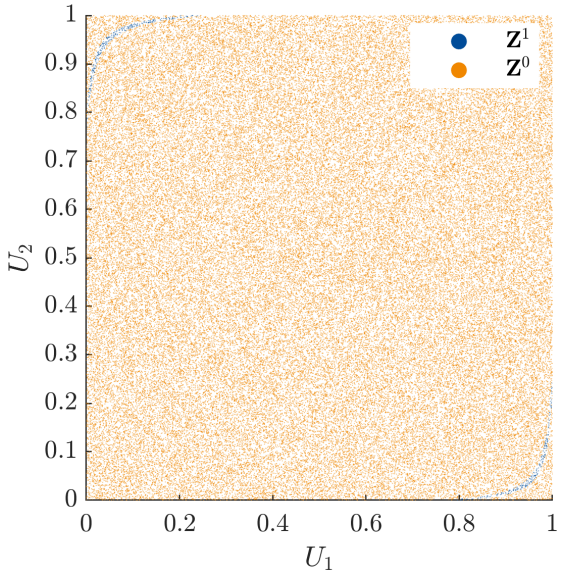

Figure 2: Sample of auxiliary random vectors $\mathbf{Z}^0$ and $\mathbf{Z}^1$ used in the partitioning strategy. The support of $\mathbf{Z}^0$ is the region of $I_m = 0$, while the support of $\mathbf{Z}^1$ is the region of $I_m = 1$. The variables are shown in the quantile space $\mathbf{U}$.

We use these auxiliary random vectors to define the location of the split. In each dimension $\xi_i \in \mathcal{D}_{X_i}, i = 1, \cdots, M$, we choose a *splitting location* that confines a maximum of $\mathbf{Z}^1$'s probability mass to one side, and $\mathbf{Z}^0$'s probability mass to the other side of the split. More formally, we pick the splitting location in dimension i as

$$\widehat{\xi}_i = \arg \max_{\xi_i \in \mathcal{D}_{X_i}} L_i(\xi_i), \tag{17}$$

with

$$L_i(\xi_i) \stackrel{\text{def}}{=} \max \begin{cases} \mathbb{P}\left[Z_i^1 \leq \xi_i\right] + \mathbb{P}\left[Z_i^0 > \xi_i\right], \\ \mathbb{P}\left[Z_i^1 > \xi_i\right] + \mathbb{P}\left[Z_i^0 \leq \xi_i\right], \end{cases} \tag{18}$$

where Z_i^0 and Z_i^1 are the marginals of the auxiliary random vectors in the i-th dimension. The objective function L_i characterizes the split properties by returning the maximum of the respective auxiliary probability masses in the subdomains resulting from the split. For illustrative purposes, this process is shown in Figure 3

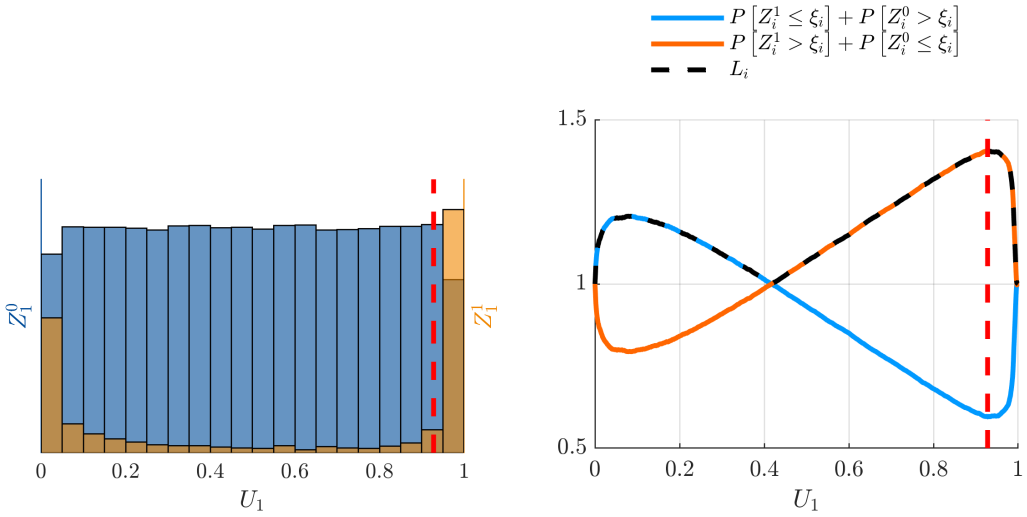

Figure 3: The function L_i defined in Eq. 18 for the example in Figure 2 and its application to the marginals of Z_i^0 and Z_i^1. The upper plot shows histograms of the marginals (in different scales). The lower plot displays L_i and the two terms it is composed of. The splitting location $\widehat{\xi}_i$ defined in Eq. 17 is shown as a vertical dashed line.

To ultimately choose a *splitting direction* $d \in \{1, \cdots, M\}$, we compare the values of the objective functions L_i and split along the dimension

$$d = \arg \max_{i \in \{1, \cdots, M\}} L_i(\widehat{\xi}_i). \tag{19}$$

The computations presented in this section are conducted with a sufficiently large sample of the auxiliary random vectors $\mathbf{Z}^0$ and $\mathbf{Z}^1$. This sample can be generated at a low computational cost, because it only requires the evaluation of the SSE model.

3.3 Proposed algorithm

We sketch here the proposed algorithm. It is based on the adaptive algorithm from **(author?)** [4].

1. **Initialization:**

   Algorithm parameters:
   - N_{ref}: size of refinement sample
   - N_{ED}: size of computational budget

   (a) $\mathcal{D}_{\mathbf{X}}^{0,1} = \mathcal{D}_{\mathbf{X}}$

   (b) Sample from input distribution $\mathcal{X} = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(2N_{\text{ref}})}\}$

   (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. 11**
   (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. 17 and 19 and update $\mathcal{T}$
   (b) **For each split** $s = \{s_1, s_2\}$
      i. Enrich sample $\mathcal{X}$ with $\min\{N_{\text{ref}}, N_{ED} - |\mathcal{X}|\}$ new points inside $\mathcal{D}_{\mathbf{X}}^{\ell+1,s}$
      ii. **If** N_{ref} points inside $\mathcal{D}_{\mathbf{X}}^{\ell+1,s}$
         A. Create the truncated expansion $\widehat{\mathcal{R}}_S^{\ell+1,s}(\mathbf{X}^{\ell+1,s})$ of the residual $\mathcal{R}^{\ell+1}(\mathbf{X}^{\ell+1,s})$ in the current subdomain using $\mathcal{X}^{\ell+1,s}$
         B. Update the residual in the current subdomain $\mathcal{R}^{\ell+2}(\mathbf{X}^{\ell+1,s}) = \mathcal{R}^{\ell+1}(\mathbf{X}^{\ell+1,s}) - \widehat{\mathcal{R}}_S^{\ell+1,s}(\mathbf{X}^{\ell+1,s})$
   (c) Retrieve P_f from Eq. 7 and its bounds from Eq. 10
   (d) **If** convergence criterion is met or less than two new expansions were created because the computational budget N_{ED} has been exhausted, terminate the algorithm, otherwise go back to 2

3. **Termination**
   (a) Return P_f and its bounds

3.4 Convergence criterion

In principle, any convergence criterion from the active learning literature can be applied to terminate the algorithm [9]. We opt for the the *beta bounds* criterion that stops refinement when

$$\frac{\left|\overline{\beta} - \underline{\beta}\right|}{\beta} \leq \varepsilon_{\text{BB}} \tag{20}$$

where β, $\overline{\beta}$ and $\underline{\beta}$ are the *reliability index* and its approximate bounds computed as $\beta = -\Phi^{-1}(P_f)$, $\overline{\beta} = -\Phi^{-1}(\underline{P_f})$ and $\underline{\beta} = -\Phi^{-1}(\overline{P_f})$ respectively. As the bounds on β overestimate the actual bounds (see Eq. 10), the convergence threshold is chosen to a relatively large value of $\varepsilon_{\text{BB}} = 3\%$. The algorithm is stopped if this criterion is met three times in a row to ensure a degree of robustness to premature convergence.

4 Application

To demonstrate the performance of our algorithm, we test it on two problems: the well-known *four-branch function* [1, 10] and a truss structure.

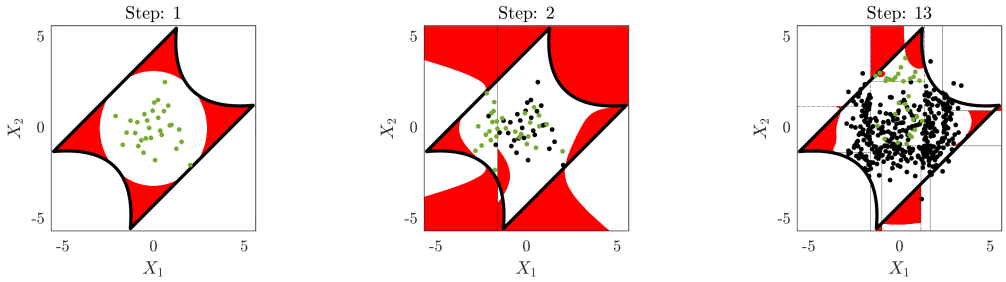

Figure 4: *Four-branch function:* Selected steps from the proposed algorithm. The limit state surface (i.e. $g(\mathbf{X}) = 0$) is shown as a thick black line. Misclassified regions are shown in red. Black points are the experimental design with the last latest additions to the experimental design highlighted in green. The domain partitions are shown as dashed lines.

4.1 Four-branch function

This 2D function simulates a series system with four failure regions. The limit state function is defined analytically by

$$g(\mathbf{X}) = \min \begin{pmatrix} 3 + 0.1(X_1 - X_2)^2 - \frac{X_1+X_2}{\sqrt{2}}, \\ 3 + 0.1(X_1 - X_2)^2 + \frac{X_1+X_2}{\sqrt{2}}, \\ (X_1 - X_2) + \frac{6}{\sqrt{2}}, \\ (X_2 - X_1) + \frac{6}{\sqrt{2}} \end{pmatrix}. \tag{21}$$

The input parameter is modelled as a bivariate independent standard Gaussian distribution, i.e. $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ which puts the failure probability from reference Monte Carlo simulations at $P_f = 4.45 \cdot 10^{-3}$ corresponding to a reliability index of $\beta = 2.61$.

We run the proposed active learning algorithm with $N_{\text{ref}} = 15$. The residual PCE expansions are constructed with the PCE module of UQLab [11] and using an adaptively chosen maximum polynomial degree of $p_{\max} = 7$.

A few selected steps of the algorithm are shown in Figure 4. The initial global expansion in Step 1 overestimates the failure probability by not accounting for the corners of the safe region. In Step 2, the algorithm partitions the domain along $d = 1$, enriches the experimental design with $N_{\text{ref}} = 15$ new points and constructs two residual expansions. This procedure is repeated until Step 13, where the convergence criterion is met. The corresponding convergence of the reliability index is shown in Figure 5

To show the robustness of our algorithm, we produce 50 replications of the analysis and show the convergence of β in Figure 6.

4.2 Truss structure

The second considered example is the static model of a truss structure as presented originally in **(author?)** [12] (Fig. 7). The truss structure is subject to six point loads $P_i, i = 1, \cdots, 6$.

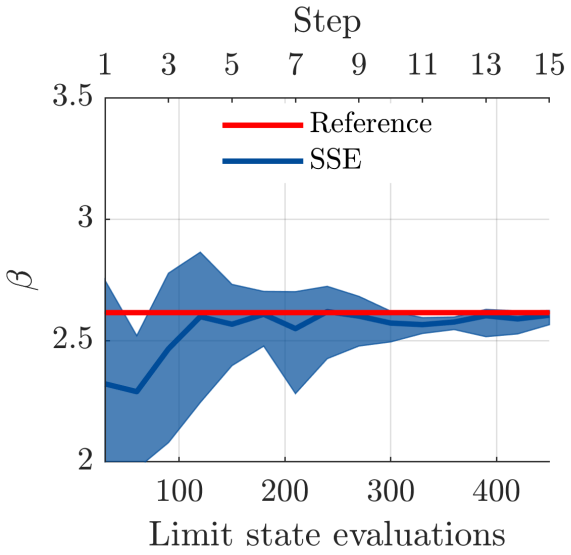

Figure 5: *Four-branch function:* Convergence of the reliability index β for a single run of the proposed algorithm. The reference value of $\beta = 2.61$ was computed with Monte Carlo simulation on the original function.

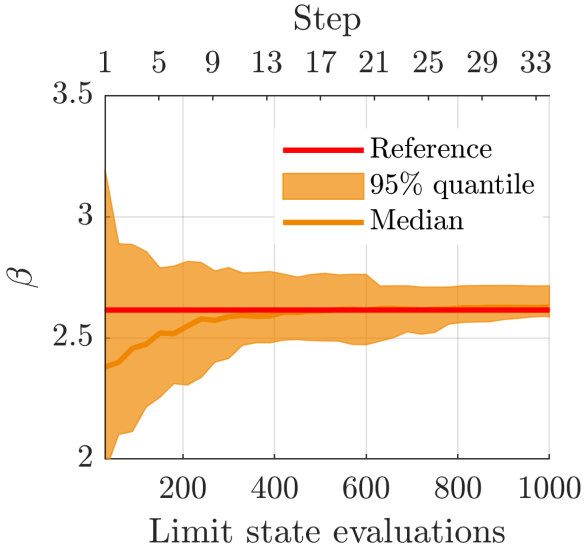

Figure 6: *Four-branch function:* Convergence of the reliability index β for 50 runs of the proposed algorithm. The reference value of $\beta = 2.61$ was computed with Monte Carlo simulation on the original function.

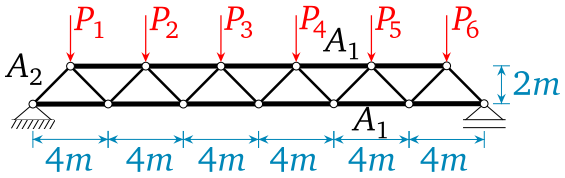

Figure 7: Sketch of the considered truss structure subject to 6 point loads.

Table 1: *Truss structure:* Input parameter marginals. The mean is denoted by μ and the standard deviation by σ.

Name	**Distribution**	μ	σ	unit
E_i	Lognormal	$2.1 \cdot 10^{11}$	$2.1 \cdot 10^{10}$	$[Pa]$
A_1	Lognormal	$2 \cdot 10^{-3}$	$2 \cdot 10^{-4}$	$[m^2]$
A_2	Lognormal	10^{-3}	10^{-4}	$[m^2]$
P_i	Gumbel	$5 \cdot 10^4$	$7.5 \cdot 10^3$	$[N]$

The bars of the chord and web are assumed to be made from materials with Young's moduli E_1 and E_2 and have cross section areas A_1 and A_2 respectively. Those quantities are unknown and modelled as independent random variables with marginals given in Table 1.

Gathering the uncertain parameters in a vector $\mathbf{X} = \{E_1, E_2, A_1, A_2, P_1, \cdots, P_6\}$, failure of the structure is assumed when the mid-span deflection w exceeds 0.13 $[m]$ leading to a limit-state function

$$g(\mathbf{X}) = 0.13 - u(\mathbf{X}). \tag{22}$$

We compute the deflection $u(\mathbf{X})$ with a simple finite element code. With this, the failure probability obtained from a reference Monte-Carlo simulation is $P_f = 2.36 \cdot 10^{-4}$, corresponding to a reliability index of $\beta = 3.5$.

We solve this problem with our proposed algorithm, setting $N_{\text{ref}} = 20$ and using local residual expansions with a maximum polynomial degree of $p_{\max} = 12$. The convergence for a single run is shown in Figure 8. The bounds on β cover the reference failure probability, but the mean prediction overestimates β by approximately 1%.

We again produce 50 replications of our analysis and show the convergence of β in Figure 9. It can be clearly seen that the algorithm slightly underestimates the failure probability, corresponding to a median overestimation of β by approximately 3%.

5 Conclusions

The paper presents an active learning SSE algorithm for structural reliability analysis. The algorithm is tested with two examples, which demonstrate good performance. However, we still

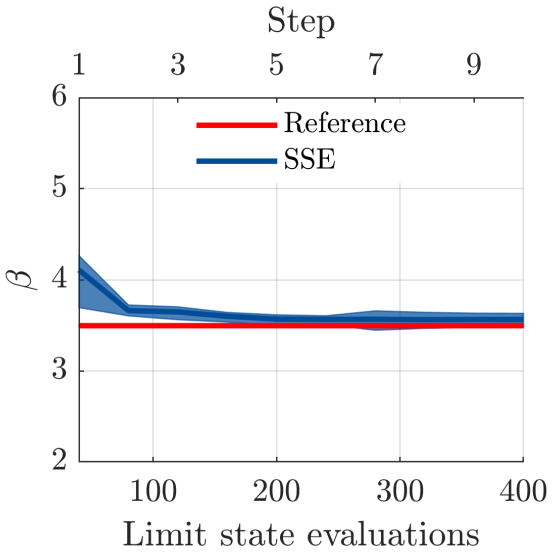

Figure 8: *Truss structure:* Convergence of the reliability index β for a single run of the proposed algorithm. The reference value of $\beta = 3.5$ was computed with Monte Carlo simulation on the original model.

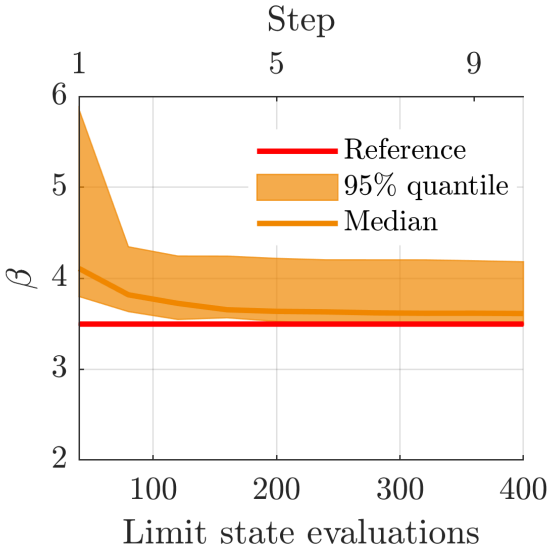

Figure 9: *Truss structure:* Convergence of the reliability index β for 50 runs of the proposed algorithm. The reference value of $\beta = 3.5$ was computed with Monte Carlo simulation on the original model.

see significant room for improvement. The three main points that we will investigate in the future pertain to: (1) underestimating failure probabilities, (2) estimating local error bounds and (3) sample enrichment. They will be considered in future contributions:

(1) The first issue can be attributed to the algorithm estimating $P_f^k = 0$ in large domains that actually have a non-zero failure probability. This is a direct result of the refinement domain selection criterion (Sec. 3.1) that will never refine large domains that (wrongly) predict $\overline{P_f^k} = \underline{P_f^k} = 0$.

(2) The local error bounds estimation is crude and, depending on the limit-state surface, can underestimate the true bounds. This is especially problematic for problems that can be approximated well globally, but show a different behaviour near the limit state surface. A more localized error metric could solve this problem.

(3) The sample enrichment strategy promises considerable room for improvement. The current approach of naively sampling uniformly in individual subdomains could be improved by sampling according to the misclassification probability.

References

[1] B. Echard, N. Gayton, and M. Lemaire. AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Structural Safety*, 33(2):145–154, 2011.

[2] S. Marelli and B. Sudret. An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis. *Structural Safety*, 75:67–74, 2018.

[3] S. Marelli, P.-R. Wagner, C. Lataniotis, and B. Sudret. Stochastic spectral embedding. *International Journal for Uncertainty Quantification*, 2021. In Press.

[4] P.-R. Wagner, S. Marelli, and B. Sudret. Bayesian model calibration with stochastic spectral embedding. *Journal of Computational Physics*, 2021. In Press.

[5] D. Xiu and G. E. Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[6] G. Blatman and B. Sudret. Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *Journal of Computational Physics*, 230:2345–2367, 2011.

[7] S K Au and J L Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, 2001.

[8] R.E. Melchers. Importance sampling in structural systems. *Structural Safety*, 6(1):3–10, jul 1989.

[9] M. Moustapha, S. Marelli, and B. Sudret. A generalized framework for active learning reliability: survey and benchmark. *Reliability Engineering & Systems Safety*, 2021. Submitted.

[10] Luc Schueremans and Dionys Van Gemert. Benefit of splines and neural networks in simulation based structural reliability analysis. *Structural Safety*, 27(3):246–261, jul 2005.

[11] S. Marelli and B. Sudret. UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich, 2019. Report # UQLab-V1.3-104.

[12] Sang Hoon Lee and Byung Man Kwak. Response surface augmented moment method for efficient reliability analysis. *Structural Safety*, 28(3):261–272, jul 2006.