Bootstrap-Polynomial Chaos Expansions and Adaptive Designs for Reliability Analysis

S. Marelli1, B. Sudret1

1Dr., ETH Zurich, Chair of Risk, Safety and Uncertainty Quantification, Stefano-Franscini-Platz 5, CH-8093 Zurich, Switzerland. (marelli@ibk.baug.ethz.ch)
2Prof., ETH Zurich, Chair of Risk, Safety and Uncertainty Quantification, Stefano-Franscini-Platz 5, CH-8093 Zurich, Switzerland. (sudret@ibk.baug.ethz.ch)

Abstract: In this contribution we propose to use polynomial chaos expansions (PCEs) as surrogate models for structural reliability. This approach is not so popular in the literature for this purpose, since PCEs are known to provide a good global approximation of the true model (which makes them very suitable for moment- and global sensitivity analysis) while being less accurate at representing the tails.

We first introduce the use of bootstrap as a means to evaluate a statistical error on the coefficients of the PCE of the limit state function and, subsequently, on the quantities of interest. This allows one to derive local error estimators similar to the Kriging variance. We investigate several experimental design enrichment criteria based on the misclassification of points (in safe/failure domain) for the various replicates. The proposed algorithm is applied to various benchmark problems in structural reliability.

Keywords: polynomial chaos expansions, bootstrap, adaptive experimental designs, structural reliability.

1. Introduction

Structural reliability analysis aims at computing the probability of failure of a system with respect to some performance criterion when its governing parameters are uncertain. To account for uncertainties, these parameters are modelled by a random vector $\mathbf{X}$ with prescribed joint probability density function $f_{\mathbf{X}}$. The limit-state function $g$ is defined over the support of $\mathbf{X}$ such that $\{ \mathbf{x} : g(\mathbf{x}) \leq 0 \}$ defines the failure domain and $\{ \mathbf{x} : g(\mathbf{x}) > 0 \}$ defines the safe domain. The probability of failure is then obtained by:

$$P_f = \int_{\{ \mathbf{x} : g(\mathbf{x}) \leq 0 \}} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}$$

Standard Monte Carlo simulation can usually not be used because of the huge number of samples required to compute small probabilities (e.g. $10^{4-6}$ for estimating a probability of order $10^{-5}$). Approximation methods such as FORM/SORM are more efficient, yet they are based on the linearization of the limit-state function. Thus they are not applicable to complex cases.

In contrast, methods based on surrogate modelling have gained interest in the last few years. Most approaches are based on Kriging surrogate models as in the EGRA method by Bichon et al. (2008, 2011), AK-MCS by Echard et al. (2011). Dubourg et al. (2013, 2014) use Kriging to devise a quasi-optimal importance sampling density. Schöbi et al. (2015) revisit the different approaches using polynomial-chaos-based Kriging as a surrogate model.

Polynomial chaos expansions (PCE) are well-known in the field of uncertainty propagation (Ghanem and Spanos (2003)) and global sensitivity analysis (Sudret, 2008), yet less applied in the context of structural reliability (Sudret & Der Kiureghian (2002)). Indeed, PCEs provide a globally accurate surrogate model (in the mean square-sense), but are not necessarily accurate in representing the tails of the output distribution, which is a key asset for reliability analysis.

One explanation is that the abovementioned Kriging-based approaches all rely upon a local error estimator (the Kriging variance) to derive their active learning algorithms. The latter are defined as the procedures for enriching the design of computer experiments from which the surrogate of the limit state function is iteratively built. No such local error estimate naturally exist for PCEs.

Based on this preliminary remark, the goal of this paper is to derive local error estimates for polynomial chaos expansions using the bootstrap approach.
technique. Once such a feature exists, one can get a large number \( B \) of surrogate models from a single experimental design, which allows in turn to estimate the quality of classification of the surrogated limit state surface \( \{ \hat{g}(\mathbf{x}) = 0 \} \). In turn, an iterative enrichment algorithm may be devised to improve simultaneously the quality of the PCE expansion in regions close to the limit state surface, and to obtain convergent estimators of the probability of failure.

2. Methodology
2.1 Polynomial chaos expansions

Polynomial chaos expansions (PCE) are a meta-modelling technique based on the spectral decomposition of a finite-variance model \( Y = M(\mathbf{X}) \) onto a set of orthogonal polynomials given by:

\[
Y = \sum_{\alpha \in \mathcal{N}^M} y_{\alpha} \Psi_{\alpha}(\mathbf{X}) \tag{1}
\]

In practice the basis polynomials are tensor products of univariate polynomials that are respectively orthogonal to the input probability density function (PDF) of each component of \( X_i \).

\[
\Psi_{\alpha}(\mathbf{X}) = \prod \varphi_{\alpha_i}(X_i)
\]

Among other techniques, the coefficients \( y_{\alpha} \) of a truncated series (coefficients in a subset \( \alpha \in A \)) can be computed by least-square analysis (Sudret, 2015). Based on an experimental design, i.e. samples drawn according to the input distribution \( f_x \), say \( \mathbf{X} = \{ \mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)} \} \), the solution reads:

\[
y_{\alpha} = \arg\min_{\alpha} \frac{1}{N} \sum_{i=1}^{N} \left[ M(\mathbf{x}^{(i)}) - \sum_{\alpha \in A} y_{\alpha} \Psi_{\alpha}(\mathbf{x}^{(i)}) \right]^2
\]

Sparse expansions may be derived to tackle high-dimensional problems using least-angle regression (Blatman and Sudret, 2011).

2.2 Local error estimation in PCE

2.2.1 Bootstrap

An important tool in assessing the variability of an estimator based on the sampling of some random vector \( \mathbf{X} \) is given by the bootstrap resampling method (Efron, 1992). Suppose that some quantity \( \theta \) is a function of a sample set \( \mathbf{X} = \{ \mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)} \} \) drawn from some random vector \( \mathbf{X} \), or, in other words, \( \theta = f(\mathbf{X}) \). The bootstrap method allows one to estimate the statistics of \( \theta \) by drawing \( B \) new samples from the original \( \mathbf{X} \) using the technique of resampling with substitution. This technique consists in constructing \( B \) new sample sets \( \{ \mathbf{x}^{(1)}, ..., \mathbf{x}^{(B)} \} \) from the original one, by randomly assembling \( N \) realizations \( \mathbf{x}^{(i)} \in \mathbf{X} \), possibly including repeatedly the same realization several times within each sample. The quantity \( \theta \) can be then estimated on each of the created \( B \) samples, effectively creating a sample of the estimator \( \mathbf{B} = \{ \theta^{(1)} = f(\mathbf{x}^{(1)}), ..., \theta^{(B)} = f(\mathbf{x}^{(B)}) \} \). This sample can be used directly to evaluate the statistics of the random variable \( \theta \) without the need of extracting additional samples other than the original \( \mathbf{X} \).

2.2.2 PC-Bootstrap

In this paper we propose to use the bootstrap technique to provide local error estimates to PCE predictions. The PCE coefficients \( y_{\alpha} \) in Eq. (1) are estimated based on an experimental design of finite size \( N \), which is a sample from the input random vector \( \mathbf{X} \). It is therefore possible to directly apply the bootstrap algorithm on the experimental design, obtaining a set of resampled EDs \( \{ \mathbf{x}^{(b)}_{\text{ED}}, \mathbf{y}^{(b)}_{\text{ED}}, b = 1, ..., B \} \). A corresponding set of coefficients \( y_{\alpha}^{(b)} \) is then calculated for each resampled ED, in fact resulting in a set of \( B \) different PCEs. From Eq. (1), their response at a point \( \mathbf{x} \) is calculated as follows:

\[
y_{\alpha}^{(b)}(\mathbf{x}) = \sum_{\alpha} y_{\alpha}^{(b)} \Psi_{\alpha}(\mathbf{x}), \tag{2}
\]

effectively yielding a response sample \( \{ y_{\text{PC}}^{(b)}(\mathbf{x}), b = 1, ..., B \} \) at each point \( \mathbf{x} \). Empirical quantiles on the response sample can
be used to give local error bounds to the PCE prediction, at the cost of a single ED.

2.3 Adaptive PCE-based reliability

The reliability algorithm presented in this section is an adaptation of the Adaptive Kriging-Monte Carlo Sampling (AK-MCS, Echard et al. (2011)) to PC-bootstrap. The original AK-MCS algorithm consists in creating a Kriging surrogate model (Santner, 2003) from a small-size initial experimental design and iteratively refining it to optimize its performance in a structural reliability context. The algorithm proposed in this paper closely follows the original implementation, with some differences in the ED enrichment phase, due to the different local error estimate for Kriging and bootstrap-PC. It can be summarized as:

0) Algorithm initialization:
   a. Generate an initial small-size ED and calculate a PC-bootstrap surrogate on it.
   b. Generate a reference MCS sample $\chi_{MCS}$ of size $N_{MCS}$ (e.g. $N_{MCS} = 10^6$).

1) Calculate the MCS estimate of $P_f$ with the current PC-bootstrap surrogate on $\chi_{MCS}$. This will yield a sample \{\hat{P}_f(b), b = 1, ..., B\}.

2) Evaluate the convergence criteria (possibly based on $P_f(b)$). If they are met, go to step 5 (termination), otherwise continue to step 3.

3) Evaluate a suitable learning function on $\chi_{MCS}$. Choose one or more $\mathbf{x} \in \chi_{MCS}$ that maximize the learning function and add them to the PC-bootstrap experimental design.

4) Calculate the PC-bootstrap surrogate on the enriched ED. Return to step 1.

5) Algorithm termination: return the last $P_f$ estimate.

2.3.1 Initial experimental design

Due to the global approximation properties of PCE, the initial experimental design is obtained from a space-filling sampling of the input distributions. In this paper, Latin Hypercube Sampling (LHS) is chosen for all the examples provided.

2.3.2 Convergence criteria

The convergence criterion chosen was developed by Schöbi et al. (2015) and is based on the stability of the $\hat{P}_f$ estimate at the current algorithm iteration. The criterion reads:

$$\frac{\hat{P}_f^+ + \hat{P}_f^-}{\hat{P}_f} \leq \varepsilon_{P_f},$$

where $P_f^+$ and $P_f^-$ are the highest and lowest PC-bootstrap estimates of the probability of failure at the current iteration, defined as:

$$\hat{P}_f^+ = \max(P_f(b)),$n

$$\hat{P}_f^- = \min(P_f(b)),$n

and $\varepsilon_{P_f}$ is typically equal to 5%. Convergence is achieved when the criterion in Eq. (4) is satisfied in at least two consecutive iterations of the algorithm.

2.3.3 Learning function

The AK-MCS variation proposed in Schöbi et al. (2015) uses a learning function that is proportional to the misclassification probability of the samples in $\chi_{MCS}$ due to the epistemic uncertainty in the underlying Gaussian process. In this paper we propose a similar enrichment criterion which does not rely on assumptions on the stochastic distribution of the epistemic error in the PC-bootstrap model. Due to the availability of the bootstrap response sample $Y_{PC}(\mathbf{x})$ in Eq. (2), it is straightforward to define the fraction of failed bootstrap replicates (FBR) $U_{FBR}(\mathbf{x})$ as follows:

$$U_{FBR}(\mathbf{x}) = \frac{B_s(\mathbf{x}) - B_f(\mathbf{x})}{B},$$

where $B_s(\mathbf{x})$ and $B_f(\mathbf{x})$ are the number of safe (resp. failed) PC-bootstrap replicates at point $\mathbf{x}$. In other words, $U_{FBR} = 1$ if all the replicates are in the safe domain or in the failure domain and it is equal to zero when half of them are in the safe
and half in the failure domain. Thus $U_{FBR}(x) \approx 0$ is minimum when the epistemic uncertainty in the classification of point $x$ is maximum.

### 2.3.4 Experimental design enrichment

Once a learning function is defined, it is possible to define a design enrichment strategy based on minimizing the global misclassification error of the surrogate model. A single-point enrichment criterion can be defined directly from Eq. (5). During the $k$-th iteration of the algorithm, $U_{FBR}$ is evaluated with the current PC-bootstrap model on the $\mathcal{X}_{MCS}$ sample and the element $x^\star$ that minimizes its absolute value is added to the experimental design:

$$
x^\star = \arg\min_{x^{(i)} \in \mathcal{X}_{MCS}} (U_{FBR}(x^{(i)})),
$$

(6)

A straightforward extension of Eq. (5) to the case when multiple points are added simultaneously at each iteration was also developed in Schöbi et al. (2015). Assuming that $K$ points are to be added at each iteration, $U_{FBR}(\mathcal{X}_{MCS})$ is first clustered in $K$ different regions $\{U_{FBR}^{(k)}, k = 1, \ldots, K\}$ by $k$-means clustering, see Zaki and Meira (2014). Then, Eq. (5) is directly applied to each of the $K$ clusters, yielding one candidate point for each of them.

### 3. Benchmark applications

The proposed adaptive algorithm is applied in this section to three benchmark studies based on Schöbi et al. (2015). The calculations have been carried out using UQLab, the Matlab-based uncertainty quantification framework developed at ETH Zurich (Marelli and Sudret, 2014), more specifically the polynomial chaos expansion module (Marelli and Sudret, 2015).

#### 3.1 Four-branch function

The four-branch function, originally proposed in Waarts (2000), is a common benchmark in structural reliability applications. It is an analytical function that reads:

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

(7)

where the two input variables $X_1$ and $X$ are modelled by two independent Gaussian random variables. A point $x$ belongs to the failure domain if $f(x) \leq 0$. This benchmark is well known because it has a complex-shaped failure domain (solid lines in Fig. 1), hence causing most of the standard reliability algorithms to underestimate the actual failure probability. The reference failure probability calculated with Monte-Carlo sampling ($N_{MCS} = 10^8$, from Schöbi et al. (2015)) is $P_f = 4.460 \cdot 10^{-3}$.

The initial experimental design consists of 50 samples drawn from the input distributions with a space-filling LHS design. The failure domain and the initial design are shown for reference in Fig. 1a. The algorithm was configured to add three samples to the experimental design at each iteration.
The algorithm converged after 78 iterations, with a total cost of 284 model evaluations. The final experimental design is shown in Fig. 1b. The adaptive ED enrichment strategy added most of the samples in close proximity to the limit-state surface. Table 1 reports the final results of the algorithm as well as the reference results.

3.2 Borehole model

The second benchmark considered is the so-called borehole model, an analytical function that predicts the water flow through a borehole (Schöbi et al. (2015)). It is described by the following 8-dimensional input function:

$$v(x) = \frac{2\pi T_u (H_u - H_l)}{\ln(r / r_w) \left[ 1 + \frac{2LT_u}{\ln(r / r_w)} \left( \frac{T_u}{T_l} \right)^2 K_w \right]}$$

where $v(x)$ (m³/year) is the water flow, $r_w$ the borehole radius, $r$ the radius of influence, $T_u$ and $T_l$ the transmissivities of the upper (resp. lower) aquifer, $H_u$ and $H_l$ the potentiometric head of the upper (resp. lower) aquifers, $L$ the length of the borehole and $K_w$ the hydraulic conductivity of the soil. The distributions of the eight input parameters in Eq. (7) are given in Table 2.

### Table 1. Final $P_f$ estimate on the four-branch function and associated costs.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$P_f$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCS</td>
<td>$4.460 \times 10^{-3}$</td>
<td>$10^8$</td>
</tr>
<tr>
<td>PC-bootstrap</td>
<td>$4.465 \times 10^{-3}$</td>
<td>284</td>
</tr>
</tbody>
</table>

The convergence of the $P_f$ estimate vs. the number of limit-state function evaluations is shown graphically in Fig. 3. The convergence criterion in Eq. (4) appears to be quite conservative for this particular example ($\epsilon_{P_f} = 5\%$).
The failure condition for this benchmark is given by $v(x) \geq 190 \text{ m}^3/\text{year}$. The initial ED comprised of 50 space-filling samples (LHS) drawn from the input distributions in Table 2. At each iteration, three additional points were added to the PC-bootstrap ED. The convergence curve of the estimation of $P_f$ and the corresponding final estimates, compared to a reference MCS consisting of $5 \cdot 10^5$ samples are reported in Fig. 3 and Table 3, respectively.

**Table 3. Final $P_f$ estimate on the borehole function and associated costs.**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$P_f$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCS</td>
<td>$1.608 \cdot 10^{-3}$</td>
<td>$5 \cdot 10^5$</td>
</tr>
<tr>
<td>PC-bootstrap</td>
<td>$1.621 \cdot 10^{-3}$</td>
<td>$251$</td>
</tr>
</tbody>
</table>

### 3.2 Two-dimensional truss structure

The final benchmark presented is a finite-element model of a truss structure subject to random loads extensively studied in the uncertainty quantification literature (see, e.g. Blatman and Sudret, (2008), Schöbi et al. (2015)). The input random vector $X$ comprises 10 variables:

$$X = [E_1, E_2, A_1, A_2, P_{(1,\ldots,6)}]^T,$$

(10)

where $E_1$ and $E_2$ are the Young’s moduli of the linear elastic material, $A_1$ and $A_2$ are the cross sections of the bars and $P_{(1,\ldots,6)}$ are the random loads. The distributions of the random parameters are given in Table 4. Fig. 4 shows the structure and the associated random parameters.

**Table 2. Distributions of the truss inputs**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Distribution</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1, E_2$ (Pa)</td>
<td>Lognormal</td>
<td>$[2.1 \cdot 10^{11}, 2.1 \cdot 10^{12}]$</td>
</tr>
<tr>
<td>$A_1$ (m$^2$)</td>
<td>Lognormal</td>
<td>$[2.0 \cdot 10^{-3}, 2.0 \cdot 10^{-4}]$</td>
</tr>
<tr>
<td>$A_2$ (m$^2$)</td>
<td>Lognormal</td>
<td>$[1.0 \cdot 10^{-3}, 1.0 \cdot 10^{-4}]$</td>
</tr>
<tr>
<td>$P_{(1,\ldots,6)}$ (N)</td>
<td>Gumbel</td>
<td>$[5 \cdot 10^4, 7.5 \cdot 10^3]$</td>
</tr>
</tbody>
</table>

The quantity of interest is the displacement at midspan $u$. The failure criterion is set as $u \geq 0.12$ m.

The initial ED for the PC-bootstrap surrogate consists of 50 space-filling (LHS) samples drawn from the input distributions. The ED is enriched with 3 samples per iteration. The resulting estimates of $P_f$ are reported in Table 5, while the corresponding convergence curve is shown in Fig. 5.

**Table 3. Final $P_f$ estimate on the truss benchmark and associated costs.**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$P_f$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCS</td>
<td>$1.462 \cdot 10^{-3}$</td>
<td>$5 \cdot 10^5$</td>
</tr>
<tr>
<td>PC-bootstrap</td>
<td>$1.387 \cdot 10^{-3}$</td>
<td>$215$</td>
</tr>
</tbody>
</table>
4. Conclusions
In this paper we proposed an extension to classical polynomial chaos extensions (PCE) to allow for local error estimates on the surrogate predictions. The bootstrap resampling technique was introduced as a means to get an error estimate on the coefficients. Resampling from the experimental design allows one to derive $B$ different PCEs, which can be used as so many surrogate models of the limit state function. This allows to derive a local misclassification function, which counts how many times a current point $x$ falls into the failure (resp. safe) domain across the bootstrap replicates. This in turn allows for an adaptive learning algorithm that selects the next points to be added to the experimental design. Different analytical examples carried out using the UQLab software have illustrated the convergence and efficiency of the proposed approach.

References


