ETH zürich Research Collection

Other Conference Item

Benchmarking active learning methods for structural reliability analysis

Author(s):
Sudret, Bruno

Publication Date:
2021-10-21

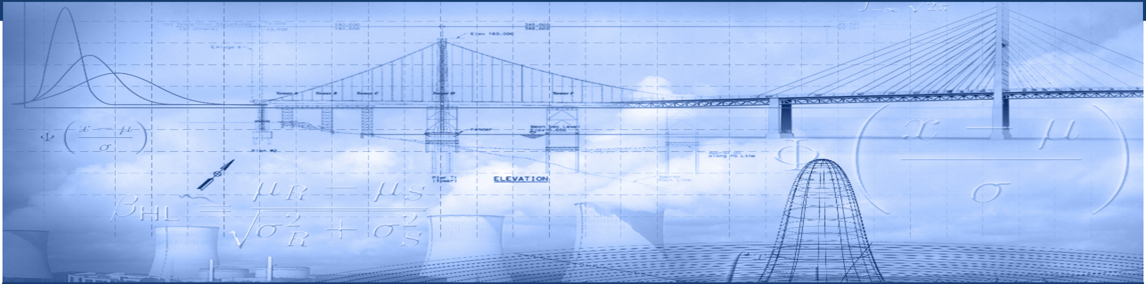
Permanent Link:
https://doi.org/10.3929/ethz-b-000511282 →

Rights / License:
In Copyright - Non-Commercial Use Permitted →

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.

ETH Library

ETH zürich



Benchmarking active learning methods for structural reliability analysis

B. Sudret

Chair of Risk, Safety and Uncertainty Quantification

How to cite?

This presentation is an invited lecture given at the 8th Workshop on Robust and Reliable Design organized by the *GST "Mécanique et Incertain"* of the *Association Française de Mécanique* on October 21, 2021.

How to cite?

Sudret, B. *Benchmarking active learning methods for structural reliability analysis*, 8ème Journée de la conception robuste et fiable – Approches universitaires et industrielles, Sigma Clermont (France), October 21, 2021 (online).

Main reference

Moustapha, M., Marelli, S. & Sudret, B. (2021) A generalized framework for active learning reliability: survey and benchmark, submitted to Structural Safety, ArXiv: 2106.01713.

Chair of Risk, Safety and Uncertainty quantification

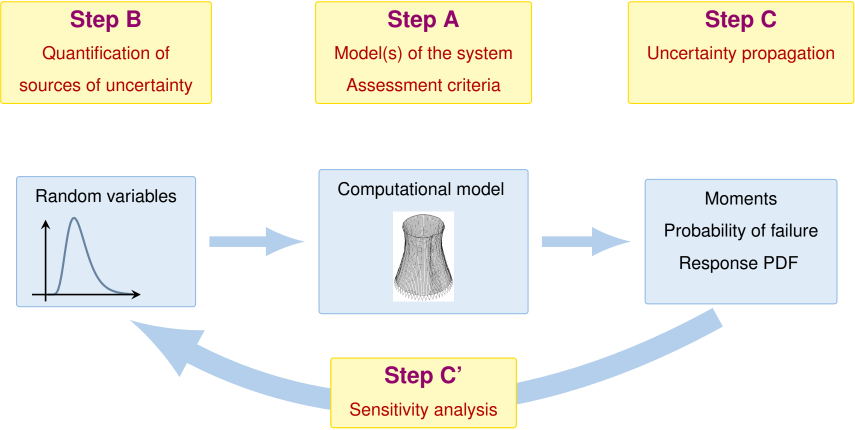
The Chair carries out research projects in the field of uncertainty quantification for engineering problems with applications in structural reliability, sensitivity analysis, model calibration and reliability-based design optimization

Research topics

- Uncertainty modelling for engineering systems
- Structural reliability analysis
- Surrogate models (polynomial chaos expansions, Kriging, support vector machines)
- Bayesian model calibration and stochastic inverse problems
- Global sensitivity analysis
- Reliability-based design optimization

http://www.rsuq.ethz.ch

Global framework for uncertainty quantification



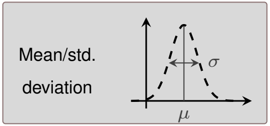
B. Sudret, *Uncertainty propagation and sensitivity analysis in mechanical models – contributions to structural reliability and stochastic spectral methods (2007)*

Step C: uncertainty propagation

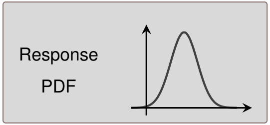
Goal: estimate the uncertainty / variability of the quantities of interest (QoI) $Y = \mathcal{M}(\boldsymbol{X})$ due to the input uncertainty $f_{\boldsymbol{X}}$

- Output statistics, *i.e.* mean, standard deviation, etc.

$$\mu_Y = \mathbb{E}_{\boldsymbol{X}}\left[\mathcal{M}(\boldsymbol{X})\right]$$
$$\sigma_Y^2 = \mathbb{E}_{\boldsymbol{X}}\left[\left(\mathcal{M}(\boldsymbol{X}) - \mu_Y\right)^2\right]$$

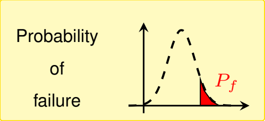


- Distribution of the QoI



- Probability of exceeding an admissible threshold y_{adm}

$$P_f = \mathbb{P}\left(Y \geq y_{adm}\right)$$

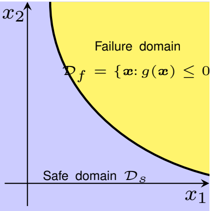


Limit state function

- For the assessment of the system's performance, failure criteria are defined, *e.g.* :

$$\text{Failure} \quad \Leftrightarrow \quad QoI = \mathcal{M}(\boldsymbol{x}) \geq y_{adm}$$

- The failure criterion is cast as a limit state function (performance function) $g: \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} \mapsto \mathbb{R}$ such that:



$g\left(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x})\right) \leq 0$	Failure domain $\mathcal{D}_f$
$g\left(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x})\right) > 0$	Safety domain $\mathcal{D}_s$
$g\left(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x})\right) = 0$	Limit state surface

e.g. $g(\boldsymbol{x}) = y_{adm} - \mathcal{M}(\boldsymbol{x})$

Probability of failure

$$P_f = \mathbb{P}\left(\left\{\boldsymbol{X} \in D_f\right\}\right) = \mathbb{P}\Big(g\left(\boldsymbol{X}, \mathcal{M}(\boldsymbol{X})\right)\Big) = \int_{\mathcal{D}_f = \{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} : \, g(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x})) \leq 0\}} f_{\boldsymbol{X}}(\boldsymbol{x})\, d\boldsymbol{x} \leq 0$$

- Multidimensional integral ($d = 10 - 100^+$), implicit domain of integration
- Failures are (usually) rare events: sought probability in the range 10^{-2} to 10^{-8}

Classical methods

Approximation methods

Hasofer & Lind (1974), Rackwitz & Fiessler (1978)

- First-/Second- order reliability method (FORM/SORM)
  - Relatively inexpensive semi-analytical methods
  - Convergence is not guaranteed (*e.g.* in presence of multiple failure regions)

Simulation methods

Melchers (1989), Au & Beck (2001), Koutsourelakis *et al.* (2001)

- Monte Carlo simulation
  - Unbiased but slow convergence rate
- Variance-reduction methods
  - *e.g.* Importance sampling, subset simulation, line sampling, etc.
  - Their computational costs remain high (*i.e.* $\mathcal{O}(10^{3-4})$ model runs)

Surrogate models can be used to leverage the computational cost of simulation methods

Outline

Surrogate models for uncertainty quantification

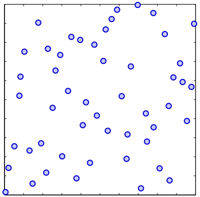
A surrogate model $\tilde{\mathcal{M}}$ is an approximation of the original computational model $\mathcal{M}$ with the following features:

- It is built from a limited set of runs of the original model $\mathcal{M}$ called the experimental design $\mathcal{X} = \left\{ \boldsymbol{x}^{(i)},\, i = 1, \dots, n \right\}$
- It assumes some regularity of the model $\mathcal{M}$ and some general functional shape

Name	Shape	Parameters
Polynomial chaos expansions	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$	$\boldsymbol{a}_{\boldsymbol{\alpha}}$
Low-rank tensor approximations	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{l=1}^{R} b_l \left(\prod_{i=1}^{M} v_l^{(i)}(x_i) \right)$	$b_l ,\ z_{k,l}^{(i)}$
Kriging (a.k.a Gaussian processes)	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \boldsymbol{\beta}^{\mathsf{T}} \cdot \boldsymbol{f}(\boldsymbol{x}) + Z(\boldsymbol{x}, \omega)$	$\boldsymbol{\beta} ,\ \sigma_Z^2 ,\ \boldsymbol{\theta}$
Support vector machines	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{n} a_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\boldsymbol{a} ,\ b$
(Deep) Neural networks	$\tilde{\mathcal{M}}(\boldsymbol{x}) = f_n \left(\cdots f_2 \left(b_2 + f_1 \left(b_1 + \boldsymbol{w}_1 \cdot \boldsymbol{x} \right) \cdot \boldsymbol{w}_2 \right) \right)$	$\boldsymbol{w}, \boldsymbol{b}$

- It is fast to evaluate

Ingredients for building a surrogate model



- Select an experimental design $\mathcal{X}$ that covers at best the domain of input parameters: Latin hypercube sampling (LHS), low-discrepancy sequences
- Run the computational model $\mathcal{M}$ onto $\mathcal{X}$ exactly as in Monte Carlo simulation
- Smartly post-process the data $\{\mathcal{X}, \mathcal{M}(\mathcal{X})\}$ through a learning algorithm

Name	Learning method
Polynomial chaos expansions	sparse grid integration, least-squares, compressive sensing
Low-rank tensor approximations	alternate least squares
Kriging	**maximum likelihood, Bayesian inference**
Support vector machines	quadratic programming

Advantages of surrogate models

Usage

$$\mathcal{M}(\boldsymbol{x}) \approx \tilde{\mathcal{M}}(\boldsymbol{x})$$

hours per run — seconds for 10^6 runs

Advantages

- Non-intrusive methods: based on runs of the computational model, exactly as in Monte Carlo simulation
- Suited to high performance computing: “embarrassingly parallel”

Challenges

- Need for rigorous validation
- Communication: advanced mathematical background

Efficiency: 2-3 orders of magnitude less runs compared to Monte Carlo

Outline

Gaussian process modelling

Gaussian process modelling (*a.k.a.* Kriging) assumes that the map $y = \mathcal{M}(\boldsymbol{x})$ is a realization of a Gaussian process:

$$Y(\boldsymbol{x}, \omega) = \sum_{j=1}^{p} \beta_j \, f_j(\boldsymbol{x}) + \sigma \, Z(\boldsymbol{x}, \omega)$$

where:

- $\boldsymbol{f} = \{f_j, j = 1, \ldots, p\}^{\mathsf{T}}$ are predefined (*e.g.* polynomial) functions which form the trend or regression part
- $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_p\}^{\mathsf{T}}$ are the regression coefficients
- σ^2 is the variance of $Y(\boldsymbol{x}, \omega)$
- $Z(\boldsymbol{x}, \omega)$ is a stationary, zero-mean, unit-variance Gaussian process

$$\mathbb{E}\left[Z(\boldsymbol{x}, \omega)\right] = 0 \qquad \mathrm{Var}\left[Z(\boldsymbol{x}, \omega)\right] = 1 \qquad \forall\, \boldsymbol{x} \in \mathbb{X}$$

The Gaussian measure artificially introduced is different from the aleatory uncertainty on the model parameters $\boldsymbol{X}$

Kriging equations

Data

- Given is an experimental design $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ and the output of the computational model $\boldsymbol{y} = \{y_1 = \mathcal{M}(\boldsymbol{x}_1), \dots, y_N = \mathcal{M}(\boldsymbol{x}_N)\}$
- We assume that $\mathcal{M}(\boldsymbol{x})$ is a realization of a Gaussian process $Y(\boldsymbol{x})$ such that the values $y_i = \mathcal{M}(\boldsymbol{x}_i)$ are known at the various points $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$
- Of interest is the prediction at a new point $\boldsymbol{x}_0 \in \mathbb{X}$, denoted by $\hat{Y}_0 \equiv \hat{Y}(\boldsymbol{x}_0, \omega)$, which will be used as a surrogate $\tilde{\mathcal{M}}(\boldsymbol{x}_0)$

$\hat{Y}_0$ is obtained as as a conditional Gaussian variable:

$$\hat{Y}_0 = Y\left(\boldsymbol{x}_0 \mid Y(\boldsymbol{x}_1) = y_1, \dots, Y(\boldsymbol{x}_N) = y_N\right)$$

Kriging mean predictor and variance

Santner, William & Notz (2003)

The conditional distribution of $\widehat{Y}_0$ given the observations $\{Y(\boldsymbol{x}_i) = y_i\}_{i=1}^n$ is a Gaussian variable:

$$\widehat{Y}_0 \sim \mathcal{N}(\mu_{\widehat{Y}_0}, \sigma^2_{\widehat{Y}_0})$$

Mean predictor : used as surrogate model

$$\mu_{\widehat{Y}_0} = \boldsymbol{f}_0^{\mathsf{T}} \widehat{\boldsymbol{\beta}} + \boldsymbol{r}_0^{\mathsf{T}} \mathbf{R}^{-1} \left(\boldsymbol{y} - \mathbf{F} \widehat{\boldsymbol{\beta}} \right)$$

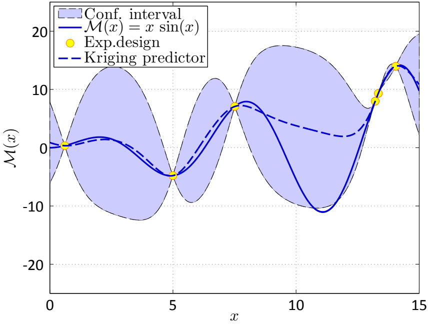
where the regression coefficients $\widehat{\boldsymbol{\beta}}$ are obtained from the generalized least-square solution:

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{F}^{\mathsf{T}} \mathbf{R}^{-1} \mathbf{F} \right)^{-1} \mathbf{F}^{\mathsf{T}} \mathbf{R}^{-1} \boldsymbol{y}$$

Kriging variance : local prediction uncertainty

$$\sigma^2_{\widehat{Y}_0} = \mathbb{E}\left[(\widehat{Y}_0 - Y_0)^2 \right] = \sigma^2 \left(1 - \boldsymbol{r}_0^{\mathsf{T}} \mathbf{R}^{-1} \boldsymbol{r}_0 + \boldsymbol{u}_0^{\mathsf{T}} \left(\mathbf{F}^{\mathsf{T}} \mathbf{R}^{-1} \mathbf{F} \right)^{-1} \boldsymbol{u}_0 \right) \qquad \boldsymbol{u}_0 = \mathbf{F}^{\mathsf{T}} \mathbf{R}^{-1} \boldsymbol{r}_0 - \boldsymbol{f}_0$$

One-dimensional example



Computational model

$$x \mapsto x \sin x \qquad \text{for } \; x \in [0,\, 15]$$

Experimental design

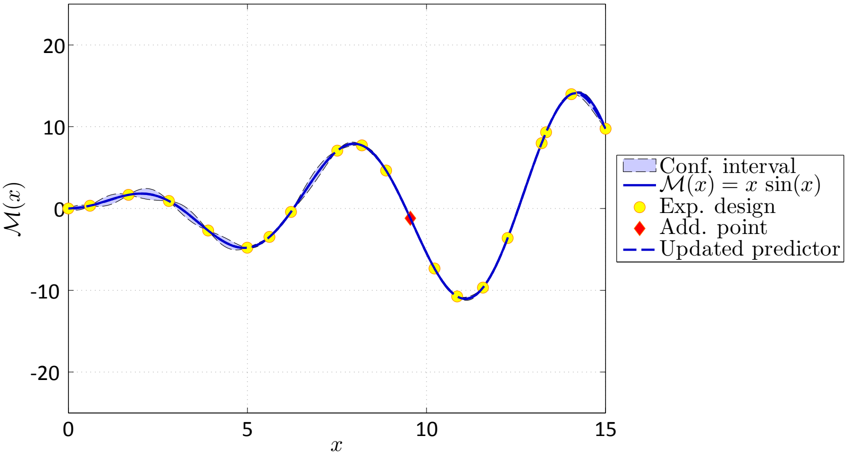
Six points selected in the range $[0,\, 15]$ using Monte Carlo simulation

Confidence intervals

With confidence level $(1-\alpha)$, *e.g.* 95%, one gets:

$$\mu_{\widehat{Y}_0} - 1.96\, \sigma_{\widehat{Y}_0} \leq \mathcal{M}(\boldsymbol{x}_0) \leq \mu_{\widehat{Y}_0} + 1.96\, \sigma_{\widehat{Y}_0}$$

Sequential updating



Outline

Active learning reliability using a Kriging surrogate

Procedure

- Start from an initial experimental design $\mathcal{X}$ and build the initial Kriging surrogate of the limit state function $\hat{g}_0$
- At each iteration k
  - Compute an estimation of P_f (and a confidence interval from the current surrogate)
  - Check a convergence criterion
  - Select the next point(s) to be added to $\mathcal{X}$: enrichment (a.k.a. in-fill) criterion
  - Update the Kriging surrogate to $\hat{g}_k$

Early approaches

- Efficient global reliability analysis (EGRA) — Bichon *et al.* (2008)
- Active Kriging - Monte Carlo simulation (AK-MCS) — Echard *et al.* (2011)

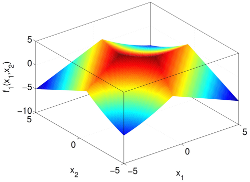
Example: series system

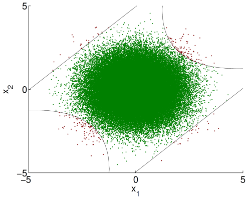
Schöbi *et al.* , ASCE J. Risk Unc. (2016)

Consider the system reliability analysis defined by:

$$g(\boldsymbol{x}) = \min \begin{pmatrix} 3 + 0.1\,(x_1 - x_2)^2 - \frac{x_1+x_2}{\sqrt{2}} \\ 3 + 0.1\,(x_1 - x_2)^2 + \frac{x_1+x_2}{\sqrt{2}} \\ (x_1 - x_2) + \frac{6}{\sqrt{2}} \\ (x_2 - x_1) + \frac{6}{\sqrt{2}} \end{pmatrix}$$

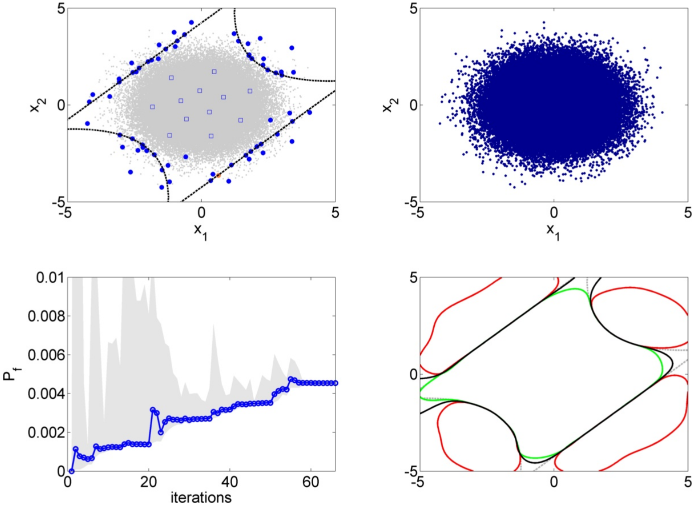
where $X_1, X_2 \sim \mathcal{N}(0, 1)$





- Initial design: LHS of size 12 (transformed into the standard normal space)
- In each iteration, one point is added (maximize the probability of missclassification)
- The mean predictor $\mu_{\widehat{\mathcal{M}}}(\boldsymbol{x})$ is used, as well as the bounds $\mu_{\widehat{\mathcal{M}}}(\boldsymbol{x}) \pm 2\sigma_{\widehat{\mathcal{M}}}(\boldsymbol{x})$ so as to get bounds on P_f: $\hat{P}_f^- \leq \hat{P}_f^0 \leq \hat{P}_f^+$

Results with classical Kriging



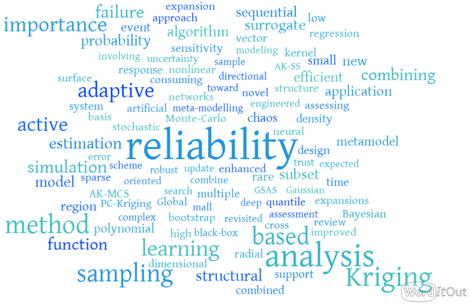
Outline

Active learning reliability methods

Teixeira *et al.* (2021), Moustapha *et al.* (2021) (submitted)

Numerous papers on active learning called AK-XXX-YYY in the last few years!

- AK-MCS is a cornerstone for the development of active learning reliability strategies
- Most methods in the literature are built by modifying:
  - the surrogate model
  - the learning function
  - the algorithm for reliability estimation
  - the stopping criterion



A module-oriented survey

Moustapha *et al.* (2021) (submitted)

	Monte Carlo simulation	Subset simulation	Importance sampling	Other
Kriging	Bichon et. al (2008) Echard et. al (2011) Hu & Mahadevan (2016) Wen et al. (2016) Fauriat & Gayton (2017) Jian et. al (2017) Peijuan et al. (2017) Sun et al. (2017) Lelievre et al. (2018) Xiao et al. (2018) Jiang et al. (2019) Tong et al. (2019) Wang & Shafieezadeh (2019) Wang & Shafieezadeh (SAMO, 2019) Zhang, Wang et al. (2019)	Huang et al. (2016) Tong et al. (2015) Ling et al. (2019) Zhang et al. (2019)	Dubourg et al. (2012) Balesdent et al. (2013) Echard et al. (2013) Cadini et al. (2014) Liu et al. (2015) Zhao et al. (2015) Gaspar et al. (2017) Razaaly et al. (2018) Yang et al. (2018) Zhang & Taflanidis (2018) Pan et al. (2020) Zhang et al. (2020)	Lv et al. (2015) Bo & HuiFeng (2018) Guo et al. (2020)
PCE	Chang & Lu (2020) Marelli & Sudret (2018) Pan et al. (2020)			
SVM	Basudhar & Missoum (2013) Lacaze & Missoum (2014) Pan et al. (2017)	Bourinet et al. (2011) Bourinet (2017)		
RSM/RBF	Li et al. (2018) Shi et al. (2019)			Rajakeshir (1993) Rousouly et al. (2013)
Neural networks	Chojazyck et al. (2015) Gomes et al. (2019) Li & Wang (2020) [Deep NN]	Sundar & Shields (2016)	Chojazyck et al. (2015)	
Other	Schoebi & Sudret (2016) Sadoughi et al. (2017) Wagner et al. (2021)			

— U — EFF — Other variance-based — Distance-based — Bootstrap-based — Sensitivity-based — Cross-validation/Ensemble-based — ad-hoc/other

General framework

Modular framework which consists of independent blocks that can be assembled in a black-box fashion

Surrogate model	Reliability estimation	Learning function	Stopping criterion
Kriging	Monte Carlo	U	LF-based
PCE	Subset simulation	EFF	Stability of β
SVR	Importance sampling	FBR	Stability of P_f
PC-Kriging	Line sampling	CMM	Bounds on β
Neural networks	Directional sampling	SUR	Bounds on P_f
...	...	...	...

Active learning for reliability analysis

Initialization

  Initial experimental design $\mathcal{ED} = \{\boldsymbol{\chi}^{(1)}, \dots, \boldsymbol{\chi}^{(n)}\}$

  Converged = FALSE

while *not(Converged)* **do**

  Train a surrogate model $\tilde{g}$ on the current experimental design

  Compute the failure probability $\hat{P}_f^0$, and its bounds $[\hat{P}_f^-, \hat{P}_f^+]$ using $\tilde{g}$

  if *Stopping criterion fulfilled* **then**

    Converged = TRUE

  else

    Evaluate the learning function LF on $\mathcal{X}$

    Enrich the ED: $\boldsymbol{\chi}^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} LF(\boldsymbol{x})$

    Update the experimental design: $\mathcal{ED} \leftarrow \mathcal{ED} \cup \{\boldsymbol{\chi}^*\}$

  end

end

Return Probability of failure $\hat{P}_f^0$ and confidence interval $[\hat{P}_f^-, \hat{P}_f^+]$

Extensive benchmark: Set-up

Reliability method	Surrogate model	Learning function	Stopping criterion	
Monte Carlo simulation Subset simulation Importance sampling	Kriging PC-Kriging	U EFF	Beta bounds Beta stability Combined	$3 \cdot 2 \cdot 2 \cdot 3 = 36$ **strategies**
Monte Carlo simulation Subset simulation Importance sampling	PCE	FBR	Beta stability	3 **strategies**
Subset simulation, Importance sampling w/o metamodel				2 **strategies**

In total $39 + 2 = 41$ strategies are tested

Moustapha, M., Marelli, S. & Sudret, B. A generalized framework for active learning reliability: survey and benchmark (2021), ArXiv: 2106.01713.

Extensive benchmark: options for the various methods

Kriging

- Trend: Constant
- Kernel: Gaussian
- Calibration: MLE

PCE

- Degree: $1 - 20$
- q-norm : 0.8
- Calibration: LAR

PC-Kriging

- Same as Kriging
- same as PCE but...
- Degree $1 - 3$

Monte Carlo simulation

- Max. sample size: 10^7
- Target C.o.V: 2.5%
- Batch size: 10^5

Importance sampling

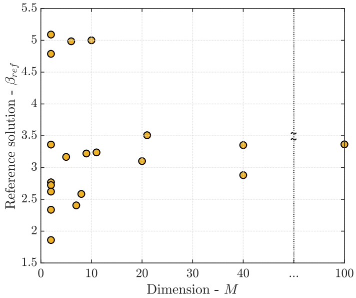
- Max. sample size: 10^4
- Target C.o.V: 2.5%
- Instrumental density: Standard Gaussian centered on the MPFP

Subset simulation

- Max. sample size: 10^7
- Target C.o.V: 2.5%
- Batch size: 10^5
- Conditional probability: $p_0 = 0.25$

Selected problems

- 20 problems selected from the literature
- 11 come from the TNO benchmark (https://rprepo.readthedocs.io/en/latest/)
- Wide spectrum of problems in terms of
  - Dimensionality
  - Reliability index $\beta = -\Phi^{-1}(P_f)$



Problem	M	$P_{f,\mathrm{ref}}$	Reference
01 (TNO RP14)	5	$7.69 \cdot 10^{-4}$	Rozsas & Slobbe 2019
02 (TNO RP24)	**2**	$2.90 \cdot 10^{-3}$	Rozsas & Slobbe 2019
03 (TNO RP28)	2	$\mathbf{1.31 \cdot 10^{-7}}$	Rozsas & Slobbe 2019
04 (TNO RP31)	2	$3.20 \cdot 10^{-3}$	Rozsas & Slobbe 2019
05 (TNO RP38)	7	$8.20 \cdot 10^{-3}$	Rozsas & Slobbe 2019
06 (TNO RP53)	2	$\mathbf{3.14 \cdot 10^{-2}}$	Rozsas & Slobbe 2019
07 (TNO RP54)	20	$9.79 \cdot 10^{-4}$	Rozsas & Slobbe 2019
08 (TNO RP63)	**100**	$3.77 \cdot 10^{-4}$	Rozsas & Slobbe 2019
09 (TNO RP7)	2	$9.80 \cdot 10^{-3}$	Rozsas & Slobbe 2019
10 (TNO RP107)	10	$2.85 \cdot 10^{-7}$	Rozsas & Slobbe 2019
11 (TNO RP111)	2	$7.83 \cdot 10^{-7}$	Rozsas & Slobbe 2019
12 (4-branch series)	2	$3.85 \cdot 10^{-4}$	Echard et al. (2011)
13 (Hat function)	2	$4.40 \cdot 10^{-3}$	Schoebi et al. (2016)
14 (Damped oscillator)	8	$4.80 \cdot 10^{-3}$	Der Kiureghian (1990)
15 (Non-linear oscillator)	6	$3.47 \cdot 10^{-7}$	Echard et al. (2011,2013)
16 (Frame)	21	$2.25 \cdot 10^{-4}$	Echard et al. (2013)
17 (HD function)	40	$2.00 \cdot 10^{-3}$	Sadoughi et al. (2017)
18 (VNL function)	40	$1.40 \cdot 10^{-3}$	Bichon et al. (2008)
19 (Transmission tower 1)	11	$5.76 \cdot 10^{-4}$	FEM (172 bars, 51 nodes)
20 (Transmission tower 2)	9	$6.27 \cdot 10^{-4}$	FEM (172 bars, 51 nodes)

Comparison of the various strategies

Approximately $12,000$ reliability analyses were run:
41 strategies - 20 problems - 15 replications

Three evaluation criteria:

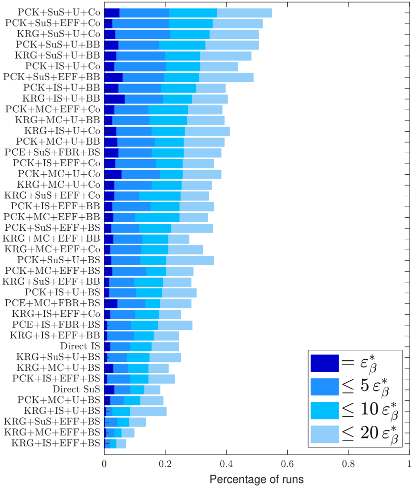
- Number of model evaluations: N_{eval}
- Accuracy: $\varepsilon = \left|\beta - \beta_{\mathrm{ref}}\right| / \beta_{\mathrm{ref}}$
- Efficiency: $\Delta = \varepsilon \, N_{\mathrm{eval}} / N_{\mathrm{med}}$

  where N_{med} is the median number of model evaluations for each problem

For each criterion:

- Ranking of the strategies as a whole
- Performance of the methods w.r.t. problem feature (dimensionality, range of P_f)

Ranking of the strategies: accuracy of β

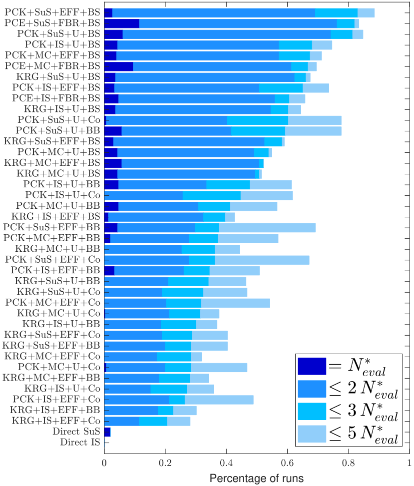


How many times a method ranks best in terms of smallest error on beta (resp. within 5, 10 or 20 times this relative error)?

$$\varepsilon = \left|\beta - \beta_{\text{ref}}\right| / \beta_{\text{ref}}$$

- Best approach: PC-Kriging + SuS + U + Combined stopping criterion
- Worst approaches: Kriging + IS + EFF + BS

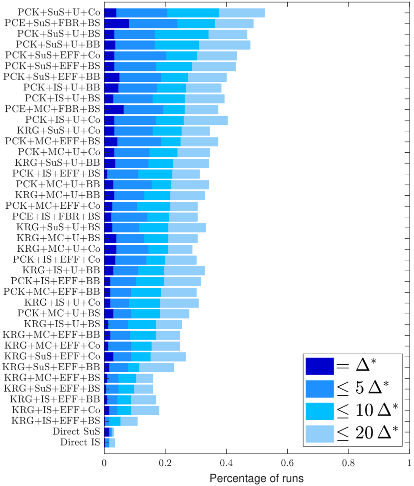
Ranking of the strategies: number of model evaluations



How many times a method ranks best (resp. within 2, 3, 5 times the lowest cost denoted N^*_{eval}) ?

- Best approach: PC-Kriging + SuS + EFF + BS
- Worst approache: Direct IS

Ranking of the strategies: efficiency



How many times a method ranks best according to efficiency Δ (resp. within 5, 10, 20 times the best)?

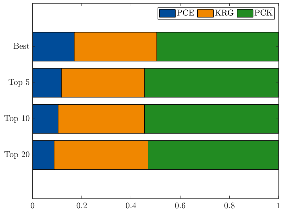
$$\Delta = \varepsilon_\beta \frac{N_{\text{eval}}}{\overline{N}_{\text{eval}}}$$

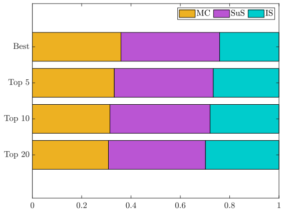
where $\overline{N}_{\text{eval}}$ is the median number of model evaluations for a particular problem (over all methods and replications)

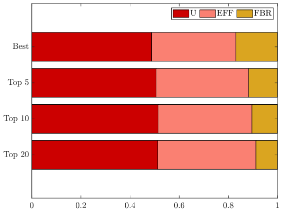
- Best approach: PC-Kriging + SuS + U + Combined stopping criterion
- Worst approaches: Direct SuS and Direct IS

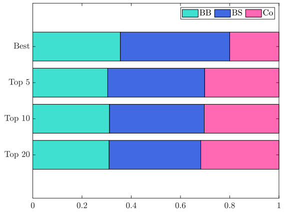
Results aggregated by method

Percentage of times a method is first or in the Top 5, 10, 20 w.r.t. Δ (regardless of the strategy)









- Surrogates: PC-Kriging dominates by far
- Reliability: Slight advantage to subset simulation
- Learning function: U dominates both EFF and FBR
- Stopping criterion: Slight advantage to the stability criterion

Summary of the results

Recommendations w.r.t. the problem feature

Module	Dimensionality		Magnitude of the reliability index	
	$M < 20$	$20 \leq M \leq 100$	$\beta < 3.5$	$\beta \geq 3.5$
Surrogate model	PCK	PCE	PCE/PCK	PCK
Reliability method	SuS	SuS	SuS	SuS
Learning function	U	FBR	U/FBR	U
Stopping criterion	β_{bo},β_{co}	β_{bo} / β_{co}	β_{bo},β_{co}	β_{bo}

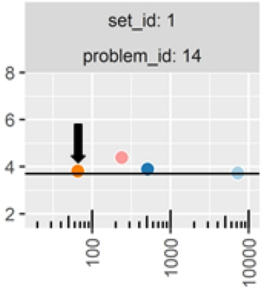
Main take-away

There is no drawback in using surrogates compared to a direct solution

TNO Benchmark: performance of UQLab "ALR" module

Rozsas & Slobbe (2019)

- Truly black-box benchmark with 27 problems
- Limit state functions not known to the participants and only accessible through an anonymous server
- Our solution: the "best approach" previously highlighted (PCK + SuS + U + Co)



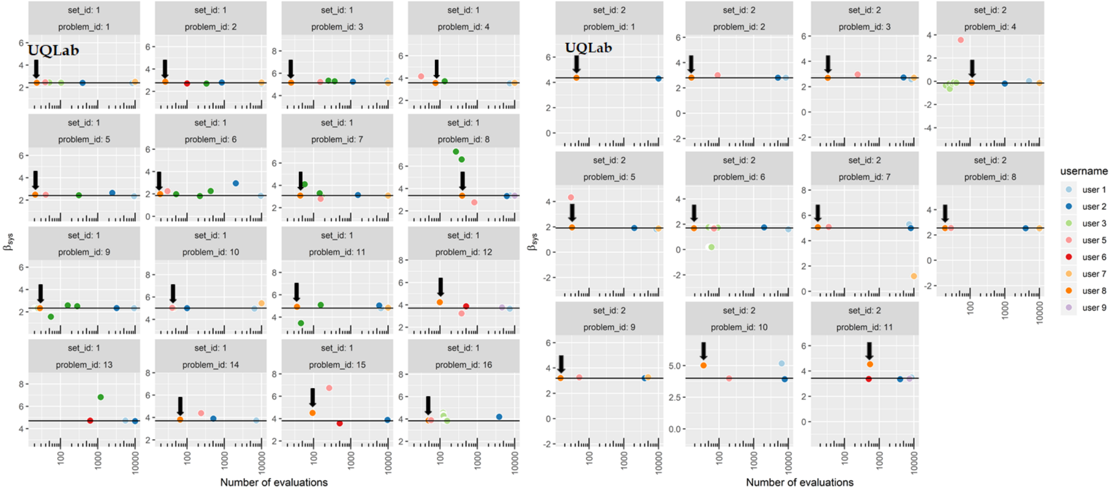
Summary plot (TNO)

- Reference solution: black line
- Zero, one or more points per participant
- X: number of runs (log scale)
- Y: obtained β index

best approach: "on the line / to the left"

TNO Benchmark: performance of UQLab “ALR” module

Rozsas & Slobbe (2019)



Component reliability

System reliability

Conclusions

- Estimating low probabilities of failure in high-dimensional problems requires more refined algorithms than plain MCS
- Recent research on surrogate models (*e.g.* Kriging and polynomial chaos expansions) and active learning has brought new extremely efficient algorithms
- Accurate estimations of P_f's (not of β !) are obtained with $\mathcal{O}(100)$ runs of the computer code regardless of their magnitude
- All the presented algorithms are available in the general-purpose uncertainty quantification software UQLab (V.1.4, "Active learning reliability" module)

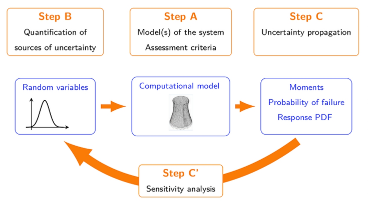
UQLab

The Framework for Uncertainty Quantification

OVERVIEW | FEATURES | DOCUMENTATION | DOWNLOAD/INSTALL | ABOUT | COMMUNITY

"Make uncertainty quantification available for anybody, in any field of applied science and engineering"

www.uqlab.com



- MATLAB®-based Uncertainty Quantification framework
- State-of-the art, highly optimized open source algorithms
- Fast learning curve for beginners
- Modular structure, easy to extend
- Exhaustive documentation

UQLab: The Uncertainty Quantification Software

- **free access to academia**
- Close to 4,000 registered users
- 1,500+ active users from 92 countries

http://www.uqlab.com

- The cloud version of UQLab, accessible via an API (SaaS)
- Available with python bindings for beta testing

https://uqpylab.uq-cloud.io/

Country	# Users
United States	620
China	571
France	362
Switzerland	302
Germany	283
United Kingdom	163
India	160
Italy	151
Brazil	144
Canada	92

As of October 19, 2021

Questions ?

Chair of Risk, Safety & Uncertainty Quantification

www.rsuq.ethz.ch

The Uncertainty Quantification Software

www.uqlab.com

The Uncertainty Quantification Community

www.uqworld.org