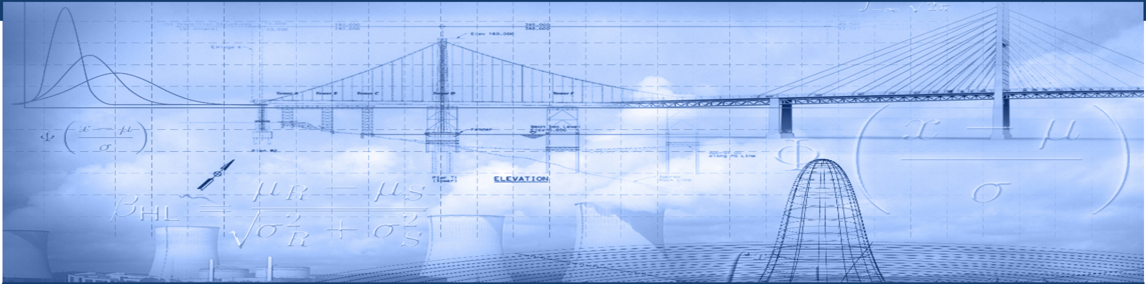
ETH zürich



Surrogate models and active learning for reliability analysis

B. Sudret, M. Moustapha, S. Marelli

Chair of Risk, Safety and Uncertainty Quantification

13th International Workshop on Rare-Event Simulation – Paris (France) – May 18-21, 2021

How to cite?

This presentation is an invited lecture given at 13th International Workshop on Rare-Event Simulation on May 19, 2021.

How to cite

Sudret, B., Moustapha, M. and Marelli, S., *Surrogate models and active learning for reliability analysis*, 13th International Workshop on Rare-Event Simulation, Paris (France), May 18-21, 2021.

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

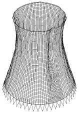
- Uncertainty modelling for engineering systems
- Structural reliability analysis
- Surrogate models (polynomial chaos expansions, Kriging, support vector machines)
- Bayesian model calibration and stochastic inverse problems
- Global sensitivity analysis
- Reliability-based design optimization

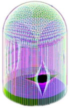
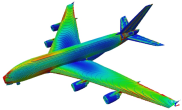
http://www.rsuq.ethz.ch

Computational models in engineering

Complex engineering systems are designed and assessed using computational models, a.k.a simulators that allows us:

- To explore the design space (“virtual prototypes”)
- To optimize the system (*e.g.* minimize the mass) under performance constraints
- To assess its robustness w.r.t uncertainty and its reliability
- Together with experimental data for calibration purposes



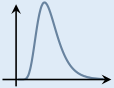
Global framework for uncertainty quantification

Step B
Quantification of sources of uncertainty

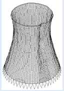
Step A
Model(s) of the system
Assessment criteria

Step C
Uncertainty propagation

Random variables



Computational model



Moments
Probability of failure
Response PDF

Step C'
Sensitivity analysis

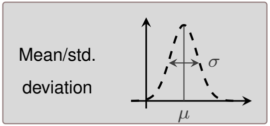
B. Sudret, *Uncertainty propagation and sensitivity analysis in mechanical models – contributions to structural reliability and stochastic spectral methods (2007)*

Step C: uncertainty propagation

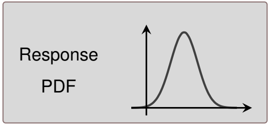
Goal: estimate the uncertainty / variability of the quantities of interest (QoI) $Y = \mathcal{M}(\boldsymbol{X})$ due to the input uncertainty $f_{\boldsymbol{X}}$

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

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

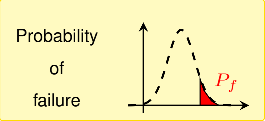


- Distribution of the QoI



- 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.* :

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

Examples:

+ admissible stress / displacements in civil engineering
+ max. temperature in heat transfer problems
+ crack propagation criterion in fracture mechanics

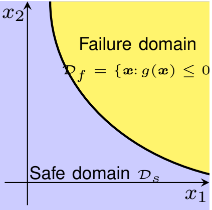
- The failure criterion is cast as a limit state function (performance function) $g: \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} \mapsto \mathbb{R}$ such that:

$g(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x})) \leq 0$ Failure domain $\mathcal{D}_f$

$g(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x})) > 0$ Safety domain $\mathcal{D}_s$

$g(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x})) = 0$ Limit state surface

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

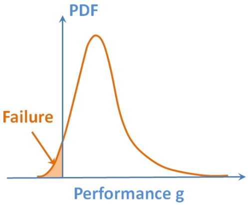


Probability of failure

Definition

$$P_f = \mathbb{P}\left(\left\{\boldsymbol{X} \in D_f\right\}\right) = \mathbb{P}\left(g\left(\boldsymbol{X}, \mathcal{M}(\boldsymbol{X})\right) \leq 0\right)$$

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



Features

- Multidimensional integral, whose dimension is equal to the number of basic input variables $M = \dim \boldsymbol{X}$
- Implicit domain of integration defined by a condition related to the sign of the limit state function:

$$\mathcal{D}_f = \{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} : \quad g(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x})) \leq 0\}$$

- 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:

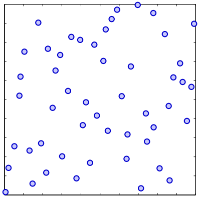
- It is built from a limited set of runs of the original model $\mathcal{M}$ called the experimental design $\mathcal{X} = \left\{\boldsymbol{x}^{(i)},\, i = 1, \dots, n\right\}$
- It assumes some regularity of the model $\mathcal{M}$ and some general functional shape

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

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

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 $\boldsymbol{X}$

Assumptions on the trend and the zero-mean process

Prior assumptions are made based on the existing knowledge on the model to surrogate (linearity, smoothness, etc.)

Trend

- Simple Kriging: known constant β
- Ordinary Kriging: $p = 1$, unknown constant β
- Universal Kriging: f_j's is a set of *e.g.* polynomial functions, *e.g.* $\left\{ f_j(x) = x^{j-1},\ j = 1, \ldots, p \right\}$ in 1D

Type of auto-correlation function of $Z(\boldsymbol{x})$

A family of auto-correlation function $R(\cdot;\ \boldsymbol{\theta})$ is selected:

$$\text{Cov}\left[Z(\boldsymbol{x}), Z(\boldsymbol{x}')\right] = \sigma^2\, R(\boldsymbol{x}, \boldsymbol{x}';\ \boldsymbol{\theta})$$

e.g. square exponential, generalized exponential, Matérn, etc.

$$R_{\text{Matérn}}(h;\ \nu = 5/2) = (1 + \sqrt{5}\, h + \frac{5}{3}\, h^2)\, \exp(-\sqrt{5}\, h)$$

Outline

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:

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

Joint distribution of the predictor / observations

- For each point $\boldsymbol{x}_i \in \mathcal{X}$, $Y_i \equiv Y(\boldsymbol{x}_i)$ is a Gaussian variable:

$$Y_i = \sum_{j=1}^{p} \beta_j \, f_j(\boldsymbol{x}_i) + \sigma Z_i = \boldsymbol{f}_i^{\mathsf{T}} \cdot \boldsymbol{\beta} + \sigma \, Z_i \qquad Z_i \sim \mathcal{N}(0,1)$$

- The joint distribution of $\{Y_0, Y_1, \ldots, Y_N\}^{\mathsf{T}}$ is Gaussian:

$$\left\{ \begin{array}{c} Y_0 \\ \boldsymbol{Y} \end{array} \right\} \sim \mathcal{N}_{1+N} \left(\left\{ \begin{array}{c} \boldsymbol{f}_0^{\mathsf{T}} \boldsymbol{\beta} \\ \mathbf{F} \, \boldsymbol{\beta} \end{array} \right\}, \sigma^2 \left[\begin{array}{cc} 1 & \boldsymbol{r}_0^{\mathsf{T}} \\ \boldsymbol{r}_0 & \mathbf{R} \end{array} \right] \right)$$

- Regression matrix $\mathbf{F}$ of size $(N \times p)$

$$\mathbf{F}_{ij} = f_j(\boldsymbol{x}_i)$$

$$i = 1, \ldots, N, \; j = 1, \ldots, p$$

- Vector of regressors $\boldsymbol{f}_0$ of size p

$$\boldsymbol{f}_0 = \{f_1(\boldsymbol{x}_0), \ldots, f_p(\boldsymbol{x}_0)\}$$

- Correlation matrix $\mathbf{R}$ of size $(N \times N)$

$$\mathbf{R}_{ij} = R(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta})$$

- Cross-correlation vector $\boldsymbol{r}_0$ of size N

$$\boldsymbol{r}_{0i} = R(\boldsymbol{x}_i, \boldsymbol{x}_0; \boldsymbol{\theta})$$

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

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

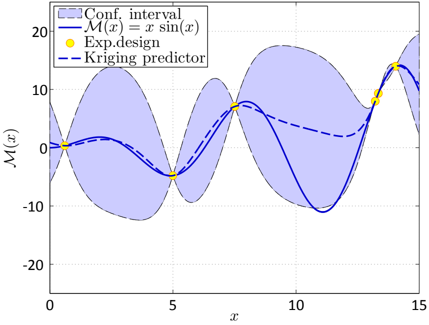
One-dimensional example

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

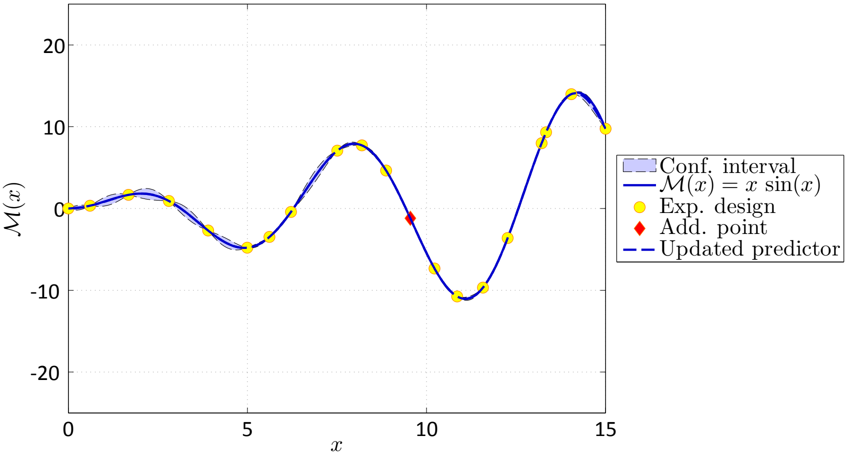


Confidence intervals

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

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

Sequential updating



Outline

Kriging for reliability analysis: basic approach

- From a given experimental design $\mathcal{X} = \left\{ \boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)} \right\}$, Kriging yields a mean predictor $\mu_{\hat{g}}(\boldsymbol{x})$ and the Kriging variance $\sigma_{\hat{g}}(\boldsymbol{x})$ of the limit state function g
- The mean predictor is substituted for the "true" limit state function, defining the surrogate failure domain

$$\mathcal{D}_f{}^0 = \left\{ \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} \; : \; \mu_{\hat{g}}(\boldsymbol{x}) \leq 0 \right\}$$

- The probability of failure is approximated by:

Kaymaz, Struc. Safety (2005)

$$P_f^0 = \mathbb{P}\left[\mu_{\hat{g}}(\boldsymbol{X}) \leq 0 \right] = \int_{\mathcal{D}_f^0} f_{\boldsymbol{X}}(\boldsymbol{x})\, d\boldsymbol{x} = \mathbb{E}\left[\mathbf{1}_{\mathcal{D}_f^0}(\boldsymbol{X}) \right]$$

- Monte Carlo simulation (resp. subset simulation, etc.) can be used on the surrogate model:

$$\widehat{P_f^0} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{1}_{\mathcal{D}_f^0}(\boldsymbol{x}_k)$$

Confidence bounds on the probability of failure

Shifted failure domains

Dubourg *et al.* , Struct. Mult. Opt. (2011)

- Let us define a confidence level $(1-\alpha)$ and $k_{1-\alpha} = \Phi^{-1}(1-\alpha/2)$, *i.e.* 1.96 if $1-\alpha = 95\%$, and:

$$\mathcal{D}_f^- = \left\{ \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} \; : \; \mu_{\hat{g}}(\boldsymbol{x}) + k_{1-\alpha}\, \sigma_{\hat{g}}(\boldsymbol{x}) \leq 0 \right\}$$

$$\mathcal{D}_f^+ = \left\{ \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} \; : \; \mu_{\hat{g}}(\boldsymbol{x}) - k_{1-\alpha}\, \sigma_{\hat{g}}(\boldsymbol{x}) \leq 0 \right\}$$

- Interpretation ($1-\alpha = 95\%$):
  - If $\boldsymbol{x} \in \mathcal{D}_f^0$ it belongs to the true failure domain with a 50% chance
  - If $\boldsymbol{x} \in \mathcal{D}_f^+$ it belongs to the true failure domain with 95% chance: conservative estimation

Bounds on the probability of failure

$$\mathcal{D}_f^- \subset \mathcal{D}_f^0 \subset \mathcal{D}_f^+ \qquad \Leftrightarrow \qquad P_f^- \leq P_f^0 \leq P_f^+$$

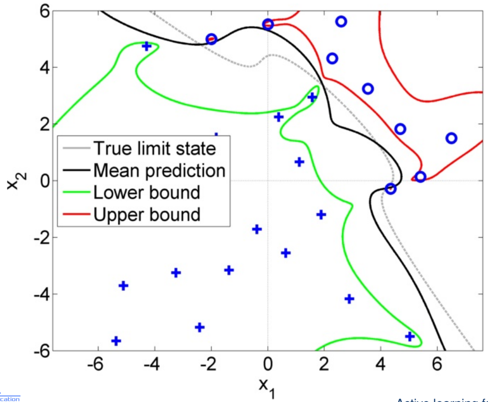
See also Picheny *et al.* (2010, 2013), Chevalier & Ginsbourger (2014), work on excursion sets by Azzimonti *et al.* (2016)

Example: hat function

Problem statement

$$g(\boldsymbol{x}) = 20 - (x_1 - x_2)^2 - 8\,(x_1 + x_2 - 4)^3$$

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



- Ref. solution:

$$P_f = 1.07 \cdot 10^{-4}$$

- Kriging surrogate:

$$P_f^- = 7.70 \cdot 10^{-6}$$

$$P_f^0 = 4.43 \cdot 10^{-4}$$

$$P_f^+ = 5.52 \cdot 10^{-2}$$

How to improve the results?

Heuristics

- The Monte Carlo estimate of P_f reads:

$$\widehat{P_f} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{1}_{\mathcal{D}_f}(\boldsymbol{x}_k) \approx \frac{1}{N} \sum_{k=1}^{N} \mathbf{1}_{\mathcal{D}_f^0}(\boldsymbol{x}_k)$$

- The Kriging-based prediction is accurate when:

$$\mathbf{1}_{\mathcal{D}_f^0}(\boldsymbol{x}_k) = \mathbf{1}_{\mathcal{D}_f}(\boldsymbol{x}_k) \quad \text{for almost all } \boldsymbol{x}_k$$

i.e. if $\mu_{\hat{g}}(\boldsymbol{x})$ is of the same sign as $g(\boldsymbol{x})$ for almost all sample points

> Ensure that the mean predictor $\mu_{\hat{g}}(\boldsymbol{x})$ classifies properly the MCS samples according to the sign of $g(\boldsymbol{x})$

Active learning reliability using a Kriging surrogate

Procedure

- Start from an initial experimental design $\mathcal{X}$ and build the initial Kriging surrogate
- At each iteration:
  - Compute an estimation of P_f and bounds 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

Early approaches

- Efficient global reliability analysis (EGRA) — Bichon *et al.* (2008)
- Active Kriging - Monte Carlo simulation (AK-MCS) — Echard *et al.* (2011)
- Both use the same reliability estimation algorithm and surrogate model
- They introduce their own learning functions for enrichment

Outline

Different enrichment criteria

Requirements

- It shall be based on the available information: $\left(\mu_{\hat{g}}(\boldsymbol{x})\,,\,\sigma_{\hat{g}}(\boldsymbol{x})\right)$
- It shall favor new points in the vicinity of the limit state surface
- If possible, it shall yield the best K points when distributed computing is available

Different enrichment criteria

- Margin indicator function — Ph.D Deheeger (2008); Bourinet *et al.* , Struc. Safety (2011)
- Margin classification function — Ph.D Dubourg (2011); Dubourg *et al.* , PEM (2013)
- Learning function U — Ph.D Échard (2012); Échard & Gayton, RESS (2011)
- Expected feasibility function — Bichon *et al.* , AIAA (2008); RESS (2011)
- Stepwise uncertainty reduction (SUR) — Bect *et al.* , Stat. Comput. (2012)
- ... **more on slide 35!**

Learning function $U(\boldsymbol{x})$

Definition

- The learning function U is defined by: Échard *et al.* (2011)

$$U(\boldsymbol{x}) = \frac{|\mu_{\hat{g}}(\boldsymbol{x})|}{\sigma_{\hat{g}}(\boldsymbol{x})}$$

Interpretation

- It describes the distance of the mean predictor $\mu_{\hat{g}}$ to zero in terms of a number of Kriging standard deviations $\sigma_{\hat{g}}$
- A small value of $U(\boldsymbol{x})$ means that:
  - $\mu_{\hat{g}}(\boldsymbol{x}) \approx 0$: $\boldsymbol{x}$ is close to the limit state surface
  - and / or $\sigma_{\hat{g}}(\boldsymbol{x}) >> 0$: the uncertainty in the prediction at point $\boldsymbol{x}$ is large
- The probability of misclassification of a point $\boldsymbol{x}$ is equal to $\Phi(-U(\boldsymbol{x}))$

Bect *et al.* , Stat. Comput. (2012)

Expected feasibility function

Heuristics

Bichon *et al.* , AIAA (2008); Bichon *et al.* , RESS (2011)

The feasibility function $FF(\boldsymbol{x})$ describes the distance from the current point $\boldsymbol{x}$ to the limit state surface

- In each point $\boldsymbol{x}$, one checks if the predictor is sufficiently close to zero: if $\left|\hat{Y}(\boldsymbol{x})\right| > \varepsilon$, $FF(\boldsymbol{x}) = 0$
- Otherwise, $FF(\boldsymbol{x})$ is the distance to the boundary of the tube $[-\varepsilon, \varepsilon]$:

$$FF(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \hat{Y}(\boldsymbol{x}) \notin [-\varepsilon, \varepsilon] \\ \varepsilon - \left|\hat{Y}(\boldsymbol{x})\right| & \text{otherwise} \end{cases} = \max\left(\varepsilon - \left|\hat{Y}(\boldsymbol{x})\right|, 0\right)$$

- As $\hat{Y}(\boldsymbol{x})$ is a Gaussian random variable, the expected feasibility function is obtained by:

$$EFF(\boldsymbol{x}) = \mathbb{E}_{GP}\left[\max\left(\varepsilon - \left|\hat{Y}(\boldsymbol{x})\right|, 0\right)\right] = \int_{\mathbb{R}} \max\left(\varepsilon - |z|, 0\right)\; f_{\hat{Y}(\boldsymbol{x})}(z)\, dz$$

where $\mathbb{E}_{GP}[\cdot]$ is the expectation with respect to the Gaussian measure associated with Kriging

Expected feasibility function

$$EFF(\boldsymbol{x}) = \int_{-\varepsilon}^{\varepsilon} (\varepsilon - |z|)\; f_{\hat{Y}(\boldsymbol{x})}(z)\, dz \qquad f_{\hat{Y}(\boldsymbol{x})}(z) = \frac{1}{\sigma_{\hat{g}}(\boldsymbol{x})}\varphi\left(\frac{z-\mu_{\hat{g}}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)$$

The expected feasibility function finally reads:

$$\begin{aligned} EFF(\boldsymbol{x}) = \mu_{\hat{g}}(\boldsymbol{x}) & \left[2\Phi\left(-\frac{\mu_{\hat{g}}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right) - \Phi\left(\frac{-\varepsilon-\mu_{\hat{g}}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right) - \Phi\left(\frac{\varepsilon-\mu_{\hat{g}}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)\right] \\ - \sigma_{\hat{g}}(\boldsymbol{x}) & \left[2\varphi\left(-\frac{\mu_{\hat{g}}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right) - \varphi\left(\frac{-\varepsilon-\mu_{\hat{g}}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right) - \varphi\left(\frac{\varepsilon-\mu_{\hat{g}}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)\right] \\ + \varepsilon & \left[\Phi\left(\frac{\varepsilon-\mu_{\hat{g}}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right) - \Phi\left(\frac{-\varepsilon-\mu_{\hat{g}}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)\right] \end{aligned}$$

NB: Usually a local value of ε is used in each point $\boldsymbol{x}$, *e.g.* $\varepsilon(\boldsymbol{x}) = 2\,\sigma_{\hat{g}}(\boldsymbol{x})$

Selection of the new ED points

Optimization of the enrichment criterion

$$\boldsymbol{x}^*_{EFF} = \arg \max_{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}} EFF(\boldsymbol{x})$$

$$\boldsymbol{x}^*_{U} = \arg \min_{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}} U(\boldsymbol{x})$$

Requires to solve a complex optimization problem in each iteration

Discrete optimization over a large Monte Carlo sample $\mathfrak{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$

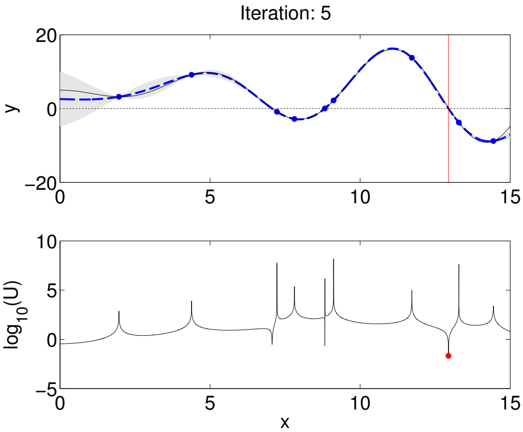
$$\boldsymbol{x}^*_{EFF} = \arg \max_{i=1,\dots,n} \{EFF(\boldsymbol{x}_1), \dots, EFF(\boldsymbol{x}_n)\}$$

$$\boldsymbol{x}^*_{U} = \arg \min_{i=1,\dots,n} \{U(\boldsymbol{x}_1), \dots, U(\boldsymbol{x}_n)\}$$

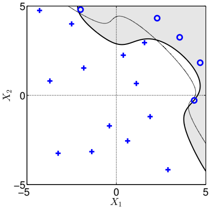
"AKMCS", Échard *et al.* (2011)

1D Application example - U function

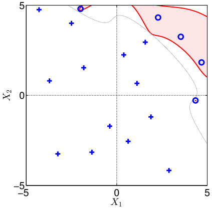
Limit state function: $g(x) = 5 - x \sin x$



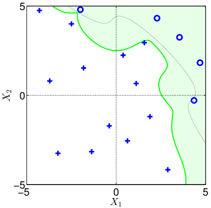
Example: hat function



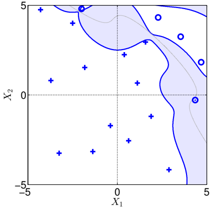
Mean $\mathcal{D}_f^0$



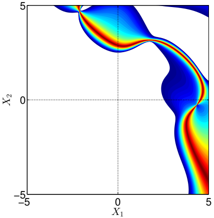
Lower bound $\mathcal{D}_f^-$



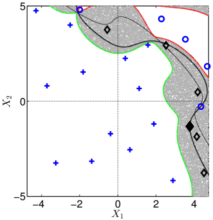
Upper bound $\mathcal{D}_f^+$



Limit-state margin



Probability of misclassification



Additional samples

PC-Kriging

Schöbi & Sudret, IJUQ (2015); Kersaudy *et al.* , J. Comp. Phys (2015), Schöbi & Sudret, ASCE/ASME JRUEng (2016)

Heuristics: Combine polynomial chaos expansions (PCE) and Kriging

- PCE approximates the global behaviour of the computational model
- Kriging allows for local interpolation and provides a local error estimate

Universal Kriging model with a sparse PC expansion as a trend

$$\mathcal{M}(\boldsymbol{x}) \approx \mathcal{M}^{(\mathsf{PCK})}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) + \sigma^2 Z(\boldsymbol{x}, \omega)$$

PC-Kriging calibration

- Sequential PC-Kriging: least-angle regression (LAR) detects a sparse basis, then PCE coefficients are calibrated together with the auto-correlation parameters
- Optimized PC-Kriging: universal Kriging models are calibrated at each step of LAR

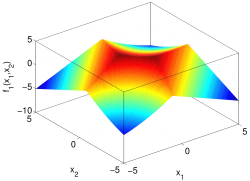
Series system

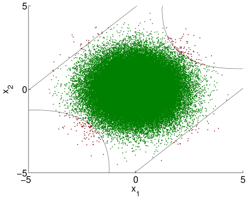
Schöbi *et al.* , ASCE J. Risk Unc. (2016)

Consider the system reliability analysis defined by:

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

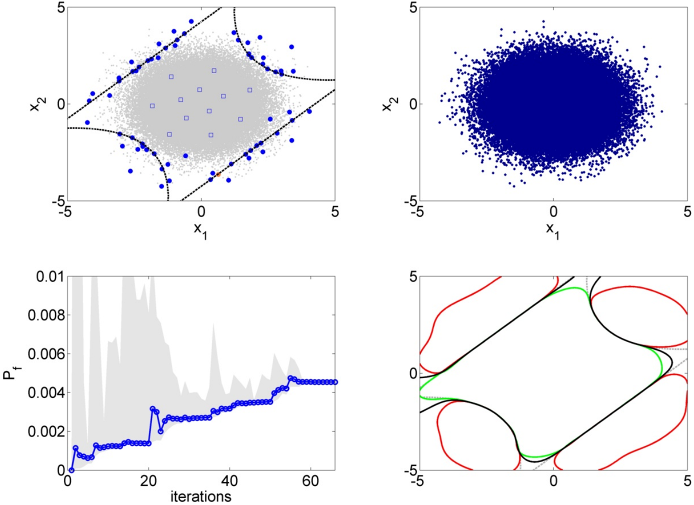
where $X_1, X_2 \sim \mathcal{N}(0, 1)$



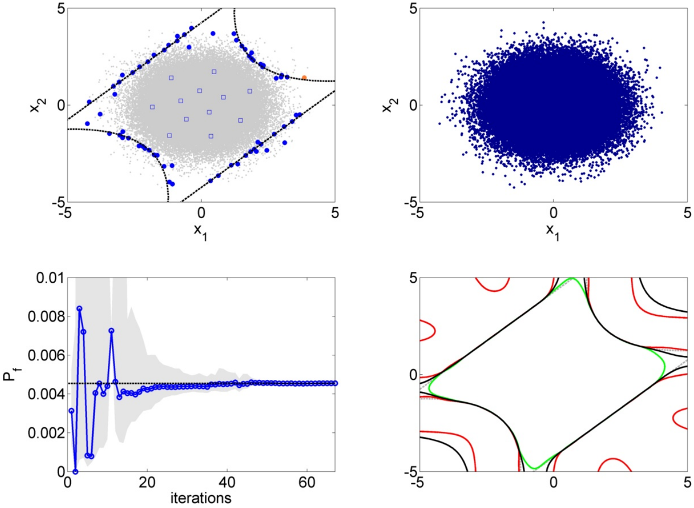


- Initial design: LHS of size 12 (transformed into the standard normal space)
- In each iteration, one point is added (maximize the probability of missclassification)
- The mean predictor $\mu_{\widehat{\mathcal{M}}}(\boldsymbol{x})$ is used, as well as the bounds $\mu_{\widehat{\mathcal{M}}}(\boldsymbol{x})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

Results with classical Kriging



Results with PC Kriging



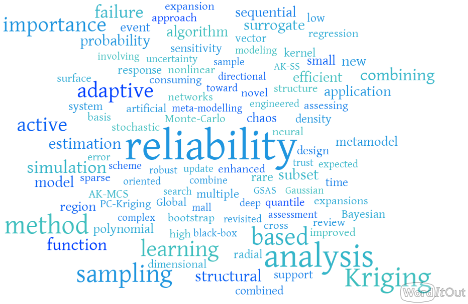
Outline

Active learning reliability methods

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

Numerous papers on active learning 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) Rous-souly 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 $[\hat{P}_f^-, \hat{P}_f^+]$

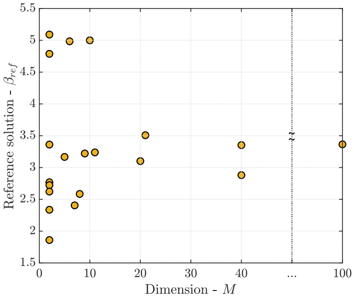
Extensive benchmark: Set-up

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

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	$P_{f,\text{ref}}$	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: $\varepsilon = |\beta - \beta_{\mathrm{ref}}| / \beta_{\mathrm{ref}}$
- Efficiency: $\Delta = \varepsilon \, N_{\mathrm{eval}} / N_{\mathrm{med}}$

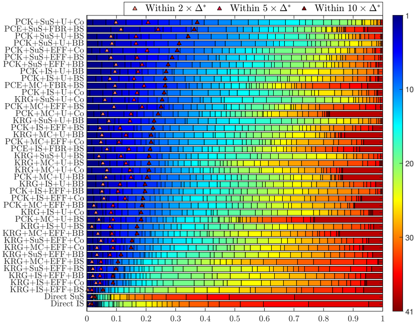
  where N_{med} is the median number of model evaluations for each problem

For each criterion:

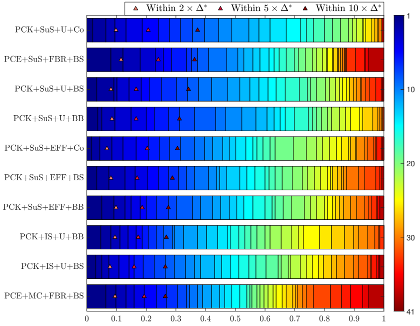
- Ranking of the strategies as a whole
- Ranking of the methods within each block
- Performance of the methods w.r.t. problem feature (dimensionality, range of P_f)

Ranking of the strategies

Percentage of times a strategy is ranked 1st, 2nd, ..., 41st w.r.t. Δ



All strategies

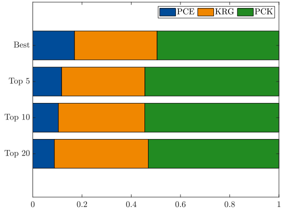


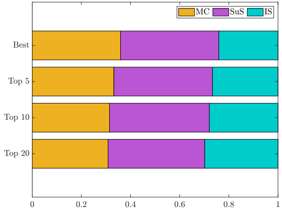
Best 10 strategies

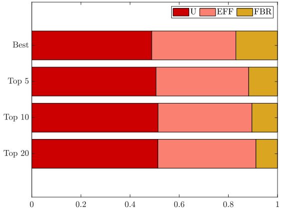
- Best approach: PC-Kriging + SuS + U + Combined stopping criterion
- Worst approaches: Direct SuS and Direct IS

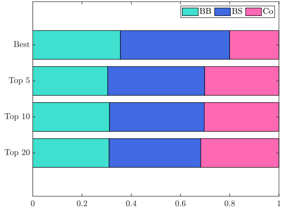
Results aggregated by method

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





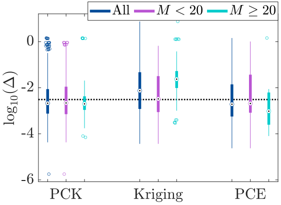


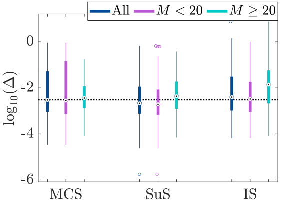


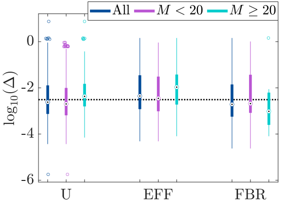
- Surrogates: PC-Kriging dominates by far
- Reliability: Slight advantage to subset simulation
- Learning function: U dominates both EFF and FBR
- Stopping criterion: Slight advantage to the stability criterion

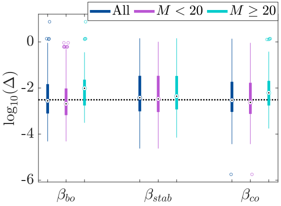
Influence of the dimension

Results aggregated w.r.t. to dimension ($M < 20$ *v.s.* $20 \leq M \leq 100$)





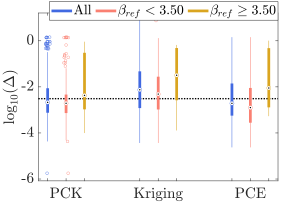


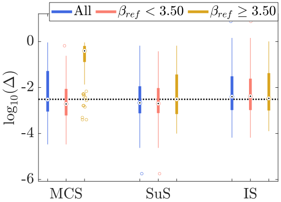


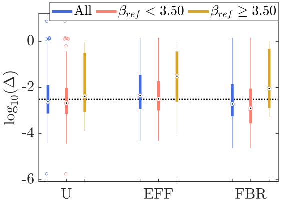
- Kriging performs worse in large dimension but not PCK and PCE
- As expected MCS is insensitive to dimension, SuS and especially IS have worse performance for $M \geq 20$
- The learning functions reproduce the performance of the surrogate models
- The β bound criterion which is based on the Kriging variance performs poorly in high dimension

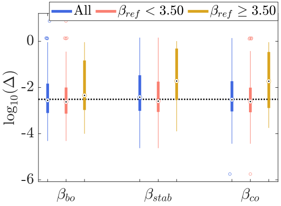
Influence of the target reliability index

Results aggregated w.r.t. to dimension ($\beta < 3.5$ vs. $\beta \geq 3.5$)





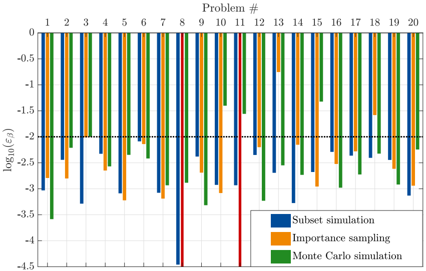




- PC-Kriging is not so much sensitive to β_{ref} while Kriging and PCE perform worse for larger values of β_{ref}
- SuS and IS are not so much affected by β_{ref} compared to MCS
- EFF performs worse than U for larger values of β_{ref}
- β-stability and combined criteria have a noticeably poorer performance for larger values of β_{ref}

Results without surrogates

Relative error for overkill reliability methods without surrogates



- Red lines correspond to non-convergence
- The problems that could not be solved with ALR were not solved by a direct approach (*i.e.* , without surrogate) either

In most cases, the surrogate model was not the cause of failure of the ALR strategy, but rather the reliability estimation algorithm

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}

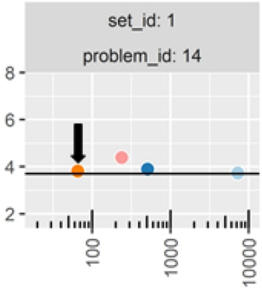
Main take-away

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)



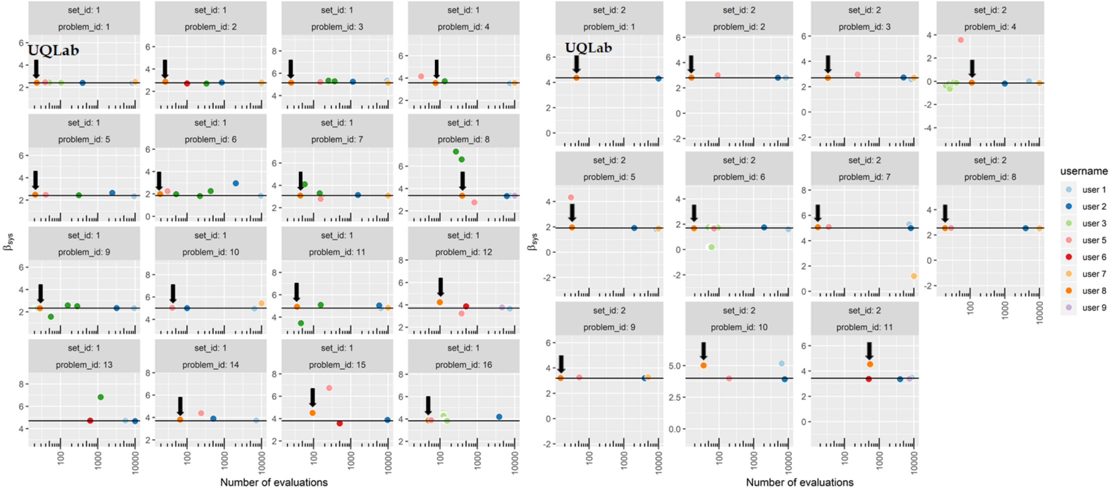
Summary plot (TNO)

- 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

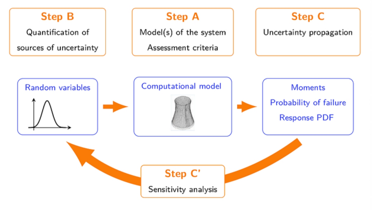
- Estimating low probabilities of failure in high-dimensional problems requires more refined algorithms than plain MCS
- Recent research on surrogate models (*e.g.* Kriging and polynomial chaos expansions) and active learning has brought new extremely efficient algorithms
- Accurate estimations of P_f's (not of β !) are obtained with $\mathcal{O}(100)$ runs of the computer code independently 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**
- More than 3,600 registered users
- 1,400+ active users from 92 countries

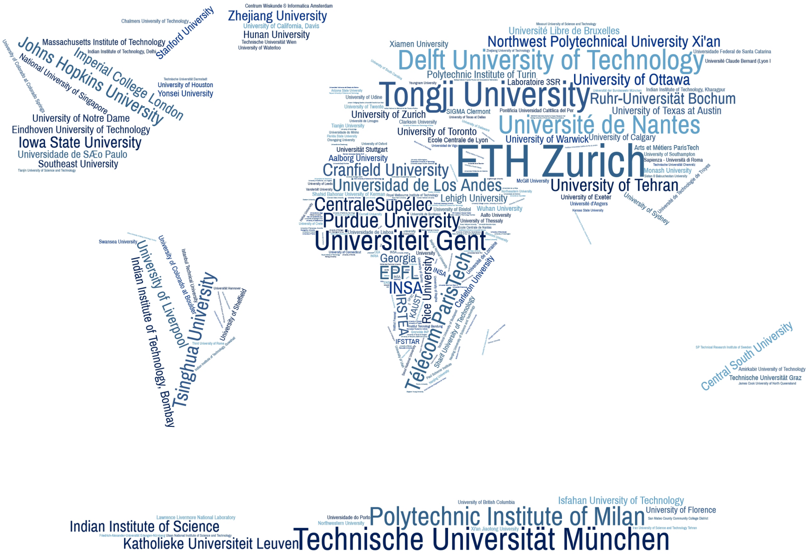
http://www.uqlab.com

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

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

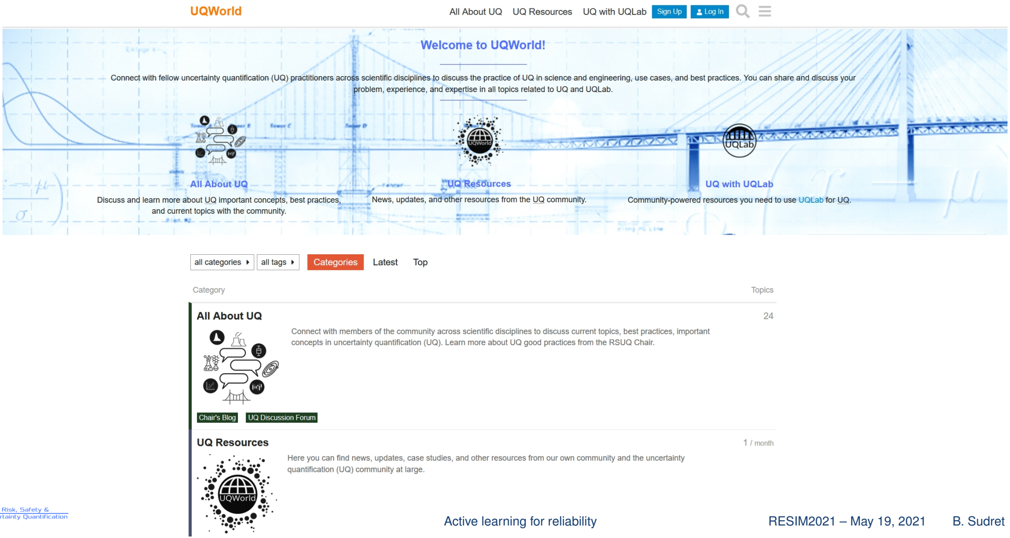
Country	# Users
United States	582
China	500
France	339
Switzerland	285
Germany	270
United Kingdom	157
Italy	145
Brazil	126
India	120
Canada	87

As of May 15, 2021


Technische Universität München
Polytechnic Institute of Milan
ETH Zurich
Tongji University
Delft University of Technology
Université de Nantes
Universiteit Gent
Purdue University
CentraleSupélec
Universidad de Los Andes
Cranfield University
University of Tehran
Ruhr-Universität Bochum
University of Ottawa
Northwest Polytechnical University Xi'an
Université Libre de Bruxelles
Polytechnic Institute of Turin
University of Zurich
University of Toronto
University of Warwick
University of Calgary
University of Texas at Austin
Lehigh University
Wuhan University
Aalborg University
Universität Stuttgart
Ecole Centrale de Lyon
Xiamen University
Zhejiang University
University of California, Davis
Hunan University
Technische Universität Wien
University of Waterloo
Centrum Wiskunde & Informatica Amsterdam
Johns Hopkins University
Imperial College London
Massachusetts Institute of Technology
Stanford University
National University of Singapore
University of Houston
Yonsei University
University of Notre Dame
Eindhoven University of Technology
Iowa State University
Universidade de SÆo Paulo
Southeast University
Chalmers University of Technology
Indian Institute of Technology, Delhi
University of Liverpool
Indian Institute of Technology, Bombay
Tsinghua University
University of Colorado at Boulder
Istanbul Technical University
University of Sheffield
Swansea University
Télécom ParisTech
Georgia
EPFL
INSA
IRSTEA
IFSTTAR
KAUST
Rice University
Carleton University
Arts et Métiers ParisTech
Monash University
University of Exeter
University of Sydney
McGill University
SIGMA Clermont
Laboratoire 3SR
Central South University
Technische Universität Graz
Amirkabir University of Technology
Isfahan University of Technology
University of Florence
Indian Institute of Science
Katholieke Universiteit Leuven
Lawrence Livermore National Laboratory
Universidade do Porto
Northwestern University
University of British Columbia
Xi'an Jiaotong University

UQWorld: the community of UQ

https://uqworld.org/



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