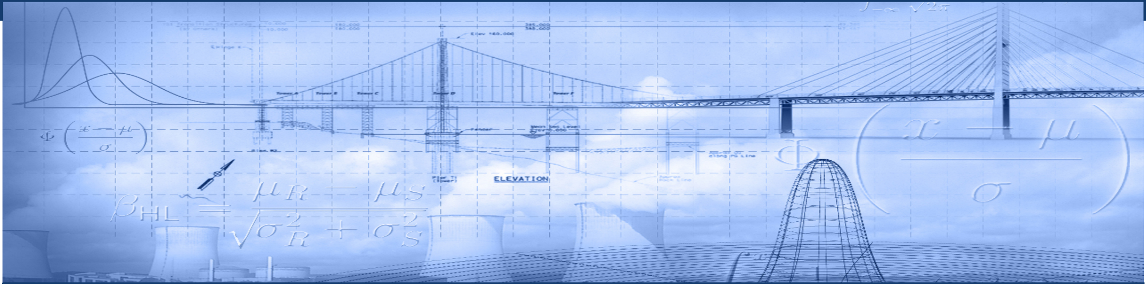
Surrogate models for uncertainty quantification and structural reliability

Presentation

Author(s):
Moustapha, Maliki; Sudret, Bruno

Publication date:
2022-02-16

Permanent link:
https://doi.org/10.3929/ethz-b-000532921



Surrogate models for uncertainty quantification and structural reliability

NARSIS Final Workshop - 16/02/2022 - Online

Maliki Moustapha, Bruno Sudret

Chair of Risk, Safety and Uncertainty Quantification | ETH Zürich

How to cite?

This presentation is an invited lecture given at the Final Workshop of the EU Project NARSIS on February 16, 2022.

How to cite

M. Moustapha and B. Sudret, *Surrogate models for uuncertainty quantification and structural reliability*, Final Workshop - EU Project NARSIS, online, February 16, 2022.

Computational models in engineering

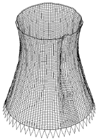
Complex engineering systems are designed and assessed using computational models, a.k.a simulators

A computational model combines:

- A mathematical description of the physical phenomena (governing equations), *e.g.* mechanics, electromagnetism, fluid dynamics, etc.

$$\operatorname{div} \boldsymbol{\sigma} + \boldsymbol{f} = \boldsymbol{0}$$
$$\boldsymbol{\sigma} = \mathbf{D} \cdot \boldsymbol{\varepsilon}$$
$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + {}^{\mathsf{T}}\nabla \boldsymbol{u}\right)$$

- Discretization techniques which transform continuous equations into linear algebra problems

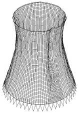


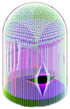
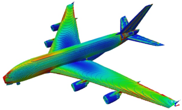
- Algorithms to solve the discretized equations

Computational models in engineering

Computational models are used:

- 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



Computational models: the abstract viewpoint

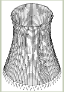
A computational model may be seen as a black box program that computes quantities of interest (QoI) (a.k.a. model responses) as a function of input parameters

Vector of input parameters $\boldsymbol{x} \in \mathbb{R}^M$

Computational model $\mathcal{M}$

Model response $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}) \in \mathbb{R}^Q$

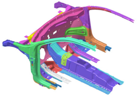
- Geometry
- Material properties
- Loading



- Analytical formula
- Finite element model
- Comput. workflow

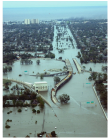
- Displacements
- Strains, stresses
- Temperature, etc.

Real world is uncertain



- Differences between the designed and the real system:
  - Dimensions (tolerances in manufacturing)
  - Material properties (*e.g.* variability of the stiffness or resistance)

- Unforecast exposures: exceptional service loads, natural hazards (earthquakes, floods, landslides), climate loads (hurricanes, snow storms, etc.), accidental human actions (explosions, fire, etc.)



Outline

Introduction

Uncertainty quantification: why surrogate models?

Polynomial chaos expansions

Structural reliability analysis

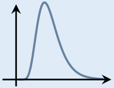
Global framework for uncertainty quantification

Step B
Quantification of sources of uncertainty

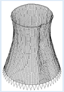
Step A
Model(s) of the system
Assessment criteria

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 B: Quantification of the sources of uncertainty

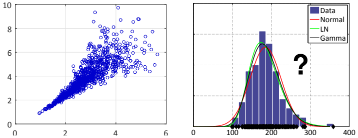
Goal: represent the uncertain parameters based on the available data and information

Probabilistic model $f_{\boldsymbol{X}}$

Experimental data is available

- What is the distribution of each parameter ?
- What is the dependence structure ?

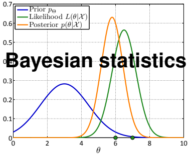
Copula theory



No data is available: expert judgment

- Engineering knowledge (*e.g.* reasonable bounds and uniform distributions)
- Statistical arguments and literature (*e.g.* extreme value distributions for climatic events)

Scarce data + expert information

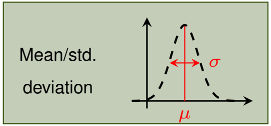


Step C: uncertainty propagation

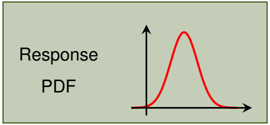
Goal: estimate the uncertainty / variability of the quantities of interest (QoI) $Y = \mathcal{M}(\boldsymbol{X})$ due to the input uncertainty $f_{\boldsymbol{X}}$

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

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

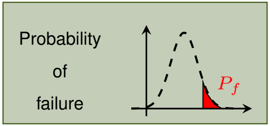


- Distribution of the QoI



- Probability of exceeding an admissible threshold y_{adm}

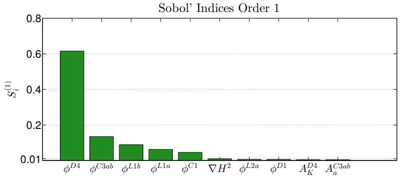
$$P_f = \mathbb{P}\left(Y \geq y_{adm}\right)$$



Step C': sensitivity analysis

Goal: determine what are the input parameters (or combinations thereof) whose uncertainty explains the variability of the quantities of interest

- Screening: detect input parameters whose uncertainty has no impact on the output variability
- Feature setting: detect input parameters which allow one to best decrease the output variability when set to a deterministic value
- Exploration: detect interactions between parameters, *i.e.* joint effects not detected when varying parameters one-at-a-time

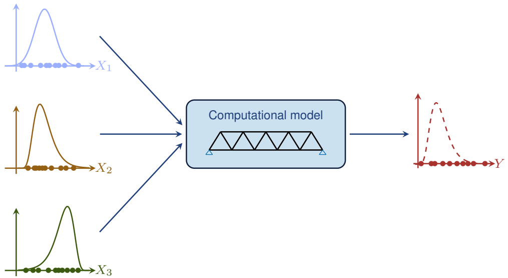


Variance decomposition (Sobol' indices)

Uncertainty propagation using Monte Carlo simulation

Principle: Generate virtual prototypes of the system using random numbers

- A sample set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is drawn according to the input distribution $f_{\boldsymbol{X}}$
- For each sample the quantity of interest (resp. performance criterion) is evaluated, say $\mathcal{Y} = \{\mathcal{M}(\boldsymbol{x}_1), \ldots, \mathcal{M}(\boldsymbol{x}_n)\}$
- The set of model outputs is used for moments-, distribution- or reliability analysis



Advantages/Drawbacks of Monte Carlo simulation

Advantages

- Universal method: only rely upon sampling random numbers and running repeatedly the computational model
- Sound statistical foundations: convergence when $n \to \infty$
- Suited to High Performance Computing: "embarrassingly parallel"

Drawbacks

- Statistical uncertainty: results are not exactly reproducible when a new analysis is carried out (handled by computing confidence intervals)
- Low efficiency: convergence rate $\propto n^{-1/2}$

Monte Carlo for reliability analysis

To compute $P_f = 10^{-k}$ with an accuracy of $\pm$10% (coef. of variation of 5%), $4 \cdot 10^{k+2}$ runs are required

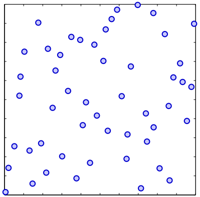
Surrogate models for uncertainty quantification

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	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{m} a_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\boldsymbol{a},\ b$

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

Introduction

Uncertainty quantification: why surrogate models?

Polynomial chaos expansions

- PCE basis
- Computing the coefficients
- Sparse PCE
- Post-processing
- Extensions

Structural reliability analysis

Polynomial chaos expansions in a nutshell

Ghanem & Spanos (1991; 2003); Xiu & Karniadakis (2002); Soize & Ghanem (2004)

- Consider the input random vector $\boldsymbol{X}$ (dim $\boldsymbol{X} = M$) with given probability density function (PDF) $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{M} f_{X_i}(x_i)$
- Assuming that the random output $Y = \mathcal{M}(\boldsymbol{X})$ has finite variance, it can be cast as the following polynomial chaos expansion:

$$Y = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} y_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$$

where :

  - $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$: basis functions
  - $y_{\boldsymbol{\alpha}}$: coefficients to be computed (coordinates)

- The PCE basis $\left\{\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}),\ \boldsymbol{\alpha} \in \mathbb{N}^M\right\}$ is made of multivariate orthonormal polynomials

Multivariate polynomial basis

Univariate polynomials

- For each input variable X_i, univariate orthogonal polynomials $\{P_k^{(i)},\ k \in \mathbb{N}\}$ are built:

$$\left\langle P_j^{(i)}, P_k^{(i)} \right\rangle = \int P_j^{(i)}(u)\ P_k^{(i)}(u)\ f_{X_i}(u)\, du = \gamma_j^{(i)}\ \delta_{jk}$$

e.g. , Legendre polynomials if $X_i \sim \mathcal{U}(-1, 1)$, Hermite polynomials if $X_i \sim \mathcal{N}(0, 1)$

- Normalization: $\Psi_j^{(i)} = P_j^{(i)} / \sqrt{\gamma_j^{(i)}} \quad i = 1, \ldots, M, \quad j \in \mathbb{N}$

Tensor product construction

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \prod_{i=1}^{M} \Psi_{\alpha_i}^{(i)}(x_i) \qquad \mathbb{E}\left[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})\Psi_{\boldsymbol{\beta}}(\boldsymbol{X})\right] = \delta_{\boldsymbol{\alpha\beta}}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ are multi-indices (partial degree in each dimension)

Outline

Introduction

Uncertainty quantification: why surrogate models?

Polynomial chaos expansions

- PCE basis
- Computing the coefficients
- Sparse PCE
- Post-processing
- Extensions

Structural reliability analysis

Computing the coefficients by least-square minimization

Isukapalli (1999); Berveiller, Sudret & Lemaire (2006)

Principle

The exact (infinite) series expansion is considered as the sum of a truncated series and a residual:

$$Y = \mathcal{M}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) + \varepsilon_P \equiv \mathbf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{X}) + \varepsilon_P(\boldsymbol{X})$$

where : $\mathbf{Y} = \{y_{\boldsymbol{\alpha}},\ \boldsymbol{\alpha} \in \mathcal{A}\} \equiv \{y_0, \ldots, y_{P-1}\}$ (P unknown coefficients)

$$\boldsymbol{\Psi}(\boldsymbol{x}) = \{\Psi_0(\boldsymbol{x}), \ldots, \Psi_{P-1}(\boldsymbol{x})\}$$

Least-square minimization

The unknown coefficients are estimated by minimizing the mean square residual error:

$$\boxed{\hat{\mathbf{Y}} = \arg\min \mathbb{E}\left[\left(\mathbf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{X}) - \mathcal{M}(\boldsymbol{X})\right)^2\right]}$$

Discrete (ordinary) least-square minimization

An estimate of the mean square error (sample average) is minimized:

$$\hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}\in\mathbb{R}^P} \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{Y}^\mathsf{T}\boldsymbol{\Psi}(\boldsymbol{x}^{(i)}) - \mathcal{M}(\boldsymbol{x}^{(i)})\right)^2$$

Procedure

- Select a truncation scheme, *e.g.* $\mathcal{A}^{M,p} = \left\{\boldsymbol{\alpha} \in \mathbb{N}^M \ : \ |\boldsymbol{\alpha}|_1 \leq p\right\}$
- Select an experimental design and evaluate the model response

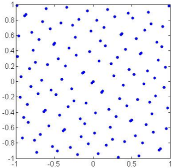
$$\mathbf{M} = \left\{\mathcal{M}(\boldsymbol{x}^{(1)}), \dots, \mathcal{M}(\boldsymbol{x}^{(n)})\right\}^\mathsf{T}$$

- Compute the experimental matrix

$$\mathbf{A}_{ij} = \Psi_j\left(\boldsymbol{x}^{(i)}\right) \quad i = 1, \dots, n\ ;\ j = 0, \dots, P-1$$

- Solve the resulting linear system

$$\hat{\mathbf{Y}} = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{M}$$



Simple is beautiful !

Error estimators

- In least-squares analysis, the generalization error is defined as:

$$E_{gen} = \mathbb{E}\left[\left(\mathcal{M}(\boldsymbol{X}) - \mathcal{M}^{\mathsf{PC}}(\boldsymbol{X})\right)^2\right] \qquad \mathcal{M}^{\mathsf{PC}}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} y_{\boldsymbol{\alpha}}\,\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$$

- The empirical error based on the experimental design $\mathcal{X}$ is a poor estimator in case of overfitting

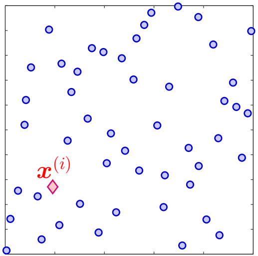
$$E_{emp} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathcal{M}(\boldsymbol{x}^{(i)}) - \mathcal{M}^{\mathsf{PC}}(\boldsymbol{x}^{(i)})\right)^2$$

- The coefficient of determination R^2 is often used as an error estimator:

$$R^2 = 1 - \frac{E_{emp}}{\mathrm{Var}\left[\mathcal{Y}\right]} \qquad \mathrm{Var}\left[\mathcal{Y}\right] = \frac{1}{n}(\mathcal{M}(\boldsymbol{x}^{(i)}) - \bar{\mathcal{Y}})^2$$

R^2 is a poor estimator of the accuracy of the PCE when there is overfitting

Leave-one-out cross validation



- An experimental design $\mathcal{X} = \{\boldsymbol{x}^{(j)},\ j = 1, \ldots, n\}$ is selected
- Polynomial chaos expansions are built using all points but one, *i.e.* based on $\mathcal{X} \backslash \boldsymbol{x}^{(i)} = \{\boldsymbol{x}^{(j)},\ j = 1, \ldots, n,\ j \neq i\}$
- Leave-one-out error (PRESS)

$$E_{LOO} \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \left(\mathcal{M}(\boldsymbol{x}^{(i)}) - \mathcal{M}^{PC\backslash i}(\boldsymbol{x}^{(i)}) \right)^2$$

- Analytical derivation from a single PC analysis

$$E_{LOO} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\mathcal{M}(\boldsymbol{x}^{(i)}) - \mathcal{M}^{PC}(\boldsymbol{x}^{(i)})}{1 - h_i} \right)^2$$

where h_i is the i-th diagonal term of matrix $\mathbf{A}(\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}$

Outline

Introduction

Uncertainty quantification: why surrogate models?

Polynomial chaos expansions

- PCE basis
- Computing the coefficients
- Sparse PCE
- Post-processing
- Extensions

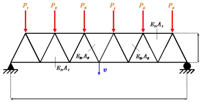
Structural reliability analysis

Curse of dimensionality

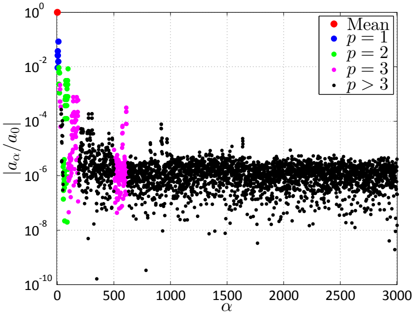
- The cardinality of the truncation scheme $\mathcal{A}^{M,p}$ is $P = \dfrac{(M+p)!}{M!\,p!}$
- Typical computational requirements: $n = OSR \cdot P$ where the oversampling rate is $OSR = 2 - 3$

However ... most coefficients are close to zero !

Example



- Elastic truss structure with $M = 10$ independent input variables
- PCE of degree $p = 5$ ($P = 3,003$ coefficients)



Hyperbolic truncation sets

Sparsity-of-effects principle

Blatman & Sudret, Prob. Eng. Mech (2010); J. Comp. Phys (2011)

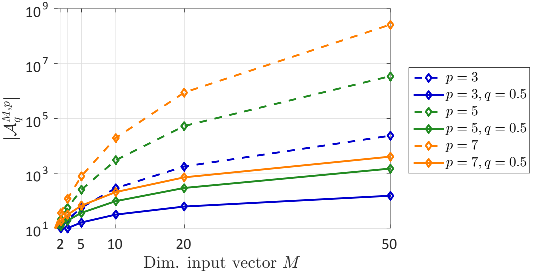
In most engineering problems, only low-order interactions between the input variables are relevant

- $q-$norm of a multi-index $\boldsymbol{\alpha}$:

$$||\boldsymbol{\alpha}||_q \equiv \left(\sum_{i=1}^{M} \alpha_i^q \right)^{1/q} \quad , \quad 0 < q \leq 1$$

- Hyperbolic truncation sets:

$$\mathcal{A}_q^{M,p} = \{\boldsymbol{\alpha} \in \mathbb{N}^M \ : \ ||\boldsymbol{\alpha}||_q \leq p\}$$



Compressive sensing approaches

Blatman & Sudret (2011); Doostan & Owhadi (2011); Sargsyan *et al.* (2014); Jakeman *et al.* (2015)

- Sparsity in the solution can be induced by ℓ_1-regularization:

$$\boldsymbol{y}_{\boldsymbol{\alpha}} = \arg\min \frac{1}{n} \sum_{i=1}^{n} \left(\mathsf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{x}^{(i)}) - \mathcal{M}(\boldsymbol{x}^{(i)}) \right)^2 + \lambda \parallel \boldsymbol{y}_{\boldsymbol{\alpha}} \parallel_1$$

- Different algorithms: LASSO, orthogonal matching pursuit, Bayesian compressive sensing

Least Angle Regression

Efron *et al.* (2004), Blatman & Sudret (2011)

- Least Angle Regression (LAR) solves the LASSO problem for different values of the penalty constant in a single run without matrix inversion
- Leave-one-out cross validation error allows one to select the best model

Outline

Introduction

Uncertainty quantification: why surrogate models?

Polynomial chaos expansions

- PCE basis
- Computing the coefficients
- Sparse PCE
- Post-processing
- Extensions

Structural reliability analysis

Post-processing sparse PC expansions

Statistical moments

- Due to the orthogonality of the basis functions ($\mathbb{E}\left[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})\Psi_{\boldsymbol{\beta}}(\boldsymbol{X})\right] = \delta_{\boldsymbol{\alpha}\boldsymbol{\beta}}$) and using $\mathbb{E}\left[\Psi_{\boldsymbol{\alpha}\neq\boldsymbol{0}}\right] = 0$ the statistical moments read:

$$\text{Mean:} \qquad \hat{\mu}_Y = y_0$$

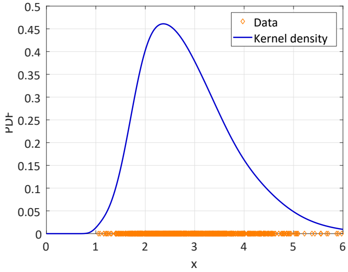
$$\text{Variance:} \qquad \hat{\sigma}_Y^2 = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\boldsymbol{0}} y_{\boldsymbol{\alpha}}^2$$

Distribution of the QoI

- The PCE can be used as a response surface for sampling:

$$\mathfrak{y}_j = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} y_{\boldsymbol{\alpha}}\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}_j) \quad j = 1, \ldots, n_{big}$$

- The PDF of the response is estimated by histograms or kernel smoothing



Sensitivity analysis

Goal

Sobol' (1993); Saltelli *et al.* (2008)

Global sensitivity analysis aims at quantifying which input parameter(s) (or combinations thereof) influence the most the response variability (variance decomposition)

Hoeffding-Sobol' decomposition

$(\boldsymbol{X} \sim \mathcal{U}([0,1]^M))$

$$\mathcal{M}(\boldsymbol{x}) = \mathcal{M}_0 + \sum_{i=1}^{M} \mathcal{M}_i(x_i) + \sum_{1 \le i < j \le M} \mathcal{M}_{ij}(x_i, x_j) + \cdots + \mathcal{M}_{12\dots M}(\boldsymbol{x})$$

$$= \mathcal{M}_0 + \sum_{\mathbf{u} \subset \{1, \dots, M\}} \mathcal{M}_{\mathbf{u}}(\boldsymbol{x}_{\mathbf{u}}) \qquad (\boldsymbol{x}_{\mathbf{u}} \stackrel{\text{def}}{=} \{x_{i_1}, \dots, x_{i_s}\})$$

- The summands satisfy the orthogonality condition:

$$\int_{[0,1]^M} \mathcal{M}_{\mathbf{u}}(\boldsymbol{x}_{\mathbf{u}})\, \mathcal{M}_{\mathbf{v}}(\boldsymbol{x}_{\mathbf{v}})\, d\boldsymbol{x} = 0 \qquad \forall\, \mathbf{u} \neq \mathbf{v}$$

Sobol' indices

Total variance

$$D \equiv \mathrm{Var}\left[\mathcal{M}(\boldsymbol{X})\right] = \sum_{\mathbf{u} \subset \{1, \dots, M\}} \mathrm{Var}\left[\mathcal{M}_{\mathbf{u}}(\boldsymbol{X}_{\mathbf{u}})\right]$$

- Sobol' indices:

$$S_{\mathbf{u}} \overset{\text{def}}{=} \frac{\mathrm{Var}\left[\mathcal{M}_{\mathbf{u}}(\boldsymbol{X}_{\mathbf{u}})\right]}{D}$$

- First-order Sobol' indices:

$$S_i = \frac{D_i}{D} = \frac{\mathrm{Var}\left[\mathcal{M}_i(X_i)\right]}{D}$$

  Quantify the additive effect of each input parameter separately

- Total Sobol' indices:

$$S_i^T \overset{\text{def}}{=} \sum_{\mathbf{u} \supset i} S_{\mathbf{u}}$$

  Quantify the total effect of X_i, including interactions with the other variables.

Link with PC expansions

Sobol decomposition of a PC expansion

Sudret, CSM (2006); RESS (2008)

Obtained by reordering the terms of the (truncated) PC expansion $\mathcal{M}^{\text{PC}}(\boldsymbol{X}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$

Interaction sets

For a given $\mathbf{u} \stackrel{\text{def}}{=} \{i_1, \ldots, i_s\}$: $\quad \mathcal{A}_{\mathbf{u}} = \{\boldsymbol{\alpha} \in \mathcal{A} \, : \, k \in \mathbf{u} \Leftrightarrow \alpha_k \neq 0\}$

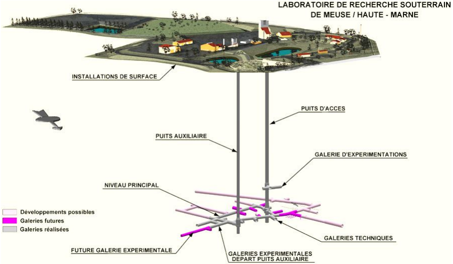
$$\mathcal{M}^{\text{PC}}(\boldsymbol{x}) = \mathcal{M}_0 + \sum_{\mathbf{u} \subset \{1, \ldots, M\}} \mathcal{M}_{\mathbf{u}}(\boldsymbol{x}_{\mathbf{u}}) \quad \text{where} \quad \mathcal{M}_{\mathbf{u}}(\boldsymbol{x}_{\mathbf{u}}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\mathbf{u}}} y_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$$

PC-based Sobol' indices

$$S_{\mathbf{u}} = D_{\mathbf{u}}/D = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\mathbf{u}}} y_{\boldsymbol{\alpha}}^2 / \sum_{\boldsymbol{\alpha} \in \mathcal{A} \setminus \mathbf{0}} y_{\boldsymbol{\alpha}}^2$$

The Sobol' indices are obtained analytically, at any order from the coefficients of the PC expansion

Example: sensitivity analysis in hydrogeology



Source: http://www.futura-sciences.com/

Source: http://lexpansion.lexpress.fr/

- When assessing a nuclear waste repository, the Mean Lifetime Expectancy MLE(x) is the time required for a molecule of water at point x to get out of the boundaries of the system

- Computational models have numerous input parameters (in each geological layer) that are difficult to measure, and that show scattering

Geological model

Deman, Konakli, Sudret, Kerrou, Perrochet & Benabderrahmane, Reliab. Eng. Sys. Safety (2016)

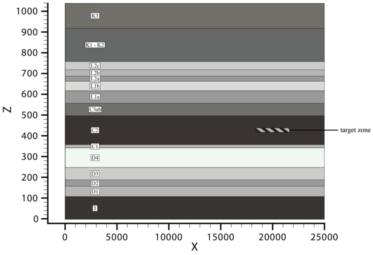
- Two-dimensional idealized model of the Paris Basin (25 km long / 1,040 m depth) with 5×5 m mesh (10^6 elements)
- Steady-state flow simulation with Dirichlet boundary conditions:

$$\nabla \cdot (\mathbf{K} \cdot \nabla H) = 0$$

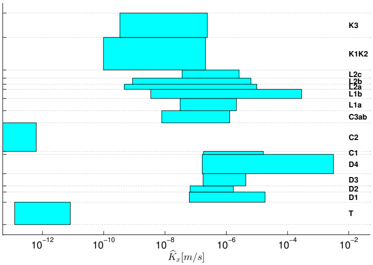
- 15 homogeneous layers with uncertainties in:
  - Porosity (resp. hydraulic conductivity)
  - Anisotropy of the layer properties (inc. dispersivity)
  - Boundary conditions (hydraulic gradients)

78 input parameters

Sensitivity analysis



Geometry of the layers



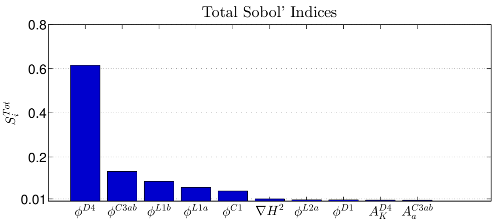
Conductivity of the layers

Question

What are the parameters (out of 78) whose uncertainty drives the uncertainty of the prediction of the mean life-time expectancy?

Sensitivity analysis: results

Technique: Sobol'indices computed from polynomial chaos expansions



Parameter	$\sum_j S_j$
ϕ (resp. K_x)	0.8664
A_K	0.0088
θ	0.0029
α_L	0.0076
A_α	0.0000
∇H	0.0057

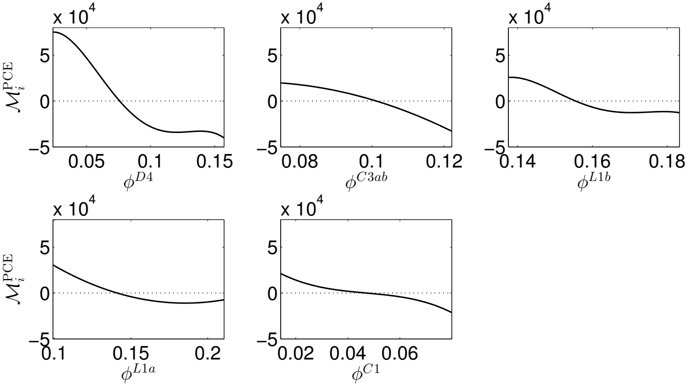
Conclusions

- Only 200 model runs allow one to detect the 10 important parameters out of 78
- Uncertainty in the porosity/conductivity of 5 layers explain 86% of the variability
- Small interactions between parameters detected

Bonus: univariate effects

The univariate effects of each variable are obtained as a straightforward post-processing of the PCE

$$\mathcal{M}_i(x_i) \overset{\text{def}}{=} \mathbb{E}\left[\mathcal{M}(\boldsymbol{X})|X_i = x_i\right], \; i = 1, \, \ldots \, , M$$



Outline

Introduction

Uncertainty quantification: why surrogate models?

Polynomial chaos expansions
- PCE basis
- Computing the coefficients
- Sparse PCE
- Post-processing
- Extensions

Structural reliability analysis

Polynomial chaos expansions in structural dynamics (1/2)

Spiridonakos et al. (2015); Mai, Spiridonakos, Chatzi & Sudret, IJUQ (2016); Mai & Sudret, SIAM JUQ (2017)

Premise

- For dynamical systems, the complexity of the map $\boldsymbol{\xi} \mapsto \mathcal{M}(\boldsymbol{\xi}, t)$ increases with time.
- Time-frozen PCE does not work beyond first time instants

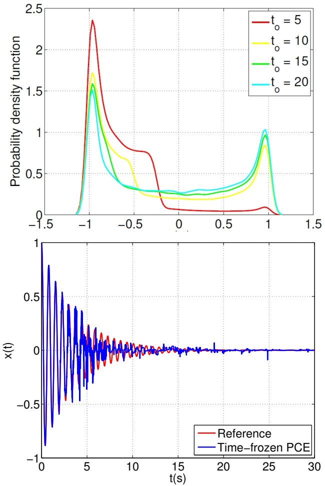
PC-NARX

- Use of non linear autoregressive with exogenous input models (NARX) to capture the dynamics:

$$y(t) = \mathcal{F}\left(x(t), \ldots, x(t-n_x), y(t-1), \ldots, y(t-n_y)\right) + \epsilon_t \equiv \mathcal{F}\left(\boldsymbol{z}(t)\right)$$

- A linear-in-parameters models is often used

$$y(t) = \sum_{i=1}^{n_g} \vartheta_i \, g_i(\boldsymbol{z}(t)) + \epsilon(t)$$



Polynomial chaos expansions in structural dynamics (2/2)

Spiridonakos et al. (2015); Mai, Spiridonakos, Chatzi & Sudret, IJUQ (2016); Mai & Sudret, SIAM JUQ (2017)

PC-NARX

- Use of non linear autoregressive with exogenous input models (NARX) to capture the dynamics:

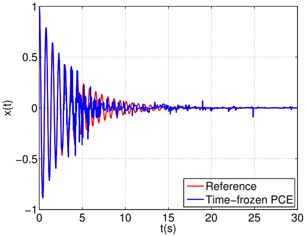
$$y(t) = \mathcal{F}\left(x(t), \ldots, x(t-n_x), y(t-1), \ldots, y(t-n_y)\right) + \epsilon_t \equiv \mathcal{F}\left(\boldsymbol{z}(t)\right) -$$

- A linear-in-parameters models is often used

$$y(t, \boldsymbol{\xi}) = \sum_{i=1}^{n_g} \vartheta_i(\boldsymbol{\xi})\, g_i(\boldsymbol{z}(t)) + \epsilon(t, \boldsymbol{\xi})$$

- Expand the NARX coefficients of different random trajectories onto a PCE basis

$$y(t, \boldsymbol{\xi}) = \sum_{i=1}^{n_g} \sum_{\boldsymbol{\alpha} \in \mathcal{A}_i} \vartheta_{i,\boldsymbol{\alpha}}\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\, g_i(\boldsymbol{z}(t)) + \epsilon(t, \boldsymbol{\xi})$$



Earthquake engineering – Bouc-Wen oscillator

Governing equations

Kafali & Grigoriu (2007), Spiridonakos & Chatzi (2015)

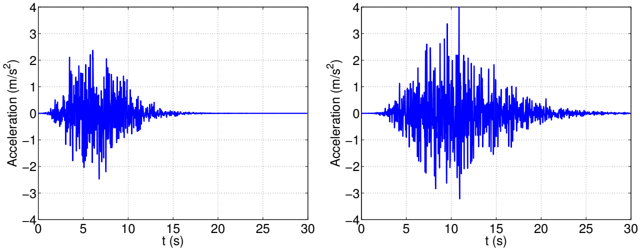
$$\ddot{y}(t) + 2\,\zeta\,\omega\,\dot{y}(t) + \omega^2(\rho\,y(t) + (1-\rho)\,z(t)) = -x(t),$$
$$\dot{z}(t) = \gamma\dot{y}(t) - \alpha\,|\dot{y}(t)|\,|z(t)|^{n-1}\,z(t) - \beta\,\dot{y}(t)\,|z(t)|^{n}\,,$$

Excitation

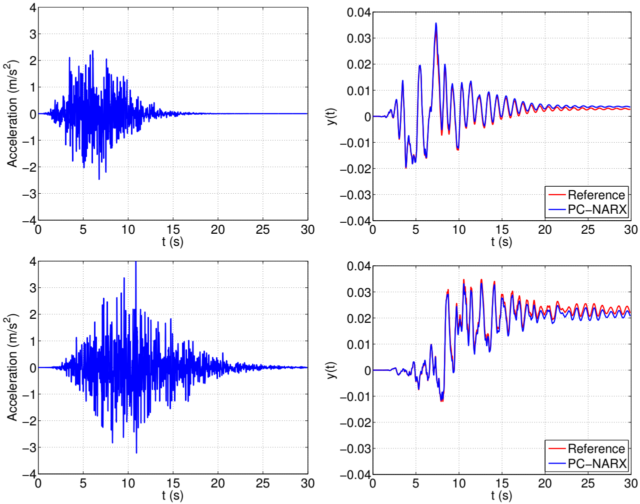
$x(t)$ is generated by a probabilistic ground motion model

Rezaeian & Der Kiureghian (2010)

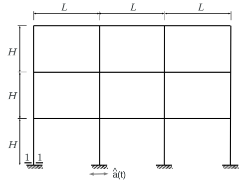
$$x(t) = q(t, \boldsymbol{\alpha}) \sum_{i=1}^{n} s_i\,(t, \boldsymbol{\lambda}(t_i))\;U_i$$



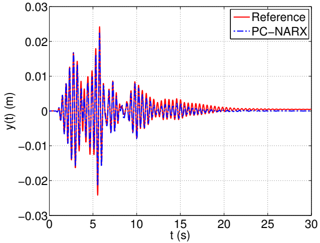
Bouc-Wen model: prediction

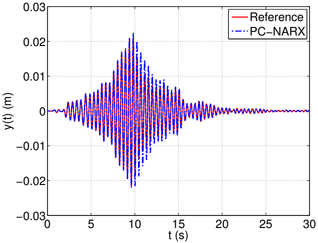


Earthquake engineering – frame structure



- 2D steel frame with uncertain properties submitted to synthetic ground motions
- Experimental design of size 300



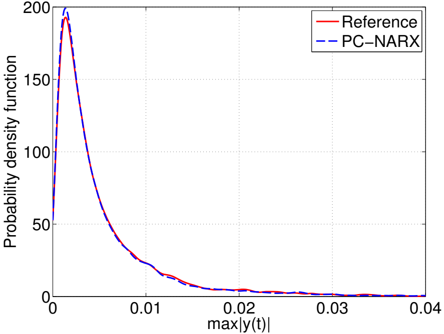


Surrogate model of single trajectories

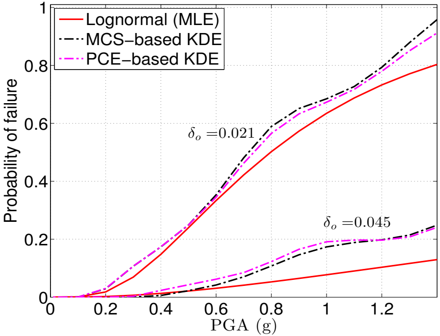
Frame structure – fragility curves

First-storey drift

- PC-NARX calibrated based on 300 simulations
- Reference results obtained from 10,000 Monte Carlo simulations



PDF of max. drift



Fragility curves for two drift thresholds

Outline

Introduction

Uncertainty quantification: why surrogate models?

Polynomial chaos expansions

Structural reliability analysis

- Reliability analysis
- Active learning
- Benchmark

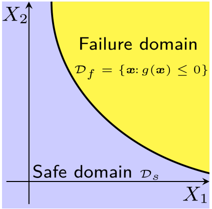
Reliability analysis

- Estimate the probability of occurence of an adverse event

$$P_f = \int_{\mathcal{D}_f} f_{\boldsymbol{X}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$$

$f_{\boldsymbol{X}}(\boldsymbol{x})$:	Joint distribution of the random vector $\boldsymbol{X}$
$\mathcal{D}_f = \{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} : g(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x}) \leq 0)\}$:	Failure domain

- Failure is assessed by a limit-state function $g : \boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} \mapsto \mathbb{R}$, based on a computational model $\mathcal{M}$
- Multi-dimensional 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

Active learning reliability

Enrich an initially poor experimental design using a learning function to improve the accuracy of the model in the vicinity of the limit-state surface

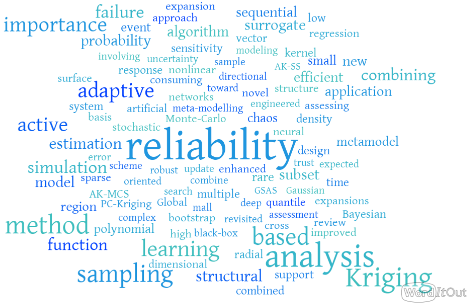
1. Generate an experimental design $\{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\} = \left\{\left(\boldsymbol{x}^{(i)}, g\left(\boldsymbol{x}^{(i)}\right)\right), i = 1, \ldots, N_0\right\}$
2. Train a surrogate model $\widetilde{g}$ using $\{\mathcal{X}, \mathcal{Y}\}$
3. Compute the failure probability $\widehat{P}_f$ using $\widetilde{g}$
4. Check whether some convergence criteria are met. If they are, stop, otherwise go to **step 5**
5. Choose the best next sample $\boldsymbol{x}^*$ to be added to $\boldsymbol{\mathcal{X}}$ based on an appropriate learning function
6. Add $\boldsymbol{x}^*$ and the corresponding response $g\left(\boldsymbol{x}^*\right)$ to the experimental design
7. Return to **step 2**

Active learning reliability methods

Teixeira *et al.* (2021), Moustapha *et al.* (2022)

> 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.* (2022)

	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
...	...	...	...

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. Active learning reliability: survey, general framework and benchmark (2022), Struct. Saf. (In press)

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

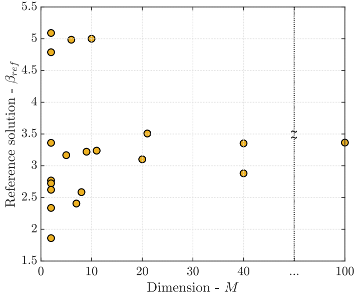
- Max. sample size: 10^4
- Target C.o.V: 2.5%
- Instrumental density: Standard Gaussian centered on the MPFP

Subset simulation

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

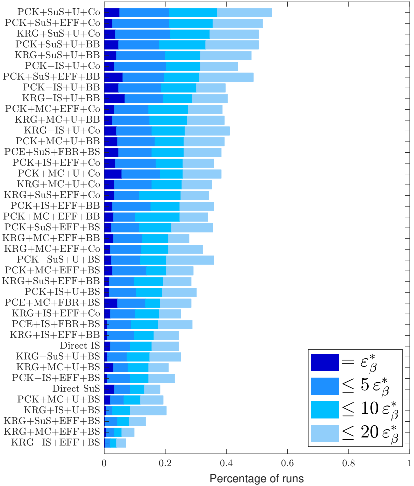
Selected problems

- 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	$P_{f,\text{ref}}$	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)

Ranking of the strategies: accuracy of β

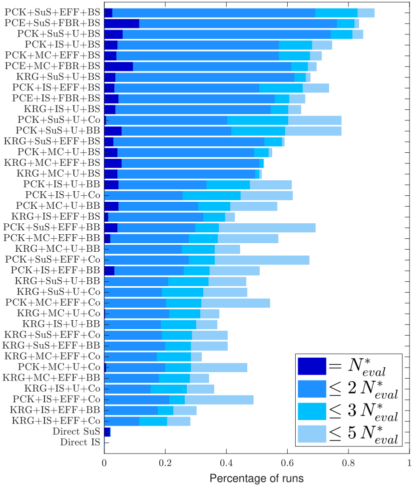


How many times a method ranks best in terms of smallest error on beta (resp. within 5, 10 or 20 times this relative error)?

$$\varepsilon = |\beta - \beta_{\mathrm{ref}}| \,/ \beta_{\mathrm{ref}}$$

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

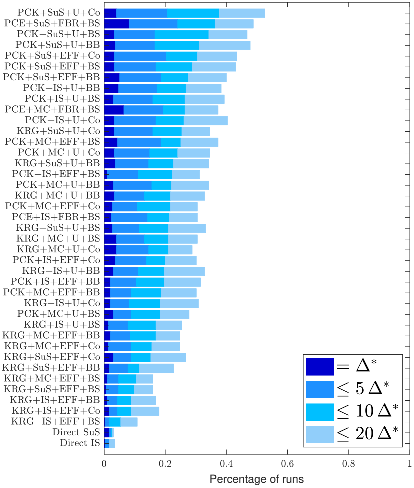
Ranking of the strategies: number of model evaluations



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 approaches: Direct SuS and 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)?

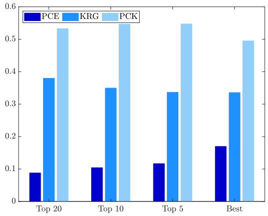
$$\Delta = \varepsilon_\beta \frac{N_{\text{eval}}}{\overline{N}_{\text{eval}}}$$

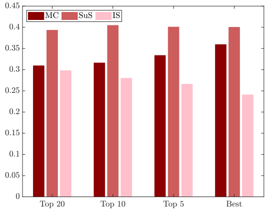
where $\overline{N}_{\text{eval}}$ is the median number of model evaluations for a particular problem (over all methods and replications)

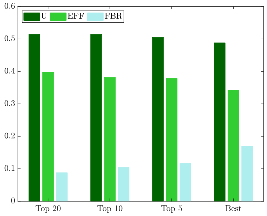
- Best approach: PC-Kriging + SuS + U + Combined stopping criterion
- Worst approaches: Direct SuS and Direct IS

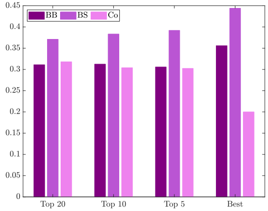
Results aggregated by method

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





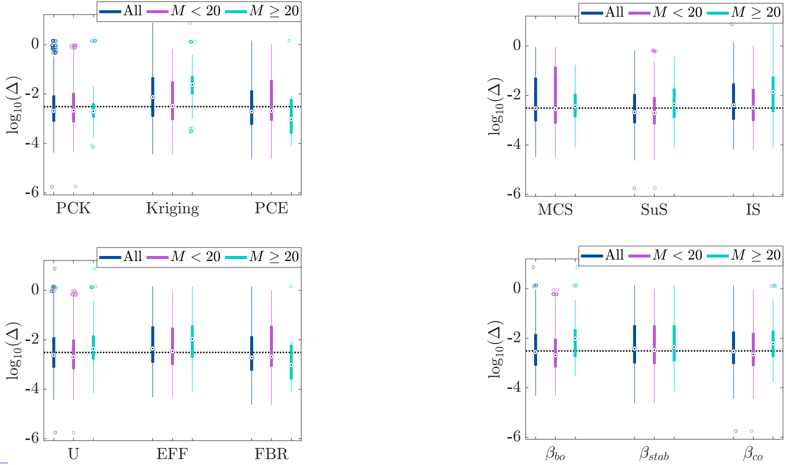




- Surrogates: PC-Kriging dominates by far
- Reliability: Slight advantage to subset simulation
- Learning function: U dominates both EFF and FBR
- Stopping criterion: Slight advantage to the stability criterion

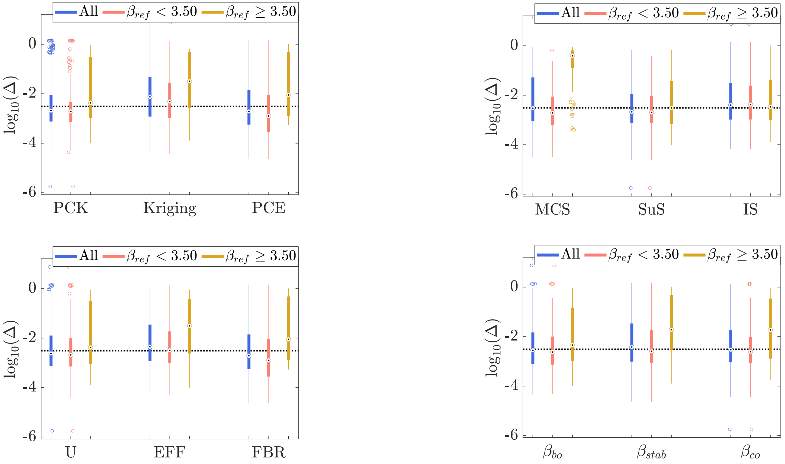
Performance w.r.t. problem feature: dimension

Results split in dimension: $M < 20$ vs. $M > 20$



Performance w.r.t. problem feature: P_f range

Results split in reliability index: $\beta < 3.5$ vs. $\beta > 3.5$



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

Conclusions

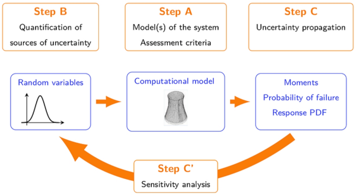
- Surrogate models are unavoidable for solving uncertainty quantification problems involving costly computational models (*e.g.* finite element models)
- Depending on the analysis, specific surrogates are most suitable: polynomial chaos expansions for distribution- and sensitivity analysis, Kriging/PC-Kriging (and low-rank tensor approximations) for reliability analysis
- Active learning enhance the performance of the most sophisticated reliability estimation algorithms
- All these techniques are non-intrusive: they rely on experimental designs, which may be adaptively built
- They are versatile, general-purpose and field-independent
- All the presented algorithms are available in the general-purpose uncertainty quantification software UQLab

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



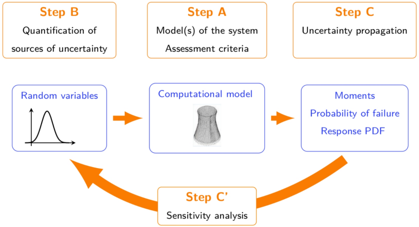
- MATLAB®-based Uncertainty Quantification framework
- State-of-the art, highly optimized open source algorithms
- Fast learning curve for beginners
- Modular structure, easy to extend
- Exhaustive documentation

UQLab: The Uncertainty Quantification Software

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

- Matlab-based platform, made of a core and a set of modules
- Project launched in January 2013, 1st beta version online on July 1st, 2015
- High quality documentation, learn-by-example tutorials
- Covers in a unique computational framework all aspects of uncertainty quantification

Philosophy



An uncertainty quantification problem is defined by:

- A computational model of the physical system
- A probabilistic model of the input uncertainties
- Analysis algorithms to solve the above-mentioned questions

UQLab is ... an implementation of the UQ framework

UQLab: The Uncertainty Quantification Software

- **Free access**
- ≈ 4,200 registered users
- ≈ 1,450+ active users from 94 countries

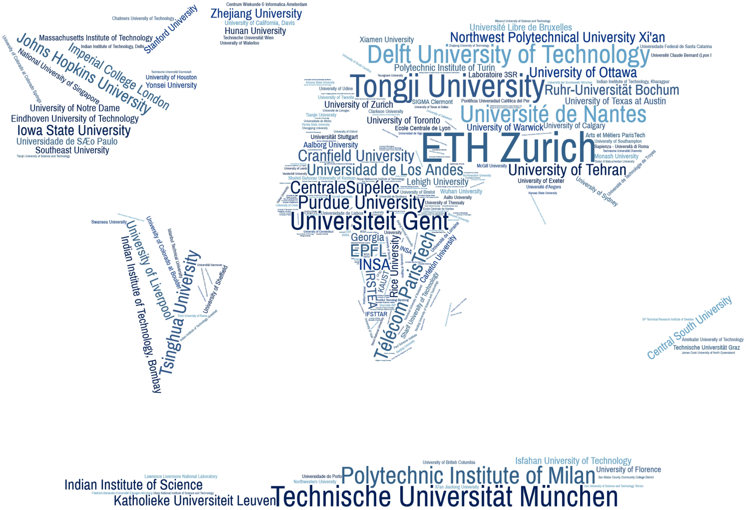
Version V.2.0 is fully open source!
Released on Feb. 1st, 2022

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

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

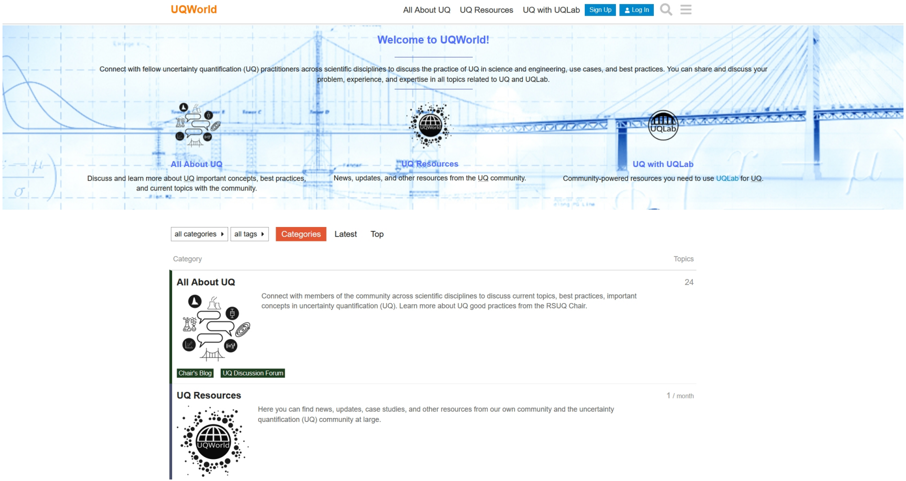
Country	# Users
United States	648
China	603
France	373
Switzerland	312
Germany	292
United Kingdom	183
India	168
Italy	161
Brazil	159
Canada	96

As of January 24, 2022


Chalmers University of Technology
Stanford University
Massachusetts Institute of Technology
Indian Institute of Technology, Delhi
Johns Hopkins University
Imperial College London
National University of Singapore
University of Colorado at Colorado Springs
Technische Universität Darmstadt
University of Houston
Yonsei University
University of Notre Dame
Eindhoven University of Technology
Iowa State University
Universidade de SÆo Paulo
Southeast University
Tianjin University of Science and Technology
Centrum Wiskunde & Informatica Amsterdam
Zhejiang University
University of California, Davis
Hunan University
Technische Universität Wien
University of Waterloo
Xiamen University
Missouri University of Science and Technology
Université Libre de Bruxelles
Northwest Polytechnical University Xi'an
Delft University of Technology
Universidade Federal de Santa Catarina
Université Claude Bernard (Lyon I
Polytechnic Institute of Turin
Laboratoire 3SR
University of Ottawa
Tongji University
Indian Institute of Technology, Kharagpur
Ruhr-Universität Bochum
University of Udine
University of Twente
University of Zurich
SIGMA Clermont
Pontificia Universidad Católica del Per
University of Texas at Austin
Clarkson University
Université de Nantes
University of Toronto
Ecole Centrale de Lyon
University of Warwick
University of Calgary
Universität Stuttgart
Aalborg University
ETH Zurich
Arts et Métiers ParisTech
University of Southampton
Sapienza - Università di Roma
Monash University
Cranfield University
Universidad de Los Andes
McGill University
University of Tehran
University of Exeter
Université d'Angers
Kansas State University
University of Sydney
Université de Technologie de Troyes
Shahid Bahonar University of Kerman
Royal Melbourne Institute of Technology
Lehigh University
CentraleSupélec
University of Bristol
Wuhan University
Aalto University
Purdue University
University of Thessaly
Ecole Centrale de Nantes
Universidade de Lisboa
Universiteit Gent
Georgia
EPFL
INSA
IRSTEA
KAUST
Rice University
Télécom ParisTech
Carleton University
Sharif University of Technology
IFSTTAR
Université de Lorraine
Swansea University
University of Liverpool
Indian Institute of Technology, Bombay
University of Colorado at Boulder
Istanbul Technical University
Tsinghua University
Universität Hannover
University of Sheffield
Third University of Rome
SP Technical Research Institute of Sweden
Central South University
Amirkabir University of Technology
Technische Universität Graz
James Cook University of North Queensland
University of British Columbia
Isfahan University of Technology
Lawrence Livermore National Laboratory
Universidade do Porto
Northwestern University
Polytechnic Institute of Milan
University of Florence
San Mateo County Community College District
Xi'an Jiaotong University
Iran University of Science and Technology Tehran
Indian Institute of Science
Katholieke Universiteit Leuven
Technische Universität München

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

UQLab

The Uncertainty Quantification Community

www.uqworld.org

UQWorld