Surrogate models for uncertainty quantification in computational sciences

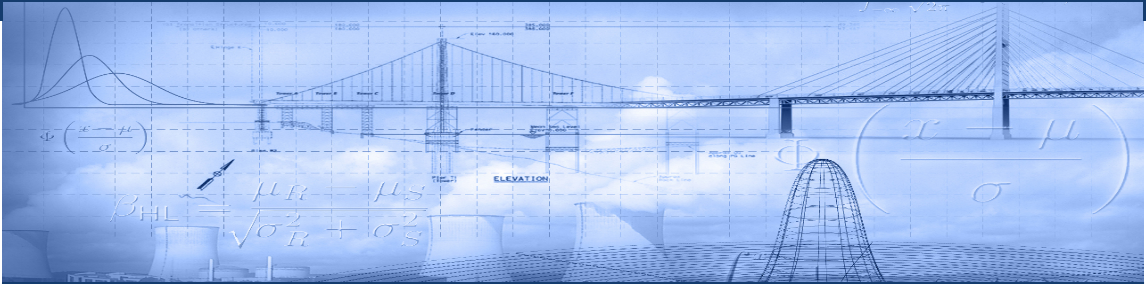
Other Conference Item

Author(s):
Sudret, Bruno

Publication date:
2022-11-22

Permanent link:
https://doi.org/10.3929/ethz-b-000583156

ETH zürich



Surrogate models for uncertainty quantification in computational sciences

Bruno Sudret

Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich

XLIII CILAMCE, Foz Do Iguaçu, Brazil – November 22, 2022

How to cite?

This presentation is a keynote lecture given at the XLIII Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE 2022) in Foz do Iguaçu, Paraná, Brazil on November 22nd, 2022.

How to cite

Sudret, B. *Surrogate models for uncertainty quantification in computational sciences*, XLIII Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE 2022), Foz do Iguaçu (Brazil), Keynote Lecture, November 22nd, 2022.

Iguaçu Falls

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

A computational model combines:

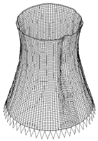
- A mathematical description of the physical phenomena (governing equations), *e.g.* mechanics, electromagnetism, fluid dynamics, etc.

$$\mathrm{div}\ \boldsymbol{\sigma} + \boldsymbol{f} = \mathbf{0}$$

$$\boldsymbol{\sigma} = \mathbf{D} \cdot \boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + {}^{\mathsf{T}}\nabla \boldsymbol{u}\right)$$

- Discretization techniques which transform continuous equations into linear algebra problems

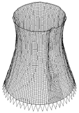


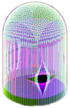
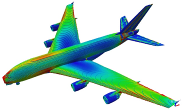
- Algorithms to solve the discretized equations

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

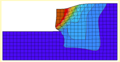
A computational model may be seen as a black box program that computes quantities of interest (QoI) (a.k.a. model responses) as a function of input parameters

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

Computational model $\mathcal{M}$

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

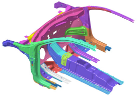
- Geometry
- Material properties
- Loading



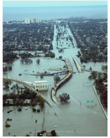
- Analytical formula
- Finite element model
- Comput. workflow

- 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

Global framework for uncertainty quantification

Step B

Quantification of sources of uncertainty

Step A

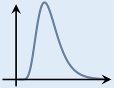
Model(s) of the system

Assessment criteria

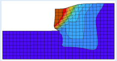
Step C

Uncertainty propagation

Random variables



Computational model



Moments

Probability of failure

Response PDF

Step C'

Sensitivity analysis

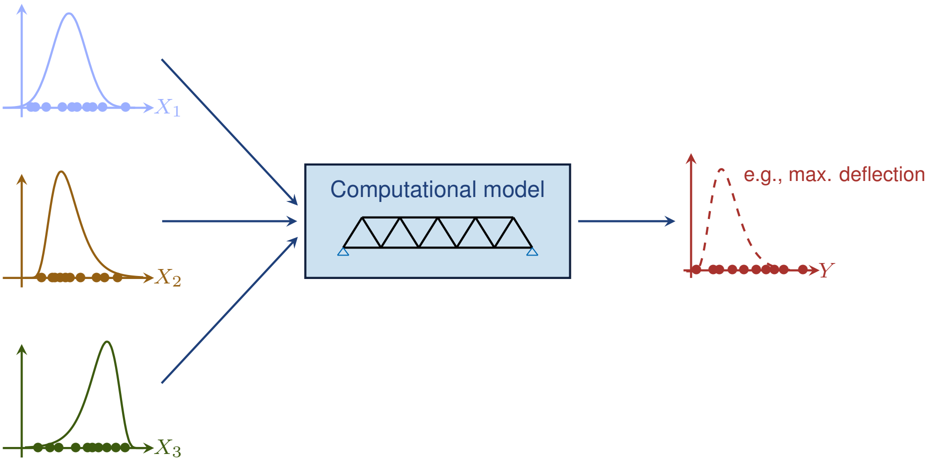
B. Sudret, Uncertainty propagation and sensitivity analysis in mechanical models – contributions to structural reliability and stochastic spectral methods (2007)

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

Uncertainty propagation using Monte Carlo simulation



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 $\propto n^{-1/2}$

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

Simulated data

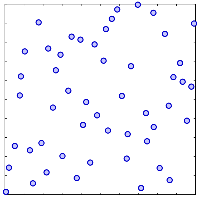
- It is fast to evaluate!

Surrogate models for uncertainty quantification

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	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{m} a_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\boldsymbol{a}\ ,\ b$
(Deep) Neural networks	$\tilde{\mathcal{M}}(\boldsymbol{x}) = f_n\left(\cdots f_2\left(b_2 + f_1\left(b_1 + \boldsymbol{w}_1 \cdot \boldsymbol{x}\right) \cdot \boldsymbol{w}_2\right)\right)$	$\boldsymbol{w}, \boldsymbol{b}$

Ingredients for building a surrogate model

- Select an experimental design $\mathcal{X}$ that covers at best the domain of input parameters:
  - (Monte Carlo simulation)
  - Latin hypercube sampling (LHS)
  - Low-discrepancy sequences



- Run the computational model $\mathcal{M}$ onto $\mathcal{X}$ exactly as in Monte Carlo simulation

Ingredients for building a surrogate model

- 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

- Validate the surrogate model, *e.g.* estimate a global error $\varepsilon = \mathbb{E}\left[\left(\mathcal{M}(\boldsymbol{X}) - \tilde{\mathcal{M}}(\boldsymbol{X})\right)^2\right]$

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

- 6-8 orders of magnitude (!) less CPU for a single run
- 2-3 orders of magnitude less runs compared to a full Monte Carlo simulation

Outline

Polynomial chaos expansions in a nutshell

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

- We assume here for simplicity that the input parameters are independent with $X_i \sim f_{X_i}, i = 1, \ldots, M$
- PCE is also applicable in the general case using an isoprobabilistic transform $\boldsymbol{X} \mapsto \boldsymbol{\Xi}$

The polynomial chaos expansion of the (random) model response reads:

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

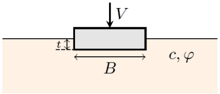
where:

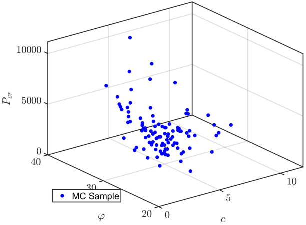
- $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$ are basis functions (multivariate orthonormal polynomials)
- $y_{\boldsymbol{\alpha}}$ are coefficients to be computed (coordinates)

Sampling (MCS) vs. spectral expansion (PCE)

Whereas MCS explores the output space /distribution point-by-point, the polynomial chaos expansion assumes a generic structure (polynomial function), which better exploits the available information (runs of the original model)

Example: load bearing capacity P_{cr} of a shallow foundation

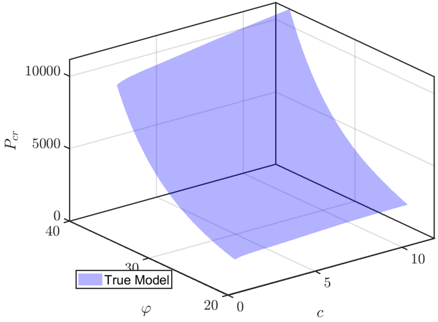




Thousands (resp. millions) of points are needed to grasp the structure of the response (resp. capture the rare events)

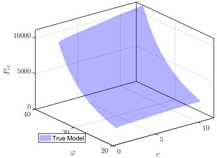
Defined as a function of the soil cohesion c and friction angle φ

Visualization of the PCE construction

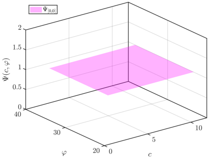


= “Sum of coefficients $\times$ basic surfaces”

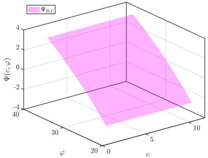
Visualization of the PCE construction



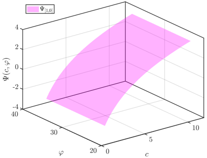
$= y_{0,0} \times$



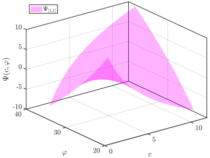
$+ y_{0,1} \times$



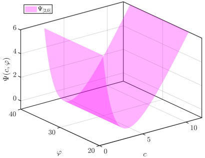
$+ y_{1,0} \times$



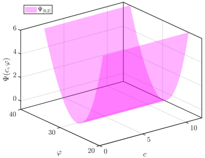
$+ y_{1,1} \times$



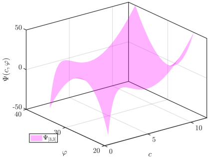
$+ y_{2,0} \times$



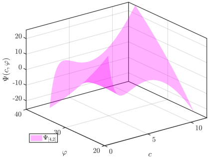
$+ \cdots + y_{0,2} \times$



$+ y_{3,3} \times$



$+ y_{4,2} \times$



Polynomial chaos expansion: procedure

$$Y^{\text{PCE}} = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$$

Four steps

- How to construct the polynomial basis $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$ for given $X_i \sim f_{X_i}$?
- How to compute the coefficients $y_{\boldsymbol{\alpha}}$?
- How to check the accuracy of the expansion ?
- How to answer the engineering questions:
  - Mean, standard deviation
  - PDF, quantiles
  - Sensitivity indices

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

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

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

Outline

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:

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

Discrete (ordinary) least-square minimization

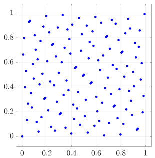
An estimate of the mean square error (sample average) is minimized:

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

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

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



- Compute the experimental matrix

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

- 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:

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

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

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

Leave-one-out cross validation

- From statistical learning theory, model validation shall be carried out using independent data

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

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

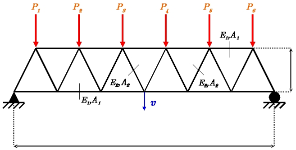
Outline

Curse of dimensionality

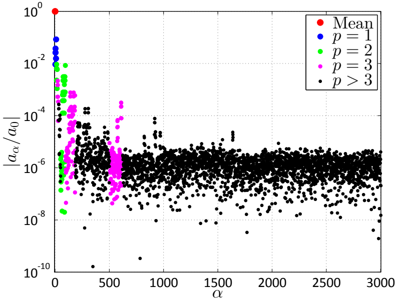
- The cardinality of the truncation scheme $\mathcal{A}^{M,p}$ is $P = \dfrac{(M+p)!}{M!\,p!}$
- Typical computational requirements: $n = OSR \cdot P$ where the oversampling rate is $OSR = 2-3$

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)



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, LARS, Bayesian compressive sensing, subspace pursuit, etc.
- State-of-the-art-review and comparisons available in:

  Lüthen, N., Marelli, S. & Sudret, B. *Sparse polynomial chaos expansions: Literature survey and benchmark*, SIAM/ASA J. Unc. Quant., 2021, 9, 593-649 https://doi.org/10.1137/20M1315774

  –, *Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications*, Int. J. Uncertainty Quantification, 2022, 12, 49-74

  https://doi.org/10.1615/Int.J.UncertaintyQuantification.2021036153

Outline

Post-processing sparse PC expansions

Statistical moments

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

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

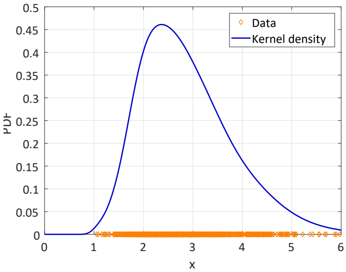
Variance: $$\hat{\sigma}_Y^2 = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} y_{\boldsymbol{\alpha}}^2$$

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

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

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

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

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

- Sobol' indices:
$$S_{\mathbf{u}} \stackrel{\text{def}}{=} \frac{\text{Var}\left[\mathcal{M}_{\mathbf{u}}(\boldsymbol{X}_{\mathbf{u}})\right]}{D}$$
- First-order Sobol' indices:
$$S_i = \frac{D_i}{D} = \frac{\text{Var}\left[\mathcal{M}_i(X_i)\right]}{D}$$
Quantify the additive effect of each input parameter separately
- Total Sobol' indices:
$$S_i^T \stackrel{\text{def}}{=} \sum_{\mathbf{u} \supset i} S_{\mathbf{u}}$$
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

Outline

Models with time-dependent outputs

Problem statement

- Consider a computational model of a dynamical system:

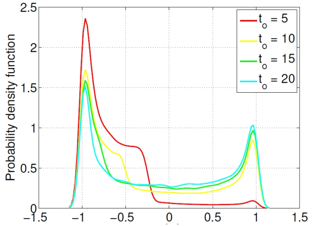
$$\mathcal{D}_{\boldsymbol{\Xi}} \times [0, T] : \ (\boldsymbol{\xi}, t) \mapsto \mathcal{M}(\boldsymbol{\xi}, t)$$

where $\boldsymbol{\Xi}$ is a random vector of uncertain parameters with given PDF $f_{\boldsymbol{\Xi}}$

- Uncertainties may be in:
  - The excitation, denoted by $x(\boldsymbol{\xi}_x, t)$
  - And/or in the system's characteristics ($\boldsymbol{\xi}_s$):

i.e.:

$$\mathcal{M}(\boldsymbol{\xi}, t) \equiv \mathcal{M}(x(\boldsymbol{\xi}_x, t), \ \boldsymbol{\xi}_s)$$



Time-dependent output PDF

Time-frozen PCE does not work!

Stochastic time warping

Problem

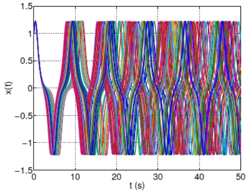
Mai & Sudret, SIAM J. Unc. Quant. (2017)

The various trajectories are "similar" yet not in phase, thus the complex time-frozen response

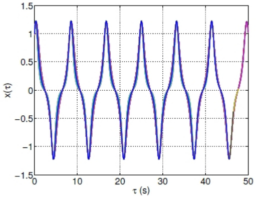
Principles of the method

- A specific warped time scale τ is introduced for each trajectory so that they become "in phase"
- Time-frozen PCE is carried out in the warped time scale using reduced-order modelling (principal component analysis)
- Predictions are carried out in the warped time scale and back-transformed in the real time line

Kraichnan Orszag model



Original trajectories



Trajectories after time warping

Example: Oregonator model

The Oregonator model represents a well-stirred, homogeneous chemical system governed by a three species coupled mechanism

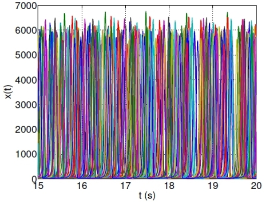
Governing equations

$$\dot{x}(t) = k_1\, y(t) - k_2\, x(t)\, y(t) + k_3\, x(t) - k_4\, x(t)^2$$
$$\dot{y}(t) = -k_1\, y(t) - k_2\, x(t)\, y(t) + k_5\, z(t)$$
$$\dot{z}(t) = k_3\, x(t) - k_5\, z(t)$$

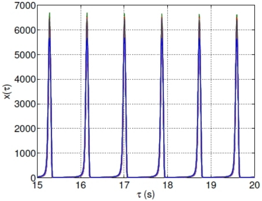
Input reaction parameters

Parameter	Distribution	Values
k_1	Uniform	$\mathcal{U}[1.8,\ 2.2]$
k_2	Uniform	$\mathcal{U}[0.095,\ 0.1005]$
k_3	Gaussian	$\mathcal{N}(104, 1.04)$
k_4	Uniform	$\mathcal{U}[0.0076, 0.0084]$
k_5	Uniform	$\mathcal{U}[23.4, 28.6]$

Le Maître et al. (2010)



Original trajectories

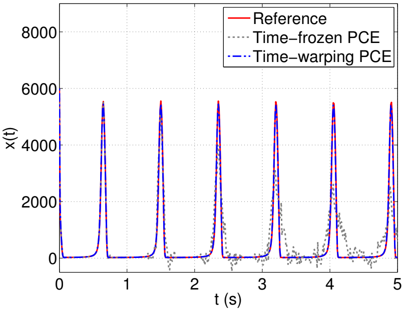


Trajectories after time warping

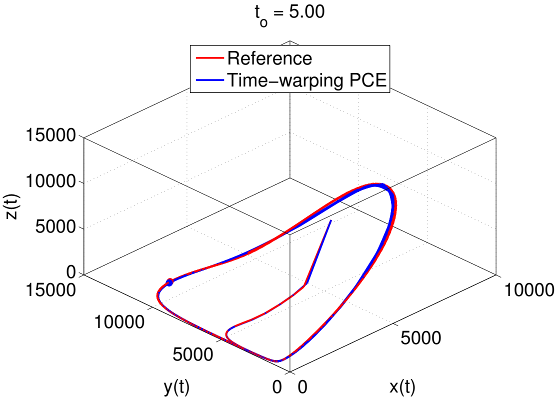
Oregonator model: prediction

Surrogate model

- Experimental design of size $n = 50$
- Validation set of size $n_{val} = 10,000$

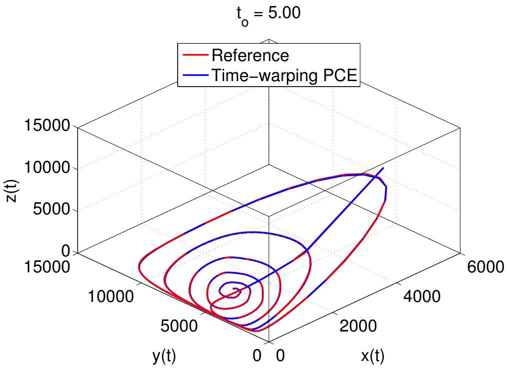


A specific trajectory ($\varepsilon = 0.0294$)

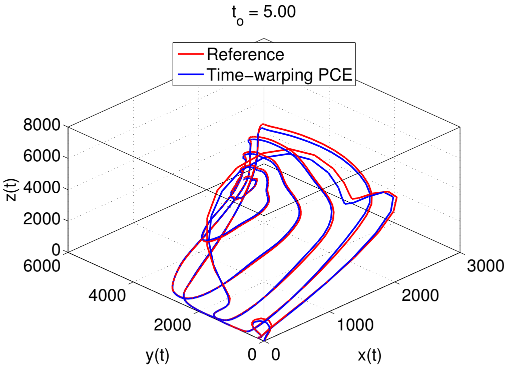


A trajectory in the state-space

Oregonator model: mean and std trajectories



Mean trajectory ($\varepsilon \approx 10^{-4}$)



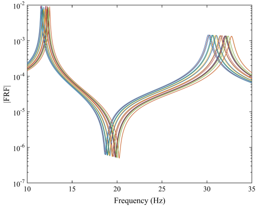
Standard deviation ($\varepsilon \approx 10^{-3}$)

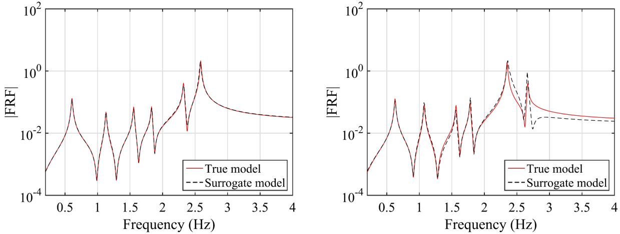
Dynamics in the frequency domain

Premise

Vaghoubi, Marelli & Sudret, Prob. Eng. Mech. (2017)

- Frequency response functions (FRF) allow one to compute the response to harmonic excitation
- In case of uncertain system properties (masses, stiffness coefficients) the resonance frequencies are shifted





(a) Typical FRF prediction (b) Worst FRF prediction

Nonlinear transient models: PC-NARX

Goal

Mai, Spiridonakos, Chatzi & Sudret, Int. J. Uncer. Quant. (2016)

Address uncertainty quantification problems for earthquake engineering, which involves transient, strongly non-linear mechanical models

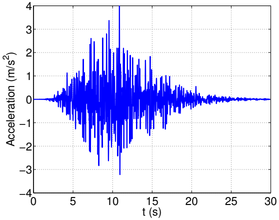
PC-NARX

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

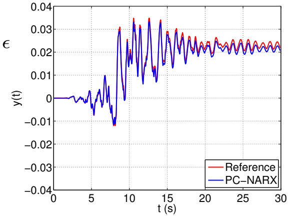
$$y(t) = \mathcal{F}\left(x(t), \ldots, x(t-n_x), y(t-1), \ldots, y(t-n_y)\right) + \epsilon_t \equiv \mathcal{F}\left(\boldsymbol{z}(t)\right) + \epsilon$$

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

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



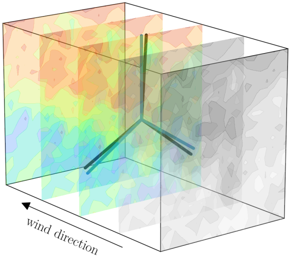
Earthquake ground motion



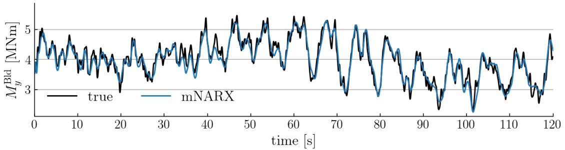
Structural response

Wind turbine simulations: mNARX surrogate

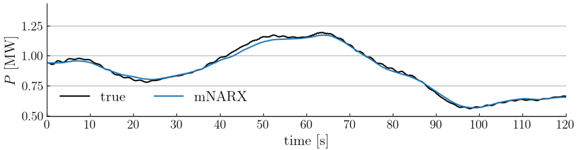
Movie-to-time series surrogate



Blade flapwise bending moment

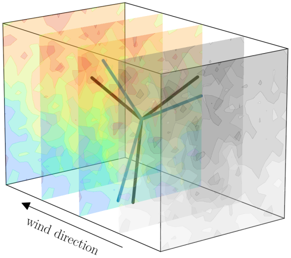


Generated power

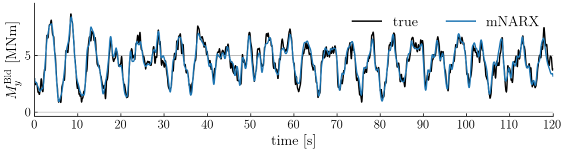


Wind turbine simulations: mNARX surrogate

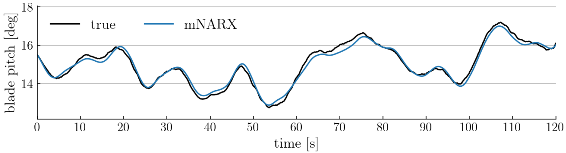
Movie-to-time series surrogate



Blade flapwise bending moment



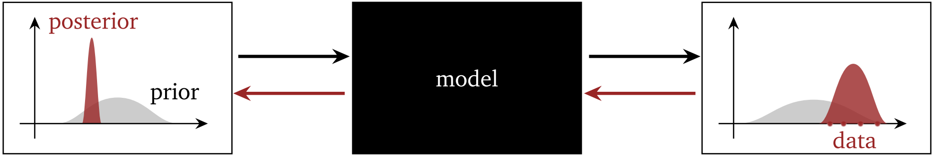
Blade pitch



Outline

Bayesian inversion: framework

Consider a computational model $\mathcal{M}$ with input parameters $\boldsymbol{X} \sim \pi(\boldsymbol{x})$



Bayesian inverse problem

$$\pi(\boldsymbol{x}|\mathcal{Y}) = \frac{\mathcal{L}(\boldsymbol{x};\mathcal{Y})\pi(\boldsymbol{x})}{Z} \quad \text{where} \quad Z = \int_{\mathcal{D}_{\boldsymbol{X}}} \mathcal{L}(\boldsymbol{x};\mathcal{Y})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

with:

- $\mathcal{L} : \mathcal{D}_{\boldsymbol{X}} \to \mathbb{R}^+$: likelihood function (measure of how well the model fits the data)
- $\pi(\boldsymbol{x}|\mathcal{Y})$: posterior density function

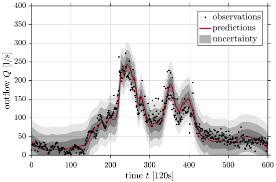
Bayesian inversion for model calibration

PCE as a surrogate of the forward model

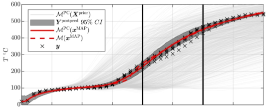
- Used in conjunction with Markov Chain Monte Carlo (MCMC) simulation

Application to sewer networks — Nagel, Rieckermann & Sudret, Reliab. Eng. Sys. Safety (2020)

Application to fire insulation panels — Wagner, Fahrni, Klippel, Frangi & Sudret, Eng.Struc. (2020)



Predicted outflow



Predicted temperature

Spectral likelihood expansions

- The likelihood function is expanded with a PCE, which leads to analytical solutions for posterior distributions and moments

Nagel & Sudret, J. Comp. Phys. (2016)

- Stochastic spectral embedding for localized posteriors and adaptive designs

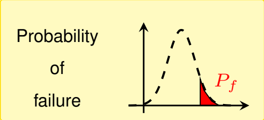
Marelli, Wagner, Lataniotis & Sudret, Int. J. Unc. Quant. (2021)

Marelli, Wagner, & Sudret, J. Comput. Phys. (2021)

Outline

Reliability analysis: problem statement

- For the assessment of the system's performance, a failure criterion is 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

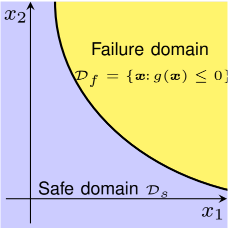
Limit state function

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

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

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

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



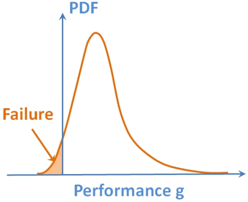
e.g. $g(\boldsymbol{x}) = q_{adm} - \mathcal{M}(\boldsymbol{x})$

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:

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

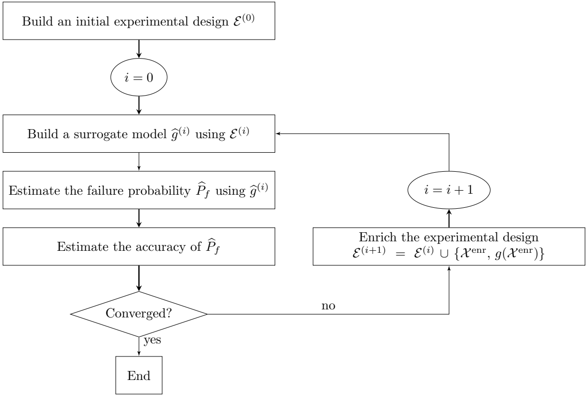
- Failures are (usually) rare events: sought probability in the range 10^{-2} to 10^{-8}

Active learning reliability framework

Bichon *et al.* (2008, 2011), Echard *et al.* (2011)

Principle

A surrogate model is built by iteratively enriching the experimental design $\mathcal{E} = \{\mathcal{X}, g(\mathcal{X})\}$ (using a learning function) so as to be accurate in the vicinity of the limit-state surface



Surrogate: Gaussian process (Kriging) model

Rasmussen & Williams (2006)

- Kriging assumes that $g(\boldsymbol{x})$ is a trajectory of an underlying Gaussian process

$$g(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x}) + \sigma^2 Z(\boldsymbol{x}; \boldsymbol{\theta})$$

$\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x})$: trend - $Z(\boldsymbol{x})$: zero-mean, Gaussian process with covariance function $\sigma^2 R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$

- The experimental design response $\mathcal{Y}$ and the response $\hat{g}(\boldsymbol{x})$ for a new point $\boldsymbol{x}$ are jointly Gaussian

$$\begin{Bmatrix} \hat{g}(\boldsymbol{x}) \\ \mathcal{Y} \end{Bmatrix} \sim \mathcal{N}_{N+1}\left(\begin{Bmatrix} \mathbf{f}(\boldsymbol{x})^T \beta \\ \mathbf{F}\beta \end{Bmatrix}, \quad \sigma^2 \begin{Bmatrix} 1 & \mathbf{r}^T(\boldsymbol{x}) \\ \mathbf{r}(\boldsymbol{x}) & \mathbf{R} \end{Bmatrix} \right)$$

- The prediction is given by the conditional mean $\mu_{\hat{g}(\boldsymbol{x})}$ and variance $\sigma^2_{\hat{g}(\boldsymbol{x})}$

$$\mu_{\hat{g}(\boldsymbol{x})} = \mathbf{f}^T(\boldsymbol{x})\hat{\beta} + \mathbf{r}^T(\boldsymbol{x})\mathbf{R}^{-1}\left(\mathcal{Y} - \mathbf{F}\hat{\beta}\right)$$

$$\sigma^2_{\hat{g}(\boldsymbol{x})} = \widehat{\sigma}^2\left(1 - \mathbf{r}^T(\boldsymbol{x})\mathbf{R}^{-1}\mathbf{r}(\boldsymbol{x}) + \mathbf{u}^T(\boldsymbol{x})(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{u}(\boldsymbol{x})\right)$$

$R_{ij} = R\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \widehat{\gamma}\right)$ - $\boldsymbol{r}(\boldsymbol{x}) = R\left(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \widehat{\gamma}\right)$ - $\boldsymbol{F} = F_{ij} = f_j\left(\boldsymbol{x}^{(i)}\right)$

- $\left\{\widehat{\beta}, \widehat{\sigma}^2, \widehat{\boldsymbol{\theta}}\right\}$ are estimated by maximum likelihood

Polynomial-Chaos Kriging

Schöebi *et al.* (2015,2016)

- Universal Kriging with a sparse PCE model as trend

$$\mathcal{M}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) + \sigma^2 Z(\boldsymbol{x})$$

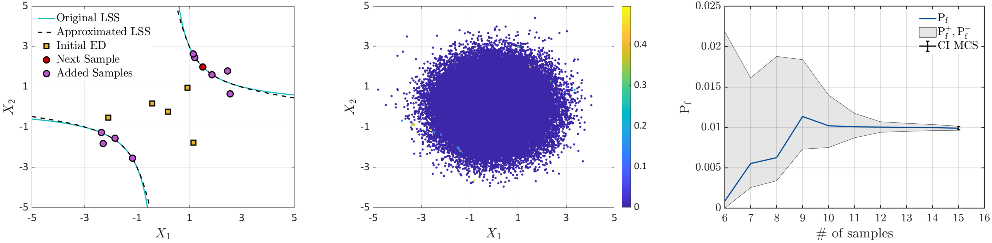
- Combines advantages of both PCE and Kriging:
  - PCE approximates the global behaviour of the model
  - Kriging captures local variations and provides the built-in local error estimation throught the Kriging variance

- Both the coefficients of the expansion $\{y_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{A}\}$ and the auto-correlation parameters $\widehat{\boldsymbol{\theta}}$ are calibrated

Active Kriging - Monte Carlo simulation (AK-MCS)

Echard *et al.* (2011)

- Gaussian process model to emulate the limit-state
- ED locally enriched using the deviation number
$$U(\boldsymbol{x}) = \frac{|\mu_{\hat{g}(\boldsymbol{x})}|}{\sigma_{\hat{g}(\boldsymbol{x})}}$$
- Probability of failure estimated using Monte Carlo simulation
- Convergence assumed when U is sufficiently large

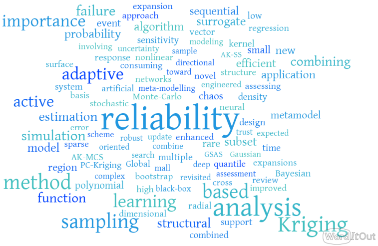


Active learning reliability methods

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 algorithm for reliability estimation
  - the learning function
  - 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) Rousouly et al. (2013)
Neural networks	Chojazyck et al. (2015) Gomes et al. (2019) Li & Wang (2020) [Deep NN]	Sundar & Shields (2016)	Chojazyck et al. (2015)	
Other	Schoebi & Sudret (2016) Sadoughi et al. (2017) Wagner et al. (2021)			

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

General framework

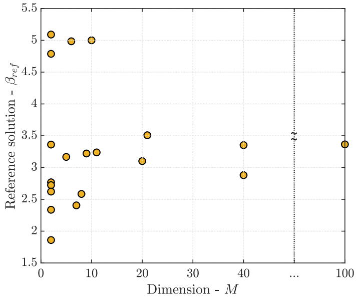
Moustapha, M., Marelli, S., Sudret, B. (2022). Active learning for structural reliability: Survey, general framework and benchmark. Structural Safety 96.

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

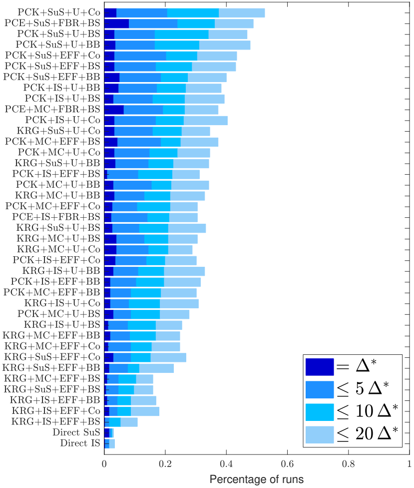
Extensive benchmark: selected problems

- 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: efficiency



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

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

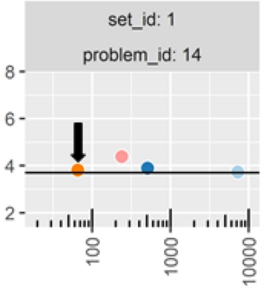
where $\overline{N}_{\text{eval}}$ is the median number of model evaluations for a particular problem (over all methods and replications)

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

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)



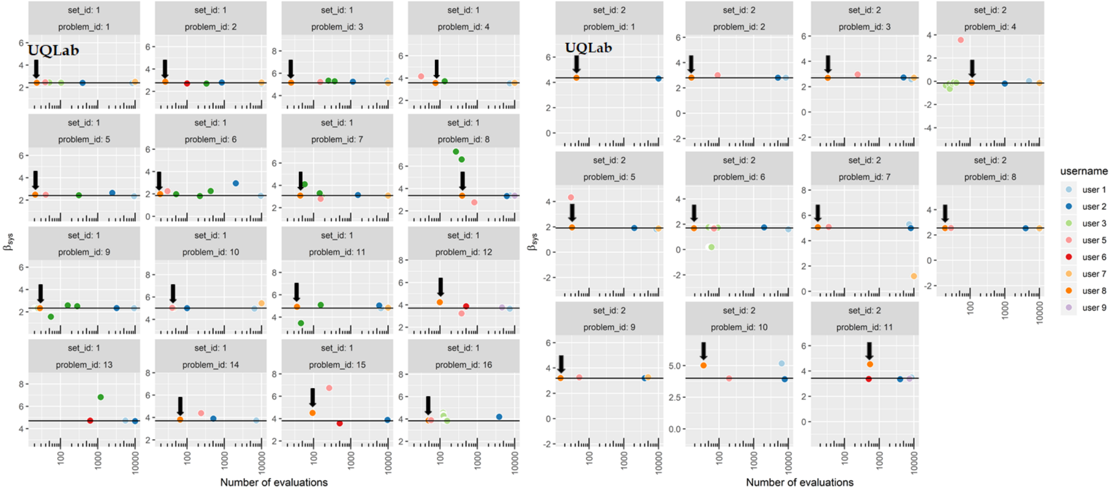
Summary plot (TNO)

- 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

- 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 (and active learning) for reliability analysis
- Sparse PCE and its extensions (time warping, PC-NARX, PC-Kriging, DRSM, etc.) allow us to address a wide range of engineering problems, including Bayesian inverse problems (without the need for MCMC simulations)
- Techniques for constructing surrogates are versatile, general-purpose and field-independent
- All the presented algorithms are available in the general-purpose uncertainty quantification software UQLab

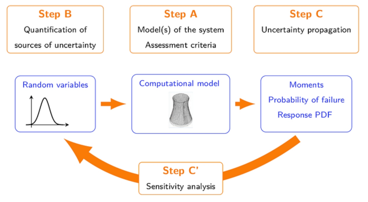
UQLab

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

- BSD 3-Clause license:

  Free access to academic, industrial, governmental and non-governmental users
- 5,200+ registered users from 94 countries since 2015

http://www.uqlab.com

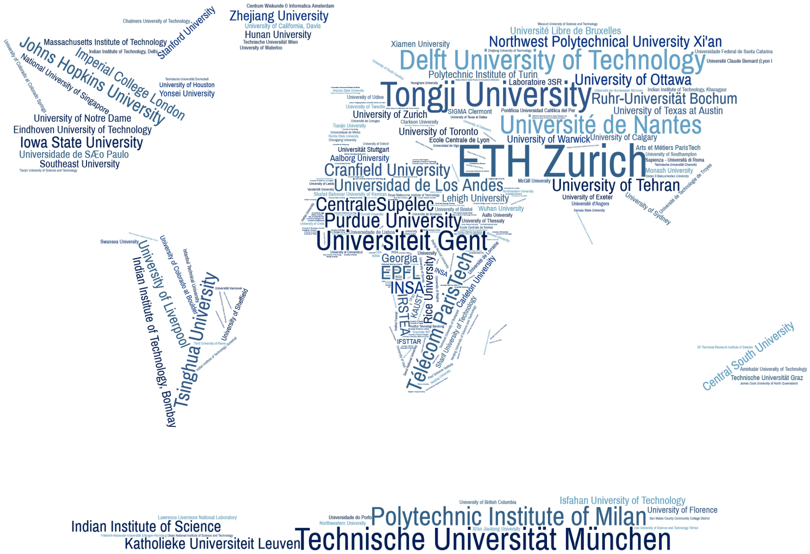
- The cloud version of UQLab, accessible via an API (SaaS)
- Available with python bindings for beta testing

UQ[py]Lab

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

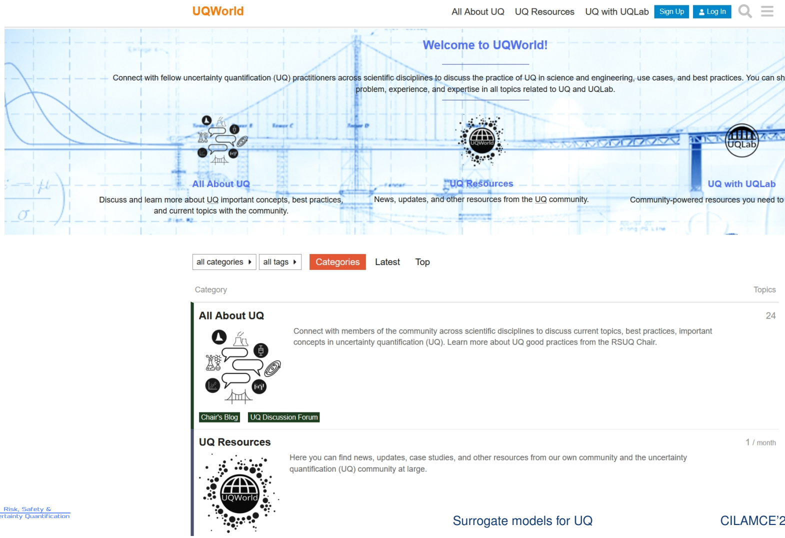
Country	# Users
China	849
United States	789
France	451
Switzerland	370
Germany	401
United Kingdom	214
India	206
Brazil	201
Italy	191
Canada	109

As of November 15, 2022


Chalmers University of Technology
Stanford University
Massachusetts Institute of Technology
Indian Institute of Technology, Delhi
Johns Hopkins University
National University of Singapore
Imperial College London
University of Colorado at Colorado Springs
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
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
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
Shahid Bahonar University of Kerman
Lehigh University
University of Exeter
Université d'Angers
CentraleSupélec
University of Bristol
Wuhan University
Aalto University
University of Sydney
Université de Technologie de Troyes
Purdue University
University of Thessaly
Universidade de Lisboa
Universiteit Gent
Swansea University
University of Liverpool
Indian Institute of Technology, Bombay
University of Colorado at Boulder
Istanbul Technical University
Tsinghua University
University of Sheffield
Universität Hannover
Georgia
EPFL
INSA
IRSTEA
Rice University
KAUST
Télécom ParisTech
Carleton University
Université de Lorraine
IFSTTAR
Sharif University of Technology
Central South University
Amirkabir University of Technology
Technische Universität Graz
University of British Columbia
Isfahan University of Technology
University of Florence
Lawrence Livermore National Laboratory
Indian Institute of Science
Universidade do Porto
Northwestern University
Polytechnic Institute of Milan
Xi'an Jiaotong University
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

www.uqpylab.uq-cloud.io

UQ[py]Lab

The Uncertainty Quantification Community

www.uqworld.org

UQWorld

Thank you very much for your attention !

References

Balesdent, M., J. Morio, and J. Marzat (2013).
Kriging-based adaptive importance sampling algorithms for rare event estimation.
Structural Safety 44, 1–10.

Berveiller, M., B. Sudret, and M. Lemaire (2006, January).
Stochastic finite elements: a non intrusive approach by regression.
Eur. J. Comput. Mech. 15(1–3), 81–92.

Bichon, B. J., M. S. Eldred, L. Swiler, S. Mahadevan, and J. McFarland (2008).
Efficient global reliability analysis for nonlinear implicit performance functions.
AIAA Journal 46(10), 2459–2468.

Bichon, B. J., J. M. McFarland, and S. Mahadevan (2011).
Efficient surrogate models for reliability analysis of systems with multiple failure modes.
Reliability Engineering & System Safety 96(10), 1386–1395.

Blatman, G. and B. Sudret (2010).
An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis.
Probabilistic Engineering Mechanics 25, 183–197.

Blatman, G. and B. Sudret (2011).
Adaptive sparse polynomial chaos expansion based on Least Angle Regression.
Journal of Computational Physics 230, 2345–2367.

Doostan, A. and H. Owhadi (2011, April).
A non-adapted sparse approximation of PDEs with stochastic inputs.
Journal of Computational Physics 230(8), 3015–3034.

Dubourg, V., B. Sudret, and J.-M. Bourinet (2011).
Reliability-based design optimization using Kriging and subset simulation.
Structural and Multidisciplinary Optimization 44(5), 673–690.

Echard, B., N. Gayton, and M. Lemaire (2011).
AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation.
Struct. Saf. 33(2), 145–154.

Ghanem, R. and P. Spanos (2003).
Stochastic Finite Elements: A Spectral Approach (2nd ed.).
Courier Dover Publications, Mineola.

Isukapalli, S. S. (1999).
Uncertainty analysis of transport-transformation models.
Ph. D. thesis, The State University of New Jersey.

Jakeman, J. D., M. S. Eldred, and K. Sargsyan (2015, May).
Enhancing ℓ_1-minimization estimates of polynomial chaos expansions using basis selection.
Journal of Computational Physics 289, 18–34.

Kersaudy, P., B. Sudret, N. Varsier, O. Picon, and J. Wiart (2015, April).
A new surrogate modeling technique combining Kriging and polynomial chaos expansions – Application to uncertainty analysis in computational dosimetry.
Journal of Computational Physics 286, 103–117.

Mai, C. V., M. D. Spiridonakos, E. N. Chatzi, and B. Sudret (2016).
Surrogate modeling for stochastic dynamical systems by combining nonlinear autoregressive with exogeneous input models and polynomial chaos expansions.
Int. J. Uncer. Quant. 6(4), 313–339.

Mai, C. V. and B. Sudret (2017).
Surrogate models for oscillatory systems using sparse polynomial chaos expansions and stochastic time warping.

SIAM/ASA J. Uncertain. Quantif. 5, 540–571.

Marelli, S., P.-R. Wagner, C. Lataniotis, and B. Sudret (2021).
Stochastic spectral embedding.
International Journal for Uncertainty Quantification 11(2), 25–47.

Moustapha, M., S. Marelli, and B. Sudret (2022, May).
Active learning for structural reliability: Survey, general framework and benchmark.
Structural Safety 96(102174).

Nagel, J., J. Rieckermann, and B. Sudret (2020).
Principal component analysis and sparse polynomial chaos expansions for global sensitivity analysis and model calibration: application to urban drainage simulation.
Reliability Engineering & System Safety 195(#106737).

Nagel, J. and B. Sudret (2016, March).
Spectral likelihood expansions for Bayesian inference.
Journal of Computational Physics 309, 267–294.

Rasmussen, C. E. and C. K. I. Williams (2006).
Gaussian processes for machine learning (Internet ed.).
Adaptive computation and machine learning. Cambridge, Massachusetts: MIT Press.

Rozsas, A. and A. Slobbe (2019).
Repository and Black-box Reliability Challenge 2019.
`https://gitlab.com/rozsasarpi/rprepo/` (Accessed: 2022-07-17).

Saltelli, A., M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola (2008).
Global Sensitivity Analysis – The Primer.
Wiley.

Santner, T. J., B. J. Williams, and W. I. Notz (2003).

The Design and Analysis of Computer Experiments.
Springer, New York.

Sargsyan, K., C. Safta, H. Najm, B. Debusschere, D. Ricciuto, and P. Thornton (2014).
Dimensionality reduction for complex models via Bayesian compressive sensing.
Int. J. Uncertainty Quantification 4(1), 63–93.

Schöbi, R., B. Sudret, and S. Marelli (2017).
Rare event estimation using Polynomial-Chaos-Kriging.
ASCE-ASME J. Risk Uncertainty Eng. Syst., Part A: Civ. Eng 3(2).
D4016002.

Schöbi, R., B. Sudret, and J. Wiart (2015).
Polynomial-chaos-based Kriging.
Int. J. Uncertainty Quantification 5(2), 171–193.

Sobol', I. M. (1993).
Sensitivity estimates for nonlinear mathematical models.
Math. Modeling & Comp. Exp. 1(4), 407–414.

Soize, C. and R. Ghanem (2004, January).
Physical systems with random uncertainties: chaos representations with arbitrary probability measure.
SIAM Journal on Scientific Computing 26(2), 395–410.

Spiridonakos, M. D. and E. N. Chatzi (2015).
Metamodeling of dynamic nonlinear structural systems through polynomial chaos NARX models.
Comput. Struct. 157, 99–113.

Sudret, B. (2007).
Uncertainty propagation and sensitivity analysis in mechanical models – Contributions to structural reliability and stochastic spectral methods.
Université Blaise Pascal, Clermont-Ferrand, France.

Habilitation à diriger des recherches, 173 pages.

Sudret, B. (2008).
Global sensitivity analysis using polynomial chaos expansions.
Reliability Engineering & System Safety 93, 964–979.

Teixeira, R., M. Nogal, and A. O'Connor (2021, March).
Adaptive approaches in metamodel-based reliability analysis: A review.
Structural Safety 89, 102019.

Wagner, P.-R., R. Fahrni, M. Klippel, A. Frangi, and B. Sudret (2020).
Bayesian calibration and sensitivity analysis of heat transfer models for fire insulation panels.
Engineering Structures 205(#110063).

Wagner, P.-R., S. Marelli, and B. Sudret (2021).
Bayesian model inversion using stochastic spectral embedding.
Journal of Computational Physics (110141).

Xiu, D. (2010).
Numerical methods for stochastic computations – A spectral method approach.
Princeton University press.

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