Surrogate models for efficient uncertainty quantification

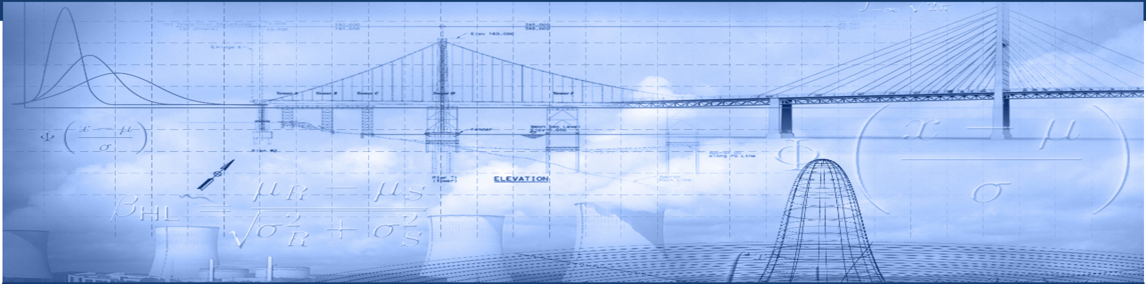
Presentation

Author(s):
Sudret, Bruno

Publication date:
2021-12-13

Permanent link:
https://doi.org/10.3929/ethz-b-000520348

ETH zürich



Surrogate models for efficient uncertainty quantification

Bruno Sudret

Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich

Euromech Colloquium 618 – University of Luxembourg – December 13, 2021

How to cite?

This presentation was given as the opening keynote lecture at the Euromech 618 Colloquium on *Uncertainty Quantification in Computational Mechanics* organized by the University of Luxembourg (online) on December 13, 2021.

Reference

Sudret, B. *Surrogate models for efficient uncertainty quantification*, Euromech 618 Colloquium on *Uncertainty Quantification in Computational Mechanics*, University of Luxembourg (online), December 13, 2021 (Keynote lecture).

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

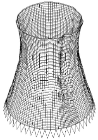
Complex engineering systems are designed and assessed using computational models, a.k.a simulators

A computational model combines:

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

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

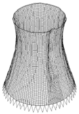
- Discretization techniques which transform continuous equations into linear algebra problems
- Algorithms to solve the discretized equations

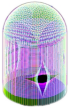


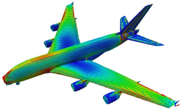
Computational models in engineering

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





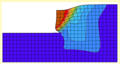


Computational models: the abstract viewpoint

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

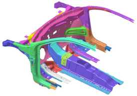
- Geometry
- Material properties
- Loading



- Analytical formula
- Finite element model
- Comput. workflow

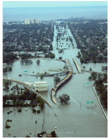
- Displacements
- Strains, stresses
- Temperature, etc.

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

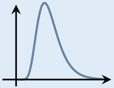
Model(s) of the system

Assessment criteria

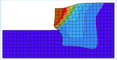
Step C

Uncertainty propagation

Random variables



Computational model



Moments

Probability of failure

Response PDF

Step C'

Sensitivity analysis

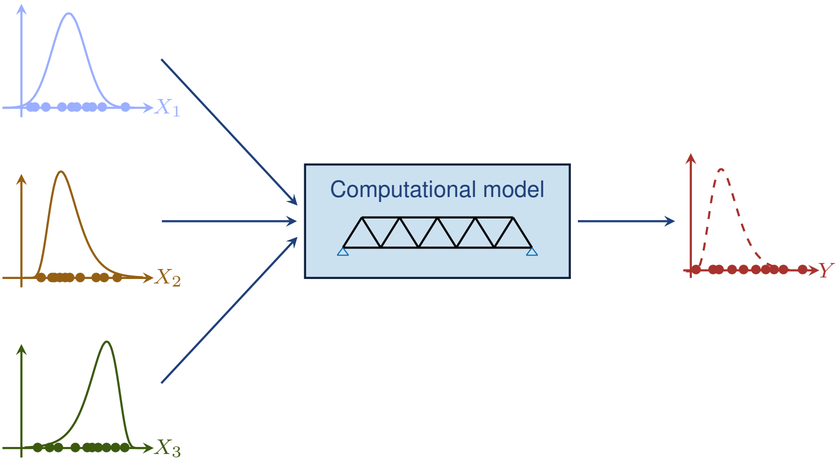
B. Sudret, *Uncertainty propagation and sensitivity analysis in mechanical models – contributions to structural reliability and stochastic spectral methods (2007)*

Uncertainty propagation using Monte Carlo simulation

Principle: Generate virtual prototypes of the system using random numbers

- A sample set $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ is drawn according to the input distribution $f_{\boldsymbol{X}}$
- For each sample the quantity of interest (resp. performance criterion) is evaluated, say $\mathcal{Y} = \{\mathcal{M}(\boldsymbol{x}_1), \dots, \mathcal{M}(\boldsymbol{x}_n)\}$
- 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 $\mathcal{X} = \left\{ \boldsymbol{x}^{(i)},\ i = 1, \ldots, N \right\}$

Simulated data

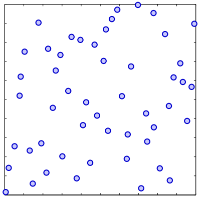
- It is fast to evaluate!

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

Surrogate modelling vs. machine learning

Features	Supervised learning	Surrogate modelling
Computational model $\mathcal{M}$	✗	✔
Probabilistic model of the input $\boldsymbol{X} \sim f_{\boldsymbol{X}}$	✗	✔
Training data: $\mathcal{X} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, n\}$	✔ Training data set (big data)	✔ Experimental design (small data)
Prediction goal: for a new $\boldsymbol{x} \notin \mathcal{X}$, $y(\boldsymbol{x})$?	$\sum_{i=1}^{m} y_i\, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}}\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$
Validation (resp. cross-validation)	✔ Validation set	✔ Leave-one-out CV

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 $X_i \sim f_{X_i},\ i = 1, \ldots, d$
- 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:

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

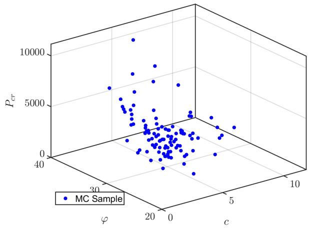
where:

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

Sampling (MCS) vs. spectral expansion (PCE)

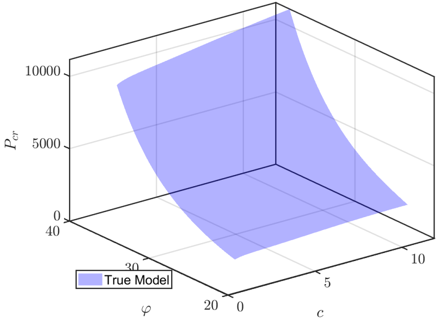
Whereas MCS explores the output space /distribution point-by-point, the polynomial chaos expansion assumes a generic structure (polynomial function), which better exploits the available information (runs of the original model)

Example: load bearing capacity as a function of (c, φ)



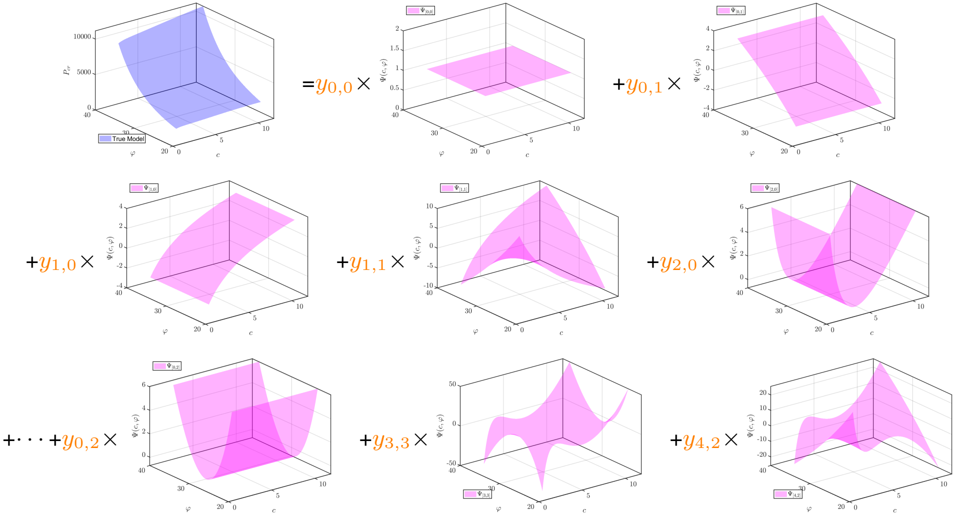
Thousands (resp. millions) of points are needed to grasp the structure of the response (resp. capture the rare events)

Visualization of the PCE construction



= "Sum of coefficients $\times$ basic surfaces"

Visualization of the PCE construction



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

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

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

Outline

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

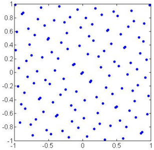
An estimate of the mean square error (sample average) is minimized:

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

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

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



- 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:

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

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

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

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 $\mathbf{A}(\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}$

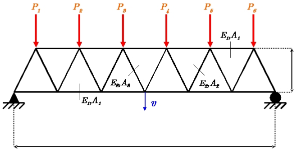
Outline

Curse of dimensionality

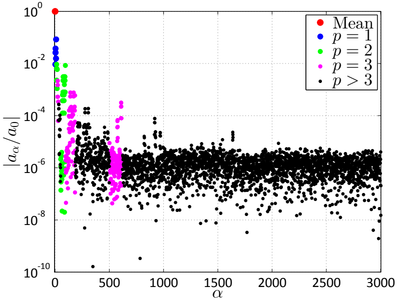
- The cardinality of the truncation scheme $\mathcal{A}^{M,p}$ is $P = \dfrac{(M+p)!}{M!\,p!}$
- Typical computational requirements: $n = OSR \cdot P$ where the oversampling rate is $OSR = 2 - 3$

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)



Hyperbolic truncation sets

Sparsity-of-effects principle

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

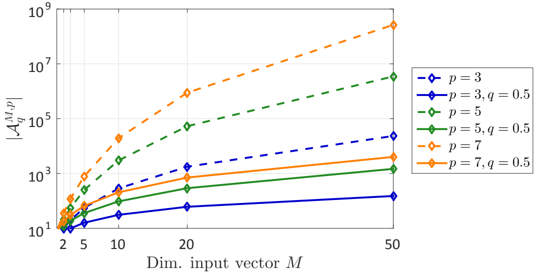
In most engineering problems, only low-order interactions between the input variables are relevant

- $q-$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, 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

  –, *Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications*, Int. J. Uncertainty Quantification, 2021, 12, 1-26

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:

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

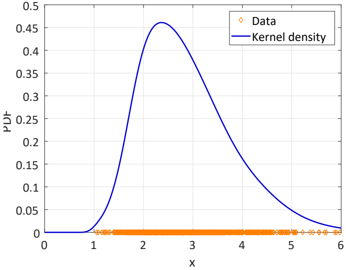
$$\text{Variance:} \qquad \hat{\sigma}_Y^2 = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} y_{\boldsymbol{\alpha}}^2$$

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

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

$$\begin{aligned}\mathcal{M}(\boldsymbol{x}) &= \mathcal{M}_0 + \sum_{i=1}^{M} \mathcal{M}_i(x_i) + \sum_{1 \le i < j \le M} \mathcal{M}_{ij}(x_i, x_j) + \dots + \mathcal{M}_{12\dots M}(\boldsymbol{x}) \\ &= \mathcal{M}_0 + \sum_{\mathbf{u} \subset \{1, \dots, M\}} \mathcal{M}_{\mathbf{u}}(\boldsymbol{x}_{\mathbf{u}}) \qquad (\boldsymbol{x}_{\mathbf{u}} \stackrel{\text{def}}{=} \{x_{i_1}, \dots, x_{i_s}\})\end{aligned}$$

- 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:

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

- 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:

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

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

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

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:

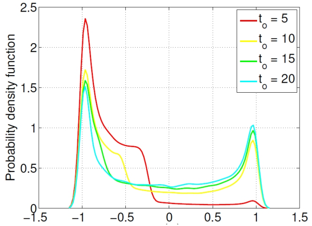
$$\mathcal{D}_{\boldsymbol{\Xi}} \times [0, T] : \; (\boldsymbol{\xi}, t) \mapsto \mathcal{M}(\boldsymbol{\xi}, t)$$

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 does not work!

Stochastic time warping

Problem

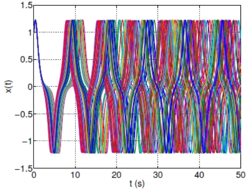
Mai & Sudret, SIAM J. Unc. Quant. (2017)

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

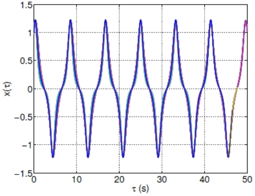
Principles of the method

- 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

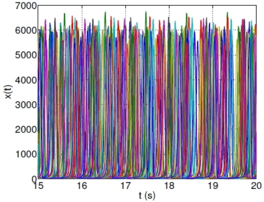
Governing equations

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

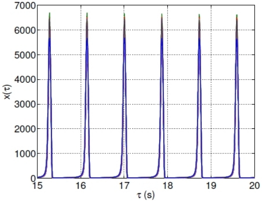
Input reaction parameters

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

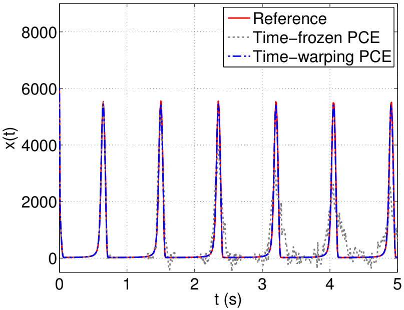


Trajectories after time warping

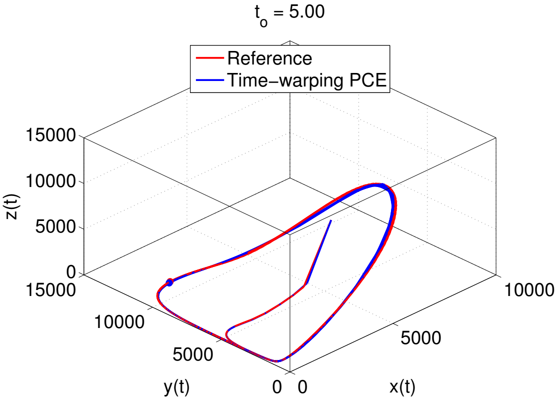
Oregonator model: prediction

Surrogate model

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

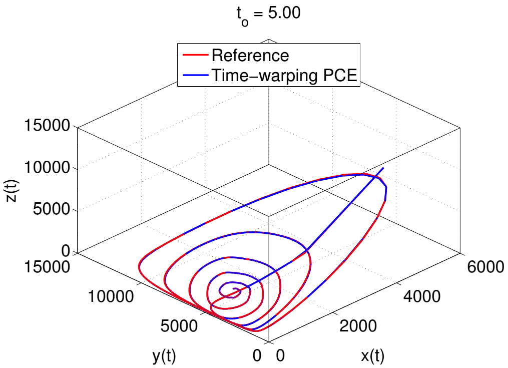


A specific trajectory ($\varepsilon = 0.0294$)

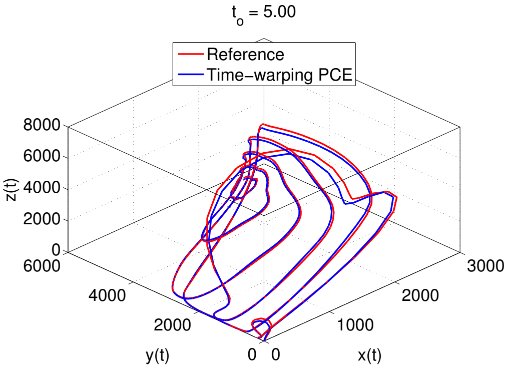


A trajectory in the state-space

Oregonator model: mean and std trajectories



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



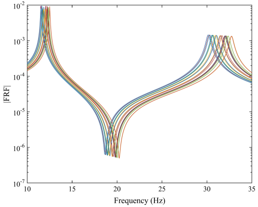
Standard deviation ($\varepsilon \approx 10^{-3}$)

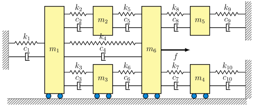
Dynamics in the frequency domain

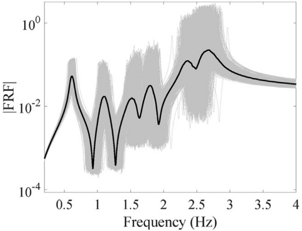
Premise

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

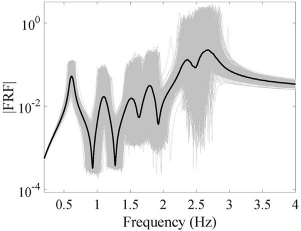
- Frequency response functions (FRF) allow one to compute the response to harmonic excitation
- In case of uncertain system properties (masses, stiffness coefficients) the resonance frequencies are shifted







(a) 6^{th} system output- True model



(b) 6^{th} system output- Surrogate model

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

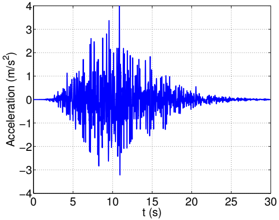
PC-NARX

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

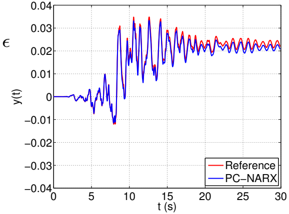
$$y(t) = \mathcal{F}\left(x(t), \ldots, x(t-n_x), y(t-1), \ldots, y(t-n_y)\right) + \epsilon_t \equiv \mathcal{F}\left(\boldsymbol{z}(t)\right) + \epsilon$$

- 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

Outline

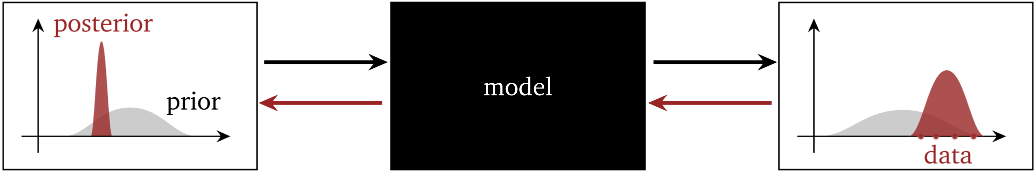
Bayesian inversion: framework

Consider a computational model $\mathcal{M}$ with input parameters $\boldsymbol{X} \sim \pi(\boldsymbol{x})$ and measurements $\mathcal{Y}$, the Bayesian inverse problem reads:

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

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



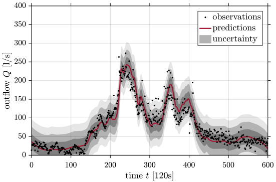
Bayesian inversion for model calibration

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

Spectral likelihood expansions

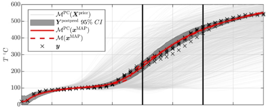
- The likelihood function is expanded with a PCE, which leads to analytical solutions for posterior distributions and moments

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)



Predicted temperature

Polymorphic (epistemic/aleatory) uncertainty propagation

Propagation of mixed epistemic/aleatory uncertainties

- Input uncertainty modelled by free (resp.) parametric p-boxes
- Uncertainty propagation using augmented spaces and optimization

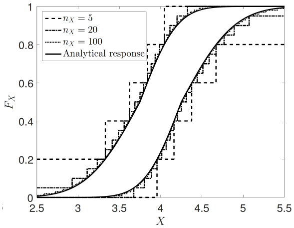
Schöbi & Sudret, J. Comp. Phys (2017)

- Structural reliability analysis

Schöbi & Sudret, Prob. Eng. Mech. (2017)

- Global sensitivity analysis extended to p-boxes inputs

Schöbi & Sudret, Reliab. Eng. Sys. Safety (2019)



Sparse polynomial chaos expansions for machine learning

Data-driven PCEs can be constructed from raw data sets using:

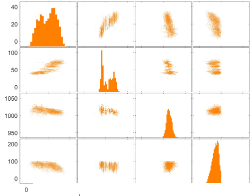
- Nonparametric representation of the input PDFs (kernel smoothing)
- Vine copulas to model the dependence

Example: combined cycle power plant (CCPP) Data set

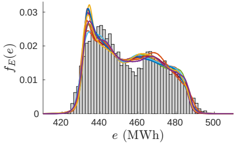
- 9,568 data points
- 4 features
- Output: net hourly electrical energy output

	MAE	min. MAE	mean-min	rMAE (%)
aPCEonX	3.11 ± 0.03	3.05	0.06	0.68 ± 0.007
BREP-NN†	$3.22 \pm$ n.a.	2.82	0.40	n.a.

Torre, Marelli, Embrechts & Sudret, J. Comput. Phys. (2019)



Raw data



PDF of CCPP

Conclusions

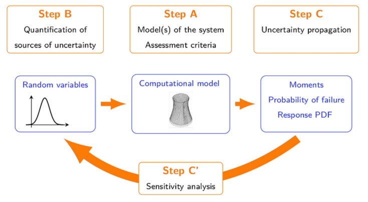
- Surrogate models are unavoidable for solving uncertainty quantification problems involving costly computational models (*e.g.* finite element models)
- Depending on the analysis, specific surrogates are most suitable: polynomial chaos expansions for distribution- and sensitivity analysis, Kriging (and 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

- **free access to academia**
- 4,120 registered users
- 1,500+ active users from 92 countries

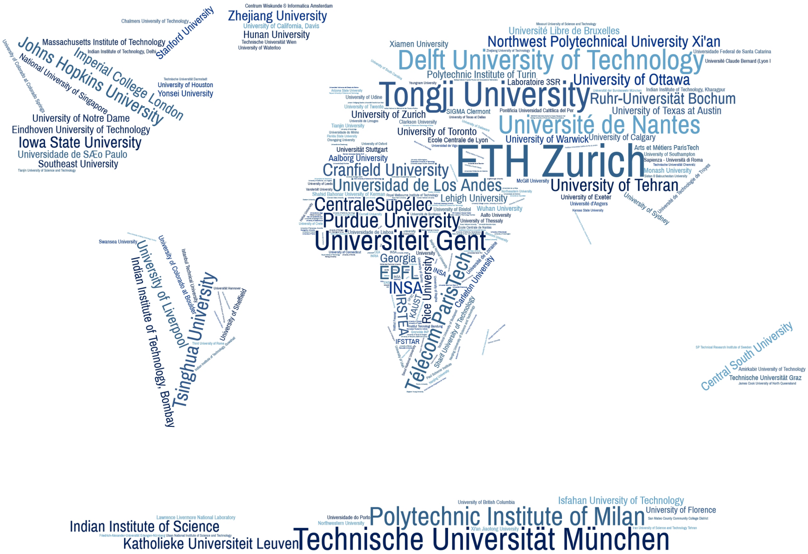
http://www.uqlab.com

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

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

Country	# Users
United States	636
China	590
France	371
Switzerland	309
Germany	286
United Kingdom	180
India	166
Brazil	158
Italy	153
Canada	94

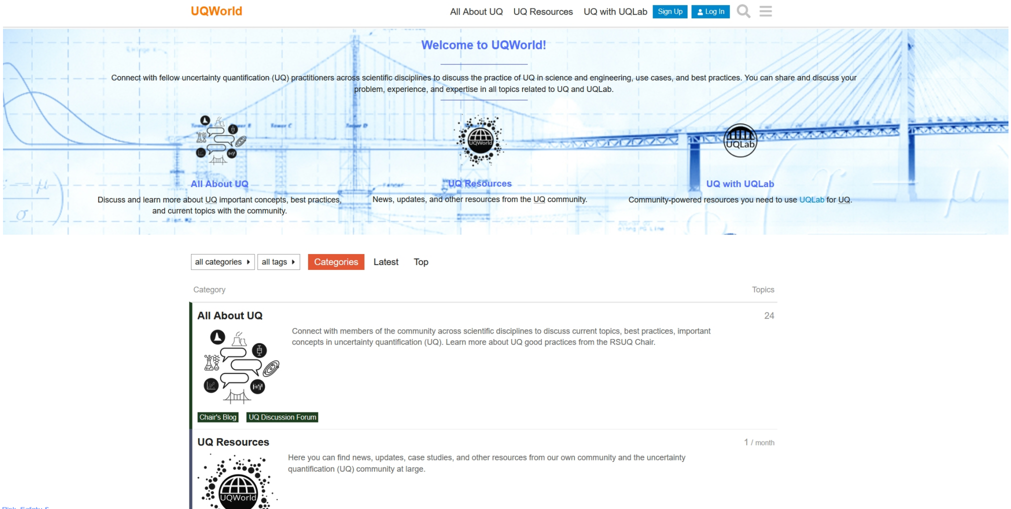
As of December 10, 2021



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

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