Surrogate models for forward and inverse uncertainty quantification

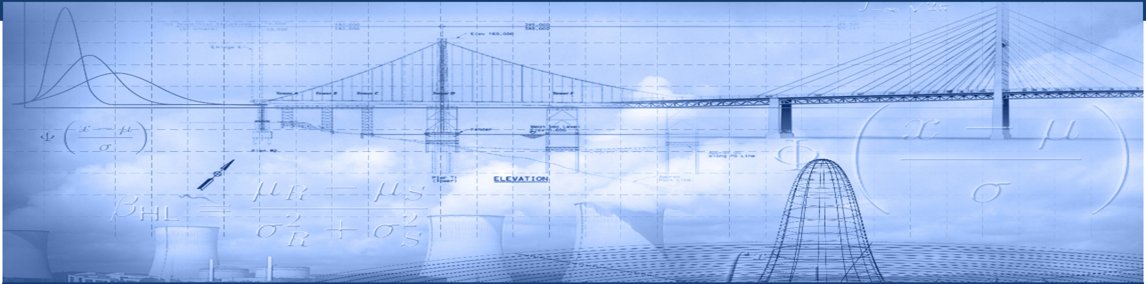
Presentation

Author(s):
Sudret, Bruno

Publication date:
2021-01-11

Permanent link:
https://doi.org/10.3929/ethz-b-000469582

Originally published in:
SSD Seminar Series



Surrogate models for forward and inverse uncertainty quantification

Bruno Sudret

Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich

How to cite?

This presentation is an invited lecture given at the RWTH Aachen University (Germany) in the International Research Training Group “Modern Inverse Problems” (IRTG-2379) on January 11, 2021.

How to cite

Sudret, B. *Surrogate models for forward and inverse uncertainty quantification*, International Research Training Group "Modern Inverse Problems", RWTH Aachen University (Germany), January 11th, 2021.

Chair of Risk, Safety and Uncertainty quantification

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

Research topics

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

http://www.rsuq.ethz.ch

Computational models in engineering

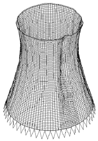
Complex engineering systems are designed and assessed using computational models, a.k.a simulators

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

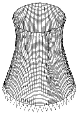


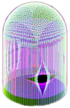
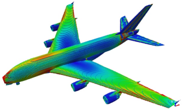
- Algorithms to solve the discretized equations

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



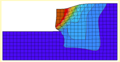



Computational models: the abstract viewpoint

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

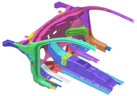
- Geometry
- Material properties
- Loading



- Analytical formula
- Finite element model
- Comput. workflow

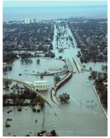
- Displacements
- Strains, stresses
- Temperature, etc.

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

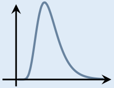
Model(s) of the system

Assessment criteria

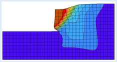
Step C

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)*

Step B: Quantification of the sources of uncertainty

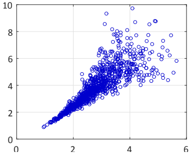
Goal: represent the uncertain parameters based on the available data and information

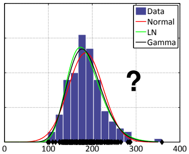
Probabilistic model $f_{\boldsymbol{X}}$

Experimental data is available

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

Copula theory

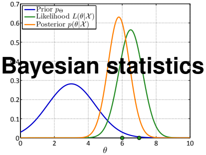




No data is available: expert judgment

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

Scarce data + expert information



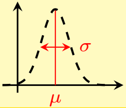
Step C: uncertainty propagation

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

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

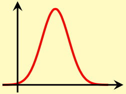
$$\mu_Y = \mathbb{E}_{\boldsymbol{X}}\left[\mathcal{M}(\boldsymbol{X})\right]$$
$$\sigma_Y^2 = \mathbb{E}_{\boldsymbol{X}}\left[\left(\mathcal{M}(\boldsymbol{X}) - \mu_Y\right)^2\right]$$

Mean/std. deviation



- Distribution of the QoI

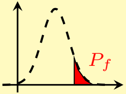
Response PDF



- Probability of exceeding an admissible threshold y_{adm}

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

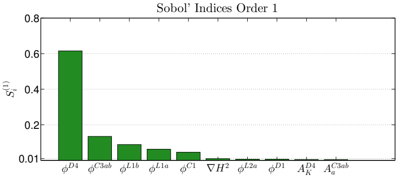
Probability of failure



Step C’: sensitivity analysis

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

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



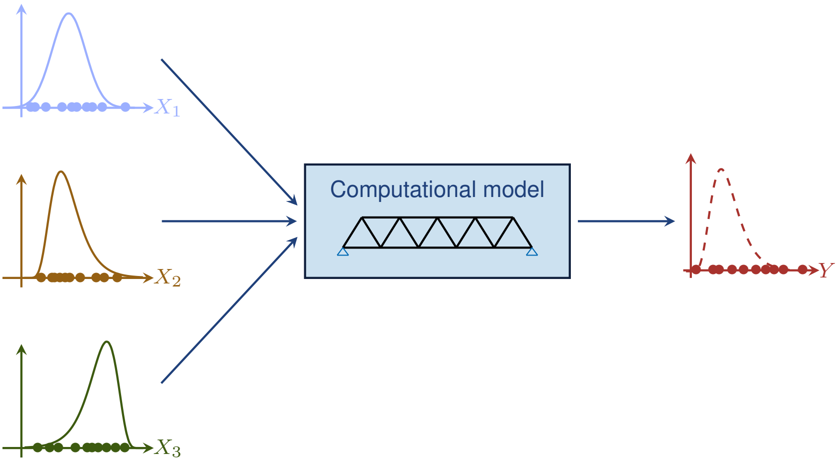
Variance decomposition (Sobol’ indices)

Uncertainty propagation using Monte Carlo simulation

Principle: Generate virtual prototypes of the system using random numbers

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

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 $\mathcal{X} = \left\{ \boldsymbol{x}^{(i)},\ i = 1, \, \dots \, , N \right\}$

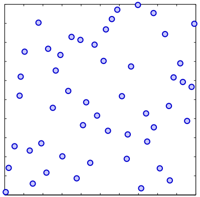
Simulated data

- 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: 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:

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

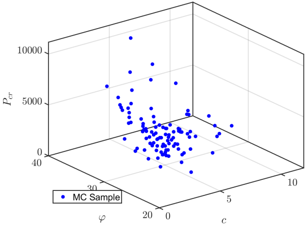
where:

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

Sampling (MCS) vs. spectral expansion (PCE)

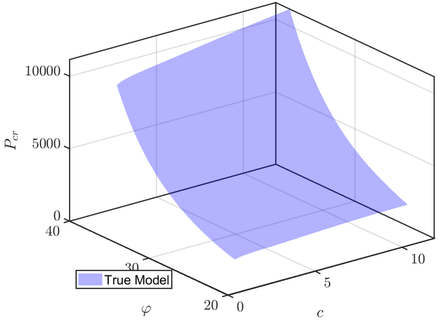
Whereas MCS explores the output space /distribution point-by-point, the polynomial chaos expansion assumes a generic structure (polynomial function), which better exploits the available information (runs of the original model)

Example: load bearing capacity as a function of $(c,\ \varphi)$



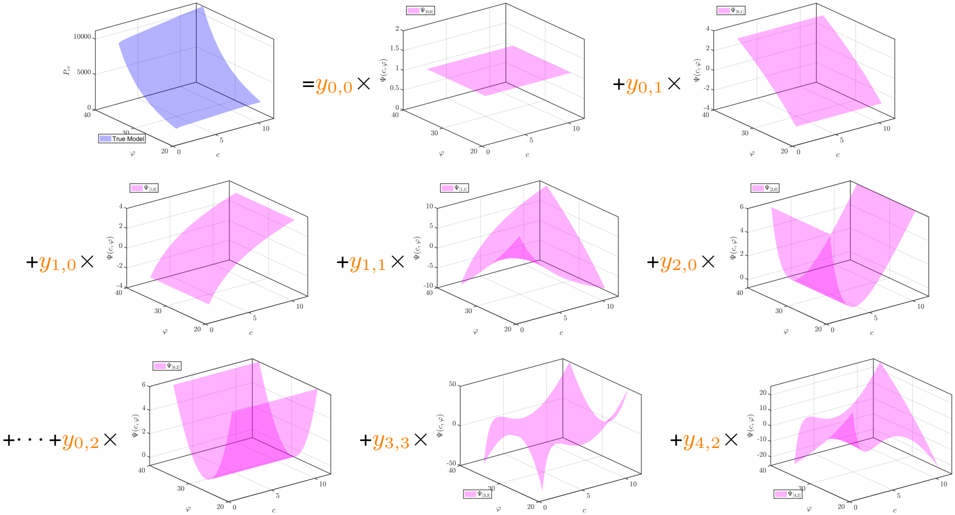
Thousands (resp. millions) of points are needed to grasp the structure of the response (resp. capture the rare events)

Visualization of the PCE construction



= “Sum of coefficients × 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

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

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

Dealing with complex input distributions

Independent variables

Input parameters with given marginal CDFs $X_i \sim F_{X_i}, \quad i = 1, \ldots, M$

- Arbitrary PCE: orthogonal polynomial computed numerically on-the-fly
- Isoprobabilistic transform through a one-to-one mapping to reduced variables, *e.g.* :

$$X_i = F_{X_i}^{-1}\left(\frac{\xi_i + 1}{2}\right) \qquad \text{if } \xi_i \sim \mathcal{U}(-1\,,\,1)$$

$$X_i = F_{X_i}^{-1}\left(\Phi(\xi_i)\right) \qquad \text{if } \xi_i \sim \mathcal{N}(0,1)$$

General case: addressing dependence

Sklar's theorem (1959)

- The joint CDF is defined through its marginals and copula

$$F_{\mathbf{X}}(\boldsymbol{x}) = \mathcal{C}\left(F_{X_1}(x_1), \ldots, F_{X_M}(x_M)\right)$$

- Rosenblatt or Nataf isoprobabilistic transform is used

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

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

Least-square minimization

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

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

Discrete (ordinary) least-square minimization

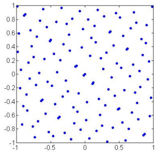
An estimate of the mean square error (sample average) is minimized:

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

Procedure

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

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



- Compute the experimental matrix

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

- Solve the resulting linear system

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

Simple is beautiful !

Error estimators

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

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

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

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

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

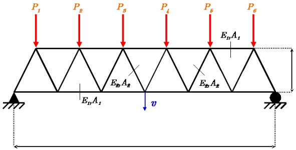
Outline

Curse of dimensionality

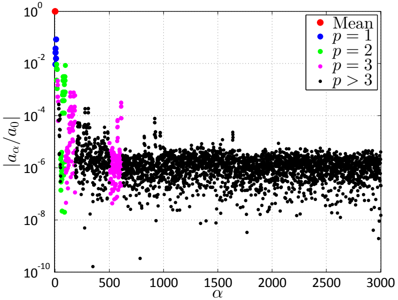
- The cardinality of the truncation scheme $\mathcal{A}^{M,p}$ is $P = \dfrac{(M+p)!}{M!\,p!}$
- Typical computational requirements: $n = OSR \cdot P$ where the oversampling rate is $OSR = 2 - 3$

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)

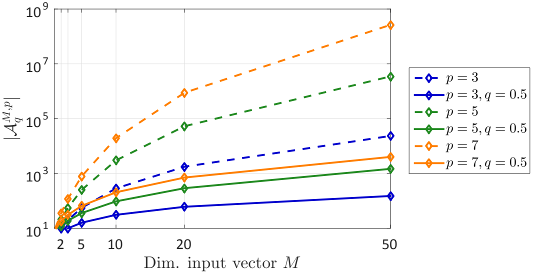
In most engineering problems, only low-order interactions between the input variables are relevant

- $q-$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

Least Angle Regression

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

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

Outline

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 $\mathbb{E}\left[\Psi_{\boldsymbol{\alpha}\neq\boldsymbol{0}}\right] = 0$ the statistical moments read:

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

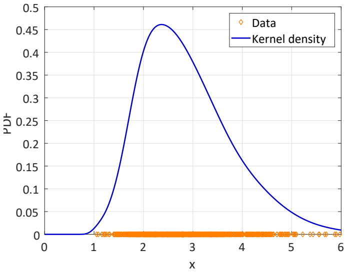
$$\text{Variance:} \qquad \hat{\sigma}_Y^2 = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\boldsymbol{0}} y_{\boldsymbol{\alpha}}^2$$

Distribution of the QoI

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

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

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



Sensitivity analysis

Goal

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

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

Hoeffding-Sobol' decomposition

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

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

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

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

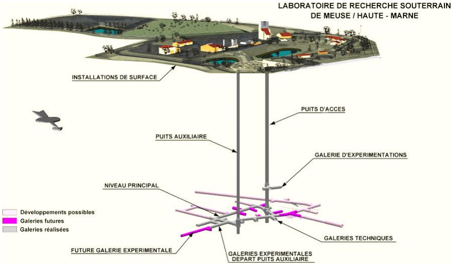
$$\mathcal{M}^{\text{PC}}(\boldsymbol{x}) = \mathcal{M}_0 + \sum_{\mathbf{u} \subset \{1, \ldots, M\}} \mathcal{M}_{\mathbf{u}}(\boldsymbol{x}_{\mathbf{u}}) \quad \text{where} \quad \mathcal{M}_{\mathbf{u}}(\boldsymbol{x}_{\mathbf{u}}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\mathbf{u}}}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

PC-based Sobol' indices

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

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

Example: sensitivity analysis in hydrogeology



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

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

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

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

Geological model

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

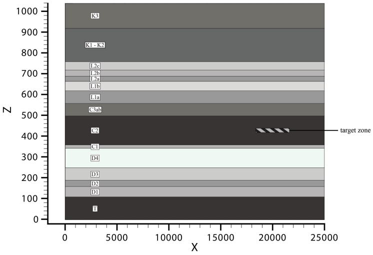
- Two-dimensional idealized model of the Paris Basin (25 km long / 1,040 m depth) with 5×5 m mesh $(10^6$ elements)
- Steady-state flow simulation with Dirichlet boundary conditions:

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

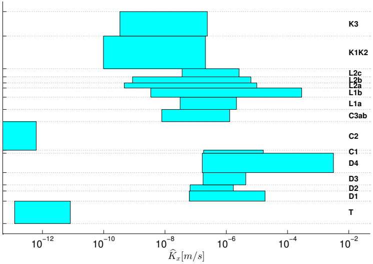
- 15 homogeneous layers with uncertainties in:
  - Porosity (resp. hydraulic conductivity)
  - Anisotropy of the layer properties (inc. dispersivity)
  - Boundary conditions (hydraulic gradients)

78 input parameters

Sensitivity analysis



Geometry of the layers



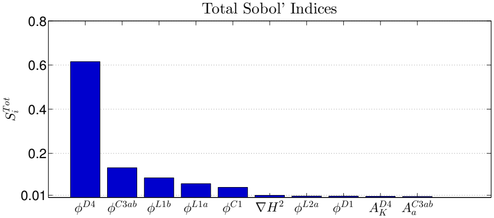
Conductivity of the layers

Question

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

Sensitivity analysis: results

Technique: Sobol'indices computed from polynomial chaos expansions



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

Conclusions

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

Outline

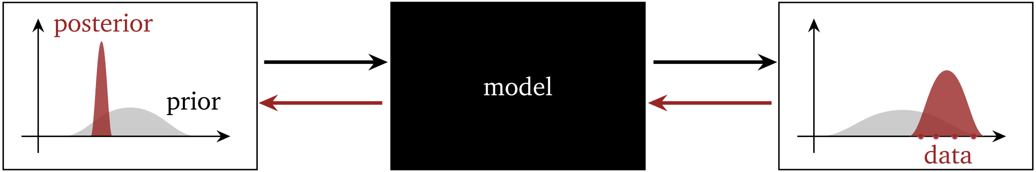
Framework

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

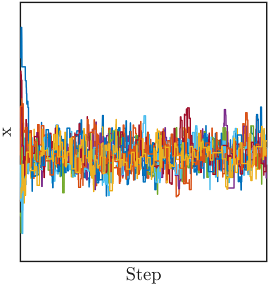


Markov-chain Monte Carlo

- Generally no analytical expression for $\pi(\boldsymbol{x}|\mathcal{Y})$ (exception: conjugate distributions)
- MCMC based approaches to generate posterior sample:

$$\boldsymbol{X}|\mathcal{Y} \sim \pi(\boldsymbol{x}|\mathcal{Y})$$

- Quantities of interest $\mathbb{E}\left[h(\boldsymbol{X})|\mathcal{Y}\right]$ estimated with this sample (e.g. posterior moments).
- Large number of MCMC algorithms (e.g. Metropolis-Hastings, Hamiltonian, affine invariant ensemble sampler)



Problems

- Require tuning & post-processing
- No clear convergence criterion
- Does not work well with multimodal posteriors
- Overall extremely computationally expensive.

MCMC + Surrogate models

- Forward model evaluations are the expensive part in solving inverse problems
- Solution:
  1. Train surrogate $\widehat{\mathcal{M}}$ of $\mathcal{M}$
  2. Formulate likelihood function $\mathcal{L}$ with $\widehat{\mathcal{M}}$
  3. Use conventional MCMC algorithms
- Speed up by orders of magnitude
- With PCE surrogate, additional sensitivity analysis for free

Problems

- Globally accurate surrogate might be inaccurate in posterior domain
- Still suffers from most MCMC problems (tuning, no convergence criterion, multimodality)

Towards a sampling-free inversion approach

Spectral likelihood embedding

Basic idea: expand the likelihood function onto a PCE

$$\mathcal{L}(\boldsymbol{X}) \approx \sum_{\boldsymbol{\alpha}\in\mathcal{A}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$$

The full posterior distribution (resp. quantities of interest) can be computed analytically: Nagel et al. (2016)

$$\hat{Z} = \mathbb{E}\left[\mathcal{L}(\boldsymbol{X})\right] = y_{\mathbf{0}}$$

$$\hat{\pi}(\boldsymbol{x}|\mathcal{Y}) = \frac{\pi(\boldsymbol{x})}{Z} \sum_{\boldsymbol{\alpha}\in\mathcal{A}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$$

$$\mathbb{E}\left[h(\boldsymbol{X})|\mathcal{Y}\right] = \frac{1}{Z} \sum_{\boldsymbol{\alpha}\in\mathcal{A}^k} y_{\boldsymbol{\alpha}} a_{\boldsymbol{\alpha}} \quad \text{with} \quad h(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} a_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$$

Requires extremely large truncated bases $\mathcal{A}$ to be accurate

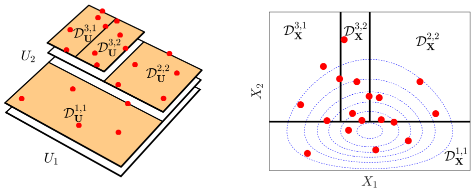
Outline

Stochastic spectral embedding - adaptive enrichment

$$\mathcal{L}(\boldsymbol{X}) \approx \sum_{k\in\mathcal{K}} \mathbf{1}_{\mathcal{D}^k_{\boldsymbol{X}}}(\boldsymbol{X})\mathcal{R}^k_{\mathrm{S}}(\boldsymbol{X}), \quad \text{where} \quad \mathcal{R}^k_{\mathrm{S}}(\boldsymbol{X}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{\alpha}\in\mathcal{A}^k} y^k_{\boldsymbol{\alpha}}\Psi^k_{\boldsymbol{\alpha}}(\boldsymbol{X}) \approx \mathcal{R}^k.$$

Sequential partitioning approach

- Adaptive experimental design enrichment
- Algorithm:
  1. initialize $\mathcal{K} = \{(0,1)\}$
  2. for refinement domain $k = \arg\max_{k\in\mathcal{T}}\{\mathcal{E}^k\}$ do:
     - 2.1 Partition domain $\mathcal{D}^k$ in half: $\mathcal{D}^{k_{\{1,2\}}}$
     - 2.2 Enrich experimental design $\mathcal{X}$
     - 2.3 Expand $\mathcal{R}^{k_{\{1,2\}}}$ to $\mathcal{R}_{\mathrm{S}}^{k_{\{1,2\}}}$
     - 2.4 Add $k_{\{1,2\}}$ to $\mathcal{K}$
  3. Stop if computational budget exhausted



Wagner et al. (2020)

Stochastic spectral likelihood embedding

After expanding the likelihood with SSLE as $\mathcal{L}(\boldsymbol{X}) \approx \sum_{k\in\mathcal{K}} \mathbf{1}_{\mathcal{D}^k_{\boldsymbol{X}}}(\boldsymbol{X})\mathcal{R}^k_{\mathrm{S}}(\boldsymbol{X})$, the full posterior distribution or the following quantities of interest can be computed analytically: Wagner et al. (2020)

$$Z = \mathbb{E}\left[\mathcal{L}(\boldsymbol{X})\right] \approx \sum_{k\in\mathcal{K}} \mathcal{V}^k y^k_{\boldsymbol{0}}$$

$$\pi(\boldsymbol{x}|\mathcal{Y}) \approx \frac{\pi(\boldsymbol{x})}{Z} \sum_{k\in\mathcal{K}} \mathbf{1}_{\mathcal{D}^k_{\boldsymbol{X}}}(\boldsymbol{x})\mathcal{R}^k_{\mathrm{S}}(\boldsymbol{x})$$

$$\mathbb{E}\left[h(\boldsymbol{X})|\mathcal{Y}\right] \approx \frac{1}{Z} \sum_{k\in\mathcal{K}} \mathcal{V}^k \cdot \sum_{\boldsymbol{\alpha}\in\mathcal{A}^k} y^k_{\boldsymbol{\alpha}} a^k_{\boldsymbol{\alpha}} \quad \text{after} \quad h(\boldsymbol{x}) \approx \sum_{\boldsymbol{\alpha}\in\mathcal{A}^k} a^k_{\boldsymbol{\alpha}} \Psi^k_{\boldsymbol{\alpha}}(\boldsymbol{x})$$

Example: Heat transfer problem

- Temperature measurements at 20 locations $\mathcal{Y} = \{T_1, \dots, T_N\}$
- Computational forward model solves the steady-state heat equation (FE-method):

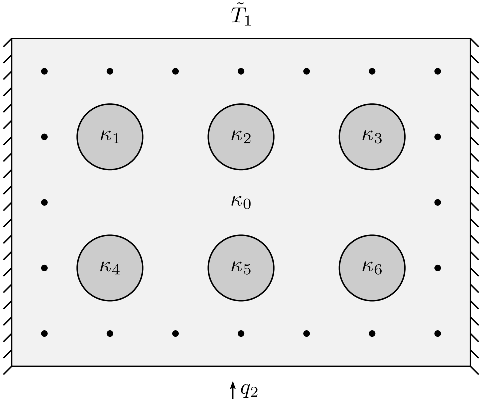
$$\nabla(\kappa \nabla T) = 0$$

- Likelihood with $\boldsymbol{\kappa} \stackrel{\text{def}}{=} (\kappa_1, \dots, \kappa_6)$:

$$\mathcal{L}(\boldsymbol{\kappa}; \mathcal{Y}) = \prod_{i=1}^{N} \mathcal{N}(T_i | \mathcal{M}(\boldsymbol{\kappa}), \sigma^2)$$

- Prior distributions:

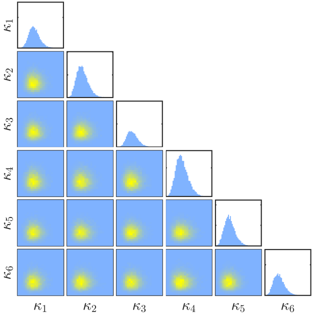
$$\pi(\boldsymbol{\kappa}) = \prod_{i=1}^{M} \mathcal{LN}(\mu = 30, \sigma = 6\ \mathrm{W/mK})$$



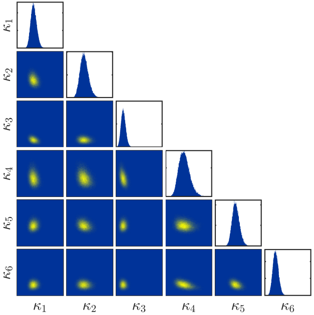
Heat transfer problem

A reference solution is obtained by MCMC (AIES, 10^5 $\mathcal{L}$ evaluations)

Prior Sample (MC)



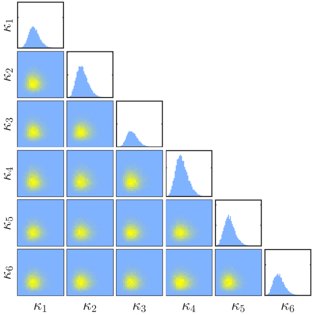
Posterior Sample (MCMC)



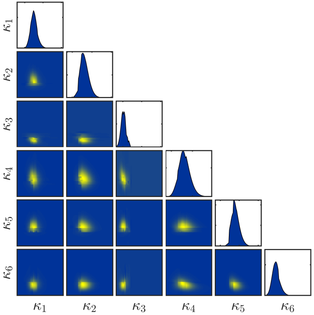
Heat transfer problem

. . . and compared to the SSLE solution (10^4 $\mathcal{L}$ evaluations)

Prior Sample (MC)



Posterior Distribution (SSE)



Wagner et al. (2021)

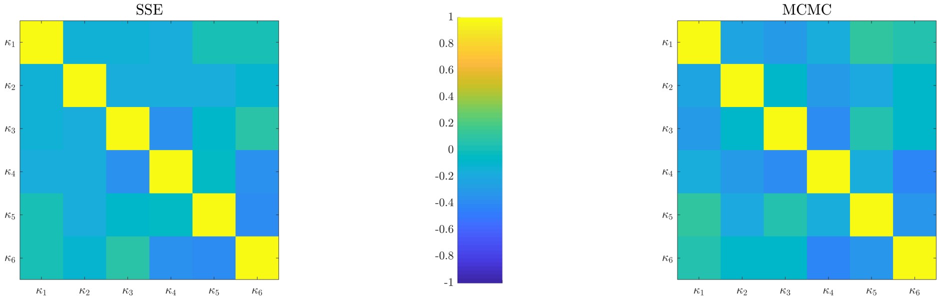
Posterior moments and correlations

SSE (10^4 $\mathcal{L}$ evaluations) vs. MCMC (10^5 $\mathcal{L}$ evaluations)

Moments

	κ_1	κ_2	κ_3	κ_4	κ_5	κ_6
$\mathbb{E}[\cdot \vert \mathcal{Y}]$ (W/mK)	30.1(29.8)	32.5(32.3)	20.7(20.7)	32.4(32.4)	36(36.4)	26.4(26.2)
$\mathrm{Var}[\cdot \vert \mathcal{Y}]$ ($\mathrm{W}^2/\mathrm{mK}^2$)	12.5(10.5)	17.4(17.9)	6.51(6.12)	27.9(26.8)	13.6(14.7)	12.8(9.02)

Correlations



Conclusions

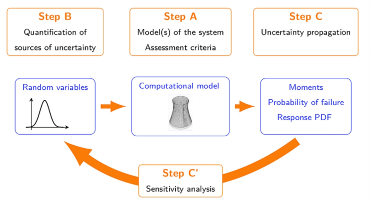
- Surrogate models are unavoidable for solving uncertainty quantification problems involving costly computational models (*e.g.* finite element models)
- Depending on the analysis, specific surrogates are most suitable: polynomial chaos expansions for distribution- and sensitivity analysis, Kriging for reliability analysis
- Bayesian inverse problems can be solved with surrogate modeling (stochastic spectral embedding), 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

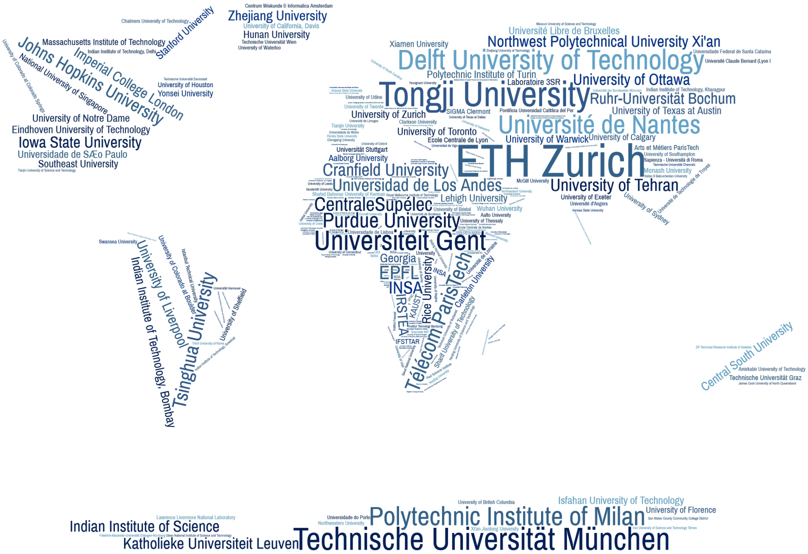
UQLab: The Uncertainty Quantification Software

http://www.uqlab.com

- ETH license:
  - + **free access to academia**
  - + yearly fee for non-academic usage
- 3,285 registered users
- 1,350+ active users from 87 countries
- About 37% license renewal after one year

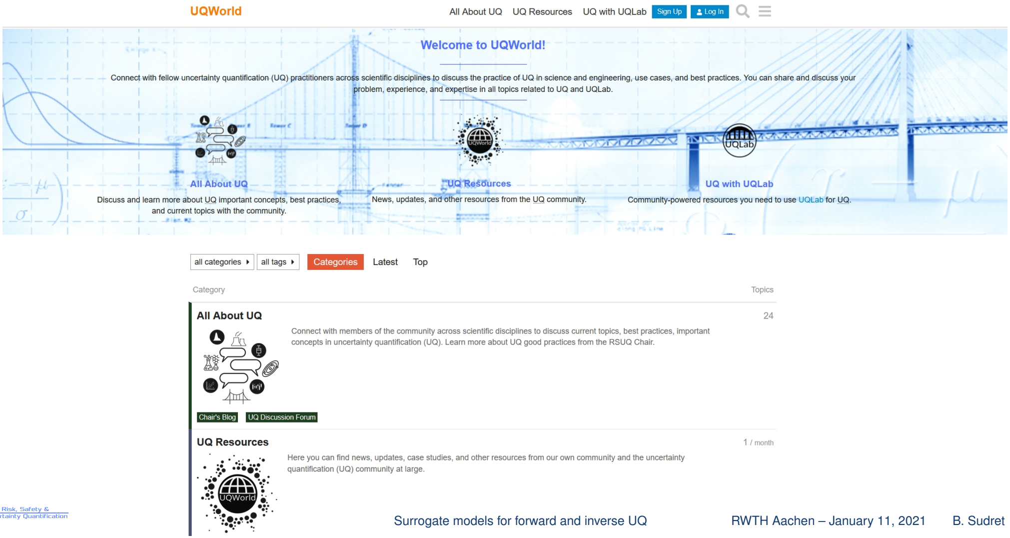
Country	# Users
United States	529
China	440
France	321
Switzerland	255
Germany	244
United Kingdom	146
Italy	125
Brazil	116
India	107
Canada	81

As of January 11, 2021

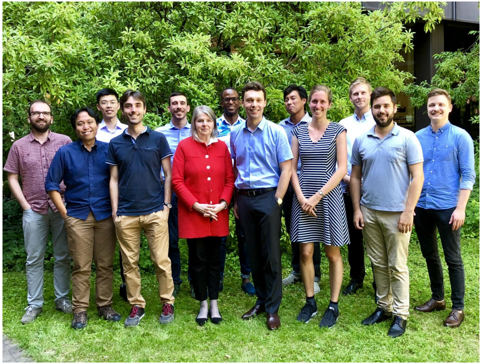

Stanford University
Zhejiang University
University of California, Davis
Hunan University
Technische Universität Wien
University of Waterloo
Massachusetts Institute of Technology
Indian Institute of Technology, Delhi
Johns Hopkins University
Imperial College London
National University of Singapore
University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

UQWorld: the community of UQ

https://uqworld.org/



Questions ?



Chair of Risk, Safety & Uncertainty Quantification

www.rsuq.ethz.ch

The Uncertainty Quantification Software

www.uqlab.com

The Uncertainty Quantification Community

www.uqworld.org