Surrogate Modelling and Uncertainty Quantification in Computational Sciences

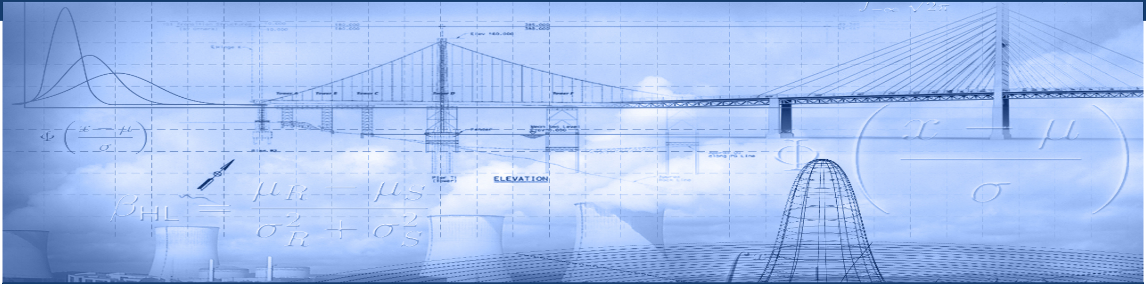
Presentation

Author(s):
Sudret, Bruno

Publication date:
2020-08-27

Permanent link:
https://doi.org/10.3929/ethz-b-000469598

ETH zürich

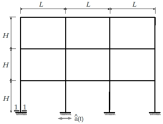


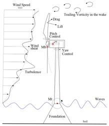
Surrogate Modelling and Uncertainty Quantification in Computational Sciences

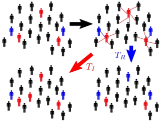
Bruno Sudret

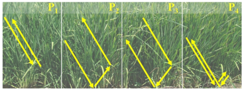
Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich

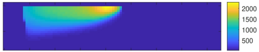
Introduction

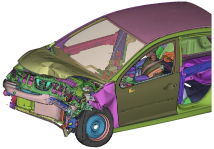


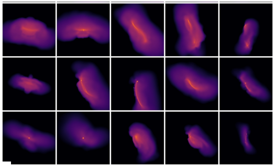


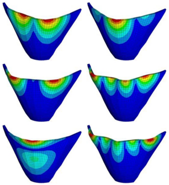


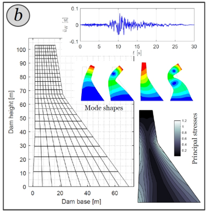


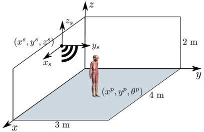












What is a computational model?

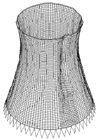
Complex natural or engineering systems are investigated / designed and assessed using computational models, a.k.a simulators

A computational model combines:

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

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

- Discretization techniques which transform continuous equations into linear algebra problems



- Algorithms to solve the discretized equations

Why do we use computational models?

- To better understand physical phenomena, *i.e.* test theories and assumptions against real-world observations

Model calibration

- To answer "what if?" questions: vary parameters within some ranges and see what happens

Parametric study

- To find out important parameters that drive the model predictions

Sensitivity analysis

Why do we use computational models (in engineering)?

- To explore the design space by creating virtual prototypes

Model exploration

- To optimize the system's performance (*e.g.* minimize its mass while ensuring certain behaviour)

Optimization

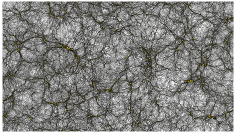
- To assess its robustness w.r.t uncertainties in the environmental & usage conditions

Uncertainty quantification / reliability

What about computational costs?

- Computer power has grown tremendously over the last decades (GigaFlops → TeraFlops → PetaFlops → ...)
- Modellers already use the available power for a single run
  e.g. "virtual universe simulation" by Teyssier et al.:
  $\sim$80 hours on 4,000$^+$ GPU nodes

Piz Daint Super Computer



Cosmic web (Image: J. Stadel)

How to carry out a parametric study / model exploration with:

- Costly simulators
- Complex input/output (nonlinear) behaviour
- High-dimensional input space

Surrogate models

Outline

Surrogate models

$$\text{Input } \boldsymbol{X} \longrightarrow \boxed{\begin{array}{c}\text{Computational model}\\ \mathcal{M}\end{array}} \longrightarrow \text{Output } \boldsymbol{Y} = \mathcal{M}(\boldsymbol{X})$$

A surrogate model $\tilde{\mathcal{M}}$ is an approximation of the original computational model $\mathcal{M}$ with the following features:

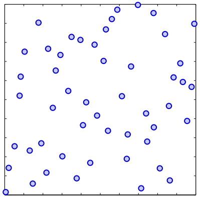
- It is built from a limited set of runs of the original model $\mathcal{M}$ called the experimental design $\mathcal{X} = \left\{ \boldsymbol{x}^{(i)}, \, i = 1, \, \dots \, , n \right\}$ that yield the model responses $\mathcal{Y} = \left\{ \boldsymbol{y}^{(i)} = \mathcal{M}\left(\boldsymbol{x}^{(i)}\right),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

Surrogate models: examples

Name	Shape	Parameters
Polynomial chaos expansions	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$	$\boldsymbol{a}_{\boldsymbol{\alpha}}$
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	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{m} a_i\, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\boldsymbol{a}\,,\ b$
Neural networks	$\tilde{\mathcal{M}}(\boldsymbol{x}) = f_2\left(b_2 + f_1\left(b_1 + \boldsymbol{w}_1 \cdot \boldsymbol{x}\right) \cdot \boldsymbol{w}_2\right)$	$\boldsymbol{w}, \boldsymbol{b}$
Low-rank tensor approximations	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{l=1}^{R} b_l \left(\prod_{i=1}^{M} v_l^{(i)}(x_i) \right)$	$b_l,\ z_{k,l}^{(i)}$

Ingredients for building a surrogate model

- Select an experimental design $\mathcal{X}$ that covers at best the domain of input parameters: Latin hypercube sampling (LHS), low-discrepancy sequences
- Run the computational model $\mathcal{M}$ onto $\mathcal{X}$



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

Wait, isn't it machine learning?

I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, MIT Press (2017)

- Machine learning aims at making predictions by building a model based on data
- Unsupervised learning aims at discovering a hidden structure within unlabelled data $\left\{\boldsymbol{x}^{(i)},\, i = 1, \ldots, n\right\}$
- Supervised learning considers a training data set:

$$\mathcal{X} = \left\{(\boldsymbol{x}^{(i)}, y^{(i)}),\, i = 1, \ldots, n\right\}$$

where:

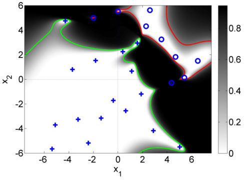
 - $\boldsymbol{x}^{(i)}$'s are the attributes / features (input space)
 - $y^{(i)}$'s are the labels (output space)

Wait, isn't it machine learning?

Classification

- In classification problems, the labels are discrete, *e.g.* $y^{(i)} \in \{-1, 1\}$. The goal is to predict the class of a new point $\boldsymbol{x}$

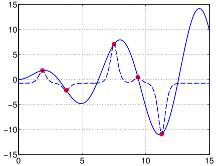
  Logistic regression - Support vector machines - (Deep) neural networks



Regression

- In regression problems, the labels are continuous, say $y^{(i)} \in \mathcal{D}_Y \subset \mathbb{R}$. The goal is to predict the value $\hat{y} = \tilde{\mathcal{M}}(\boldsymbol{x})$ for a new point $\boldsymbol{x}$

  Neural networks - Gaussian process models - Support vector regression



Bridging supervised learning and surrogate modelling

Features	Supervised learning	Surrogate modelling
Computational model $\mathcal{M}$	✗	✔
Input space $\boldsymbol{X} \sim f_{\boldsymbol{X}}$	✗	✔
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 $\boldsymbol{x} \notin \mathcal{X}$, $y(\boldsymbol{x})$?	$\sum_{i=1}^{m} y_i\, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	
Validation (resp. cross-validation)	✔ Validation set	✔ Leave-one-out CV

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
- Suited to high performance computing: "embarrassingly parallel"
- Similarities with big data analysis

Challenges

- Need for rigorous validation
- Communication: advanced mathematical background

Outline

Global framework for uncertainty quantification

Step B

Quantification of sources of uncertainty

Step A

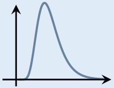
Model(s) of the system

Assessment criteria

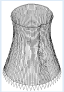
Step C

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

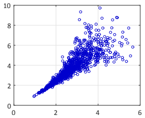
Goal: represent the uncertain parameters based on the available data and information

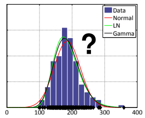
Probabilistic model $f_{\boldsymbol{X}}$

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

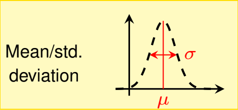
Bayesian statistics

Step C: uncertainty propagation

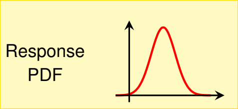
Goal: estimate the uncertainty / variability of the quantities of interest (QoI) $Y = \mathcal{M}(\boldsymbol{X})$ due to the input uncertainty $f_{\boldsymbol{X}}$

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

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

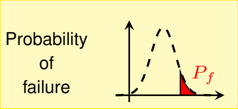


- Distribution of the QoI



- Probability of exceeding an admissible threshold y_{adm}

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



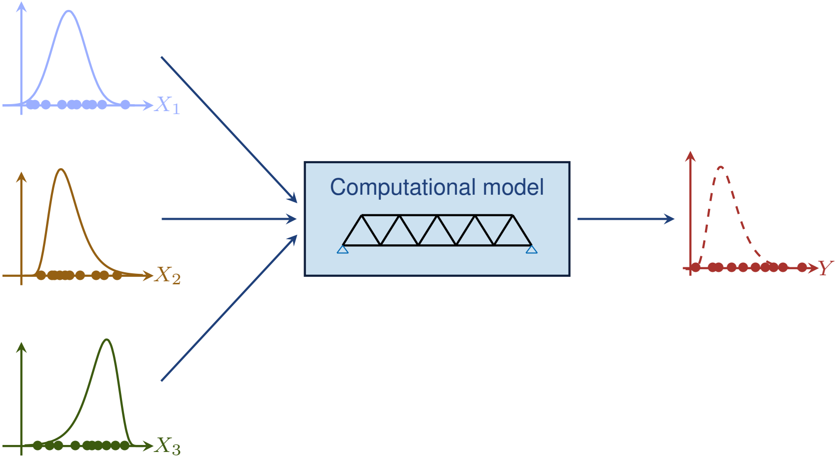
Uncertainty propagation using Monte Carlo simulation

Principle

Generate virtual realizations 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-, quantile- 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 $\propto n^{-1/2}$

Monte Carlo for reliability analysis

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

Need for surrogate models !

Outline

Polynomial chaos expansions in a nutshell

Ghanem & Spanos (1991); Sudret & Der Kiureghian (2000)
Xiu & Karniadakis (2002); Soize & Ghanem (2004)

- Input $\boldsymbol{X}$ with given PDF $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{M} f_{X_i}(x_i)$ (dim $\boldsymbol{X} = M$)
- Output $Y = \mathcal{M}(\boldsymbol{X})$ cast as the following polynomial chaos expansion:

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

where :

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

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

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \prod_{i=1}^{M} \Psi_{\alpha_i}^{(i)}(x_i)$$

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

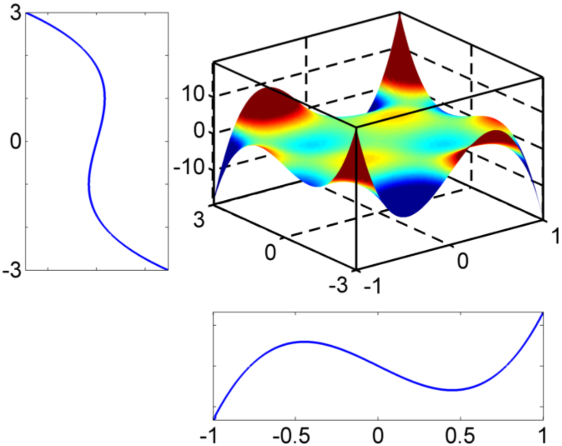
Tensor product construction

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

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

Multivariate polynomial basis $M = 2$

$\boldsymbol{\alpha} = [3\,,\,3]$ $\qquad \Psi_{(3,3)}(\boldsymbol{x}) = \tilde{P}_3(x_1) \cdot \tilde{He}_3(x_2)$



- $X_1 \sim \mathcal{U}(-1,1)$: Legendre polynomials
- $X_2 \sim \mathcal{N}(0,1)$: Hermite polynomials

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:

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

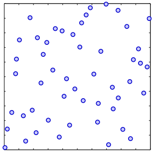
Discrete (ordinary) least-square minimization

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

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 !

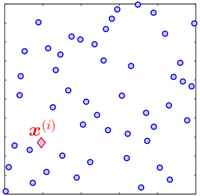
Validation: error estimators

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

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

Leave-one-out cross validation

- From statistical learning theory, model validation shall be carried out using independent data
- LOO cross-validation for PCE emulates it using all data at once



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

where h_i is the i-th diagonal term of matrix $\mathbf{A}(\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}$, $\mathbf{A}_{ij} = \Psi_j(\boldsymbol{x}^{(i)})$

Outline

Surrogate models

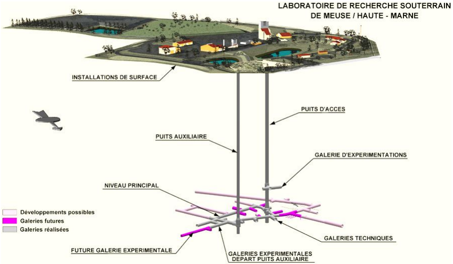
Basics of uncertainty quantification

Polynomial chaos expansions

Applications

- Subsurface flow
- Machine learning benchmarks

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)

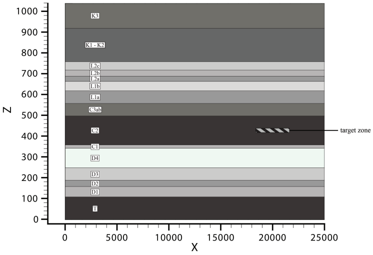
- Two-dimensional idealized model of the Paris Basin (25 km long / 1,040 m depth) with 5×5 m mesh $(10^6$ elements)
- Steady-state flow simulation with Dirichlet boundary conditions:

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

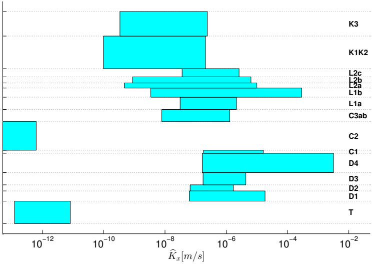
- 15 homogeneous layers with uncertainties in:
  - Porosity (resp. hydraulic conductivity)
  - Anisotropy of the layer properties (inc. dispersivity)
  - Boundary conditions (hydraulic gradients)

78 input parameters

Sensitivity analysis



Geometry of the layers



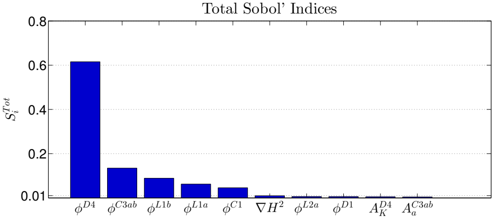
Conductivity of the layers

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

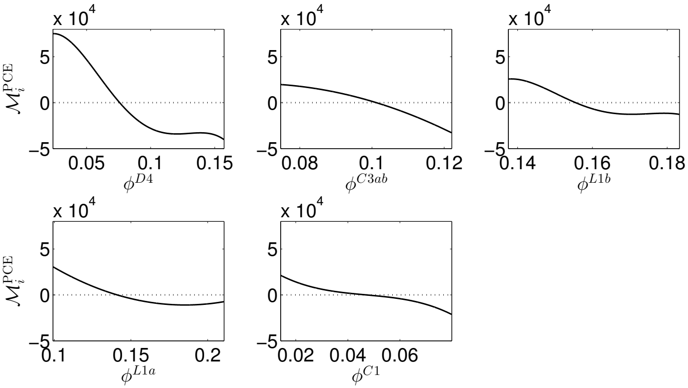
Conclusions

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

Bonus: univariate effects

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

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



Outline

Combined cycle power plant (CCPP)

Data set

UC Irvine Machine Learning Repository

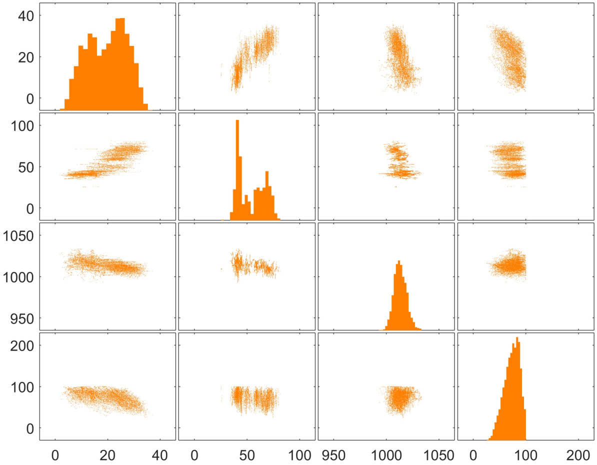
- 9,568 data points
- 4 features:
  - Temperature $T \in [1.81, 37.11]$ °C
  - Exhaust vacuum in the steam turbine $V \in [25.36, 81.56]$ cm Hg
  - Ambient pressure $P \in [992.89, 1033.30]$ mB
  - Relative humidity in the gas turbine $RH \in [25.56 - 100.16]\%$
- Output: net hourly electrical energy output $EP \in [420.26, 495.76]$ MW

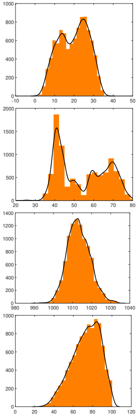
Reference approach

Tüfekci, P. (2014), *Int. J. Elec. Power & Energy Systems*

- 13 ML techniques including regression trees, ANN and SVR
- 10 pairs of training / validation sets of size 4,784
- Best approach: *bagging reduced error pruning (BREP) regression tree*

CCPP: Training data (X-space)





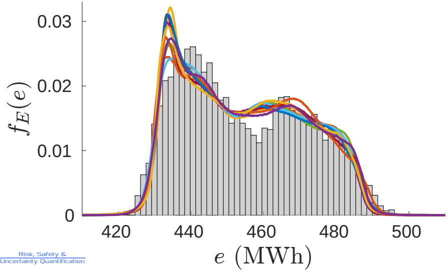
CCPP: Results

Relative mean absolute error

$$MAE = \frac{1}{n_{\text{val}}} \sum_{(\boldsymbol{x},y)\in\mathcal{X}_{\text{val}}} |y - \mathcal{M}^{\text{PC}}(\boldsymbol{x})|$$

	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.

† Tüfekci *et al.* (2014)



Estimated PDF of the energy produced by the CCPP:

- Histogram of raw data
- PDF obtained by PCE (10 diff. training sets) for input dependencies modelled by C-vines

Airfoil

Data set

UC Irvine Machine Learning Repository

- 750 training points, 750 validation points
- 41 features:
  - Frequency, in Hertz
  - Angle of attack, in degrees
  - Chord length, in meters
  - Free-stream velocity, in meters per second.
  - Suction side displacement thickness, in meters
  - 36 noise variables (standard normal)
- Output: Scaled sound pressure level, in decibels

Reference approach

K. Kandasamy & Y. Yu, ICML16 Proc. of the 33rd Int. Conf. on Machine Learning (2016)

- Sparse LASSO regression (SALSA)
- Beats 13 other regression models, incl. neural networks

Airfoil: Results

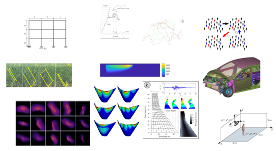
(Relative) mean absolute error (MAE)

	MAE (dB)	**rMAE (%)**
aPCEonX	3.04 ± 0.07	2.4 ± 0.06
SALSA†	3.81 ± 0.06	3.1 ± 0.04

† Kandasamy & Yu (2016)

Conclusions

- Surrogate models are unavoidable when dealing with costly computational models for uncertainty quantification, sensitivity analysis or optimization
- Depending on the analysis, specific surrogates are most suitable, *e.g.* polynomial chaos expansions for distribution- and sensitivity analysis, Kriging for reliability analysis
- All these techniques are non-intrusive: they rely on experimental designs, the size of which is a user's choice
- They are versatile, general-purpose and field-independent
- All the presented algorithms are available in the general-purpose uncertainty quantification software UQLab



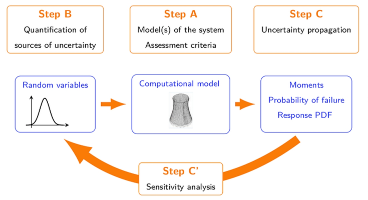
UQLab

The Framework for Uncertainty Quantification

OVERVIEW FEATURES DOCUMENTATION DOWNLOAD/INSTALL ABOUT COMMUNITY

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

www.uqlab.com

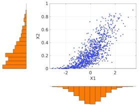


- MATLAB®-based Uncertainty Quantification framework
- State-of-the art, highly optimized open source algorithms
- Fast learning curve for beginners
- Modular structure, easy to extend
- Exhaustive documentation

UQLab features

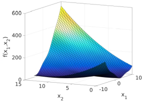
Probabilistic Input Modelling

- Common marginals
- Support for user-defined marginals
- Support for bounds on all distributions (including user-defined)
- Gaussian copula



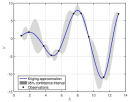
Modelling facilities

- Simple text strings
- MATLAB m-files
- MATLAB handles
- UQLINK: easily connect UQLAB to third party modelling software



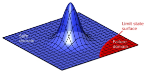
Advanced metamodelling

- Sparse degree-adaptive Polynomial Chaos Expansions
- Gaussian process modelling (Kriging)
- Polynomial-Chaos Kriging
- Low-rank tensor approximations
- Support vector machines



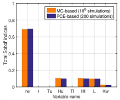
Reliability Analysis (Rare Event Estimation)

- FORM/SORM approximation
- Monte Carlo Simulation (MCS)
- Importance Sampling
- Subset Simulation
- Adaptive Kriging (AK-MCS)



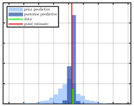
Sensitivity Analysis

- Correlation-based indices
- Standard Regression Coefficients
- Cotter measure
- Morris indices
- Sampling-based Sobol' indices
- PCE- and LRA-based Sobol' indices
- Borgonovo δ indices
- Support for dependent inputs



Bayesian inversion

- Intuitive problem statement
- Advanced MCMC algorithms
- Multi-model support (joint inversion)
- Support for custom likelihood



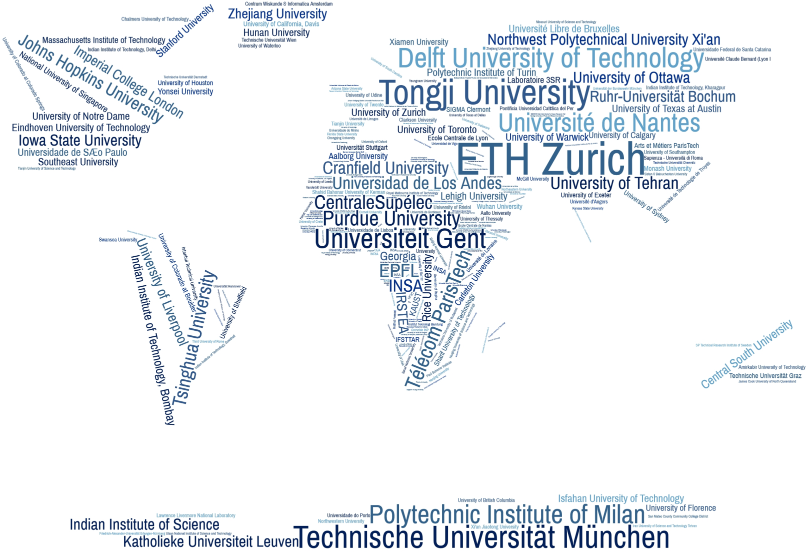
UQLab: The Uncertainty Quantification Software

http://www.uqlab.com

- ETH license:
  - free access to academia
  - yearly fee for non-academic usage
- 2,900+ registered users
- 1,280 active users from 87 countries
- About 37% license renewal after one year

Country	# Users
United States	493
China	365
France	301
Switzerland	238
Germany	221
United Kingdom	134
Italy	110
Brazil	96
India	88
Canada	77

As of August 24, 2020

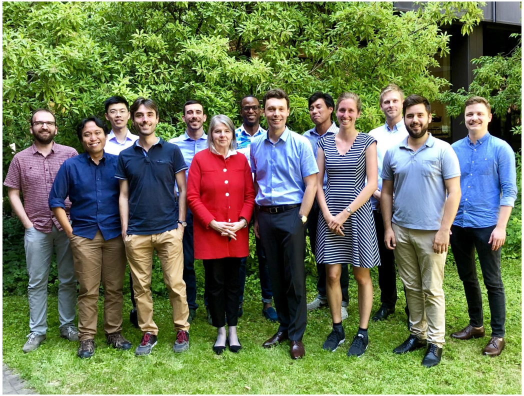


Chalmers University of Technology
Stanford University
Massachusetts Institute of Technology
Indian Institute of Technology, Delhi
Johns Hopkins University
Imperial College London
National University of Singapore
University of Colorado at Colorado Springs
University of Houston
Yonsei University
University 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
Indian Institute of Technology, Kharagpur
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
Royal Melbourne Institute of Technology
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
Universität Hannover
University of Sheffield
Georgia
EPFL
INSA
IRSTEA
KAUST
Rice University
Télécom ParisTech
Université de Lorraine
Carleton University
IFSTTAR
Sharif University of Technology
Central South University
Amirkabir University of Technology
Technische Universität Graz
University of British Columbia
Isfahan University of Technology
Lawrence Livermore National Laboratory
Indian Institute of Science
Universidade do Porto
Northwestern University
Polytechnic Institute of Milan
University of Florence
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