ETH zürich

Risk, Safety & Uncertainty Quantification

Méta-modèles pour la propagation d'incertitudes et l'analyse de sensibilité

Bruno Sudret

Chair of Risk, Safety and Uncertainty Quantification

ETH Zurich

Séminaire F2M – Ecole Nationale Supérieure des Arts et Métiers

16 Juin 2016

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

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

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.

$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

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.

$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

- Discretization techniques which transform continuous equations into linear algebra problems

- Algorithms to solve the discretized equations

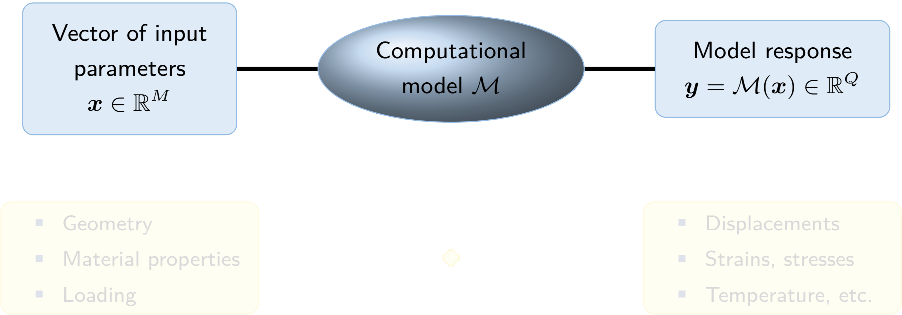
Computational models in engineering

Computational models are used:

- Together with experimental data for calibration purposes
- 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

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



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)

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)*

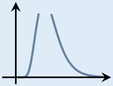
Global framework for uncertainty quantification

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)*

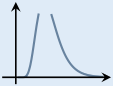
Global framework for uncertainty quantification

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)*

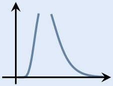
Global framework for uncertainty quantification

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)*

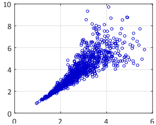
Step B: Quantification of the sources of uncertainty

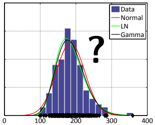
Goal: represent the uncertain parameters based on the *available data and information*

Experimental data is available

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

Copula theory





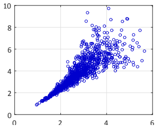
Step B: Quantification of the sources of uncertainty

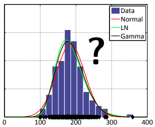
Goal: represent the uncertain parameters based on the *available data and information*

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)

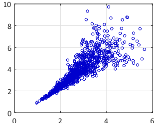
Step B: Quantification of the sources of uncertainty

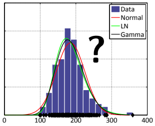
Goal: represent the uncertain parameters based on the *available data and information*

Experimental data is available

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

Copula theory

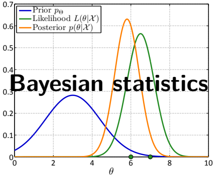




No data is available: expert judgment

- 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 B: Quantification of the sources of uncertainty

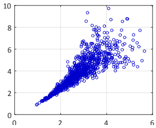
Goal: represent the uncertain parameters based on the *available data and information*

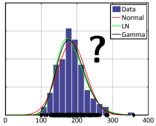
Probabilistic model $f_{\boldsymbol{X}}$

Experimental data is available

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

Copula theory

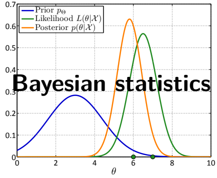




No data is available: expert judgment

- 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



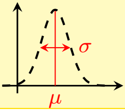
Step C: uncertainty propagation

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

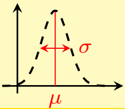
Step C: uncertainty propagation

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.

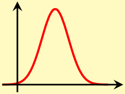
$$\mu_Y = \mathbb{E}_{\boldsymbol{X}}\left[\mathcal{M}(\boldsymbol{X})\right]$$
$$\sigma_Y^2 = \mathbb{E}_{\boldsymbol{X}}\left[\left(\mathcal{M}(\boldsymbol{X}) - \mu_Y\right)^2\right]$$

Mean/std. deviation



- Distribution of the QoI

Response PDF



- Probability of exceeding an admissible threshold y_{adm}

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

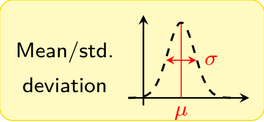
Probability of failure

Step C: uncertainty propagation

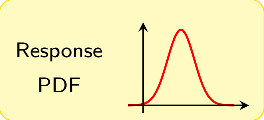
Goal: estimate the uncertainty / variability of the quantities of interest (QoI) $Y = \mathcal{M}(\boldsymbol{X})$ due to the input uncertainty $f_{\boldsymbol{X}}$

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

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

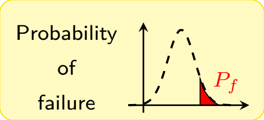


- Distribution of the QoI



- Probability of exceeding an admissible threshold y_{adm}

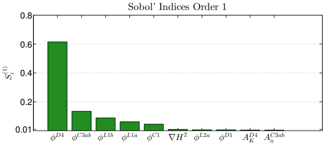
$$P_f = \mathbb{P}\left(Y \geq y_{adm}\right)$$



Step C': sensitivity analysis

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

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

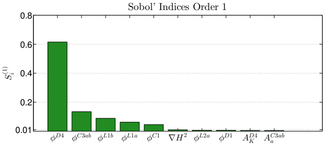


Variance decomposition (Sobol' indices)

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

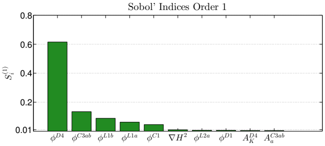


Variance decomposition (Sobol' indices)

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

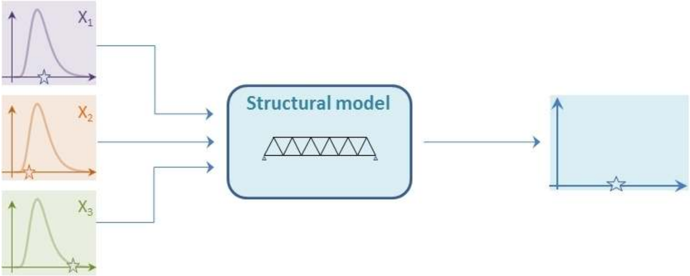
Principle: Generate virtual prototypes of the system using random numbers

- 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 quantities of interest is used for moments-, distribution- or reliability analysis

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 $\mathcal{Y} = \{\mathcal{M}(\boldsymbol{x}_1), \dots, \mathcal{M}(\boldsymbol{x}_n)\}$

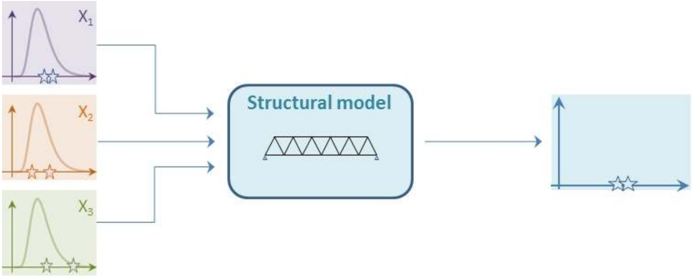


- The set of quantities of interest is used for moments-, distribution- or reliability analysis

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 $\mathcal{Y} = \{\mathcal{M}(\boldsymbol{x}_1), \ldots, \mathcal{M}(\boldsymbol{x}_n)\}$

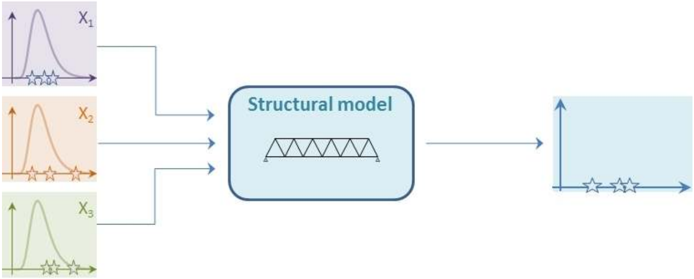


- The set of quantities of interest is used for moments-, distribution- or reliability analysis

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 $\mathcal{Y} = \{\mathcal{M}(\boldsymbol{x}_1), \ldots, \mathcal{M}(\boldsymbol{x}_n)\}$

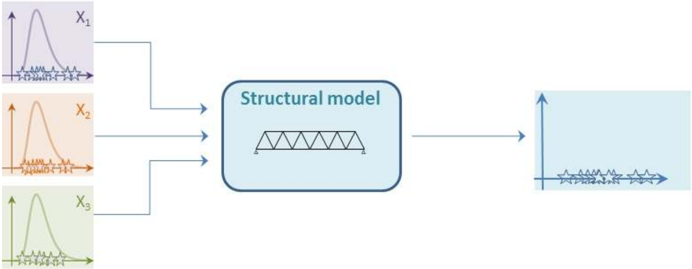


- The set of quantities of interest is used for moments-, distribution- or reliability analysis

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 $\mathcal{Y} = \{\mathcal{M}(\boldsymbol{x}_1), \ldots, \mathcal{M}(\boldsymbol{x}_n)\}$

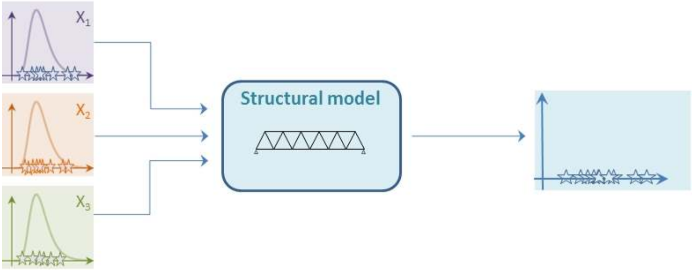


- The set of quantities of interest is used for moments-, distribution- or reliability analysis

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 quantities of interest is used for moments-, distribution- or reliability analysis

Advantages/Drawbacks of Monte Carlo simulation

Advantages

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

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 $N_{MCS} \to \infty$
- 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 $\propto n^{-1/2}$

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 $N_{MCS} \to \infty$
- 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 $\pm10\%$ (coef. of variation of 5%), $4 \cdot 10^{k+2}$ runs are required

Surrogate models for uncertainty quantification

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

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

Surrogate models for uncertainty quantification

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

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

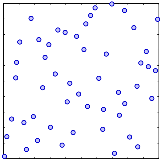
Surrogate models for uncertainty quantification

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

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

Name	Shape	Parameters
Polynomial chaos expansions	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}}\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$	$\boldsymbol{a}_{\boldsymbol{\alpha}}$
Low-rank tensor approximations	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{l=1}^{R} b_l \left(\prod_{i=1}^{M} v_l^{(i)}(x_i) \right)$	$b_l,\, z_{k,l}^{(i)}$
Kriging (a.k.a Gaussian processes)	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \boldsymbol{\beta}^{\mathsf{T}} \cdot \boldsymbol{f}(\boldsymbol{x}) + Z(\boldsymbol{x}, \omega)$	$\boldsymbol{\beta}\,,\, \sigma_Z^2\,,\, \boldsymbol{\theta}$
Support vector machines	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{m} a_i\, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\boldsymbol{a}\,,\, b$

Ingredients for building a surrogate model

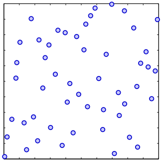


- Select an experimental design $\mathcal{X}$ that covers at best the domain of input parameters: Latin hypercube sampling (LHS), low-discrepancy sequences
- Run the computational model $\mathcal{M}$ onto $\mathcal{X}$ exactly as in Monte Carlo simulation
- Smartly post-process the data $\{\mathcal{X}, \mathcal{M}(\mathcal{X})\}$ through a learning algorithm

Name	Learning method
Polynomial chaos expansions	sparse grid integration, least-squares, compressive sensing
Low-rank tensor approximations	alternate least squares
Kriging	maximum likelihood, Bayesian inference
Support vector machines	quadratic programming

Ingredients for building a surrogate model

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



- Run the computational model $\mathcal{M}$ onto $\mathcal{X}$ exactly as in Monte Carlo simulation
- Smartly post-process the data $\{\mathcal{X}, \mathcal{M}(\mathcal{X})\}$ through a learning algorithm

Name	**Learning method**
Polynomial chaos expansions	sparse grid integration, least-squares, compressive sensing
Low-rank tensor approximations	alternate least squares
Kriging	maximum likelihood, Bayesian inference
Support vector machines	quadratic programming

Advantages of surrogate models

Usage

$$\mathcal{M}(\boldsymbol{x}) \quad \approx \quad \tilde{\mathcal{M}}(\boldsymbol{x})$$

hours per run — seconds for 10^6 runs

Advantages

- Non-intrusive methods: based on runs of the computational model, exactly as in Monte Carlo simulation
- Suited to high performance computing: "embarrassingly parallel"

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

Advantages of surrogate models

Usage

$$\mathcal{M}(\boldsymbol{x}) \quad \approx \quad \tilde{\mathcal{M}}(\boldsymbol{x})$$

hours per run — seconds for 10^6 runs

Advantages

- Non-intrusive methods: based on runs of the computational model, exactly as in Monte Carlo simulation
- Suited to high performance computing: "embarrassingly parallel"

Challenges

- Need for rigorous validation
- Communication: advanced mathematical background

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

Outline

Polynomial chaos expansions in a nutshell

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

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

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

  where :

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

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

Multivariate polynomial basis

Univariate polynomials

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

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

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

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

Tensor product construction

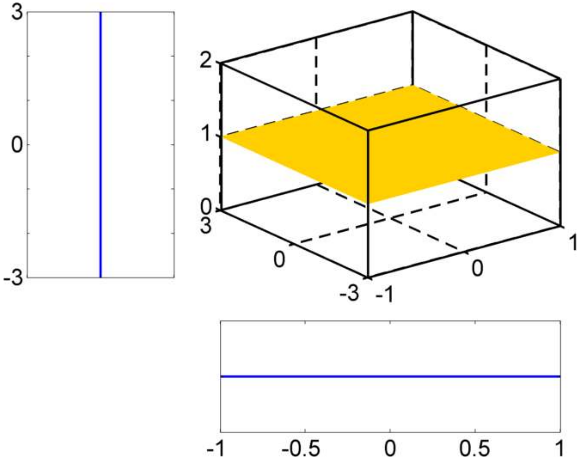
$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \overset{\text{def}}{=} \prod_{i=1}^{M} \Psi_{\alpha_i}^{(i)}(x_i) \qquad \mathbb{E}\left[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})\Psi_{\boldsymbol{\beta}}(\boldsymbol{X})\right] = \delta_{\boldsymbol{\alpha}\boldsymbol{\beta}}$$

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

Example: $M = 2$

Xiu & Karniadakis (2002)

$\boldsymbol{\alpha} = [0\,,\,0]$ $\Psi_{(0,0)}(\boldsymbol{x}) = \tilde{P}_0(x_1) \cdot \tilde{He}_0(x_2)$

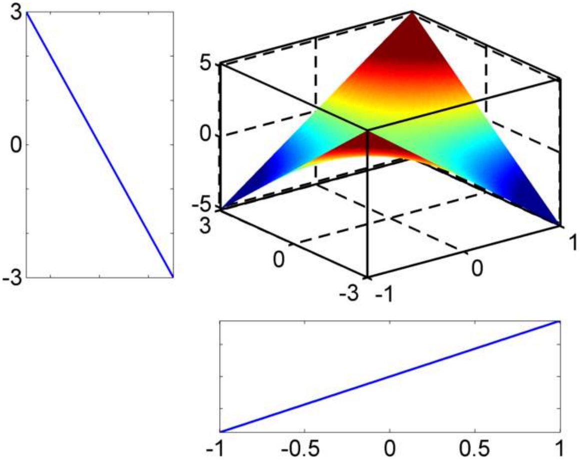


- $X_1 \sim \mathcal{U}(-1, 1)$: Legendre polynomials
- $X_2 \sim \mathcal{N}(0, 1)$: Hermite polynomials

Example: $M = 2$

Xiu & Karniadakis (2002)

$\boldsymbol{\alpha} = [1\ ,\ 1]$ $\qquad \Psi_{(1,1)}(\boldsymbol{x}) = \tilde{P}_1(x_1) \cdot \tilde{He}_1(x_2)$

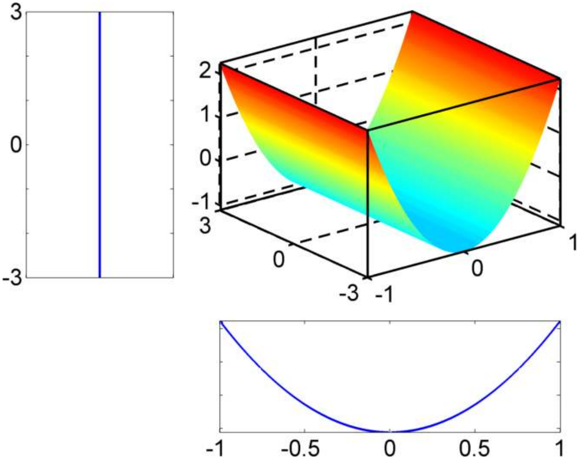


- $X_1 \sim \mathcal{U}(-1, 1)$: Legendre polynomials
- $X_2 \sim \mathcal{N}(0, 1)$: Hermite polynomials

Example: $M = 2$

Xiu & Karniadakis (2002)

$\boldsymbol{\alpha} = [2\,,\,0]$ $\quad \Psi_{(2,0)}(\boldsymbol{x}) = \tilde{P}_2(x_1) \cdot \tilde{He}_0(x_2)$

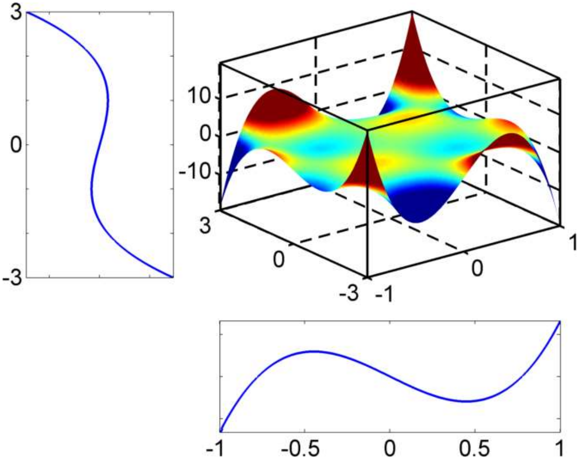


- $X_1 \sim \mathcal{U}(-1, 1)$: Legendre polynomials
- $X_2 \sim \mathcal{N}(0, 1)$: Hermite polynomials

Example: $M = 2$

Xiu & Karniadakis (2002)

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



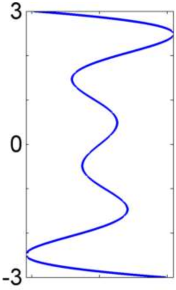
- $X_1 \sim \mathcal{U}(-1, 1)$: Legendre polynomials
- $X_2 \sim \mathcal{N}(0, 1)$: Hermite polynomials

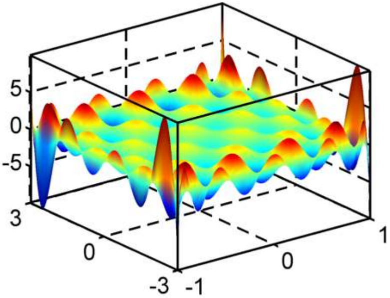
Example: $M = 2$

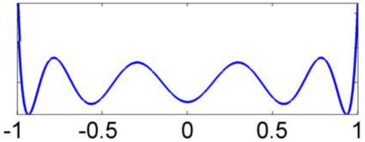
Xiu & Karniadakis (2002)

$\boldsymbol{\alpha} = [10\ ,\ 11]$

$$\Psi_{(10,11)}(\boldsymbol{x}) = \tilde{P}_{10}(x_1) \cdot \tilde{He}_{11}(x_2)$$







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

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:

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

where : $\mathsf{Y} = \{y_{\boldsymbol{\alpha}},\, \boldsymbol{\alpha} \in \mathcal{A}\} \equiv \{y_0, \, \dots \, , y_{P-1}\}$ (P unknown coef.)

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

Least-square minimization

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

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

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 coef.)

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

Least-square minimization

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

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

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

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

Discrete (ordinary) least-square minimization

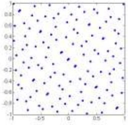
An estimate of the mean square error (sample average) is minimized:

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

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

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

Discrete (ordinary) least-square minimization

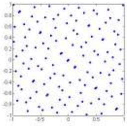
An estimate of the mean square error (sample average) is minimized:

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

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

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

Discrete (ordinary) least-square minimization

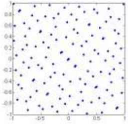
An estimate of the mean square error (sample average) is minimized:

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

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

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

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

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

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

Least-squares analysis: Wrap-up

Algorithm 1: Ordinary least-squares

Input: Computational budget n
Initialization
    Experimental design $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$
    Run model $\mathcal{Y} = \{\mathcal{M}(\boldsymbol{x}^{(1)}), \ldots, \mathcal{M}(\boldsymbol{x}^{(n)})\}$
PCE construction
    for $p = p_{\min} : p_{\max}$ **do**
        Select candidate basis $\mathcal{A}^{M,p}$
        Solve OLS problem
        Compute $e_{\mathsf{LOO}}(p)$
    end
    $p^* = \arg\min e_{\mathsf{LOO}}(p)$
Return Best PCE of degree p^*

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 $OSR = 2 - 3$

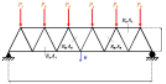
However ... most coefficients are close to zero !

Curse of dimensionality

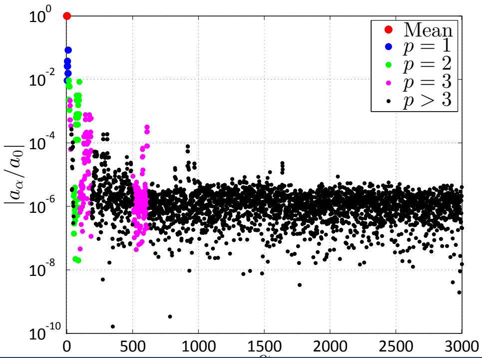
- The cardinality of the truncation scheme $\mathcal{A}^{M,p}$ is $P = \dfrac{(M+p)!}{M!\,p!}$
- Typical computational requirements: $n = OSR \cdot P$ where the oversampling rate is $OSR = 2 - 3$

However ... most coefficients are close to zero !

Example



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



Hyperbolic truncation sets

Sparsity-of-effects principle

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

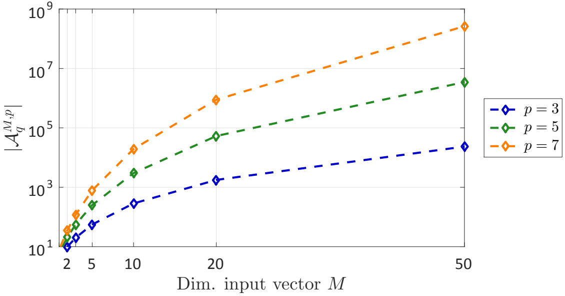
In most engineering problems, only low-order interactions between the input variables are relevant

- $q-$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:

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



Hyperbolic truncation sets

Sparsity-of-effects principle

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

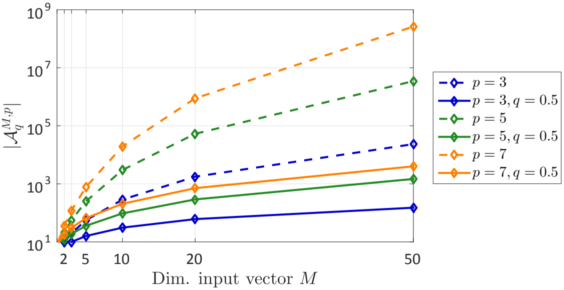
In most engineering problems, only low-order interactions between the input variables are relevant

- $q-$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); Ian, Guo, Xiu (2012); 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

Compressive sensing approaches

Blatman & Sudret (2011); Doostan & Owhadi (2011); Ian, Guo, Xiu (2012); 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

Sparse PCE: wrap up

Algorithm 2: LAR-based Polynomial chaos expansion

Input: Computational budget n
Initialization
    Sample experimental design $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\}$
    Evaluate model response $\mathcal{Y} = \{\mathcal{M}(\boldsymbol{x}^{(1)}), \dots, \mathcal{M}(\boldsymbol{x}^{(n)})\}$
PCE construction
    for $p = p_{\min} : p_{\max}$ **do**
        for $q \in \mathcal{Q}$ **do**
            Select candidate basis $\mathcal{A}_q^{M,p}$
            Run LAR for extracting the optimal sparse basis $\mathcal{A}^*(p,q)$
            Compute coefficients $\{y_{\boldsymbol{\alpha}},\ \boldsymbol{\alpha} \in \mathcal{A}^*(p,q)\}$ by OLS
            Compute $e_{\mathrm{LOO}}(p,q)$
        end
    end
    $(p^*, q^*) = \arg\min e_{\mathrm{LOO}}(p,q)$
Return Optimal sparse basis $\mathcal{A}^*(p,q)$, PCE coefficients, $e_{\mathrm{LOO}}(p^*, q^*)$

Outline

Post-processing sparse PC expansions

Statistical moments

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

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

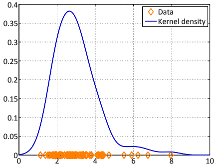
$$\text{Variance:} \qquad \hat{\sigma}_Y^2 = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} y_{\boldsymbol{\alpha}}^2$$

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.* (2000)

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

Hoeffding-Sobol' decomposition

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

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

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

- The summands satisfy the orthogonality condition:

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

Sobol' indices

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

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

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

PC-based Sobol' indices

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

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

Example: sensitivity analysis in hydrogeology

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

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

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

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

Geological model

Joint work with University of Neuchâtel

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

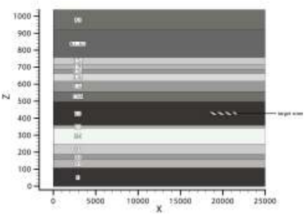
- Two-dimensional idealized model of the Paris Basin (25 km long / 1,040 m depth) with 5×5 m mesh (10^6 elements)
- Steady-state flow simulation with Dirichlet boundary conditions:

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

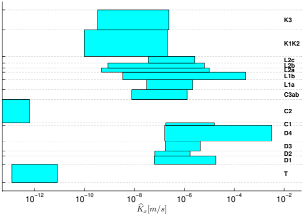
- 15 homogeneous layers with uncertainties in:
  - Porosity (resp. hydraulic conductivity)
  - Anisotropy of the layer properties (inc. dispersivity)
  - Boundary conditions (hydraulic gradients)

78 input parameters

Sensitivity analysis



Geometry of the layers



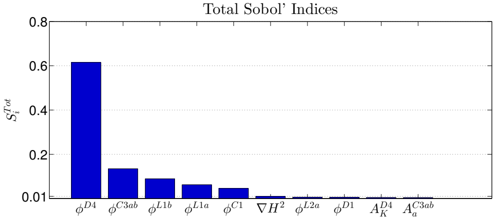
Conductivity of the layers

Question

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

Sensitivity analysis: results

Technique: Sobol'indices computed from polynomial chaos expansions



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

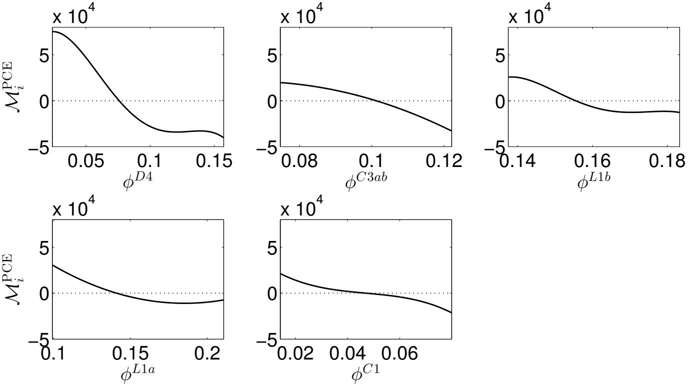
Conclusions

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

Bonus: univariate effects

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

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

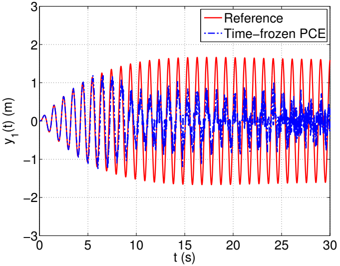


Polynomial chaos expansions in structural dynamics

Spiridonakos et al. (2015); Mai & Sudret, ICASP'2015; Mai *et al.* , 2016

Premise

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



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

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

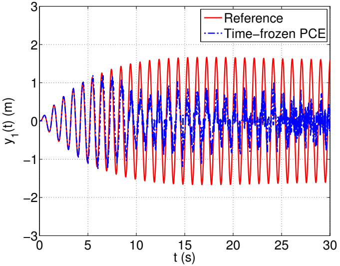
- Expand the NARX coefficients of different random trajectories onto a PCE basis

Polynomial chaos expansions in structural dynamics

Spiridonakos et al. (2015); Mai & Sudret, ICASP'2015; Mai *et al.* , 2016

Premise

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



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

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

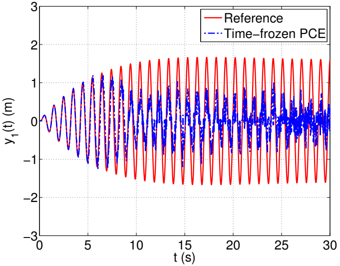
- Expand the NARX coefficients of different random trajectories onto a PCE basis

Polynomial chaos expansions in structural dynamics

Spiridonakos et al. (2015); Mai & Sudret, ICASP'2015; Mai *et al.* , 2016

Premise

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



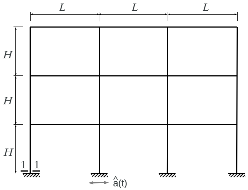
- Use of non linear autoregressive with exogenous input models (NARX) to capture the dynamics:

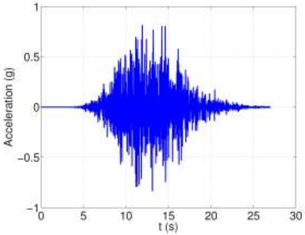
$$y(t) = \mathcal{F}\left(x(t), \ldots, x(t - n_x), y(t-1), \ldots, y(t - n_y)\right) + \epsilon_t$$

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

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

Application: earthquake engineering



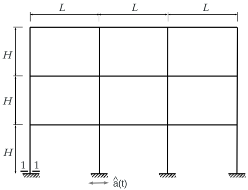


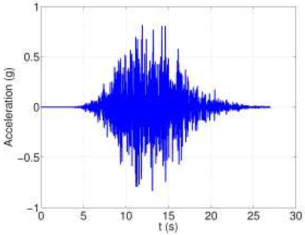
Rezaeian & Der Kiureghian (2010)

- 2D steel frame with uncertain properties submitted to synthetic ground motions
- Experimental design of size 300
- Ground motions obtained from modulated, filtered white noise

$$x(t) = q(t, \boldsymbol{\alpha}) \sum_{i=1}^{n} s_i \left(t, \boldsymbol{\lambda}(t_i)\right) \cdot \xi_i \qquad \xi_i \sim \mathcal{N}(0,1)$$

Application: earthquake engineering





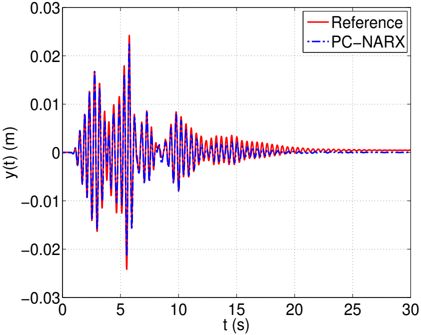
Rezaeian & Der Kiureghian (2010)

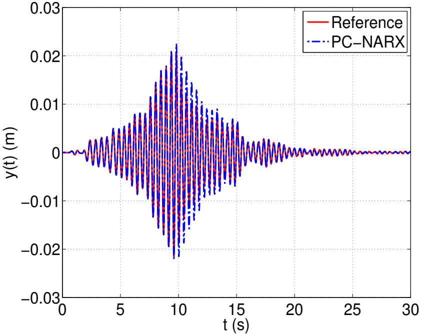
- 2D steel frame with uncertain properties submitted to synthetic ground motions
- Experimental design of size 300
- Ground motions obtained from modulated, filtered white noise

$$x(t) = q(t, \boldsymbol{\alpha}) \sum_{i=1}^{n} s_i \left(t, \boldsymbol{\lambda}(t_i)\right) \cdot \xi_i \quad \xi_i \sim \mathcal{N}(0, 1)$$

Application: earthquake engineering

Surrogate model of single trajectories

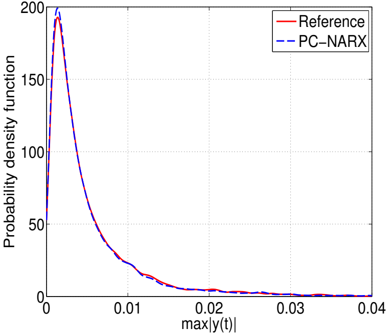




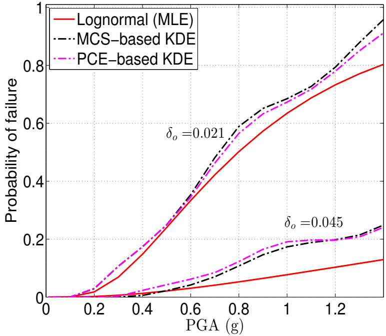
Application: earthquake engineering

First-storey drift

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



PDF of max. drift



Fragility curves for two drift thresholds

Outline

Introduction

- Polynomial chaos expansions (PCE) represent the model output on a fixed, predetermined basis:

$$Y = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} y_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \qquad \qquad \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) = \prod_{i=1}^{M} P^{(i)}_{\alpha_i}(X_i)$$

- Sparse PCEs are built from a pre-selected set of candidate basis functions $\mathcal{A}$
- High-dimensional problems (*e.g.* $M > 50$) may still be challenging for sparse PCE in case of small experimental designs ($n < 100$)

Low-rank tensor representations

Rank-1 function

A rank-1 function of $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}}$ is a product of univariate functions of each component:

$$w(\boldsymbol{x}) = \prod_{i=1}^{M} v^{(i)}(x_i)$$

Canonical low-rank approximation (LRA)

A canonical decomposition of $\mathcal{M}(\boldsymbol{x})$ is of the form

Nouy (2010)

$$\mathcal{M}^{\text{LRA}}(\boldsymbol{x}) = \sum_{l=1}^{R} b_l \left(\prod_{i=1}^{M} v_l^{(i)}(x_i) \right)$$

where:

- R is the rank (# terms in the sum)
- $v_l^{(i)}(x_i)$ are univariate function of x_i
- b_l are normalizing coefficients

Low-rank tensor representations

Polynomial expansions

Doostan et al., 2013

By expanding $v_l^{(i)}(X_i)$ onto polynomial basis orthonormal w.r.t. f_{X_i} one gets:

$$\widehat{Y} = \sum_{l=1}^{R} b_l \left(\prod_{i=1}^{M} \left(\sum_{k=0}^{p_i} z_{k,l}^{(i)} \, P_k^{(i)}(X_i) \right) \right)$$

where:

- $P_k^{(i)}(X_i)$ is k-th degree univariate polynomial of X_i
- p_i is the maximum degree of $P_k^{(i)}$
- $z_{k,l}^{(i)}$ are coefficients of $P_k^{(i)}$ in the l-th rank-1 term

Complexity

Assuming an isotropic representation $(p_i = p)$, the number of unknown coefficients is $R(p \cdot M + 1)$

Linear increase with dimensionality M

Low-rank tensor representations

Polynomial expansions

Doostan et al., 2013

By expanding $v_l^{(i)}(X_i)$ onto polynomial basis orthonormal w.r.t. f_{X_i} one gets:

$$\widehat{Y} = \sum_{l=1}^{R} b_l \left(\prod_{i=1}^{M} \left(\sum_{k=0}^{p_i} z_{k,l}^{(i)} \, P_k^{(i)}(X_i) \right) \right)$$

where:

- $P_k^{(i)}(X_i)$ is k-th degree univariate polynomial of X_i
- p_i is the maximum degree of $P_k^{(i)}$
- $z_{k,l}^{(i)}$ are coefficients of $P_k^{(i)}$ in the l-th rank-1 term

Complexity

Assuming an isotropic representation $(p_i = p)$, the number of unknown coefficients is $R(p \cdot M + 1)$

Linear increase with dimensionality M

Greedy construction of the LRA

Chevreuil et al. (2015); Konakli & Sudret (2016)

- An greedy construction is carried out by iteratively adding rank-1 terms. The r-th approximation reads $\widehat{Y}_r = \mathcal{M}_r(\boldsymbol{X}) = \sum_{l=1}^{r} b_l w_l(\boldsymbol{X})$
- In each iteration, alternate least-squares are used (correction and updating steps)

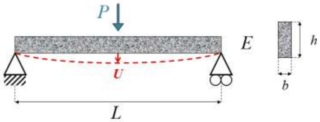
Correction step: sequential updating of $\boldsymbol{z}_r^{(j)}$, $j = 1, \dots, M$, to build w_r:

$$\boldsymbol{z}_r^{(j)} = \arg\min_{\boldsymbol{\zeta} \in \mathbb{R}^{p_j}} \left\| \mathcal{M} - \widehat{\mathcal{M}}_{r-1} - \left(\prod_{i \neq j} \sum_{k=0}^{p_i} z_{k,r}^{(i)} \; P_k^{(i)} \right) \left(\sum_{k=0}^{p_j} \zeta_k \; P_k^{(j)} \right) \right\|_{\mathcal{E}}^2$$

Updating step: evaluation of normalizing coefficients $\{b_1, \dots, b_r\}$:

$$\boldsymbol{b} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^r} \left\| \mathcal{M} - \sum_{l=1}^{r} \beta_l w_l \right\|_{\mathcal{E}}^2$$

Application: Simply supported beam



The maximum deflection U at midspan is of interest

Structural model

$$U = \frac{P\,L^3}{48E\,\frac{bh^3}{12}} = \frac{P\,L^3}{4\,Ebh^3}$$

(Rank-1 function!)

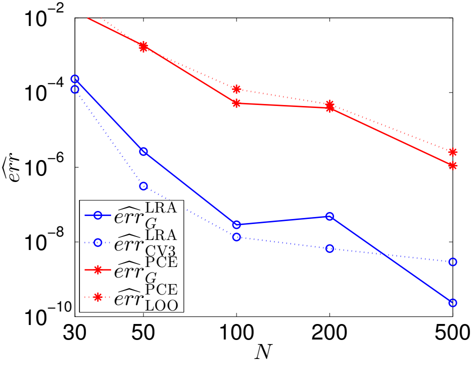
Probabilistic model:

independent lognormal random variables

Variable	Mean	Coef. of variation
Beam width b [m]	0.15	0.05
Beam height h [m]	0.3	0.05
Beam span L [m]	5	0.01
Young's modulus E [MPa]	30,000	0.15
Uniform load p [KN]	10	0.20

Comparison of surrogate modelling errors

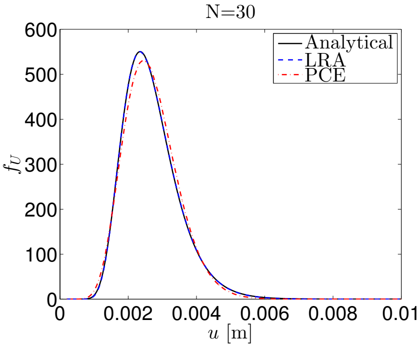
- Models built using experimental designs of increasing size
- Validation error computed from large Monte Carlo sampling
- Compared to leave-one-out cross-validation error (for PCE), resp. 3-fold cross validation (for LRA)



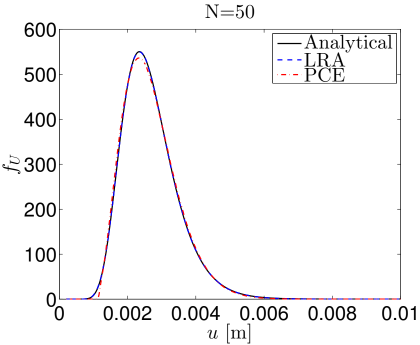
PDF of the beam deflection

Size of the experimental design: 30 (resp. 50) samples from Sobol' sequence

Kernel density estimates of the PDF **in the linear scale**



30 samples

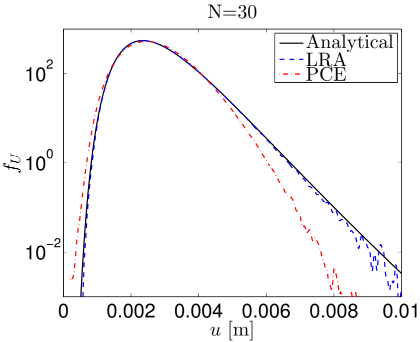


50 samples

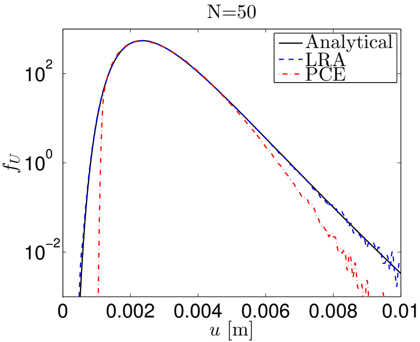
PDF of the beam deflection

Size of the experimental design: 30 (resp. 50) samples from Sobol' sequence

Kernel density estimates of the PDF **in the log scale**



30 samples

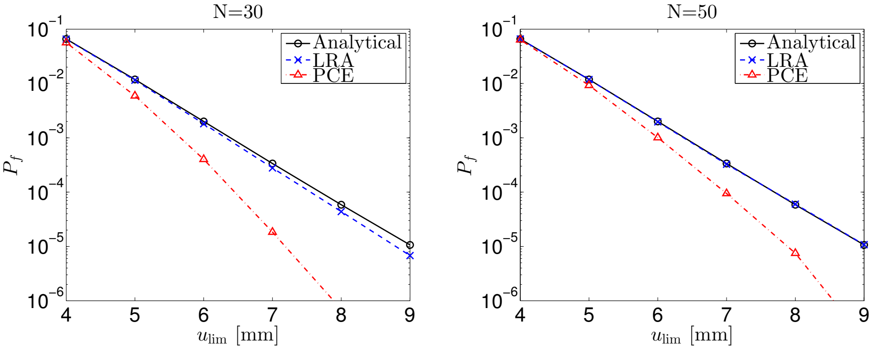


50 samples

Beam deflection - reliability analysis

Probability of failure

$$P_f = \mathbb{P}\left(U \geq \boldsymbol{u}_{\text{lim}}\right)$$

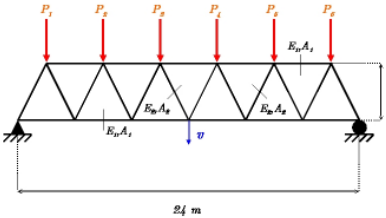


30 samples **50 samples**

Elastic truss

Structural model

Blatman & Sudret (2011)



- Response quantity: maximum deflection U
- Reliability analysis:

$$P_f = \mathbb{P}\left(U \geq u_{\text{lim}}\right)$$

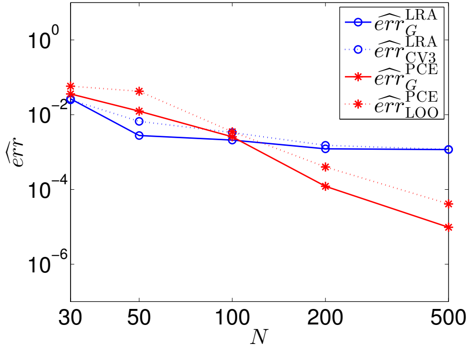
Probabilistic model

Variable	Distribution	mean	CoV
Hor. bars cross section A_1 [m]	Lognormal	0.002	0.10
Oblique bars cross section A_2 [m]	Lognormal	0.001	0.10
Young's moduli E_1, E_2 [MPa]	Lognormal	210,000	0.10
Loads $P_1, \ldots, P_6$ [KN]	Gumbel	50	0.15

Elastic truss

Konakli & Sudret, Prob. Eng. Mech (2016)

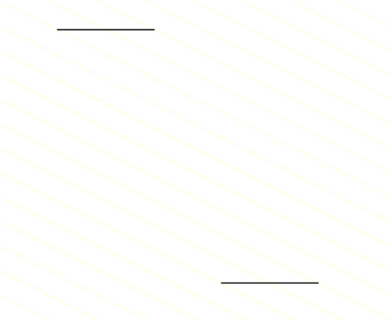
Surrogate modelling error



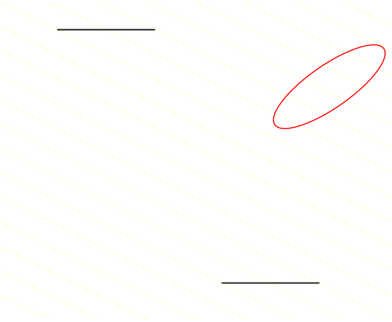
- Smaller validation error for LRA when ED is small ($N < 100$)
- Faster error decrease for PCE
- However ...

Elastic truss: validation plots

Konakli & Sudret, Prob. Eng. Mech (2016)



Low-rank approximation



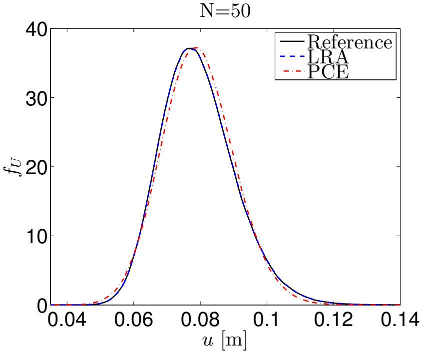
Polynomial chaos expansion

Polynomial chaos approximation is biased in the high values

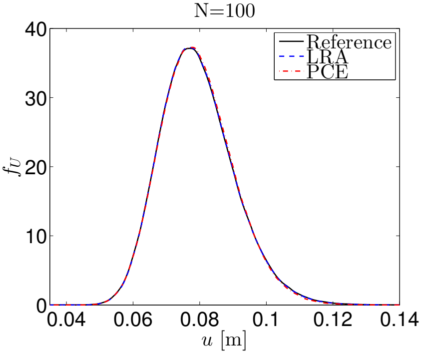
PDF of the truss deflection

Size of the experimental design: 50 (resp. 100) samples from Sobol' sequence

Kernel density estimates of the PDF **in the linear scale**



50 samples

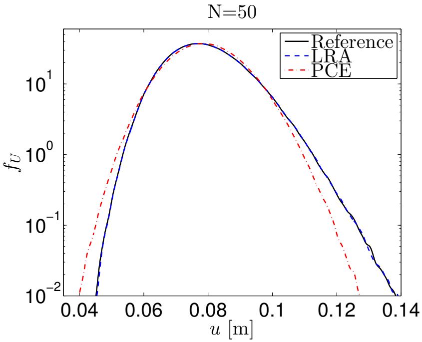


100 samples

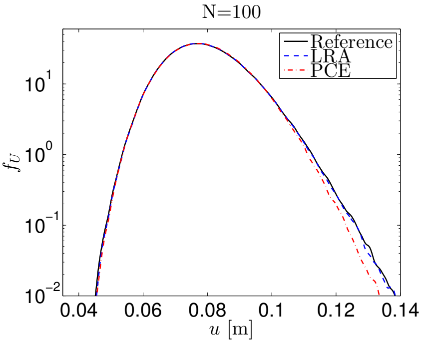
PDF of the truss deflection

Size of the experimental design: 50 (resp. 100) samples from Sobol' sequence

Kernel density estimates of the PDF **in the log scale**



50 samples

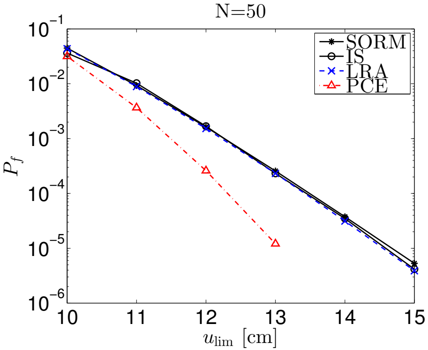


100 samples

Truss deflection - reliability analysis

Probability of failure

- LRA/PCE built from 50 samples
- Post-processing by crude Monte Carlo simulation: $P_f = \mathbb{P}\left(U \geq \boldsymbol{u}_{\text{lim}}\right)$



Number of model evaluations

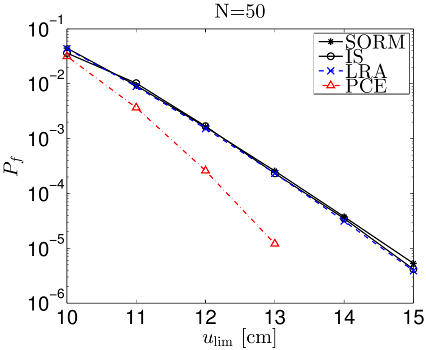
u_{lim}(m)	SORM	IS
0.10	387	375
0.11	365	553
0.12	372	660
0.13	367	755
0.14	379	1,067
0.15	391	1,179

Full curve at the cost of 50 finite element analyses

Truss deflection - reliability analysis

Probability of failure

- LRA/PCE built from 50 samples
- Post-processing by crude Monte Carlo simulation: $P_f = \mathbb{P}\left(U \geq \boldsymbol{u}_{\lim}\right)$



Number of model evaluations

$u_{\lim}$(m)	SORM	IS
0.10	387	375
0.11	365	553
0.12	372	660
0.13	367	755
0.14	379	1,067
0.15	391	1,179

Full curve at the cost of 50 finite element analyses

Conclusions

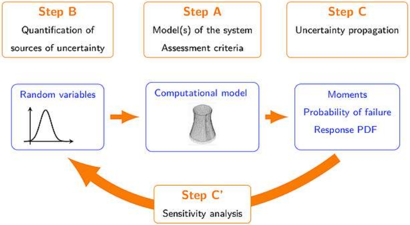
- Surrogate models are unavoidable for solving uncertainty quantification problems involving costly computational models (*e.g.* finite element models)
- Depending on the analysis, specific surrogates are most suitable: polynomial chaos expansions for distribution- and sensitivity analysis, low-rank tensor approximations for reliability analysis
- Kriging (a.k.a. Gaussian process modelling) and PC-Kriging are suitable for adaptive algorithms (enrichment of the experimental design)
- 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

UQLab

The Framework for Uncertainty Quantification

OVERVIEW | FEATURES | DOWNLOAD/INSTALL | DOCUMENTATION | EXAMPLES | ABOUT | CONTACT US

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



- MATLAB-based Uncertainty Quantification framework
- State-of-the art, highly optimized algorithms
- Easy to use and deploy
- Designed to be extended by users

http://www.uqlab.com

Probabilistic input description

Define, transform and sample random distributions

Key features:
- Usual and custom marginals
- Dependence modelling through copulas
- Monte Carlo & Latin Hypercube sampling, low-discrepancy sequences

Modelling

Connect your own simulation models to UQLab

Key features:
- Strings and inline function handles for analytical models
- MATLAB m-files
- Easy plugging of third party codes through wrappers

Polynomial chaos expansions

Compute fast surrogate models using polynomial chaos expansions

Key features:
- Full and sparse PC expansions
- Quadrature, sparse grids, least-squares and least-angle regression
- Advanced truncation schemes, custom basis specification

Kriging/Gaussian Processes

Compute robust surrogate models using Gaussian processes

Key features:
- Highly customizable trend and correlation functions
- Maximum likelihood and cross-validation for estimating hyperparameters
- Gradient-based and global optimizers

Sensitivity analysis

Identify the important input variables and their interactions

Key features:
- Screening (Morris method)
- Linear measures: Taylor series expansion (perturbation), standard regression coefficients
- ANOVA: Sobol' indices through Monte Carlo and polynomial chaos expansions

Reliability analysis/Rare events

NEW!

Estimate probabilities of failure and distribution tails

Key features:
- FORM/SORM approximation methods
- Sampling methods (Monte Carlo, importance sampling, subset simulation)
- Kriging-based adaptive methods (AK-MCS)

http://www.uqlab.com

Questions ?

Acknowledgements:
K. Konakli, C.V. Mai, S. Marelli, R. Schöbi

Chair of Risk, Safety & Uncertainty Quantification

www.rsuq.ethz.ch

The Uncertainty Quantification Laboratory

www.uqlab.com

Thank you very much for your attention !