ETH zürich

ETH Library

Surrogate modelling using sparse polynomial chaos expansions: a machine learning flavour

Other Conference Item

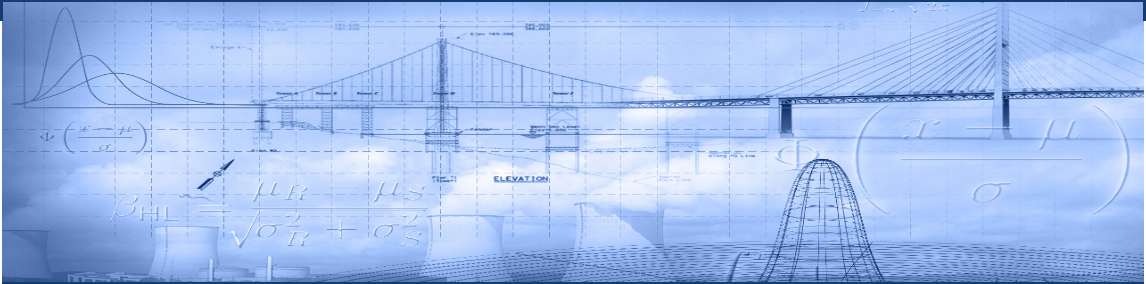
Author(s):
Sudret, Bruno

Publication date:
2024-06-19

Permanent link:
https://doi.org/10.3929/ethz-b-000680984

Rights / license:
In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.



Surrogate modelling using sparse polynomial chaos expansions: a machine learning flavour

1st CEACM Int. Conf. Synergy between Multiphysics/Multiscale Modelling and Machine Learning

Bruno Sudret

How to cite?

This presentation is a distinguished lecture given at the 1st CEACM International Conference on Synergy between Multiphysics/Multiscale Modelling and Machine Learning on June 19, 2024 in Prag (Czech Republic).

How to cite

Sudret, B. *Surrogate modelling using sparse polynomial chaos expansions: a machine learning flavour*, 1st CEACM International Conference on Synergy between Multiphysics/Multiscale Modelling and Machine Learning, *Distinguished Lecture*, June 19, 2024.

Computational models in engineering

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

- Governing equations
- Discretization techniques
- Solvers

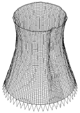
$$\mathrm{div}\ \boldsymbol{\sigma} + \boldsymbol{f} = \mathbf{0}$$

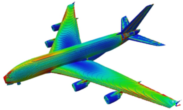
$$\boldsymbol{\sigma} = \mathbf{D} \cdot \boldsymbol{\varepsilon}$$

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

Computational models are used:

- To explore the design space ("virtual prototypes")
- To optimize the system (*e.g.* minimize the mass) under performance constraints
- To assess its robustness w.r.t uncertainty and its reliability
- Together with experimental data for calibration purposes



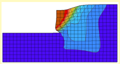


Computational models: the abstract viewpoint

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

Vector of input parameters $\boldsymbol{x} \in \mathbb{R}^M$ — Computational model $\mathcal{M}$ — Model response $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}) \in \mathbb{R}^Q$

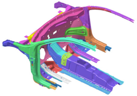
- Geometry
- Material properties
- Loading



- Analytical formula
- Finite element model
- Comput. workflow

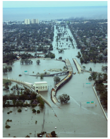
- Displacements
- Strains, stresses
- Temperature, etc.

Real world is uncertain



- Differences between the designed and the real system:
  - Dimensions (tolerances in manufacturing)
  - Material properties (*e.g.* variability of the stiffness or resistance)

- Unforecast exposures: exceptional service loads, natural hazards (earthquakes, floods, landslides), climate loads (hurricanes, snow storms, etc.), accidental human actions (explosions, fire, etc.)



Outline

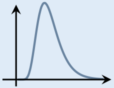
Global framework for uncertainty quantification

Step B
Quantification of sources of uncertainty

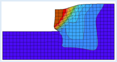
Step A
Model(s) of the system
Assessment criteria

Step C
Uncertainty propagation

Random variables



Computational model



Moments
Probability of failure
Response PDF

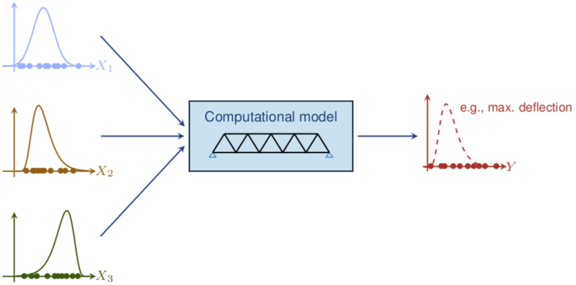
Step C'
Sensitivity analysis

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

Uncertainty propagation using Monte Carlo simulation

Principle: Generate virtual prototypes of the system using random numbers

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



Surrogate models for uncertainty quantification

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

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

Simulated data

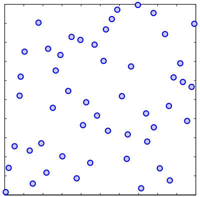
- It is fast to evaluate!

Surrogate models for uncertainty quantification

Name	Shape	Parameters
Polynomial chaos expansions	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$	$\boldsymbol{a}_{\boldsymbol{\alpha}}$
Low-rank tensor approximations	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{l=1}^{R} b_l \left(\prod_{i=1}^{M} v_l^{(i)}(x_i) \right)$	$b_l, \; z_{k,l}^{(i)}$
Kriging (a.k.a Gaussian processes)	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \boldsymbol{\beta}^{\mathsf{T}} \cdot \boldsymbol{f}(\boldsymbol{x}) + Z(\boldsymbol{x}, \omega)$	$\boldsymbol{\beta} \, , \; \sigma_Z^2 \, , \; \boldsymbol{\theta}$
Support vector machines	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{m} a_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\boldsymbol{a} \, , \; b$
(Deep) Neural networks	$\tilde{\mathcal{M}}(\boldsymbol{x}) = f_n \left(\cdots f_2 \left(b_2 + f_1 \left(b_1 + \boldsymbol{w}_1 \cdot \boldsymbol{x} \right) \cdot \boldsymbol{w}_2 \right) \right)$	$\boldsymbol{w}, \boldsymbol{b}$

Ingredients for building a surrogate model

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



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

Ingredients for building a surrogate model

- Smartly post-process the data $\{\mathcal{X}\,,\,\mathcal{M}(\mathcal{X})\}$ through a learning algorithm

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

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

Advantages of surrogate models

Usage

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

hours per run — seconds for 10^6 runs

Advantages

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

Challenges

- Need for rigorous validation
- Communication: advanced mathematical background

Efficiency

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

Surrogate modelling vs. machine learning

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

Outline

Polynomial chaos expansions in a nutshell

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

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

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

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

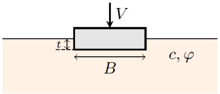
where:

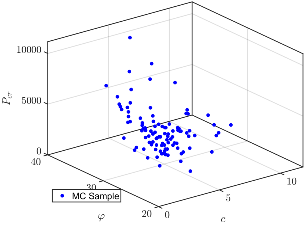
- $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$ are basis functions (multivariate orthonormal polynomials)
- $y_{\boldsymbol{\alpha}}$ are coefficients to be computed (coordinates)

Sampling (MCS) vs. spectral expansion (PCE)

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

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

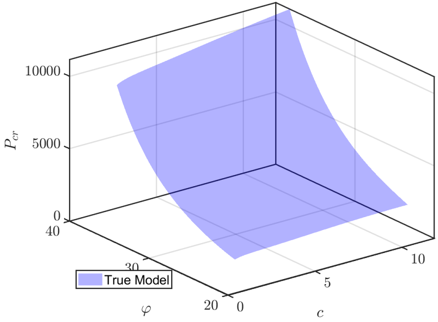




Thousands (resp. millions) of points are needed to grasp the structure of the response (resp. capture the rare events)

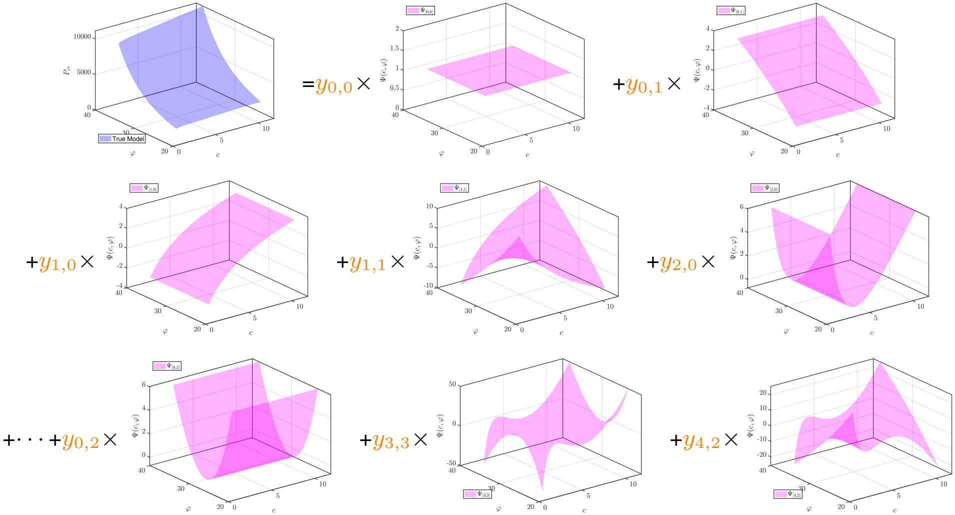
Defined as a function of the soil cohesion c and friction angle φ

Visualization of the PCE construction



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

Visualization of the PCE construction



Computing the coefficients by least-square minimization

Isukapalli (1999); Berveiller, Sudret & Lemaire (2006)

Principle

The exact (infinite) series expansion is considered as the sum of a truncated series and a residual:

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

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

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

Least-square minimization

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

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

Discrete (ordinary) least-square minimization

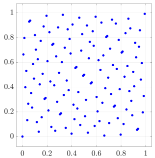
An estimate of the mean square error (sample average) is minimized:

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

Procedure

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

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



- Compute the experimental matrix

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

- Solve the resulting linear system

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

Simple is beautiful !

Error estimators

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

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

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

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

Leave-one-out cross validation

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

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

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

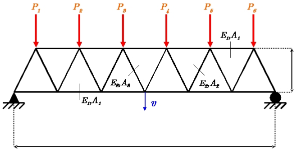
Outline

Curse of dimensionality

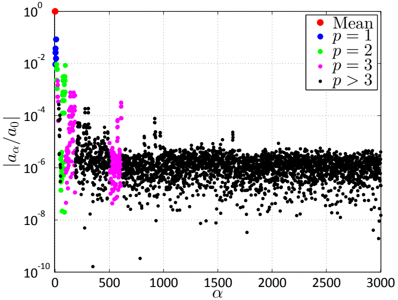
- The cardinality of the truncation scheme $\mathcal{A}^{M,p}$ is $P = \dfrac{(M+p)!}{M!\,p!}$
- Typical computational requirements: $n = OSR \cdot P$ where the oversampling rate is $OSR = 2 - 3$

However ... most coefficients are close to zero !

Example



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



Compressive sensing approaches

Blatman & Sudret (2011); Doostan & Owhadi (2011); Sargsyan *et al.* (2014); Jakeman *et al.* (2015)

- Sparsity in the solution can be induced by ℓ_1-regularization:

$$\boldsymbol{y}_{\boldsymbol{\alpha}} = \arg\min \frac{1}{n} \sum_{i=1}^{n} \left(\mathsf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{x}^{(i)}) - \mathcal{M}(\boldsymbol{x}^{(i)}) \right)^2 + \lambda \parallel \boldsymbol{y}_{\boldsymbol{\alpha}} \parallel_1$$

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

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

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

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

Outline

Post-processing sparse PC expansions

Statistical moments

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

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

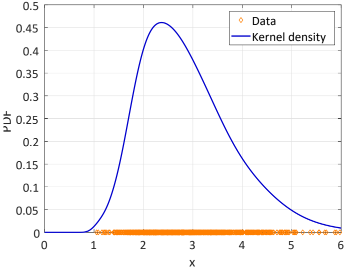
$$\text{Variance:} \quad \hat{\sigma}_Y^2 = \sum_{\boldsymbol{\alpha}\in\mathcal{A}\setminus\mathbf{0}} y_{\boldsymbol{\alpha}}^2$$

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



Sobol' indices

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

Variance decomposition

$$\mathrm{Var}\left[Y\right] = \sum_{i=1}^{M} D_i + \sum_{1 \leq i < j \leq M} D_{ij} + \cdots + D_{12\ldots M}$$

Sobol' indices

First order: $S_i = \dfrac{D_i}{\mathrm{Var}\left[Y\right]}$

Second order: $S_{ij} = \dfrac{D_{ij}}{\mathrm{Var}\left[Y\right]}$

Total: $S_i^T = \sum_{\mathbf{u} \supset i} S_{\mathbf{u}}$

Sobol decomposition of a PC expansion

Sudret, *Global sensitivity analysis using polynomial chaos expansion, 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

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

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

Example: strip foundation

Load bearing capacity

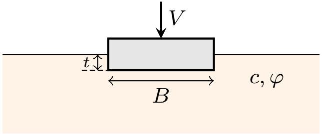
$$P_{cr} = B\,\sigma_{cr} = B\left[c\,N_c + \gamma t\,N_q + \frac{1}{2}\gamma\,B N_\gamma\right]$$

with the load bearing factors:

$$N_q = e^{\pi\tan\varphi}\,\frac{1+\sin\varphi}{1-\sin\varphi}$$

$$N_c = (N_q - 1)/\tan\varphi$$

$$N_\gamma = 2\,(N_q - 1)\tan\varphi$$



Variable	Description	Distribution	Moments
γ	Self-weight	Gaussian	$\mu_\gamma = 21\ kN/m^3,\ COV_\gamma = 5\%$
c	Cohesion	Lognormal	$\mu_c = 5\ kPa,\ COV_c = 30\%$
φ	Effective friction angle	Lognormal	$\mu_\varphi = 30^\circ,\ COV_\varphi = 8\%$
B	Width	Deterministic	$3\ m$
t	Depth	Gaussian	$\mu_t = 0.5\ m,\ COV_t = 20\%$

PCE vs. Monte Carlo simulation: moments

Monte-Carlo simulation

N_{MCS}	100	$1,000$	$10,000$	$100,000$	$1,000,000$
Mean	3216	3082	3121	3125	3124
95% CI	$[2942 - 3378]$	$[3057 - 3201]$	$[3105 - 3150]$	$[3115 - 3133]$	$[3122 - 3127]$
Std. dev	1109	1080	1188	1173	1174
95% CI	$[966 - 1565]$	$[1099 - 1313]$	$[1145 - 1207]$	$[1163 - 1185]$	$[1171 - 1178]$

Polynomial chaos expansion

	Experimental design of size $N_{\text{ED}} = 100$
Mean	3123
95% CI	$[3121 - 3125]$
Std. dev	1169
95% CI	$[1162 - 1171]$

PCE vs. Monte Carlo simulation: Sobol' indices

Monte-Carlo simulation

N_{MCS}	100	1,000	10,000	100,000	1,000,000
γ	$[0.007 - 0.020]$	$[0.013 - 0.017]$	$[0.014 - 0.015]$	$[0.015 - 0.015]$	$[0.015 - 0.015]$
c	$[0.006 - 0.018]$	$[0.013 - 0.019]$	$[0.013 - 0.015]$	$[0.014 - 0.015]$	$[0.015 - 0.015]$
φ	$[0.917 - 1.201]$	$[0.872 - 1.014]$	$[0.965 - 1.003]$	$[0.958 - 0.969]$	$[0.963 - 0.966]$
t	$[0.004 - 0.012]$	$[0.009 - 0.013]$	$[0.011 - 0.012]$	$[0.011 - 0.012]$	$[0.012 - 0.012]$
N_{TOT}	600	6,000	60,000	600,000	**6,000,000**

Polynomial chaos expansion

	Experimental design of size $N_{\text{ED}} = 100$
γ	$[0.015 - 0.016]$
c	$[0.014 - 0.014]$
φ	$[0.962 - 0.964]$
t	$[0.011 - 0.012]$
N_{TOT}	**100**

Outline

Models with time-dependent outputs

Problem statement

- Consider a computational model of a dynamical system:

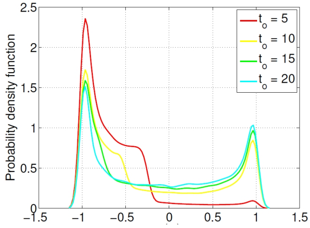
$$\mathcal{D}_{\boldsymbol{\Xi}} \times [0, T] : \; (\boldsymbol{\xi}, t) \mapsto \mathcal{M}(\boldsymbol{\xi}, t)$$

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

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

i.e.:

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



Time-dependent output PDF

Point-in-time PCE does not work!

Stochastic time warping

Problem

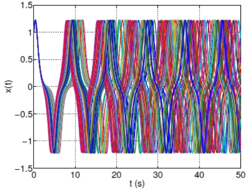
Mai & Sudret, SIAM J. Unc. Quant. (2017)

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

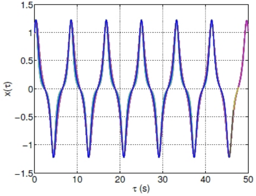
Principles of the method

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

Kraichnan Orszag model



Original trajectories



Trajectories after time warping

Example: Oregonator model

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

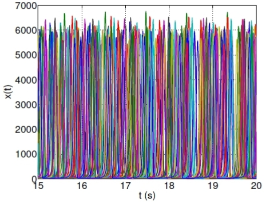
Governing equations

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

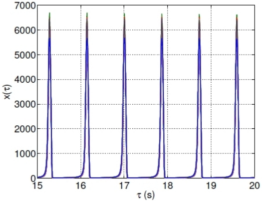
Input reaction parameters

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

Le Maître et al. (2010)

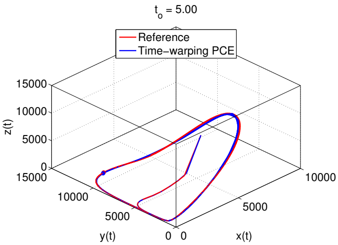


Original trajectories

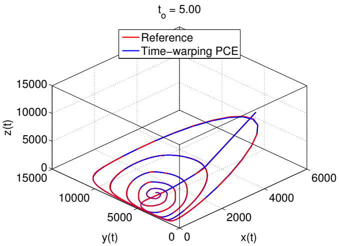


Trajectories after time warping

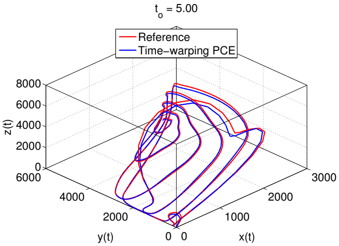
Oregonator model: trajectories



A trajectory in the state-space



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



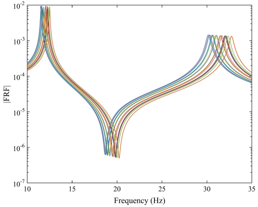
Standard deviation ($\varepsilon \approx 10^{-3}$)

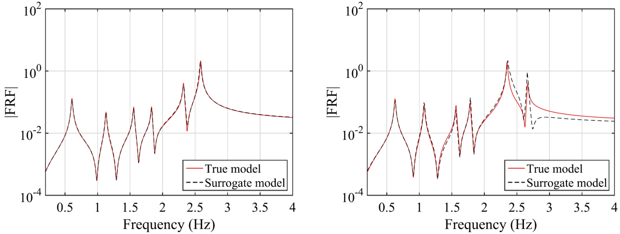
Dynamics in the frequency domain: frequency warping

Premise

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

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





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

Nonlinear transient models: PC-NARX

Goal

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

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

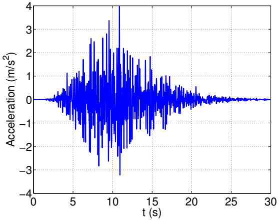
PC-NARX

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

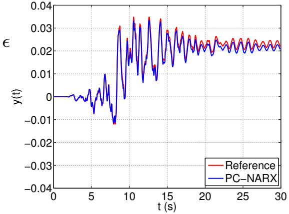
$$y(t) = \mathcal{F}\left(x(t),\ \dots,\ x(t-n_x), y(t-1),\ \dots,\ y(t-n_y)\right) + \epsilon_t \equiv \mathcal{F}\left(\boldsymbol{z}(t)\right) + \epsilon$$

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

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



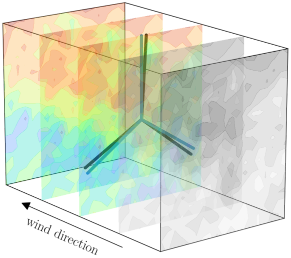
Earthquake ground motion



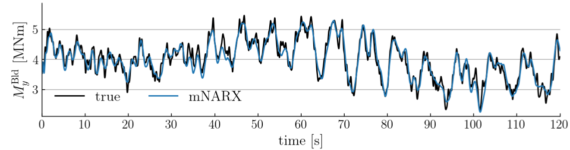
Structural response

Wind turbine simulations: mNARX surrogate

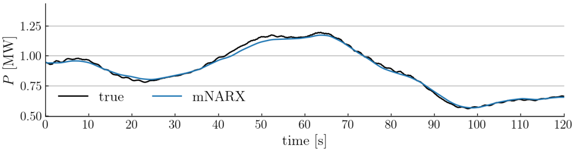
Movie-to-time series surrogate



Blade flapwise bending moment

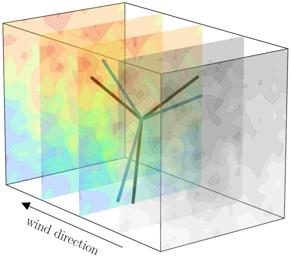


Generated power

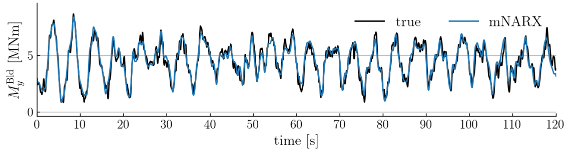


Wind turbine simulations: mNARX surrogate

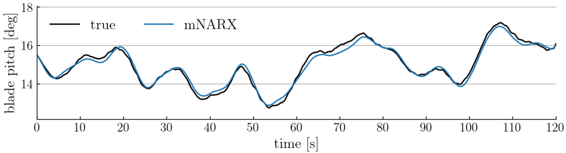
Movie-to-time series surrogate



Blade flapwise bending moment



Blade pitch



Conclusions

- Surrogate models are unavoidable for solving uncertainty quantification problems involving costly computational models (*e.g.* finite element models)
- Sparse polynomial chaos expansions are extremely efficient for distribution- and sensitivity analysis
- Extensions using time warping, PC-NARX, etc. allow us to address a wide range of engineering problems, including dynamics and Bayesian inverse problems
- Techniques for constructing surrogates are versatile, general-purpose and field-independent
- All the presented algorithms are available in the general-purpose uncertainty quantification software UQLab

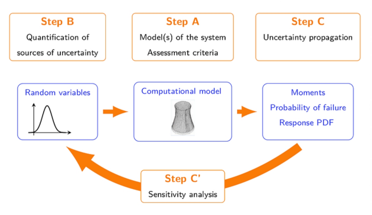
UQLab

The Framework for Uncertainty Quantification

OVERVIEW FEATURES DOCUMENTATION DOWNLOAD/INSTALL ABOUT COMMUNITY

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

www.uqlab.com



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

UQLab: The Uncertainty Quantification Software

- BSD 3-Clause license:

  Free access to academic, industrial, governmental and non-governmental users
- ∼7,200+ registered users from 94 countries since 2015 (450 in 2024)

http://www.uqlab.com

UQ[py]Lab

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

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

Country	# Users
China	1232
United States	983
France	534
Germany	417
Switzerland	453
United Kingdom	277
India	269
Brazil	247
Italy	248
Canada	133
Belgium	127
The Netherlands	119

As of May 21, 2024

Questions ?

Chair of Risk, Safety & Uncertainty Quantification

www.rsuq.ethz.ch

The Uncertainty Quantification Software

www.uqlab.com

UQLab

www.uqpylab.uq-cloud.io

UQ[py]Lab

The Uncertainty Quantification Community

www.uqworld.org

UQWorld

Thank you very much for your attention !