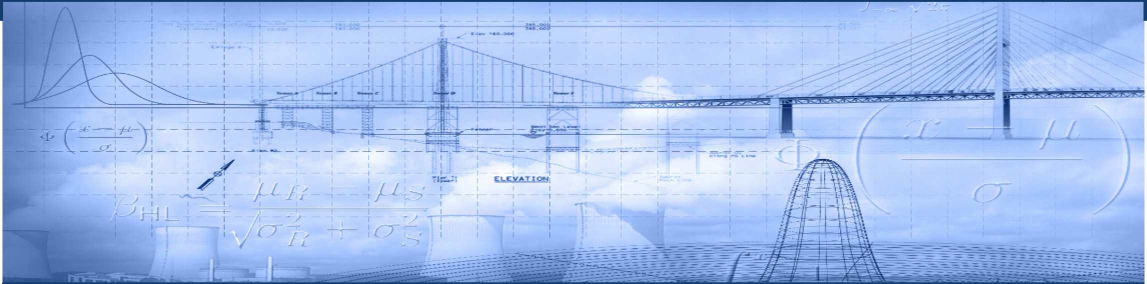
ETH zürich



Surrogate models for stochastic simulators: an overview with a focus on generalized lambda models

Bruno Sudret, Xujia Zhu

How to cite?

This presentation is an invited lecture given at the MascotNum Workshop on “Stochastic simulators” on March 11, 2021.

How to cite

Sudret, B. and Zhu, X., *Surrogate models for stochastic simulators: an overview with a focus on generalized lambda models*, MascotNum Workshop on “Stochastic simulators” (online), March 11th, 2021.

Chair of Risk, Safety and Uncertainty quantification

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

Research topics

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

`http://www.rsuq.ethz.ch`

The SAMOS project

- The SAMOS project (“SurrogAte Modelling for stOchastic Simulators”) is funded by the Swiss National Science Foundation under Grant # 175524 (May 2018 - April 2022).
- It is devoted to the development of innovative methods to ... build surrogate models for stochastic simulators!

Xujia Zhu

Today’s talk on “Stochastic polynomial chaos expansions for emulating stochastic simulators”

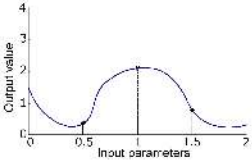
Nora Lüthen

Today’s talk on “Surrogating stochastic simulators using spectral methods and advanced statistical modeling”

Deterministic vs. stochastic simulators

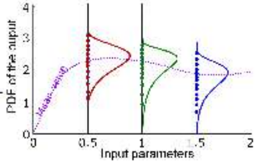
Simulators

Deterministic simulators



Output $\mathcal{M}_d(x)$ is a real number

Stochastic simulators



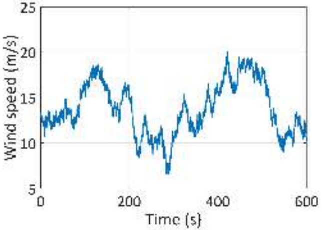
Output $\mathcal{M}_s(x)$ is a random variable

Example: wind turbine simulation

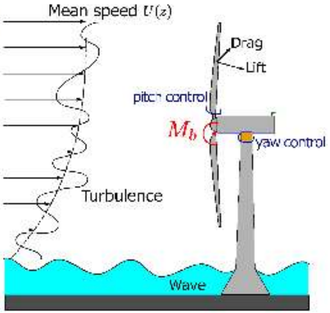
Input parameters

- Mean speed
- Turbulence intensity
- Shear exponent
- Air density
- Inclination angle

Wind generation →



Aero-servo-elastic simulator →



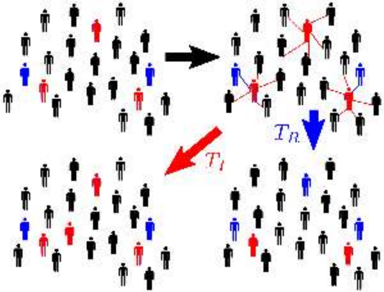
Example: epidemiology

Terminology

- S_t: number of susceptible individuals at time t
- I_t: number of infected individuals at time t
- R_t: number of recovered individuals at time t

System dynamics

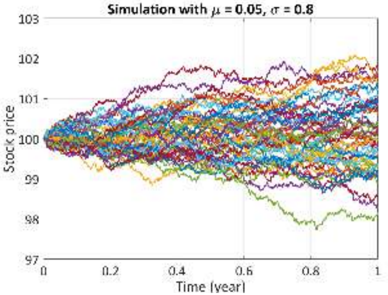
- Susceptible individuals can get infected due to close contact with infected individuals ($S \to I$)
- Infected individuals can recover and becomes immune to future infections ($I \to R$)
- Random contact and recovery modelled with Poisson processes



Example: mathematical finance

Geometrical Brownian motion

$$\mathrm{d}S_t = \mu\, S_t\, \mathrm{d}t + \sigma\, S_t\, \mathrm{d}W_t$$



- S_t: stock price, W_t: Wiener process
- μ: drift, σ: volatility

Asian option

- The payoff (of a call option) is contingent on the average price of the underlying asset

$$C = \max\left\{A_T - K, 0\right\}, \text{ with } A_t = \frac{1}{t}\int_0^t S_u \mathrm{d}u.$$

Outline

Formal definition

- A (scalar) stochastic simulator $\mathcal{M}_s$ is a mapping:

$$\mathcal{M}_s : \mathcal{D}_{\boldsymbol{X}} \times \Omega \to \mathbb{R}$$
$$(\boldsymbol{x}, \omega) \mapsto \mathcal{M}_s(\boldsymbol{x}, \omega)$$

  where $\mathcal{D}_{\boldsymbol{X}}$ is the input parameters space and $\{\Omega, \mathcal{F}, \mathbb{P}\}$ is a probability space

- When fixing $\boldsymbol{x} = \boldsymbol{x}_0$, the output is a random variable $Y|\boldsymbol{X} = \boldsymbol{x}_0 \equiv \mathcal{M}_s(\boldsymbol{x}_0, \omega)$
- When fixing the seed $\omega = \omega_0$ we get a deterministic simulator $\boldsymbol{x} \mapsto \mathcal{M}_s(\boldsymbol{x}, \omega_0)$ (a.k.a. trajectory)

Latent variables

$\mathcal{M}_s$ can be seen as a deterministic function $\mathcal{M}$ of input parameters $\boldsymbol{x}$ and latent variables $\boldsymbol{Z}$:

$$\mathcal{M}_s(\boldsymbol{x}, \omega) = \mathcal{M}\left(\boldsymbol{x},\ \boldsymbol{Z}(\omega)\right)$$

Computational costs induced by stochastic simulators

- Replications are needed to estimate the PDF of $Y|\boldsymbol{X} = \boldsymbol{x}$
- Many runs must be carried out by varying $\boldsymbol{X}$ for uncertainty propagation, sensitivity analysis, optimization, etc.
- Realistic simulators (e.g. for wind turbine design) are costly

Need for surrogate models

Requirements

Our goal is to develop a methodology that is:

- Non-intrusive (*i.e.* that considers the stochastic simulator as a black box)
- General-purpose: no restrictive assumption (*e.g.* Gaussian) on the family of the output distribution is made
- Able to tackle the full distribution of $Y|\boldsymbol{X} = \boldsymbol{x}$, but also quantities of interest (*e.g.* mean, variance, quantiles)
- Providing a representation of $Y|\boldsymbol{X} = \boldsymbol{x}$ easy to sample from

Existing approaches

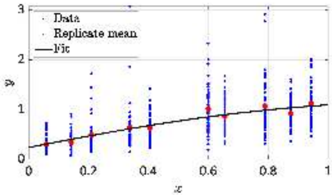
The literature on stochastic simulators is both old and new:

- Replication-based approaches
- Gaussian models
- Estimation of the conditional distribution
- Latent variable models
- Random field representations
- Quantile regression

Replication-based approaches

Main idea

- Estimate distributions/QoIs based on replications
- Treat the estimated parameters as outputs from a deterministic simulator and apply standard surrogate models



Literature

- Stochastic Kriging: Ankenman *et al.* (2006) *Stochastic Kriging for simulation metamodeling*, Oper. Res.
- Quantile Kriging: Plumlee & Tuo (2014) *Building accurate emulators for stochastic simulations via quantile Kriging*, Technometrics
- Kernel density estimation: Moutoussamy *et al.* (2015) *Emulators for stochastic simulation codes*, ESAIM: Math. Model. Num. Anal.
- Generalized lambda model: Zhu & Sudret (2020) *Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions*, Int. J. Uncertainty Quantification

Assuming normality: Kriging models

Main idea

- Response distributions are normal
- Mean function $\mu(\boldsymbol{x})$ and log-variance function $\log(V(\boldsymbol{x}))$ are modeled by Gaussian processes

Literature

- Full Bayesian setup: Goldberg *et al.* (1997) *Regression with input-dependent noise: a Gaussian process treatment*, NIPS10
- Iterative fitting: Marrel *et al.* (2012) *Global sensitivity analysis of stochastic computer models with joint metamodels*, Stat. Comput.
- Maximum likelihood: Binois *et al.* (2018) *Practical heteroscedastic Gaussian process modeling for large simulation experiments*, J. Comput. Graph. Stat.

Conditional distribution estimation

Main approaches

- Estimate the joint distribution of $(\boldsymbol{X}, Y)$ by kernel smoothing, then compute the conditional PDF by:

$$f(y \mid \boldsymbol{x}) = \frac{f(\boldsymbol{x}, y)}{f(\boldsymbol{x})}$$

- Use parametric models to represent the conditional distribution directly

Literature

- Kernel smoothing: Hall *et al.* (2004) *Cross-validation and the estimation of conditional probability densities*, J. Amer. Stat. Assoc.
- Vine copula: Kraus & Czado (2017) *D-vine copula based quantile regression*, Comput. Stat. Data Anal.
- Generalized lambda model: Zhu & Sudret (2021) *Emulation of stochastic simulators using generalized lambda models*, Submitted to SIAM/ASA J. Unc. Quant.

Latent variable models

Main idea

- Introduce explicitly latent variables $\tilde{Z}$ into a deterministic model to emulate the random nature of stochastic simulators

$$Y(\boldsymbol{x}) \overset{\text{d}}{=} \tilde{\mathcal{M}}(\boldsymbol{x}, \tilde{Z})$$

Literature

- Yan & Perdikaris (2019) *Conditional deep surrogate models for stochastic, high-dimensional, and multi-fidelity systems*, Comput. Mech.
- Stochastic polynomial chaos expansions for emulating stochastic simulators (Talk X. Zhu)

Random field approaches

Main idea

- Consider the stochastic simulator as a random field indexed by the input variables:

$$Y_{\boldsymbol{x}}(\omega) = \mathcal{M}\left(\boldsymbol{x}, \boldsymbol{Z}(\omega)\right)$$

- Fixing the internal stochasticity ($\omega = \omega_0$) gives access to trajectories $\boldsymbol{x} \mapsto \mathcal{M}(\boldsymbol{x}, \boldsymbol{Z}(\omega_0))$

Literature

- Azzi *et al.* (2019) *Surrogate modeling of stochastic functions-application to computational electromagnetic dosimetry*, Int. J. Uncertainty Quantification
- Azzi *et al.* (2020) *Sensitivity analysis for stochastic simulators using differential entropy*, Int. J. Uncertainty Quantification
- Surrogating stochastic simulators using spectral methods and advanced statistical modeling (Talk N. Lüthen)

Outline

Definition

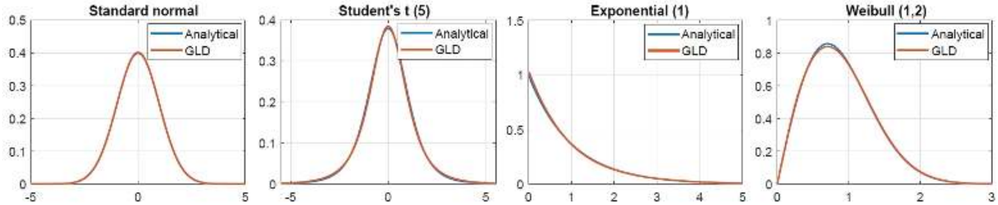
- The Freimer-Mudholkar-Kollia-Lin (FMKL) lambda distribution is defined through its quantile function $Q(u; \boldsymbol{\lambda})$ by 4 parameters

$$Q(u; \boldsymbol{\lambda}) = \lambda_1 + \frac{1}{\lambda_2} \left(\frac{u^{\lambda_3} - 1}{\lambda_3} - \frac{(1-u)^{\lambda_4} - 1}{\lambda_4} \right)$$

  where:
  - λ_1 is the location parameter
  - $\lambda_2 > 0$ is the scale parameter
  - λ_3, λ_4 are shape parameters
- The PDF is obtained by:

$$f_Y(y; \boldsymbol{\lambda}) = \frac{1}{Q'(u; \boldsymbol{\lambda})} = \frac{\lambda_2}{u^{\lambda_3 - 1} + (1-u)^{\lambda_4 - 1}} \qquad \text{with } u = Q^{-1}(y; \boldsymbol{\lambda})$$

Properties



- GLDs approximate well unimodal PDFs (bell-, U-shaped, bounded and unbounded)
- λ_3 and λ_4 control the shape and boundedness

$$B_l(\boldsymbol{\lambda}) = \begin{cases} -\infty, & \lambda_3 \leq 0 \\ \lambda_1 - \frac{1}{\lambda_2 \lambda_3}, & \lambda_3 > 0 \end{cases}, \quad B_u(\boldsymbol{\lambda}) = \begin{cases} +\infty, & \lambda_4 \leq 0 \\ \lambda_1 + \frac{1}{\lambda_2 \lambda_4}, & \lambda_4 > 0 \end{cases}$$

Moments

- The mean value and variance read:

$$\mu = \lambda_1 - \frac{1}{\lambda_2}\left(\frac{1}{\lambda_3+1} - \frac{1}{\lambda_4+1}\right)$$

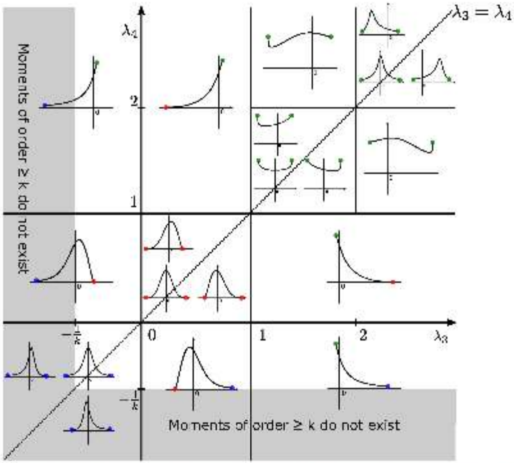
$$V \overset{\text{def}}{=} \sigma^2 = \frac{(d_2 - d_1^2)}{\lambda_2^2}$$

where (B is the Beta function):

$$d_1 = \frac{1}{\lambda_3}\,\mathrm{B}(\lambda_3+1,1) - \frac{1}{\lambda_4}\,\mathrm{B}(1,\lambda_4+1)$$

$$d_2 = \frac{1}{\lambda_3^2}\,\mathrm{B}(2\lambda_3+1,1) - \frac{2}{\lambda_3\lambda_4}\,\mathrm{B}(\lambda_3+1,\lambda_4+1) + \frac{1}{\lambda_4^2}\,\mathrm{B}(1,2\lambda_4+1)$$

Summary chart



- Blue points: infinite support
- Red points: finite support, with PDF = 0 at the bound
- Green points: finite support, with PDF $\neq 0$ at the bound

Zhu & Sudret (2020). *Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions*, Int. J. Uncertainty Quantification

Outline

Introduction and literature

Generalized lambda distributions

Generalized lambda/PCE models

- With replications
- Without replications

Application examples

Conclusions & outlook

Problem statement

Let us consider a stochastic simulator $\mathcal{M}_S$ and the following experimental design with replications

- Experimental design of size N in the $\boldsymbol{X}$-space: $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(N)}\right\}$
- R replications for each $\boldsymbol{x}^{(i)} \in \mathcal{X}$: $\mathcal{Y}^{(i)} = \left\{y^{(i,1)}, y^{(i,2)}, \dots, y^{(i,R)}\right\}$

Two approaches

Zhu & Sudret (2020) *Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions*, Int. J. Uncertainty Quantification

- Infer-and-fit: infer a lambda distribution for each point $\boldsymbol{x}^{(i)}$ of the experimental design, then fit a sparse polynomial chaos expansion to the parameters $\boldsymbol{\lambda}$
- Joint inference: improve the previous results by maximum likelihood optimization

Local inference of lambda distributions ("Infer")

Estimate $\lambda^{(i)}$ based on $\mathcal{Y}^{(i)} = \left\{ y^{(i,1)}, \ldots, y^{(i,R)} \right\}$

- Maximum likelihood estimation

$$\hat{\boldsymbol{\lambda}}^{(i)} = \arg\max_{\boldsymbol{\lambda}} \sum_{r=1}^{R} \log \left(\frac{\lambda_2}{u_{i,r}^{\lambda_3 - 1} + (1 - u_{i,r})^{\lambda_4 - 1}} \right)$$

where

$$u_{i,r} = Q^{-1}\left(y^{(i,r)}; \boldsymbol{\lambda} \right), \quad y^{(i,r)} = Q(u_{i,r}; \boldsymbol{\lambda}) = \lambda_1 + \frac{1}{\lambda_2} \left(\frac{u_{i,r}^{\lambda_3} - 1}{\lambda_3} - \frac{(1 - u_{i,r})^{\lambda_4} - 1}{\lambda_4} \right)$$

- Nonlinear equation that can be solved numerically (Q is a monotonically increasing function)

Polynomial chaos expansions

- Fit 4 PCEs from the lambda data points $\Lambda = \left\{ \hat{\boldsymbol{\lambda}}^{(1)}, \ldots, \hat{\boldsymbol{\lambda}}^{(N)} \right\}$

Polynomial chaos expansions in a nutshell

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

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

The polynomial chaos expansion for a deterministic simulator $\boldsymbol{X} \mapsto \mathcal{M}_d(\boldsymbol{X})$ reads:

$$\mathcal{M}_d(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d} a_{\boldsymbol{\alpha}}\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$$

where:

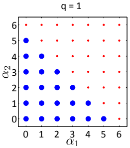
- $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$ are basis functions (multivariate orthonormal polynomials)
- $a_{\boldsymbol{\alpha}}$ are coefficients to be computed (coordinates)

Polynomial chaos expansions

Truncation schemes

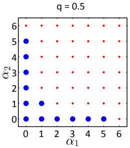
- "Full basis" of degree p:

$$\boldsymbol{\alpha} \in \mathcal{A}^{p,M} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}^M ; \sum_{i=1}^{M} \alpha_i \leq p \right\}$$



- q-norm truncation:

$$\boldsymbol{\alpha} \in \mathcal{A}^{p,q,M} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}^M ; \; \|\boldsymbol{\alpha}\|_q \stackrel{\text{def}}{=} \left(\sum_{i=1}^{M} \alpha_i^q \right)^{\frac{1}{q}} \leq p \right\}, \qquad 0 < q < 1$$



- Sparse expansion: $\mathcal{A}$ only contains relevant basis functions taken out of a candidate basis

Fitting polynomial chaos expansions ("Fit")

- Gather the data from the "Fit"-step: $\left\{\left(\boldsymbol{x}^{(1)}, \boldsymbol{\lambda}^{(1)}\right), \left(\boldsymbol{x}^{(2)}, \boldsymbol{\lambda}^{(2)}\right), \ldots, \left(\boldsymbol{x}^{(N)}, \boldsymbol{\lambda}^{(N)}\right)\right\}$
- Represent the distribution parameters $\boldsymbol{\lambda}$ by polynomial chaos expansions

$$\lambda_k(\boldsymbol{x}) \approx \lambda_k^{\mathsf{PC}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_k} a_{k,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \qquad \text{for } k = 1, 3, 4$$

$$\lambda_2(\boldsymbol{x}) \approx \lambda_2^{\mathsf{PC}}(\boldsymbol{x}; \boldsymbol{a}) = \exp\left(\sum_{\boldsymbol{\alpha} \in \mathcal{A}_2} a_{2,\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right)$$

- PCE coefficients are calibrated by the sparse regression method Hybrid-LARS

Blatman & Sudret (2011) *Adaptive sparse polynomial chaos expansion based on Least Angle Regression*, J. Comput. Phys.

Improvement: joint fitting

Rationale

- The results of the Infer-and-fit algorithm depends on the accuracy of the local inference
- **Idea:** build a global model for the joint distribution of inputs and outputs:

$$f_{\boldsymbol{X},Y}(\boldsymbol{x}, y) = f_{Y|\boldsymbol{X}}(y \mid \boldsymbol{x}) \cdot f_{\boldsymbol{X}}(\boldsymbol{x})$$

where the conditional PDF is represented by a lambda model:

$$f^{\text{GLD}}_{\boldsymbol{X},Y}(\boldsymbol{x}, y; \boldsymbol{a}) = f^{\text{GLD}}_{Y|\boldsymbol{X}}\left(y; \boldsymbol{\lambda}^{\text{PC}}(\boldsymbol{x}; \boldsymbol{a})\right) \cdot f_{\boldsymbol{X}}(\boldsymbol{x})$$

Procedure

Find the optimal PCE coefficients $\boldsymbol{a}$ that minimize the Kullback-Leibler divergence between $f_{\boldsymbol{X},Y}$ and $f^{\text{GLD}}_{\boldsymbol{X},Y}$:

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} D_{KL}\left(f_{\boldsymbol{X},Y} \,||\, f^{\text{GLD}}_{\boldsymbol{X},Y}(\,.\,; \boldsymbol{a})\right)$$

Corresponding loss function

Solution

After some basic algebra, the minimization problem reduces to:

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \mathbb{E}_{\boldsymbol{X},Y}\left[\log \hat{f}_{Y|\boldsymbol{X}}\left(Y;\, \boldsymbol{\lambda}^{\mathsf{PC}}(\boldsymbol{X};\, \boldsymbol{a})\right)\right]$$

Estimator

- The PCE bases for $\boldsymbol{\lambda}^{\mathsf{PC}}(\boldsymbol{x})$ are taken from the "Infer-and-Fit" approach
- The expectation w.r.t. $\boldsymbol{X}$ and Y is computed from the experimental design $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$, with R replications in each point (outputs $y^{(i,r)}$):

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \frac{1}{N\,R} \sum_{i=1}^{N} \sum_{r=1}^{R} \log \hat{f}_{Y|\boldsymbol{X}}\left(y^{(i,r)};\, \boldsymbol{\lambda}^{\mathsf{PC}}(\boldsymbol{x}^{(i)};\, \boldsymbol{a})\right)$$

Illustration of the algorithm

Toy example

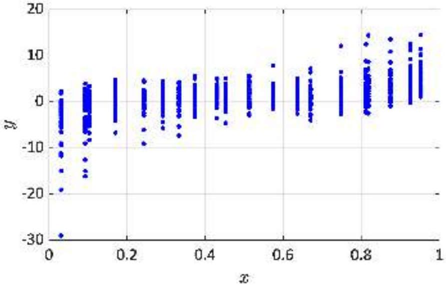
- Analytical lambda model

$$\lambda_1(x) = 50x^3 - 75x^2 + 35x - 4$$

$$\lambda_2(x) = \exp(-3x^2 + 3x - 1)$$

$$\lambda_3(x) = -0.2 + 0.7x$$

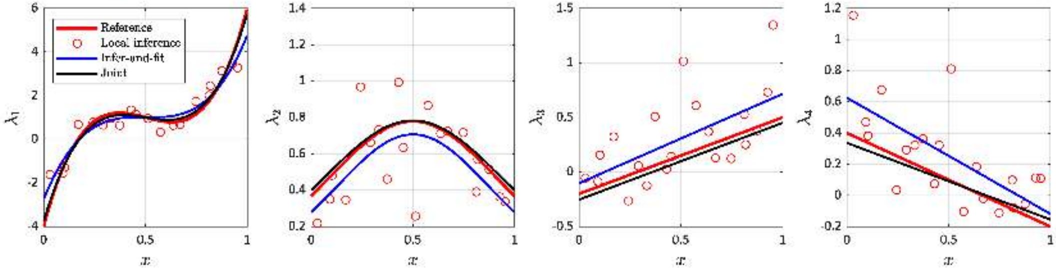
$$\lambda_4(x) = 0.4 - 0.6x$$



- Experimental design of $N = 20$, replications $R = 40$

Illustration of the algorithm

Joint fitting



Outline

Introduction

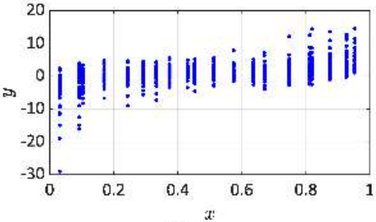
Goal

Propose a framework to build a stochastic emulator without replications (... that can also be applied if there is replicated data)

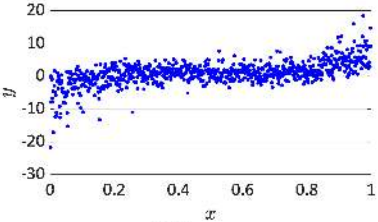
Rationale

The joint fitting estimator does not require replications!

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \mathbb{E}_{\boldsymbol{X},Y}\left[\log \hat{f}_{Y|\boldsymbol{X}}\left(Y; \boldsymbol{\lambda}^{\text{PC}}(\boldsymbol{X}; \boldsymbol{a})\right)\right]$$



With



Without

Procedure

Ingredients

- An experimental design $\mathcal{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$ and model evaluations $y^{(i)} \stackrel{\text{def}}{=} \mathcal{M}(\boldsymbol{x}(i), \omega_i)$
- Pre-selected PC bases for the four polynomial chaos expansions of $\lambda_i(\boldsymbol{x}),\ i = 1, \ldots, 4$

Selection of PCE bases

- PCE models for the mean and variance of the model output built using the feasible generalized least-square method
- Use the PCE basis of $\mu(\boldsymbol{x})$ (resp. $\log \sigma^2(\boldsymbol{x})$) for λ_1 (resp. λ_2)
- PCE of degree 1 for λ_3 and λ_4 (it is assumed that the shape of the response distribution does not vary nonlinearly with $\boldsymbol{x}$)

Algorithm 1: Feasible generalized least-squares (FGLS)

Input: Computational budget N

Initialization

  Experimental design $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$

  Model evaluations $\mathcal{Y} = \{y^{(i)} \stackrel{\text{def}}{=} \mathcal{M}(\boldsymbol{x}(i), \omega_i), i = 1, \ldots, N\}$

  Set $k = 0$. Estimate the mean function $\hat{\mu}_0(\boldsymbol{x})$ by ordinary least-squares % Fix the basis after (p, q) search

FGLS

  while *NotConverged* **do**

    Subtract the estimated mean from the data, *i.e.* $r^{(i)} = y^{(i)} - \hat{\mu}_k\left(\boldsymbol{x}^{(i)}\right)$

    Estimate the variance function $\hat{\sigma}_k^2(\boldsymbol{x})$ based on $\left\{\left(\boldsymbol{x}^{(i)}, 2\log\left|r^{(i)}\right|\right), i = 1, \ldots, N\right\}$

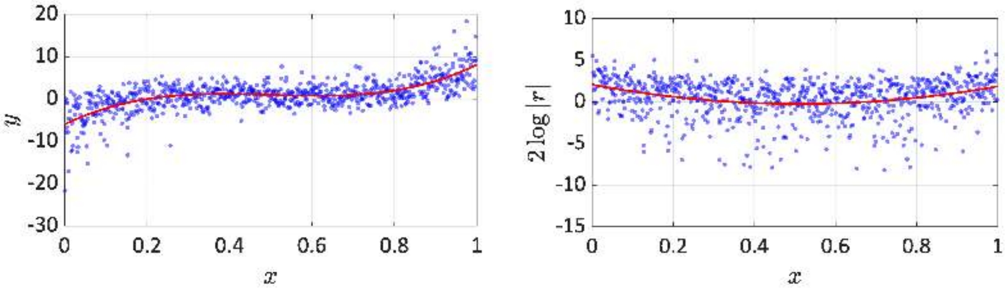
    $k \leftarrow k + 1$

    Use $\hat{\sigma}_k^2(\boldsymbol{x})$ as weight to estimate $\hat{\mu}_k(\boldsymbol{x})$ by weighted least-squares

  end

Return PCE expansions of $\mu(\boldsymbol{x})$ and $\log\sigma^2(\boldsymbol{x})$

Feasible generalized least-squares (FGLS): illustration



Iteration 10

Outline

Error metrics for stochastic emulators

Wasserstein distance

- It is the L^2 distance between the quantile functions:

$$d^2_{\mathrm{WS}}(Y, \hat{Y}) \stackrel{\text{def}}{=} \|Q_Y - \hat{Q}_Y\|^2_{L^2} = \int_0^1 \left(Q_Y(u) - \hat{Q}_Y(u)\right)^2 du$$

Normalized Wasserstein distance

$$\varepsilon = \frac{\mathbb{E}_{\boldsymbol{X}}\left[d^2_{\mathrm{WS}}\left(Y(\boldsymbol{X}), Y^{\text{GLaM}}(\boldsymbol{X})\right)\right]}{\mathrm{Var}\left[Y\right]}$$

Geometric Brownian motion

$$\mathrm{d}S_t = x_1\, S_t\, \mathrm{d}t + x_2\, S_t\, \mathrm{d}W_t$$

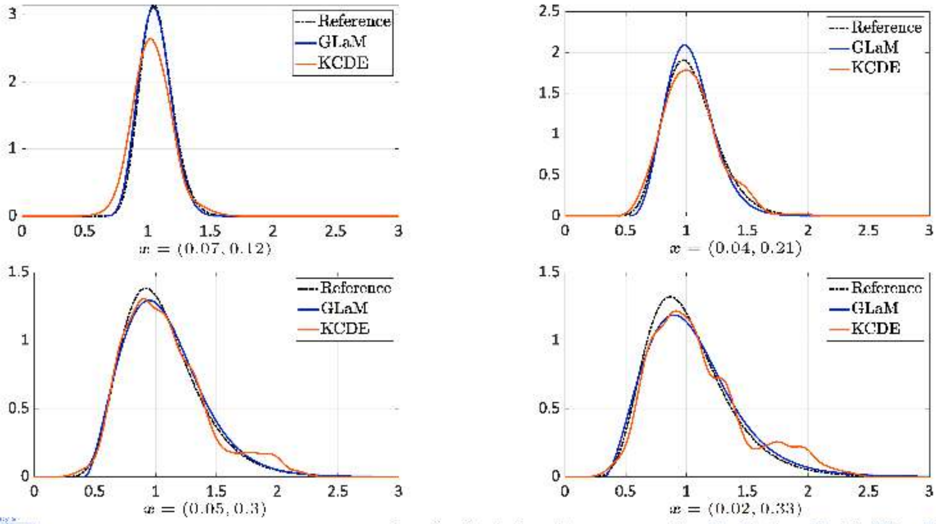
- S_t: stock price, W_t: Wiener process
- x_1: drift, x_2: volatility
- The analytical distribution of S_t exists (Itô's calculus)

$$S_t(\boldsymbol{x})/S_0 \sim \mathcal{LN}\left(\left(x_1 - x_2^2/2\right) t, x_2\, t\right)$$

Setup

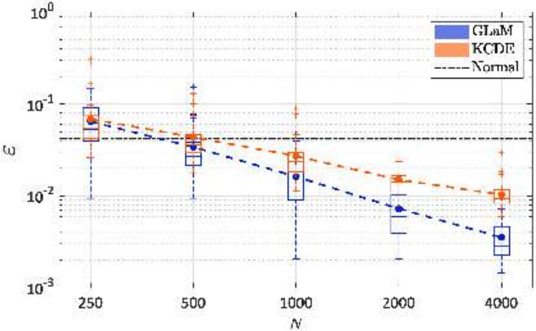
- $X_1 \sim \mathcal{U}(0, 0.1)$, $X_2 \sim \mathcal{U}(0.1, 0.4)$
- $Y = S_1$ for $t = 1$ is of interest, *i.e.* $Y(\boldsymbol{x}) \sim \mathcal{LN}\left(x_1 - x_2^2/2, x_2\right)$
- Experimental design: $\boldsymbol{\mathcal{X}}$ are generated using the Latin hypercube sampling
- No replications

PDF predictions (ED with $N = 500$)



Convergence study

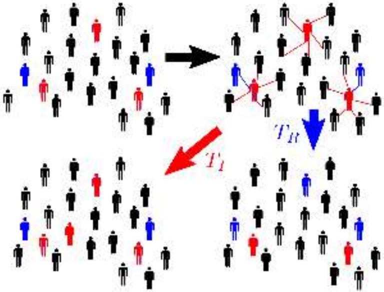
- Experimental design of size 250, 500, 1000, 2000, 4000
- 50 independent runs for each scenario
- Normalized Wasserstein distance as a performance indicator



Outline

Stochastic SIR model

- $M_t = S_t + I_t + R_t$: total population
- S_t: number of susceptible individuals at time t
- I_t: number of infected individuals at time t
- R_t: number of recovered individuals at time t

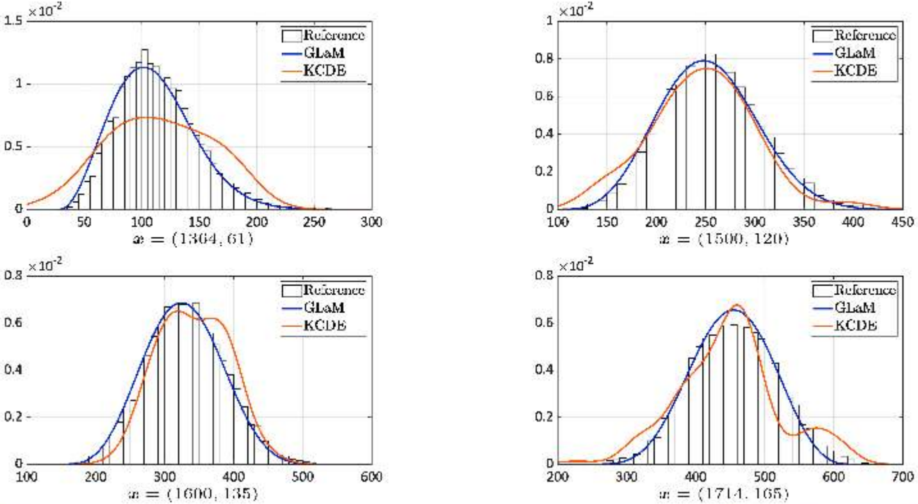


Setup

- Total population $M_t = 2000$
- The initial condition: $S_0 \sim \mathcal{U}(1300, 1800)$, $I_0 \sim \mathcal{U}(20, 200)$
- $Y(x)$: total number of infected individuals during the outbreak (without counting I_0)

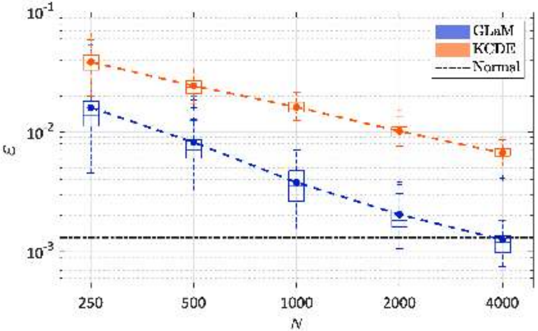
Binois et al. (2018) *Practical heteroscedastic Gaussian process modeling for large simulation experiments*, J. Comput. Graph. Stat.

PDF predictions (ED with $N = 500$)



Convergence study

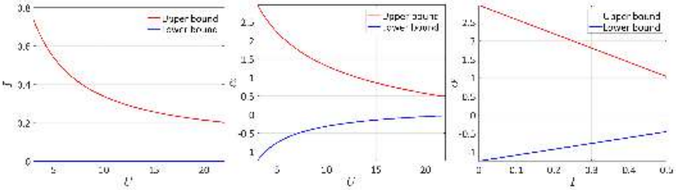
- Experimental design of size 250, 500, 1000, 2000, 4000
- 50 independent runs for each scenario
- Normalized Wasserstein distance as a performance indicator

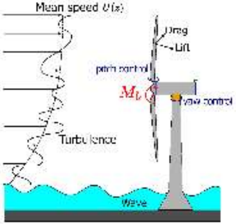


Outline

Wind turbine application

- **Five input variables:**
  - Mean speed U, turbulence intensity I and shear exponent α are uniformly distributed in the following domain





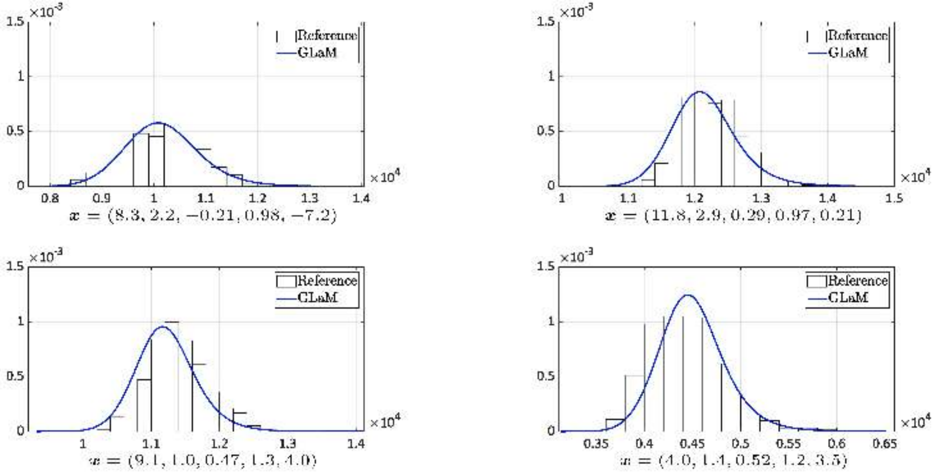
  - Air density $\rho \sim \mathcal{U}(0.8, 1.4)$, inclination angle $\beta \sim \mathcal{U}(-10, 10)$
- **Model output:** maximum flapwise bending moment $Y = M_b$

Simulation

- 485 training points with 50 replications
- 120 test points with 500 replications

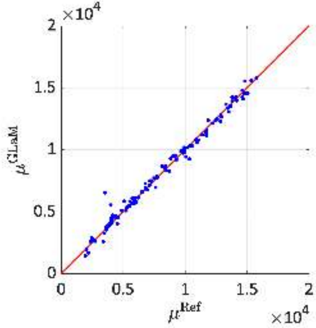
PDF predictions

The normalized Wasserstein distance is $\varepsilon = 0.013$

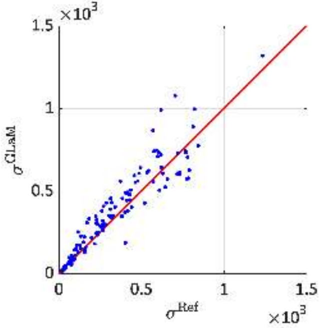


Predictions of quantities of interest

Mean and standard deviation



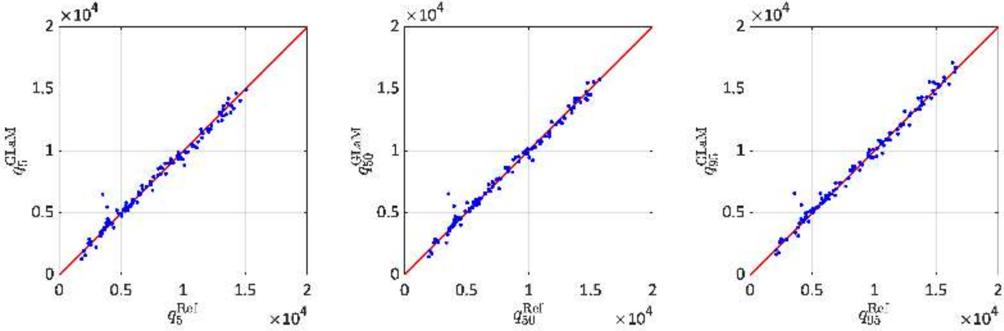
$$R^2 = 0.987$$



$$R^2 = 0.836$$

QoI predictions

Quantiles



$R^2 = 0.983$ $R^2 = 0.987$ $R^2 = 0.988$

Conclusions

- Stochastic simulators are used in many fields of applied sciences and engineering
- Building general-purpose emulators is necessary for optimization, sensitivity analysis, etc.
- We propose a framework based on generalized lambda distributions and polynomial chaos expansions
- Replications are not mandatory ... but can be used !
- Extensions with other surrogates (*e.g.* Gaussian processes) and sparse techniques are under investigation

Thank you very much for your attention!

Questions ?

Chair of Risk, Safety & Uncertainty Quantification

www.rsuq.ethz.ch

The Uncertainty Quantification Software

www.uqlab.com

The Uncertainty Quantification Community

www.uqworld.org