Polynomial Chaos Expansions for Wind Energy: Tutorial

Juan P. Murcia: PhD. student, jumu@dtu.dk

Pierre-E. Réthoré: Supervisor, Senior Scientist

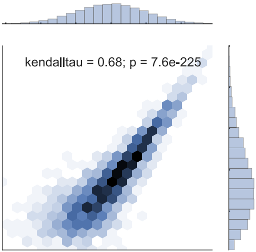
Outline

1. (A very short) Introduction to Polynomial Chaos Expansions
2. A1. 2D Not correlated Normal
3. A2. 2D MvNormal
4. A3. 2D Conditionally Correlated
5. A4. 2D Copula Correlated
6. Conclusions

Uncertainty propagation problem

Input variables
$\mathbf{x} \in \mathbb{R}^M$ → Model $\mathcal{M}(\mathbf{x})$ → Output variables $\mathbf{y} = \mathcal{M}(\mathbf{x}) \in \mathbb{R}^L$

Random inputs
$\mathrm{PDF}(\mathbf{x})$ → Model $\mathcal{M}(\mathbf{x})$ → **Random** outputs $\mathrm{PDF}(\mathbf{y})$



?

(A very short) Introduction to PCE

Monte-Carlo simulation

- Obtain a response sample by evaluating the model in each input realization: $\mathbf{y}_i = \mathcal{M}(\mathbf{x}_i)$
- Input sample, $\mathbf{x}_i$, can be generated using advanced sampling methods such as: Latin Hypercube sampling (LHS), Halton or Hammersley sequences.
- **Pros:** Very robust and easy to implement and parallelize
- **Cons:** Convergence is slow ($\propto N^{-1/2}$)

Polynomial Chaos expansion

- Build a polynomial surrogate of the model: $y(\mathbf{x}) \approx \sum c_l \, \mathbf{\Psi}_l(\mathbf{x})$
- A polynomial basis, $\mathbf{\Psi}_l(\mathbf{x})$, is built with respect to $\mathrm{PDF}(\mathbf{x})$
- The model is evaluated, $\mathbf{y}_i = \mathcal{M}(\mathbf{x}_i)$, and *projected/fitted* to the polynomial basis.
- The mean $\mathbb{E}(\mathbf{y})$, variance $\mathbb{V}(\mathbf{y})$ and Sobol's sensitivity index for each input S_i are obtain from the polynomial coefficients c_l.
- **Pros:** Convergence is fast ($\propto N^{-m}$, $m > 1$, m is problem dependent)
- **Cons:** How to define the order of the polynomials in each variable? How to avoid over fitting the model (Gibbs oscillations)?

(A very short) Introduction to PCE

Single uncertain variable x

$$y(x) \approx \sum_{l=0}^{P} c_l \, \boldsymbol{\Psi}_l(x)$$

Define an inner product using $\mathrm{PDF}(x)$:

$$\langle f, g \rangle = \int f(x)\, g(x)\, \mathrm{PDF}(x)\, dx$$

The polynomial basis is constructed such that $\boldsymbol{\Psi}_0 = 1$ and:

$$\langle \boldsymbol{\Psi}_l, \boldsymbol{\Psi}_k \rangle = \begin{cases} 1 & \text{if } l = k \\ 0 & \text{if } l \neq k \end{cases}$$

$$\langle 1, \boldsymbol{\Psi}_l \rangle = 0 \quad \forall l > 0 \qquad \Longleftrightarrow \qquad \int \boldsymbol{\Psi}_l(x)\, \mathrm{PDF}(x)\, dx = 0 \quad \forall l > 0$$

Methods to find the coefficients c_l

Semi-Spectral projection (quadrature integration)

- Use a quadrature rule to approximate the integrals (nodes, x_i and weights ω_i). Gaussian quadrature is widely used.

$$c_l = \langle y, \mathbf{\Psi}_l \rangle = \int y(x)\, \mathbf{\Psi}_l(x)\, \mathrm{PDF}(x)\, dx \approx \sum_{i=0}^{N} \omega_i\, y(x_i)\, \mathbf{\Psi}_l(x_i)$$

- **Pros:** Very good for low number of dimensions
- **Cons:** Unstable for heavy tailed PDFs. Quadrature rules fail with most correlated variables

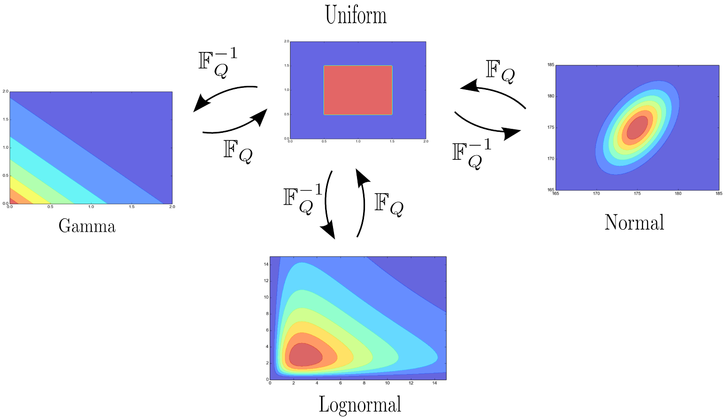
Point collocation (polynomial fit)

- Generate a small sample and fit the polynomial basis using Least squares or some other optimization method (e.g. LAR, LASSO).
- **Pros:** Very robust. Optimization algorithms are design to handle large number of dimensions (sparsity) and correlated inputs.
- **Cons:** Not as efficient as semi-spectral collocation.

How to deal with correlated inputs?

Rosenblatt Transformation [Rosenblatt 1952]

- Transforms the correlated input variables $(\mathbf{x})$ into a multi-dimensional uncorrelated uniform space $(\mathbf{w})$. Solve the propagation problem in the uncorrelated space: $\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbb{F}_Q^{-1}(\mathbf{w}))$. Use Legendre polynomials for uniform variables.
- It consists in using the inverse of the CDF of each variable in sequence. Chaospy includes this transformation [Feinberg 2015]. Graph reproduced from Chaospy tutorials.



Outline

Case A: A simple model
Model Description

Input variables		Model		Output variables
$\mathbf{x} = (I; a) \in \mathbb{R}^2$	$\rightarrow$	$\mathcal{M}(\mathbf{x}; t) = I\, e^{-a\,t}$	$\rightarrow$	$\mathbf{y} = \mathcal{M}(\mathbf{x}; t) \in \mathbb{R}^L$

Variables without uncertainty

- t: Location of the evaluation.

Uncertain variables

- I: Initial condition.
- a: Rate of dissipation.

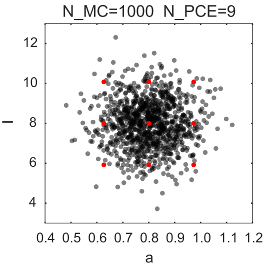
Case A1: A simple model

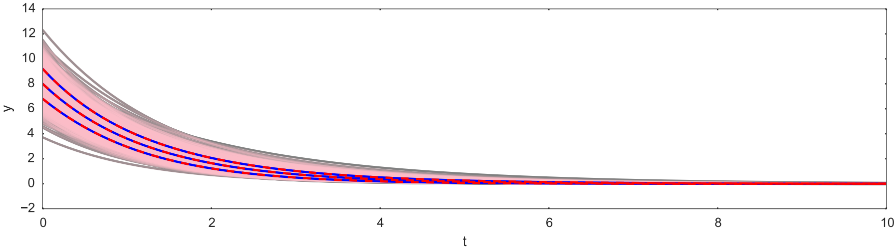
Inputs: 2D Not correlated Normal

$$y = I\, e^{-a\, t}$$

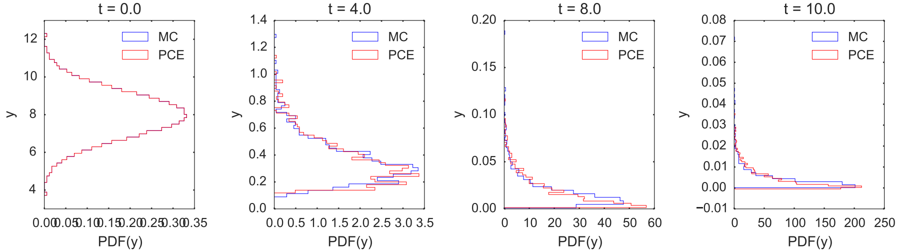
$$I \sim Normal(\mu_I = 8., \sigma_I = 2^{1/2})$$

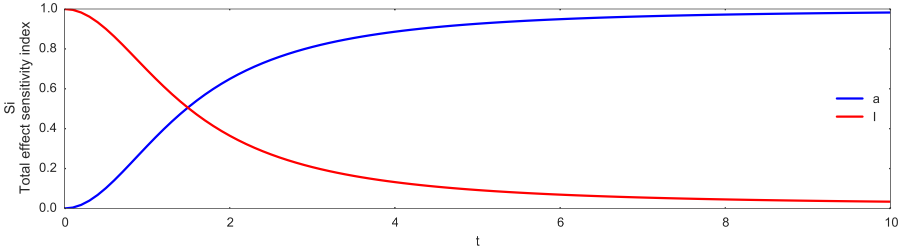
$$a \sim Normal(\mu_a = 0.8, \sigma_a = 0.01^{1/2})$$





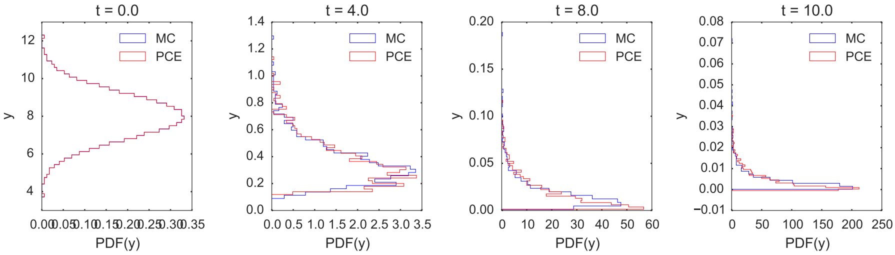
Case A1: A simple model
Inputs: 2D Not correlated Normal

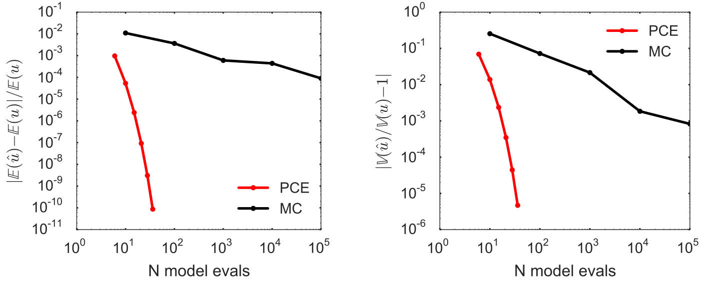




Case A1: A simple model

Inputs: 2D Not correlated Normal





Outline

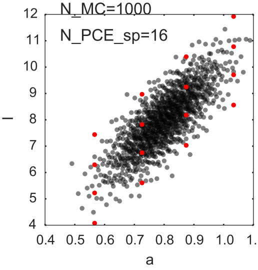
1. (A very short) Introduction to Polynomial Chaos Expansions
2. A1. 2D Not correlated Normal
3. A2. 2D MvNormal
4. A3. 2D Conditionally Correlated
5. A4. 2D Copula Correlated
6. Conclusions

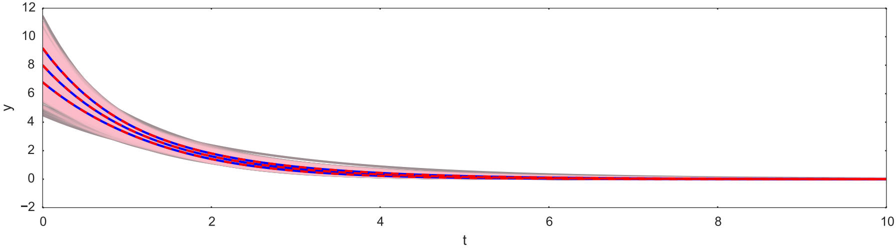
Case A2: A simple model
Inputs: 2D Correlated Normal

$$y = I\, e^{-a\, t}$$

$$\begin{bmatrix} a \\ I \end{bmatrix} \sim Normal(\mu, \mathbb{C})$$

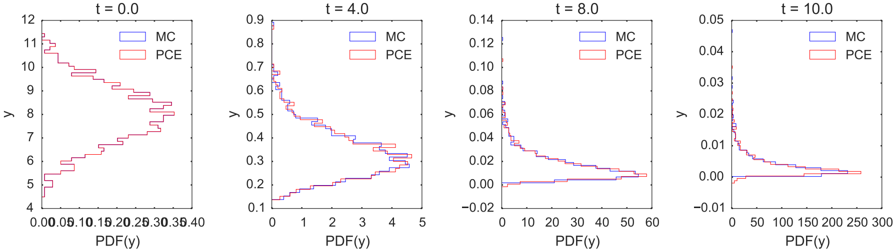
$$\mu = \begin{bmatrix} 0.8 \\ 8. \end{bmatrix} \quad \mathbb{C} = \begin{bmatrix} 0.01 & 0.1 \\ 0.1 & 2. \end{bmatrix}$$

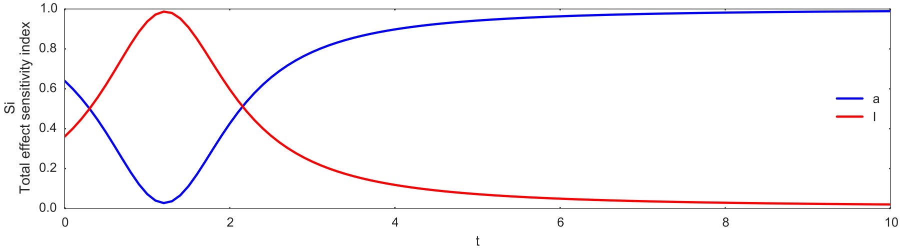




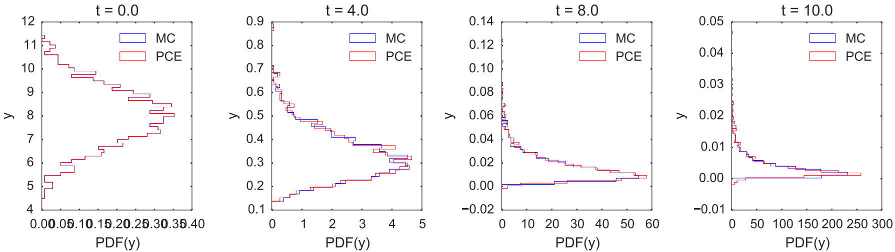
Case A2: A simple model

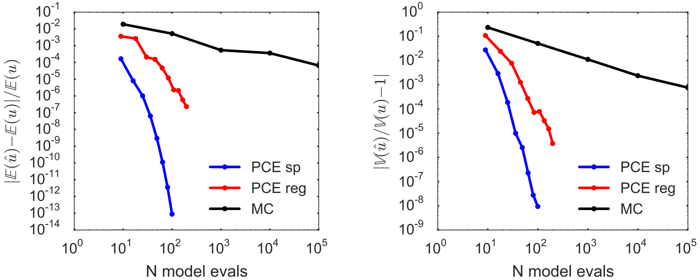
Inputs: 2D Correlated Normal





Case A2: A simple model
Inputs: 2D Correlated Normal





Outline

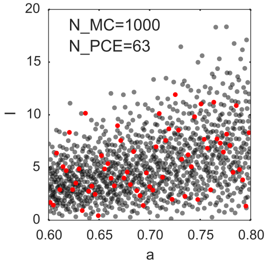
Case A3: A simple model

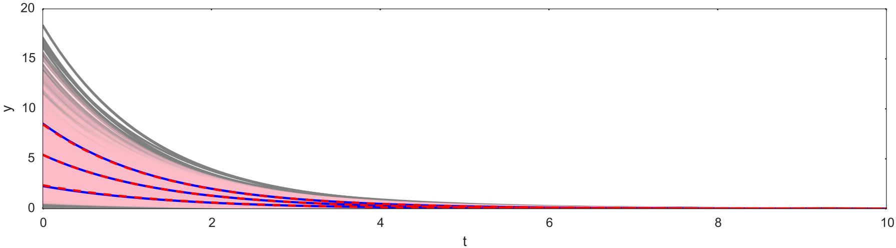
Inputs: 2D Conditionally Correlated

$$y = I\, e^{-a\,t}$$

$$a \sim Uniform(0.6, 0.8)$$

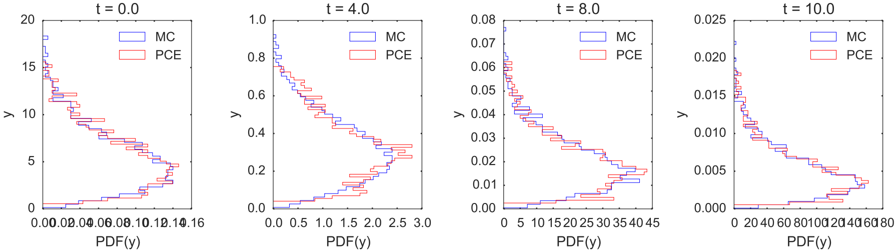
$$I \sim Weibull(k = 2.,$$
$$A = 6(0.3 + a)^4)$$

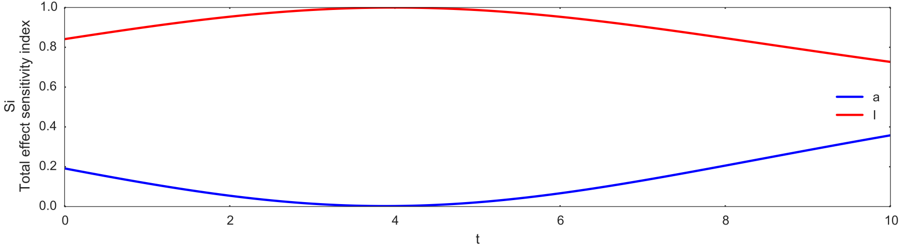




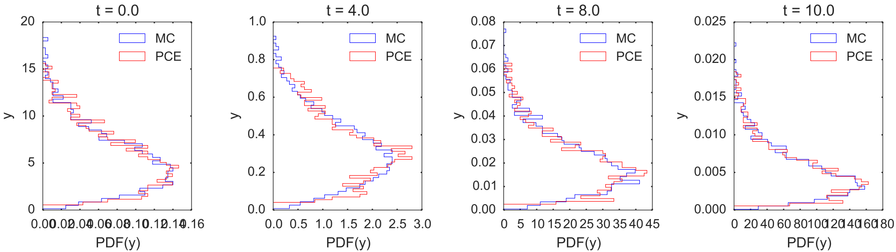
Case A3: A simple model

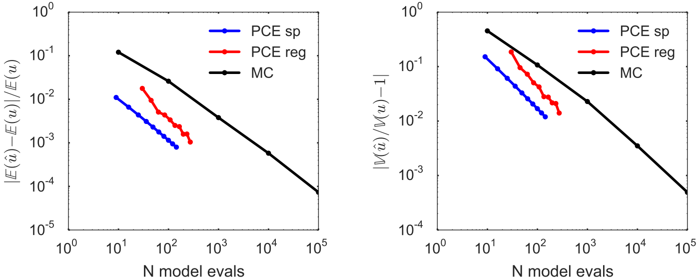
Inputs: 2D Conditionally Correlated





Case A3: A simple model
Inputs: 2D Conditionally Correlated





Options: A better surrogate and MC

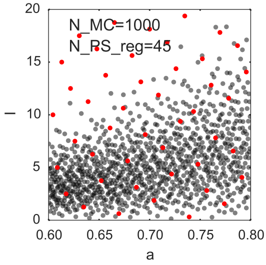
Case A3: A simple model

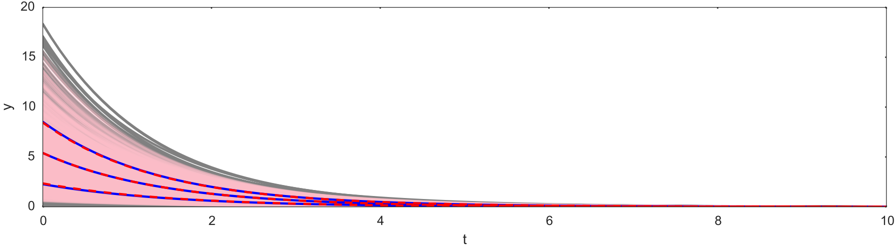
Inputs: 2D Conditionally Correlated - A better surrogate and MC

$$y = I\, e^{-a\, t}$$

$$a \sim Uniform(0.6, 0.8)$$

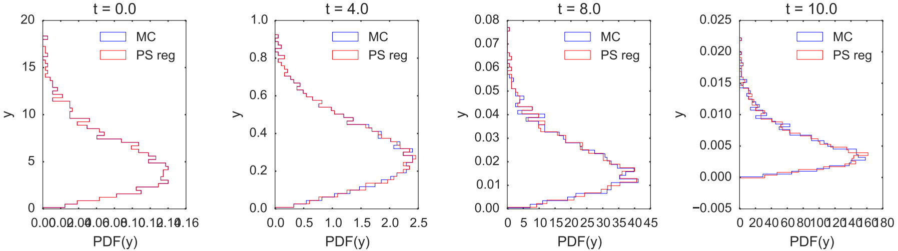
$$I \sim Weibull(k = 2.,$$
$$A = 6(0.3 + a)^4)$$

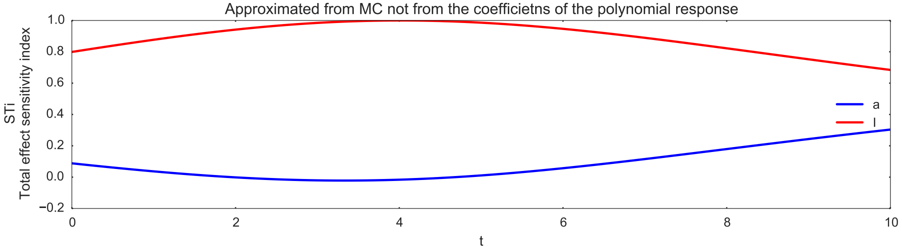




Case A3: A simple model

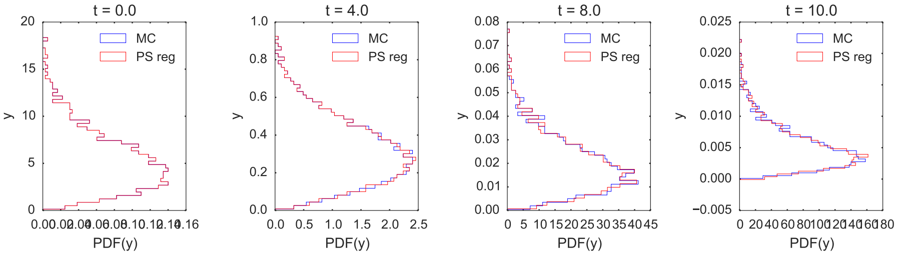
Inputs: 2D Conditionally Correlated - A better surrogate and MC

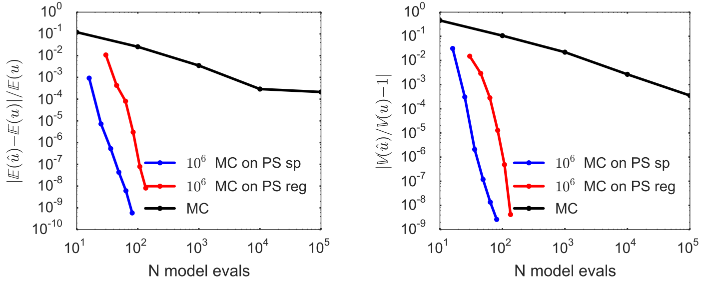




Case A3: A simple model

Inputs: 2D Conditionally Correlated - A better surrogate and MC





Outline

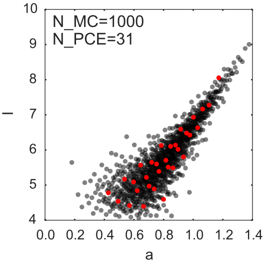
Case A4: A simple model
Inputs: 2D Copula Correlated

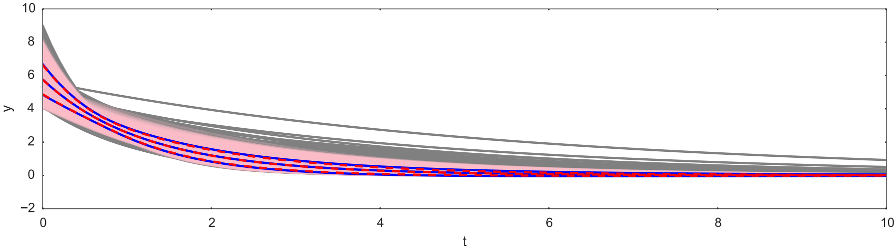
$$y = I\, e^{-a\, t}$$

$$a \sim Uniform(0.6, 0.8)$$

$$I \sim Weibull(k = 2., A = 2)$$

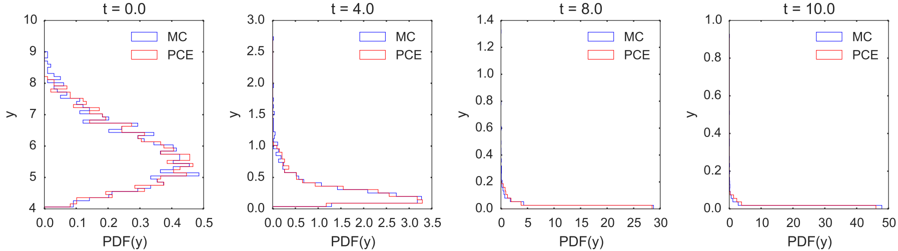
$$Joe(CDF(a), CDF(I), theta = 5.)$$

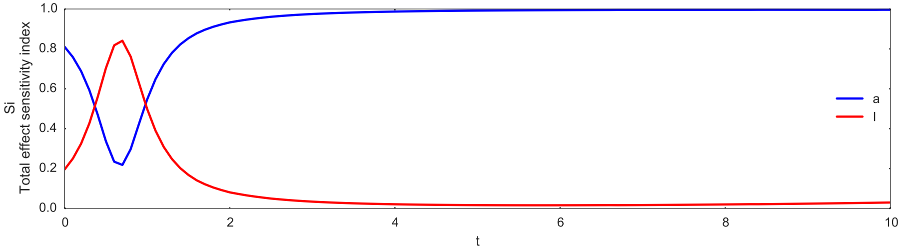




Case A4: A simple model

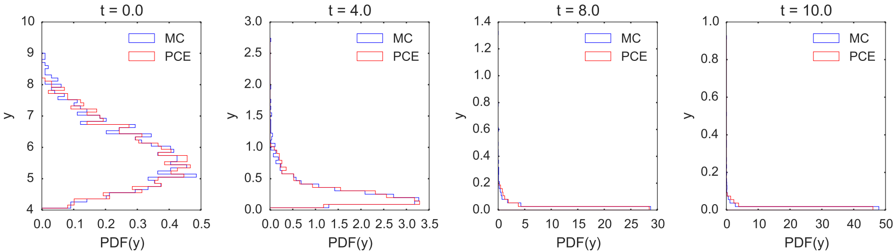
Inputs: 2D Copula Correlated

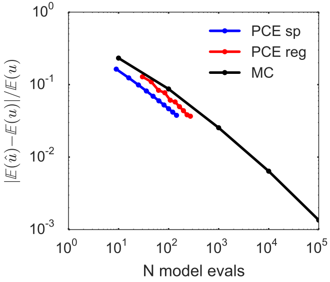


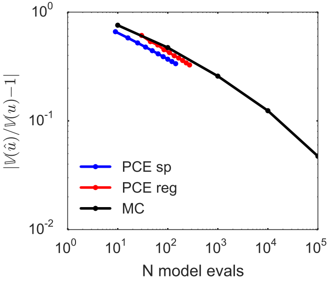


Case A4: A simple model

Inputs: 2D Copula Correlated







Options: A better surrogate and MC

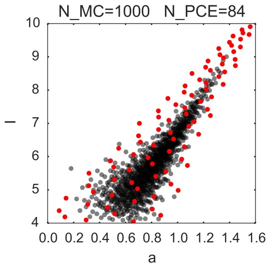
Case A4: A simple model
Inputs: 2D Copula Correlated - A better surrogate and MC

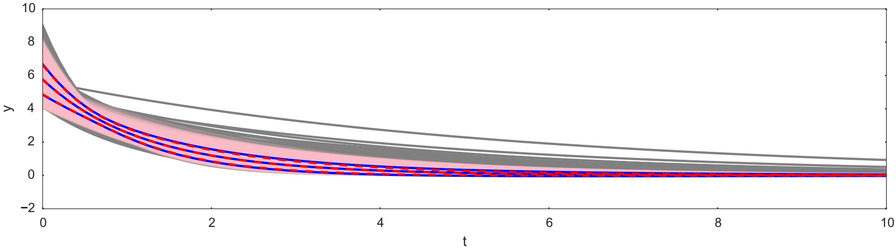
$$y = I\, e^{-a\, t}$$

$$a \sim Uniform(0.6, 0.8)$$

$$I \sim Weibull(k = 2., A = 2)$$

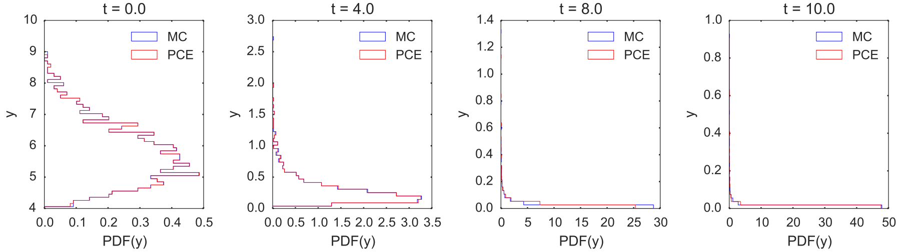
$$Joe(CDF(a), CDF(I), theta = 5.)$$

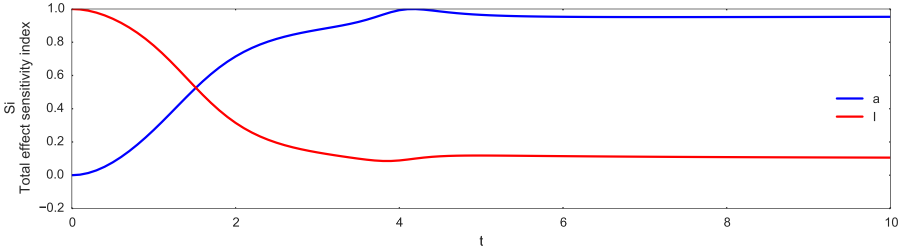




Case A4: A simple model

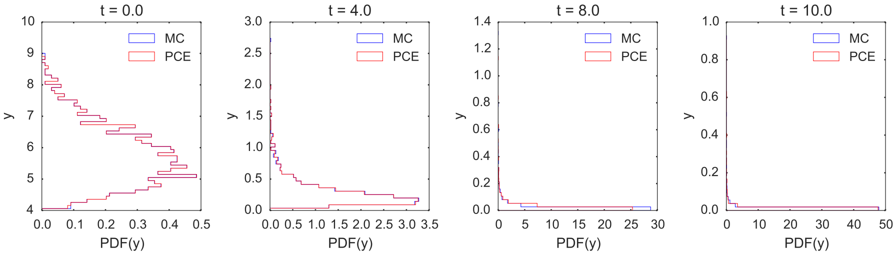
Inputs: 2D Copula Correlated - A better surrogate and MC

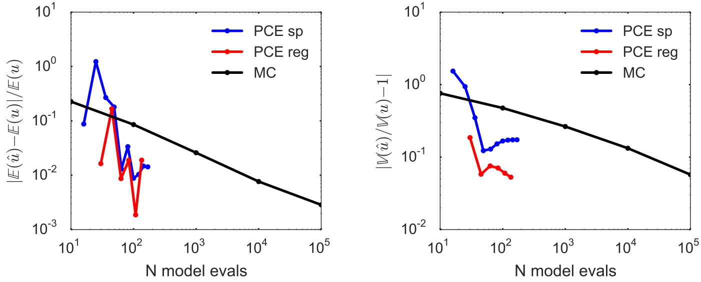




Case A4: A simple model

Inputs: 2D Copula Correlated - A better surrogate and MC



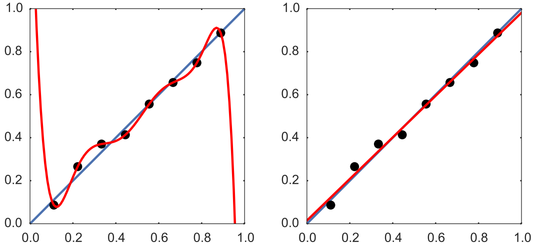


Outline

How to avoid over fitting and achieve sparsity?
Sparse linear model regression

- Least Absolute Shrinkage and Selection Operator problem (LASSO) is useful to avoid over-fitting and achieve sparsity in the PCE.
- LASSO is a least squares minimization problem with a l_1 penalization on the coefficients ($\mathbf{c}$):

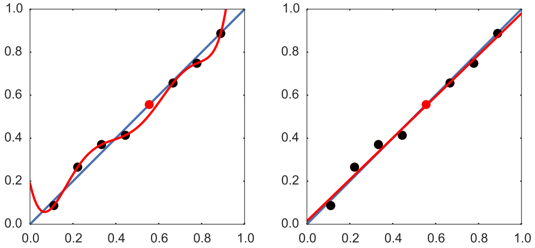
$$\min_{\mathbf{c}} ||\mathbf{c}\boldsymbol{\Psi} - \mathbf{y}_k||_2^2 + \alpha_k ||\mathbf{c}||_1 = \min_{\mathbf{c}} \sum_{i=0}^{N-1} \left[\sum_{l=0}^{N_c-1} c_l \boldsymbol{\Psi}_l(\mathbf{w}_i) - y_k(\mathbf{x}_i) \right]^2 + \alpha_k \sum_{l=0}^{N_c-1} |c_l|$$



How to select the right sparsity?

k-fold cross validation

- It divides the dataset in k groups and uses k-1 groups ("folds") for training and the remaining for validation. Repeat this process until all the groups have been the validation set.
- k-fold cross validation is repeated for multiple values of the sparsity parameter.
- As a result it gives the optimal sparsity parameter (α_k).



PCE of complex cases

Variable transformations steps

Correlated inputs PDF($\mathbf{x}$) → Rosenblatt transformation → Uniform **independent** PDF($\mathbf{w}$) → PCE $\mathbf{z} = \hat{\mathcal{M}}(\mathbf{w})$ → Overlimited outputs PDF($\mathbf{z}$) → Logistic transformation → **Correlated** outputs PDF($\mathbf{y}$)

Rosenblatt transformation

- Used to decorrelate the variables
- Variables can be transformed to independent Uniform or Normal
- Inverse transformation used for MC sample. Use efficient sampling techniques in the unitary uniform uncorrelated space.

PCE model surrogate

- Polynomial chaos expansion working on the uncorrelated space.
- Trained using k-Fold validation to avoid over-fitting and prefer lower order polynomials (Least absolute shrinkage and selection operator - LASSO problem).

Logistic transformation

- Used to force fixed constrains in the outputs: i.e. to avoid overshoots.
- Can be used to smooth discontinuities and to impose only positive values.

References I

[Rosenblatt 1952] Rosenblatt, M 1952 Remarks on a multivariate transformation. Annals of Mathematical Statistics Vol. 23, pp 470-472.]

[Feinberg 2015] Feinberg, J., & Langtangen, H. P. (2015). Chaospy: An open source tool for designing methods of uncertainty quantification. Journal of Computational Science, 11, 46-57.

Questions?

J. P. Murcia
+45 2339 7790
jumu@dtu.dk
PhD Student
DTU Wind Energy

Technical University of Denmark (DTU)
Building 101
Risø Campus
Frederiksborgvej 399
4000 Roskilde, Denmark