Surrogate Modeling

Prof. Americo Cunha Jr

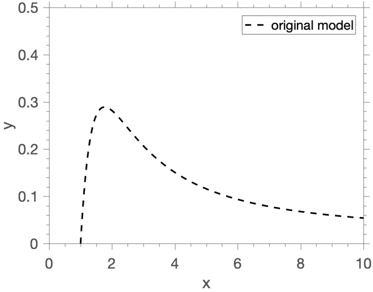
americocunha.org

@AmericoCunhaJr

@AmericoCunhaJr @AmericoCunhaJr

What is a surrogate model ?

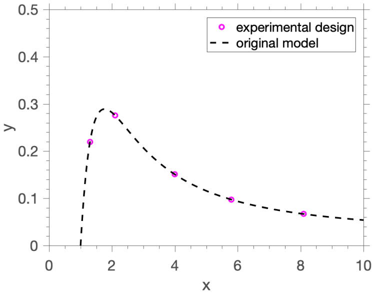
A surrogate model (a.k.a. metamodel) is an approximation for the original model which is constructed from a limited number of original model executions.



B. Sudret, **Recent developments in surrogate modelling for uncertainty quantification**, 2018.

What is a surrogate model ?

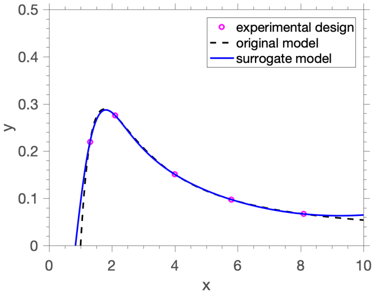
A surrogate model (a.k.a. metamodel) is an approximation for the original model which is constructed from a limited number of original model executions.



B. Sudret, **Recent developments in surrogate modelling for uncertainty quantification**, 2018.

What is a surrogate model ?

A surrogate model (a.k.a. metamodel) is an approximation for the original model which is constructed from a limited number of original model executions.

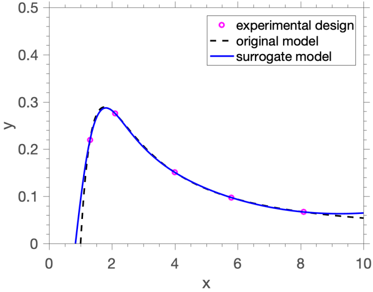


B. Sudret, **Recent developments in surrogate modelling for uncertainty quantification**, 2018.

What is a surrogate model ?

A surrogate model (a.k.a. metamodel) is an approximation for the original model which is constructed from a limited number of original model executions.

It can be thought as a kind of "response surface" that approximates well the original model behavior (locally).



B. Sudret, **Recent developments in surrogate modelling for uncertainty quantification**, 2018.

Surrogate model features

$$\underset{\text{hours/days per run}}{\mathcal{M}(\mathbf{x})} \approx \underset{\text{seconds for } 10^6 \text{ runs}}{\tilde{\mathcal{M}}(\mathbf{x})}$$

Advantages:

- Non-intrusive calculation
- Embarrassingly parallel

Drawbacks:

- Demands rigorous validation
- The math is hard!

Context of use:

- Uncertainty propagation
- Parametric studies
- Optimization
- Inversion
- ...

B. Sudret, **Recent developments in surrogate modelling for uncertainty quantification**, 2018.

Types of surrogates

- Polynomial chaos expansions
- Low-rank tensor approximations
- Gaussian processes interpolation (a.k.a Kriging)
- Support vector machines
- Neural networks
- many others

B. Sudret, **Recent developments in surrogate modelling for uncertainty quantification**, 2018.

Types of surrogates

- **Polynomial chaos expansions**
- Low-rank tensor approximations
- Gaussian processes interpolation (a.k.a Kriging)
- Support vector machines
- Neural networks
- many others

B. Sudret, **Recent developments in surrogate modelling for uncertainty quantification**, 2018.

Polynomial Chaos Expansion (PCE) surrogate

$$Y = \sum_{\alpha=0}^{\infty} y_\alpha \, \psi_\alpha(\mathbf{X})$$

- y_α – real numbers to be determined
- $\psi_\alpha(\mathbf{X})$ – multi-dimensional orthonormal polynomials

Properties:

- $\mathbb{E}\{\psi_0\} = 1$
- $\mathbb{E}\{\psi_\alpha\} = 0 \quad \alpha > 0$
- $\mathbb{E}\left\{\psi_\alpha \psi_\beta\right\} = \delta_{\alpha\beta}$
- $y_\alpha = \mathbb{E}\left\{ Y \, \psi_\alpha(\mathbf{X}) \right\}$

Computational implementation of PCE

$$Y \approx \sum_{\alpha=0}^{P} y_\alpha \, \psi_\alpha(\mathbf{X})$$

- $P + 1$ coefficients
- $P = (K + N)!/(K!N!)$
- N is the dimension of $\mathbf{X}$
- K the highest degree of the polynomials $\psi_\alpha\,(\mathbf{X})$
- P grows very fast with N (curse-of-dimensionality)

PCE via projection

1. Expand Y via PCE

$$Y \approx \sum_{\alpha=0}^{P} y_\alpha \, \psi_\alpha(\mathbf{X})$$

2. Plug the expansion into model equation

$$\sum_{\alpha=0}^{P} y_\alpha \, \psi_\alpha(\mathbf{X}) = \mathcal{M}(\mathbf{X})$$

3. Project into span $< \psi_0, \cdots, \psi_P >$

$$y_\beta = \mathbb{E}\left\{\mathcal{M}(\mathbf{X})\psi_\beta\right\}, \quad \beta = 0, \cdots, P$$

4. Compute the (deterministic) coefficients
   (quadrature schemes, Monte Carlo integration, etc)

PCE via regression

Obtain M samples of Y computed from i.i.d. realizations of $\mathbf{X}$:
(this is a Monte Carlo calculation)

$$\begin{array}{ccc} \mathbf{X}_1 & \longrightarrow & Y_1 = \mathcal{M}(\mathbf{X}_1) \\ \vdots & & \vdots \\ \mathbf{X}_M & \longrightarrow & Y_M = \mathcal{M}(\mathbf{X}_M) \end{array}$$

Is it possible to estimate y_α from these M samples ?

PCE via regression

$$Y_\beta = \mathcal{M}(\mathbf{X}_\beta) \approx \sum_{\alpha=0}^{P} y_\alpha \, \psi_\alpha(\mathbf{X}_\beta), \quad \beta = 1, \cdots, M$$

$$\Longleftrightarrow$$

$$\begin{bmatrix} \psi_0\left(\mathbf{X}_1\right) & \cdots & \psi_P\left(\mathbf{X}_1\right) \\ \vdots & \ddots & \vdots \\ \psi_0\left(\mathbf{X}_M\right) & \cdots & \psi_P\left(\mathbf{X}_M\right) \end{bmatrix} \begin{bmatrix} y_0 \\ \\ \vdots \\ \\ y_P \end{bmatrix} \approx \begin{bmatrix} \mathcal{M}(\mathbf{X}_1) \\ \vdots \\ \mathcal{M}(\mathbf{X}_M) \end{bmatrix}$$

$$P + 1 \gg M \text{ (in general)}$$

Least-squares problem

Simple is beautiful !

PCE error estimation

- Experimental design

$$\mathcal{X} = \{\mathbf{X}_\beta \, , \beta = 1 \, , \, \cdots \, , M\}$$

- Empirical error

$$E_{emp} = \frac{1}{M} \sum_{\beta=1}^{M} \Big(\mathcal{M}(\mathbf{X}_\beta) - \mathcal{M}^{\text{PCE}}(\mathbf{X}_\beta) \Big)^2$$

- Coefficient of determination

$$R^2 = 1 - \frac{E_{emp}}{Var\{Y\}}$$

$$Var\{Y\} = \frac{1}{M} \sum_{\beta=1}^{M} \left(\mathcal{M}(\mathbf{X}_\beta) - \bar{Y} \right)^2 \quad \bar{Y} = \frac{1}{M} \sum_{\beta=1}^{M} \mathcal{M}(\mathbf{X}_\beta)$$

PCE error estimation

- Experimental design

$$\mathcal{X} = \{\mathbf{X}_\beta \,, \beta = 1 \,, \cdots \,, M\}$$

- Empirical error

$$E_{emp} = \frac{1}{M} \sum_{\beta=1}^{M} \Big(\mathcal{M}(\mathbf{X}_\beta) - \mathcal{M}^{\text{PCE}}(\mathbf{X}_\beta) \Big)^2$$

- Coefficient of determination

$$R^2 = 1 - \frac{E_{emp}}{Var\{Y\}}$$

$$Var\{Y\} = \frac{1}{M} \sum_{\beta=1}^{M} \left(\mathcal{M}(\mathbf{X}_\beta) - \bar{Y} \right)^2 \quad \bar{Y} = \frac{1}{M} \sum_{\beta=1}^{M} \mathcal{M}(\mathbf{X}_\beta)$$

Both are poor estimators in case of overfitting!

Leave-one-out (LOO) cross-validation

- Experimental design

$$\mathcal{X} = \{\mathbf{X}_\beta \,, \beta = 1 \,, \cdots \,, M\}$$

- Construct a family of PCE using all points but one, i.e.

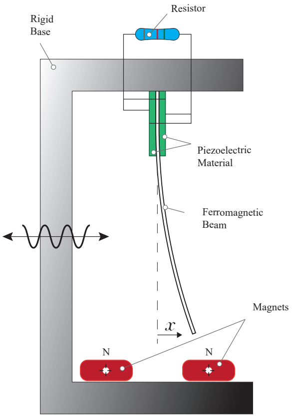
$$\mathcal{X} \backslash \alpha = \{\mathbf{X}_\beta \,, \beta = 1 \,, \cdots \,, M \; \beta \neq \alpha\}$$

- Leave-one-out error

$$E_{LOO} = \frac{\displaystyle\sum_{\beta=1}^{M} \left(\mathcal{M}(\mathbf{X}_\beta) - \mathcal{M}^{\mathrm{PCE}\backslash\beta}(\mathbf{X}_\beta) \right)^2}{\displaystyle\sum_{\beta=1}^{M} \left(\mathcal{M}(\mathbf{X}_\beta) - \bar{Y} \right)^2}$$

- Choose the PCE with the least-error

Bistable Energy Harvester



$$\ddot{x} + 2\xi\dot{x} - \frac{1}{2}x(1 - x^2) - \chi v = f\cos(\Omega t)$$

$$\dot{v} + \lambda v + \kappa\dot{x} = 0$$

$$+ \ \text{initial conditions}$$

Mean power:

$$P_{avg} = \frac{1}{T}\int_{t_o}^{t_o+T} \lambda\, v\,(t)^2 \; dt$$

J. P. Norenberg, A. Cunha Jr, S. da Silva, P. S. Varoto, *Global sensitivity analysis of (a)symmetric energy harvesters*, **arXiv:2107.04647**, 2021. https://arxiv.org/abs/2107.04647

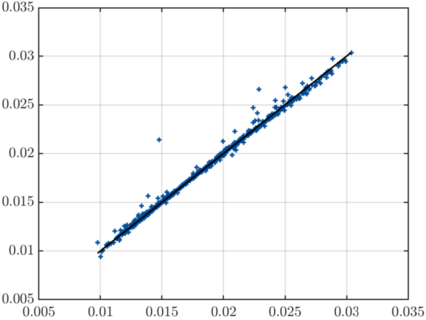
Model parameters

Parameter	Distribution	c.v.	Nominal value
ξ	Uniform	20%	0.01
χ	Uniform	20%	0.05
λ	Uniform	20%	0.05
κ	Uniform	20%	0.5
f	Constant	—	0.200
Ω	Constant	—	0.8

Example prepared by J. P. Norenberg

PCE validation (full-order model vs surrogate)

Number of samples: 500

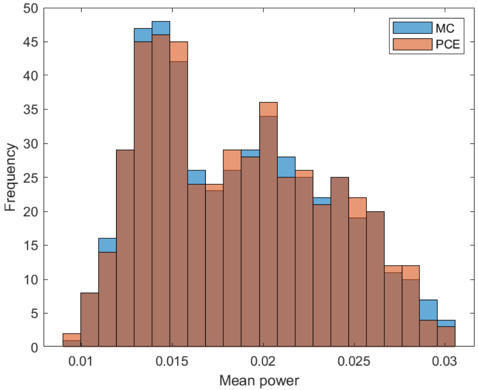


PCE degree	8
Exp. Design	100
LOO error	1.639×10^{-3}
training time*	54.02 s

*Intel i7-9750H 2.60GHz 8GB 2666GHz DDR4

Example prepared by J. P. Norenberg

Monte Carlo x PCE



MC time *	133.18 s
PCE time*	0.0382 s

*Intel i7-9750H 2.60GHz 8GB 2666GHz DDR4

`Example prepared by J. P. Norenberg`

Poisson problem with random conductivity

$$
\begin{aligned}
(\xi + 2)\, \nabla^2 u(x,y) &= 1 \qquad (x,y) \in \quad \mathcal{D} \subset \mathbb{R}^2 \\
u(x,y) &= 0 \qquad (x,y) \in \partial\mathcal{D}
\end{aligned}
$$

Deterministic solver: Finite difference with 16×16 uniform mesh
Probabilistic model: $\xi \sim \mathcal{U}(-1,1)$
PCE:

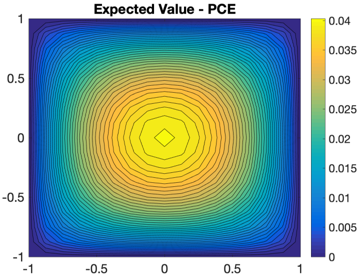
$$
u(x,y) \approx \sum_{\beta=0}^{P} y_\beta(x,y)\, \psi_\beta(\xi), \quad P = (K+1)!/K!
$$

- $y_\beta(x,y)$ are deterministic functions of x and y
- $y_\beta(x,y)$ are computed via Galerkin projection
- $\psi_k(\xi)$ are Legendre polynomials up to degree $K = 3$

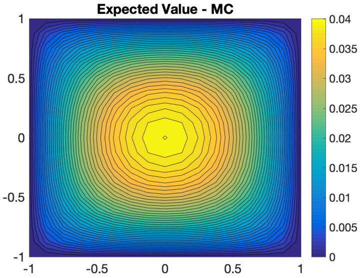
P. Constantine, **A Primer on Stochastic Galerkin Methods**, Lecture Notes, 2007.

Poisson problem with random conductivity

temperature mean value



(a) Polynomial Chaos

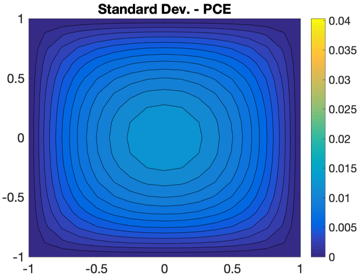


(b) Monte Carlo Method

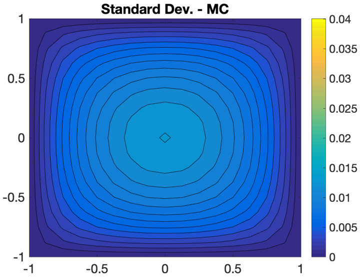
P. Constantine, **A Primer on Stochastic Galerkin Methods**, Lecture Notes, 2007.

Poisson problem with random conductivity

temperature standard deviation



(c) Polynomial Chaos



(d) Monte Carlo Method

P. Constantine, **A Primer on Stochastic Galerkin Methods**, Lecture Notes, 2007.

Poisson problem with random conductivity

Maximum norm of the difference between PCE and MC solutions:

K	mean value	standard deviation
3	0.00026079	0.00040902
5	0.00023984	0.00023604
7	0.00023973	0.00023440
10	0.00023972	0.00023439
20	0.00023972	0.00023439

P. Constantine, **A Primer on Stochastic Galerkin Methods**, Lecture Notes, 2007.

Poisson problem with random conductivity

Speed-up of PCE compared to MC solution (2600 samples):

K	speed-up*
3	4.1
5	3.7
7	3.5
10	2.0
20	1.3

* PCE educational code, not optimized for HPC

P. Constantine, **A Primer on Stochastic Galerkin Methods**, Lecture Notes, 2007.

Polynomial chaos software

- UQLab
  http://www.uqlab.com
- Korali
  https://www.cse-lab.ethz.ch/korali
- MUQ
  http://muq.mit.edu
- DAKOTA
  http://dakota.sandia.gov
- UQ Toolkit
  http://www.sandia.gov/UQToolkit
- UQ-PyL
  http://www.uq-pyl.com
- Chaospy
  http://github.com/jonathf/chaospy

References

P. Constantine, **A Primer on Stochastic Galerkin Methods**, Lecture Notes, 2007.

N. Lüthen, S. Marelli and Bruno Sudret, *Sparse Polynomial Chaos Expansions: Literature Survey and Benchmark*, **SIAM/ASA Journal on Uncertainty Quantification**, 9: 593-649, 2021.

R. G. Ghanen and P. D. Spanos, **Stochastic Finite Element Method: A Spectral Approach**, Dover Publications, Revised Version, 2003.

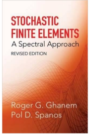
D. Xiu, **Numerical Methods for Stochastic Computations: A Spectral Method Approach**, Princeton Universty Press, 2010.

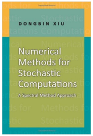
O. Le Maitre and O. M. Knio, **Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics**, Springer, 2010.

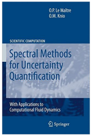
M. P. Pettersson, G. Iaccarino amd J. Nordström, **Polynomial Chaos Methods for Hyperbolic Partial Differential Equations: Numerical Techniques for Fluid Dynamics Problems in the Presence of Uncertainties** , Springer, 2015.

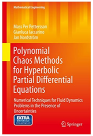
R. Ghanem, D. Higdon and H. Owhadi (Editors), **Handbook of Uncertainty Quantification**, Springer, 2017.

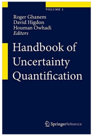
P. Benner, M. Ohlberger, A. Cohen, K. E. Wilcox (Editors), **Model Reduction and Approximation: Theory and Algorithms**, SIAM, 2017.

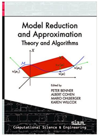












How to cite this material?

A. Cunha Jr, *Surrogate Modeling*, 2021.

@AmericoCunhaJr

@AmericoCunhaJr

@AmericoCunhaJr