Performance of Surrogate Modeling Techniques in Structural Reliability

Henrique M. Kroetz
Graduate Student, Dept. of Structural Engineering, University of São Paulo, São Carlos, Brazil

André T. Beck
Assoc. Prof., Dept. of Structural Engineering, University of São Paulo, São Carlos, Brazil

ABSTRACT: Solution of structural reliability and uncertainty propagation problems by Monte Carlo simulation can be a demanding task, since complex mechanical models usually have to be solved repeated times. Therefore, surrogate models are often required to reduce the computational burden. This article compares the performance of three surrogate modeling techniques in the solution of structural reliability problems. It addresses artificial neural networks, polynomial chaos and kriging meta-modeling, associated with LHS and Monte-Carlo simulation. A simple procedure for mapping input data for uncertainty quantification problems is also proposed.

Structural reliability problems often involve complex mechanical models, usually associated to low failure probabilities. Most computational methods used to solve such problems (e.g. Monte Carlo simulations) require repeated calls of a numerical model of the structural system (e.g. Finite Element, Boundary Element models). Solving reliability problems this way can be a very demanding task, leading to impracticable computational costs. In the past few decades, these difficulties led researchers to search for alternative approaches and opened the so-called meta-modeling field of research.

The first meta-modeling applications in structural reliability involved polynomial response surfaces, and in this context, the original methods were called *Response Surface Methods*. More advanced techniques have been developed over time, that are able to solve the same problems with more complex, yet greater accuracy surrogate models.

The basic idea is to replace the original detailed numerical model with a substitute, which mimics its behavior, having similar or even identical outputs, for the same input data. In general, a design of experiment is selected following some sampling strategy (e.g. randomly, a design of experience, Latin Hypercube sample) and the surrogate model is calibrated using this information. The next step is to solve the problem considering no longer the original model, but the surrogate one.

This paper presents three meta-modeling techniques that have been successfully employed in structural reliability problems, and aims to compare their performance. Artificial Neural Networks (ANN), Polynomial Chaos Expansions (PCE) and Kriging are briefly explained, and then used to solve the same problems, and their accuracy is compared for fixed amounts of training data.

A simple data mapping procedure that has shown to be useful for ANN and Kriging is also discussed.

1. PROBLEM STATEMENT

Let $\boldsymbol{X}$ be a vector that gathers together the m random input parameters of the model (e.g. geometrical and material properties, loads) with prescribed density function $f_{\boldsymbol{X}}(\boldsymbol{x})$. The uncertainty implies in the possibility of undesirable structural responses, mathematically described by a limit state function $g(\boldsymbol{x})$, such that:

$\Omega_f = \{\boldsymbol{x}|g(\boldsymbol{x}) \leq 0\}$ is the failure domain (1)

$\Omega_s = \{\boldsymbol{x}|g(\boldsymbol{x}) > 0\}$ the survival domain (2)

The probability of failure of the system is defined by:

$$P_f = \int_{\Omega_f} f_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \quad (3)$$

The integral represented by Eq. (3) usually requires numerical solutions, since its integration domain is implicitly defined, hence the usage of Monte Carlo simulations (MCS). In this technique, the failure probability is interpreted as the mean value of a stochastic experiment where a large number of samples of the random variables are generated *(Ditlevsen and Madsen, 2007).* An indicator function $I(\boldsymbol{x})$, that assumes value 0 over the survival domain and 1 over de failure domain is considered, and n samples of $\boldsymbol{X}$ are generated following $f_{\boldsymbol{X}}(\boldsymbol{x})$. The probability of failure is then estimated as:

$$P_f = \frac{1}{n}\sum_{i=1}^{n} I(\boldsymbol{x}_i) \quad (4)$$

Since in MCS convergence is achieved with increasing values of n, it may be useful to evaluate at this point simpler surrogate models instead of the original mechanical models, reducing the computational burden.

2. META-MODELING TECHNIQUES

2.1 Artificial Neural Networks (ANNs)

ANNs are numerical algorithms inspired on the structure of the brain. They have been applied in structural mechanics by *Berke and Haleja (1992)* and *Bento (1998)*, and coupled with MCS in the works of *Papadrakakis et al. (1996), Elhewy et al. (2006)* and *Cardoso et al. (2008)*.

A neuron is a logical structure that receives information subjected to characteristic weight functions w. In the model adopted in this work, for a neuron n receiving data from L input channels, the neuron weighted input information is subjected to a sum function, resulting in a linear combination of all the received information. A bias term b is added to this result, so that the neuron works with non zero values even when all the input data are zero. This sum results in the neuron activation value a. Equation (5) gives the activation value of a neuron n.

$$a_n = \sum_{i=1}^{L} w_{ni}x_i + b_n \quad (5)$$

The neuron output information is obtained subjecting the activation value to a *transfer* or *activation* function f, usually an S shaped function such as hyperbolic, sigmoid, tangent or similar.

Neurons are organized in layers, composing the network. The capability to recognizing patterns, and therefore working as a surrogate model, is defined by the architecture of the network (number of layers and number of neurons per layer), the available training data, and its training algorithm, the later defining weight w for each neuron. Usually, some rule is adopted for initialization of the weights, and an algorithm corrects the weights based on available data about the original model. In the present work, feed-forward networks are used, meaning that the neurons may only send the output information to their next layer. Figure 1 illustrates the operation of a feed forward multi layer neural network.

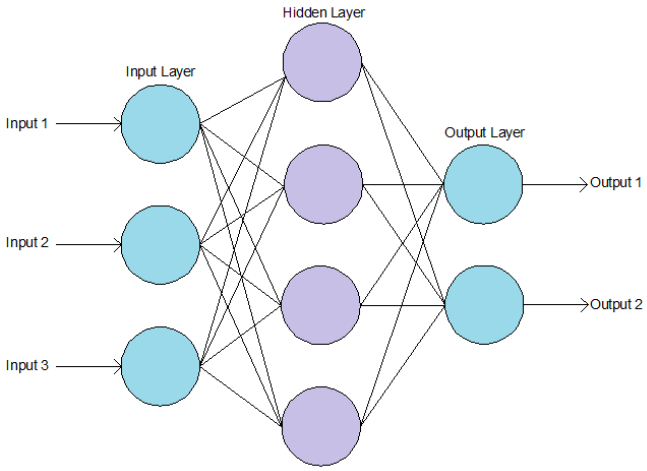

Figure 1: Feed forward multi layer neural network.

2.1. Polynomial Chaos Expansions (PCE)

PCE has been used since de 90's in the solution of stochastic finite elements problems, but only in the early 2000's the expansions began to be seen as meta-models. Broader reviews can be found in the works of *Panacyrici and Schuëller (2011)* and *Sudret (2012).*

In this section, the input random variables will be considered independent for a matter of simplicity.

With PCE, the response *Y* of a structural system is considered as a random variable that belongs to a specific space (e.g. space of random variables with finite variance), and can be represented in a basis of this space, as shown in equation (6):

$$Y = \sum_{\boldsymbol{\alpha}} a_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}} \tag{6}$$

The $\Psi_{\boldsymbol{\alpha}}$ are the multivariate polynomial basis functions, that are orthonormal with respect to $f_{\boldsymbol{X}}(\boldsymbol{x})$, so that :

$$E\left[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \Psi_{\boldsymbol{\beta}}(\boldsymbol{X})\right] = \delta_{\alpha\beta} \tag{7}$$

where $\delta_{\alpha\beta}$ = 1 if $\alpha = \beta$ and 0 otherwise. The $a_{\boldsymbol{\alpha}}$ are coefficients to be computed. In the case of independent input random variables, they can be transformed in standard normal random variables, so that $\boldsymbol{X}$ is represented by a standard normal Gaussian vector $\boldsymbol{\xi} = (\xi_1, \xi_2, \dots, \xi_m)$. Hermite polynomials are a suitable option for the orthonormal basis functions. For practical computational reasons, the series must be truncated and only the polynomials of degree p or less are adopted. Such basis can be obtained from the tensorization of the truncated univariate expansions of the m input random variables that compose $\boldsymbol{X}$. The number of *P* basis functions adopted this way is given by:

$$P = \frac{(m+p)!}{m!p!} \tag{8}$$

A practical algorithm to build the multivariate basis is presented in detail by *Sudret et al. (2006).*

The $a_{\boldsymbol{\alpha}}$ may be obtained non-intrusively by least squares regression. A set of *N* (at least *P*) training data (realizations of $\boldsymbol{X}$) is selected, and the original model is solved for them, generating the system response vector $\boldsymbol{\Upsilon}$. The coefficients are then given by:

$$\boldsymbol{a} = (\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^{\mathrm{T}}\,\boldsymbol{\Upsilon} \tag{9}$$

where $\boldsymbol{a}$ is a vector that gathers the coefficients $a_{\boldsymbol{\alpha}}$ and $\boldsymbol{\Psi}$ is the data matrix defined by:

$$\boldsymbol{\Psi}_{ij} = \Psi_{\alpha j}(\mathbf{x_i})/\mathrm{i} = 1,\dots,\mathrm{N};\ \mathrm{j} = 1,\dots,\mathrm{p}\text{-}1 \tag{10}$$

With the basis defined and the coefficients determined, the expansion can be used as an approximation for the original model (e.g. a limit state equation).

2.2. Kriging

The Kriging technique consists in predicting data using interpolation models, which are assembled considering a spatial correlation between the values of the function to be approximated. Kriging has first appeared in the field of geology in the sixties, but only recently, since the works of *Romero et al. (2004)* and *Kaymaz (2005)*, the method started to be employed in structural reliability. A historical review of the method is presented by *Cressie (1990).*

Kriging models can be seen as the realization of a stochastic field, and thus the function to be surrogated may be written as follows:

$$g(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{\beta} + z(\boldsymbol{x}) \tag{11}$$

where the first term defines a deterministic response surface that works as the mean of the process. $\boldsymbol{f}(\boldsymbol{x})^T$ is a set of basis functions (e.g. polynomials), and $\boldsymbol{\beta}$ is a vector that gathers together the coefficients of the surface:

$$\boldsymbol{\beta} = (\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{F}^{\mathbf{T}})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\boldsymbol{\Upsilon} \tag{12}$$

Matrix $\boldsymbol{R}$ is the correlation matrix, given by equation (15) for each pair of training data, and matrix **F** is the regression matrix. The second term of equation (11) is a homogeneous Gaussian random field with zero mean and constant variance σ_Z^2:

$$\sigma_Z^2 = \frac{1}{\mathrm{m}}(\boldsymbol{\Upsilon} - \mathbf{F}\boldsymbol{\beta})^{\mathrm{T}}\mathbf{R}^{-1}(\boldsymbol{\Upsilon} - \mathbf{F}\boldsymbol{\beta}) \tag{13}$$

It's autocorrelation function is given by:

$$Cov[z(\boldsymbol{x}), z(\boldsymbol{x}')] = \sigma_Z^2 R(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\theta}) \quad (14)$$

with

$$R(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\theta}) = \prod_{i=1}^{m} e^{(-\theta_i d_i^2)} \quad (15)$$

where $d_i = (x_i - x_i')$ is the distance between two studied points in the i_{th} direction. Other autocorrelation functions may also be applied, but the anisotropic Gaussian function is the most common one (*Gaspar et al., 2014*).

Vector $\boldsymbol{\theta}$ gathers the parameters that define the inverse of the correlation length. Since the correlation matrix $\boldsymbol{R}$, hence $\boldsymbol{\beta}$ and σ_Z^2, depend on $\boldsymbol{\theta}$, the later vector has to be first calculated. This is done by minimizing the function $\mathcal{L}(\boldsymbol{\theta})$, using the method of maximum likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = |\boldsymbol{R}(\boldsymbol{\theta})|^{\frac{1}{m}} \sigma_Z^2(\boldsymbol{\theta}) \quad (16)$$

Once $\boldsymbol{\theta}$ is defined, all the kriging parameters can be obtained, and a limit state function can be finally predicted by equation (11), with:

$$z(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})^{\mathrm{T}} \mathbf{R}^{-1} (\boldsymbol{\Upsilon} - \mathbf{F}\boldsymbol{\beta}) \quad (17)$$

and

$$\boldsymbol{r}_i(\boldsymbol{x}')^{\mathrm{T}} = [R(\boldsymbol{x} - \boldsymbol{x}_i', \boldsymbol{\theta})] \quad (18)$$

3. APPLICATION EXAMPLES

3.1. Some details on the employed Meta-Models

For every application, a Multi-Layer Perceptron ANN is adopted with only one hidden layer (as shown to be enough by *Hornik et al., 1990*). The first layer has one neuron for each input random variable and the last layer has one neuron for each response of the model. The number of neurons of the hidden layer is twice the number of input parameters plus one, as stated by the Kolmogorov–Nielsen theorem. A tangent-sigmoid transfer function was adopted for the neurons on the hidden layer, and linear functions on the other ones. When the number of training points is stated on the examples, it already considers the 20% of the information that was used as validation data. The Levenberg–Marquardt method was used to train the networks, with 2000 iterations. More details about this training method can be found in *Gomes and Beck (2013)*.

Two full expansion PCEs are used in the analytical examples, with the max (multi-index) polynomial degree of 4 and 7. In section 4.3, a full PCE expansion of degree 3 is considered, same order used by *Sudret et al. (2011)* in the same example. The PCE coefficients were obtained with the regression approach discussed in section 2.2.

In Kriging models, the correlation is defined by an isotropic Gaussian correlation function, and the design of experiment is centered on the random variable's means. For the response surface, up to second order polynomials with no cross-terms were considered as basis functions.

In all cases, the training points were defined by a random Latin Hypercube Sampling technique. Once the surrogate models are established, crude Monte Carlo simulation with 10^6 samples is used on the original model as a reference, and then for each analytical technique. Only 10^4 samples were used in the numerical example, since the accuracy of the meta-models, and not the exact solution, is studied.

In the present work, as a mapping strategy, the input random variables were divided by their mean before feeding the models. In the few cases where the mean was zero, input random variables were divided by their standard deviation. This has shown to be enough for overcoming scaling problems.

3.2. Analytical Limit State Examples

Four limit state equations that can be analytically represented are considered in this section. They are solved for 4 different amounts of training data: the minimum required for the full 2nd, 4th and 7th degree PCE, as given by equation 8 (i.e. the minimum required for all the techniques to work), and twice the 7th degree PCE required number. Eventually, a few training points were added so that data matrix became well conditioned. The random variables used in this

section are listed in Table 1. Table 2 shows the limit state equations addressed in this section.

Table 1: Random Variables

Variable	*Distribution*
X_1, X_2, X_3	*Standard Normal*
X_4, X_5, X_6	*Uniform*~$[-\pi, \pi]$

Table 2: Studied Limit State Equations.

Eq. Type	$g(\boldsymbol{x})$	*Eq. #*
Linear	$2X_1 - X_2 + 5$	*(19)*
Polynomial	$4X_1 + 2X_2^2 - X_1X_3^3 + 10$	*(20)*
Exponential	$e^{(0.2X_1+1.4)} - X_2$	*(21)*
Ishigami	$\sin X_4 + 7\sin^2 X_5 + 0.1X_6^4 \sin X_4 + 8$	*(22)*

The reliability index β obtained with each technique is shown in Tables 3 to 6 for its respective number of training points. A positive constant was added to equation (22) to maintain its complexity but make it more suitable for the limit state equation behavior of a hypothetical structural reliability problem.

The used meta-models appear to satisfactorily represent all the studied analytical equations, with improved accuracy as the amount of training data increases. Some difficulties were found when dealing with the ANN, for slightly changing the number of training points implied in a considerable gain or loss of accuracy on the reliability index. Also, for PCE, the data matrix often became ill-posed, the selection of a different design of experiment being required for an accurate solution. Still, in both cases the errors appear to be local effects, the accuracy still increasing with the amount of training data, taken away the punctual errors.

Table 3: β for for PCE_2 minimum amount of data

Eq.	*Ref.*	*NNA*	PCE_2	Kriging
19	*2,23*	*2,21*	*2,25*	*2,25*
20	*2,11*	*2,83*	*1,94*	*3,19*
21	*3,36*	*3,29*	*3,29*	*3,29*
22	*1,00*	*0,57*	*0,77*	*0,34*

Table 4: β for PCE_4 minimum amount of data.

Eq.	*Ref.*	*NNA*	PCE_4	Kriging
19	*2,23*	*2,25*	*2,23*	*2,23*
20	*2,11*	*2,54*	*2,11*	*2,18*
21	*3,36*	*3,35*	*3,36*	*3,36*
22	*1,00*	*0,80*	*0,71*	*0,63*

Table 5: β for PCE_7 minimum amount of data.

Eq.	*Ref.*	*NNA*	PCE_7	Kriging
19	*2,23*	*2,23*	*2,23*	*2,23*
20	*2,11*	*2,07*	*2,11*	*2,11*
21	*3,36*	*3,38*	*3,36*	*3,36*
22	*1,00*	*1,00*	*0,88*	*1,00*

Table 6: β for more training data

Eq.	*Ref.*	*NNA*	PCE_7	Kriging
19	*2,23*	*2,23*	*2,23*	*2,23*
20	*2,11*	*2,11*	*2,11*	*2,11*
21	*3,36*	*3,36*	*3,36*	*3,36*
22	*1,00*	*1,00*	*0,99*	*0,99*

3.3. Finite Element Truss

A truss structure first studied by *Blatman et al. (2007)* is considered, as represented by figure 2.

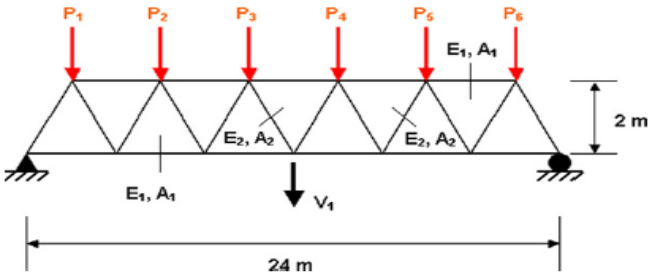

Figure 2: Studied truss structure.

It is composed by 23 bars and 13 nodes, and it is subjected to 6 vertical random loads on the upper nodes. The input random parameters are described in Table 7. The limit state equation is defined implicitly as the result of a finite element model, taking into consideration the vertical displacement of the mid-span node, as represented in Figure 2.

Table 7: Truss random parameters.

Variable	*Distribution*	*Mean*	*St. Dev.*
$E_1, E_2 (Pa)$	*Lognormal*	2.1×10^{11}	2.1×10^{10}
$A_1 (m^2)$	*Lognormal*	2×10^{-3}	2×10^{-4}
$A_2 (m^2)$	*Lognormal*	1×10^{-3}	1×10^{-4}
$P_1 - P_6 (N)$	*Gumbel*	5×10^{4}	7.5×10^{3}

The serviceability of the truss relates to a threshold of 10cm defined as limit displacement, thus the limit state equation reads:

$$g(\boldsymbol{x}) = 10 - |V_1(\boldsymbol{x})| \qquad (23)$$

The reliability indexes obtained with each technique for various quantities of training data are plotted in Figure 3.

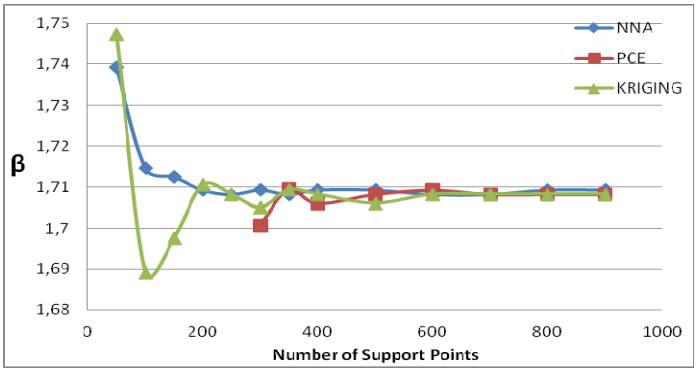

Figure 3: β for different quantities of training data.

All the techniques converge to the reference value of $\beta = 1{,}71$ as the amount of quantities of training data increase. The full PCE expansion shows good convergence, but it requires use of at least 286 training points, while ANN and Kriging give reasonably accurate results with as few as 50 training points. The fastest convergence was presented by the ANNs, even though the training process may be time demanding in comparison to the definition of the kriging meta-model.

4. CONCLUSIONS

Three meta-modeling techniques were employed in the solution of five different problems. Their accuracy has been compared, and in all cases found to be satisfactory. Their convergence with increasing number of training data has been verified in all the studied examples. A suggestion for the often required data mapping was discussed in detail. Some difficulties which may be found when dealing with each technique were pointed out, and a few more are briefly addressed here:

- In the regression performed by PCE and Kriging, the data matrix may, in some cases, not be invertible. Regularization of the matrix by adding small quantities to its main diagonal, using a different design of experiment and replacing the inverse matrix by the corresponding pseudo-inverse are ways of over-coming this problem.

- For mechanical models with very large number of random input parameters, use of PCE may become impracticable, since the minimum number of times the original model must be called is too large. The sparse PCE approach presented in *Sudret and Blatman (2010)* mitigates this difficulty.

- The ANNs showed great accuracy for all studied examples, but was by far the most time consuming technique. Also, the division between training and validation data may change this models accuracy, even for the same design of experiment

Further work is in progress towards addressing more challenging structural reliability problems and employing the studied techniques in solution of risk optimization problems.

ACKNOWLEDGMENTS

Financial support by Brazilian Agencies CAPES and FAPESP is greatly acknowledged.

REFERENCES

Berke L, Hajela P. (1992). “Applications of artificial neural nets in structural mechanics” *Structural Optimization*, Springer-Verlag” 4, 90-98.

Bento JP (1998). “Modeling mechanical behavior without mechanics”, *Development of Knowledge-Based Systems for Engineering*, Springer-Verlag, 37-58.

Blatman G, Sudret B, Berveiller M. (2007). “Quasi-random numbers in stochastic finite element analysis”. *Mécanique & Industries* 8:289_97.

Cardoso, J.B., Almeida, J.R., Dias, J.M., Coelho, P.G.(2008), "Structural reliability analysis using Monte Carlo simulation and neural networks", *Advances in Engineering Software*, Vol. 39/6, 505-513,.

Cressie, N. (1990), "The Origins of Kriging", *Mathematical Geology*, Vol 22, Issue 3, 239-252.

Dietlevsen O., Madsen H. O. (2007). "Structural Reliability Methods". Technical University of Denmark.

Elhewy. A. H., Mesbahi, E. Pu, Y. (2006). "Reliability analisys of structures using neural network method". *Probabilistic Engineering Mechanics*, 21, 44-53

Gaspar, B., Teixeira, A. P., Guedes Soares, C. (2014) "Assessment of the efficiency of Kriging surrogate models for structural reliability analysis". *Probabilistic Engineering Mechanics*, 37, 24-34

Gomes. W., Beck, A. T. (2013) "Global structural optimization considering expected consequences of failure and using ANN surrogates" *Computers and Structures*, 126, 56-68

Hornik K, Stinchcombe M, White H. (1990) "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks." *Neural Networks* 3,551–560.

Kaymaz, I. (2005), "Application of kriging method to structural reliability problems". *Structural Safety*, vol 27, 133-151

Panayirci, H. M., Schuëller, G. I. (2011) "On the Capabilities of the Polynomial Chaos Expansion Method within SFE Analysis - An Overview", *Archives of Computational Methods in Engineering*, Barcelona.

Papadrakakis M, Papadopoulos V, Lagaros ND (1996). "Structural reliability of elastoplastic structures using neural networks and Monte Carlo simulation", *Computer Methods in Applied Mechanics and Engineering*, 136, 145-163.

Romero. V. J., Swiler. L. P., Giunta. A. A. (1994) "Construction of response surfaces based on progressive-lattice-sampling experimental designs with application to uncertainty propagation", "*Structural Safety"*, Vol 26, Issue 2, 201-219.

Sudret, B, Berveiller M, Lemaire M (2006), "A stochastic finite element procedure for moment and reliability analisys", *European Journal of Computational Mechanics*. vol 15/7-8 - 2006 - pp.825-866

Sudret, B, Blatman, G. (2010): "An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis", *Probabilistic Engineering Mechanics*, 25, 183-197

Sudret, B, Blatman, G. Berveiller, M. (2011): "Response surfaces based on polynomial chaos expansions", *Construction Reliability*, edited by Baroth. J., Schoefs, F., Breyssep. D., p. 147-167

Sudret, B, (2012): "Meta-models for structural reliability and uncertainty quantifications", *5th Asian-Pacific Symposium on Structural Reliability and its Applications*, 53-76.