ETH zürich

Combining Polynomial Chaos Expansions and Kriging

R. Schöbi, P. Kersaudy, B. Sudret and J. Wiart

Chair of Risk, Safety and Uncertainty Quantification
Stefano-Franscini-Platz 5
CH-8093 Zürich

Risk, Safety and Uncertainty Quantification

Report Data Sheet

Report Ref.	RSUQ-2014-001
Title:	Combining Polynomial Chaos Expansions and Kriging
Authors:	R. Schöbi[1], P. Kersaudy[2,3], B. Sudret[1] and J. Wiart[2]
	[1] ETH Zürich, Switzerland
	[2] Orange Labs, Issy-les-Moulineaux, France
	[3] Université Paris-Est Marne-la-Vallée, France
Date:	April 10th, 2014
Acknowledgements:	The present research was supported by the ETH / Orange Labs research agreement n° C12063.

Abstract

Computer simulation has emerged as a key tool for designing and assessing engineering systems in the last two decades. Uncertainty quantification has become popular more recently as a way to model all the uncertainties affecting the system and their impact onto its performance.

In this respect meta-models (a.k.a. surrogate models) have gained interest. Indeed dealing with uncertainties requires running the computer model many times, which may not be affordable for complex models. Surrogate models mimic the behaviour of the original model while being cheap to evaluate.

Polynomial chaos expansion (PCE) and Kriging are two popular techniques, which have been developed with very little interaction so far. In this report we present a new approach, called *PC-Kriging*, that combines the two tools. The algorithm is based on the universal Kriging model where the trend is represented by a set or orthonormal polynomials.

Various aspects of the new metamodelling technique are presented and investigated in details. The discussion starts with a survey on methods for generating an optimal design of experiments (DOE). The PC-Kriging algorithm inherits many parameters and sub-methods such as the number of polynomial terms and the choice of the autocorrelation kernel. A variety of kernels are presented and discussed.

The methods are compared on analytical benchmark functions. The conclusion of this report is that PC-Kriging performs better or at least as well as PCE or Kriging taken separately in terms of relative generalized error (L_2-error).

Keywords: metamodelling, polynomial chaos expansions, Kriging, PC-Kriging, experimental design

Contents

1 Introduction

Executing physical experiments is time- and money consuming. Thus the number of experiments is limited by the available resources. A similar phenomenon appears for computational experiments. Up-to-date computer models of physical behaviours require large computational power and execution time. In a practical case, the experiments are used to understand the unknown behaviour of a system represented by a computational model and finally to assess some kind of performance. Examples for problems in which computational models are used are optimization (*i.e.* searching for minimum of a cost function) Rackwitz (2001); Rasmussen and Williams (2006); Dubourg et al. (2011) and reliability analysis (computing a probability that the model output exceeds a prescribed threshold) (Kaymaz, 2005; Bect et al., 2012; Echard et al., 2011). Solving these problems requires a large number of model evaluations (computer experiments) to assess the performance with reasonable accuracy. The resources are often below the amount necessary to reach an acceptable accuracy. Thus the idea of *metamodels*, also known as *surrogate models*, has emerged in the past decade.

Metamodelling decreases the computational effort of time-consuming computational simulations by approximating the underlying computational model with a simple and easy-to-evaluate function. A small and carefully chosen design of experiments (DOE) contains the support points of the metamodel. Under the assumption of smoothness of the underlying model, a response surface is built up on the basis of the values of the original model at the support points. The required data is the set of input/output values. Such approaches are called *non-intrusive* since there is no need to know the machinery of the model, *i.e.* it is considered as a black-box). The metamodel (also called response surface) simplifies the prediction of the output value for samples which have not been considered in the experimental design. The metamodel can then be used to predict system responses at an affordable effort.

This report is based on two distinct non-intrusive metamodelling approaches. Polynomial Chaos Expansions (PCE) approximates the underlying model by a set of orthonormal polynomials in the input variables (Ghanem and Spanos, 2003). A sparse set of polynomials may be determined by a selection algorithm (*e.g.* least-angle regression) and the coefficients are obtained by least-square minimization (Blatman and Sudret, 2011).

A second metamodelling technique is Kriging. It originates from interpolating geographical data in mining (Krige, 1951) and is today also known as Gaussian process regression (Santner et al., 2003; Rasmussen and Williams, 2006).

Based on these two approaches a new metamodelling technique is presented in this report as a combination of both. The new approach is called *Polynomial-Chaos-Kriging* (PC-Kriging) which combines the advantages of both approaches: the approximation of the global behaviour of PCE and the local interpolating behaviour of Kriging, as described below. As shown in this report, this hybrid modelling leads to an accuracy that is better than or at least as good as either of the simple methods.

This report is organized as follows. The two metamodelling approaches PCE and Kriging are presented in Sections 2 and 3 respectively. Then the new metamodelling approach called PC-Kriging is introduced in Section 4. In Section 5 different sampling methods for the design of experiments are presented. Then the performance of the three approaches are compared on analytical benchmark functions in Section 6. The report ends with conclusions and an outlook to further research in Section 7.

2 Polynomial Chaos Expansion

2.1 Problem definition

Consider a system whose behaviour is represented by a computational model $\mathcal{M} : \boldsymbol{x} \in \mathcal{D}_X \subset \mathbb{R}^M \mapsto \mathbb{R}$. Due to uncertainties in the input parameters, the latter one is represented by a random vector $\boldsymbol{X}$ whose components are assumed independent in this report. The output is represented by the random variable Y such that:

$$Y = \mathcal{M}(\boldsymbol{X}) \tag{1}$$

This random variable (or more generally, random vector in case of a vector-valued model) is obtained by propagating the uncertainty in $\boldsymbol{X}$ through the computational model $\mathcal{M}$. The model $\mathcal{M}$ is a deterministic function which means that repeated evaluations with the same realization of the input vector lead to the same output value. In this report, the model $\mathcal{M}$ is assumed to be a black-box-type model, meaning that information is available only about the input vector and output value: no knowledge is required on the inner structure of the model. Assuming the output Y has a finite variance, the polynomial chaos expansion of Y (PCE) represents the computational model by sums of orthonormal polynomials in $\boldsymbol{X}$:

$$Y \equiv \mathcal{M}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} \boldsymbol{a}_{\boldsymbol{\alpha}} \, \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \tag{2}$$

where $\{\boldsymbol{a}_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathbb{N}^M\}$ are coefficients to be determined, M is the number of input variables in $\boldsymbol{X}$, $\psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$ are multivariate orthonormal polynomials, $\boldsymbol{\alpha}$ is the multi-index $\{\alpha_1, \ldots, \alpha_M\}$. Polynomial $\psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$ is built as a product of univariate polynomials:

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) = \prod_{i=1}^{M} \psi_{\alpha_i}^{(i)}(X_i) \tag{3}$$

where $\psi_{\alpha_i}^{(i)}$ is the polynomial in the i-th variable of degree α_i. The input random variables $\boldsymbol{X} = \{X_i, i = 1, \ldots, M\}$ (supposed independent here for the clarity of the derivations) may be modelled by different probability density functions (PDF). For each type of input PDF there exists a set of orthogonal polynomials which can be transformed into an orthonormal polynomial basis. To define the orthonormality of a polynomial basis, first the functional inner product of two functions $\phi_1(x), \phi_2(x)$ with respect to a prescribed

probability density function f_{X_i} is defined:

$$\langle \phi_1, \phi_2 \rangle_i = \int_{\mathcal{D}_i} \phi_1(x)\, \phi_2(x)\, f_{X_i}(x) dx \tag{4}$$

In PCE, orthonormal polynomial bases have the property that the functional inner product is equal to 1 for identical polynomials $P_j^{(i)} = P_k^{(i)}$ and is equal to zero for different polynomials $P_j^{(i)} \neq P_k^{(i)}$, for each variable $i = 1, \ldots, M$:

$$\langle P_j^{(i)}, P_k^{(i)} \rangle = \int_{\mathcal{D}_i} P_j^{(i)}(x)\, P_k^{(i)}(x)\, f_{X_i}(x)\, dx = \delta_{jk} \tag{5}$$

where $\mathcal{D}_i$ is the support of the random variable X_i and δ_{jk} is the Kronecker delta which is equal to 1 for $j = k$ and equal to 0 otherwise. Some of the classical polynomial basis are shown in Table 1 (Sudret, 2012).

Table 1: Classical Orthogonal Polynomials

Distr.	PDF	Orthogonal polynomials	Orthonormal basis
Uniform	$\mathbf{1}_{]-1,1[}(x)/2$	Legendre $P_k(x)$	$P_k(x)/\sqrt{\frac{1}{2k+1}}$
Gaussian	$\frac{1}{\sqrt{2\pi}} e^{-x^2/2}$	Hermite $H_{e_k}(x)$	$H_{e_k}(x)/\sqrt{k!}$
Gamma	$x^a e^{-x} \mathbf{1}_{\mathbb{R}^+}(x)$	Laguerre $L_k^a(x)$	$L_k^a(x)/\sqrt{\frac{\Gamma(k+a+1)}{k!}}$
Beta	$\mathbf{1}_{]-1,1[}(x) \frac{(1-x)^a (1+x)^b}{B(a)B(b)}$	Jacobi $J_k^{a,b}(x)$	$J_k^{a,b}(x)/\mathcal{J}_{a,b,k}$
		$\mathcal{J}^2_{a,b,k} = \frac{2^{a+b+1}}{2k+a+b+1} \frac{\Gamma(k+a+1)\Gamma(k+b+1)}{\Gamma(k+a+b+1)\Gamma(k+1)}$	

Eq. (2) shows that any second order vector $Y = \mathcal{M}(\boldsymbol{X})$ can be cast as an infinite series. For practical purposes though, truncated series have to be handled. The goal is to find a sparse set of multi-indexes $\boldsymbol{\alpha} \in \mathcal{A} \subset \mathbb{N}^M$ so that the model output is approximated with sufficient accuracy.

$$Y \approx \mathcal{M}^{(\mathsf{PCE})}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \boldsymbol{a}_{\boldsymbol{\alpha}}\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \tag{6}$$

where $\mathcal{A}$ is a *finite* set of multi-indices. There are several ways to select the number of polynomials included in $\mathcal{A}$. A simple truncation scheme that is commonly used consists of limiting the *total degree* of polynomials with an upper bound. The total degree of a polynomial is defined as (Blatman, 2009)

$$|\boldsymbol{\alpha}| = \sum_{i=1}^{M} \alpha_i \tag{7}$$

The corresponding set of multi-indices is defined as $\mathcal{A}^{M,p} = \{\boldsymbol{\alpha} \in \mathbb{N}^M : |\boldsymbol{\alpha}| \leq p\}$ where p is the maximal total polynomial degree. When the number of input variables is M, the cardinal of this set $\mathcal{A}$ reads:

$$|\mathcal{A}_{M,p}| = \frac{(M+p)!}{M!\, p!} \tag{8}$$

which grows polynomially both in M and p. Such a truncation scheme would lead to non tractable problems if the model response is highly nonlinear in its input parameters (need for large p) and/or the size of the input vector $\boldsymbol{X}$ is large (say, $M > 10$). This problem is known as *the curse of dimensionality*.

2.2 Determination of the coefficients

Having defined the polynomial set, the next step is to determine the expansion coefficients. Ghiocel and Ghanem (2002), Le Maître et al. (2002), Keese and Matthies (2005) use projection methods; Xiu and Hesthaven (2005), Xiu (2009) use stochastic collocation methods; and Berveiller et al. (2006), Blatman (2009), Blatman and Sudret (2010, 2011) use least-square-minimization methods. The least-square-minimization method now presented in details: the expansion coefficients $\boldsymbol{a} = \{\boldsymbol{a}_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{A}\}$ are calculated by minimizing the least square residual

$$\boldsymbol{a} = \underset{\boldsymbol{a}\in\mathbb{R}^{|\mathcal{A}|}}{\arg\min}\, \mathbb{E}\left[\left(Y - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \boldsymbol{a}_{\boldsymbol{\alpha}}\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{X})\right)^2\right] \tag{9}$$

Given an experimental design $\boldsymbol{\mathcal{X}} = \{\boldsymbol{\chi}^{(1)}, \ldots, \boldsymbol{\chi}^{(N)}\}$ of N samples $\boldsymbol{\chi}^{(i)} \in \mathbb{R}^M$ and the associated model output $\mathcal{Y} = \{\mathcal{M}(\boldsymbol{\chi}^{(1)}), \ldots, \mathcal{M}(\boldsymbol{\chi}^{(N)})\} \equiv \{\mathcal{Y}^{(1)}, \ldots, \mathcal{Y}^{(N)}\}$, Eq. (9) is transformed into a discretized mean-square error problem:

$$\hat{\boldsymbol{a}} = \underset{\boldsymbol{a}\in\mathbb{R}^{|\mathcal{A}|}}{\arg\min}\, \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{Y}^{(i)} - \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \boldsymbol{a}_{\boldsymbol{\alpha}}\, \psi_{\boldsymbol{\alpha}}(\boldsymbol{\chi}^{(i)})\right)^2 \tag{10}$$

Blatman (2009) shows that the optimal expansion coefficients $\hat{\boldsymbol{a}}$ are determined as:

$$\hat{\boldsymbol{a}} = (\boldsymbol{F}^{\mathsf{T}}\, \boldsymbol{F})^{-1}\, \boldsymbol{F}^{\mathsf{T}}\, \mathcal{Y} \tag{11}$$

where $\boldsymbol{F}$ is the information matrix of size $N \times |\mathcal{A}|$ whose generic term reads:

$$F_{ij} = \psi_j(\boldsymbol{\chi}^{(i)}) \tag{12}$$

In practice the set of polynomials obtained from the common truncation scheme (see Eq. (8)) includes many elements with negligible influence on the resulting model. Blatman and Sudret (2008) proposed to use *hyperbolic index sets* to a priori neglect the polynomials associated with a high degree of interaction. They observed that many real systems tend to have low degree interaction predictors. Thus it is not necessary to compute all interaction terms of higher total polynomial degree. The derived *hyperbolic index set* is based on q-norms:

$$\mathcal{A}_q^{M,p} \equiv \{\boldsymbol{\alpha} \in \mathbb{N}^M : \|\boldsymbol{\alpha}\|_q \leq p\} \tag{13}$$

where

$$\|\boldsymbol{\alpha}\|_q \equiv \left(\sum_{i=1}^{M} \alpha_i^q \right)^{\frac{1}{q}} \tag{14}$$

where $0 < q < 1$. Decreasing q leads to a smaller number of interactive predictors. In the limit $q \to 0$, only univariate polynomials would be retained in the expansion, what is called an additive model.

The system behaviour usually depends only on a limited number of predictors in case of a smooth function. Thus a further reduction in high-dimensional space of the set of predictors is possible and suitable to better understand the system. Several approaches are known for regression with a large set of predictors:

- *Stepwise regression* which is known in statistics to be greedy and unstable (Hesterberg et al., 2008);
- *All-subsets regression* is a variation of the stepwise regression and is computationally demanding as it considers a large amount of possible metamodels (Furnival and Wilson, 1974);
- *LASSO* (Least Absolute Shrinkage and Selection Operator) (Tibshirani, 1996);
- *Forward Stagewise Regression* (Hastie et al., 2001) picks the predictor which is most correlated with the output vector in each iteration;
- *Least Angle Regression* (LAR) (Efron et al., 2004) can be viewed as a modified Forward Stagewise Regression;
- *Dantzig selector* (Candes and Tao, 2007) transforms the setup into a constrained optimization problem which can be solved by linear programming.

In this report, the LAR algorithm is implemented using hyperbolic index sets, following the developments in Blatman (2009); Blatman and Sudret (2011).

2.3 Error estimation

The PCE predictions differ from the exact system response due to the truncation of the expansion. Thus error measures are developed to quantify the deviation between model output $\mathcal{Y}^{(\mathsf{PCE})} = \mathcal{M}^{(\mathsf{PCE})}(\boldsymbol{\mathcal{X}})$ and the exact output $\mathcal{Y} = \{\mathcal{Y}^{(i)}, i = 1, \ldots, N\}$. Computing the expectation of the squared output residuals leads to the *generalization error* (also called L^2-error):

$$Err_{L^2} = \mathbb{E}\left[\left(Y - Y^{(\mathsf{PCE})} \right)^2 \right] \tag{15}$$

The numerical formulation (based on a discrete number of samples n) of the latter error estimate is:

$$\widehat{Err}_{L^2} \equiv \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \mathcal{M}^{(\mathsf{PCE})}(\boldsymbol{x}_i) \right)^2 \tag{16}$$

where n is the number of elements in the validation set and its input vectors $\boldsymbol{x}_i$ and output values y_i. The validation set is a large collection (n large) of input/ouput samples which are different from the support points/experimental design of the metamodel.

This error is only affordable for analytical functions. In practice the latter error is not accessible and may be estimated by the *empirical error* computed from the experimental design:

$$Err_{emp} \equiv \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{Y}^{(i)} - \mathcal{M}^{(\mathsf{PCE})}(\boldsymbol{\mathcal{X}}^{(i)})\right)^2 \tag{17}$$

where $\boldsymbol{\mathcal{X}} = \{\boldsymbol{\mathcal{X}}^{(i)}, i = 1,\ldots,N\}$ is the set of input realizations and $\mathcal{Y} = \{\mathcal{Y}^{(i)}, i = 1,\ldots,N\}$ are the corresponding output values of the computational model. The *relative empirical error* is a dimensionless version of the empirical error where the empirical error is normalized by the output variance:

$$\epsilon_{emp} \equiv \frac{\sum_{i=1}^{N}\left(\mathcal{Y}^{(i)} - \mathcal{M}^{(\mathsf{PCE})}(\boldsymbol{\mathcal{X}}^{(i)})\right)^2}{\sum_{i=1}^{N}\left(\mathcal{Y}^{(i)} - \mu_{\mathcal{Y}}\right)^2} \tag{18}$$

where $\mu_{\mathcal{Y}}$ is the mean value of the output values in $\mathcal{Y}$.

The empirical error generally underpredicts the generalization error because of the so-called *overfitting*. When increasing the number of polynomials in the PCE metamodel $\mathcal{M}^{(\mathsf{PCE})}$ in Eq. (17) the empirical error ϵ_{emp} decreases. In the extreme case of $N = |\mathcal{A}|$ the predictors may interpolate the output values leading to $\epsilon_{emp} = 0$. Thus a better estimate for the global error is found in the leave-one-out (LOO) error (Stone, 1974; Geisser, 1975). The general formulation of the leave-one-out error is

$$Err_{LOO} \equiv \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{Y}^{(i)} - \mathcal{M}^{(\mathsf{PCE})}_{(-i)}(\boldsymbol{\mathcal{X}}^{(i)})\right)^2 \tag{19}$$

where $\mathcal{M}^{(\mathsf{PCE})}_{(-i)}(\cdot)$ is a PCE model built from the experimental design $\boldsymbol{\mathcal{X}}^{(-i)} = \boldsymbol{\mathcal{X}} \backslash \boldsymbol{\mathcal{X}}^{(i)} \equiv \{\boldsymbol{\mathcal{X}}^{(j)}, j = 1,\ldots,i-1,i+1,\ldots,N\}$, and $\mathcal{Y} = \{\mathcal{Y}^{(i)}, i = 1,\ldots,N\}$ are the output values of the original computational model $\mathcal{M}$. The leave-one-out error is a special case of the ν-fold cross-validation error. In the leave-one-out error formulation each sample of the experimental design is predicted using a model based on all other samples of the experimental design what is denoted $\mathcal{M}^{(\mathsf{PCE})}_{(-i)}(\boldsymbol{\mathcal{X}}^{(i)})$. This may lead to a large computational effort as N surrogate models are to be determined.

In the special case of linearly parameterized regression, it is possible to calculate the leave-one-out error analytically without building explicitly the N models. The leave-one-out error is then (Saporta, 2006):

$$Err_{LOO} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathcal{Y}^{(i)} - \mathcal{M}^{(\mathsf{PCE})}(\boldsymbol{\mathcal{X}}^{(i)})}{1-h_i}\right)^2 \tag{20}$$

where h_i is the i^{th} diagonal term of the matrix $\boldsymbol{F}(\boldsymbol{F}^{\mathsf{T}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathsf{T}}$ and $\boldsymbol{F}$ defined in Eq. (12).

The potential overfitting is still seizable. Thus there are empirical methods to decrease the sensitivity to overfitting. One of the methods by Blatman (2009) takes into consideration the number of samples and the number of predictors to produce an *adjusted empirical error* of the form:

$$Err^*_{emp} = Err_{emp} \times T(P, N) \tag{21}$$

where N is the number of samples, P is the number of predictors, i.e. number of polynomials and $T(P, N)$ is the adjustment factor to the empirical error given by:

$$T(P, N) = \frac{N-1}{N-P-1} \tag{22}$$

The adjusted empirical error increases with an increasing number of predictors P and decreases with an increasing number of samples N. Another empirical rule is proposed by Chapelle et al. (2002) and applied by Blatman (2009) which is valid for regressions with a small experimental design. The correction factor is dependent on the number of predictors and number of samples as well as on the information matrix (Eq. (12)):

$$T(P, N) \equiv \frac{N}{N-P}\left(1 + \frac{\mathsf{tr}(C^{-1}_{emp})}{N}\right) \tag{23}$$

$$C_{emp} \equiv \frac{1}{N}\mathbf{F}^{\mathsf{T}}\,\mathbf{F} \tag{24}$$

The same procedure can be used to adjust the leave-one-out error to a corrected leave-one-out error.

$$Err^*_{LOO} = Err_{LOO} \times T(P, N) \tag{25}$$

3 Kriging

3.1 Problem definition

Kriging (a.k.a. Gaussian process modelling) is a stochastic interpolation algorithm which assumes that the model output $\mathcal{M}(\boldsymbol{x})$ is a realization of a Gaussian process indexed by $\boldsymbol{x} \in \mathcal{D}_X \subset \mathbb{R}^M$.

$$\mathcal{M}(\boldsymbol{x}) \approx \mathcal{M}^{(\mathsf{K})}(\boldsymbol{x}) = \boldsymbol{\beta}^{\mathsf{T}} \cdot \boldsymbol{f}(\boldsymbol{x}) + \sigma^2\, Z(\boldsymbol{x}, \omega) \tag{26}$$

where $\boldsymbol{\beta}^{\mathsf{T}} \cdot \boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{P} \beta_j\, f_j(\boldsymbol{x})$ is the mean value of the Gaussian process (a.k.a. *trend*) and $Z(\boldsymbol{x}, \omega)$ is a zero mean, unit variance Gaussian process. The set of hyper-parameters $\boldsymbol{\theta}$ describes the autocorrelation $R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$ between the two points $\boldsymbol{x}$ and $\boldsymbol{x}'$. In this notation, $\{f_j(\boldsymbol{x}),\, j = 1, \ldots, P\}$ is a set of predefined functions (*e.g.* polynomials), $\boldsymbol{\beta}$ is a set of parameters to the computed and $\boldsymbol{\theta}$ is the set of parameters describing the autocorrelation function. Examples of autocorrelation functions are given in Section 3.2. In the literature distinction is made between three different types of Kriging (Echard, 2012; Dubourg, 2011), namely:

- *Simple Kriging*: The simple Kriging assumes that the trend has a *known* constant

value.

$$\boldsymbol{\beta}^{\mathsf{T}} \cdot \boldsymbol{f}(\boldsymbol{x}) \equiv \beta_0$$

- *Ordinary Kriging*: The ordinary Kriging approach assumes that the trend is a constant but has a unknown value. ($P = 1$; $f_1(\boldsymbol{x}) = 1$; β_1 is unknown)

- *Universal Kriging*: The universal Kriging method is the most general and flexible formulation and assumes that the trend is a linear combination of prescribed functions, *e.g.* polynomials.

$$\boldsymbol{\beta}^{\mathsf{T}} \cdot \boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{P} \beta_j \, f_j(\boldsymbol{x})$$

  where P is the number of polynomials in the trend part of the Kriging algorithm.

From the above it is clear that simple Kriging and ordinary Kriging are special cases of universal Kriging.

3.2 Autocorrelation functions

The autocorrelation function describes the correlation between two points in the M-dimensional space. Various formulations of autocorrelation functions can be found in the literature. Some of the widely used autocorrelation functions in Kriging approaches are listed here (Echard, 2012; Dubourg, 2011):

- *Dirac (a.k.a. nugget) autocorrelation function*:

$$R(\boldsymbol{x}, \boldsymbol{x}') = \prod_{i=1}^{M} \delta(x_i - x_i'), \ \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{D}_X \tag{27}$$

  where δ is the Dirac function which is 1 for $x_i = x_i'$ and 0 otherwise;

- *Linear autocorrelation function*:

$$R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{l}) = \prod_{i=1}^{M} \max\left(0, 1 - \frac{|x_i - x_i'|}{l_i}\right) \tag{28}$$

  where $\boldsymbol{l} : \{l_i > 0, \, i = 1, \ldots, M\}$ are the scale parameters. The correlation is limited to a defined range described by the parameter $\boldsymbol{l}$. This corresponds to the assumption that there is no correlation beyond a distance l_i in each dimension;

- *Exponential autocorrelation function*:

$$R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{l}) = \exp\left(-\sum_{i=1}^{M} \frac{|x_i - x_i'|}{l_i}\right) \tag{29}$$

  where $\boldsymbol{l} : \{l_i > 0, \, i = 1, \ldots, M\}$ are the scale parameters;

- *Squared exponential autocorrelation function*:

$$R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{l}) = \exp\left(-\sum_{i=1}^{M}\left(\frac{x_i - x_i'}{l_i}\right)^2\right) \tag{30}$$

  where $\boldsymbol{l} : \{l_i > 0,\, i = 1, \ldots, M\}$ are the scale parameters. This autocorrelation function is also called *Gaussian autocorrelation function*;

- *Matérn autocorrelation function*:

$$R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{l}, \nu) = \prod_{i=1}^{M} \frac{1}{2^{\nu-1}\,\Gamma(\nu)} \left(\sqrt{2\nu}\,\frac{|x_i - x_i'|}{l_i}\right)^{\nu} \kappa_\nu\left(\sqrt{2\nu}\,\frac{|x_i - x_i'|}{l_i}\right) \tag{31}$$

  where $\boldsymbol{l} : \{l_i > 0,\, i = 1, \ldots, M\}$ are the scale parameters, $\nu \geq 1/2$ is the shape parameter, $\Gamma(\cdot)$ is the Euler Gamma function and $\kappa_\nu(\cdot)$ is the modified Bessel function of the second kind (also known as Bessel function of the third kind). In practice the values $\nu = 3/2$ and $\nu = 5/2$ are the most popular forms. The formulation then simplifies into (Roustant et al., 2012):

$$R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{l}, \nu = 3/2) = \prod_{i=1}^{M}\left(1 + \frac{\sqrt{3}\,|x_i - x_i'|}{l_i}\right)\exp\left(-\frac{\sqrt{3}\,|x_i - x_i'|}{l_i}\right) \tag{32}$$

$$R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{l}, \nu = 5/2) = \prod_{i=1}^{M}\left(1 + \frac{\sqrt{5}\,|x_i - x_i'|}{l_i} + \frac{5\,(x_i - x_i')^2}{3\,l_i^2}\right)\exp\left(-\frac{\sqrt{5}\,|x_i - x_i'|}{l_i}\right) \tag{33}$$

Graphs of the behaviour of the various autocorrelation functions can be found in Dubourg (2011). The listed autocorrelation functions belong to the family of the separative functions (*i.e.* products of univariate functions) which work well for low-dimensional problems. Recently, new *additive* autocorrelation functions were proposed (Ginsbourger et al., 2013; Durrande et al., 2012, 2013; Duvenaud et al., 2012) which seem to be more suitable for high-dimensional problems. Additive kernels inherit a summation in the autocorrelation function instead of a multiplication, i.e. for the linear autocorrelation function:

$$R^{(\text{additive})}(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{l}) = \frac{1}{M}\sum_{i=1}^{M}\max\left(0, 1 - \frac{|x_i - x_i'|}{l_i}\right) \tag{34}$$

By choosing adequate additive kernels, the behaviour of the Gaussian process is modelled more accurately (Duvenaud et al., 2012).

For the benchmark problems in Section 6, the Matérn autocorrelation function is implemented because the applications are low-dimensional. Additionally, the Matérn kernel is a generalization of the Gaussian and the exponential kernel. The influence of the function family is low, thus all benchmark applications in this report are computed with the same autocorrelation function family, namely the Matérn with $\nu = 5/2$.

3.3 Estimation of the Kriging model parameters

The solution of Eq. (26) with respect to $\{\boldsymbol{\beta}, \sigma^2\}$ may be obtained analytically conditionally to the correlation parameter $\boldsymbol{\theta}$:

$$\boldsymbol{\beta}(\boldsymbol{\theta}) = \left(\boldsymbol{F}^{\mathsf{T}} \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}\, \boldsymbol{R}^{-1} \mathcal{Y} \tag{35}$$

$$\sigma_y^2(\boldsymbol{\theta}) = \frac{1}{N} \left(\mathcal{Y} - \boldsymbol{F}\, \boldsymbol{\beta}\right)^{\mathsf{T}} \boldsymbol{R}^{-1} \left(\mathcal{Y} - \boldsymbol{F}\, \boldsymbol{\beta}\right) \tag{36}$$

where $\boldsymbol{R}_{ij} = R(\boldsymbol{\chi}^{(i)}, \boldsymbol{\chi}(j); \boldsymbol{\theta})$ is the correlation matrix of the points in the experimental design, $\boldsymbol{F}_{ij} = f_j(\boldsymbol{\chi}^{(i)})$ is the Vandermonde matrix and $\mathcal{Y} = \{\mathcal{Y}^{(i)},\, i = 1, \ldots, N\}$ are the model responses of the computational model on the experimental design $\boldsymbol{\mathcal{X}} = \{\boldsymbol{\chi}^{(i)},\, i = 1, \ldots, N\}$. The prediction of the Kriging model at a new sample $\boldsymbol{x}$, denoted by $\hat{Y}(\boldsymbol{x})$, is a Gaussian variable defined by a mean value $\mu_{\hat{Y}}$ and a variance $\sigma^2_{\hat{Y}}$ (Sudret, 2012):

$$\mu_{\hat{Y}}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{\beta} + \boldsymbol{r}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{R}^{-1} \left(\mathcal{Y} - \boldsymbol{F}\, \boldsymbol{\beta}\right) \tag{37}$$

$$\sigma^2_{\hat{Y}}(\boldsymbol{x}) = \sigma_y^2 \left(1 - \langle \boldsymbol{f}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{r}(\boldsymbol{x})^{\mathsf{T}} \rangle \begin{bmatrix} \boldsymbol{0} & \boldsymbol{F}^{\mathsf{T}} \\ \boldsymbol{F} & \boldsymbol{R} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}) \\ \boldsymbol{r}(\boldsymbol{x}) \end{bmatrix}\right) \tag{38}$$

where $r_i(\boldsymbol{x}) = R(\boldsymbol{x}, \boldsymbol{\chi}^{(i)}; \boldsymbol{\theta})$ is the correlation between the new sample $\boldsymbol{x}$ and the samples points in the experimental design.

An important generalized case is when the correlation parameter $\boldsymbol{\theta}$ are a priori unknown. The optimal correlation parameters can either be obtained by a maximum likelihood estimate (ML) (Marrel et al., 2008; Dubourg et al., 2011) or a leave-one-out cross-validation estimate (CV) (Bachoc, 2013). The corresponding objective function to be minimized for ML and CV respectively read:

$$f_{ML}(\boldsymbol{\theta}) = \frac{1}{N} \log\det(\boldsymbol{R}(\boldsymbol{\theta})) + \log\left(\mathcal{Y}^{\mathsf{T}}\, \boldsymbol{R}(\boldsymbol{\theta})^{-1}\, \mathcal{Y}\right) \tag{39}$$

$$f_{CV}(\boldsymbol{\theta}) = \mathcal{Y}^{\mathsf{T}} \boldsymbol{R}(\boldsymbol{\theta})^{-1} \mathrm{diag}\left(\boldsymbol{R}(\boldsymbol{\theta})^{-1}\right)^{-2} \boldsymbol{R}(\boldsymbol{\theta})^{-1}\, \mathcal{Y} \tag{40}$$

The derivations of equations (39) and (40) can be found in Bachoc (2013). The comparison of the two approaches is carried out in Bachoc (2013) and the summarized results are:

- ML is preferable to CV in well-specified cases, i.e. when the selected covariance function family is the true one, a feature which is not known in many real problems;
- CV is preferable in misspecified cases and not too regular design of experiments because of a smaller bias in the results;
- The estimation benefits from irregular sampling in general;
- The asymptotic variance of CV is higher than ML in all studied cases;

- CV is less robust for regular designs with miss-specification of the autocorrelation function.

The conclusion is that in the practical case where the correlation structure of the Gaussian process is unknown, the use of the cross-validation leads to more robust results than the maximum likelihood estimate.

3.4 Error estimation

The Kriging prediction is stochastic, i.e. the mean and variance of the prediction are provided at any prediction point $\boldsymbol{x}$. The prediction variance Eq. (38) is a local error measure. The local error estimate is useful when searching for regions of the input space with low prediction accuracy. By placing new samples in these regions the overall accuracy of the model can be increased in an iterative manner: this technique is often referred to as adaptive metamodeling algorithms (Bichon et al., 2011; Echard et al., 2011).

When comparing different models a global error measure is needed. The empirical error Eq. (18) cannot be used for a global error estimate because Kriging is an exact interpolation algorithm in the experimental design samples. An approach is the leave-one-out cross-validation which is described in Section 2.3 in the case of polynomial chaos expansions. Dubrule (1983) derived an analytical formulation of the leave-one-out error for universal Kriging:

$$Err_{LOO} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{Y}^{(i)} - \mu_{\hat{Y}_{(-i)}}(\boldsymbol{\mathcal{X}}^{(i)})\right)^2 \tag{41}$$

where $\mu_{\hat{Y}_{(-i)}}(\boldsymbol{\mathcal{X}}^{(i)})$ is the mean prediction at sample point $\boldsymbol{\mathcal{X}}^{(i)}$ on the modified experimental design $\boldsymbol{\mathcal{X}}^{(-i)} = \boldsymbol{\mathcal{X}} \backslash \boldsymbol{\mathcal{X}}^{(i)}$. $\mathcal{Y}^{(i)}$ are the system response of the computational model. The prediction mean and variance are given by:

$$\mu_{\hat{Y}_{(-i)}} = \sum_{j=1, j\neq i}^{N} \frac{B_{ij}}{B_{ii}}\,\mathcal{Y}^{(j)} = \sum_{j=1}^{N} \frac{B_{ij}}{B_{ii}}\,\mathcal{Y}^{(j)} - \mathcal{Y}^{(j)} \tag{42}$$

$$\sigma^2_{\hat{Y}_{(-i)}} = \frac{1}{B_{ii}} \tag{43}$$

where $\boldsymbol{B}$ is a square matrix of size $(N + P)$, N being the number of samples in the experimental design $\boldsymbol{\mathcal{X}}$, P the number of polynomials in the trend:

$$\boldsymbol{B} = \begin{bmatrix} \sigma_y^2 \boldsymbol{R} & \boldsymbol{F} \\ \boldsymbol{F}^{\mathsf{T}} & \boldsymbol{0} \end{bmatrix}^{-1} \tag{44}$$

where σ_y^2 is the Kriging variance of the Kriging model involving the set of all sample points $\boldsymbol{\mathcal{X}}$ (Eq. (36)). Note that matrix $\boldsymbol{B}$ includes the information of the points of the initial experimental design, namely N samples. The generalized formulation for k-fold cross-validation can also be found in the referred paper (Dubrule, 1983).

4 PC-Kriging

4.1 Idea

Polynomial Chaos Kriging (PC-Kriging) is a combination of the two metamodelling techniques PCE and Kriging. PCE handles the global behaviour of the model using a set of polynomials and Kriging interpolates local variations as a function of the nearby sample points. The combination (PC-Kriging) behaves as an approximation on two levels: on the global scale via PCE and on the local scale via Kriging.

Basis for the combination of both approaches is the universal Kriging which combines a trend (regression) part and a correlation part as described in Eq. (26). The PC Kriging metamodel $\mathcal{M}^{(\mathsf{PCK})}$ reads as follows:

$$\mathcal{M}(\boldsymbol{x}) \approx \mathcal{M}^{(\mathsf{PCK})} = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \, \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) + \sigma^2 \, Z(\boldsymbol{x}, \omega) \tag{45}$$

where $\boldsymbol{a}^{\mathsf{T}} \cdot \boldsymbol{\psi}_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \, \psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$ is the mean value of the Gaussian process (a.k.a. trend), $P = |\mathcal{A}|$ is the number of polynomials and $Z(\boldsymbol{x}, \omega)$ is a zero mean, unit variance Gaussian process with the set of hyper-parameters $\{\sigma^2, \boldsymbol{\theta}\}$. The autocorrelation function $R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$ describes the correlation between two samples given the correlation parameters $\boldsymbol{\theta}$. In this notion, $\{\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \, \boldsymbol{\alpha} \in \mathcal{A}\}$ is a set orthonormal polynomials indexed by the hyper-indices $\boldsymbol{\alpha}$, $\boldsymbol{a}$ is a set of parameters and $\boldsymbol{\theta}$ is the set of parameters describing the autocorrelation function.

The new metamodelling algorithm consists of two parts:

- the determination of a set of polynomials for the trend part
- the determination of the optimal correlation parameter $(\sigma^2, \boldsymbol{\theta})$ and trend parameters $\{\boldsymbol{a}_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{A}\}$.

The set of polynomials is computed using the PCE framework, i.e. the LARS algorithm. The computation of the parameters in the trend and in the correlation parts is carried out as in the universal Kriging algorithm. The two tasks are processed in series because the set of polynomials can be determined independently from the Kriging analysis.

Note that PCE can be interpreted as Kriging with uncorrelated samples, i.e. using a correlation matrix which is the identity matrix (Dirac autocorrelation function).

4.2 Algorithm

The link between the two distinct frameworks can be achieved in various ways. Two approaches are proposed here and validated on benchmark problems in Section 6.

- *The sequential approach in a global sense*: The first step is to find the optimal polynomial set of predictors using the PCE framework, i.e. the polynomial set with the smallest corrected leave-one-out error (Eq. (20)). Once the sparse PC basis is set, the universal Kriging algorithm is used. The error is estimated once at the end of

the algorithm where the performance is assessed using the leave-one-out error of Eq. (41). The underlying assumption is that the optimal set of polynomials that arises from the PCE algorithm is also the optimal (or close to optimal) set for the combined algorithm. In the following examples this approach is denoted as *FPC-Kriging* (Full-PC-Kriging);

- *The sequential approach in an iterative sense*: The second approach is used to test the assumption of optimality in FPC-Kriging. The procedure starts identically: A PCE algorithm selects the optimal set of polynomials. Inside PCE, the polynomials are chosen with the LARS algorithm: the orthogonal polynomial that are most correlated with the current residuals are chosen first. Then the computation of the universal Kriging metamodel is carried out iteratively at each LARS step. The initialisation is a universal Kriging algorithm with one single polynomial in the trend part. Iteratively, polynomials are added one-by-one according to the largest correlation with the current residuals till the error estimate converges to a minimum. The evolution of the metamodel accuracy is determined by using the leave-one-out error in Eq. (41). This approach is denoted as *OPC-Kriging* (Optimal-PC-Kriging).

The accuracy of both approaches are compared in Section 6 to the disjoined algorithms, namely PCE and Kriging.

4.3 Error estimation

PC-Kriging is a universal Kriging metamodel with an advanced trend part. Thus the error estimates of Section 3.4 are valid. In particular, the leave-one-out error in Eq. (41) is used to compare different PC-Kriging models and to compute the optimal metamodel in OPC-Kriging.

4.4 Implementation

The main program used for this calculation is Matlab. In Matlab the sparse set of polynomials is computed by the PCE method using the LARS algorithm described in Blatman (2009). The results are reliable and fast and thus no alternative approach is used.

The Kriging problem with different trend parts is solved using two open-source toolboxes:

- *DACE* (Lophaven et al., 2002) is a rather old Matlab toolbox. It can handle user-defined trend terms and uses a constrained optimization algorithm, called *boxmin*, to find the optimal hyperparameters of the Kriging model. The accessibility and adjustability is very user-friendly;

- *DiceKriging* (Roustant et al., 2013, 2012) is a toolbox written in the programming language R. A Matlab-based interface for R is available, i.e. R can be executed via Matlab. DiceKriging uses the BFGS-algorithm (Fletcher, 1970; Shanno, 1970) to

optimize the hyperparameter values, which is a quasi-Newton method. For more complex problems the optimization algorithm can be switched to a hybrid method which combines the BFGS-algorithm with a genetic algorithm. The genetic algorithm is used to search globally for the optimum whereas the BFGS-algorithm optimizes locally to a local optimum.

Although handling the DiceKriging toolbox requires more attention than handling DACE, DiceKriging outperforms DACE in terms of stability. DACE's optimization algorithm is weaker than DiceKriging's optimization algorithm as it stops in local minima or stops without convergence more often than DiceKriging. This becomes important for high-dimensional problems where the search for optimal parameters becomes more challenging. Moreover DiceKriging provides up-to-date Kriging features like correlation functions such as the Matérn autocorrelation function or the inbuilt "nugget" option. The nugget option may be used in order to regularize the problem in cases when the correlation matrix of the experimental design is close to singularity.

As conclusion, the combination of the Matlab PCE algorithm and the DiceKriging toolbox is implemented to calculate the optimal metamodels in Section 6.

5 Experimental design

In metamodelling the choice of the experimental design, i.e. the set of input samples, is crucial for an accurate representation of the computational model. Various approaches are available, from purely deterministic to fully stochastic sampling techniques. The key point is that the behaviour of the computational model is estimated with only a few data points, which are carefully chosen so that the experimental design covers the entire space of input parameters.

5.1 Regular grid design

Intuitively, the regular grid is a good choice to cover the whole space of input variables in a deterministic way. It is a full factorial design, i.e. all regions are covered regularly with the same density of samples in each subdomain. The regular gird also includes the boundary values at the limits of the space of input variables where the support is bounded.

A drawback of the regular grid is that the design is *full factorial*. This implies that there are a large number of samples involved, i.e. $N = l^M$ where l is the number of levels along each dimension and M is the number of variables (dimensions).

Another issue is the handling of Gaussian variables. Modelling Gaussian variables as the inverse Gaussian CDF of a uniform variable leads to samples located towards infinity. This causes numerical issues in metamodelling and may decrease the accuracy of the metamodel.

5.2 Monte Carlo sampling

In theory, Monte Carlo sampling is a purely stochastic design of experiments. The values of the samples are determined randomly according to an assumed probability density function of the input variables. In practice, random number generators like the Mersenne twister generate a sequence of pseudo-random numbers that resembles a set of random numbers.

The probability is high though that not the entire variable space is covered evenly by the experimental design points. Especially for small sample sizes there is a bias on the samples which may translate into an inaccurate metamodel. The bias reduces when the number of samples is increased which results in higher computational cost and time. Thus semi-random designs were created, *i.e.* designs which include randomness and also a deterministic part. Some of the various methods are described in the following Sections 5.3–5.6.

5.3 Latin-hypercube sampling

Latin-Hypercube sampling is the special case of an orthogonal array (presented more in details in Section 5.5) with a strength equal to 1. We assume here uniformly distributed, independent random variables for the sake of clarity. The property of this experimental design is that the projection onto any axis in the M-dimensional space results in a uniform distribution (McKay et al., 1979). In the 2-dimensional case the space of input parameters is defined by a regular grid. The samples are then arranged so that there is one sample in each column and in each row of the grid. Inside each square, the sample coordinates are chosen randomly. The algorithm to determine the sample placement reads as follows:

1. Generate $i = 1 \ldots, M$ random permutations $\boldsymbol{s}^{(i)}$ of the integer series $\{1, \ldots, N\}$ (N is the target number of samples). The matrix of the random permutations is $\boldsymbol{S}_{ij} = \boldsymbol{s}_j^{(i)}$ where $j = 1, \ldots, M$.

2. Compute the lower left corner coordinate of each hypercube by $\widehat{S}_{ij} = (S_{ij} - 1)/N$.

3. Sample uniformly in each hypercube resulting in the experimental design $\boldsymbol{\mathcal{X}} = \{\boldsymbol{\chi}^{(1)}, \ldots, \boldsymbol{\chi}^{(N)}\} = \widehat{S}^{\mathsf{T}} + U$ where U is a $M \times N$ matrix containing uniform random values over $[0, \frac{1}{N})$.

An illustrative sketch of a 2-dimensional sample space is shown in Figure 1 for a experimental design of $N = 10$ samples. Each axis is divided into $N = 10$ equal intervals which results in 100 subregions in the space of the input variables x_1 and x_2. The gray subregions represent the determined regions for samples. The matrix of the random permutations is in this case:

$$S^{\mathsf{T}} = \begin{bmatrix} 9 & 4 & 8 & 2 & 3 & 10 & 7 & 5 & 1 & 6 \\ 1 & 6 & 8 & 9 & 3 & 7 & 4 & 2 & 5 & 10 \end{bmatrix}$$

The samples are then chosen randomly in each region of the sample space as shown in Figure 1.

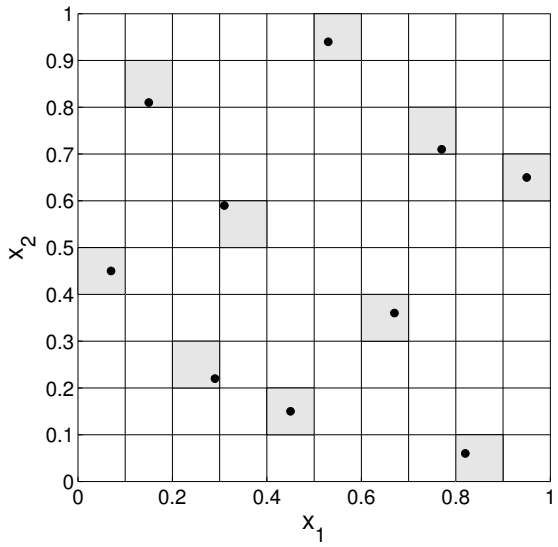


Figure 1: Latin-hypercube experimental design in the 2-dimensional space using 10 samples

Latin-Hypercube sampling is implemented in Matlab under the function name `lhsdesign`.

5.4 Quasi-random sequences

Quasi-random sequences are an alternative to Monte Carlo sampling. Also known as low-discrepancy sequences, they are deterministic sequences of points which cover almost uniformly the unit hypercube. Popular quasi-random sequences are the Sobol' sequences (Sobol, 1967), the Faure sequences (Faure, 1982), the Halton series (Halton, 1960) and the Niederreiter series (Niederreiter, 1988). The implementation of such sequences is available on the internet in different formats. For the Sobol' sequence is also implemented in Matlab under the command `sobolset`.

5.5 Orthogonal arrays

An orthogonal array is an $(N \times M)$-array from the variable set S with s levels. The notation for an orthogonal array A is given as $OA(N, M, s, t)$. N is the number of samples, M is the number of variables, s is the number of levels (different states), t is the strength of the orthogonal array. For all sub-array of A consisting of all N samples and $t \leq M$ variables (array of size $N \times t$), each possible permutation of levels can be found equally often, namely $\lambda = N/s^t$ times.

A simple example is presented here to show how OAs work. Consider a problem with $M = 4$ variables and two levels, namely 0 and 1 (e.g. on/off). Eight experiments are planned. One possible OA is then given in Table 2.

Table 2: Orthogonal array in four dimensions

Sample #	x_1	x_2	x_3	x_4
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

The strength t of this OA is determined by checking the entries of the sub-arrays of size $N \times t$. Strength $t = 1$ means that for any variable x_i each level appears the same number of times, i.e. four times (0) and four times (1). For strength $t = 2$ the sub-arrays of any two $x_i, x_j (i \neq j)$ are checked (arrays of size $N \times 2$). The number of each possible combination of values appear the same number of times, i.e. the combinations (0,0), (0,1), (1,0), and (1,1). Indeed for all possible sub-arrays each combination appears twice. The array in Table 2 is of strength $t = 3$, since the combinations (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), and (1,1,1) appear equally often, i.e. they appear once. Thus the sample array is called $OA(8, 4, 2, 3)$.

This framework can be adapted to more levels and different strengths. For the applications in Section 6, OA's of strength $t = 2$ are used. Each variable level s is interpreted as an interval. The samples are chosen randomly within the interval in order to include the idea of randomness in the designs.

Generating OAs is a complex topic. Computational approaches can be found in Zhang (2007); Suen and Kuhfeld (2005); Suen (2006); Heydayat et al. (1999), to name a few.

5.6 K-means clustering

Clustering is a numerical technique which allows one to classify a large sample set (say n samples) into k clusters. Monte Carlo sampling may be used to generate a large random sample set. The samples are separated into a smaller number of clusters given an objective function. For k-means clustering, the objective function can be the distance between the samples and their centroid. The algorithm iterates until the optimal centroids are found and summarizes as follows:

1. Generate a large sample set via Monte Carlo sampling;
2. Choose randomly k centroids;
3. Compute the distance of each sample to the centroids and determine the closest centroid of each sample;

4. Each cluster is formed by the samples which are closest to the centroid of the cluster;
5. Compute a new centroid point for each cluster (e.g. the nucleus of the subset);
6. Go back to step 3 unless the centroids do not change significantly anymore;
7. Finally, choose the centroids as the samples of the experimental design;

This simple clustering algorithm is available in Matlab called `kmeans(X,k)`. In the literature this approach is also called *Centroidal Voronoi Tessellation* (Du et al., 1999) because the result actually provides the centroids of a Voronoi tessellation.

The centroids are then taken as the samples of the experimental design. When assuming a uniform distribution for the underlying samples, the resulting centroids do not cover properly the uniform input space. Indeed, the uniformity of the final design is determined by projecting the centroids onto each dimension. Numerical experiments show that there is a trend to having more centroids towards the limits of the interval $[0, 1]$ for small number of clusters. This behaviour is explained by the radial character of the objective function (*i.e.* Euclidean distance). Considering for instance a 2-dimensional problem with 5 clusters, the optimal solution will be a central cluster and one cluster in each corner. Thus the bound values of the distribution are more represented than the values inside the domain.

Figure 2 illustrates the clustering in two dimensions using 10 clusters out of 10^5 uniformly distributed samples. The resulting cluster resembles a Voronoi tessellation. Note that when projecting the clusters to the horizontal axis, the clusters lie around the three values $0.15, 0.5, 0.85$, i.e. there is no uniformity in the distribution of the centroids.

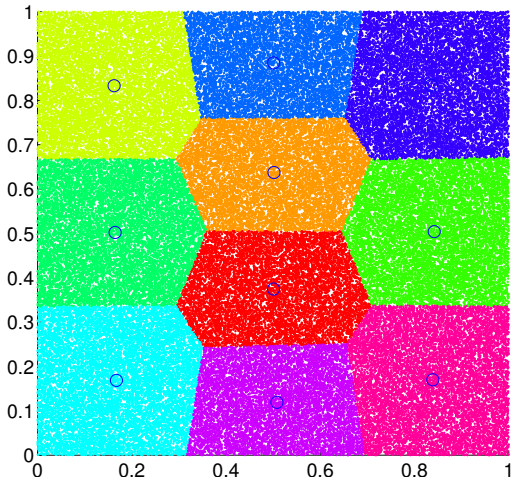
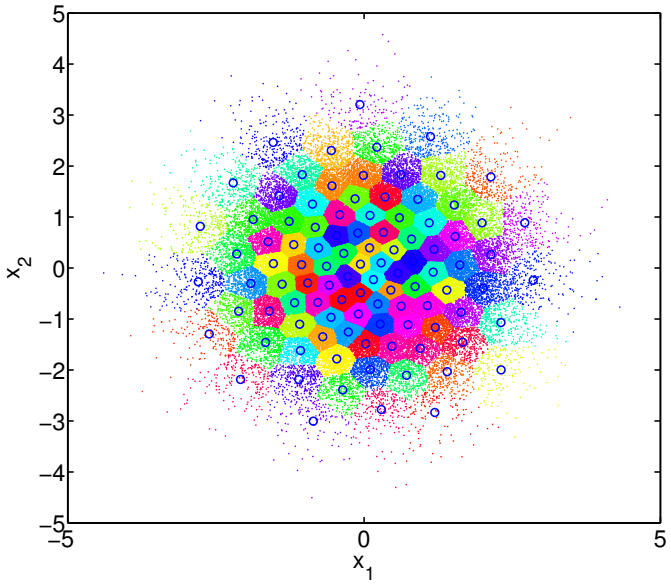
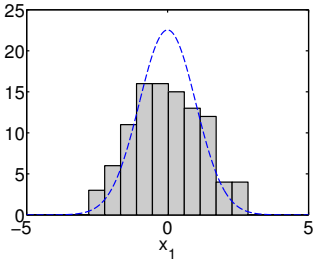


Figure 2: k-means clustering on 2-dimensional uniform variables (10 clusters, $n = 10^5$ samples): each cluster in a separate color, samples are marked with dots and centroids are marked with circles

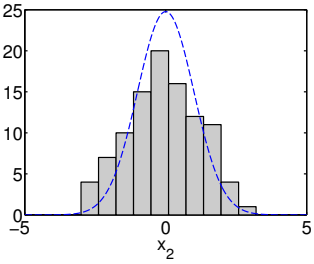
In the case of Gaussian input variables, the centroids are distributed closer to the original sampling distribution, i.e. the centroids are close to normally distributed. Figure 3(a) shows the same setting as in Figure 2 but with Gaussian variables and using 100 clusters. Additionally, the histograms of the projection onto the two axes are plotted and compared to the theoretical optimal Gaussian PDF in Figure 3(b) and Figure 3(c) respectively. The results are graphically much more satisfactory than in the uniform case.



(a) Clustering



(b) Projection on x_1



(c) Projection on x_2

Figure 3: k-means clustering on 2-dimensional Gaussian variables (100 clusters, $n = 10^5$ samples): Fig. 3(a) shows the samples (dots) and the clusters centroids (circles), each cluster in a different color; Fig. 3(b) and Fig. 3(c) show the histogram of the centroids projected onto the coordinate axes and the theoretical Gaussian PDF

6 Benchmark problems

6.1 Setup

The PC-Kriging approach is used to metamodel the following analytical functions which are benchmark problems in the field of metamodelling. Four functions with uniformly distributed input variables and two with Gaussian input are considered:

- *Ishigami function*: The input vector $\boldsymbol{X}$ contains 3 independent random variables which are uniformly distributed over the interval $[-\pi, \pi]$:

$$f_1(\boldsymbol{X}) = \sin X_1 + 7 \sin^2 X_2 + 0.1 X_3^4 \sin X_1 \tag{46}$$

- *Sobol' function*: The input vector $\boldsymbol{X}$ contains 8 independent random variables which are uniformly distributed over the interval $[0, 1]$. The function parameter $\boldsymbol{c}$ is chosen as $\boldsymbol{c} = (1, 2, 5, 10, 20, 50, 100, 500)^\mathsf{T}$ as in Sudret (2008).

$$f_2(\boldsymbol{X}) = \prod_{i=i}^{M} \frac{|4 X_i - 2| + c_i}{1 + c_i} \tag{47}$$

- *Rosenbrock function*: The input random variables $\boldsymbol{X} = \{X_i,\, 1, \ldots, M\}$ are independent uniformly distributed variables over $[-2, 2]$. The size of $\boldsymbol{X}$ can be varied to investigate the effect of the dimensionality M.

$$f_3(\boldsymbol{X}) = \sum_{i=1}^{M-1} \left[100 \left(X_{i+1} - X_i^2\right)^2 + (1 - X_i)^2\right] \tag{48}$$

  In this report M is set equal to two:

$$f_3(\boldsymbol{X}) = 100 \left(X_2 - X_1^2\right)^2 + (1 - X_1)^2 \tag{49}$$

- *Morris function*: $\boldsymbol{X}$ is a vector containing 20 independent variables, uniformly distributed over $[0, 1]$.

$$f_4(\boldsymbol{X}) = \sum_{i=1}^{20} \beta_i \, w_i + \sum_{i<j}^{20} \beta_{ij} \, w_i w_j + \sum_{i<j<l}^{20} \beta_{ijl} \, w_i w_j w_l + 5 \, w_1 w_2 w_3 w_4 \tag{50}$$

  where $w_i = 2\,(X_i - 1/2)$ for all i except for $i = 3, 5, 7$ where $w_i = 2\,(\dfrac{1.1\, X_i}{X_i + 0.1} - 1/2)$. The coefficients are defined as:

$$\begin{aligned} \beta_i &= 20,\ i = 1, \ldots, 10 \\ \beta_{ij} &= -15,\ i, j = 1, \ldots, 6 \\ \beta_{ijl} &= -10,\ i, j, l = 1, \ldots, 5 \end{aligned}$$

  The remaining coefficients are set to $\beta_i = (-1)^i$ and $\beta_{ij} = (-1)^{i+j}$ (Blatman, 2009),

which is a slight modification of the original version by Morris (1991) with standard Gaussian distributed parameters β_i and β_{ij}.

- *Rastrigin function*: $\boldsymbol{X}$ is composed of 2 independent standard Gaussian variables.

$$f_5(\boldsymbol{X}) = 10 - \sum_{i=1}^{2}(X_i^2 - 5\,\cos(2\pi\,X_i)) \tag{51}$$

- *O'Hagan function*: $\boldsymbol{X}$ is a vector of length 15 containing independent standard Gaussian variables. The vectors $\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}$ and the matrix $\boldsymbol{Q}$ are chosen as shown in Oakley and O'Hagan (2004), see Appendix A for the values.

$$f_6(\boldsymbol{X}) = \boldsymbol{a_1}^\mathsf{T}\boldsymbol{X} + \boldsymbol{a_2}^\mathsf{T}\sin(\boldsymbol{X}) + \boldsymbol{a_3}^\mathsf{T}\cos(\boldsymbol{X}) + \boldsymbol{X}^\mathsf{T}\,\boldsymbol{Q}\,\boldsymbol{X} \tag{52}$$

Note that the first four functions f_1 – f_4 use uniform random variables as input and Legendre polynomials in the polynomial chaos expansions whereas the functions f_5 and f_6 use Gaussian random variables as input and Hermite polynomials accordingly.

6.2 Analysis

As the analytical benchmark functions are fast to evaluate, the accuracy of the metamodel can be assessed by the relative generalized error Eq. (18) is obtained from a large validation set of size $n = 10^5$:

$$\epsilon_{gen} \equiv \frac{\sum_{i=1}^{n}\left(\mathcal{M}(\boldsymbol{x}^{(i)} - \widehat{\mathcal{M}}(\boldsymbol{x}^{(i)}\right)^2}{\sum_{i=1}^{n}\left(\mathcal{M}(\boldsymbol{x}^{(i)} - \mu_y\right)^2} \tag{53}$$

where $\mathbb{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$ contains the input sample values sampled according to the variable's input distributions and $\mathbb{Y} = \{y^{(i)}, \ldots, y^{(n)}\}$ contains the corresponding output sample values. $\widehat{\mathcal{M}}(\boldsymbol{x}^{(i)})$ is the metamodel evaluated on the sample $\boldsymbol{x}^{(i)}$ and μ_y is the mean value of the output over the validation set $\mathbb{Y}$. This is the basis for comparing different setups of the combination of PCE and Kriging. The surrogate models which are compared in the following subsection are:

- *PCE*: The metamodel is based on the description in Section 2. A sparse predictor set is determined using the LARS algorithm according to the smallest corrected leave-one-out error (Eq. (25)).

$$\mathcal{M}^{(\mathsf{PCE})}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} a_{\boldsymbol{\alpha}}\,\psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{54}$$

- *Kriging*: The metamodel consists of an ordinary Kriging model, where the regression part is a constant (unknown) value β_0:

$$\mathcal{M}^{(\mathsf{OK})}(\boldsymbol{x}) = \beta_0 + \sigma^2\,Z(\boldsymbol{x}, \omega) \tag{55}$$

- *FPC-Kriging*: This is the first sequential procedure described in Section 4. First, the optimal predictors are determined as in the classical PCE procedure. Then this optimal set of predictors is taken as the trend (regression) component of a universal Kriging metamodel;
- *OPC-Kriging*: This metamodel takes the set of predictors of the case *P*CE as input set of possible regression terms of a universal Kriging model. The predictor terms are added one-by-one, based on the most correlated predictor, as found by LARS. A Kriging model is fitted for each different trend. The evolution of the leave-one-out error (Eq. (41)) is monitored and the setup with the minimum error is eventually selected.

The generalized empirical error Eq. (53) with a large number of Monte Carlo samples is used to validate and compare the different approaches. This is reasonable for these analytical functions as they are fast to evaluate. The leave-one-out cross-validation error Eq. (41) is also provided because this error measure is based on the experimental design points only. It would be the only measure of error available when dealing with expensive-to-evaluate models. Note that it is also used in the OPC-Kriging case to choose the optimal configuration (Section 4).

The comparison of the various approaches is carried out using a varying number of samples in the experimental design. Latin-hypercube sampling (McKay et al., 1979) is used to generate the experimental designs (Section 5.3). To ensure the statistical stability, 50 independent runs are carried out and the results are presented using box plots. In the box plot the central mark is the median value (over the 50 independent runs), the edges are the 25th and the 75th percentile. The whiskers describe the boundary to the outliers. Outliers are defined as smaller than $q_{25} - 1.5\,(q_{75} - q_{25})$ or larger than $q_{75} + 1.5\,(q_{75} - q_{25})$ where q_{25} and q_{75} are the values of the 25th and 75th percentiles.

6.3 Results

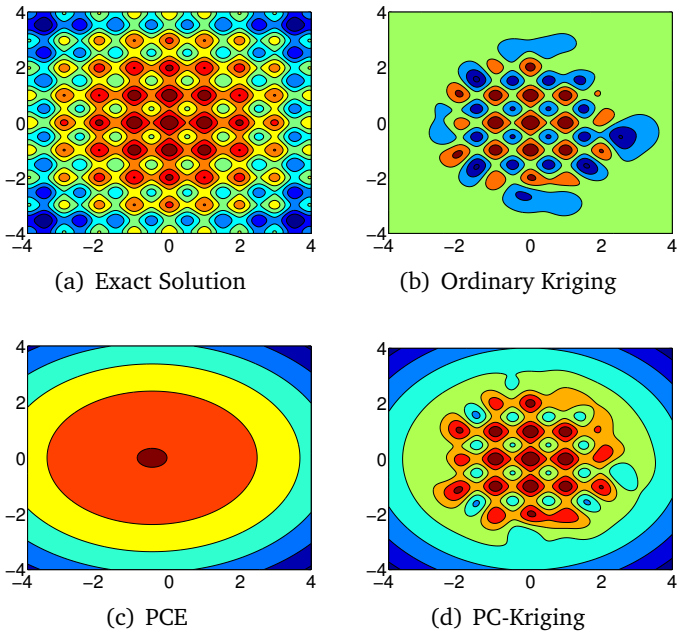
6.3.1 Illustration of the behaviour of PC-Kriging

In Figure 4, the contour plot of the Rastrigin function (Eq. (51)) in two dimensions is plotted. The plot compares the exact model to the three surrogate-modelling techniques, namely PCE, ordinary Kriging and PC-Kriging (exactly FPC-Kriging). The Rastrigin function results in highly volatile output values over the entire variable space due to the high-frequent cosine function inside the model. This behaviour is difficult to approximate with a small number of samples as many local extrema are missed out. The PCE method manages to model only the global behaviour (quadratic function) which can be seen in Fig 4(c).

On the other hand, the ordinary Kriging model (Fig. 4(b)) interpolates the experimental design exactly and thus captures the volatile output well in the regions with many samples. Note that the input variables are standard Gaussian random variables

and thus most of the samples are located around the origin.

The combination of both approaches in PC-Kriging (Fig. 4(d)) leads to a better overall surrogate model. The figure shows the combination of the characteristics of the PCE and the ordinary Kriging model, i.e. the global behaviour is modelled by PCE and the local variability by Kriging.



(a) Exact Solution (b) Ordinary Kriging

(c) PCE (d) PC-Kriging

Figure 4: Composition of the PC-Kriging on the Rastrigin function ($N = 128$ points in the experimental design)

6.3.2 Comparing different surrogate-modelling techniques

The performance of the four surrogate-modelling approaches is illustrated on the six analytical functions. The comparison is carried out via the relative generalized error in Eq. (53) based on $n = 10^5$ random samples in the validation set. The experimental designs are Latin-hypercube sampling designs of various sizes. Figures 5–10 show the boxplots of the generalized errors representing 50 independent runs of the same algorithm as a function of the number of samples in the experimental design. In each figure (a) shows the ordinary Kriging model, (b) shows the PCE with the optimal maximum degree of polynomials. The results of the new PC-Kriging approaches are reported in (c) and (d). The number of samples in the experimental design is chosen in a way that a large range of resulting relative generalized error is covered and focus on small experimental designs.

In Figure 5 the generalized error estimate of the Ishigami function is plotted. For $N = 20$ samples ordinary Kriging performs the best in terms of median value. OPC-Kriging is a little less accurate in the median but varies much more than the other methods. This means that also for only 20 samples, very low errors may be achieved. When increasing

the number of samples, PC-Kriging performs better than the distinct techniques on their own. OPC-Kriging has slightly smaller error estimates as FPC-Kriging.

The Rosenbrock function (in Figure 6) results in low error estimates for small experimental designs. Similarly to the Ishigami function, OPC-Kriging performs better than the other methods, even for the smallest sample size ($N = 8$). $N = 20$ samples are sufficient to metamodel this function with a high accuracy due to the polynomial formulation of the function.

Figure 7 presents the Sobol' function. Focusing on the very low sample size ($N = 16,\ 32$) OPC-Kriging performs best when comparing the median value of the 50 independent runs. Ordinary Kriging and PCE perform very similar but worse than PC-Kriging.

For the Morris function in Figure 8 OPC-Kriging and ordinary Kriging behave similar, though PC-Kriging has slightly smaller errors. PCE performs worse than the other techniques in this setting. Considering that the Morris function has a 20-dimensional input vector, the $N = 256$ samples result in a low error.

The first function with Gaussian input variables, namely the Rastrigin function, is shown in Figure 9. Due to the nature of the function many samples are needed to obtain a low error estimate ($N = 32-256$). Throughout the whole range presented in the figure, OPC-Kriging outperforms the other three techniques. Ordinary Kriging performs worst in this setting.

The sixth function is the O'Hagan function which is presented in Figure 10. The performance of PC-Kriging resembles more the ordinary Kriging results than the PCE results. The errors of ordinary Kriging and OPC-Kriging are comparable, though OPC-Kriging performs slightly better.

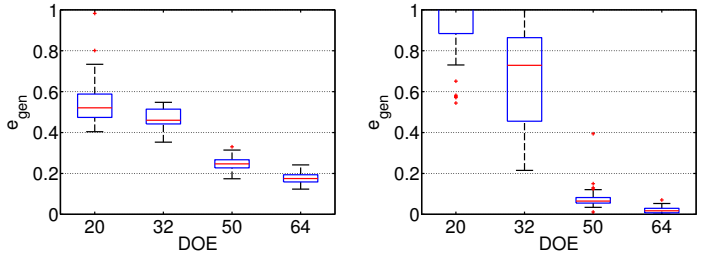
Considering all six figures, the behaviour of the meta-modelling approaches differs in the various analytical functions $f_1 - f_6$. Functions like O'Hagan and Morris are hard to model with PCE. When summarizing the results of all six analytical functions, the generalized error is similar for the two simple approaches (for small experimental designs), namely ordinary Kriging and PCE. The new PC-Kriging approaches are at least as good as the PCE and the ordinary Kriging. In many cases the error obtained in the PC-Kriging is smaller than in the simple approaches PCE and Kriging.

Within the PC-Kriging approaches ((c) and (d)), the OPC-Kriging appears to be more accurate than the FPC-Kriging. This is explained by the setup of the two approaches. OPC-Kriging takes the optimal set of polynomials of the combined calculation of PCE and Kriging. This reduces the number of polynomials but also decreases the mean-square error. FPC-Kriging optimizes the set of polynomials *only once* based on the information of the sole LAR selection algorithm and thus cannot be optimal in all cases.

The increased accuracy comes with a higher computational cost. The single approaches PCE and Kriging have the lowest computational costs, FPC-Kriging has an intermediate cost and OPC-Kriging has the highest cost. The high costs in OPC-Kriging results from the iterative calculation of the optimal set of polynomials which implies the computation of many PC-Kriging models. For a single run of the Ishigami-function and 128

samples in the experimental design, the CPU times (using a 3.2GHz, 4-core computer) for PCE, Kriging, FPC-Kriging and OPC-Kriging are 0.4 sec., 1.8 sec., 7.0 sec., and 120 sec. respectively.

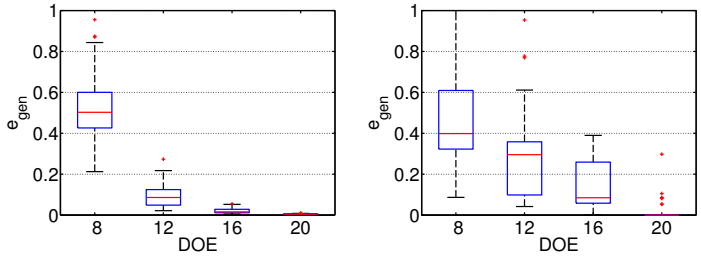
However it is intended to apply these techniques to real problems in which a single evaluation of the original computational model may last minutes to hours. So the apparent overload of OPC-Kriging is not an issue in practice.



(a) Ordinary Kriging (b) PCE

(c) FPC-Kriging (d) OPC-Kriging

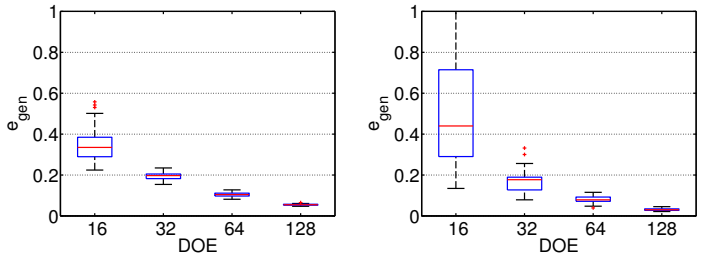
Figure 5: Ishigami function for the various meta-modelling approaches



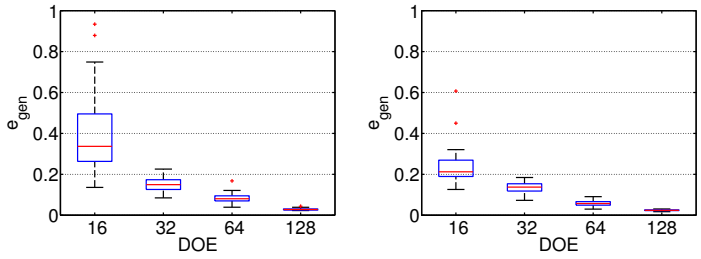
(a) Ordinary Kriging (b) PCE

(c) FPC-Kriging (d) OPC-Kriging

Figure 6: Rosenbrock function for the various meta-modelling approaches

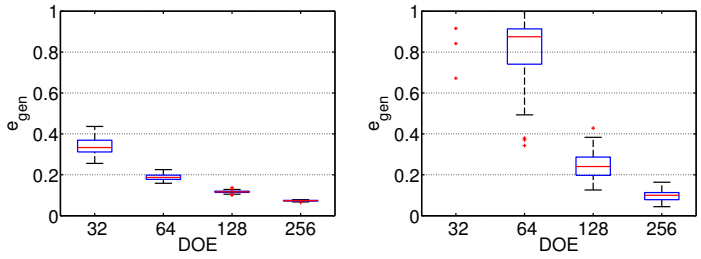


(a) Ordinary Kriging (b) PCE

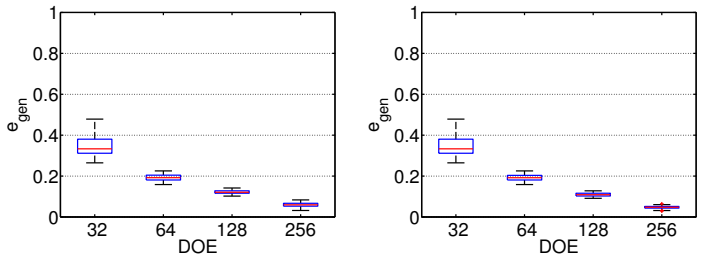


(c) FPC-Kriging (d) OPC-Kriging

Figure 7: Sobol' function for the various meta-modelling approaches

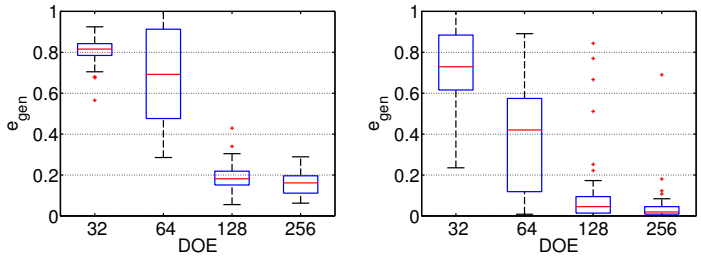


(a) Ordinary Kriging (b) PCE

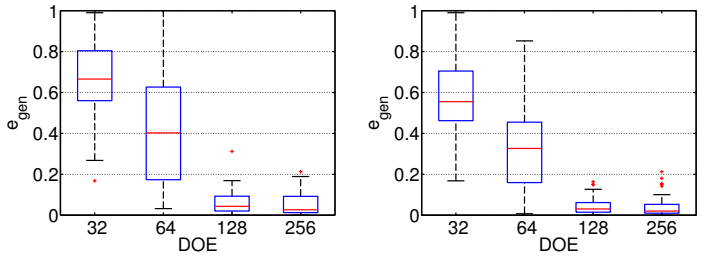


(c) FPC-Kriging (d) OPC-Kriging

Figure 8: Morris function for the various meta-modelling approaches

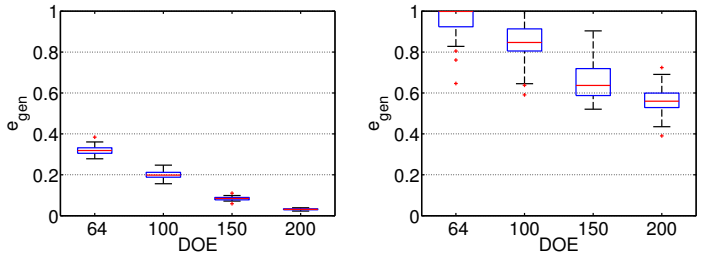


(a) Ordinary Kriging (b) PCE

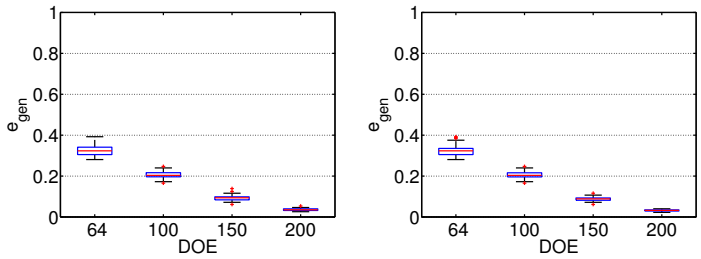


(c) FPC-Kriging (d) OPC-Kriging

Figure 9: Rastrigin function for the various meta-modelling approaches



(a) Ordinary Kriging (b) PCE



(c) FPC-Kriging (d) OPC-Kriging

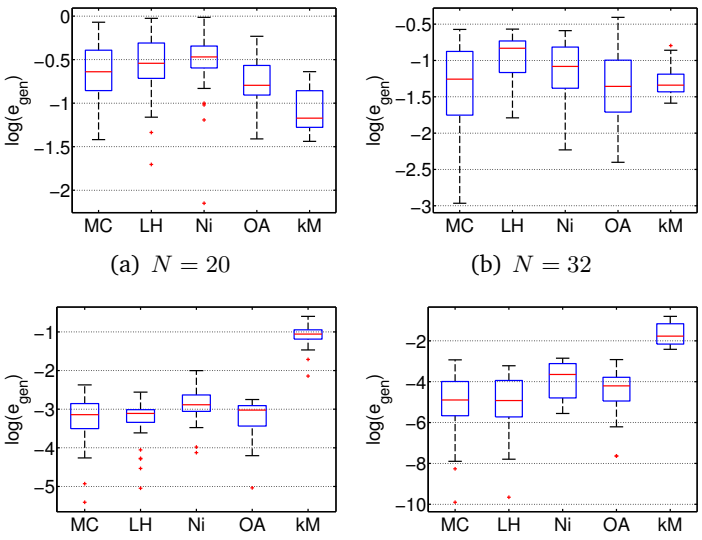
Figure 10: O'Hagan function for the various meta-modelling approaches

6.3.3 Comparing different sampling techniques

In Figure 11 and Figure 12 the design of experiments is generated through various sampling algorithms. The figures include the Monte Carlo sampling (denoted *MC* in the figures), Latin-Hypercube sampling (*LH*), Niederreiter sequences (*Ni*), orthogonal arrays of strength $t = 2$ (*OA*), and k-means clustering (*kM*). The experimental setup is similar to the one previously mentioned: 50 independent runs of OPC-Kriging with $n = 10^5$ random samples to assess the relative generalized error Eq. (53) of each experimental design. Figure 11 illustrates the behaviour of metamodelling the Ishigami function (Legendre polynomials) whereas Figure 12 illustrates the Rastrigin function with Hermite polynomials. The error estimate is calculated on a logarithmic scale (base 10) in order to make the ranges of the error more visible. In Figure 11 and Figure 12 the error range from 10^0 for $N = 20$ to 10^{-10} and 10^{-12} respectively for $N = 64$.

The functions have been chosen to present one example with uniform input variables and one with Gaussian input variables. The corresponding set of orthonormal polynomials used in the PC expansions are the Legendre polynomials and the Hermite polynomials respectively. The sampling methods are compared to the random sampling method of Monte Carlo sampling. The quasi-random sampling methods are in general better than the Monte Carlo sampling comparing the median values of the relative generalized error Eq. (53). LHS, Niederreiter sequences, and OA of strength 2 behave very similar. The error is comparable and the ranking varies depending on the function and on the sample size. None of the quasi-random approaches outperforms the others in this setting.

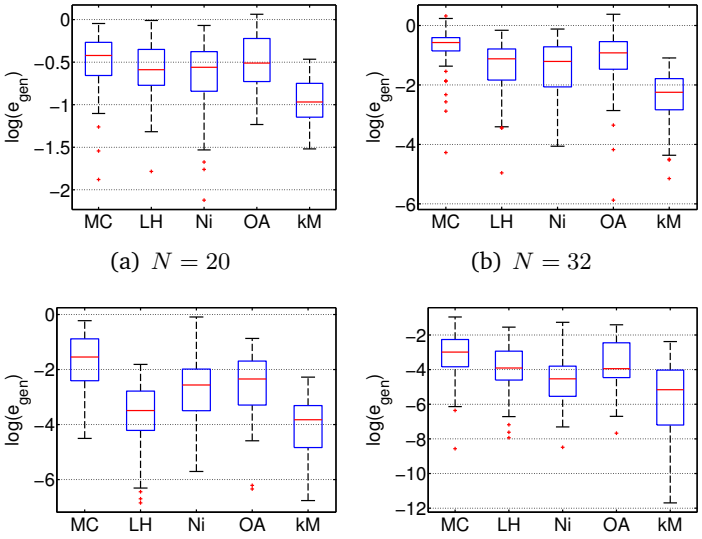
One special case is the k-means clustering algorithm though, which behaves differently for uniform and Gaussian random variables. When increasing the number of k-means clustering samples in the Ishigami function the relative generalized error decreases much more slowly than when using the other quasi-random sampling methods. The reason is the effect described in Section 5.6, i.e. there are more clusters towards the boundaries of the sampling domain rather than inside the domain. For the case of Gaussian random variables, this effect is reversed. The k-means clustering method leads to considerably lower error estimates and thus to better metamodels than the other sampling methods. Note that very low errors are more likely to appear in the k-means clustering method than in the other methods.



(a) $N = 20$ (b) $N = 32$

(c) $N = 50$ (d) $N = 64$

Figure 11: Ishigami function – Several experimental designs compared with respect to the associated relative generalized error; *MC* stands for Monte Carlo sampling, *LH* for Latin-Hypercube Sampling, *Ni* for Niederreiter sequences, *OA* for orthogonal arrays and *kM* for k-means clustering sampling.



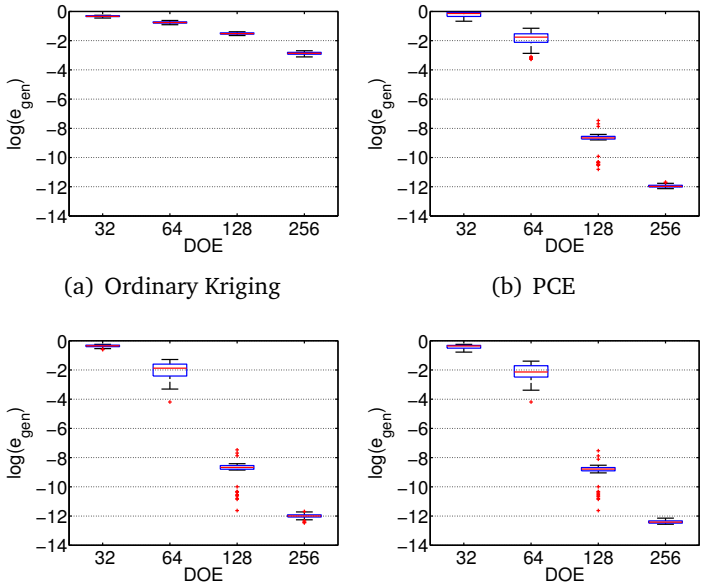
(a) $N = 20$ (b) $N = 32$

(c) $N = 50$ (d) $N = 64$

Figure 12: Rastrigin function – Several experimental designs compared with respect to the associated relative generalized error; *MC* stands for Monte Carlo sampling, *LH* for Latin-Hypercube Sampling, *Ni* for Niederreiter sequences, *OA* for orthogonal arrays and *kM* for k-means clustering sampling.

6.3.4 Behaviour of large experimental designs

In the previous sections the focus was on small sample sets because this is an important case for realistic problems. The behaviour for large sample sizes is shown in this section to illustrate the convergence rate as a function of the number of samples. Figure 13 illustrates the evolution of the relative generalized error (on logarithmic scale, base 10) from small to large sample sizes. Latin-hypercube sampling is used and the boxplots inherit the information of 50 distinct, independent runs. The figure shows the evolution of the four approaches, namely (a) ordinary Kriging, (b) PCE, (c) FPC-Kriging and (d) OPC-Kriging.



(a) Ordinary Kriging (b) PCE

(c) FPC-Kriging (d) OPC-Kriging

Figure 13: Ishigami function – large range of number of samples

The relative generalized error decreases fast with enlarging the sample set because the Ishigami function is a rather smooth function. It consists of sine and cosine functions (over a bounded domain) which can be approximated well by a sum of polynomials, i.e. a Taylor expansion. Thus the PCE is the dominating effect in the PC-Kriging in this example. The evolution of the error of the PC-Kriging approaches is very similar to the evolution of the error in the pure PCE case. The Kriging algorithm has the same order of magnitude of error for small sample sizes. For large sample sizes, PCE outperforms Kriging though.

Kriging has an issue with large sample sizes as the interpolation algorithm is based on the inversion of matrices. The size of these matrices is proportional to the squared number of samples $N \times N$. The more points added to the experimental design, the less stable the inversion due to singularities and bad conditioning in the inversion matrix. Thus Kriging is only suitable for rather small sample sizes. If a large sample size is available, then regional Kriging models on a subset of samples (i.e. the neighbouring

samples) are more suitable (Dubrule, 1983).

6.3.5 Evolution of the leave-one-out error in PC-Kriging

Figure 14 shows the evolution of the leave-one-out error in the case of a single run of OPC-Kriging. The set of polynomials in the trend part of PC-Kriging increases starting from the most correlated polynomial term towards the least correlated terms. The red, dotted line represents the leave-one-out error calculated from Eq. (41) using only the information in the experimental design. The black, solid line represents the empirical validation error calculated with $n = 10^5$ independent random samples based on Eq. (53). The end of the lines at 56 polynomials corresponds to the model which is chosen as the optimal set of polynomials by the LARS algorithm in PCE, i.e. FPC-Kriging. Thus OPC-Kriging is obviously not equivalent to the optimal model FPC-Kriging. The minimum error is found when using 41 polynomials instead of 56 according to the respective relative generalized errors. The leave-one-out error models nicely the evolution of the relative generalized error for comparably small numbers of polynomials. At the stage where the error becomes stable (number of polynomials larger than 28), the algorithm should terminate with a stopping criterion.

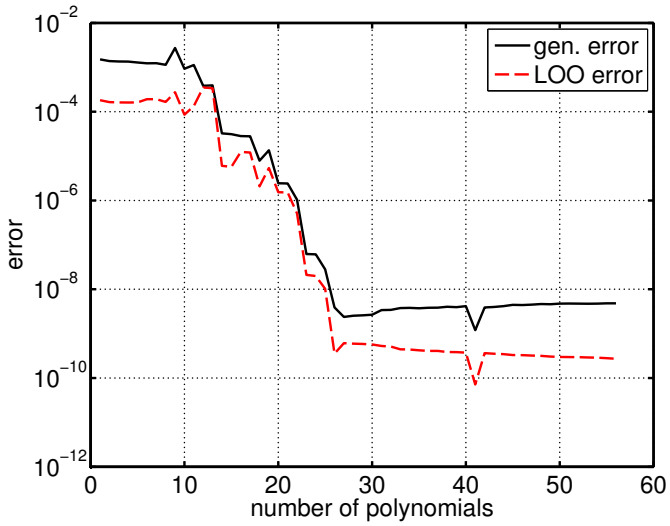


Figure 14: Evolution of the LOO and relative generalized error inside the OPC-Kriging algorithm as a function of the number of polynomials in the regression part

A simple stopping criterion is to compute all PC-Kriging models possible in the OPC-Kriging approach (here 56) and then choosing the set of polynomials with the minimal relative generalized error. A different stopping criterion may identify the plateau and return the last number of polynomials which significantly reduced the leave-one-out error.

6.3.6 Conclusion

The PC-Kriging model is at least as good as the single approaches PCE and ordinary Kriging. In practice, as there is usually no prior knowledge of the underlying process, this

leads to a metamodel with a performance close to the best model of either PCE or Kriging. In reality, it is not known which approach (PCE or Kriging) models the hidden process more accurately. Depending on the computational cost one wants to invest, either the OPC-Kriging or the FPC-Kriging is appropriate for the analysis of the problem. Considering that the purpose of metamodelling is avoiding many computations of the expensive real process and that the computational cost of the original computational model $\mathcal{M}$ is usually orders of magnitude higher, the OPC-Kriging should be used in any case.

The use of quasi-random experimental designs is preferable to crude Monte Carlo sampling. Dependent on the problem one or the other quasi-random sampling approach performs slightly better, which in practical applications is not known in advance. As illustrated for the Rastrigin function, the k-means clustering algorithm is preferred for Gaussian input variables but not for uniform input variables.

7 Conclusion and outlook

Polynomial chaos expansion (PCE) and Kriging are two distinct non-intrusive metamodelling techniques. In this report these approaches are discussed and a new technique called PC-Kriging is proposed. PC-Kriging is based on the universal Kriging algorithm where the trend is represented by a sum of orthonormal polynomials. The least-angle regression algorithm, which is used in PCE algorithms, determines the optimal sparse set of orthonormal polynomials in coherency with the input variables' distributions.

The combination of PCE and Kriging opens new opportunities to better surrogate computational models in a non-intrusive setup, as validated on several analytical benchmark functions. PC-Kriging behaves better or at least as good as the single approaches PCE and Kriging separately. In terms of prediction error, PC-Kriging provides a framework to determine new sample points which opens the path to adaptive designs and to further research in applying PC-Kriging in reliability analysis.

Thus the next issue is the validation of the theory on real engineering problems. Engineering problems focus on different aspects compared to the aspects covered here, e.g. on the reliability of structures. Reliability assessment is based on the evaluation of a large number of virtual systems in order to estimate a probability of failure. As failure probabilities are small in engineering practice, the generation and the prediction accuracy of extreme events is crucial to the accuracy of the reliability. This leads to an adaptive design of experiments. In this report, the experimental design is based on the input distribution of the random variables. In reliability problems, the focus is on the failure domain though. Adaptive designs generating additional samples in the region of interest increase the predictability of the metamodel. Related topics in research are importance sampling and subset simulation which might be incorporated into an adaptive framework.

A O'Hagan function

$\boldsymbol{X}$ is a standard Gaussian variables vector of dimension 15. Vectors $\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}$ and the matrix $\boldsymbol{Q}$ are chosen as shown in Oakley and O'Hagan (2004).

$$f(\boldsymbol{X}) = \boldsymbol{a_1}^\mathsf{T}\boldsymbol{X} + \boldsymbol{a_2}^\mathsf{T}\sin(\boldsymbol{X}) + \boldsymbol{a_3}^\mathsf{T}\cos(\boldsymbol{X}) + \boldsymbol{X}^\mathsf{T}\boldsymbol{Q}\,\boldsymbol{X} \tag{56}$$

In this paper the vectors $\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}$ and the matrix $\boldsymbol{Q}$ are given by:

$$\boldsymbol{a}_1 = \begin{pmatrix} 0.0118 \\ 0.0456 \\ 0.2297 \\ 0.0393 \\ 0.1177 \\ 0.3865 \\ 0.3897 \\ 0.6061 \\ 0.6159 \\ 0.4005 \\ 1.0741 \\ 1.1474 \\ 0.7880 \\ 1.1242 \\ 1.1982 \end{pmatrix}, \boldsymbol{a}_2 = \begin{pmatrix} 0.4341 \\ 0.0887 \\ 0.0512 \\ 0.3233 \\ 0.1489 \\ 1.0360 \\ 0.9892 \\ 0.9672 \\ 0.8977 \\ 0.8083 \\ 1.8426 \\ 2.4712 \\ 2.3946 \\ 2.0045 \\ 2.2621 \end{pmatrix}, \boldsymbol{a}_3 = \begin{pmatrix} 0.1044 \\ 0.2057 \\ 0.0774 \\ 0.2730 \\ 0.1253 \\ 0.7526 \\ 0.8570 \\ 1.0331 \\ 0.8388 \\ 0.7970 \\ 2.2145 \\ 2.0382 \\ 2.4004 \\ 2.0541 \\ 1.9845 \end{pmatrix}$$

$$
\boldsymbol{Q} = \begin{bmatrix}
-0.0225 & -0.1850 & 0.1342 & 0.3687 & 0.1717 & 0.1365 & -0.4403 & -0.0814 & 0.7132 & -0.4436 & 0.5038 & -0.0241 & -0.0459 & 0.2167 & 0.0559 \\
0.2566 & 0.0538 & 0.2580 & 0.2380 & -0.5913 & -0.0816 & -0.2875 & 0.4158 & 0.4975 & 0.0839 & -0.1106 & 0.0332 & -0.1398 & -0.0310 & -0.2232 \\
-0.0560 & 0.1954 & 0.0955 & -0.2863 & -0.1444 & 0.2237 & 0.1453 & 0.2900 & 0.2311 & -0.3193 & -0.2904 & -0.2096 & 0.4314 & 0.02440 & 0.0449 \\
0.6645 & 0.4307 & 0.2992 & -0.1620 & -0.3148 & -0.3903 & 0.1768 & 0.0580 & 0.1723 & 0.1347 & -0.3528 & 0.2515 & -0.0188 & 0.3648 & -0.3250 \\
-0.1213 & 0.1246 & 0.1066 & 0.0466 & -0.2168 & 0.1949 & -0.0655 & 0.02440 & -0.0968 & 0.1937 & 0.3335 & 0.3130 & -0.0836 & -0.2534 & 0.3733 \\
-0.2838 & -0.3282 & -0.1050 & -0.2207 & -0.1371 & -0.1443 & -0.1150 & 0.2242 & -0.0304 & -0.5151 & 0.0173 & 0.0390 & 0.3607 & 0.3090 & 0.0500 \\
-0.0779 & 0.0037 & 0.8869 & -0.2659 & -0.0793 & -0.0427 & -0.1865 & -0.3560 & -0.1750 & 0.0887 & 0.4003 & -0.0560 & 0.1372 & 0.2149 & -0.0113 \\
-0.0923 & 0.5921 & 0.0313 & -0.0331 & -0.2431 & -0.0998 & 0.0345 & 0.0951 & -0.3380 & 0.0064 & -0.6121 & 0.0813 & 0.8868 & 0.1425 & 0.1478 \\
-0.1319 & 0.5288 & 0.1265 & 0.0451 & 0.5837 & 0.3729 & 0.1140 & -0.2948 & -0.5701 & 0.4629 & -0.0941 & 0.1396 & -0.3861 & -0.449 & -0.1460 \\
0.0581 & -0.3229 & 0.0931 & 0.0724 & -0.5692 & 0.5255 & 0.2366 & -0.0118 & 0.0718 & 0.0783 & -0.1336 & 0.2272 & 0.1437 & -0.452 & -0.5557 \\
0.6615 & 0.3463 & 0.1410 & 0.5188 & -0.2802 & -0.1603 & -0.0684 & -0.2043 & 0.0697 & 0.2311 & -0.0444 & -0.1646 & 0.2162 & 0.0043 & -0.0874 \\
0.3160 & -0.0276 & 0.1343 & 0.1350 & 0.0540 & -0.1737 & 0.1753 & 0.0603 & -0.1791 & -0.3106 & -0.2536 & 0.0258 & -0.4301 & -0.6227 & -0.0340 \\
-0.2904 & 0.0341 & 0.0349 & -0.1212 & 0.0260 & -0.3355 & -0.4142 & 0.0532 & -0.271 & -0.0263 & 0.4102 & 0.2664 & 0.1558 & -0.1867 & 0.0199 \\
-0.2439 & -0.4410 & 0.0126 & 0.2495 & 0.0711 & 0.2462 & 0.1748 & 0.0085 & 0.2515 & -0.1466 & -0.0846 & 0.3693 & -0.2996 & 0.1104 & -0.7569 \\
0.0415 & -0.2598 & 0.4640 & -0.3611 & -0.9498 & -0.1650 & 0.0031 & 0.0528 & 0.2252 & 0.3839 & 0.4556 & -0.1863 & 0.0082 & 0.1667 & 0.1605
\end{bmatrix}
$$

References

Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecifications. *Comp. Stat. Data An. 66*, 55–69.

Bect, J., D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Stat. Comput. 22*(3), 773–793.

Berveiller, M., B. Sudret, and M. Lemaire (2006). Stochastic finite elements: a non intrusive approach by regression. *Eur. J. Comput. Mech. 15*(1-3), 81–92.

Bichon, B., J. McFarland, and S. Mahadevan (2011). Efficient surrogate models for reliability analysis of systems with multiple failure modes. *Reliab. Eng. Sys. Safety 96*(10), 1386–1395.

Blatman, G. (2009). *Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis*. Ph. D. thesis, Université Blaise Pascal, Clermont-Ferrand.

Blatman, G. and B. Sudret (2008). Adaptive sparse polynomial chaos expansions - application to structural reliability. In *Proc. 4th Int. ASRANet Colloquium, Athens, Greece*.

Blatman, G. and B. Sudret (2010). An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Prob. Eng. Mech. 25*(2), 183–197.

Blatman, G. and B. Sudret (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *J. Comput. Phys 230*, 2345–2367.

Candes, E. and T. Tao (2007). The Dantzig selector: statistical estimator when P is much larger than N. *Ann. Statist. 35*, 2313 – 2351.

Chapelle, O., V. Vapnik, and Y. Bengio (2002). Model selection for small sample regression. *Mach. Learn. 48*, 9 – 23.

Du, Q., V. Faber, and M. Gunzberger (1999). Centroidal Voronoi tessellations: applications and algorithms. *SIAM Review 41*(4), 637 – 676.

Dubourg, V. (2011). *Adaptive surrogate models for reliability analysis and reliability-based design optimization*. Ph. D. thesis, Université Blaise Pascal, Clermont-Ferrand, France.

Dubourg, V., J.-M. Bourinet, B. Sudret, and M. Cazuguel (2011). Reliability-based design optimization of an imperfect submarine pressure hull. In M. Faber (Ed.), *Proc. 11th Int. Conf. on Applications of Stat. and Prob. in Civil Engineering (ICASP11), Zurich, Switzerland*.

Dubourg, V., B. Sudret, and J.-M. Bourinet (2011). Reliability-based design optimization using kriging and subset simulation. *Struct. Multidisc. Optim. 44*(5), 673–690.

Dubrule, O. (1983). Cross validation of Kriging in a unique neighbourhood. *Math. Geol. 15*(6), 687–698.

Durrande, N., D. Ginsbourger, and O. Roustant (2012). Additive covariance kernels for high-dimensional Gaussian process modeling. *Annales de la Faculté de Sciences de Toulouse Tome 21 3*, 481–499.

Durrande, N., D. Ginsbourger, and O. Roustant (2013). ANOVA kernels and RKHS of zero mean function for model-based sensitivity analysis. *J. Multivariate Anal. 115*, 57–67.

Duvenaud, D., H. Nickisch, and C. E. Rasmussen (2012). Additive Gaussian processes. *Adv. Neur. In. 25*, 1–8.

Echard, B. (2012). *Évaluation par krigeage de la fiabilité des structures sollicitées en fatigue*. Ph. D. thesis, Université Blaise Pascal - Clermont II.

Echard, B., N. Gayton, and M. Lemaire (2011). AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Struct. Saf. 33*(2), 145–154.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *Annals of Statistics 32*, 407–499.

Faure, H. (1982). Discrepance de suites associées à un système de numération (en dimension s). *Acta Arith. 41*, 337 – 351.

Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal 13*, 317–322.

Furnival, G. and R. Wilson (1974). Regression by leaps and bounds. *Technometrics 16*, 499 – 511.

Geisser, S. (1975). The predictive sample reuse method with applications. *J. Am. Stat. Assoc. 70*, 320 – 328.

Ghanem, R. and P. Spanos (2003). *Stochastic Finite Elements : A Spectral Approach*. Courier Dover Publications.

Ghiocel, D. and R. Ghanem (2002). Stochastic finite element analysis of seismic soil-structure interaction. *J. Eng. Mech. 128*, 66 – 77.

Ginsbourger, D., N. Durrande, and O. Roustant (2013). Kernels and designs for modelling invariant functions: From group invariance to additivity. Technical Report hal-00731657, INRIA.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of point in evaluating multi-dimensional integrals. *Numer. Math. 2*, 84 – 90.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The elements of statistical learning: Data mining, inference and prediction*. Springer, New York.

Hesterberg, T., N. Choi, L. Meier, and C. Fraley (2008). Least angle and L_1 penalized regression: A review. *Statistics Surveys 2*, 61 – 93.

Heydayat, A., N. Sloane, and J. Stufken (1999). *Orthogonal Arrays - Theory and Applications*. Springer Series in Statistics.

Kaymaz, I. (2005). Application of Kriging method to structural reliability problems. *Struct. Saf. 27*(2), 133–151.

Keese, A. and H.-G. Matthies (2005). Hierarchical parallelisation for the solution of stochastic finite element equations. *Computers & Structures 83*, 1033–1047.

Krige, D. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *J. of the Chem., Metal. and Mining Soc. of South Africa 52*(6), 119–139.

Le Maître, O., M. Reagan, H. Najm, R. Ghanem, and O. Knio (2002). A stochastic projection method for fluid flow – II. Random process. *J. Comput. Phys. 181*, 9–44.

Lophaven, S., H. Nielsen, and J. Søndergaard (2002). *DACE, A Matlab Kriging Toolbox*. Technical University of Denmark.

Marrel, A., B. Iooss, F. Van Dorpe, and E. Volkova (2008). An efficient methodology for modeling complex computer codes with Gaussian processes. *Comput. Stat. Data Anal. 52*, 4731–4744.

McKay, M. D., R. J. Beckman, and W. J. Conover (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics 2*, 239–245.

Morris, M. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics 33*(2), 161–174.

Niederreiter, H. (1988). Low-discrepancy and low-dispersion sequences. *J. Number Theory 30*, 51–70.

Oakley, J. and A. O'Hagan (2004). Probabilistic sensitivity analysis of complex models: a Bayesian approach. *J. Royal Stat. Soc., Series B 66*, 751–769.

Rackwitz, R. (2001). Reliability anaylsis - a review and some perspectives. *Struct. Saf. 23*(4), 365 – 395.

Rasmussen, C. and C. Williams (2006). *Gaussian processes for machine learning* (Internet ed.). Adaptive computation and machine learning. Cambridge, Massachusetts: MIT Press.

Roustant, O., D. Ginsbourger, and Y. Deville (2012). DiceKriging, DiceOptim: Two R package for the analysis of computer experiments by Kriging-based metamodeling and optimisation. *J. Stat. Softw. 51*, 1–55.

Roustant, O., D. Ginsbourger, and Y. Deville (2013). *Package DiceKriging - Kriging methods for computer experiments*.

Santner, T., B. Williams, and W. Notz (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

Saporta, G. (2006). *Probabilités, analyse des données et statistique* (2nd ed.). Editions Technip.

Shanno, D. F. (1970). Conditioning of quasi-Newton methods for function minimization. *Math. Comput. 24*, 647–656.

Sobol, L. M. (1967). The distribution of point in a cube and the approximate evaluation of integrals. *USSR. Comput. Math. Math. Phys. 7*(4), 86 – 112.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *J. Royal Stat. Soc., Series B 36*, 111–147.

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Sys. Safety 93*, 964–979.

Sudret, B. (2012). Meta-models for structural reliability and uncertainty quantification. In K. Phoon, M. Beer, S. Quek, and S. Pang (Eds.), *Proc. 5th Asian-Pacific Symp. Struct. Reliab. (APSSRA'2012), Singapore*, pp. 53–76. Keynote lecture.

Suen, C. (2006). Some mixed orthogonal arrays obtained by orthogonal projection matrices. *J. Stat. Plan. Infer. 137*, 1704–1710.

Suen, C. and W. F. Kuhfeld (2005). On the construction of mixed orthogonal arrays of strength two. *J. Stat. Plan. Infer. 133*, 555–560.

Tibshirani, R. (1996). Regression shrinkage ans selection via LASSO. *J. Roy. Stat. Soc. B 58*, 267 – 288.

Xiu, D. (2009). Fast numerical methods for stochastic computations: a review. *Comm. Comput. Phys. 5*(2-4), 242–272.

Xiu, D. and J. Hesthaven (2005). High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput. 27*(3), 1118–1139.

Zhang, Y. (2007). Orthogonal arrays obtained by repeating-column difference matrices. *Discrete Math. 307*, 246–261.