



## A dedicated multiparametric strategy for the fast construction of a cokriging metamodel

Luc Laurent, Pierre-Alain Boucard, Bruno Soulier

### ► To cite this version:

Luc Laurent, Pierre-Alain Boucard, Bruno Soulier. A dedicated multiparametric strategy for the fast construction of a cokriging metamodel. *Computers & Structures*, 2013, KRETA, 124, pp.61 - 73. 10.1016/j.compstruc.2013.03.012 . hal-01376464

**HAL Id: hal-01376464**

**<https://hal.science/hal-01376464>**

Submitted on 16 Jan 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# A Dedicated Multiparametric Strategy for the Fast Construction of a Cokriging Metamodel

L. LAURENT<sup>a</sup>, P.-A. BOUCARD<sup>a</sup> AND B. SOULIER<sup>a</sup>

a. *LMT-Cachan, (ENS Cachan/CNRS/UPMC/PRES UniverSud Paris), 61 avenue du président Wilson,  
F-94235 Cachan Cedex, France  
{laurent,boucard,soulier}@lmt.ens-cachan.fr*

## Abstract

This paper deals with the building of a gradient-based metamodel using a dedicated strategy for solving structural assemblies problems. This work is the first part of a two-levels global optimization strategy. The general objective is to reduce computation costs; here, we focus on the costs which are associated with the generation of the metamodel. Our goal is achieved through the introduction of two main elements: what we call a “multiparametric strategy” based on the LATIN method, which reduces the computation costs when the parameters vary, and the use of a cokriging metamodel taking gradients into account. Several examples illustrate the efficiency of these two elements.

**Keywords:** metamodel, cokriging, multiparametric strategy, LATIN method, assemblies, two-levels optimization.

## 1. Introduction

Structural optimization involves two very different domains of expertise: numerical simulation and optimization. Of the two, simulation is the only one which generates high computation costs. The main difficulty in the context of optimization is that the optimizer often requires a large number of simulations to locate the optimum of an objective function. (In contact and friction problems, these simulations are nonlinear and, thus, even more costly.) Therefore, new methods for the resolution of complex optimization problems are needed. In this context we proposed to use a two-levels model optimization. Here, we focus on the first level. Thus, this paper deals with the aim of reducing the computational cost associated to the building of kriging metamodels. Therefore we introduced a gradient-based kriging metamodel and a MultiParametric Strategy.

In the first part of this paper, the context of two-levels model optimization is reviewed. In the second part, a specific mechanical resolution process is introduced and its performance is discussed in detail. The third part concerns the kriging class of metamodels. Finally, the last part presents the application of the coupled multiparametric-metamodel approach to a mechanical test case.

## 2. Two-level model optimization

The proposed multilevel model strategy is based on the approach developed in [1]. In our case, we consider the two-levels strategy illustrated in Figure 1. The first level consists in a metamodel defined

using a limited amount of data. The zones where the optimum can be located are determined using, for example, a genetic algorithm. This information is transferred to the second level, where a precise search of the minimum is carried out. Therefore, the second level consists in an optimization process based on the full mechanical model. Two elements contribute to the computation cost: the construction of the metamodel from the responses of the full mechanical model, and the direct optimization based on the full mechanical model. Both elements are involved both levels of the two-levels model strategy. Therefore, the strategy proposed here leads to a reduction in the computation cost of both phases.

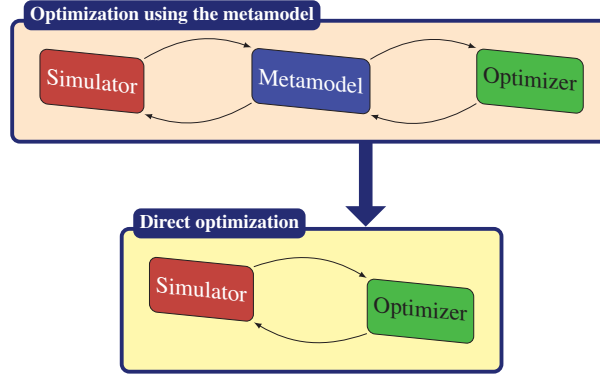


Figure 1: The multilevel optimization

The first step of our two-level optimization is a classical approach that is often designated as surrogate-based optimization and, overall, our work fits in the context of parametric optimization. More details on the use of metamodels in the context of parametric optimization can be found in [2, 3, 4, 5].

The scope of this paper is limited to a study of the cost associated with the generation of the metamodel.

Many types of metamodels can be used to find an approximate solution: polynomial regression [6], methods based on neural networks [7], radial basis functions [8], etc... We chose to use a particular category of metamodels called kriging approximations and, more precisely, a cokriging metamodel using derivatives. These approximations are presented in Section 4.

### 3. The multiparametric strategy

This chapter introduces the dedicated strategy called multiparametric strategy for solving problems on structural assemblies. Due to a judicious reinitialisation of the LATIN algorithm - described in Sect. 3.2 - multiparametric strategy enables one to reduce significantly the computation cost. The multiparametric strategy is presented in Sect 3.3.1 and example of 2D mechanical benchmark with two, three and four design variables is used for showing the performance of the multiparametric strategy (See Sect. 3.3.3).

#### 3.1. The context

We are considering assemblies of linear elastic structures under the assumption of small perturbations. The only nonlinearities are considered to be due to contact or friction between parts. The strategy we use to solve this type of problem, based on twenty years of development at LMT-Cachan, relies on an iterative algorithm introduced by P. Ladev  ve [9, 10].

In the context of assemblies, this strategy remains on three main points:

- the structure being studied is divided into *substructures* and *interfaces*. In the context of assembly problem, the natural way is to considering *substructures* as the parts of the assembly and *interfaces* as the contact zones among parts. Moreover a mixed domain decomposition is considered (force and displacement are unknowns at the interfaces)

- a dedicated iterative algorithm is used to solve the mechanical problem;
- the operators of the method remain constant and depend neither on the loading nor on the parameters (friction coefficient, gap) of the interface.

### 3.2. The resolution process

The resolution strategy, known as the LATIN (LArge Time INcrement) approach [10], consists of alternative resolutions of two groups of equations. The first group contains the *local* equations  $\Gamma$  (which may be nonlinear) related to the interfaces, and the second group contains the *linear* equations  $\mathbf{A}_d$  related to the substructures. The iterations between the two groups are handled through the use of search directions which are parameters of the method. This resolution process, shown in Figure 2, leads to the solution defined as the intersection of spaces  $\Gamma$  and  $\mathbf{A}_d$ .

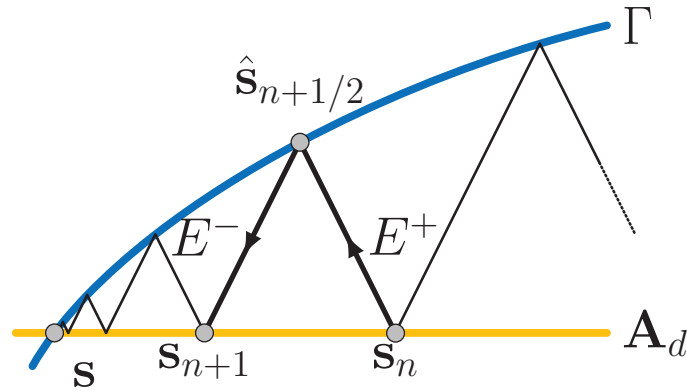


Figure 2: Schematic representation of the LATIN process

Friction and contact problems along with their specific approaches and laws [11] are part of equation group  $\Gamma$ . The schematic *linear* representation of  $\mathbf{A}_d$  is due to the *linear* behavior of the substructures. In the context of elastic assemblies, the boundary conditions (forces  $\mathbf{F}_i$  and displacements  $\mathbf{W}_i$ ) at each interface  $i$  between two substructures are sufficient to define the solution: the internal behavior is the result of a classical elastic problem. Thus, an approximate solution is described entirely by  $\mathbf{s} = \bigcup_{\text{all the interfaces}} (\mathbf{W}_i, \mathbf{F}_i)$ .

Readers can found additional details in [9, 10] for the LATIN method and in [11, 12] for its application to assemblies.

### 3.3. A method for reducing computation costs

A parametric optimization involves many calculations, each carried out with a different set of the parameters of the problem (design variables such as friction coefficients, preloads, gaps, etc...). This leads to the resolution of many similar problems in the sense that only the parameters of the problems vary. Therefore, it is essential to find a method to accelerate these calculations. The method we use, called the Multiparametric Strategy (MPS), was introduced in [13, 14], then studied and applied to various sample problems in [15, 16, 17].

#### 3.3.1. Principle of the method

The main idea of the multiparametric strategy is to take advantage of a feature of the LATIN algorithm: at each iteration, the solver yields an approximate solution over the whole loading path and at all points in the structure. Then, if a new calculation associated with other values of the parameters is requested, the algorithm can be reinitialized using a previously converged solution. Here, we assume

that the only parameters which are different are the interface parameters, so space  $\Gamma$  alone is affected by the change.

Each new converged solution enriches a database of possible restart solutions. Figure 3 illustrates the use of the multiparametric strategy to obtain an approximate solution starting from a previous calculation: the schematic representation shows that the required number of iterations to carry out the converged solution  $s_1$  is bigger than the required number of iterations to obtain the other converged solutions  $s_2$  or  $s_3$ . Moreover, in order to obtain the solution  $s_3$ , the LATIN algorithm can be reinitialized with one of the two available converged solutions  $s_1$  or  $s_2$ . We observe that the benefit depends on the choice of the starting point. Here, we will use only a "closest point" strategy: for each new requested solution the LATIN algorithm is reinitialized with the converged solution associated with the set of design parameters which is closest to the one being considered. Some indications for choosing the best initialization strategy were given in [18].

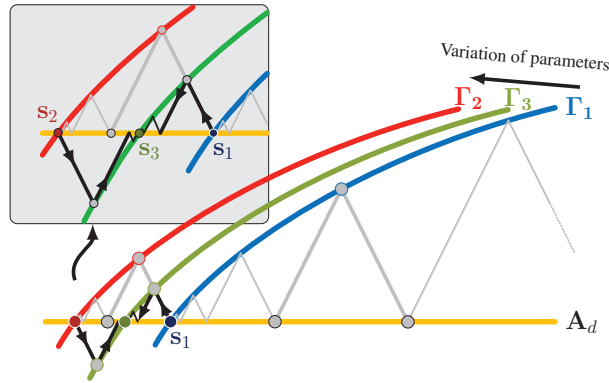


Figure 3: The multiparametric strategy using the LATIN algorithm

With this approach, the solution converges in fewer iterations of the algorithm and, therefore, in less time.

### 3.3.2. One Reduced Order Model: the Proper Generalized Decomposition (PGD)

Each converged solution used for reinitializing the LATIN algorithm can be considered as a reduced order model (ROM) of the solution in the sense that, as we deal here with elastic assemblies, the whole solution can be only described by the boundary conditions (more details can be found in [13]). In case of problems with nonlinear material behaviour for example this idea must be extended with the use of a PGD basis as ROM.

The Proper Generalized Decomposition is initially based on the space-time separated representations as a part of the LATIN method which was designed to deal with non-linearities such as plasticity or viscoplasticity. Then many developments have been achieved during the past three decades. Readers can find details in [19, 20] for the developments of the PGD approximations and applications. Recently the PGD was used on parametric models and this approach is initiated the use of the PGD to deal with parametric problems such as parametric optimization problems, inverse identification problems or real time simulation. In order to achieve these parametric problems the classical time-space separated approximation presented by the Equation 1 can be rewritten as presented by Equation 2 where  $k$  denoted a parameter of the model. The PGD can also be used as space-parameter decomposition as presented by Equation 3.

$$u(\mathbf{x}, t) \approx \sum_{i=1}^Q X_i(\mathbf{x}) \cdot T_i(t) \quad (1)$$

$$u(\mathbf{x}, t, k) \approx \sum_{i=1}^Q X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k) \quad (2)$$

$$u(\mathbf{x}, k) \approx \sum_{i=1}^Q X_i(\mathbf{x}) \cdot K_i(k) \quad (3)$$

Such a decomposition allows to reduce significantly the CPU time associated to the computation of the responses of the model on requested values of the time, space and/or parameter. Examples of the use of the parametric modelling can be found in [21, 22, 23, 24, 25]. PGD was also carried out in the context of Multiparametric Strategy [26]. Finally the PGD was also used to build response surface models [27]. Actually, whether it is for optimization, the building of a response surface or, more generally, the building of a metamodel, it requires values of an objective function obtained for different values of the parameters of the model. The PGD enables to reduce significantly the computation cost associated to these required values.

### 3.3.3. Illustration of the performance of the method

#### A two design variables example.

The example considered here is a quasi-static academic problem which was presented in [16]. Figure 4 shows the geometry of the problem, which consists of three square parts ( $h = 50mm$ , Young's modulus  $E = 2 \cdot 10^5 MPa$  and Poisson's coefficient  $\nu = 0.3$ ) which are in contact with friction. Each part was represented by a single substructure discretized into  $24 \times 24$  bilinear quadrangles. The parametric study consisted in varying the friction coefficients  $\mu_1$  and  $\mu_2$  of the two contact interfaces. The loading consisted of two stages: first, a progressive vertical pressure  $P_1$  up to a maximum of  $50MPa$  applied at the top of substructure  $\Omega_3$  (the preloading stage); then, a progressive horizontal load from 0 to  $30MPa$  applied to substructure  $\Omega_2$ .

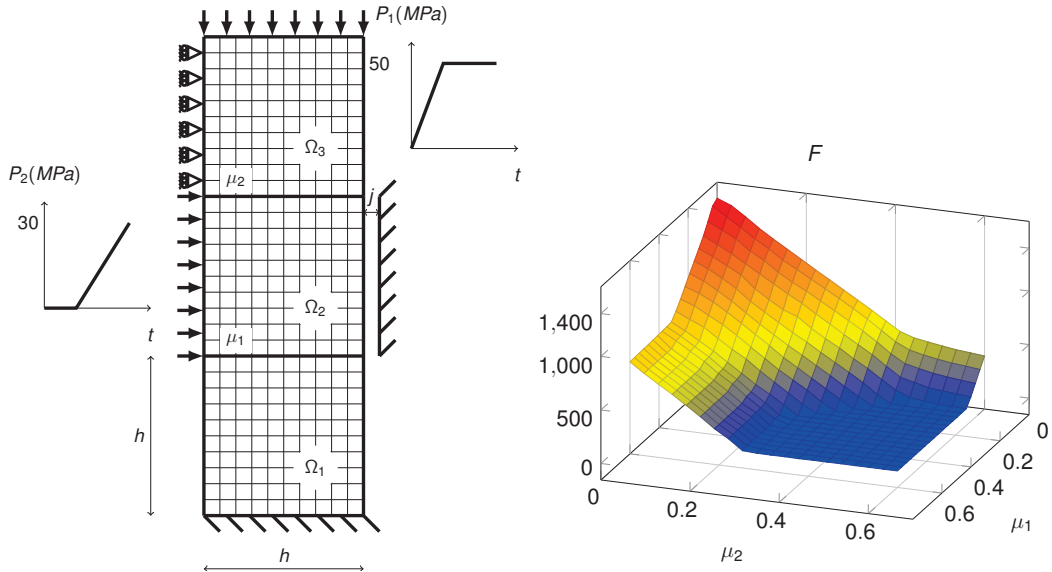


Figure 4: The geometry of the problem and the response surface

In this test case, variations of the friction coefficients  $\mu_1$  and  $\mu_2$  between 0 and 0.6 were considered, and the function studied was the reaction force on the rigid wall. Figure 4 shows the response surface of this function obtained with  $18 \times 18$  values of the friction coefficients.

In order to illustrate the performance of the multiparametric strategy, the two-variable design space was sampled with a  $4 \times 4$  regular grid. For each sample, the force and gradients were calculated (using a classical finite difference method for the gradients). Table 1 summarizes the characteristics of these calculations:

The gain obtained with our method (compared to a classical calculation without the multiparametric strategy) was estimated using the following expression:

$$\text{Gain} = \frac{\text{Number of calculations} \times \text{CPU time of the first calculation}}{\text{CPU time using the multiparametric strategy}} \quad (4)$$

We considered that the time of each calculation without the multiparametric strategy was almost constant and equal to the time of the first calculation (8.2s).

	Eval.	Eval. gradients	Total
Number of evaluations	16	$2 \times 16$	48
Total CPU time	42.3s	58.1s	100.4s
Avg CPU time per calc.	2.7s	1.8s	2.1s
Min CPU time	0.2s	0.2s	0.2s
Max CPU time	8.2s	6s	8.2s
Nb. of iterations	1,896	2,580	4,476
Avg nb. of iterations (per calc.)	118.5	80.6	93.3
Min nb. of iterations	5	5	5
Max nb. of iterations	381	260	381
Gain	3.11	4.54	3.94

Table 1: The mechanical calculations using the multiparametric strategy

The results presented in Table 1 show that the solver can reuse a previously converged solution to accelerate the resolution of similar problems. The most remarkable point is the gain obtained in the evaluation of the gradients. These were calculated using a finite difference method: each gradient required the value of the response at 3 points (a sample point plus 2 points obtained by very small variations of the parameters). The results show that for the same number of evaluations the cost was less when the gradients were calculated along with the responses than when the responses alone were calculated: in our example, 16 evaluations took 42.3s whereas 16 gradients (32 evaluations) took 58s.

#### *A four design variables example.*

In this paragraph we proposed a modified version of the previous two-variables example. By modifying the substructure  $\Omega_1$  and replace it with two new substructures, a four-variables is introduced. Figure 5 shows the geometry of the problem. The material characteristics and the loading are retained and two new interfaces are introduced: a gap between substructures  $\Omega_1$  and  $\Omega_4$  and a new interface with friction between substructures  $\Omega_2$  and  $\Omega_4$  are introduced. In this context, two new design variables are added to the two existing design variables: thus we consider three friction coefficients  $\mu_1$ ,  $\mu_2$  and  $\mu_3$  (between 0 and 0.6) and one gap  $j_1$  (between  $-28\mu\text{m}$  and  $48\mu\text{m}$ ).

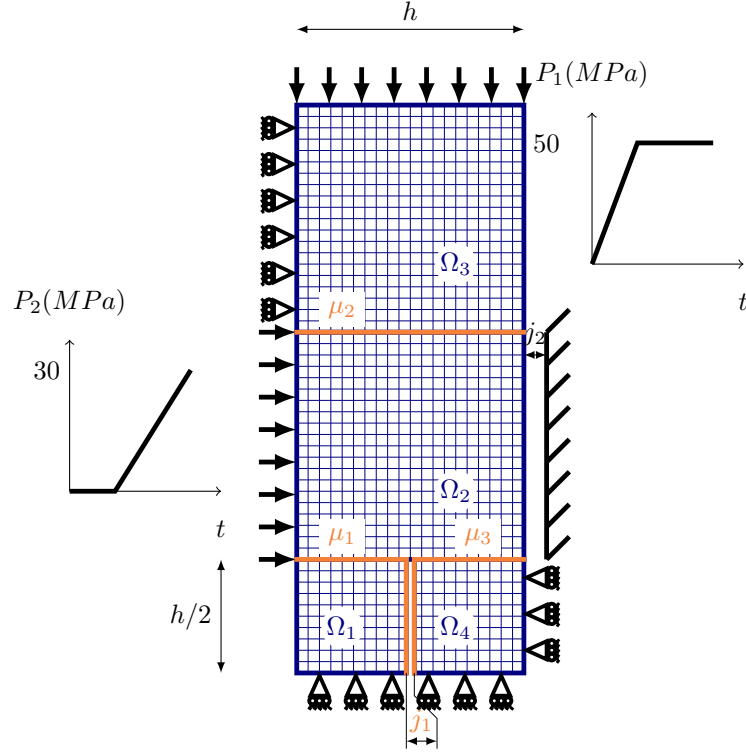


Figure 5: The geometry of the problem of the four-variables example

Figure 6 shows the computational cost and the gain of the MPS associated to the computation of the responses on few sets of values of the design parameters (from 5 to 50 sample points are considered). Although we use a first-order finite difference method to compute the gradients and the fact that  $n_d$  responses (in addition to the first calculation) of the objective function are needed to obtain the gradient on one sample point, the required time to evaluate the gradients is the same than the time required to evaluate all responses. For this example we obtain a gain around 4.5 for computing the responses and the gradients.

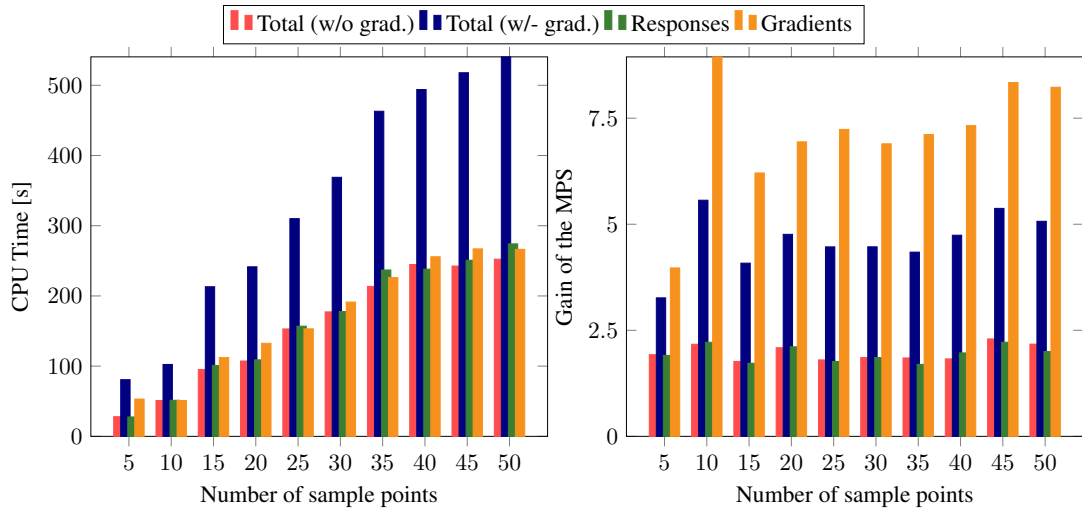


Figure 6: CPU time and gain of the MPS associated two different sets of sample points for the four-variables test-case



Thus, the MPS is particularly efficient in calculating the gradients, thanks to the ability of the method to reduce the computation time for sample points which are close together. This led to our decision to use a dedicated gradients metamodel.

#### 4. The cokriging metamodel

This type of metamodel is similar to the kriging metamodel proposed by [28] and developed by [29, 30]. The cokriging strategy stems from multivariate geostatistics [31]. In our case, the cokriging metamodel is built using not only the responses, but also their gradients [32, 33, 34].

##### 4.1. Notations

In this section, we will use the following notations:  $\mathbf{x}^{(i)}$  denotes a point in the design space  $\mathcal{D}$  ( $\mathbf{x}^{(i)}$  is one of the  $n_s$  sample points, while  $\mathbf{x}^{(0)}$  is a non-sample point);  $x_i$  denotes the  $i^{th}$  coordinate of a point  $\mathbf{x}$ ;  $Y(\mathbf{x})$  and  $\tilde{Y}(\mathbf{x})$  denote respectively the response of the analytical function (or the response of the mechanical model) and the approximate response given by the metamodel; finally,  $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$  is a correlation function.

##### 4.2. Principle

The model can be viewed as the sum of two components: a linear model (which represents the trend of the data) and a departure from the linear model (which represents the fluctuations around the trend).

$$\tilde{Y}(\mathbf{x}^{(0)}) = \mu(\mathbf{x}^{(0)}) + Z(\mathbf{x}^{(0)}), \quad \forall \mathbf{x}^{(0)} \in \mathcal{D} \quad (5)$$

$$\mathbb{E}(Z) = 0, \quad Cov[Z(\mathbf{x}^{(i)}), Z(\mathbf{x}^{(j)})] \neq 0, \quad \forall (\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \in \mathcal{D}^2 \quad (6)$$

where  $\mathbb{E}$  and  $Cov$  denote the classical statistical expected value and covariance.

The main idea is to consider a covariance relation between the discrete responses of the deterministic function. This covariance depends only on the distances among the samples. Thus the departure is represented by a zero-mean, second-order stationary process. (The mean and the variance are constant, with a correlation depending on the distance.) Depending on the definition of the function  $\mu$ , one can build different types of kriging or cokriging metamodels (simple kriging, where  $\mu$  is the average of the response at sample design points; ordinary kriging, where  $\mu$  is an unknown constant; or universal kriging, where  $\mu$  is a polynomial function).

Function  $Z$  has a zero expected value and its covariance structure is a function of a generalized distance among the sample responses. The covariance structure can be written as:

$$\forall (\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \in \mathcal{D}^2, \quad cov[Z(\mathbf{x}^{(i)}), Z(\mathbf{x}^{(j)})] = \sigma^2 R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sigma^2 c_{ij} \quad (7)$$

In our case, the correlation function chosen is a Gaussian function or a Matérn function[35, 36].

In the case of a cokriging metamodel, additional covariance relations involving the different variables are introduced [32]. In our case, the primary variables are the evaluations of the function being studied, and the secondary variables are the gradients:

$$Cov\left[\frac{\partial Z}{\partial x_k}(\mathbf{x}^{(i)}), Z(\mathbf{x}^{(j)})\right] = -\sigma^2 \frac{\partial R}{\partial x_k}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sigma^2 c_{kij} \quad (8)$$

$$Cov\left[Z(\mathbf{x}^{(i)}), \frac{\partial Z}{\partial x_k}(\mathbf{x}^{(j)})\right] = -\sigma^2 \frac{\partial R}{\partial x_k}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sigma^2 c_{kij} \quad (9)$$

$$Cov\left[\frac{\partial Z}{\partial x_k}(\mathbf{x}^{(i)}), \frac{\partial Z}{\partial x_l}(\mathbf{x}^{(j)})\right] = -\sigma^2 \frac{\partial^2 R}{\partial x_k \partial x_l}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sigma^2 c_{kilj} \quad (10)$$

With this relation, one can also build the kriging or cokriging metamodel by considering the following linear predictors of the non-sample point  $\mathbf{x}^{(0)}$ :

Kriging	$\tilde{Y}(\mathbf{x}^{(0)}) = \sum_{i=1}^{n_s} \lambda_i(\mathbf{x}^{(0)}) Y(\mathbf{x}^{(i)})$
Cokriging	$\tilde{Y}(\mathbf{x}^{(0)}) = \sum_{i=1}^{n_s} \lambda_{0i}(\mathbf{x}^{(0)}) Y(\mathbf{x}^{(i)}) + \sum_{j=1}^{n_d} \sum_{i=1}^{n_s} \lambda_{ij}(\mathbf{x}^{(0)}) \frac{\partial Y}{\partial x_j}(\mathbf{x}^{(i)})$

Thus, the determination of the Best Linear Unbiased Predictor (BLUP) leads to the  $\lambda$ 's which, for a kriging metamodel, minimize the *Mean Square Error*:

$$\text{MSE}[\tilde{Y}(\mathbf{x}^{(0)})] = \mathbb{E} \left[ \left( \sum_{i=1}^{n_s} \lambda_i(\mathbf{x}^{(0)}) Y_i - Y(\mathbf{x}^{(0)}) \right)^2 \right] \quad (11)$$

subject to the unbiasedness condition:

$$\mathbb{E} \left[ \sum_{i=1}^{n_s} \lambda_i(\mathbf{x}^{(0)}) Y(\mathbf{x}^{(i)}) \right] = \mathbb{E} [Y(\mathbf{x}^{(0)})] \quad (12)$$

The construction of a cokriging metamodel follows the same process.

In the universal kriging case, the best linear predictor of  $Y(\mathbf{x}^{(0)})$  can be written in matrix form [37] as:

$$\tilde{Y}(\mathbf{x}^{(0)}) = \mathbf{x}_0^T \hat{\boldsymbol{\beta}} + \mathbf{c}_0^T \mathbf{C}^{-1} (\mathbf{Y}_s - \mathbf{X} \hat{\boldsymbol{\beta}}) \quad (13)$$

where

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\mathbf{X}^T \mathbf{C}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{C}^{-1} \mathbf{Y}_s \\ \mathbf{x}_0 &= [f_1(\mathbf{x}^{(0)}) \quad f_2(\mathbf{x}^{(0)}) \quad \cdots \quad f_k(\mathbf{x}^{(0)})] \\ \mathbf{c}_0 &= [c_{01} \quad c_{02} \quad \cdots \quad c_{0n_s}] \\ (\mathbf{X})_{ij} &= f_j(\mathbf{x}^{(i)}) \\ (\mathbf{C})_{ij} &= c_{ij} \end{aligned}$$

The regressor vector  $\hat{\boldsymbol{\beta}}$  is the generalized least-squares approximation of  $\boldsymbol{\beta}$ . Thus, the first part of Equation 13 is the generalized least-squares prediction at point  $\mathbf{x}^{(0)}$ . The second part can be viewed as a correction of the generalized least-squares response surface to obtain the interpolating kriging model.

The formulation of the cokriging metamodel is very similar to that of the kriging metamodel. For simplicity's sake, following the idea in [34], we propose to use only an ordinary cokriging metamodel:

$$\tilde{Y}(\mathbf{x}^{(0)}) = \hat{\boldsymbol{\beta}}_c + \mathbf{c}_{c0}^T \mathbf{C}_c^{-1} (\mathbf{Y}_{sc} - \mathbf{X}_c \hat{\boldsymbol{\beta}}_c) \quad (14)$$

where

$$\begin{aligned} \hat{\boldsymbol{\beta}}_c &= (\mathbf{X}_c^T \mathbf{C}_c^{-1} \mathbf{X}_c)^{-1} \mathbf{X}_c^T \mathbf{C}_c^{-1} \mathbf{Y}_{sc} \\ \mathbf{c}_{c0} &= [c_{01} \quad c_{02} \quad \cdots \quad c_{0n_s} \quad c_{110} \quad c_{210} \quad \cdots \quad c_{n_d n_s 0}] \\ \mathbf{X}_c &= [1, 1, \dots, 1, 0, 0, \dots, 0]^T \\ \mathbf{C}_c &= \begin{bmatrix} \mathbf{C} & \mathbf{C}_d \\ \mathbf{C}_d^T & \mathbf{C}_{dd} \end{bmatrix} \end{aligned}$$

and where

$$\begin{aligned}
(\mathbf{C})_{ij} &= c_{ij} \\
(\mathbf{C}_d)_{ij} &= \begin{bmatrix} c_{111} & c_{211} & \cdots & c_{n_d 11} & c_{121} & \cdots & c_{n_d n_s} \\ c_{112} & c_{212} & \cdots & c_{n_d 12} & c_{122} & \cdots & c_{n_d n_s 2} \\ c_{113} & c_{213} & \cdots & & & & \\ \vdots & & \ddots & & & & \vdots \\ c_{11 n_s} & \cdots & & & & & c_{n_d n_s n_s} \end{bmatrix} \\
(\mathbf{C}_{dd})_{ij} &= \begin{bmatrix} c_{1111} & c_{1121} & \cdots & c_{11 n_d 1} & c_{1112} & \cdots & c_{11 n_d n_s} \\ c_{2111} & c_{2121} & \cdots & & & & \vdots \\ \vdots & & \ddots & & & & \\ & & & c_{n_d 1 n_d 1} & c_{n_d 112} & \cdots & \\ & & & & \ddots & & \\ c_{n_d n_s 11} & \cdots & & & & & c_{n_d n_s n_d n_s} \end{bmatrix}
\end{aligned}$$

In the kriging case, vector  $\mathbf{Y}_s$  contains only the responses of the function at the sample point, whereas in the cokriging case vector  $\mathbf{Y}_{s_c}$  contains both the responses and the gradients.

With these formulations, the models can supply approximate responses of the actual function at all the points in the design space. In our case, we consider the response of the function to be deterministic and we obtain both of the interpolating models. This type of metamodel has other advantages: for example, it provides statistical information (the expected value and the variance of the process). Due to the use of the unbiasedness condition, the expected value of  $\tilde{Y}$  is given by the trend model  $\mu$  and the mean square error of  $\tilde{Y}$  [37]:

$$\text{MSE} \left[ \tilde{Y} \left( \mathbf{x}^{(0)} \right) \right] = \sigma^2 \left[ 1 - [\mathbf{x}_0 \quad \mathbf{c}_0] \begin{bmatrix} \mathbf{0} & \mathbf{X}^T \\ \mathbf{X} & \mathbf{C} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{c}_0 \end{bmatrix} \right] \quad (15)$$

This relation also holds in the cases of simple or ordinary kriging or cokriging models, provided the appropriate forms of vectors and matrices  $\mathbf{x}_0$ ,  $\mathbf{c}_0$ ,  $\mathbf{X}$  and  $\mathbf{C}$  are used.

#### 4.3. Estimation of the parameters

The model's parameters (such as the characteristic correlation length scale  $l$ , the variance  $\sigma$  of the random process  $Z$  or the regression coefficients) can be determined by maximizing the likelihood [38]. In our case, we use this technique to determine  $l$  and  $\sigma$ :

$$(l, \sigma^2) = \arg \min_{l, \sigma^2} \left[ -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2} \log(|\mathbf{C}(l)|) - \frac{1}{2\sigma^2} (\mathbf{Y}_s - \mathbf{X}\hat{\beta})^T \mathbf{C}(l)^{-1} (\mathbf{Y}_s - \mathbf{X}\hat{\beta}) \right] \quad (16)$$

with

$$N = \begin{cases} n_s & \text{for kriging} \\ n_s(n_d + 1) & \text{for cokriging} \end{cases}$$

The variance  $\sigma^2$  can be determined analytically through the derivation of the likelihood function:

$$\hat{\sigma}^2 = \frac{1}{N} (\mathbf{Y}_s - \mathbf{X}\hat{\beta})^T \mathbf{C}(l)^{-1} (\mathbf{Y}_s - \mathbf{X}\hat{\beta}) \quad (17)$$

One can also use an optimizer to determine the value of the correlation length scale. This method has some drawbacks [39, 40, 41]: in many cases with very few points, the log-likelihood is monotonous; often, the correlation matrix is affected by conditioning problems which make finding the minimum difficult. When such problems arise, one sets the parameters (in particular the correlation length scale) to fixed values.

#### 4.4. Analytical applications

In this section, we apply kriging and cokriging to one- and two-dimensional analytical test functions. We will use the abbreviations OK and OCK to designate respectively **Ordinary Kriging** and **Ordinary CoKriging**.

##### 4.4.1. The one-dimensional test function

First, we applied the two types of metamodels to an analytical function, chosen to be  $y(x) = \exp(-x/10) \cos(x) + x/10$ . We used 5 sample responses of the analytical function to build the OK metamodel, and an additional 5 sample derivatives to build the OCK metamodel. The correlation function was the Matérn function. The sample points were obtained using *Latin Hypercube Sampling*.

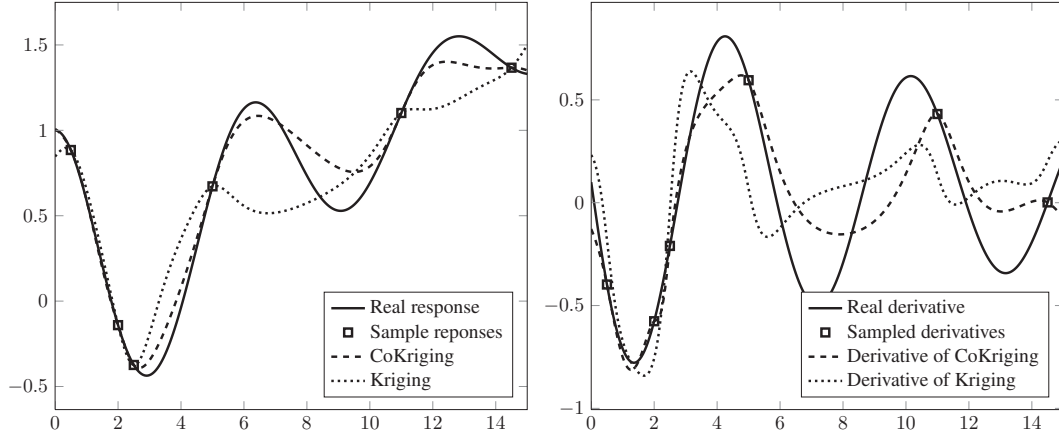


Figure 7: The ordinary kriging and cokriging metamodels and their derivatives

Figure 7 illustrates the capability of the cokriging metamodel to interpolate not only the values of the responses, but also the sample derivatives. In this example, the quality of the cokriging metamodel was better. This statement remained true as long as we worked with only a few points. We can also observe that for a relatively smooth function the kriging metamodel converges quickly toward a good approximation of the real function when the number of points becomes large enough.

The characteristics of the two metamodels are summarized in Table 2.

	$R^2$	RAAE	RMAE	$Q_1$	$Q_2$	$Q_3$
OK	0.4442	0.5036	1.726	0.4192	23.46	$7.796 \cdot 10^{-2}$
OCK	0.9719	0.1153	0.4087	$2.351 \cdot 10^{-2}$	1.186	$3.940 \cdot 10^{-3}$

Table 2: Characteristics of the two previous metamodels

Criteria RAAE and RMAE stand for *Relative Average Absolute Error* and *Relative Maximum Absolute Error*. Criteria  $Q_i$ , which compare the actual response and the responses of the metamodels at

$n_c$  points with  $n_c \gg n_s$ , were calculated as follows:

$$\begin{aligned}
Q_1 &= \sup_{i \in \{1, 2, \dots, n_c\}} e_i \\
Q_2 &= \sum_{i=1}^{n_c} e_i \\
Q_3 &= \frac{Q_2}{n_c}
\end{aligned}$$

where  $\forall i \in \{1, 2, \dots, n_c\}$ ,  $e_i = \frac{(Y(\mathbf{x}^{(i)}) - \tilde{Y}(\mathbf{x}^{(i)}))^2}{\sup_{j \in \{1, 2, \dots, n_c\}} Y(\mathbf{x}^{(j)})^2}$

Based on the statistical information from the kriging and cokriging metamodels, one can derive confidence intervals. The two diagrams of Figure 8 show the 95% Confidence Intervals (CIs) obtained with Expression 18. In these types of metamodels, the size of the confidence envelopes is determined mainly by the distance between each pair of neighboring points.

$$CI^\pm(\mathbf{x}) = \tilde{Y}(\mathbf{x}) \pm 2\sqrt{MSE[\tilde{Y}(\mathbf{x})]} \quad (18)$$

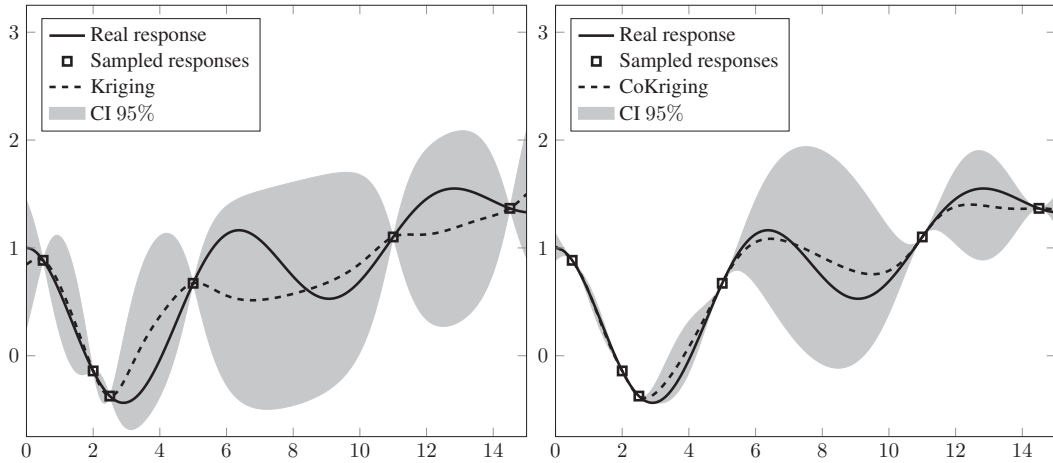


Figure 8: The confidence envelopes for the kriging and cokriging metamodels

#### 4.4.2. The two-dimensional test function

The same two types of metamodels were used to approximate an analytical function of two variables. In order to illustrate the performance of the cokriging metamodel, we chose a very bumpy function: the *six-hump camel back function* ( $\forall (x_1, x_2) \in [-3, 3] \times [-2, 2]$ ,  $f(x_1, x_2) = (4 - 2.1x_1^2 + x_1^4)x_2^2 + x_1x_2 + 4(x_2^2 - 1)x_2^2$ ). Figure 9 shows the response surface of this function. The two metamodels were constructed using 16 evaluations of the function for the kriging metamodel, and an additional 16 evaluations of the gradients for the cokriging metamodel. In both cases the correlation function was the Matérn function.

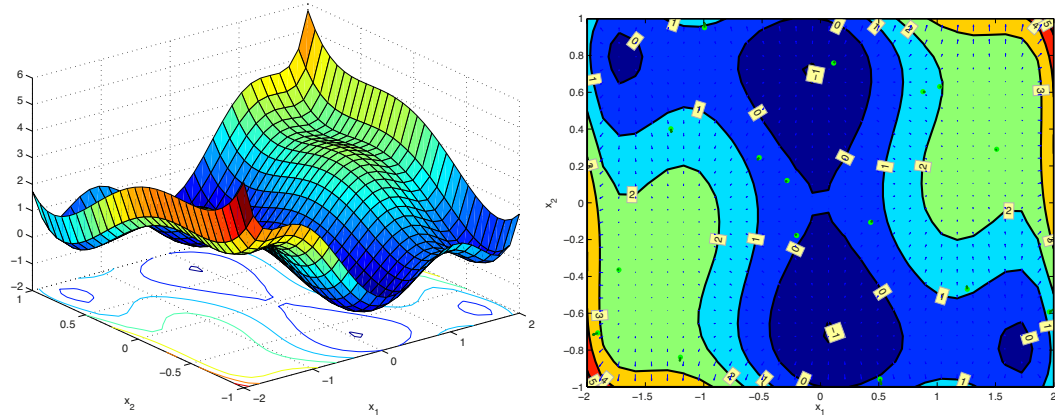


Figure 9: The actual function and its gradients (blue arrows)

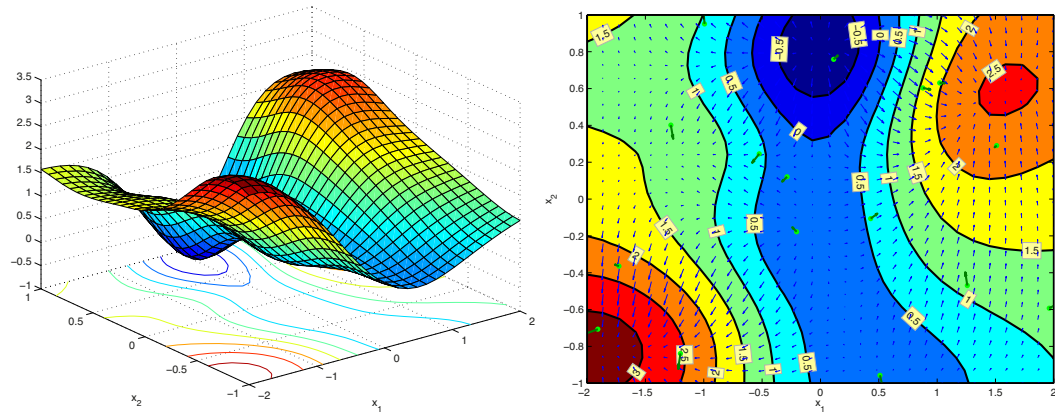


Figure 10: The kriging function and its gradients (blue arrows)

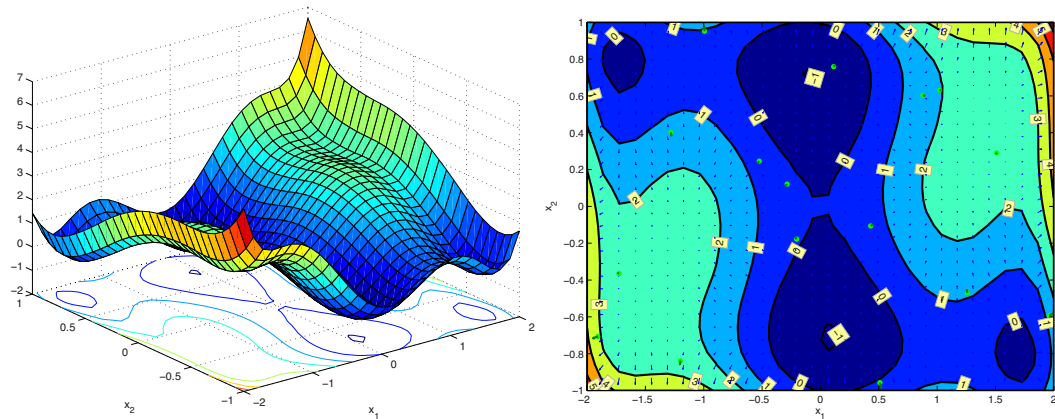


Figure 11: The cokriging function and its gradients (blue arrows)

The characteristics of these two metamodels are given in Table 3.

	$R^2$	RAAE	RMAE	$Q_1$	$Q_2$	$Q_3$
OK	0.6991	0.4042	2.845	0.3946	13.192	$1.466 \cdot 10^{-2}$
OCK	0.9958	$3.027 \cdot 10^{-2}$	$5.330 \cdot 10^{-1}$	$1.391 \cdot 10^{-2}$	0.1824	$2.027 \cdot 10^{-4}$

Table 3: Characteristics of the two previous metamodells (Figure 10-11)

For this 2D test function, the cokriging metamodell led to a relatively accurate approximation of the actual function using only a few sample points. Taking into account the gradients, we were able to develop more efficient approximate models. But for a problem involving the evaluation of mechanical responses the computation cost for the determination of  $n_s$  responses and  $n_s$  gradients is, of course, higher than that required to obtain  $n_s$  responses alone. (In the former case, due to the use of finite differences to obtain gradients, the construction of the metamodell requires  $3n_s$  evaluations of the mechanical model.)

Now let us take another approach: the idea is to build metamodells using the same number of evaluations. In the following case, we used the same six-hump function, but the kriging metamodell was constructed based on the responses at 27 sample points (Figure 12), while the cokriging metamodell was still constructed using the responses and the gradients at 9 sample points (Figure 13).

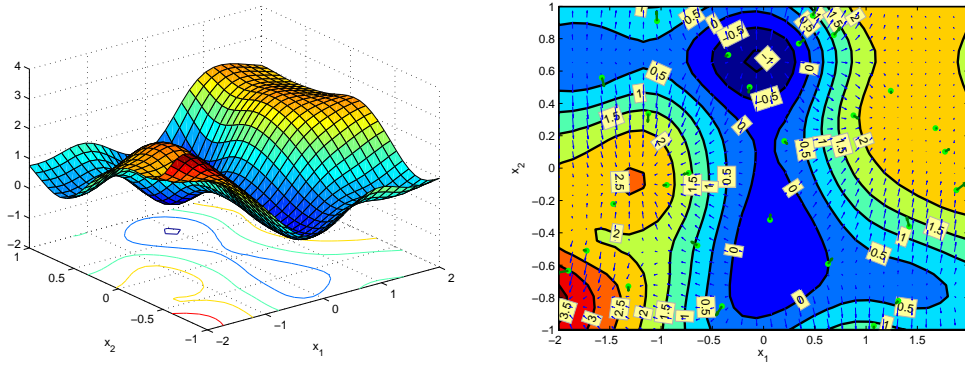


Figure 12: The kriging surface obtained with 27 responses (27 evaluations of the function)

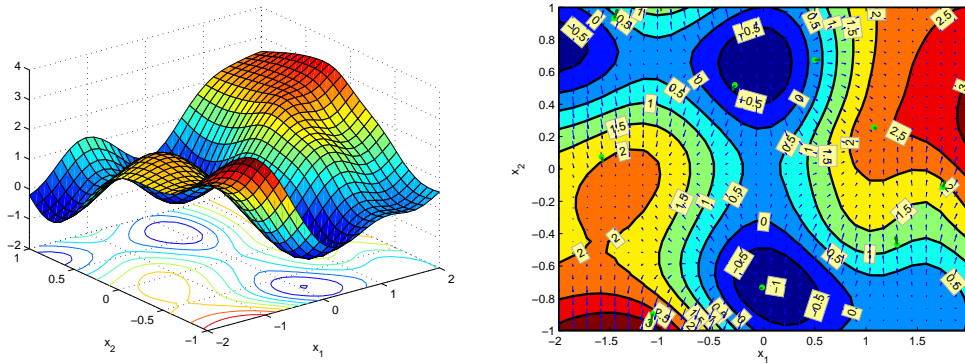


Figure 13: The cokriging surface obtained with 9 responses and 9 gradients (27 evaluations of the function)

	$R^2$	RAAE	RMAE	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>
OK	0.8175	0.2566	2.933	0.4195	8.001	$8.890 \cdot 10^{-3}$
OCK	0.8160	0.2580	2.541	0.3150	8.066	$8.9627 \cdot 10^{-3}$

Table 4: Characteristics of the two previous metamodells (Figure 12-13)

Table 4 shows that the quality of the prediction achieved with the two metamodells was similar, but a very important advantage of the cokriging metamodel was that it led to more zones where a minimum could be found than the kriging metamodel. If the problem involved the calculation of mechanical responses, the computation cost associated with the 9 points and their gradients, thanks to the multiparametric strategy and as shown in Table 1, would be much lower than that associated with the 27 points.

#### 4.4.3. The five- and ten-dimensional test functions

In this section, we consider two analytical functions defined for  $n_d$  variables:

- the *Rosenbrock function*, which is a smooth function, can be defined as follow:

$$\forall \mathbf{x} \in [-2, 2]^{n_d}, f(\mathbf{x}) = \sum_{i=1}^{n_d-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] \quad (19)$$

- the *Schwefel function*, which is a very irregular function, can be defined as follow:

$$\forall \mathbf{x} \in [-500, 500]^{n_d}, f(\mathbf{x}) = 418.9829 \times n_d - \sum_{i=1}^{n_d} [x_i \sin(\sqrt{|x_i|})] \quad (20)$$

Here, we only use these analytical function with 5 or 10 variables (*i.e.*  $n_d = 5$  or  $n_d = 10$ ). Using a Latin Hypercube Sampling 10 (for  $n_d = 5$ ) and 2 (for  $n_d = 10$ ) sets of sample points are generated for each number of sample points. Here we consider the number of sample points in the horizontal axis, thus for  $n_s$  sample points the kriging metamodel is built from  $n_s$  responses and the cokriging metamodels is built from  $n_s$  responses and  $n_s$  gradients (each gradient comprises  $n_d$  components). The mean of the criteria  $R^2$  and  $Q_3$  is calculated for each number of sample points: metamodells are built with responses (and gradients) for each set of sample points obtained for each number of sample points; then for each metamodel the quality criteria are computed and finally for each number of sample points we compute the means of the two criteria.

Figures 14 and 15 show the means of the quality criteria versus the number of sample points for the kriging and cokriging metamodels built using responses and gradients provided by the Rosenbrock and Schwefel functions with 5 variables. Figures 16 and 17 show the same results obtained with the ten-dimensional functions.



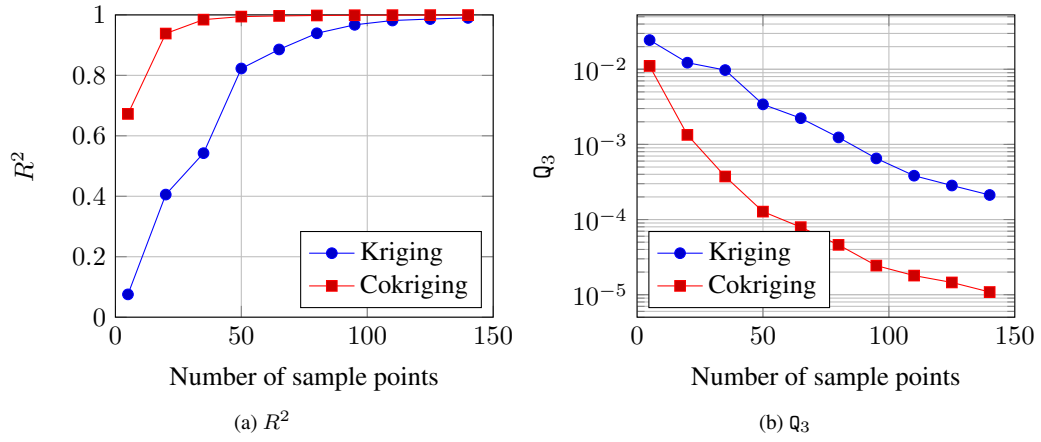


Figure 14: Quality of kriging and cokriging metamodels applied on five-dimensional Rosenbrock function

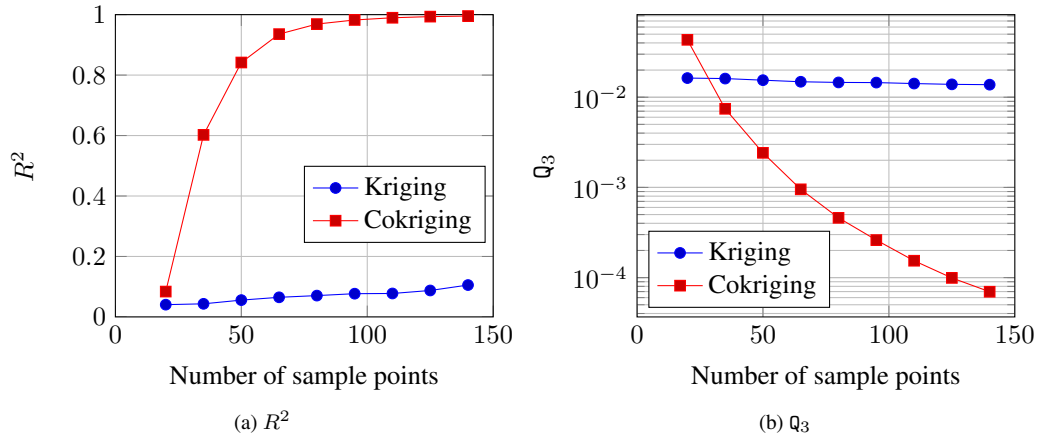


Figure 15: Quality of kriging and cokriging metamodels applied on five-dimensional Schwefel function

We can observe on the Figures 14 and 16 that the cokriging metamodel provides a better approximation than kriging metamodel. Due to the fact that the *Rosenbrock function* is a very smooth function, kriging reaches an acceptable quality of approximation. However on a very irregular function (See Fig. 15 and 17) only cokriging metamodel provides a good approximation of the analytical test function.

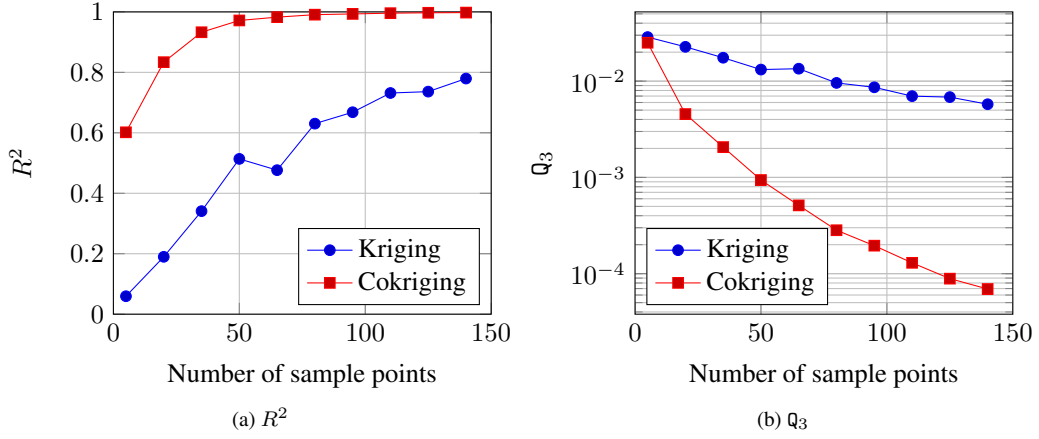


Figure 16: Quality of kriging and cokriging metamodels applied on ten-dimensional Rosenbrock function

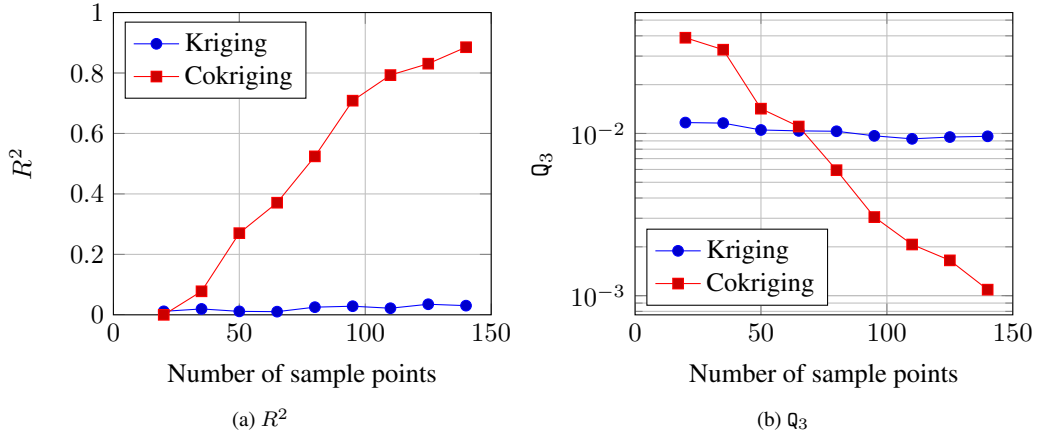


Figure 17: Quality of kriging and cokriging metamodels applied on ten-dimensional Schwefel function

Moreover, if we consider the case of the five-dimensional Schwefel function with the same total number of evaluations (*i.e.*, if we consider  $n_s$  responses and  $5 \times n_s$  gradients for the cokriging, we can use  $n_s + 5 \times n_s$  responses for the kriging), the quality is always better with the cokriging metamodel as soon as  $n_s$  is greater than 30. As we have shown before that the MPS is particularly efficient in calculating the gradients, coupling a cokriging metamodel with the MPS should lead to a very efficient strategy to build metamodels. This is what we illustrate in the next part.

## 5. Application to a mechanical problem

In this section, we present the construction of the two previous metamodels (kriging and cokriging) using the multiparametric strategy presented in the first part of this paper and the actual responses of a mechanical model. The test case being considered is the *three-squares example* presented in Page 5. We carried out two studies: one with a fixed number of mechanical calls, and the other with a fixed quality of the metamodels.

### 5.1. Case of a fixed number of mechanical calls

Both metamodels were constructed using 15 mechanical evaluations. The kriging metamodel was determined using 15 values of the force, and the cokriging metamodel was determined with 5 values of

the force and 5 gradients. The sample points were obtained through *Latin Hypercube Sampling*. Our calculations led to the following two response surfaces (Figures 18 and 19) to be compared with the actual response surface of Figure 4:

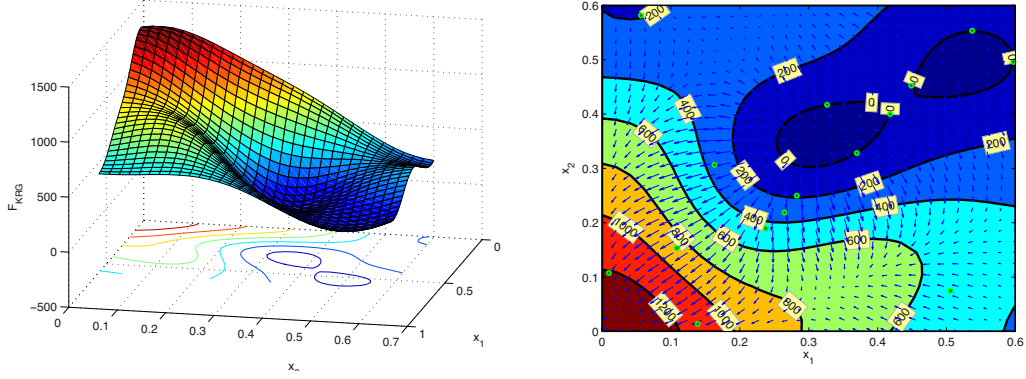


Figure 18: The kriging response surface of the *three-squares test case* (using 15 responses)

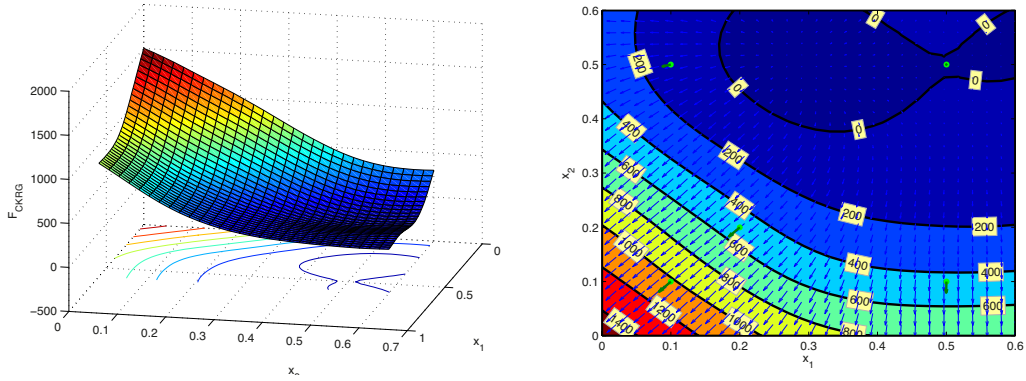


Figure 19: The cokriging response surface of the *three-squares test case* (using 5 responses and 5 gradients)

Tables 5 and 6 show the characteristics of the calculations and of the metamodels.

Gradients	without	with
Total CPU Time	44.3s	36.4s
Avg CPU Time (/calc)	2.95s	2.13s
Min CPU Time	0.6s	0.2s
Max CPU Time	8.8s	8.2s
LATIN iterations	2,031	1,611
Avg iterations (/calc)	135	107
Min iterations	25	5
Max iterations	411	391
Gain	2.97	3.40

Table 5: Characteristics of the mechanical calculations

	$R^2$	RAAE	RMAE	$Q_1$	$Q_2$	$Q_3$
OK	0.9865	$8.751 \cdot 10^{-2}$	0.6624	$2.746 \cdot 10^{-2}$	0.3385	$8.463 \cdot 10^{-4}$
OCK	0.9822	0.1015	0.4024	$1.014 \cdot 10^{-2}$	0.4441	$1.110 \cdot 10^{-3}$

Table 6: Characteristics of the metamodels (OK: ordinary kriging/OCK: ordinary cokriging)

In this case, the cokriging metamodel led to a much better approximation of the mechanical model than the kriging metamodel. Moreover, this result was obtained at a lower computation cost.

### 5.2. Case of a fixed metamodel quality

Another way to study the cost of the metamodels consists in constructing kriging and cokriging metamodels with the same quality. For this study, we constructed the metamodels using sample points obtained through full factorial sampling. We chose to build the kriging metamodel with  $9 \times 9$  samples and the cokriging metamodel with  $3 \times 3$  and  $5 \times 5$  samples. The results are shown in Table 7.

$5 \times 5$  full factorial sampling led to a cokriging metamodel with the same quality as the  $9 \times 9$  kriging metamodel. As in the previous case with a fixed number of mechanical calls, the cost with the cokriging metamodel was lower. In the case of more complex problems and more parameters, one can expect even better gains.

	OK $9 \times 9$	OCK $3 \times 3$	OCK $5 \times 5$
Total CPU Time	154.7s	61.6s	127s
Avg CPU Time (/calc)	1.9s	2.3s	1.7s
Min CPU Time	0.2s	0.2s	0.2s
Max CPU Time	8.9s	8.1s	9.1s
LATIN iterations	6,576	2,731	5,436
Avg iterations (/calc)	81.2	101	72.5
Min iterations	5	5	5
Max iterations	381	381	381
Gain	4.65	3.53	5.39
$R^2$	0.9878	0.9702	0.9942
RAAE	$2.511 \cdot 10^{-2}$	0.1234	$5.147 \cdot 10^{-2}$
RMAE	0.1442	0.5703	0.3243
$Q_1$	$1.301 \cdot 10^{-3}$	$2.036 \cdot 10^{-2}$	$6.585 \cdot 10^{-3}$
$Q_2$	$3.044 \cdot 10^{-2}$	0.7439	0.1450
$Q_3$	$7.610 \cdot 10^{-5}$	$1.860 \cdot 10^{-3}$	$3.626 \cdot 10^{-4}$

Table 7: The metamodels constructed using full factorial sampling

## 6. Conclusion

In this paper, we associated a cokriging metamodel with a dedicated multiparametric strategy. These two elements enabled us to achieve a significant reduction in the computation cost of constructing the metamodel.

Our multiparametric strategy takes advantage of a property of the LATIN method which is that the result of a previous calculation constitutes an efficient starting point for a subsequent calculation. This property is particularly advantageous for the evaluation of gradients, which are obtained using finite differences (*i.e.* with very small variations of the parameters). Thus, our strategy lends itself naturally to the use of a gradient-based metamodel such as the cokriging model presented in the paper.

Moreover, since this metamodel is part of a multilevel model optimization strategy, besides reducing the computation cost associated with the construction of the cokriging metamodel, one can also reduce

the cost of the second level, in which algorithms based on gradient descent are used. This will be the subject of an upcoming work.

## Acknowledgment

This work was supported by the *Agence Nationale de la Recherche* as part as ANR-08-COSI-007-10: Distributed Multi-Disciplinary Optimization.

- [1] G. Robinson, A. Keane, "A case for multi-level optimisation in aeronautical design", *Aeronautical Journal*, 103(1028): 481–485, 1999, ISSN 0001-9240.
- [2] J. Barthelemy, R. Haftka, "Approximation concepts for optimum structural design - a review", *Structural and Multidisciplinary Optimization*, 5(3): 129–144, 1993, ISSN 1615-147X.
- [3] A. Booker, J. Dennis, P. Frank, D. Serafini, V. Torczon, M. Trosset, "A rigorous framework for optimization of expensive functions by surrogates", *Structural and Multidisciplinary Optimization*, 17(1): 1–13, 1999, ISSN 1615-147X.
- [4] N. Queipo, R. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. Tucker, "Surrogate-based analysis and optimization", *Progress in Aerospace Sciences*, 41(1): 1–28, 2005, ISSN 0376-0421, URL <http://www.sciencedirect.com/science/article/B6V3V-4FWV2CX-1/2/1c6d71acaf8a60864cd5ddf136601fa8>.
- [5] R. Barton, M. Meckesheimer, "Metamodel-based simulation optimization", *Handbooks in operations research and management science*, 13: 535–574, 2006, ISSN 0927-0507.
- [6] G. Box, K. Wilson, "On the Experimental Attainment of Optimum Conditions", *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1): 1–45, 1951, ISSN 00359246, URL <http://www.jstor.org/stable/2983966>.
- [7] W. McCulloch, W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biology*, 5(4): 115–133, Dec. 1943, URL <http://dx.doi.org/10.1007/BF02478259>.
- [8] R. Hardy, "Multiquadric Equations of Topography and Other Irregular Surfaces", *J. Geophys. Res.*, 76: 1905–1915, 1971, URL <http://dx.doi.org/10.1029/JB076i008p01905>.
- [9] P. Ladevèze, "Sur une famille d'algorithmes en mécanique des structures", *Compte rendu de l'académie des Sciences*, 300(2): 41–44, 1985.
- [10] P. Ladevèze, *Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation*, Springer Verlag, 1999.
- [11] C. Blanzé, L. Champaney, J. Cognard, P. Ladevèze, "A modular approach to structure assembly computations: application to contact problems", *Engineering Computations*, 13(1): 15–32, 1995, ISSN 0264-4401.
- [12] L. Champaney, J. Cognard, P. Ladeveze, "Modular analysis of assemblages of three-dimensional structures with unilateral contact conditions", *Computers & Structures*, 73(1-5): 249–266, 1999.
- [13] P.A. Boucard, P. Ladevèze, "A multiple solution method for non-linear structural mechanics", *Mechanical Engineering*, 50(5): 317–328, 1999.
- [14] P.A. Boucard, "Application of the LATIN method to the calculation of response surfaces", in *Proceeding of the First MIT Conference on Computational Fluid and Solid Mechanics, Cambridge, USA*, Volume 1, pages 78–81, Juin 2001.
- [15] O. Allix, P. Vidal, "A new multi-solution approach suitable for structural identification problems", *Computer Methods in Applied Mechanics and Engineering*, 191(25-26): 2727–2758, 2002.
- [16] P.A. Boucard, L. Champaney, "A suitable computational strategy for the parametric analysis of problems with multiple contact", *International Journal for Numerical Methods in Engineering*, 57(9): 1259–1281, 2003.
- [17] L. Champaney, P.A. Boucard, S. Guinard, "Adaptive multi-analysis strategy for contact problems with friction", *Computational Mechanics*, 42(2): 305–315, 2008.
- [18] B. Soulier, P.A. Boucard, "A multiparametric strategy for the large-scale multilevel optimization of structural assemblies", in *8th World Congress on Structural and Multidisciplinary Optimization, Lisbon, Portugal*, 2009.
- [19] F. Chinesta, A. Ammar, E. Cueto, "Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models", *Archives of Computational methods in Engineering*, 17(4): 327–350, 2010.
- [20] F. Chinesta, P. Ladeveze, E. Cueto, "A short review on model order reduction based on proper generalized decomposition", *Archives of Computational Methods in Engineering*, pages 1–10, 2011.
- [21] H. Lamari, F. Chinesta, A. Ammar, E. Cueto, "Nonconventional numerical strategies in the advanced simulation of materials and processes", *Int J Mod Manuf Technol*, 1: 49–56, 2009.
- [22] H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Chinesta, F. Jacquemin, "Routes for efficient computational homogenization of nonlinear materials using the proper generalized decompositions", *Archives of Computational methods in Engineering*, 17(4): 373–391, 2010.
- [23] A. Ammar, M. Normandin, F. Chinesta, "Solving parametric complex fluids models in rheometric flows", *Journal of Non-Newtonian Fluid Mechanics*, 165(23): 1588–1601, 2010.
- [24] E. Pruliere, F. Chinesta, A. Ammar, "On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition", *Mathematics and Computers in Simulation*, 81(4): 791–810, 2010.
- [25] B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, A. Poitou, "Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity", *Computer Methods in Applied Mechanics and Engineering*, 201: 1–12, 2012.
- [26] C. Heyberger, P.A. Boucard, D. Néron, "Multiparametric analysis within the proper generalized decomposition framework", *Computational Mechanics*, 49(3): 277–289, 2012.
- [27] F. El Halabi, D. González, A. Roca, M. Doblaré, "Multiparametric Response Surface Construction by means of Proper Generalized Decomposition: An extension of the PARAFAC procedure", *Computer Methods in Applied Mechanics and Engineering*, 2012.

- [28] D. Krige, “A statistical approach to some mine valuation and allied problems on the Witwatersrand”, Master’s thesis, 1951.
- [29] G. Matheron, “Traité de géostatistique appliquée”, *Memoires du Bureau de Recherches Geologiques et Minieres*, 14, 1962.
- [30] G. Matheron, “Principles of geostatistics”, *Economic geology*, 58(8): 1246, 1963, ISSN 0361-0128.
- [31] H. Wackernagel, *Multivariate geostatistics: an introduction with applications*, Springer Verlag, 1995.
- [32] M. Morris, T. Mitchell, D. Ylvisaker, “Bayesian design and analysis of computer experiments: use of derivatives in surface prediction”, *Technometrics*, 35(3): 243–255, 1993, ISSN 0040-1706.
- [33] J. Koehler, A. Owen, “Computer experiments”, *Handbook of statistics*, 13: 261–308, 1996.
- [34] H. Chung, J. Alonso, “Using gradients to construct cokriging approximation models for high-dimensional design optimization problems”, in *40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*. Citeseer, 2002.
- [35] B. Matérn, *Spatial Variation (Lecture Notes in Statistics 36)*, Springer, Berlin, 1960.
- [36] M. Stein, *Interpolation of Spatial Data: some theory for kriging*, Springer Verlag, 1999, ISBN 0387986294.
- [37] J. Sacks, W. Welch, T. Mitchell, H. Wynn, “Design and analysis of computer experiments”, *Statistical science*, 4(4): 409–423, 1989, ISSN 0883-4237.
- [38] K. Mardia, R. Marshall, “Maximum likelihood estimation of models for residual covariance in spatial regression”, *Biometrika*, 71(1): 135, 1984, ISSN 0006-3444.
- [39] J. Barnes, B. Ripley, “Problems with likelihood estimation of covariance functions of spatial Gaussian processes”, *Biometrika*, 74(3): 640, 1987, ISSN 0006-3444.
- [40] K. Mardia, A. Watkins, “On multimodality of the likelihood in the spatial linear model”, *Biometrika*, 76(2): 289, 1989, ISSN 0006-3444.
- [41] M. Sasena, *Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations*, PhD thesis, University of Michigan, 2002.