Kriging-based sequential design strategies using fast cross-validation techniques with extensions to multi-fidelity computer codes

Loic Le Gratiet, Claire Cannamela

Kriging-based sequential design strategies using fast cross-validation techniques with extensions to multi-fidelity computer codes

Loic Le Gratiet [†] [‡] and Claire Cannamela [‡]

[†] Université Paris Diderot 75205 Paris Cedex 13

[‡] CEA, DAM, DIF, F-91297 Arpajon, France

October 29, 2012

1 Abstract

Kriging-based surrogate models have become very popular during the last decades to approximate a computer code output from few simulations. In practical applications, it is very common to sequentially add new simulations to obtain more accurate approximations. We propose in this paper a method of kriging-based sequential design which combines both the error evaluation providing by the kriging model and the observed errors of a Leave-One-Out cross-validation procedure. This method is proposed in two versions, the first one selects points one at-a-time. The second one allows us to parallelize the simulations and to add several design points at-a-time. Then, we extend these strategies to multi-fidelity co-kriging models which allow us to surrogate a complex code using fast approximations of it. The main advantage of these extensions is that it not only provides the new locations where to perform simulations but also which versions of code have to be simulated (between the complex one or one of its fast approximations).

A real multi-fidelity application is used to illustrate the efficiency of the proposed approaches. In this example, the accurate code is a two-dimensional finite element model and the less accurate one is a one-dimensional approximation of the system.

Keywords : kriging, co-kriging, sequential design, cross-validation, resource allocations, multi-fidelity codes.

2 Introduction

Kriging-based surrogate models have become very popular during the last decades to design and analyze computer experiments [Sacks et al., 1989]. The reader is referred to the books of [Stein, 1999], [Santner et al., 2003] and [Rasmussen and Williams, 2006] for more detail about kriging models. Usually, in real applications, two stages are performed to surrogate a computer code with a kriging model. The first one consists in building a kriging model from simulations

coming from an initial experimental design set. Many methods exist to build the initial design set, in order to ensure appropriate space filling properties, the reader is referred to [Fang et al., 2006] for a non-exhaustive review of them. The second stage consists in adding simulations sequentially at new design points which complete the initial one. The selection of the new points are usually based on criteria to improve the global accuracy of the kriging model and this will be our goal in this paper. To be complete, we mention that sequential kriging has also been widely used in optimization (see [Jones et al., 1998], [Picheny et al., 2012]) and to estimate probabilities of failure [Bect et al., 2012]

Kriging models are a powerful tool to enrich an experimental design set since it provides through the kriging variance - also called predictor's Mean Squared Error (MSE) or variance of prediction - an estimation of the model MSE. Kriging literature provides lot of criteria usually based on the kriging variance for sequentially design the experiments [Sacks et al., 1989]. Furthermore, [Bates et al., 1996] and [Picheny et al., 2010] propose more efficient criteria by considering the Integrated MSE (IMSE). It consists in integrating the mean value of the MSE integrated over the input parameter space. We note though that the IMSE can be computationally expensive to assess, especially when the dimension increases. Although these criteria are efficient for many cases, they can suffer from an important flaw when the accuracy of the kriging model is not homogeneous over the input parameter space. Indeed, the kriging variance is determined by the distances between prediction and design points but not by the real model errors. To fix this important flaw, we can use the Empirical IMSE suggested in [Sacks et al., 1989] which evaluates the model errors through a test set. Nevertheless, in a complex computer code framework, it could be too expensive to consider an external test set and cross-validation (CV) based criteria are more significant. As an illustration [Kleijnen and van Beers, 2004] and [van Beers and Kleijnen, 2008] combine a bootstrapping and a CV procedure to evaluate the predictor's MSE. Although this method improves the classical approach, it still does not take into account the real model errors. We note that a strength of the method proposed by [Kleijnen and van Beers, 2004] is that it can be applied to others type of surrogate models than the kriging one.

The first focus of this paper is on sequential design to improve the accuracy of a kriging model. In particular, we propose new criteria combining the kriging variance and the Leave-One-Out CV (LOO-CV) errors. The CV errors allow for focusing the new observations on regions where the real model errors are large. Furthermore, thanks to the equations presented in [Dubrule, 1983], the LOO-CV equations are fast to compute and thus the suggested approach is not expensive.

Kriging has recently been extended to allow for the use of coarse versions of a complex computer code to improve the accuracy of its approximation. These so-called multi-fidelity surrogate models have become of growing interest. A first one was proposed in [Craig et al., 1998] which is based on a linear regression formulation and was improved by [Cumming and Goldstein, 2009] through a Bayes linear formulation. The first multi-fidelity model using co-kriging was suggested by [Kennedy and O'Hagan, 2000]. Then, several works dealing with this model have been developed [Kennedy and O'Hagan, 2001], [Higdon et al., 2004], [Reese et al., 2004], [Qian and Wu, 2008]. Defining sequential design strategies in a multi-fidelity framework is also of interest and is still an open problem. A method based on nested Latin hypercube designs is suggested in [Xiong and Qian, 2012]. However, it does not allow for adding few simulations (e.g. it cannot perform an one step at-a-time sequential design) and it does not take into account the accuracy of the coarse code versions and the time ratios between two code levels.

The second focus of this paper is on sequential design for co-kriging model. We adapt the new strategies suggested for the kriging model to the multi-fidelity co-kriging one. The strength of the proposed extensions is that they not only provide the new points where to perform new simulations but they also determine which version of code is worth being simulated. These new criteria take into account the computational time ratios between code versions. They are based on a proxy of the IMSE reduction and on an original result giving the contribution of each code on the total variance of the model. We note that sequential design in a multi-fidelity framework has also been applied for optimization purposes [Forrester et al., 2007] and [Huang et al., 2006].

The paper is organized as follows. First, we introduce the kriging model and present our CV-based sequential design strategies. We illustrate these strategies in tabulated functions. Secondly, we present the proposed co-kriging multi-fidelity model and the extensions of the previous strategies. Finally, we apply the sequential co-kriging approach to a mechanical example.

3 Kriging models and sequential designs

In this Section, we briefly introduce the kriging model and some of its classical sequential design criteria. Then, we will present our sequential strategies to enhance kriging models considering the region with large LOO-CV errors.

3.1 The Kriging model

The kriging model is a widely used method to surrogate the output of a computer code from few simulations [Sacks et al., 1989]. Let us denote by $y(x)$ the output of the code at point $x \in Q$. Here, $y(x)$ is a scalar and $Q \subset \mathbb{R}^d$. Furthermore, we denote by $\mathbf{D} = \{x_1, \ldots, x_n\}$ the experimental design set and $\mathbf{y}_n = y(\mathbf{D})$ the value of $y(x)$ at points in $\mathbf{D}$.

In a kriging framework, we set that the prior knowledges about the code can be modeled by a Gaussian process $Y_0(x)$. Usually, we consider a Gaussian process with mean of the form $m_0(x) = \mathbf{f}'(x)\beta$, with $\mathbf{f}'(x) = (f_1(x), \ldots, f_p(x))$ and with covariance function $k_0(x, \tilde{x}) = \sigma^2 r(x, \tilde{x}; \theta)$. Then, the kriging equations are given by the Gaussian process $Y_0(x)$ conditioned by its known values $\mathbf{y}_n$ at points in $\mathbf{D}$:

$$Y_n(x) = [Y_0(x) | Y_0(\mathbf{D}) = \mathbf{y}_n] \sim \mathcal{GP}\left(m_n(x), k_n(x, \tilde{x})\right) \tag{1}$$

where:

$$m_n(x) = \mathbf{f}'(x)\hat{\beta} + \mathbf{r}'(x)\mathbf{R}^{-1}(\mathbf{y}_n - \mathbf{F}\hat{\beta}) \tag{2}$$

and:

$$k_n(x, \tilde{x}) = \sigma^2 \left(r(x, \tilde{x}) - \begin{pmatrix} \mathbf{f}'(x) & \mathbf{r}'(x) \end{pmatrix} \begin{pmatrix} 0 & \mathbf{F}' \\ \mathbf{F} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f}(x) \\ \mathbf{r}(x) \end{pmatrix} \right) \tag{3}$$

where $'$ stands for the transpose, $\mathbf{F}$ are the values of $\mathbf{f}'(x)$ at points in $\mathbf{D}$, $\mathbf{r}(x)$ is the correlation vector between $\mathbf{D}$ and x with respect to the correlation function $r(x, \tilde{x})$, $\mathbf{R}$ is the correlation matrix of $\mathbf{D}$ with respect to $r(x, \tilde{x})$ and $\hat{\beta} = (\mathbf{F}'\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathbf{y}_n$ is the usual least-squares estimates of β. The model parameters σ^2 and θ could be estimated by maximizing their Likelihood [Santner et al., 2003] or with a cross-validation procedure [Rasmussen and Williams, 2006]. Furthermore, the Maximum Likelihood Estimate (MLE) of σ^2 is given by $\hat{\sigma}^2 = (\mathbf{y}_n - \mathbf{F}\hat{\beta})'\mathbf{R}^{-1}(\mathbf{y}_n - \mathbf{F}\hat{\beta})/(n-p)$.

1 point at-a-time Sequential design

Now, let us suppose that we want to add a new point x_{n+1} in $\mathbf{D}$ in order to enhance the accuracy of the kriging model. From the kriging variance $k_n(x,x)$ - representing the model MSE - some sequential design methods have been derived [Sacks et al., 1989], [Bates et al., 1996] and [Picheny et al., 2010]. A first one consists in adding x_{n+1} where the kriging variance is the largest (see [Sacks et al., 1989]):

$$x_{n+1} = \arg\max_x k_n(x,x) \tag{4}$$

However, as presented in [Kleijnen and van Beers, 2004], its performance is poor. Then, it has been improved with a criterion which consists in adding the new point which leads the most important IMSE reduction (see [Bates et al., 1996] and [Picheny et al., 2010]):

$$x_{n+1} = \arg\max_x \int_{u\in Q} k_n(u,u) - k_{n+1}(u,u)\, du \tag{5}$$

Here, the covariance kernels $k_{n+1}(u,\tilde{u})$ corresponds to the one of the Gaussian process $Y_n(u)$ (1) conditioned by a new observation at x. Furthermore, the equation (3) shows that the kriging variance does not depend on the observations if we consider known the parameters σ^2 and θ. Therefore, in that case, $k_{n+1}(u,u)$ can be computed without new simulations. We denote by MinIMSE this criterion. Finally, we also consider the criterion presented by [Kleijnen and van Beers, 2004] using a Jackknife estimator for the predictor's variance. Its principle is the following one. Let us consider $m_{n,-i}(x)$ the kriging mean built without the ith observation, the jackknife variance is given by:

$$s^2_{jack}(x) = \frac{1}{n(n-1)} \sum_{i=1}^{n} (\tilde{y}_i - \bar{\tilde{y}})^2 \tag{6}$$

where $\tilde{y}_i = n m_n(x) - (n-1) m_{n,-i}(x)$ and $\bar{\tilde{y}} = \sum_{i=1}^{n} \tilde{y}_i / n$. Then, we consider candidate points coming from a maximin LHS Design [Fang et al., 2006] and we add those which maximize the jackknife variance. We denote by KleiCrit this criterion.

q points at-a-time Sequential design

There is a natural way to extend these algorithms when the simulations can be performed simultaneously. Indeed, the covariance kernel $k_{n+1}(x,\tilde{x})$ of the Gaussian process $Y_n(x)$ conditioned by the new observation at point x_{n+1} can be computed without knowing $y(x_{n+1})$ when we consider the model parameters σ^2 and θ as known. Then, from $k_{n+1}(x,\tilde{x})$, we can find a new point x_{n+2} where to perform a new simulation using the same criterion as in equation (5) and the kernel $k_{n+2}(x,\tilde{x})$. Thus, considering the parameters σ^2 and θ as known (they are fixed to their estimated values), we can determine with this procedure q good locations where to perform simulations. We call this method the liar sequential kriging. This idea is also presented in a framework of kriging-based optimization in [Ginsbourger et al.,].

3.2 LOO-CV based strategies for kriging sequential design

We present in this subsection new sequential-kriging strategies. The main difference between these new strategies and the previous ones is that they take into account the real model errors through the LOO-CV equations.

The proposed sequential methods are based on the following original proposition. It gives the closed form expressions of the LOO-CV equations where the model parameters β and σ^2 are re-estimated after each removed point. This result is already known when σ^2 is fixed [Dubrule, 1983] and [Rasmussen and Williams, 2006].

Notations: $\mathbf{A}_{i,i}$ is the ith element of the main diagonal of $\mathbf{A}$, $\mathbf{A}_i$ is the ith row of the matrix $\mathbf{A}$, $\mathbf{A}_{-i}$ is the matrix $\mathbf{A}$ without its ith row, $\mathbf{A}_{-i,i}$ is the ith column of $\mathbf{A}$ without its ith element, $\mathbf{A}_{i,-i} = \mathbf{A}'_{-i,i}$ and $\mathbf{A}_{-i,-i}$ is the matrix $\mathbf{A}$ without the ith row and column.

Proposition 1 *Let us denote by $Y_{n,-i}(x)$ the Gaussian process $Y_0(x)$ conditioned by the values $\mathbf{y}_{n,-i} = y(\mathbf{D}) \setminus y(x_i)$. Then, the predictive mean of $Y_{n,-i}(x)$ at point x_i is given by:*

$$m_{n,-i}(x_i) = y(x_i) - \left[\mathbf{R}^{-1}(\mathbf{y}_{n,-i} - \mathbf{F}_{-i}\hat{\beta}_{-i})\right]_i / \left[\mathbf{R}^{-1}\right]_{i,i} \tag{7}$$

where $\hat{\beta}_{-i} = (\mathbf{F}'_{-i}\mathbf{K}_i\mathbf{F}_{-i})^{-1}\mathbf{F}'_{-i}\mathbf{K}_i\mathbf{y}_{n,-i}$ and $\mathbf{K}_i = \left[\mathbf{R}^{-1}\right]_{-i,-i} - \left[\mathbf{R}^{-1}\right]_{-i,i}\left[\mathbf{R}^{-1}\right]_{i,-i} / \left[\mathbf{R}^{-1}\right]_{i,i}$. Furthermore, the predictive variance of $Y_{n,-i}(x)$ at point x_i is given by:

$$k_{n,-i}(x_i) = \sigma^2_{-i} / \left[\mathbf{R}^{-1}\right]_{i,i} + \varsigma_{-i}(x_i) \tag{8}$$

where $\varsigma_{-i}(x_i) = \left(\left[\mathbf{R}^{-1}\mathbf{F}\right]_i / \left[\mathbf{R}^{-1}\right]_{i,i}\right)' (\mathbf{F}'_{-i}\mathbf{K}_i\mathbf{F}_{-i})^{-1} \left(\left[\mathbf{R}^{-1}\mathbf{F}\right]_i / \left[\mathbf{R}^{-1}\right]_{i,i}\right)$ *and*
$\sigma^2_{-i} = \left(\mathbf{y}_{n,-i} - \mathbf{F}_{-i}\hat{\beta}_{-i}\right)' \mathbf{K}_i \left(\mathbf{y}_{n,-i} - \mathbf{F}_{-i}\hat{\beta}_{-i}\right) / (n - p - 1)$

The previous proposition provides a powerful tool to compute the LOO-CV predictive means and variances. Indeed, several elements of the equations have been already computed during the models construction (e.g. the inverse of the matrix $\mathbf{R}$). Consequently, the LOO-CV equations are fast to compute and can be easily recomputed at each step of the sequential strategy. We note that the originality of this result is the estimation of σ^2_{-i}. As we use the value of $k_{n,-i}(x_i)$ strongly depending on σ^2_{-i} in our forthcoming developments, it is important to well estimate it.

Now, let us denote by $\mathbf{e}^2_{\text{LOO-CV}} = \left[\left(y(x_i) - m_{n,-i}(x_i)\right)^2\right]_{i=1,\dots,n}$ the vector of the LOO-CV squared errors and $\mathbf{s}^2_{\text{LOO-CV}} = [k_{n,-i}(x_i)]_{i=1,\dots,n}$ the vector of the LOO-CV variances. Furthermore, let us consider the Voronoi cells $(V_i)_{i=1,\dots,n}$ associated with the points $(x_i)_{i=1,\dots,n}$:

$$V_i = \{x \in Q,\, ||x - x_i|| \leq ||x - x_j||,\, \forall j \neq i\},\, i, j = 1, \dots, n \tag{9}$$

In the remainder of this section, we present two strategies to sequentially add simulations which use $\mathbf{e}^2_{\text{LOO-CV}}$, $\mathbf{s}^2_{\text{LOO-CV}}$ and V_i. The intuitive idea of the suggested criteria is to enhance the predictive variance in the locations where the LOO-CV errors are important.

LOO-CV-based 1 point at-a-time Sequential design

Let us denote by x_{n+1} the new point that we want to add to $\mathbf{D}$. We consider the point solving the following problem:

$$x_{n+1} = \arg\max_x k_n(x, x) \left(1 + \sum_{i=1}^{n} \frac{[\mathbf{e}^2_{\text{LOO-CV}}]_i}{[\mathbf{s}^2_{\text{LOO-CV}}]_i} \mathbb{I}_{x \in V_i}\right) \tag{10}$$

This criterion considers the predictor's MSE $k_n(x,x)$ adjusted with the LOO-CV errors and variances. For equivalent $k_n(x,x)$, the criterion favors the points close to an experimental design point with large LOO-CV errors. Furthermore, if two points are in the same Voronoi cell, the one with the largest predictor's MSE is considered. Therefore, a sequential strategy with this criterion focus on the regions of Q where the LOO-CV errors are the largest. We note that the standardization with $\mathbf{s}^2_{\mathrm{LOO-CV}}$ is important since it is not necessary to enlarge the predictor's MSE in the regions where it is well or over estimated.

LOO-CV-based q points at-a-time Sequential design

We extend here the previous criterion for a q points at-a-time sequential design. First, we emphasize that the liar sequential kriging is not relevant for this new criterion. Indeed, conditioning on model parameters, with a liar method we can compute the kriging variances $(k_{n+i}(x,x))_{i=1,\ldots,q}$ but not the LOO-CV equations. Therefore, we use another strategy to propose q new locations where to perform the simulations. This approach is proposed in [Dubourg et al., 2011] in a different framework. The idea of the suggested method is to select the q best points with respect to the criterion (10) from N candidate points. These N candidate points are chosen with the following algorithm.

1. Generate N_{MCMC} samples with respect to the probability density function proportional to $k_n(x,x)$ with a suitable Markov Chain Monte Carlo (MCMC) technique [Robert and Casella, 2004].

2. Extract from these samples N representative points with a N-means clustering technique [MacQueen, 1967].

As presented in [Dubourg et al., 2011] the use of this algorithm to select N candidate points in a kriging framework is efficient. Indeed, it allows us to concentrate the points at the modes of the kriging variance. In the proposed strategy, we always take $N \geq q$ and we choose from the N cluster centers $(C_i)_{i=1\ldots,N}$ the q points where $k_{n,\mathrm{adj}}(x,x) = k_n(x,x)\left(1+\sum_{i=1}^{n}\frac{[\mathbf{e}^2_{\mathrm{LOO-CV}}]_i}{[\mathbf{s}^2_{\mathrm{LOO-CV}}]_i}\mathbb{I}_{x\in V_i}\right)$ is the largest. For the MCMC procedure, we use a Metropolis-Hastings (M-H) algorithm with a Gaussian jumping distribution. It is centered on the last sample point and has a standard deviation such that the acceptance rate is around 30% (see [Robert and Casella, 2004]). Furthermore, we set N_{MCMC} such that $N_{\mathrm{MCMC}} \gg N$. For the N-means procedure, we choose the value of N with respect to the following criterion:

$$\max_{N\geq q}\ \min_{x\in(C_i)_i} k_n(x,x) \tag{11}$$

where $(C_i)_{i=1\ldots,N}$ are the cluster centers. This criterion prevents from having a cluster center in a region where the kriging variance is close to zero. Furthermore, if the number of clusters is too high, the cluster centers get away from the modes and consequently the value of $\min_{x\in(C_i)_{i=1\ldots,N}} k_n(x,x)$ decreases. Therefore, this criterion also prevents from having a number of clusters too large.

4 Sequential design in a multi-fidelity framework

A computer code can often be run at different levels of accuracy. In this case, it can be worth using low-accuracy versions of a code to improve its approximation. Co-kriging models are well suited to build such multi-fidelity surrogate models (see [Kennedy and O'Hagan, 2000]

and [Qian and Wu, 2008]). In this section, we present the considered multi-fidelity co-kriging models and we extend the previous sequential design strategies in this framework. We note that, in a multi-fidelity framework, the search for the best locations where to run the code is not the only point of interest. Indeed, once the best locations are determined, we also have to decide which code level is worth being run. This will not only depend on the time-ratios between the code levels but also on the contribution of each code level to the total predictor's MSE.

4.1 Multi-fidelity co-kriging models

Let us suppose that we want to surrogate a computer code output $y^s(x)$ and that coarse versions of this code $(y^l(x))_{l=1,\dots,s-1}$ are available. These codes are sorted by order of fidelity from the less accurate $y^1(x)$ to the most accurate $y^{s-1}(x)$. All code levels are modeled by Gaussian processes $(Y^l(x))_{l=1,\dots,s}$ with respect to the following relationship with $l = 2, \dots, s$:

$$\begin{cases} Y^l(x) = \rho_{l-1}\tilde{Y}^{l-1}(x) + \delta^l(x) \\ \tilde{Y}^{l-1}(x) \perp \delta^l(x) \\ \tilde{Y}^{l-1}(x) \sim [Y^{l-1}(x)|\mathbf{Y}^{(l-1)} = \mathbf{y}^{(l-1)}] \end{cases} \tag{12}$$

where $\mathbf{Y}^{(l)} = (\mathbf{Y}^1, \dots, \mathbf{Y}^l)$ with $\mathbf{Y}^l = Y^l(\mathbf{D}^l)$, $\mathbf{y}^{(l)} = (\mathbf{y}^1, \dots, \mathbf{y}^l)$ with $\mathbf{y}^l = y^l(\mathbf{D}^l)$ and $(\mathbf{D}^l)_{l=1,\dots,s}$ are the experimental design sets at level l with n_l points and such that $\mathbf{D}^s \subseteq \mathbf{D}^{s-1} \subseteq \cdots \subseteq \mathbf{D}^1$. We note that the nested property is not necessary to build the model but allows for efficient parameter estimations. Furthermore, conditioning on parameters β_l, σ_l^2 and θ_l, $\delta^l(x)$ is a Gaussian process of mean $\mathbf{f}_l'(x)\beta_l$ with $\mathbf{f}_l'(x) = (f_1^l(x), \dots, f_{p_l}^l(x))$ and covariance kernel $\sigma_l^2 r_l(x, \tilde{x}, \theta_l)$. By convention $Y^1(x)$ has the same distribution as $\delta^1(x)$.

The multi-fidelity co-kriging model at level $l = 2, \dots, s$ is given by the following distribution:

$$Y_{n_l}^l(x) = [Y^l(x)|\mathbf{Y}^{(l)} = \mathbf{y}^{(l)}] \sim \mathcal{GP}(\mu_{n_l}^l(x), k_{n_l}^l(x, \tilde{x})) \tag{13}$$

with:

$$\mu_{n_l}^l(x) = \hat{\rho}_{l-1}\mu_{n_{l-1}}^{l-1}(x) + \mathbf{f}_l(x)\hat{\beta}_l + \mathbf{r}_l'(x)\mathbf{R}_l^{-1}(\mathbf{y}^l - \mathbf{F}_l\hat{\beta}_l - \hat{\rho}_{l-1}y^{l-1}(\mathbf{D}^l)) \tag{14}$$

and:

$$k_{n_l}^l(x, \tilde{x}) = \hat{\rho}_{l-1}^2 k_{n_{l-1}}^{l-1}(x, \tilde{x}) + \sigma_l^2 \left(r_l(x, \tilde{x}) - \begin{pmatrix} \mathbf{h}_l'(x) & \mathbf{r}_l'(x) \end{pmatrix} \begin{pmatrix} 0 & \mathbf{H}_l' \\ \mathbf{H}_l & \mathbf{R}_l \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{h}_l(\tilde{x}) \\ \mathbf{r}_l(\tilde{x}) \end{pmatrix} \right) \tag{15}$$

where $\mathbf{F}_l$ are the values of $\mathbf{f}_l'(x)$ at points in $\mathbf{D}^l$, $\mathbf{r}_l(x)$ is the correlation vector between $\mathbf{D}^l$ and x with respect to the correlation function $r_l(x, \tilde{x})$, $\mathbf{R}_l$ is the correlation matrix of $\mathbf{D}^l$ with respect to $r_l(x, \tilde{x})$, $\mathbf{h}_l'(x) = \left[\mu_{n_{l-1}}^{l-1}(x)\ \mathbf{f}_l'(x)\right]$ and $\begin{pmatrix} \hat{\rho}_{l-1} \\ \hat{\beta}_l \end{pmatrix} = (\mathbf{H}_l'\mathbf{R}_l^{-1}\mathbf{H}_l)^{-1}\mathbf{H}_l'\mathbf{R}_l^{-1}\mathbf{y}^l$ with $\mathbf{H}_l = [y^{l-1}(\mathbf{D}^l)\, F_l]$ is the usual least-squares estimates of $\begin{pmatrix} \rho_{l-1} \\ \beta_l \end{pmatrix}$. Furthermore, the restricted maximum likelihood estimate of σ_l^2 is given by $\hat{\sigma}_l^2 = \left(\mathbf{y}^l - \mathbf{H}_l \begin{pmatrix} \hat{\rho}_{l-1} \\ \hat{\beta}_l \end{pmatrix}\right)' \mathbf{R}_l^{-1} \left(\mathbf{y}^l - \mathbf{H}_l \begin{pmatrix} \hat{\rho}_{l-1} \\ \hat{\beta}_l \end{pmatrix}\right) / (n_l - p_l - 1)$ [Santner et al., 2003].

Remark. The important property of this co-kriging model is that its MSE (15) provides through the term $\rho_{l-1}^2 k_{n_{l-1}}^{l-1}$ the contribution of the code level $l-1$ to the total predictor's MSE. Therefore, it can allow us to determine which code level is worth being simulated at a new location x.

4.2 Sequential design for multi-fidelity co-kriging models

The aim of this subsection is to extend the sequential kriging strategies proposed in Subsection 3.2 to the suggested multi-fidelity co-kriging model. These extensions are based on the variance decomposition property presented in Subsection 4.1 in equation (15) and on the following extension of Proposition 1:

Proposition 2 *For $l = 1, \dots, s$, let us consider the multi-fidelity model (12) with $\mathbf{D}^l \subseteq \mathbf{D}^{l-1} \subseteq \cdots \subseteq \mathbf{D}^1$. We denote by ξ_j the index of $\mathbf{D}^j$ corresponding to the i^{th} point x_i^l of $\mathbf{D}^l$ with $1 \leq j \leq l$. Then, if we note $\varepsilon_{\mathrm{LOO-CV},l}(x_i^l)$ the LOO-CV error (i.e. real value minus predicted value) at level l and point x_i^l, we have:*

$$\varepsilon_{\mathrm{LOO-CV},l}(x_i^l) \;=\; \hat{\rho}_{l-1}\varepsilon_{\mathrm{LOO-CV},l-1}(x_i^l) + \left[\mathbf{R}_l^{-1}\left(\mathbf{y}^l - \mathbf{H}_l \begin{pmatrix} \hat{\rho}_{l-1} \\ \hat{\beta}_l \end{pmatrix}\right)\right]_{\xi_l} / \left[\mathbf{R}_l^{-1}\right]_{\xi_l,\xi_l} \quad (16)$$

where $\varepsilon_{\mathrm{LOO-CV},1}(x_i^l)$ is given by Proposition 1, $\begin{pmatrix} \hat{\rho}_{l-1} \\ \hat{\beta}_l \end{pmatrix} = (\mathbf{H}'_{l,-\xi_l}\mathbf{K}_l\mathbf{H}_{l,-\xi_l})^{-1}\mathbf{H}'_{l,-\xi_l}\mathbf{K}_l\mathbf{y}^l_{-\xi_l}$ and $\mathbf{K}_l = \left[\mathbf{R}_l^{-1}\right]_{-\xi_l,-\xi_l} - \left[\mathbf{R}_l^{-1}\right]_{-\xi_l,\xi_l}\left[\mathbf{R}_l^{-1}\right]_{\xi_l,-\xi_l} / \left[\mathbf{R}_l^{-1}\right]_{\xi_l,\xi_l}$.

Furthermore, if we note $\sigma^2_{\mathrm{LOO-CV},l}(x_i^l)$ the variance of the LOO-CV, we have:

$$\sigma^2_{\mathrm{LOO-CV},l}(x_i^l) = \hat{\rho}_{l-1}^2\sigma^2_{\mathrm{LOO-CV},l-1}(x_i^l) + \sigma^2_{l,-\xi_l} / \left[\mathbf{R}_l^{-1}\right]_{\xi_l,\xi_l} + \varsigma_l \quad (17)$$

with $\varsigma_l = \mathrm{u}_l^2\left(\mathbf{H}'_{l,-\xi_l}\mathbf{K}_l\mathbf{H}_{l,-\xi_l}\right)^{-1}$, $\mathrm{u}_l = \left[\mathbf{R}_l^{-1}\mathbf{H}_l\right]_{\xi_s} / [\mathbf{R}_l^{-1}]_{\xi_l,\xi_l}$ and:

$$\sigma^2_{l,-\xi_l} = \frac{\left(\mathbf{y}^l_{-\xi_l} - \mathbf{H}_{l,-\xi_l}\begin{pmatrix} \hat{\rho}_{l-1} \\ \hat{\beta}_l \end{pmatrix}\right)' \mathbf{K}_s \left(\mathbf{y}^l_{-\xi_l} - \mathbf{H}_{l,-\xi_l}\begin{pmatrix} \hat{\rho}_{l-1} \\ \hat{\beta}_l \end{pmatrix}\right)}{n_l - p_l - 2} \quad (18)$$

where $\mathbf{H}_{l,-\xi_l} = [\mathbf{y}^l_{-\xi_l} \quad \mathbf{F}_{l,-\xi_l}]$.

Proposition 2 provides closed form expressions for the LOO-CV errors and variances. From them, the LOO-CV equations are fast to compute and consequently they can be used in a sequential procedure with a low computational cost. Furthermore, since the experimental design sets are nested, we state that during the LOO-CV procedure at level l, the points are removed from all code levels. Finally, from these equations, we can adjust the co-kriging variances $\left(k_{n_l}^l(x,\tilde{x})\right)_{l=1,\dots,s}$ at each level using the same method as presented in equation (10).

1 point at-a-time sequential co-kriging. First, let us consider x_{new} the point solving the problem:

$$x_{\mathrm{new}} = \arg\max_x k_{n_s}^s(x,x) \quad (19)$$

Therefore, we want to compute a new simulation at point where the predictor's MSE is maximal. Now, let us consider two successive code levels $l-1$ and l. The question of interest is to estimate which of these two code levels is worth being simulated.

First, thanks to the equation (15), we can deduce the contribution of each code levels to the predictor's MSE. Let us define the following notation for $l = 2, \ldots, s$:

$$\sigma^2_{\delta^l}(x) = \sigma^2_l \left(1 - \begin{pmatrix} \mathbf{h}'_l(x) & \mathbf{r}'_l(x) \end{pmatrix} \begin{pmatrix} 0 & \mathbf{H}'_l \\ \mathbf{H}_l & \mathbf{R}_l \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{h}_l(x) \\ \mathbf{r}_l(x) \end{pmatrix} \right) \tag{20}$$

and $\sigma^2_{\delta^1}(x) = k^1_{n_1}(x,x)$. Then, we have:

$$k^l_{n_l}(x,x) = \sum_{i=1}^{l} \sigma^2_{\delta^i}(x) \prod_{j=i}^{l-1} \rho^2_j \tag{21}$$

Let us consider that the parameters $(\theta_l)_{l=1,\ldots,s}$ define the characteristic length-scales of the kernels $((r_l(x,\tilde{x};\theta_l))_{i=1,\ldots,s}$ (see [Rasmussen and Williams, 2006] p.83). Then, we can approximate the reduction of the IMSE after adding a new point x_{new} at level l with the following formula:

$$\text{IMSE}^l_{\text{red}}(x_{\text{new}}) = \sum_{i=1}^{l} \sigma^2_{\delta^i}(x_{\text{new}}) \prod_{j=i}^{l-1} \rho^2_j \prod_{m=1}^{d} \theta^m_i \tag{22}$$

with $\theta_l = (\theta^1_l, \ldots, \theta^d_l)$. Indeed, at each stage, $\sigma^2_{\delta^i}(x_{\text{new}}) \prod_{j=i}^{l-1} \rho^2_j$ represents the contribution of the bias $\delta^i(x)$ to the co-kriging variance and $\prod_{m=1}^{d} \theta^m_i$ represents the volume of influence of x_{new} at level j. This criterion is justify by the fact that the reduction of IMSE^l defined by $\text{IMSE}^l = \int_Q \sigma^2_{\delta^l}(x)\, dx$ after adding a new point x_{new} has the same order of magnitude than $\sigma^2_{\delta^i}(x_{\text{new}})$ times the volume of influence $\prod_{m=1}^{d} \theta^m_i$ of x_{new}.

Now, let us consider that the ratio of computational times between the codes $y^l(x)$ and $y^{l-1}(x)$ equals $B_{l/l-1}$. It is worth running the code $y^{l-1}(x)$ if $B_{l/l-1}\text{IMSE}^{l-1}_{\text{red}}(x_{\text{new}}) > \text{IMSE}^l_{\text{red}}(x_{\text{new}})$, i.e. if the potential uncertainty reduction by running $B_{l/l-1}$ times $y^{l-1}(x)$ is greater than the one when we run one simulation on $y^l(x)$. From this criterion, we can deduce the following algorithm for an one at-a-time sequential co-kriging model taking into account both the computational ratios between the different code levels and the contribution of each level to the total co-kriging variance.

Algorithm 1 One point at-a-time sequential co-kriging

Find x_{new} such that $x_{\text{new}} = \arg\max_x k^s_{n_s}(x,x)$
for $l = 2, \ldots, s$ **do**
  if $\left(\sigma^2_{\delta^l}(x_{\text{new}}) < \text{IMSE}^l\right)$ **then**
    Run $y^{l-1}(x_{\text{new}})$
    end for
  else
    if $\left(\text{IMSE}^{l-1}_{\text{red}}(x_{\text{new}})/\text{IMSE}^{l}_{\text{red}}(x_{\text{new}}) > 1/B_{l/l-1}\right)$ **then**
      Run $y^{l-1}(x_{\text{new}})$
      end for
    end if
  end if
end for
if $(l = s)$ **then**
  Run $y^l(x_{\text{new}})$
end if

Remarks: Algorithm 1 evaluates for two successive code levels $l-1$ and l, which one is worth being simulated. It starts with the levels one and two, then two and three and so on. When it finds that the level $l-1$ is more promising than the level l, it stops the loop and simulate x_{new} at code levels $y^1(x), \ldots, y^{l-1}(x)$. Since the loop is defined from level 1 to level s, it favors simulations at low code levels. Therefore, it will tend to learn the coarse code versions before learning the accurate ones. We note that during the loop of the algorithm 1, the parameters are not re-estimated. In fact, they are re-estimated after adding the new point x_{new}. Moreover, the first test $\sigma^2_{\delta^l}(x_{\text{new}}) < \text{IMSE}^l$ checks in averaged if the code level l at point x_{new} is worth being run. Then, the test $\text{IMSE}^{l-1}_{\text{red}}(x_{\text{new}})/\text{IMSE}^{l}_{\text{red}}(x_{\text{new}}) > 1/B_{l/l-1}$ evaluates which code levels between l and $l-1$ is the most promising. Finally, if we consider that the code level l is more promising than the code level $l-1$, we confront it to the following code level $l+1$. We note that the algorithm 1 is reiterated until a prescribed accuracy is reached or the computational time budget is spent.

1 point at-a-time sequential co-kriging with adjusted predictor's MSE. From Proposition 2, Algorithm 1 and Equation (22), we can extend the criterion (10) to the multi-fidelity co-kriging model. Let us consider the following quantity:

$$\begin{aligned}\text{IMSE}^l_{\text{red,adj}}(x_{\text{new}}) &= \sum_{i=1}^{l} \sigma^2_{\delta^i}(x_{\text{new}}) \left(1 + \sum_{j=1}^{n_i} \frac{\left(\varepsilon_{\text{LOO-CV},i}(x^i_j) - \hat{\rho}_{i-1}\varepsilon_{\text{LOO-CV},i-1}(x^i_j)\right)^2}{\sigma^2_{\text{LOO-CV},i}(x^i_j) - \hat{\rho}^2_{i-1}\sigma^2_{\text{LOO-CV},i-1}(x^i_j)}\right) \\ &\times \prod_{j=i}^{l-1} \rho_j^2 \prod_{m=1}^{d} \theta_i^m \end{aligned} \tag{23}$$

with the convention $\hat{\rho}_0 = 0$ and $(\hat{\rho}_i)_{i=1,\ldots,l}$ is given by Proposition 2. In equation (23), the kriging variances $\sigma^2_{\delta^i}(x)$ in equation (21) is replaced with the adjusted kriging variance presented in Subsection 3.2. We note that $\left(\varepsilon_{\text{LOO-CV},i}(x^i_j) - \hat{\rho}_{i-1}\varepsilon_{\text{LOO-CV},i-1}(x^i_j)\right)^2$ is the part of the LOO-CV squared error explained by the bias $\delta^i(x)$ and $\sigma^2_{\text{LOO-CV},i}(x^i_j) - \hat{\rho}^2_{i-1}\sigma^2_{\text{LOO-CV},i-1}(x^i_j)$ is the corresponding LOO-CV predictive variance. To adapt the adjusted co-kriging variance in a multi-fidelity framework, we just have to replace $\text{IMSE}^l_{\text{red}}(x)$ with $\text{IMSE}^l_{\text{red,adj}}(x)$ in the

algorithm 1.

$(\mathbf{q^i})_{\mathbf{i=1,\ldots,s}}$ points at-a-time sequential co-kriging. In this paragraph, we propose an extension for the multi-fidelity model of the q points at-a-time sequential design presented in Subsection 3.2. Its principle is the following one. First, we select q^l new points for the code $y^l(x)$ with the method presented in Subsection 3.2 "LOO-CV based q points at-a-time Sequential design". Then, we consider these points as known for the code $y^{l-1}(x)$ and we select q^{l-1} new points for this code with the same method. We note that, as presented in Subsection 3.1, we can use a liar method to compute the new co-kriging variance without simulating $y^{l-1}(x)$ at the q^l new points. Finally, we repeat this procedure for all code levels from $y^{l-2}(x)$ to $y^1(x)$. At the end of the procedure, we have $\sum_{i=j}^{l} q^i$ new points at level j and we want to find the allocation $\{q^1,\ldots,q^l\}$ leading the largest potential uncertainty reduction and under the constraint of a constant CPU time budget. We note the CPU time budget $T=\sum_{j=1}^{l}\sum_{i=j}^{l} q^i t^j$ where $(t^i)_{i=1,\ldots,s}$ represents the CPU times of codes $(y^i(x))_{i=1,\ldots,s}$. The algorithm 2 presents the suggested q points at-a-time sequential co-kriging.

Algorithm 2 $(q^i)_{i=1,\ldots,s}$ points at-a-time sequential co-kriging

1: Set the budget $T>0$ and the allocation $\{q^1,\ldots,q^l\}$ such that $\sum_{j=1}^{l}\sum_{i=j}^{l} q^i t^j = T$
2: Set $(N^i_{\mathrm{MCMC}})_{i=1,\ldots,l}$ for the M-H procedures.
3: Generate N^l_{MCMC} samples with respect to $k^l_{n_l}(x,x)$.
4: Find the N^l cluster centers $(C^l_i)_{i=1,\ldots,N^l}$ such that $N^l = \max_{N\geq q^l}\min_{x\in(C^l_i)_i} k^l_{n_l}(x,x)$
5: Select from $(C^l_i)_{i=1,\ldots,N^l}$ the q^l points $(x^l_{\mathrm{new,i}})_{i=1,\ldots,q_l}$ where $k^l_{n_l,\mathrm{adj}}(x,x)$ is the largest.
6: **for** $m=l-1,\ldots,1$ **do**
7: Compute $k^m_{n_m+\sum_{i=m+1}^{l} q^i}(x,x)$ with the new points $\left((x^j_{\mathrm{new,i}})_{i=1,\ldots,q_l}\right)_{j=m+1,\ldots,l}$
8: Generate N^m_{MCMC} samples with respect to $k^m_{n_m+\sum_{i=m+1}^{l} q^i}(x,x)$.
9: Find the N^m cluster centers $(C^m_i)_{i=1,\ldots,N^m}$ such that $N^m = \max_{N\geq q^m}\min_{x\in(C^m_i)_i} k^m_{n_m+\sum_{i=m+1}^{l} q^i}(x,x)$
10: Select from $(C^m_i)_{i=1,\ldots,N^m}$ the q^m points $(x^m_{\mathrm{new,i}})_{i=1,\ldots,q_m}$ where $k^m_{n_m+\sum_{i=m+1}^{l} q^i,\mathrm{adj}}(x,x)$ is the largest.
11: **end for**

In Algorithm 2, $k^l_{n_l+\sum_{i=l+1}^{s} q^i}(x,x)$ corresponds to the kernel of the Gaussian process $Y^l_{n_l}(x)$ conditioned by the observations $\left((x^j_{\mathrm{new,i}})_{i=1,\ldots,q^s}\right)_{j=l+1,\ldots,s}$ when the parameters $(\sigma^2_i)_{i=1,\ldots,l}$ and $(\theta_i)_{i=1,\ldots,l}$ are considered as known (i.e. this corresponds to a liar method). Furthermore, $k^l_{n_l+\sum_{i=l+1}^{s} q^i,\mathrm{adj}}(x,x)$ corresponds to the predictor's variance $k^l_{n_l+\sum_{i=l+1}^{s} q^i}(x,x)$ adjusted with the LOO-CV errors and variances:

$$\begin{aligned} k^l_{n_l+\sum_{i=l+1}^{s} q^i,\mathrm{adj}}(x,x) &= \sum_{i=1}^{l}\sigma^2_{\delta^i+\sum_{i=l+1}^{s} q^i}(x)\left(1+\sum_{j=1}^{n_i}\frac{\left(\varepsilon_{\mathrm{LOO-CV},i}(x^i_j)-\hat{\rho}_{i-1}\varepsilon_{\mathrm{LOO-CV},i-1}(x^i_j)\right)^2}{\sigma^2_{\mathrm{LOO-CV},i}(x^i_j)-\hat{\rho}^2_{i-1}\sigma^2_{\mathrm{LOO-CV},i-1}(x^i_j)}\right) \\ &\times \prod_{j=i}^{l-1}\rho^2_j\prod_{m=1}^{d}\theta^m_i \end{aligned} \tag{24}$$

where $k^1_{n_1+\sum_{i=l+1}^{s} q^i}$ and $\sigma^2_{\delta^i+\sum_{i=l+1}^{s} q^i}(x_{\mathrm{new}})$ are deduced from the equation (15). We note that for the M-H procedures, we use a Gaussian jumping distribution with a standard deviation such that acceptance rate is around 30%.

Furthermore, let us consider the following quantity

$$\mathrm{IMSE}_{\mathrm{red,q}} = \sum_{i=1}^{l} \sum_{r=1,\ldots,q^i} \sigma^2_{\delta^i}(x^i_{\mathrm{new},r}) \prod_{j=i}^{l-1} \rho_j^2 \prod_{m=1}^{d} \theta_i^m \tag{25}$$

We consider the allocation $\{q^1, \ldots, q^l\}$ which solves the following optimization problem:

$$\{q^1, \ldots, q^l\} = \arg \max_{\{q^1,\ldots,q^l\}} \mathrm{IMSE}_{\mathrm{red,q}} \text{ such that } \sum_{j=1}^{l} \sum_{i=j}^{l} q^i t^j = T \tag{26}$$

i.e. we look for the allocation leading the maximal uncertainty reduction. This optimization problem is very complex to solve and a sub-optimal solution will be often considered. Nevertheless, when the number of code levels and the budget T are low (e.g. $s = 2$ in our application) an exhaustive exploration of the allocation $\{q^1, \ldots, q^l\}$ can be performed. We are in that case in the presented application . Furthermore, we note that $\mathrm{IMSE}_{\mathrm{red,q}}$ is a proxy on the IMSE reduction when we add $\left((x^m_{\mathrm{new,i}})_{i=1,\ldots,q_m}\right)_{m=1,\ldots,l}$ at code levels $(y^m(x))_{m=1,\ldots,l}$.

In practical application, the algorithm 2 is reiterated until we reach a prescribed precision or the computational time budget is exhausted.

5 Applications

We compare in this Section the MinIMSE, KleiCrit and AdjMMSE criteria on toy examples and on an application concerning a spherical tank under pressure. We present both the cases of 1 point at-a-time and q points at-a-time sequential kriging. Then, we compare on the tank application, the suggested sequential kriging and co-kriging methods with $s = 2$ levels. The purpose of this section is to emphasize the efficiency of the LOO-CV-based criteria and to highlight that a multi-fidelity analysis can be worthwhile. Finally, for the multi-fidelity sequential co-kriging, we present the allocation of the simulations between the coarse code and the accurate one. We note that for the different examples, we compare the different methods given a prescribed computational time budget.

5.1 Comparison between sequential kriging criteria

In this subsection, the 1 point at-a-time sequential kriging criteria (MinIMSE, KleiCrit, AdjMMSE) are compared on three tabulated functions:

- Ackley's function on $[-2, 2]^2$ [Ackley, 1987]:

$$f(x, y) = -20\exp\left(-0.2\sqrt{\frac{x^2 + y^2}{2}}\right) - \exp\left(\frac{\cos(2\pi x) + \cos(2\pi y)}{2}\right) + 20 + \exp(1)$$

- Shubert's function on $[-2, 2]^2$ [Xian, 2001]:

$$f(x, y) = \left(\sum_{k=1}^{5} k\cos\left((k + 1)x + k\right)\right)\left(\sum_{k=1}^{5} k\cos\left((k + 1)y + k\right)\right)$$

- Michalewicz's function on $[0, \pi]^2$ [Michalewicz, 1992]:

$$f(x,y) = -\sin(x)\left(\sin\left(\frac{x^2}{\pi}\right)\right)^{20} - \sin(y)\left(\sin\left(\frac{y^2}{\pi}\right)\right)^{20}$$

The comparison is performed on a test set $\mathbf{D}_{\text{test}}$ composed of $n_{\text{test}} = 1000$ points uniformly spread on the input parameter space and from 50 different initial experimental design sets. We compare the different methods with respect to the Normalized RMSE:

$$\text{Norm RMSE} = \frac{\sqrt{\sum_{i=1}^{n_{\text{test}}} \left(y_{\text{real}}(x_{\text{test}}^i) - y_{\text{pred}}(x)\right)^2 / n_{\text{test}}}}{\max_{x\in\mathbf{D}_{\text{test}}} y_{\text{real}}(x) - \min_{x\in\mathbf{D}_{\text{test}}} y_{\text{real}}(x)} \tag{27}$$

where $y_{\text{real}}(x)$ is the real value of the output and $y_{\text{pred}}(x)$ the predicted one. The initial experimental design sets are LHS designs of 10 points optimized with respect to the S-optimality [Stocki, 2005]. From these designs, 50 sequential krigings are performed and the convergence of the mean and the quantiles of the Normalized RMSE are computed for the three criteria. Furthermore, after each added point, the parameters of the kriging model are re-estimated with a maximum likelihood method. These estimations are performed thanks to the R library 'DiceKriging' [Roustant et al., 2012].

Figure 1 illustrates the efficiency of the criterion AdjMMSE. Indeed, for the Shubert's and Michalewicz's functions, we see that the accuracy of the 1 point at-a-time kriging with this criterion is significantly better than the one of the others criteria (both in terms of mean and quantiles of the Normalized RMSE). In fact, these functions have the particularity to have non-stationarity in some areas of the input parameter space. Thus, the errors are more important in these locations and the suggested criterion focus the new points on it. Furthermore, the non-stationarity is particularly important for the Shubert's function. For this reason, the IMSE criterion performed very bad in that case. Indeed, this criterion is efficient for stationary functions (i.e. when the predictor's MSE well predicts the real model errors). In contrast, the jackknife predictor's MSE provided by the criterion KleiCrit manages to catch this non-stationarity and it performs better than the IMSE criterion. Moreover, we see that the performance of the AdjMMSE and IMSE criteria are equivalent for the Ackley's function. We note that the Ackley's function is perfectly stationary and the errors have thus the same order of magnitude over the input parameter space.

These examples illustrate the fact that our criterion is more efficient than the other criteria when the functions are non-stationary and it remains efficient even in the cases where the functions are stationary (its efficiency is equivalent to the one of the IMSE criterion).

5.2 Spherical tank under internal pressure example

In this section, we deal with an example about a spherical tank under internal pressure. We are interested in the von Mises stresses on the three points labeled in figure 2. Indeed, we want to prevent from material yielding which occurs when the von Mises stress reaches the critical yield strength.

The system illustrated in figure 2 depends on the following parameters:

- $P\,(MPa) \in [30, 50]$: the value of the internal pressure.
- $R_{int}\,(mm) \in [1500, 2500]$: the length of the internal radius of the shell.
- $T_{shell}\,(mm) \in [300, 500]$: the thickness of the shell.

- $T_{cap}\,(mm) \in [100, 300]$: the thickness of the cap.
- $E_{shell}\,(GPa) \in [63, 77]$: the Young's modulus of the shell material.
- $E_{cap}\,(GPa) \in [189, 231]$: the Young's modulus of the cap material.
- $\sigma_{y,shell}\,(MPa) \in [200, 300]$: the yield stress of the cap material.
- $\sigma_{y,cap}\,(MPa) \in [400, 800]$: the yield stress of the cap material.

The accurate code output $y^2(x)$ is the value of the von Mises stress provided by an Aster finite elements code (`http://www.code-aster.org`) modelling the system presented in figure 2. We use the notation $x = (P, R_{int}, T_{shell}, T_{cap}, E_{shell}, E_{cap}, \sigma_{y,shell}, \sigma_{y,cap})$. We note that the material properties of the shell correspond to high quality aluminum and the ones of the cap corresponds to steel from classical to high quality. Then, the coarse code output $y^1(x)$ is the value of the von Mises stress given by the 1D simplification of the tank (28) (it corresponds to a perfect spherical tank under pressure, i.e. without cap).

$$y^1(x) = \frac{3}{2}\frac{(R_{int} + T_{shell})^3}{(R_{int} + T_{shell})^3 - R_{int}^3}P \tag{28}$$

According to equation (28), the actual input dimension of $y^1(x)$ is three (it depends only on P, R_{int} and T_{shell}) while a sensitivity analysis performed with a Sobol decomposition gives that the accurate code depends essentially on four parameters (P, R_{int}, T_{shell} and T_{cap}). Furthermore, the response is highly stationary. Therefore, only few points are necessary to well predict the output of the code. For these reasons, we can start the sequential strategies from an initial experimental design set with only 10 points.

Thus, for the different comparisons, we use a S-optimal LHS design $\mathbf{D}^2$ of 10 points for the code $y^2(x)$. For the coarse code $y^1(x)$, we start with a design $\mathbf{D}^1$ of 20 points. It is created with the following procedure. First, we create a S-Optimal design $\tilde{\mathbf{D}}^1$ of 20 points. Second, we remove from $\tilde{\mathbf{D}}^1$ the 10 points that are the closest to those of $\mathbf{D}^2$. Finally, $\mathbf{D}^1$ is the concatenation of $\mathbf{D}^2$ and $\tilde{\mathbf{D}}^1$ (this procedure ensures the nested property $\mathbf{D}^2 \subset \mathbf{D}^1$). We note that the CPU time is around 1 minute for the accurate code and 10^{-8} seconds for the coarse code. Nevertheless, to be in a more realistic case, we consider that the CPU time ratio between $y^2(x)$ and $y^1(x)$ equals $B_{2/1} = 10$. Furthermore, each sequential procedure is performed with 40 different initial design sets. Then, the mean and the quantiles of probabilities 90% and 10% of the empirical Normalized MSE are computed from a test set composed of 1000 points uniformly spread on the input parameter space. Finally, for the M-H procedure, we use a Gaussian jumping distribution such that the acceptance rate is around 30% and we set $N_{\text{MCMC}} = 50000$ (we use 5 000 samples for the the burn-in procedure of the M-H method, see [Robert and Casella, 2004]). For the M-H procedure, we use the package R CRAN mcmc. We note that after each added points, the parameters of the kriging or co-kriging models are re-estimated with a maximum likelihood method.

The remainder of this section is organized as follows. First we compare the MSE of the 1 point at-a-time sequential kriging with the one of the $q = 5$ points at-a-time one. Second, we compare for a given CPU time budget the sequential kriging and cokriging strategies. In the forthcoming developments, the response $i = 1, 2, 3$ refers to the value of the von Mises stress at point i on figure 2.

5.2.1 Comparison between sequential kriging criteria

Figure 3 compares the different criteria of the 1 point at-a-time and the $q = 5$ points at-a-time sequential kriging. We see that the criteria MinIMSE and AdjMMSE give equivalent values for the MSE for the 1 point at-a-time procedure and they perform better than the KleiCrit criterion. There are equivalent since the output $y^2(x)$ is perfectly stationary. Nevertheless, the criterion AdjMMSE is the most efficient for the $q = 5$ points at-a-time procedure. Indeed, the 5 points provided by a liar method with the MinIMSE criterion are not necessarily those which maximize the reduction of the IMSE. The method suggested in section 4.2 seems to give better solution.

5.2.2 Comparison between kriging and co-kriging sequential analysis

In this section, we compare the sequential kriging strategy with the sequential co-kriging with respect to the AdjMMSE criterion. Figure 4 gives the convergence of the empirical normalized MSE for the response 1. We see that the sequential co-kriging performs better than the kriging one. Furthermore, at the beginning of the method, the proportion of runs for the accurate code is very low. Indeed, the coarse code and the accurate code are extremely correlated for this response (around 99%) and thus, during the sequential strategy, the bias between the two codes is well estimated. Then, when the coarse code is well approximated, the sequential strategy starts to run the accurate one (for a CPU time around 500).

Figure 5 gives the convergence of the errors for the response 2. For this response, the correlation between the coarse and the accurate code is around 80%. Therefore, the proportion of runs for the accurate code determined by the sequential strategy is more important than in Figure 4. Furthermore, we see that this proportion increases with the CPU time. It means that the sequential co-kriging improves the approximation of the coarse code at the beginning of the procedure and then focuses on the accurate code. As a result, we see that the sequential co-kriging strategy is substantially better than the kriging one.

Figures 4 and 5 illustrate the efficiency of the sequential co-kriging when the coarse code bring information on the accurate code. For the response 3, the coarse code is weakly correlated with the accurate code (around 45%). This is due to the fact that the coarse code models the von Mises stress in a perfect spherical tank whereas the response 3 corresponds to the one in the cap. Figure 6 shows that in this case, the sequential co-kriging model manages to determine that the coarse code is not worth being simulated. Indeed, the proportion of runs for the accurate code is very low. Furthermore, it shows that the co-kriging sequential design performs as well as the kriging one when the coarse code is non-informative.

Finally, Figure 7 shows the efficiency of the (q^1, q^2) at-a-time sequential co-kriging. We set in the algorithm 2 that $T = q^1 + q^2 + 10q^2 = 120$ where the CPU time of the coarse code is 1 and the one of the accurate code is 10. For the the sequential kriging, we use a $q = 10$ at-a-time sequential procedure. Furthermore, Figure 7 shows that at the beginning of the procedure, the sequential co-kriging focuses on the approximation of the coarse code whereas at the end the number of runs for the accurate code is maximal. We note that the allocation of runs for the accurate code in figure 7 agrees with the proportion of runs given in Figure 5.

The results of the sequential co-kriging on the different responses show that the criterion suggested in section 4.1 performs very well. Indeed, it is always better than the sequential kriging when the coarse code is informative and its performance is equivalent to it when the coarse code is not useful. Furthermore, the different proportions of runs for the accurate code

emphasizes that the criterion accurately determines the contribution of each code to the total model error and the optimal run allocation between the accurate and the coarse codes.

6 Conclusion

This paper deals with sequential strategies for kriging and co-kriging models. The kriging model is used to approximate the output of a complex computer code and the co-kriging one allows to improve this approximation thanks to coarse versions of the code.

First, we have presented classical sequential criteria for the kriging model and we have suggested another criterion based on the Leave-One-Out cross validation errors. This criterion has allowed us to set the new observations at locations where the model error is important. The examples presented in the last section have highlighted the efficiency of the suggested criterion. Indeed, for non-stationary functions, it provides results significantly better than classical criteria and for stationary ones its performance is equivalent to them. We have also emphasized the performance of the suggested criterion on a real application. Furthermore, we show in the application that when the simulations can be performed in parallel, our method has performed better.

Second, we have presented the extension of our criterion to multi-fidelity co-kriging models. We have shown in the application that performing a multi-fidelity sequential co-kriging is worthwhile when the coarse code versions are informative (i.e. highly correlated with the accurate code). Furthermore, a strength of the proposed approach is that it performs as well as a sequential kriging when the coarse code versions are not informative. In fact, the proposed extension takes into account the contribution of each code to the total predictor's mean squared errors and it determines the best run allocation between accurate and coarse code versions given a CPU time budget.

7 Acknowledgements

The author particularly thanks Professor Josselin Garnier for supervising his work and for his fruitful guidance. He is also grateful to Dr. Gilles Defaux for providing the application.

References

[Ackley, 1987] Ackley, D. H. (1987). *A connectionist machine for genetic hillclimbing*. Kluwer Academic Publishers, Boston.

[Bates et al., 1996] Bates, R. A., Buck, R., Riccomagno, E., and Wynn, H. (1996). Experimental design and observation for large systems. *Journal of the Royal Statistical Society, Series B*, 58 (1):77–94.

[Bect et al., 2012] Bect, J., Ginsbourger, D., Li, L., Picheny, V., and Vazquez, E. (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22:773–793.

[Craig et al., 1998] Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A. (1998). Constructing partial prior specifications for models of complex physical systems. *Applied Statistics*, 47:37–53.

[Cumming and Goldstein, 2009] Cumming, J. A. and Goldstein, M. (2009). Small sample bayesian designs for complex high-dimensional models based on information gained using fast approximations. *Technometrics*, 51:377–388.

[Dubourg et al., 2011] Dubourg, V., Sudret, B., and Bourinet, J.-M. (2011). Reliability-based design optimization using kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44 - 5:673–690.

[Dubrule, 1983] Dubrule, O. (1983). Cross validation of kriging in a unique neighborhood. *Mathematical Geology*, 15:687–699.

[Fang et al., 2006] Fang, K.-T., Li, R., and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*. Chapman & Hall - Computer Science and Data Analysis Series, London.

[Forrester et al., 2007] Forrester, A. I. J., Sobester, A., and Keane, A. J. (2007). Multi-fidelity optimization via surrogate modelling. *Proc. R. Soc. A*, 463:3251–3269.

[Ginsbourger et al.,] Ginsbourger, D., Le Riche, R., and Carraro, L. *Kriging is well-suited to parallelize optimization. In Computational Intelligence in Expensive Optimization Problems*, pages 131–162. Adaptation Learning and Optimization. Springer, Berlin.

[Higdon et al., 2004] Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004). Combining field data and computer simulation for calibration and prediction. *SIAM Journal on Scientific Computing*, 26:448–466.

[Huang et al., 2006] Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A. (2006). Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32:369–382.

[Jones et al., 1998] Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Biometrika*, 13:455–492.

[Kennedy and O'Hagan, 2000] Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87:1–13.

[Kennedy and O'Hagan, 2001] Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B*, 63:425–464.

[Kleijnen and van Beers, 2004] Kleijnen, J. and van Beers, W. (2004). Application-driven sequential designs for simulation experiments: Kriging metamodelling. *Journal of the Operational Research Society*, 55:876–883.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *In: Le Cam J LM Neyman (ed) Proc. 5th Berkeley Symp. on Math. Stat. & Prob., University of California Press, Berkeley, CA*, 1:281–297.

[Michalewicz, 1992] Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, New York.

[Picheny et al., 2012] Picheny, V., Ginsbourger, D., Richet, Y., and Caplin, G. (2012). Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*.

[Picheny et al., 2010] Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R. T., and Nam-Ho, K. (2010). Adaptative designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132:071008–1 – 071008–9.

[Qian and Wu, 2008] Qian, P. Z. G. and Wu, C. F. J. (2008). Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50:192–204.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge.

[Reese et al., 2004] Reese, C. S., Wilson, A. G., Hamada, M., Martz, H. F., and Ryan, K. J. (2004). Integrated analysis of computer and physical experiments. *Technometrics*, 46:153–164.

[Robert and Casella, 2004] Robert, C. and Casella, G. (2004). *Monte Carlo statistical methods (2nd edition)*. Springer Series in Statistics, Springer Verlag, New York.

[Roustant et al., 2012] Roustant, O., Ginsbourger, D., and Deville, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55.

[Sacks et al., 1989] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, 4:409–423.

[Santner et al., 2003] Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

[Stein, 1999] Stein, M. L. (1999). *Interpolation of Spatial Data*. Springer Series in Statistics, New York.

[Stocki, 2005] Stocki, R. (2005). A method to improve design reliability using optimal latin hypercube sampling. *Computer Assisted Mechanics and Engineering Sciences*, 12:87–105.

[van Beers and Kleijnen, 2008] van Beers, W. and Kleijnen, J. (2008). Customized sequential designs for random simulation experiments: Kriging metamodelling and bootstrapping. *European journal of operational research*, 186:1099–1113.

[Xian, 2001] Xian, L. (2001). Finding global minima with a computable filled function. *Journal of Global Optimization*, 19:191–204.

[Xiong and Qian, 2012] Xiong, S. and Qian, Peter Z. G.and Jeff Wu, C. F. (2012). Sequential design and analysis of high-accuracy and low-accuracy computer codes. *Technometrics*, DOI:10.1080/00401706.2012.723572.

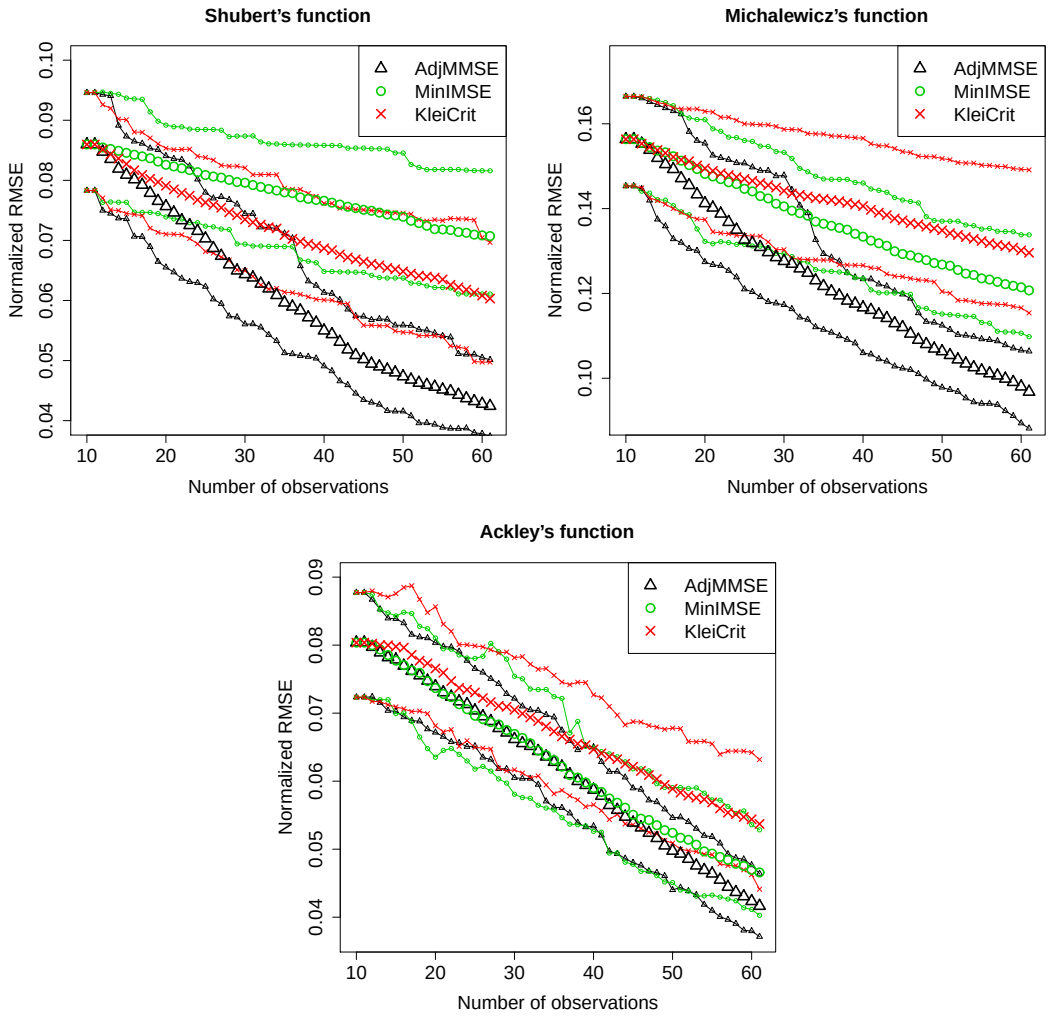

Figure 1: Comparison between 1 point at-a-time sequential kriging criteria on toy examples. The bold triangles represent the mean of the empirical MSE for the AdjMMSE criterion, the bold circles represent it for the MinIMSE criterion and the bold Crosses represent it for the KleiCrit criterion. Furthermore, the thin triangles, circles and crosses represent the quantiles of probabilities 10% and 90% of the empirical MSE.

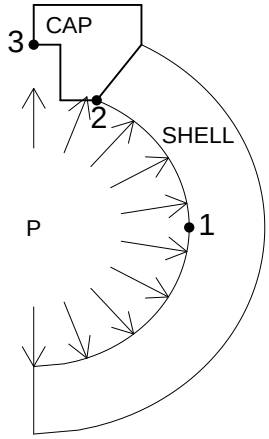

Figure 2: Scheme of the spherical tank under pressure.

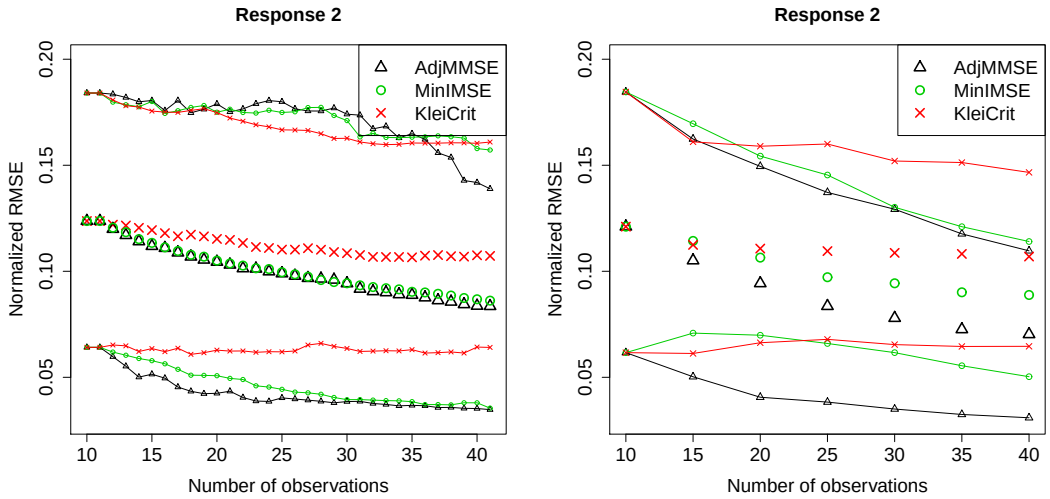

Figure 3: Comparison between 1 point at-a-time sequential kriging criteria (on left) and $q = 5$ points at-a-time sequential kriging criteria (on right) on the spherical tank example. The bold triangles represent the mean of the empirical MSE for the AdjMMSE criterion, the bold circles represent it for the MinIMSE criterion and the bold Crosses represent it for the KleiCrit criterion. Furthermore, the thin triangles, circles and crosses represent the quantiles of probabilities 10% and 90% of the empirical MSE.

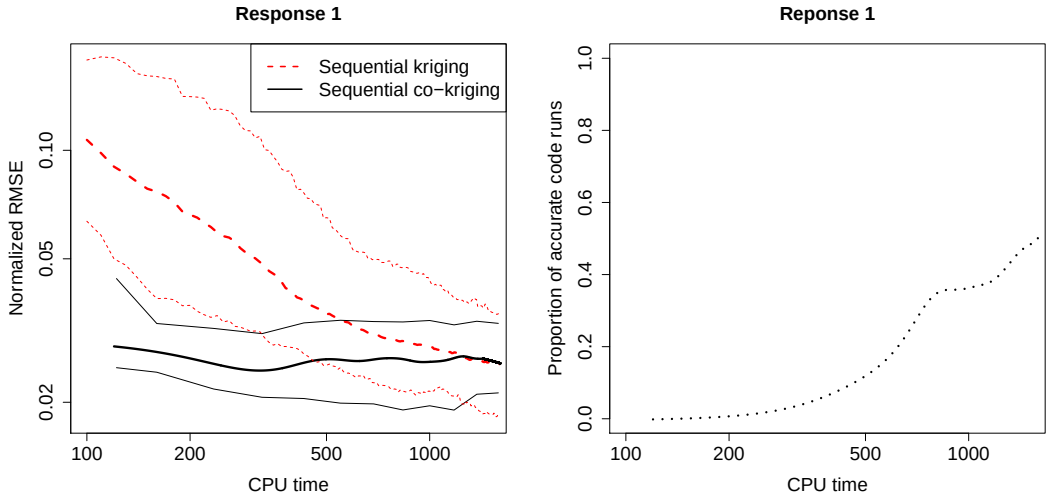

Figure 4: Comparison between 1 point at-a-time sequential kriging and co-kriging on the response 1 of the spherical tank example with respect to the AdjMMSE criterion (on the left). The thick dashed line represents the mean of the empirical MSE for the sequential kriging and the thick solid line represents it for the sequential co-kriging. The thin lines represent the quantiles of probabilities 10% and 90% of the empirical MSE. Figure on the right represents the proportion of runs allocated to the accurate code.

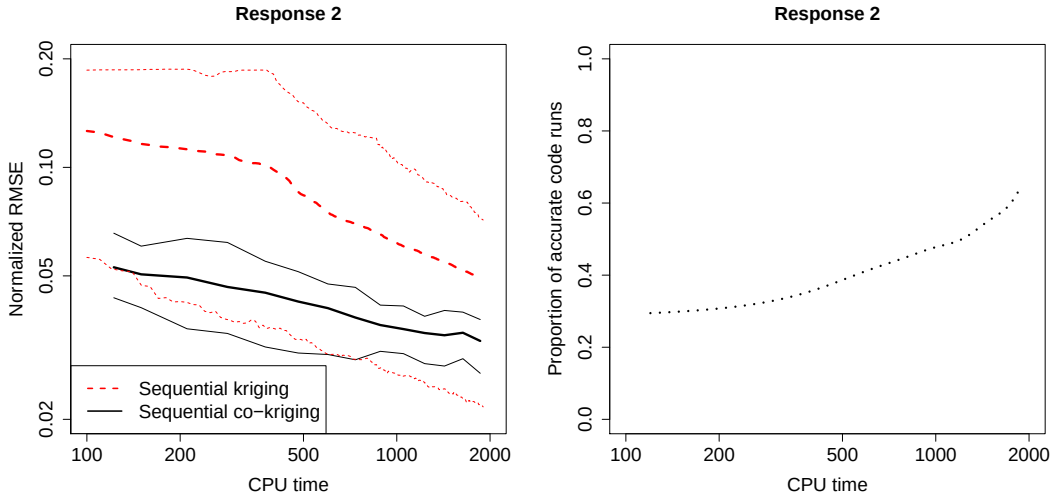

Figure 5: Comparison between 1 point at-a-time sequential kriging and co-kriging on the response 2 of the spherical tank example with respect to the AdjMMSE criterion (on the left). The thick dashed line represents the mean of the empirical MSE for the sequential kriging and the thick solid line represents it for the sequential co-kriging. The thin lines represent the quantiles of probabilities 10% and 90% of the empirical MSE. Figure on the right represents the proportion of runs allocated to the accurate code.

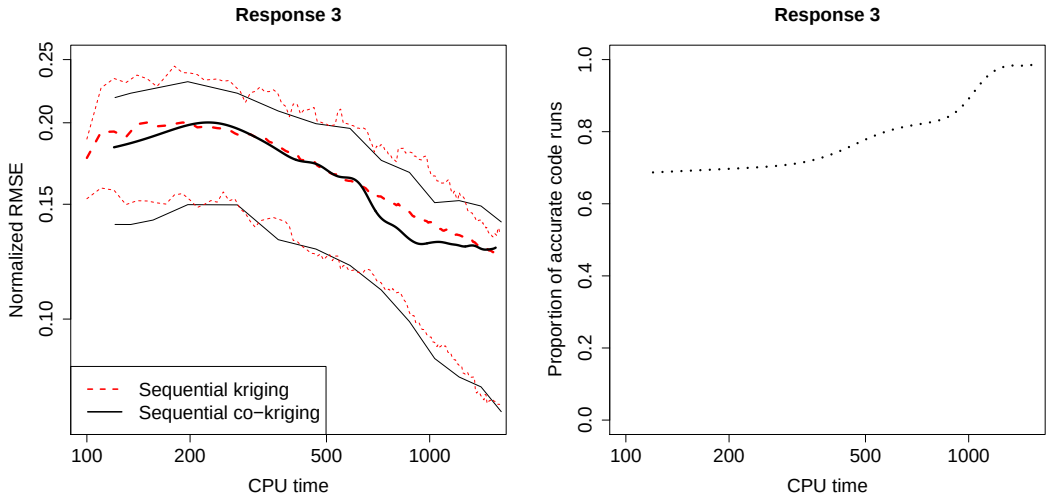

Figure 6: Comparison between 1 point at-a-time sequential kriging and co-kriging on the response 3 of the spherical tank example with respect to the AdjMMSE criterion (on the left). The thick dashed line represents the mean of the empirical MSE for the sequential kriging and the thick solid line represents it for the sequential co-kriging. The thin lines represent the quantiles of probabilities 10% and 90% of the empirical MSE. Figure on the right represents the proportion of runs allocated to the accurate code.

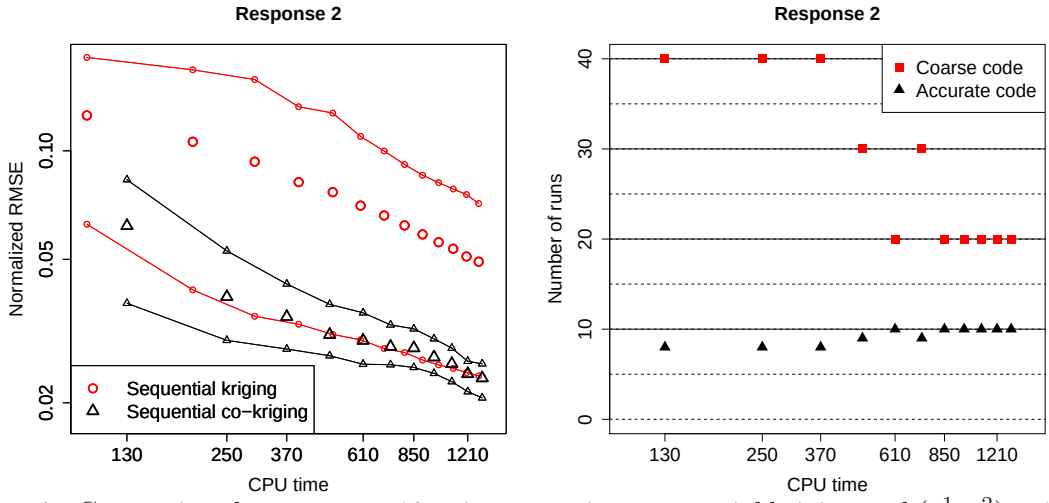

Figure 7: Comparison between $q = 10$ points at-a-time sequential kriging and (q^1, q^2) points at-a-time sequential co-kriging. On the left, the bold circles represents the mean of the empirical MSE for the sequential kriging and the bold triangles represent the one of the sequential co-kriging. Furthermore, the thin circles and triangles represent the quantiles of probabilities 10% and 90% of the empirical MSE. On the right, the squares represent the median number of runs for the coarse code during the sequential co-kriging and the triangles represent it for the accurate code.