



HAL
open science

Bayesian analysis of hierarchical multi-fidelity codes.

Loic Le Gratiet

► **To cite this version:**

Loic Le Gratiet. Bayesian analysis of hierarchical multi-fidelity codes.. 2012, pp.RSPA-2011-0742.
hal-00654716v1

HAL Id: hal-00654716

<https://hal.science/hal-00654716v1>

Submitted on 22 Dec 2011 (v1), last revised 24 Sep 2012 (v2)

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.

Bayesian analysis of hierarchical multi-fidelity codes.

Loic Le Gratiet
CEA, DAM, DIF, F-91297 Arpajon, France
loic.le-gratiet@cea.fr

December 22, 2011

1 Abstract

This paper deals with the Gaussian process based approximation of a code which can be run at different levels of accuracy. This co-kriging method allows us to improve a surrogate model of a complex computer code using fast approximations of it. In particular, we focus on the case of a large number of code levels on the one hand and on a Bayesian approach when we have 2 levels on the other hand. Moreover, based on a Bayes linear formulation, an extension of the universal kriging equations are provided for the co-kriging model. We also address the problem of nested space-filling design for multi-fidelity computer experiments and we provide a significant simplification of the computation of the co-kriging cross-validation equations. A hydrodynamic simulator example is used to illustrate the comparison Bayesian versus non-Bayesian co-kriging. A thermodynamic example is used to illustrate the comparison between 2-level and 3-level co-kriging.

Keywords: surrogate models, co-kriging, multi-fidelity computer experiment, Bayesian analysis, cross-validation, nested space-filling design.

2 Introduction

Large computer codes are widely used in science and engineering to study physical systems since real experiments are often costly and sometimes impossible. Nevertheless, simulations can sometimes be costly and time-consuming as well. In this case, conception based on an exhaustive exploration of the input space of the code is generally impossible under reasonable time constraints. Therefore, a mathematical approximation of the output of the code - also called surrogate or metamodel - is often built with a few simulations to represent the real system.

Gaussian Process regression is a particular class of surrogate which makes the assumption that prior beliefs about the code can be modelled by a Gaussian Process. We focus here on this metamodel and on its extension to multiple response models. The reader is referred to [Santner, Williams & Notz (2003)] and [Rasmussen & Williams (2006)] for further detail about Gaussian Process models.

Actually, a computer code can often be run at different levels of complexity and a hierarchy of levels of code can hence be obtained. The aim of this paper is to study the use of several levels of a code to predict the output of a costly computer code.

A first metamodel for multi-level computer codes was built by [Kennedy & O'Hagan (2000)]

using a spatial stationary correlation structure. This multi-stage model is a particular case of co-kriging which is a well known geostatistical method. Then, [Qian et al. (2006)] built an extension to this model in a case of non spatial stationarity and [Forrester, Sobester & Keane (2007)] went into more detail about the estimation of the model parameters. Furthermore, Forrester *et al.* presented the use of co-kriging for multi-fidelity optimization based on the EGO (Efficient Global Optimization) algorithm created by [Jones, Schonlau & Welch (1998)]. A Bayesian approach was also proposed by [Qian & Wu (2008)] which was computationally expensive and does not provide explicit formulas for the joint distribution of the parameters.

This paper presents a new approach to estimate the parameters of the model which is effective in the case of non-spatial stationarity and when many levels of codes are available. Furthermore, this approach allows us to consider prior information in the estimation of the parameters. We also address the problem of the inversion of the co-kriging covariance matrix when the number of levels is large. A solution to this problem is provided which shows that the inverse can be easily calculated. Moreover, it is known that with a non-Bayesian approach, the variance of the predictive distribution may be underestimated [Kennedy & O’Hagan (2000)]. This paper suggests a Bayesian modelling different from the one presented in [Qian & Wu (2008)] which provides an explicit representation of the joint distribution for the parameters and avoids prohibitive implementation. Furthermore, thanks to the joint density of the parameters, we can deduce closed form formulas for the mean and covariance of the posterior predictive distribution. Due to their similarities with the universal kriging equations, we call these formulas the universal co-kriging equation. Then, we suggest a new experimental design strategy for multi-fidelity computer experiments which is more flexible than the previous ones and not time-consuming. Finally, we present a fast method to compute the cross-validation equations of the co-kriging surrogate model.

3 Building a surrogate model based on a hierarchy of s levels of code

Let us assume that we have s levels of code $z_1(x), \dots, z_s(x)$, $x \in \mathbb{R}^d$, $d > 0$. For all $t = 1, \dots, s$ the t^{th} scalar output $z_t(x)$ is modelled by $z_t(x) = Z_t(x, \omega)$ where $Z_t(x, \omega)$, $\omega \in \Omega$ is a realization of the Gaussian process $Z_t(x)$. We will introduce below a consistent set of hypotheses so that the joint process $(Z_t(x))_{x \in \mathbb{R}^d, t=1, \dots, s}$ is Gaussian given a certain set of parameters.

[Kennedy & O’Hagan (2000)] suggest an autoregressive model to build a metamodel based on a multi-level computer code. Hence, we have a hierarchy of s levels of code - from the less accurate to the most accurate - and for each level, the conditional distribution of the Gaussian process $Z_t(x)$ knowing $Z_1(x), \dots, Z_{t-1}(x)$ is entirely determined by $Z_{t-1}(x)$. Let us introduce here the mathematical formalism that we will use in this paper.

$Q \subset \mathbb{R}^d$ is a compact subset of \mathbb{R}^d called the input space or the domain of interest. For $t = 1, \dots, s$, $D_t = \{x_1^{(t)}, \dots, x_{n_t}^{(t)}\}$ is the experimental design set at level t containing n_t points in Q . Let $\mathcal{Z}_t = Z_t(D_t) = (Z_t(x_1^{(t)}), \dots, Z_t(x_{n_t}^{(t)}))^T$ be the random Gaussian vector containing the values of $Z_t(x)$ for $x \in D_t$. Let $\mathcal{Z} = (\mathcal{Z}_1^T, \dots, \mathcal{Z}_s^T)^T$ be the Gaussian random vector containing the values of the processes $(Z_t(x))_{t=1, \dots, s}$ at the points of the design sets $(D_t)_{t=1, \dots, s}$. We assume here that the code output is observed without measurement error. The column vector of responses is written $z = (z_1^T, \dots, z_s^T)^T$, where $z_t = (z_t(x_1^{(t)}), \dots, z_t(x_{n_t}^{(t)}))^T$ is the output vector for the level t and T stands for the transpose.

If we consider $Z_s(x)$, the Gaussian process modelling the most accurate code, we want to determine the predictive distribution of $Z_s(x_0)$, $x_0 \in Q$ given $\mathcal{Z} = z$, *i.e.* the following conditional distribution: $[Z_s(x_0)|\mathcal{Z} = z]$.

We assume the Markow property introduced in [Kennedy & O'Hagan (2000)]:

$$\text{Cov}(Z_t(x), Z_{t-1}(x')|Z_{t-1}(x)) = 0 \quad \forall x \neq x' \quad (1)$$

This means that if $Z_{t-1}(x)$ is known, then nothing more can be learnt about $Z_t(x)$ from any other run of the cheaper code $Z_{t-1}(x')$ for $x' \neq x$. This assumption leads to the following autoregressive model:

$$Z_t(x) = \rho_{t-1}(x)Z_{t-1}(x) + \delta_t(x) \quad t = 2, \dots, s \quad (2)$$

where $\delta_t(x)$ is a Gaussian process independent of $Z_{t-1}(x), \dots, Z_1(x)$ and $\rho_{t-1}(x)$ represents a scale factor between $Z_t(x)$ and $Z_{t-1}(x)$. We assume that $\rho_{t-1}(x)$, $t = 2, \dots, s$ is a linear regression function:

$$\forall t = 2, \dots, s \quad \rho_{t-1}(x) = f_{\rho_{t-1}}(x)^T \beta_{\rho_{t-1}} \quad (3)$$

where $f_{\rho_{t-1}}(x) = (f_{\rho_{t-1}}^1(x), \dots, f_{\rho_{t-1}}^{q_{t-1}}(x))^T$ is a vector of q_{t-1} regression functions - generally including the constant function : $x \in Q \rightarrow 1$ - and $\beta_{\rho_{t-1}} \in \mathbb{R}^{q_{t-1}}$.

Conditioning on parameters σ_t , β_t and θ_t , $\delta_t(x)$ is assumed to be a Gaussian process with mean $f_t(x)^T \beta_t$, where $f_t(x)$ is a p_t -dimensional vector of regression functions, and with a covariance function of the form $c_t(x, x') = \text{cov}(\delta_t(x), \delta_t(x')) = \sigma_t^2 r_t(x - x'; \theta_t)$, where σ_t^2 is the variance of the Gaussian process and θ_t are the hyper parameters of the correlation function r_t . Moreover, conditioning on parameters σ_1 , β_1 and θ_1 , the simplest code $Z_1(x)$ is modelled as a Gaussian process with mean $f_1(x)^T \beta_1$ and with covariance function $c_1(x, x') = \sigma_1^2 r_1(x - x'; \theta_1)$. With this consistent set of hypotheses, the joint process $(Z_1(x), \dots, Z_t(x))_{x \in Q, t=1, \dots, s}$ given $\sigma^2 = (\sigma_i^2)_{i=1, \dots, t}$, $\theta = (\theta_i)_{i=1, \dots, t}$, $\beta = (\beta_i)_{i=1, \dots, t}$ and $\beta_\rho = (\beta_{\rho_{i-1}})_{i=2, \dots, t}$, is Gaussian with mean:

$$\mathbb{E}[Z_t(x)|\sigma^2, \theta, \beta, \beta_\rho] = h'_t(x)^T \beta \quad (4)$$

$$h'_t(x)^T = \left(\left(\prod_{i=1}^{t-1} \rho_i(x) \right) f_1^T(x), \left(\prod_{i=2}^{t-1} \rho_i(x) \right) f_2^T(x), \dots, \rho_{t-1}(x) f_{t-1}^T(x), f_t^T(x) \right) \quad (5)$$

and covariance:

$$\text{cov}(Z_t(x), Z_t(x')|\sigma^2, \theta, \beta, \beta_\rho) = \sum_{j=1}^t \sigma_j^2 \left(\prod_{i=j}^{t-1} \rho_i^2(x) \right) r_j(x - x'; \theta_j) \quad (6)$$

For each level $t = 2, \dots, s$, the experimental design D_t is assumed to be such that $D_t \subseteq D_{t-1}$. Note that this assumption is not necessary but allows us to have closed form expression for the parameter estimation formula. Furthermore, we denote by $R_t(D_k, D_l)$ the correlation matrix between points in D_k and D_l , $1 \leq k, l \leq s$. $R_t(D_k, D_l)$ is a $(n_k \times n_l)$ matrix with (i, j) entry given by:

$$[R_t(D_k, D_l)]_{i,j} = r_t(x_i^{(k)} - x_j^{(l)}; \theta_t) \quad 1 \leq i \leq n_k \quad 1 \leq j \leq n_l \quad \forall x_i^{(k)} \in D_k$$

We will use the notation: $R_t(D_k) = R_t(D_k, D_k)$.

[Kennedy & O'Hagan (2000)] have presented the case where $\forall t \in [2, s]$, $\rho_{t-1}(x) = \rho_{t-1}$ are constant and [Qian et al. (2006)] the case where $f_t(x) = 1$ and $t = 2$, *i.e.* the case of 2 levels. Here, we will consider the general model presented in equations (2) and (3). We will also propose a new approach to estimate the coefficients $(\beta_t, \beta_{\rho_{t-1}})_{t=2, \dots, s}$ based on a Bayesian estimation, which allows us to get information about their uncertainties. In the following section, we describe the case of 2 levels of code where the scaling coefficient ρ is constant and then we will extend it for s levels in Section 9. The general case in which ρ depends on x , is addressed in Appendix A.

4 Building a model with 2 levels of code

Let us assume that we have 2 levels of code $z_2(x)$ and $z_1(x)$. From the previous section we assume that:

$$\begin{cases} Z_2(x) = \rho Z_1(x) + \delta(x), & x \in Q \\ (Z_1(x))_{x \in Q} \perp (\delta(x))_{x \in Q} \end{cases} \quad (7)$$

The goal of this section is to build a surrogate model for $Z_2(x)$ given the observations $\mathcal{Z} = z$ with an uncertainty quantification. The strategy is the following one. In Subsection 4.1 we describe the statistical distribution of the output $Z_2(x_0)$ at a new point x_0 given the parameters (β_1, β_2, ρ) , (σ_1^2, σ_2^2) and (θ_1, θ_2) and the observations z . In Subsection 4.2 we describe the Bayesian estimation of the parameters (β_1, β_2, ρ) and (σ_1^2, σ_2^2) given the observations. As pointed out at the end of Subsection 4.2 the hyper-parameters (θ_1, θ_2) are estimated using a concentrated restricted log-likelihood method.

4.1 Conditional distribution of the output

For a point $x_0 \in Q$ we determine in this subsection the distribution of $[Z_2(x_0) | \mathcal{Z} = z, (\beta_1, \beta_2, \rho), (\sigma_1^2, \sigma_2^2), (\theta_1, \theta_2)]$. Standard results for normal distribution give that:

$$[Z_2(x_0) | \mathcal{Z} = z, (\beta_1, \beta_2, \rho), (\sigma_1^2, \sigma_2^2), (\theta_1, \theta_2)] \sim \mathcal{N}(m_{Z_2}(x_0), s_{Z_2}^2(x_0)) \quad (8)$$

with mean function:

$$m_{Z_2}(x) = h'(x)^T \beta + t(x)^T V^{-1} (z - H\beta) \quad (9)$$

and variance:

$$s_{Z_2}^2(x) = \rho^2 \sigma_1^2 + \sigma_2^2 - t(x)^T V^{-1} t(x) \quad (10)$$

where we have denoted:

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \quad z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

and where H is defined by:

$$H = \begin{pmatrix} f_1^T(x_1^{(1)}) & 0 \\ \vdots & \vdots \\ f_1^T(x_{n_1}^{(1)}) & 0 \\ \rho f_1^T(x_1^{(2)}) & f_2^T(x_1^{(2)}) \\ \vdots & \vdots \\ \rho f_1^T(x_{n_2}^{(2)}) & f_2^T(x_{n_2}^{(2)}) \end{pmatrix} = \left(\begin{array}{c|c} F_1(D_1) & 0 \\ \hline \rho F_1(D_2) & F_2(D_2) \end{array} \right)$$

with the notation $F_i(D_j) = \begin{pmatrix} f_i^T(x_{n_1}^{(j)}) \\ \vdots \\ f_i^T(x_{n_j}^{(j)}) \end{pmatrix}$. Furthermore, we have:

$$\begin{aligned} t(x)^T &= \text{Cov}(Z_2(x), \mathcal{Z}) \\ &= (\rho\sigma_1^2 R_1(\{x\}, D_1), \rho^2\sigma_1^2 R_1(\{x\}, D_2) + \sigma_2^2 R_2(\{x\}, D_2)) \\ h'(x) &= (\rho f_1^T(x), f_2^T(x)) \end{aligned} \quad (11)$$

The covariance matrix V of the Gaussian vector $\mathcal{Z} = \begin{pmatrix} \mathcal{Z}_1 \\ \mathcal{Z}_2 \end{pmatrix}$ can be written :

$$V = \begin{pmatrix} \sigma_1^2 R_1(D_1) & \rho\sigma_1^2 R_1(D_1, D_2) \\ \rho\sigma_1^2 R_1(D_2, D_1) & \rho^2\sigma_1^2 R_1(D_2) + \sigma_2^2 R_2(D_2) \end{pmatrix} \quad (12)$$

4.2 Bayesian estimation of the parameters with 2 levels of code

In this Subsection, we describe the estimation of the parameters $(\beta_1, \beta_2, \rho, \sigma_1^2, \sigma_2^2, \theta_1, \theta_2)$ for the 2-level model given the observations $\mathcal{Z} = z$. Due to the conditional independence between $Z_1(x)$ and $\delta(x)$, it is possible to estimate separately the parameters $(\beta_1, \sigma_1^2, \theta_1)$ and $(\beta_2, \rho, \sigma_2^2, \theta_2)$. We first describe the estimations of (β_1, σ_1^2) given θ_1 and $(\beta_2, \sigma_2^2, \rho)$ given θ_2 , which can be obtain in closed forms. We then describe how to estimate θ_1 and θ_2

Firstly, we consider the parameters $(\beta_1, \sigma_1^2, \theta_1)$. We choose prior distribution with the following form :

$$p(\beta_1 | \sigma_1^2, \theta_1) \propto 1 \quad p(\sigma_1^2) \propto \frac{1}{\sigma_1^2} \quad (13)$$

These priors are non-informative, they correspond to the ‘‘Jeffreys priors’’ [Jeffreys (1961)]. Considering the likelihood:

$$p(z_1 | \beta_1, \sigma_1^2, \theta_1) = \frac{1}{(2\pi\sigma_1^2)^{\frac{n_1}{2}} \sqrt{\det(R_1(D_1))}} e^{-\frac{(z_1 - F_1(D_1)\beta_1)^T R_1(D_1)^{-1} (z_1 - F_1(D_1)\beta_1)}{2\sigma_1^2}}$$

and the Bayes formula, the posterior distribution of $[\beta_1 | z_1, \sigma_1^2, \theta_1]$ is :

$$[\beta_1 | z_1, \sigma_1^2, \theta_1] \sim \mathcal{N}_{p_1} \left([F_1^T R_1(D_1)^{-1} F_1]^{-1} [F_1^T R_1(D_1)^{-1} z_1], [F_1^T \frac{R_1(D_1)^{-1}}{\sigma_1^2} F_1]^{-1} \right) \quad (14)$$

Then, using the Bayes formula:

$$p(\sigma_1^2 | y) \propto \frac{p(y | \sigma_1^2, \beta) p(\beta | \sigma_1^2) p(\sigma_1^2)}{p(\beta | \sigma_1^2, y)}$$

we obtain that the posterior distribution of $[\sigma_1^2 | z_1, \theta_1]$ is:

$$[\sigma_1^2 | z_1, \theta_1] \sim \mathcal{IG}(\alpha_{\sigma_1^2 | n_1}, \frac{Q_1}{2}) \quad (15)$$

where $\mathcal{IG}(\alpha, Q)$ stands for the inverse gamma distribution with density function

$$p_{\alpha, Q}(x) = \frac{Q^\alpha}{\Gamma(\alpha)} \frac{e^{-\frac{Q}{x}}}{x^{\alpha+1}} \quad x > 0$$

and the parameters are given by:

$$\alpha_{\sigma_1^2|n_1} = \frac{n_1 - p_1}{2} \quad (16)$$

and:

$$\begin{aligned} Q_1 &= z_1^T [R_1(D_1)^{-1} - R_1(D_1)^{-1} F_1 (F_1^T R_1(D_1)^{-1} F_1)^{-1} F_1^T R_1(D_1)^{-1}] z_1 \\ &= (z_1 - F_1 \hat{\beta}_1)^T R_1(D_1)^{-1} (z_1 - F_1 \hat{\beta}_1) \end{aligned} \quad (17)$$

$$\hat{\beta}_1 = E[\beta_1 | z_1, \sigma_1^2, \theta_1] = [F_1^T R_1(D_1)^{-1} F_1]^{-1} [F_1^T R_1(D_1)^{-1} z_1] \quad (18)$$

Bayesian estimation of parameters with non-informative ‘‘Jeffreys priors’’ [Jeffreys (1961)] gives the same results as maximum likelihood estimation for the parameter β_1 . For the parameter σ_1^2 , the estimation given by $\frac{Q_1}{2\alpha_{\sigma_1^2|n_1}}$ is identical to the one obtained with the restricted maximum likelihood method. This method was introduced by [Patterson & Thompson (1971)] in order to reduce the bias of the maximum likelihood estimator.

Secondly, let us consider the set of parameters $(\beta_2, \rho, \sigma_2^2, \theta_2)$. In order to have closed form formulas for the estimation of (β_2, ρ) , we estimate them together. The idea to carry out a joint estimation is proposed for the first time in this paper and we believe it is important. Indeed, if the cheaper code is perfectly known, it can be considered as a regression function and so ρ will be a regression parameter. In this case, an independent estimation of β_2 and ρ will not be consistent.

Using similar Jeffrey prior distributions as in (13) and the same methodology as for the estimation of (β_1, σ_1^2) , we find that:

$$[(\rho, \beta_2) | z_1, z_2, \sigma_2^2, \theta_2] \sim \mathcal{N}_{p_2+1} \left([F^T R_2(D_2)^{-1} F]^{-1} [F^T R_2(D_2)^{-1} F], [F^T \frac{R_2(D_2)^{-1}}{\sigma_2^2} F]^{-1} \right) \quad (19)$$

and:

$$[\sigma_2^2 | z_2, z_1, \theta_2] \sim \mathcal{IG}(\alpha_{\sigma_2^2|n_2}, \frac{Q_2}{2}) \quad (20)$$

where:

$$\alpha_{\sigma_2^2|n_2} = \frac{n_2 - p_2 - 1}{2} \quad (21)$$

and:

$$\begin{aligned} Q_2 &= z_2^T [R_2(D_2)^{-1} - R_2(D_2)^{-1} F (F^T R_2(D_2)^{-1} F)^{-1} F^T R_2(D_2)^{-1}] z_2 \\ &= (z_2 - F \hat{\lambda})^T R_2(D_2)^{-1} (z_2 - F \hat{\lambda}) \end{aligned} \quad (22)$$

$$\hat{\lambda} = E[(\rho, \beta_2) | z_1, z_2, \sigma_2^2, \theta_2] = [F^T R_2(D_2)^{-1} F]^{-1} [F^T R_2(D_2)^{-1} z_2] \quad (23)$$

The design matrix F is such that $F = [\rho z_1(D_2) \quad F_2]$. Furthermore, the estimation of σ_2^2 given by $\frac{Q_2}{2\alpha_{\sigma_2^2|n_2}}$ is the same as the restricted maximum likelihood one.

The hyper-parameters θ_1 and θ_2 are found by minimizing the opposite of the concentrated restricted log-likelihoods:

$$\log(|\det(R_1(D_1))|) + (n_1 - p_1) \log(\hat{\sigma}_1^2) \quad (24)$$

and:

$$\log(|\det(R_2(D_2))|) + (n_2 - p_2 - 1)\log(\hat{\sigma}_2^2) \quad (25)$$

These equations must be numerically minimized with a global optimization method. We use an evolutionary method coupled with a BFGS algorithm. The drawback of the maximum likelihood estimation (see [Lehmann & Casella (1998)]) is that, contrarily to Bayes estimation, we do not have any information about the variance of the estimator. Nevertheless, Bayes estimation of the hyper parameters θ_1 and θ_2 are prohibitive and as noted in [Santner, Williams & Notz (2003)] the choice of the prior distribution is non trivial. Therefore, we will always consider these parameters as known and we will estimate them with a concentrated restricted likelihood method.

5 Bayesian prediction for a code with 2 levels

The aim of a Bayesian prediction is to provide a predictive distribution integrating the posterior distributions of the parameters and hence taking into account their uncertainty.

A Bayesian prediction for a code with 2 levels was suggested by [Qian & Wu (2008)]. Nevertheless, we propose here a new Bayesian approach with some significant differences. First, we assume that the adjustment coefficient is a regression function whereas [Qian & Wu (2008)] model it with a Gaussian process. Secondly, we use different prior distributions for the parameter estimation. More specifically, according to the Bayesian estimation of parameters previously presented, we use a joint prior distribution for (β_2, ρ) conditioned by σ_2^2 whereas [Qian & Wu (2008)] use separated prior distributions with ρ not conditioned by σ_2^2 . Then, we use a hierarchy between the different parameters. At the lowest level is the regressor parameter β . At the second level is the variance parameter σ^2 which controls the distribution of the parameter β . At the top level is the parameter θ which controls the distribution of the parameters at the bottom levels. It is common to use a hierarchical specification of models for Bayesian prediction as presented in [Rasmussen & Williams (2006)]. This strategy will allow us to obtain explicit formulas for the joint distribution of the parameters and above all, to reduce the cost of the numerical implementation of the complete Bayesian prediction.

We will also present the case in which we do not have any prior information about the parameters. In order to avoid computationally expensive implementation, we will consider the hyper parameter θ to be known. In practice, it is estimated by minimizing the opposite of the concentrated restricted log-likelihood.

5.1 Prior distributions and Bayesian estimation of the parameters

Many choices of priors can be made for the Bayesian modelling. Here we study the two following cases:

- (I) Priors for each parameter are informative.
- (II) Priors for each parameter are non-informative.

For the non-informative case (II), we use the improper distributions corresponding to the ‘‘Jeffreys prior’’ and then the posterior distributions are given in Section 4.2. Note that non-informative distributions are used when we do not have prior knowledge. For the informative

case (I), we will consider the following prior distributions:

$$[\beta_1 | \sigma_1^2] \sim \mathcal{N}_{p_1}(b_1, \sigma_1^2 V_1), \quad [(\rho, \beta_2) | z_1, \sigma_2^2] \sim \mathcal{N}_{1+p_2} \left(b_\lambda = \begin{pmatrix} b_\rho \\ b_2 \end{pmatrix}, \sigma_2^2 V_\lambda = \sigma_2^2 \begin{pmatrix} V_\rho & 0 \\ 0 & V_2 \end{pmatrix} \right)$$

$$[\sigma_1^2] \sim \mathcal{IG}(\alpha_1, \gamma_1), \quad [\sigma_2^2 | z_1] \sim \mathcal{IG}(\alpha_2, \gamma_2)$$

where $b_1 \in \mathbb{R}^{p_1}$, $b_\lambda \in \mathbb{R}^{1+p_2}$, V_1 is a $(p_1 \times p_1)$ diagonal matrix, V_λ is a $((1+p_2) \times (1+p_2))$ diagonal matrix and $\alpha_1, \gamma_1, \alpha_2, \gamma_2 > 0$. The forms of the priors are chosen in order to be able to get closed form expressions for the posterior distributions. Note that there are enough free parameters in the priors to allow the user to prescribe their means and variances. From the previous prior definitions, the posterior distributions of the parameters are:

$$[\beta_1 | z_1, \sigma_1^2] \sim \mathcal{N}_{p_1}(A_i^1 \nu_i^1, A_i^1) \quad [(\rho, \beta_2) | z_1, z_2, \sigma_2^2] \sim \mathcal{N}_{p_2+1}(A_i^\lambda \nu_i^\lambda, A_i^\lambda) \quad (26)$$

where:

$$A_i^1 = \begin{cases} [F_1^T \frac{R_1^{-1}(D_1)}{\sigma_1^2} F_1 + \frac{V_1^{-1}}{\sigma_1^2}]^{-1} & i = (\text{I}) \\ [F_1^T \frac{R_1^{-1}(D_1)}{\sigma_1^2} F_1]^{-1} & i = (\text{II}) \end{cases} \quad \nu_i^1 = \begin{cases} [F_1^T \frac{R_1^{-1}(D_1)}{\sigma_1^2} z_1 + \frac{V_1^{-1}}{\sigma_1^2} b_1] & i = (\text{I}) \\ [F_1^T \frac{R_1^{-1}(D_1)}{\sigma_1^2} z_1] & i = (\text{II}) \end{cases} \quad (27)$$

$$A_i^\lambda = \begin{cases} [F^T \frac{R_2^{-1}(D_2)}{\sigma_2^2} F + \frac{V_\lambda^{-1}}{\sigma_2^2}]^{-1} & i = (\text{I}) \\ [F^T \frac{R_2^{-1}(D_2)}{\sigma_2^2} F]^{-1} & i = (\text{II}) \end{cases} \quad \nu_i^\lambda = \begin{cases} [F^T \frac{R_2^{-1}(D_2)}{\sigma_2^2} z_2 + \frac{V_\lambda^{-1}}{\sigma_2^2} b_\lambda] & i = (\text{I}) \\ [F^T \frac{R_2^{-1}(D_2)}{\sigma_2^2} z_2] & i = (\text{II}) \end{cases} \quad (28)$$

and $F = [\rho z_1(D_2) \quad F_2]$. Furthermore, we have:

$$[\sigma_1^2 | z_1] \sim \mathcal{IG}(\alpha_i^{\sigma_1^2 | n_1}, \frac{Q_i^1}{2}), \quad [\sigma_2^2 | z_2, z_1] \sim \mathcal{IG}(\alpha_i^{\sigma_2^2 | n_2}, \frac{Q_i^2}{2}) \quad (29)$$

where:

$$Q_i^1 = \begin{cases} \gamma_1 + (b_1 - \hat{\beta}_1)^T (V_1 + [F_1^T R_1^{-1}(D_1) F_1]^{-1})^{-1} (b_1 - \hat{\beta}_1) + Q_2^1 & i = (\text{I}) \\ z_1^T [R_1^{-1}(D_1) - R_1^{-1}(D_1) F_1 (F_1^T R_1^{-1}(D_1) F_1)^{-1} F_1^T R_1^{-1}(D_1)] z_1 & i = (\text{II}) \end{cases}$$

$$Q_i^2 = \begin{cases} \gamma_2 + (b_\lambda - \hat{\lambda})^T (V_\lambda + [F^T R_2^{-1}(D_2) F]^{-1})^{-1} (b_\lambda - \hat{\lambda}) + Q_2^2 & i = (\text{I}) \\ z_2^T [R_2^{-1}(D_2) - R_2^{-1}(D_2) F (F^T R_2^{-1}(D_2) F)^{-1} F^T R_2^{-1}(D_2)] z_2 & i = (\text{II}) \end{cases}$$

$$\hat{\beta}_1 = (F_1^T R_1^{-1}(D_1) F_1)^{-1} F_1^T R_1^{-1}(D_1) z_1 \quad \hat{\lambda} = (F^T R_2^{-1}(D_2) F)^{-1} F^T R_2^{-1}(D_2) z_2$$

$$\alpha_i^{\sigma_1^2 | n_1} = \begin{cases} \frac{n_1}{2} + \alpha_1 & i = (\text{I}) \\ \frac{n_1 - p_1}{2} & i = (\text{II}) \end{cases} \quad \alpha_i^{\sigma_2^2 | n_2} = \begin{cases} \frac{n_2}{2} + \alpha_2 & i = (\text{I}) \\ \frac{n_2 - p_2 - 1}{2} & i = (\text{II}) \end{cases}$$

Mixing of informative and non-informative priors are of course possible and easy to implement. As we will discuss in Subsection 5.4 and see in the examples of Section 7, the use of informative priors has minor impact on the mean estimation but may have a strong impact on variance estimation.

5.2 Predictive distributions when $\beta_2, \rho, \sigma_1^2$ and σ_2^2 are known

As a preliminary step towards the Bayesian prediction carried out in the next subsection, we give here Bayesian prediction in the form of closed form expressions when the parameters $\beta_2, \rho, \sigma_1^2$ and σ_2^2 are known. Then the conditional distribution of $[Z_2(x)|Z = z, \beta_2, \rho, \sigma_1^2, \sigma_2^2]$ is given by:

$$[Z_2(x)|\mathcal{Z} = z, \beta_2, \rho, \sigma_1^2, \sigma_2^2] \sim \mathcal{N}(\mu_i(x), \sigma_i^2(x)) \quad (30)$$

where:

$$\begin{aligned} \mu_i(x) &= h'(x)^T \begin{pmatrix} A_i^1 \nu_i^1 \\ \beta_2 \end{pmatrix} + t(x)^T V^{-1} \left(z - H \begin{pmatrix} A_i^1 \nu_i^1 \\ \beta_2 \end{pmatrix} \right) \\ \sigma_i^2(x) &= s_{Z_2}^2(x) + k_1 A_i^1 k_1^T \end{aligned}$$

and A_i^1 and ν_i^1 are defined by (27).

Note that the estimated variance is augmented by the term $k_1 A_i^1 k_1^T$ which quantifies the uncertainty due to the estimation of β_1 . k_1 is a $(1 \times p_1)$ vector composed of the p_1 first elements of the $(1 \times p_1, 1 \times p_2)$ vector k given by:

$$k = (k_1, k_2) = h'(x)^T - t(x)^T V^{-1} H$$

H is given by (4.1). The existence of closed form formulas is important as it will allow for a fast numerical implementation.

5.3 Bayesian prediction

Before performing the Bayesian prediction we note that - thanks to the explicit joint prior distribution for β_2 and ρ , the independance hypotheses and the hierarchical specification of the paramaters - conditioning on θ , we have an explicit formula for the following joint density:

$$p(\beta_1, \beta_2, \rho, \sigma_1^2, \sigma_2^2 | z_1, z_2) = p(\beta_1 | \sigma_1^2 | z_1) p(\beta_2, \rho | \sigma_2^2 | z_1, z_2) p(\sigma_1^2 | z_1) p(\sigma_2^2 | z_1, z_2) \quad (31)$$

This explicit joint density is an original result which contrasts with [Qian & Wu (2008)] and which allows us to avoid prohibitive implementation for the Bayesian analysis.

First, we consider the predictive distribution with σ_1^2 and σ_2^2 known. Considering the conditional independence assumption between $(\delta(x))_{x \in Q}$ and $(Z_1(x))_{x \in Q}$, the probability density function of $[Z_2(x)|\mathcal{Z} = z, \sigma_1^2, \sigma_2^2]$ can be deduced from the following integral:

$$p(z_2(x) | z_1, z_2, \sigma_1^2, \sigma_2^2) = \int_{\mathbb{R}^{1+p_2}} p(z_2(x) | z_1, z_2, \beta_2, \rho, \sigma_1^2, \sigma_2^2) p(\rho, \beta_2 | z_1, z_2, \sigma_2^2) d\rho d\beta_2 \quad (32)$$

where $p(z_2(x) | z_1, z_2, \beta_2, \rho, \sigma_1^2, \sigma_2^2)$ is given by (8). This integral has to be numerically evaluated. Since $[\rho, \beta_2 | z_1, z_2, \sigma_2^2]$ has a known normal distribution given by (26), we here use a crude Monte-Carlo algorithm when the dimension of β_2 and ρ is high, or a trapezoidal quadrature method when it is low.

Then, we infer from the parameters σ_1^2 and σ_2^2 . Due to the independence between $(\delta(x))_{x \in Q}$ and $(Z_1(x))_{x \in Q}$, the probability density function of $[Z_2(x)|\mathcal{Z} = z]$ is:

$$p(z_2(x) | z_1, z_2) = \int_{\mathbb{R}^2} p(z_2(x) | z_1, z_2, \sigma_1^2, \sigma_2^2) p(\sigma_1^2 | z_1) p(\sigma_2^2 | z_1, z_2) d\sigma_1^2 d\sigma_2^2 \quad (33)$$

where $p(\sigma_1^2|z_1)$ and $p(\sigma_2^2|z_1, z_2)$ are given by (29). This integral has also to be numerically evaluated. Since we have a double integration, a quadrature method will be efficient. We use here a trapezoidal numerical integration, defining the region of integration $[\sigma_{1_{inf}}^2, \sigma_{1_{sup}}^2] \times [\sigma_{2_{inf}}^2, \sigma_{2_{sup}}^2]$ from the equation (29) and such that $p(\sigma_{1_{inf}}^2|z_1)$, $p(\sigma_{1_{sup}}^2|z_1)$, $p(\sigma_{2_{inf}}^2|z_1, z_2)$ and $p(\sigma_{2_{sup}}^2|z_1, z_2)$ are close to 0. This region essentially contains the support of the function. Furthermore, we create a non-uniform integration grid distributed with a geometric progression.

Finally $p(z_2(x)|z_1, z_2)$ is a predictive density function integrating the posterior distribution of parameters $(\beta_2, \rho, \beta_1, \sigma_1^2, \sigma_2^2)$. We hence have a predictive distribution taking into account the uncertainties due to the parameter estimations. This predictive distribution is clearly not Gaussian but we have observed in practice that it is extremely close to normality. Therefore, it is relevant to consider in our analysis only the mean $\mathbb{E}[Z_2(x)|\mathcal{Z}_1 = z_1, \mathcal{Z}_2 = z_2]$ and the variance $\text{Var}(Z_2(x)|\mathcal{Z}_1 = z_1, \mathcal{Z}_2 = z_2)$. With classical formulas of the total mean, variance and covariance, parameter estimations in (26) and results of Subsection 9.2, it can be shown that:

$$\mathbb{E}[Z_2(x)|\mathcal{Z} = z] = \left(\mathbb{E}[Z_1(x)|\mathcal{Z}_1 = z_1] \quad f_1^T(x) \right) A_i^\lambda \nu_i^\lambda + R_2(\{x\}, D_2) R_2^{-1} \left(z_2 - F A_i^\lambda \nu_i^\lambda \right) \quad (34)$$

$$\mathbb{E}[Z_1(x)|\mathcal{Z}_1 = z_1] = f_1^T(x) A_i^1 \nu_i^1 + R_1(\{x\}, D_1) R_1^{-1} (z_1 - F_1 A_i^1 \nu_i^1) \quad (35)$$

$$\begin{aligned} \text{Var}(Z_2(x)|\mathcal{Z} = z) &= \hat{\rho}^2 \text{Var}(Z_1(x)|\mathcal{Z}_1 = z_1) + (h_2^T A_i^\lambda h_2) \\ &\quad + \frac{Q_i^2}{2(\alpha_i^{\sigma_2^2|n_2} - 1)} (1 - R_2(\{x\}, D_2) R_2^{-1} R_2(D_2, \{x\})) \end{aligned} \quad (36)$$

where $h_2^T = \left(\mathbb{E}[Z_1(x)|\mathcal{Z}_1 = z_1] \quad f_1^T(x) \right) - R_2(\{x\}, D_2) R_2^{-1} F$, $\hat{\rho} = A_i^\lambda \nu_i^\lambda(1)$ and:

$$\text{Var}(Z_1(x)|\mathcal{Z}_1 = z_1) = \frac{Q_i^1}{2(\alpha_i^{\sigma_1^2|n_1} - 1)} (1 - R_1(\{x\}, D_1) R_1^{-1} R_1(D_1, \{x\})) + (h_1^T A_i^1 h_1) \quad (37)$$

where $h_1^T = (f_1^T(x) - R_1(\{x\}, D_1) R_1^{-1} F_1)$.

We note that, in the mean of the predictive distribution, the parameters have been just replaced by their posterior means. Furthermore, in the variance of the predictive distribution, the variance parameters have been replaced by their posterior means and two terms have been added: $h_1^T A_i^1 h_1$ and $h_2^T A_i^\lambda h_2$. They represent the uncertainty due to the estimation of the regression parameters (including the adjustment coefficient). We call these formulas the universal co-kriging equations due to their similarities with the well known universal kriging equations. These formulas can naturally be extended for the case of ρ depending on x and with more than 2-levels of code.

5.4 Discussion about the numerical evaluations of the integrals

We saw in the previous section that we can obtain an analytical prediction when β_2 , ρ , σ_1^2 and σ_2^2 are known. From this, we can have a Bayesian prediction with only two nested simple integrations. One of them can be approximated with a quadrature or a crude Monte Carlo method, which is not too expensive. The other is a double integration approximated with a quadrature method which is efficient and not expensive. Therefore, we do not use any Markov

chain Monte Carlo method and we considerably reduce the time and the complexity of the method. This allows us to easily build an accurate Bayesian metamodel. Note that this metamodel is build with two nested integrations. Indeed, at each integration points used to evaluate the integral (33), we evaluate the integral (32) with a crude Monte-Carlo procedure. Practically, we use 441 integration points to approximate (33) and 1000 Monte-Carlo particles to approximate (32). Therefore, we have 441000 call to the predictive density function (30).

To avoid a prohibitive implementation, another approach has also been proposed by [Cumming & Goldstein (2009)]. They adopt a Bayes linear formulation which requires only the specification of the means, variances and covariance. See [Goldstein & Wooff (2007)] for further details about the Bayes linear approach. The strength of this method is that its computationally cost is low. Nonetheless, since it only focuses on posterior means and covariances, it does not provide the full posterior predictive distribution. Moreover, [Cumming & Goldstein (2009)] provide a multi-level analysis considering (β_1, σ_1^2) as known and their approach does not provide posterior distributions for the parameters. The universal co-kriging equations presented in Subsection 5.3 can be viewed as an extension of the ones of [Cumming & Goldstein (2009)]. Indeed, we provide a full linear Bayesian formulation by inferring from all the known posterior distributions of the parameters.

Finally, we highlight the fact that our Bayesian procedure can be used to perform multi-fidelity analysis with more than 2 levels of code whereas the cost of the one presented by [Qian & Wu (2008)] is too high to allow such analysis. We illustrate in Section 11 through an industrial case the importance of using more than 2 levels of code.

6 Experimental design

As presented in Section 3 we consider nested experimental designs $\forall t = 2, \dots, s \quad D_t \subseteq D_{t-1}$. Therefore, we have to adopt particular design strategies to uniformly spread the inputs for all D_t . Space-filling designs are widely used in computer experiments, such as Latin hypercube (see [McKay, Beckman & Conover (1979)], [Morris & Mitchell (1995)]), Orthogonal array-based Latin hypercube (see [Owen (1992)], [Tang (1993)]) and uniform designs (see [Fang, Lin, Winker & Zhang (2000)]). A strategy based on Orthogonal array-based Latin hypercube for nested space-filling designs is proposed by [Qian, Ai & Wu (2009)] and [Qian, Tang & Wu (2009)].

We consider here another strategy for space-filling design, described in the following algorithm, which is very simple and not time-consuming. The number of points n_t for each design D_t is prescribed by the user, as well as the experimental design method applied to determine the coarsest grid D_s used for the most expensive code z_s .

ALGORITHM

build $D_s = \{x_j^{(s)}\}_{j=1, \dots, n_s}$ with the experimental design method prescribed by the user.

for $t = s$ to 2 **do**:

 build design \tilde{D}_{t-1} with the experimental design method prescribed by the user.

for $i = 1$ to n_t **do**:

 find $\tilde{x}_j^{(t-1)} \in \tilde{D}_{t-1}$ the closest point from $x_i^{(t)} \in D_t$ where $j \in [1, n_{t-1}]$.

remove $\tilde{x}_j^{(t-1)}$ from \tilde{D}_{t-1} .

end for

$D_{t-1} = \tilde{D}_{t-1} \cup D_t$.

end for

This strategy allows us to use any space-filling design method. Therefore it is more flexible than the one presented by [Qian, Ai & Wu (2009)]. Furthermore, it conserves the initial structure of the experimental design D_s of the most accurate code, contrarily to a strategy based on selection of subsets of an experimental design for the less accurate code as presented by [Kennedy & O’Hagan (2000)], [Floater & Iske (1996)] and [Forrester, Sobester & Keane (2007)]. We hence can ensure that D_s has excellent space-filling properties. Moreover, the experimental design D_{t-1} being equal to $\tilde{D}_{t-1} \cup D_t$, this method ensures the nested property.

7 Toy examples

We will present in this section some co-kriging metamodels using one-dimensional functions inspired by the example presented in [Forrester, Sobester & Keane (2007)]. For the following examples, we will use a non-Bayesian co-kriging model - *i.e.* the one presented in [Kennedy & O’Hagan (2000)] - but with a Bayesian estimation of the parameters (see Section 4.2) and for the second example we will also use a Bayesian co-kriging. Furthermore, the correlation kernels are assumed to be:

$$r_t(x_i^{(k)} - x_j^{(l)}; \theta_t) = \exp\left(-\frac{\|x_i^{(k)} - x_j^{(l)}\|^2}{\theta_t^2}\right)$$

where:

$$t, k, l = 1, 2 \quad 1 \leq i \leq n_1 \quad 1 \leq j \leq n_2 \quad x_i^{(k)} \in D_k \quad x_j^{(l)} \in D_l$$

Example 1. We assume that the expensive code is given by $z_2(x) = (6x - 2)^2 \sin(12x - 4)$ and the cheaper code by $z_1(x) = 0.5z_2(x) + 10(x - 0.5) - 5$. The experimental design set of the cheapest code is $D_1 = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and the one of the expensive code is $D_2 = \{0, 0.4, 0.6, 1\}$. This example is identical to the one-dimensional demonstration presented in [Forrester, Sobester & Keane (2007)]. Figure 1 shows the functions $x \mapsto z_2(x)$ and $x \mapsto z_1(x)$, the training data for z_2 and z_1 , the ordinary kriging using only the expensive data and the co-kriging using expensive and cheap data. To validate the model, the Root-mean-square errors (RMSE) and Q_2 coefficient (38) are computed:

$$Q_2 = 1 - \frac{\sum_{x \in T} (m_{Z_2}(x) - z_2(x))^2}{\sum_{x \in T} (m_{Z_2}(x) - \bar{z}_2)^2} \quad (38)$$

The test set T is composed of a regular grid points sampled from 0 to 1 with a grid step equal to 0.01 and \bar{z}_2 is the empirical mean evaluated in T . The estimated RMSE is 5.68×10^{-2} and the coefficient Q_2 is 99.98%, so we have a prediction error closed to 0. The Bayesian estimation of the parameters of co-kriging are given in Table 1. Furthermore, the estimations of the hyper-parameters (θ_1, θ_2) , calculated by maximizing the concentrated log-likelihoods (24) and (25), are $\hat{\theta}_1 = 0.25$ and $\hat{\theta}_2 = 0.80$. D_1 being a regular grid with a grid step equal to

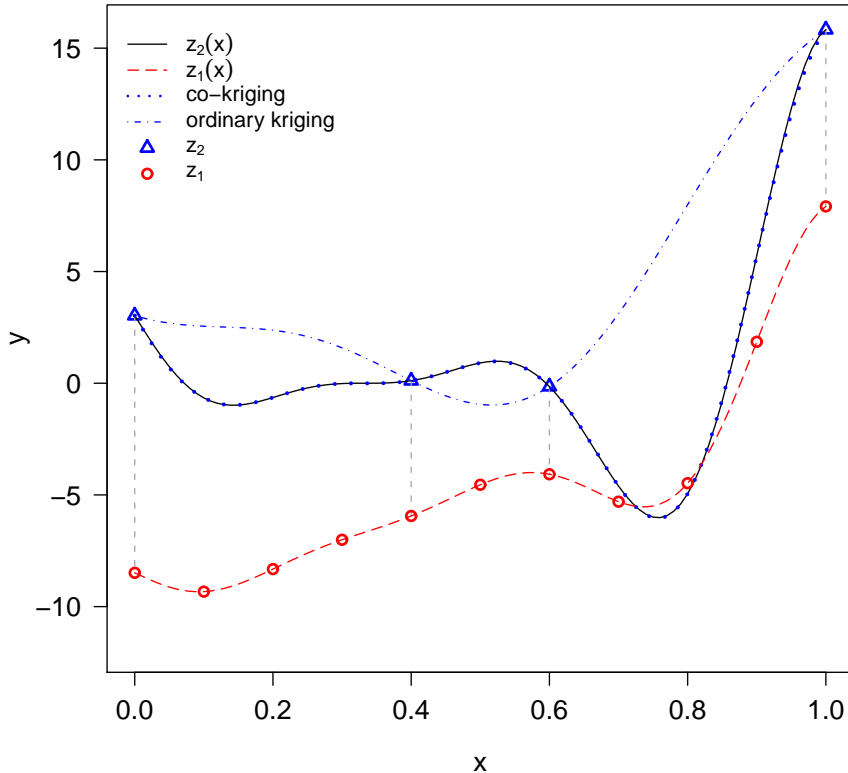


Figure 1: A co-kriging example with one-dimensional functions. The co-kriging metamodel is very close to the expensive output $z_2(\cdot)$ and improves significantly the ordinary kriging metamodel using the small design D_2 .

0.1 and D_2 being composed of points sampled from 0 to 1, points of the experimental designs are hence strongly correlated which will imply a smooth surrogate model.

Coefficient	Estimation
ρ	2
β_2	(20, -20)
β_1	-3.49
σ_1^2	32.75
σ_2^2	7.02×10^{-30}

Table 1: A co-kriging example with one-variable functions. Bayesian estimation of parameters.

We see that the Bayesian estimation of parameters is very effective since the estimations of parameters ρ and β_2 are perfect. Nevertheless this example does not highlight the strength of the method since there is a relation between $z_2(x)_{x \in [0,1]}$ and $z_1(x)_{x \in [0,1]}$ which exactly cor-

responds to the equation (2) with the error δ_2 that can be written in terms of the regression functions f_2 exactly. Therefore, if the cheap code is well modelled, like in our case, the co-kriging is equivalent to a linear regression. Moreover, the very small value of σ_2^2 illustrates this.

Example 2. We assume that the expensive code is given by $z_2(x) = (6x - 2)^2 \sin(12x - 4) + \sin(10 \cos(5x))$ and the cheaper code is given by $z_1(x) = 0.5((6x - 2)^2 \sin(12x - 4)) + 10(x - 0.5) - 5$. Through the term $\sin(10 \cos(5x))$, the expensive code has high frequencies which are not captured by the cheap code and the error δ_2 is not a simple linear combination of the regression functions f_2 . Figure 2 shows the results of kriging and co-kriging for these two functions. The estimated RMSE is 1.05 and the coefficient Q_2 is 93.57%, we still have

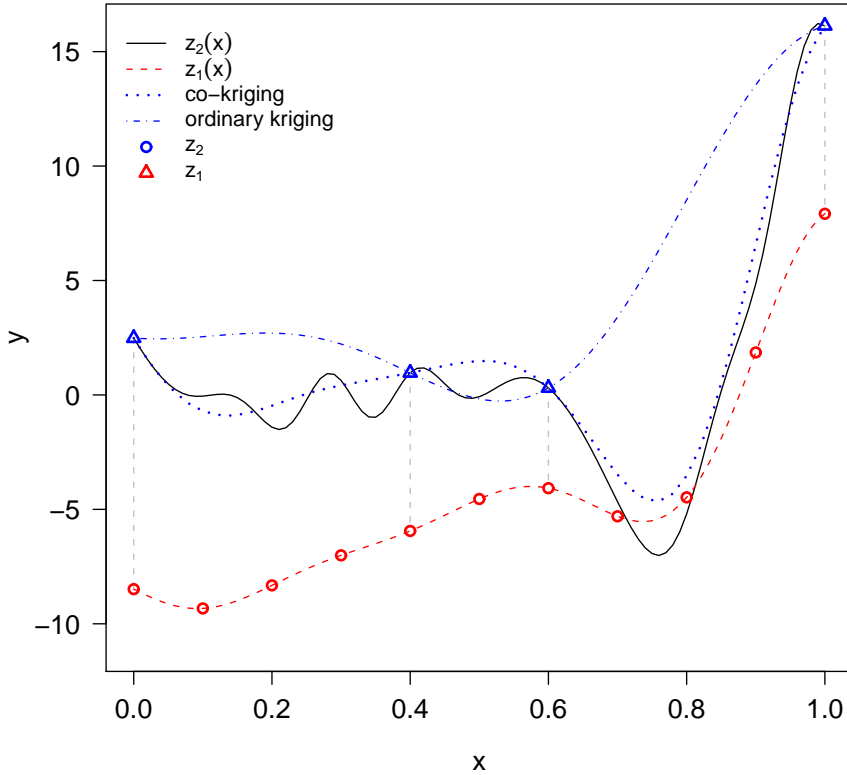


Figure 2: A co-kriging example with one-dimensional functions. The high frequency components of the expensive code are not predicted since they are not captured by the cheap code and the coarse grid used for the expensive code cannot detect them either. Nevertheless, the co-kriging improves the ordinary kriging metamodel since the cheap code allows us to predict the low frequencies of the expensive code accurately.

a good prediction. The Bayesian estimations of the parameters are given in Table 2 and we have $\hat{\theta}_1 = 0.25$ and $\hat{\theta}_2 = 0.07$. The values of these parameters have been fixed according to the following arguments. As the cheap code is the same as the one of the Example 1, we keep the

same estimation for θ_1 . Then, we consider that there are not enough points to carry out a significant estimation of θ_2 . Therefore, we fix the value of $\hat{\theta}_2$ according to the high frequencies introduced by the term $\sin(10 \cos(5x))$.

Coefficient	Estimation
ρ	1.86
β_2	(18.39, -17.00)
β_1	-3.49
σ_1^2	32.75.03
σ_2^2	0.30

Table 2: A co-kriging example with one-dimensional functions. Bayesian estimation of parameters.

Due to the additional term $\sin(10 \cos(5x))$, the estimation of the parameter ρ is less effective than in the first example. This highlights the dependence between the estimation of ρ and the mean of $\delta(x)_{x \in [0,1]}$.

Furthermore, Figure 3 represents the confidence interval at plus or minus twice the standard deviation of the predictive distribution in the Bayesian and non-Bayesian case. We see that we underestimate the variance of the predictive distribution in the non-Bayesian case. This estimation is adjusted in the Bayesian case; nevertheless, it seems to be slightly overestimated. According to the universal co-kriging equations presented in Subsection 5.3, the means of the predictive distributions for the two cases are equivalent. We finally consider the case in which we have prior information:

$$[(\rho, \beta_2) | z_1, \sigma_2^2] \sim \mathcal{N} \left(\begin{pmatrix} 2 \\ 20 \\ -20 \end{pmatrix}, \sigma_2^2 \begin{pmatrix} 0.05 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.05 \end{pmatrix} \right), \quad [\sigma_2^2 | z_1] \sim \mathcal{IG}(3, 1)$$

Figure 4 shows the result of the Bayesian co-kriging with the given prior information. The estimated RMSE is 0.79 and the coefficient Q_2 is 96.57%, we hence improve the accuracy of the metamodel. The predictive mean is closer to the true function and the predictive variance is reduced compared to the non-informative Bayesian case, with the confidence interval that still contains the true function. The posterior estimations of the parameters are given in Table 3 and we have $\hat{\theta}_1 = 0.25$ and $\hat{\theta}_2 = 0.07$.

Coefficient	Estimation
ρ	2.00
β_2	(20.12, -19.81)
β_1	-3.49
σ_1^2	32.75
σ_2^2	0.29

Table 3: A co-kriging example with one-dimensional functions and prior information. Posterior estimation of parameters.

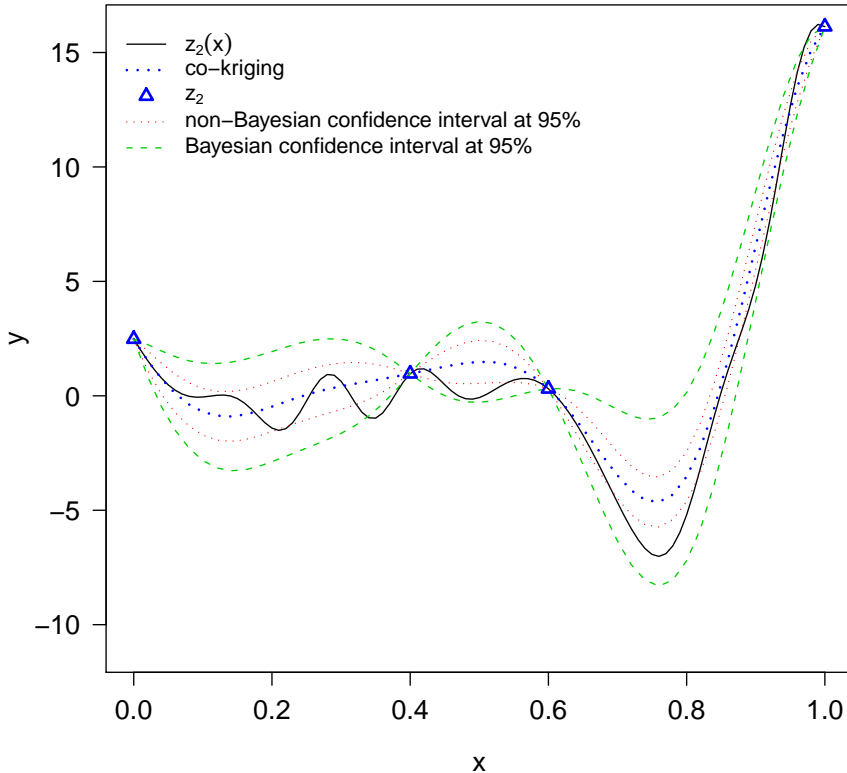


Figure 3: A co-kriging example with one-dimensional functions and without any prior information. Comparison between Bayesian and non-Bayesian co-kriging. The thick dotted line represents the prediction mean, the thin dotted lines represent the confidence interval at plus or minus twice the standard deviation in the non-Bayesian case and the dashed lines represent the same confidence interval in the Bayesian case.

8 Example 1: hydrodynamic simulator

This example illustrates the comparison between Bayesian and non-Bayesian co-kriging. The co-kriging method is applied to a hydrodynamic code named “MELTEM”. This code simulates a second-order turbulence model for gaseous mixtures induced by Richtmyer-Meshkov instability [Gregoire, Souffland & Gauthier (2005)]. We consider here two parameters X_1 and X_2 which are phenomenological coefficients used in the equations of the energy of dissipation of the turbulent flow. These two coefficients vary into the region $[0.5, 1.5] \times [1.5, 2.3]$. The considered code output, called R , is the ratio between the longitudinal and the transversal speed variations in the turbulence area. The simulator is a finite-elements code which can be run at different levels of accuracy by altering the finite-elements mesh. The simple code $z_1(\cdot)$, using a coarse mesh, takes 20 seconds to produce an output whereas the complex code $z_2(\cdot)$, using a fine mesh, takes 8 minutes. A Latin hypercube design of 200 points was built in

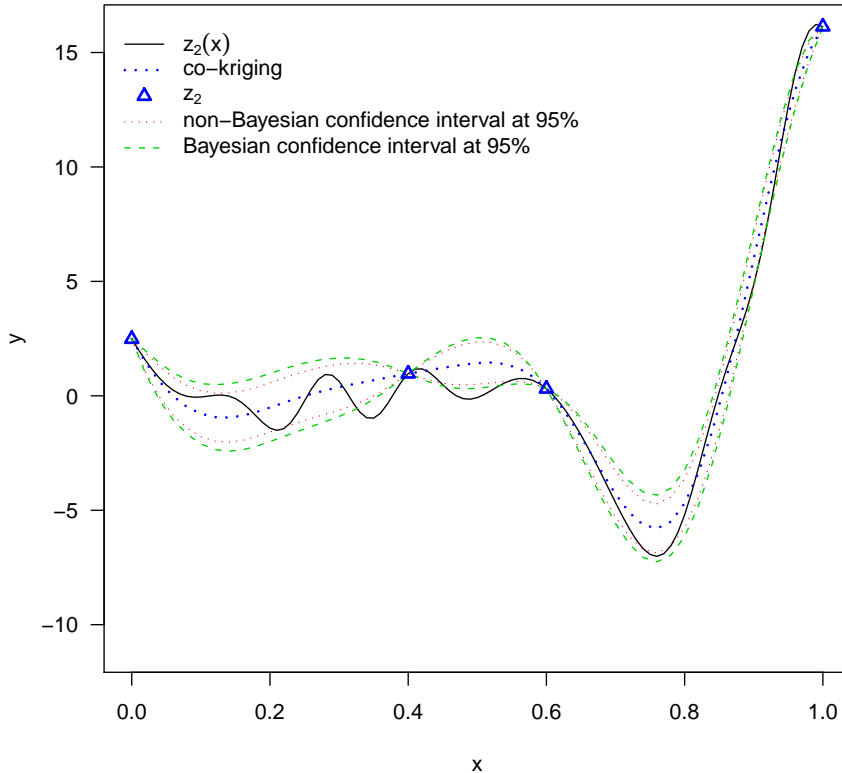


Figure 4: A Bayesian co-kriging example with one-dimensional functions and prior information. The prior information improves the accuracy of the co-kriging metamodel and the variance of the predictive distribution has decreased.

the input parameter space and optimized with a maximin criterion. The two codes were run on this experimental design set. The aim of the study is to build a prediction as accurate as possible using only a few runs of the complex code. Therefore, we extract a subset from the outputs of the complex code that we use as data (the complementary subset is used for the validation step). Furthermore, no prior information is available: we are in the non-informative case.

8.1 Comparison between ordinary kriging and non Bayesian co-kriging

Figure 5 shows the prediction RMSE for ordinary kriging and non Bayesian co-kriging when the number of runs for the complex code varies. We use for both ordinary kriging and co-kriging a Matern $\frac{5}{2}$ covariance kernel and we consider that $f_\rho^T(x) = (1, x_1)$ (see Section A for the case of ρ depending on x), $f_2(x) = 1$ and $f_1(x) = 1$. For the co-kriging, we use the 200 runs of the Latin hypercube for the fast code and the RMSE is estimated with the complex-code outputs that had not been used to build the model. Furthermore, for a fixed number of runs for the accurate simulator, the figure gives the average RMSE calculated from 20 different

Latin hypercube designs. In the Figure 5, we see that the error saturates when there are

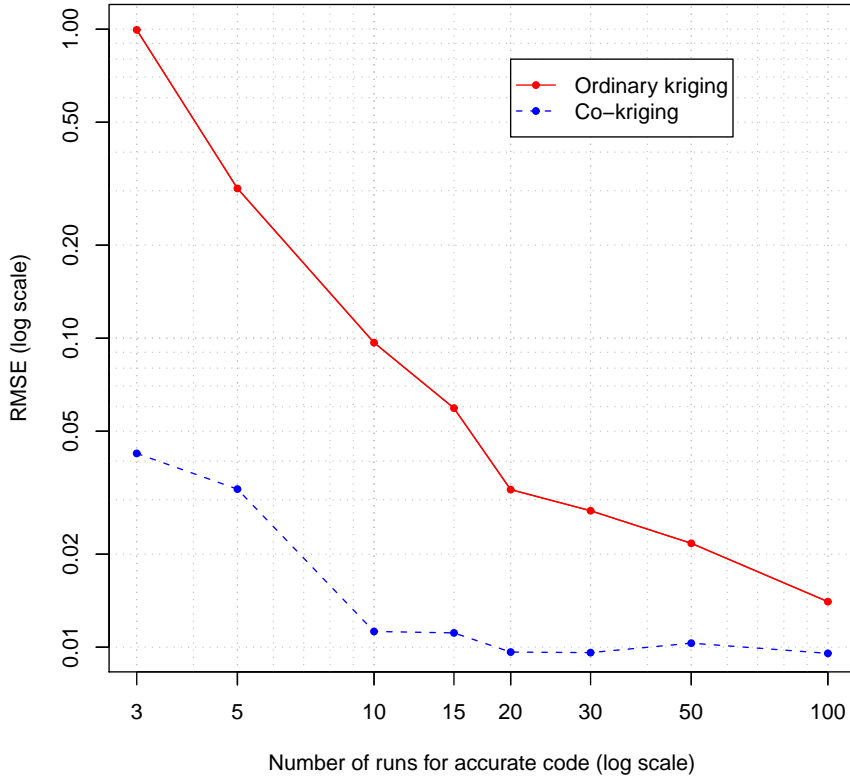


Figure 5: Comparison between ordinary kriging and non-Bayesian co-kriging. Co-kriging predictions are better than the ordinary kriging ones and with only 10 simulations we already have an excellent model with co-kriging.

more than 10 runs for the accurate simulator. Therefore, 10 will be the number of calls of the complex code in the remainder of this study.

8.2 Comparison between non Bayesian co-kriging and Bayesian co-kriging

In this section, we compare a model obtained with a non-Bayesian co-kriging - as presented in [Kennedy & O’Hagan (2000)] and [Forrester, Sobester & Keane (2007)] - and a model obtained with a Bayesian co-kriging as presented in this paper. We use 200 simulations for the cheap code and 10 for the expensive code. The 190 others simulations of the complex code are used to validate and compare our models. To build the different covariance matrices, we consider a matern- $\frac{5}{2}$ kernel (see [Rasmussen & Williams (2006)]), $f_{\rho}^T(x) = (1, x_1)$, $f_2(x) = 1$, $f_1(x) = 1$ and, using the concentrated maximum likelihood, we have the following estimation for the hyper-parameters of correlation: $\hat{\theta}_1 = (0.47, 1.59)$; $\hat{\theta}_2 = (0.18, 1.42)$. According to the values of the hyper-parameter estimates, the co-kriging model is very smooth since the

correlation length is large compared to the size of the input parameter space. Furthermore, the estimated correlation between the two codes is 98.96%, the cheap code hence well approximates the response.

Table 3 presents the Bayesian estimation of the parameters.

Regression coefficient	Posterior mean	Posterior Covariance
β_1	3.12	σ_t^2 0.40
$\begin{pmatrix} \beta_\rho \\ \beta_2 \end{pmatrix}$	$\begin{pmatrix} 1.04 \\ -0.15 \\ -0.02 \end{pmatrix}$	$\begin{pmatrix} 0.21 & -0.16 & -0.21 \\ -0.16 & 0.36 & -0.57 \\ -0.21 & -0.57 & 2.84 \end{pmatrix}$
Variance coefficient	Q_t	α_t
σ_1^2	74.14	99.5
σ_2^2	0.30	3.5

Table 4: Example: hydrodynamic simulator. Bayesian estimation of the parameters (26) and (29).

We see in Table 4 that the correlation between β_ρ and β_2 is non-negligible which highlights the importance of taking into account the correlation between these two coefficients. We also see that the adjustment parameter β_ρ is close to 1 with a linear trend with a smooth slope, both code have hence the same order of magnitude. Finally, the variance of the bias between the two codes is ten times lower than the one of the cheap code due to the fact that it is much easier to model than the cheap code itself.

Table 5 compares the prediction accuracy of the Bayesian and the non-Bayesian co-kriging. The different coefficients (MaxAE: Maximal Absolute Error, RMSE, Q_2 , ...) are estimated with the 190 responses of the complex code that have not been used to build the model.

	Q_2	RMSE	MaxAE
Bayesian co-kriging	99.86%	0.042	0.111
Non-Bayesian co-kriging	99.86%	0.042	0.111
	Average Std. dev.	Median Std. dev.	Maximal Std. dev.
Bayesian co-kriging	0.068	0.065	0.153
Non-Bayesian co-kriging	0.041	0.038	0.091

Table 5: Example: hydrodynamic simulator. Comparison between Bayesian and non-Bayesian co-kriging. The non-Bayesian predictions are identical to the Bayesian ones and the variance of the predictive distribution in the Bayesian case is slightly larger than the one in the non-Bayesian case (Std. dev represents the standard deviation of the predictive distribution).

We see that the accuracies of the two models are identical. Indeed, according to (34), the posterior means of these two models are equivalent. Nevertheless, the average standard deviation of the prediction is slightly larger in the Bayesian case than in the non-Bayesian one. Comparing the RMSE and the average standard deviation estimations in Table 5, it

seems that we slightly overestimate a little the variance of the predictive distribution in the Bayesian case whereas in the non-Bayesian case, this estimation seems correct.

Figure 6 shows the experimental designs for the 2 levels of code. Figure 7 represents the mean and the confidence interval at plus or minus twice the standard deviation of the predictions for points along the lines 1 and 2 plotted in Figure 6. In particular in Figure 7 in

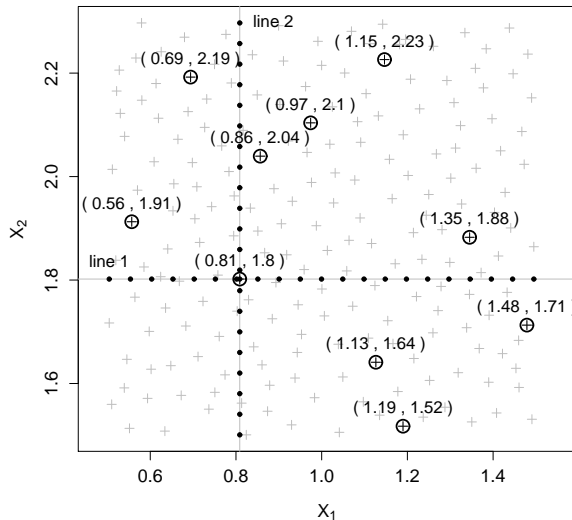


Figure 6: Comparison between Bayesian co-kriging and non Bayesian co-kriging for the hydrodynamic simulator. The crosses represent the experimental design set for the cheap code, the circled crosses represent the experiments design set for the complex code, and the thick points represent the set at which predictions are calculated and reported in Figure 7.

line 1, we see necked points at coordinate $(0.55, 1.15, 1.25)$ since, in the direction of X_2 , the hyper-parameter of correlation for $(\delta(x))_{x \in Q}$ is 1.42 and points of D_2 have almost the same coordinate: $(0.56, 1.13, 1.135)$.

Finally, we see that in this case the difference between the Bayesian and the non-Bayesian confidence interval is less important than the one in the example 2 Section 7. This is due to the fact that the number of data to evaluate the hyper-parameters is here larger than in Section 7 and their estimations are hence less uncertain. Moreover, the high correlation degree between the two codes indicates that it is very easy to learn the bias between them and so the predictive variance of the bias is small.

9 The case of s levels of code

The aim of this Section is to perform a multi-level co-kriging with any number of codes. Let us consider s levels of code. The generalization of the previous model is straightforward. Actually, if we note $\beta = (\beta_1^T, \dots, \beta_s^T)^T$, $\rho = (\rho_1, \dots, \rho_{s-1})$, $\sigma^2 = (\sigma_1^2, \dots, \sigma_s^2)$ and $\theta = (\theta_1, \dots, \theta_s)$, we have:

$$\forall x \in Q \quad [Z_s(x) | \mathcal{Z} = z, \beta, \rho, \sigma^2, \theta] \sim \mathcal{N}(m_{Z_s}(x), s_{Z_s}^2(x))$$

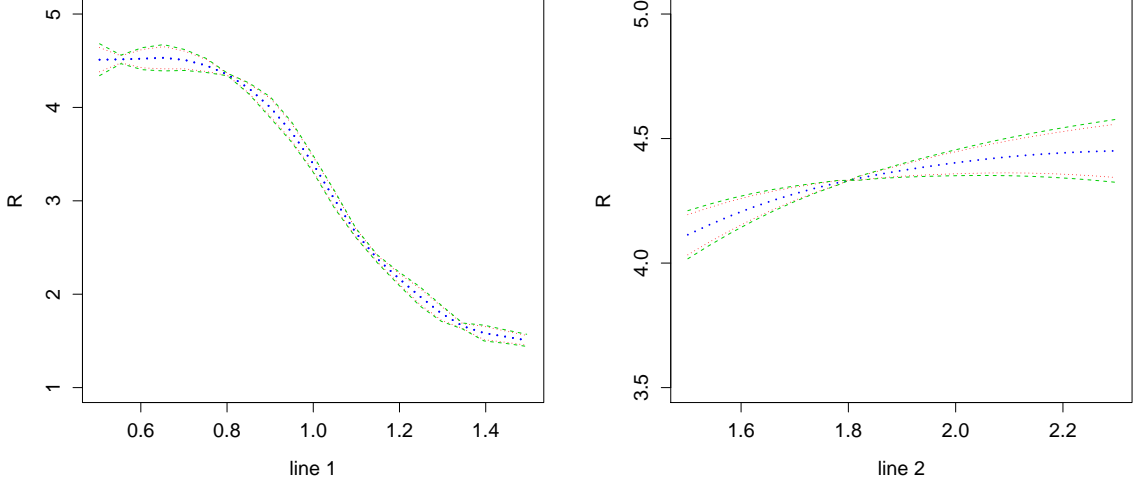


Figure 7: Comparison between Bayesian co-kriging and non Bayesian co-kriging. The thick dotted lines represent the prediction mean, the thin dashed lines represent the prediction confidence interval at plus or minus twice the standard deviation in the Bayesian case and the thin dotted lines represent the prediction confidence interval at plus or minus twice the standard deviation in the non-Bayesian case. The locations of line 1 and line 2 in the input space are plotted in Figure 6.

where:

$$m_{Z_s}(x) = h'_s(x)^T \beta + t_s(x)^T V_s^{-1} (z - H_s \beta) \quad (39)$$

and:

$$s_{Z_s}^2(x) = \sigma_{Z_s}^2 - t_s(x)^T V_s^{-1} t_s(x) \quad (40)$$

Furthermore, let us denote by $R_t = R_t(D_t)$ the correlation matrix for D_t and $\rho_s = 0, \forall s \leq 0$. The matrix V_s has the form:

$$V_s = \begin{pmatrix} V^{(1,1)} & \dots & V^{(1,s)} \\ \vdots & \ddots & \vdots \\ V^{(s,1)} & \dots & V^{(s,s)} \end{pmatrix} \quad (41)$$

The s diagonal blocks of size $n_t \times n_t$ are defined by:

$$V^{(t,t)} = \sigma_t^2 R_t(D_t) + \sigma_{t-1}^2 \rho_{t-1}^2 R_{t-1}(D_t) + \dots + \sigma_1^2 \left(\prod_{i=1}^{t-1} \rho_i^2 \right) R_1(D_t) \quad (42)$$

and the off-diagonal blocks of size $n_t \times n_{t'}$ are given by:

$$V^{(t,t')} = \left(\prod_{i=t}^{t'-1} \rho_i \right) V^{(t,t)}(D_t, D_{t'}) \quad (43)$$

9.2 Some important results about the covariance matrix V_s

V_s is an $(\sum_{i=1}^s n_i \times \sum_{i=1}^s n_i)$ matrix, its inverse can hence be difficult to process. We present in this Subsection a method to reduce the complexity of the processing of V_s^{-1} . From the previous section the covariance matrix V_s can be written as:

$$V_s = \begin{pmatrix} V_{s-1} & U_{s-1} \\ U_{s-1}^T & V^{(s,s)} \end{pmatrix} \quad U_{s-1} = \begin{pmatrix} V^{(1,s)} \\ \vdots \\ V^{(s-1,s)} \end{pmatrix} = \begin{pmatrix} \rho_{s-1} V^{(1,s-1)}(D_1, D_s) \\ \vdots \\ \rho_{s-1} V^{(s-1,s-1)}(D_{s-1}, D_s) \end{pmatrix}$$

By sorting the experimental design sets such that:

$$\forall t = 2, \dots, s \quad D_{t-1} = (x_1^{(t-1)}, \dots, x_{n_{t-1}-n_t}^{(t-1)}, x_1^{(t)}, \dots, x_{n_t}^{(t)}) = (D_{t-1} \setminus D_t, D_t)$$

it can be shown that $\forall t = 2, \dots, s$ the inverse of the matrix V_s has the form:

$$V_s^{-1} = \begin{pmatrix} V_{s-1}^{-1} + \begin{pmatrix} 0 & 0 \\ 0 & \rho_{s-1}^2 \frac{R_s^{-1}}{\sigma_s^2} \end{pmatrix} & - \begin{pmatrix} 0 \\ \rho_{s-1} \frac{R_s^{-1}}{\sigma_s^2} \end{pmatrix} \\ - \begin{pmatrix} 0 & \rho_{s-1} \frac{R_s^{-1}}{\sigma_s^2} \end{pmatrix} & \frac{R_s^{-1}}{\sigma_s^2} \end{pmatrix} \quad (51)$$

$$V_1^{-1} = \frac{R_1^{-1}}{\sigma_1^2}$$

with V_{s-1}^{-1} an $(\sum_{i=1}^{s-1} n_i \times \sum_{i=1}^{s-1} n_i)$ matrix and R_s^{-1} an $(n_s \times n_s)$ matrix. This is a very important result since it shows that we can deduce V_s^{-1} from R_t^{-1} , $t = 1, \dots, s$. Therefore, the complexity of the processing of V_s^{-1} is $\mathcal{O}(\sum_{i=1}^s n_i^3)$ instead of $\mathcal{O}((\sum_{i=1}^s n_i)^3)$. Furthermore, from the equation (51) and the Bayesian estimation of parameters presented in Section 9.1, we have shown here that building a s -level co-kriging is equivalent to build s independent krigings.

Since $(t_1^*(x, D_1)^T, \dots, t_{s-1}^*(x, D_{s-1})^T) = \rho_{s-1} t_{s-1}^T(x)$ it can also be shown that in the equation (39):

$$t_s(x)^T V_s^{-1} = \left(\rho_{s-1} t_{s-1}^T(x) V_{s-1}^{-1} - [0_{1 \times (\sum_{i=1}^{s-1} n_i - n_s)}, \rho_{s-1} R_s(\{x\}, D_s) R_s^{-1}], R_s(\{x\}, D_s) R_s^{-1} \right) \quad (52)$$

Therefore, $t_s(x)^T V_s^{-1}$ is independent of σ_s^2 . Since $t_1(x)^T V_1^{-1} = R_1(\{x\}, D_1) R_1^{-1}$ does not depend on σ_1^2 , by induction, $t_s(x)^T V_s^{-1}$ is independent of σ_i^2 for all $1 \leq i \leq s$. We have just shown here that the co-kriging mean does not depend on the variance coefficients.

Finally, the determinant of the covariance matrix is given by:

$$|V_s| = \prod_{i=1}^s (\sigma_i^2)^{n_i} |R_i|$$

Therefore, $|V_s|$ does not depend on the adjustment coefficients. It strengthens the result presented in Section 9.1 which shows that we can independently estimate the variance parameters $(\sigma_i^2)_{i=1, \dots, s}$ and the hyper-parameters $(\theta_t)_{t=1, \dots, s}$ as well.

10 Fast cross-validation for kriging and co-kriging surrogate models

The idea of a cross-validation procedure is to split the training set into two disjoint sets, one is used for training and the other is used to monitor the performance of the model. Then, the performance on the validation set is used as a proxy for the generalization error. A particular case of this method is the Leave-One-Out Cross-Validation (noted LOO-CV) where n validation sets are obtained by removing one observation at a time. This procedure can be time-consuming for a kriging model but [Dubrule (1983)] shows that there are computational shortcuts. These shortcuts are also presented by [Marcotte (1995)], [Rasmussen & Williams (2006)] and [Zhang & Wang (2009)] and we present in this section their adaptation for co-kriging models. Furthermore, the cross-validation equations proposed in this section extend the ones of [Dubrule (1983)] since they do not suppose that the regression and the variance coefficients are known. Therefore, only the hyper-parameters of the correlation function are fixed and the other parameters are re-estimated at each training set. We note that the re-estimation of the variance coefficient is an original result which is important since fixing this parameter can lead huge errors for the estimation of the cross-validation predictive variance when the number of observations is small or when the number of points in the validation set is important.

If we denote by ξ_s the n_{train} indices of points in D_s constituting the training set D_{train} and ξ_t with $1 \leq t < s$ the corresponding points in D_t - indeed, we have $D_s \subset D_{s-1} \subset \dots \subset D_1$, therefore $D_{train} \subset D_t$ and if we denote $D_t^{\xi_t} = (x_{\xi_t(1)}^{(t)}, x_{\xi_t(2)}^{(t)}, \dots, x_{\xi_t(n_{train})}^{(t)})$, we have $D_t^{\xi_t} = D_{train}$ for all $1 \leq t \leq s$. The nested experimental design assumption implies that, in the cross-validation procedure, if we remove a point from D_s we also have to remove it from D_t , $t < s$. Considering, the hyper-parameters θ as known, the Woodbury formula presented in [Harville (1997)] and the results of Section 9.2, it can be shown that the vectors of the cross-validation predictive errors ϵ_{Z_s, ξ_s} and variances ς_{Z_s, ξ_s} at points in the training set D_{train} are given by the recursive equations (53) and (54).

$$(\epsilon_{Z_t, \xi_t} - \rho_{t-1} \epsilon_{Z_{t-1}, \xi_{t-1}}) [R_t^{-1}]_{[\xi_t, \xi_t]} = [R_t^{-1} (z_t - H_t \lambda_{t, -\xi_t})]_{[\xi_t]} \quad (53)$$

$$\varsigma_{Z_t, \xi_t} = \rho_{t-1}^2 \varsigma_{Z_{t-1}, \xi_{t-1}} + \sigma_{t, -\xi_t}^2 \text{diag} \left(\left([R_t^{-1}]_{[\xi_t, \xi_t]} \right)^{-1} \right) \quad (54)$$

where $1 \leq t \leq s$, $H_t = [\rho_{t-1} z_{t-1}(D_t) \quad F_t]$ $t > 1$, $H_1 = F_1$ and:

$$\lambda_{t, -\xi_t} (H_{t, -\xi_t}^T K_t H_{t, -\xi_t}) = H_{t, -\xi_t}^T K_t z_t(D_{t, -\xi_t}) \quad (55)$$

$$\sigma_{t, -\xi_t}^2 = \frac{(z_t(D_{t, -\xi_t}) - H_{t, -\xi_t} \lambda_{t, -\xi_t})^T K_t (z_t(D_{t, -\xi_t}) - H_{t, -\xi_t} \lambda_{t, -\xi_t})}{n_t - p_t - q_{t-1} - n_{train}} \quad (56)$$

$$K_t = [R_t^{-1}]_{[-\xi_t, -\xi_t]} - [R_t^{-1}]_{[-\xi_t, \xi_t]} \left([R_t^{-1}]_{[\xi_t, \xi_t]} \right)^{-1} [R_t^{-1}]_{[\xi_t, -\xi_t]} \quad (57)$$

We note that we can easily adapt these formulas if we just remove points from D_t , $t > t_0 \geq 1$, since we will so have $\epsilon_{Z_r, \xi_r} = 0$ and $\varsigma_{Z_r, \xi_r} = 0$ for $r \leq t_0$. Furthermore, these equations are also valid when $s = 1$, i.e. for kriging model. We hence have closed form expression for the equations of a k -fold cross-validation with a re-estimation of the regression and variance parameters and directly deductible from the co-kriging equations. The complexity of this

procedure is monitored by the inversion of the matrix $[R_t^{-1}]_{[\xi_t, \xi_t]}$ of size $n_{train} \times n_{train}$. We also note that if we suppose parameters of variance and/or regression as known, we do not have to compute $\sigma_{t, -\xi_t}^2$ and/or $\lambda_{t, -\xi_t}$ which reduces substantially the complexity of the method. When the variance parameter is known we find the equations presented by [Dubrule (1983)].

Notations: $A_{[\xi, \xi]}$ is the submatrix of elements $\xi \times \xi$ of A , $a_{[\xi]}$ is the subvector of elements ξ of a , $B_{-\xi}$ represents the matrix B minus the rows of indice ξ , $C_{[-\xi, -\xi]}$ is the submatrix of C in which we remove the elements of indice $-\xi \times -\xi$ and $C_{[-\xi, \xi]}$ is the submatrix of C in which we remove the row of indice ξ and keep the column of indice ξ .

11 Example 2: Fluidized-Bed Process

This example illustrates the comparison between 2-level and 3-level co-kriging. A 3-level co-kriging method is applied to a physical experiment modelled by a computer code. The experiment, which is the measurement of the temperature of the steady-state thermodynamic operation point for a fluidized-bed process, was presented by [Dewettinck, De Visscher, Deroo, Huyghebaert (1999)], who developed a computer model named ‘‘Topsim’’ to calculate the measured temperature. The code, developed for a Glatt GPCG-1, fluidized-bed unit in the top-spray configuration, can be run at 3 levels of complexity. We hence have 4 available responses:

1. T_{exp} : the experimental response.
2. T_3 : the most accurate code modelling the experiment.
3. T_2 : a simplified version of T_3 .
4. T_1 : the lowest accurate code modelling the experiment.

The differences between T_1 , T_2 and T_3 are discussed by Dewettinck et al. (1999). The aim of this study is to predict the experimental response T_{exp} given the two levels of code T_3 and T_2 . We only focus on a 3-level co-kriging using T_3 and T_2 to predict T_{exp} since 28 responses available for each level is not enough for a relevant 4-level co-kriging. The experimental design set and the responses T_1 , T_2 , T_3 and T_{exp} are given by [Qian & Wu (2008)] who have presented a 2-level co-kriging using T_{exp} and T_2 . Furthermore, the responses are parameterized by a 6-dimensional input vector presented by Dewettinck et al. (1999).

11.1 Building the 3-level co-kriging

To build the 3-level co-kriging, we use 10 measures of T_{exp} (measures 1, 3, 8, 10, 12, 14, 18, 19, 20, 27 in Table 4 in [Qian & Wu (2008)]), 20 simulations of T_3 (runs 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 19, 20, 22, 24, 27) and the 28 simulations of T_2 and the input vector is scaled between 0 and 1. The last 18 measures of T_{exp} are used for validation. The design sets are nested such that $(D_{t-1} = (D_{t-1} \setminus D_t, D_t))_{t=2,3}$ and we use a Matern $_{\frac{5}{2}}$ kernel for the three covariance functions. The estimations of the hyper-parameters which represent correlation lengths of the three covariance kernels are given in Table 6.

$\hat{\theta}_1$	1.790	3.988	1.218	1.790	3.595	0.722
$\hat{\theta}_2$	1.810	1.842	2.008	1.036	0.001	0.345
$\hat{\theta}_3$	0.890	0.721	2.008	2.952	1.790	0.241

Table 6: Example: fluidized-bed process. Estimation of the hyper-parameters (correlation lengths) for the 3-level co-kriging.

The estimations of hyper-parameters in Table 6 show us that the surrogate model will be very smooth in the first four directions. For the fifth direction the Gaussian processes modelling the cheap code T_2 and the bias between T_{exp} and T_3 are very smooth and the one modelling the bias between T_3 and T_2 is close to a regression. Finally, the model is sharper in the sixth direction in particular for the two biases where correlation lengths are around 0.3.

Furthermore, Table 7 gives the estimation of the variance and regression parameters (see section 9.1).

Regression coefficient	Posterior mean	Posterior Covariance σ_t^2
β_1	47.02	0.134
$\begin{pmatrix} \beta_{\rho_1} \\ \beta_2 \end{pmatrix}$	$\begin{pmatrix} 0.97 \\ -0.17 \end{pmatrix}$	$\begin{pmatrix} 0.001 & -0.034 \\ -0.034 & 1.610 \end{pmatrix}$
$\begin{pmatrix} \beta_{\rho_2} \\ \beta_3 \end{pmatrix}$	$\begin{pmatrix} 0.95 \\ 1.93 \end{pmatrix}$	$\begin{pmatrix} 0.003 & -0.121 \\ -0.121 & 5.188 \end{pmatrix}$
Variance coefficient	Q_t	α_t
σ_1^2	1032	13.5
σ_2^2	5.30	9
σ_3^2	8.39	4

Table 7: Example: fluidized-bed process. Bayesian estimation of the variance and regression parameters for the 3-level co-kriging.

Table 7 shows that the responses have approximately the same scale since the adjustment coefficients are close to 1. Furthermore, we see an important bias between T_3 and T_2 with $\beta_3 = 1.93$. Finally, the variance coefficients for the biases indicate that they are possibly much simpler to model than the cheap code T_2 as their estimations are smaller.

11.2 3-level co-kriging prediction: predictions when code output is available

The aim of this Section is to show that co-kriging can improve significantly the accuracy of the surrogate model at points where at least one level of responses is available.

The predictions of the 3-level co-kriging are here presented and compared with the predictions obtained with a 2-level co-kriging using only the 10 responses of T_{exp} and the 20 responses of T_3 . The predictions for the 2-level and the 3-level co-krigings vs. the real values (i.e., the measured temperature T_{exp}) are shown in Figure 8. The 3-level co-kriging gives us the same

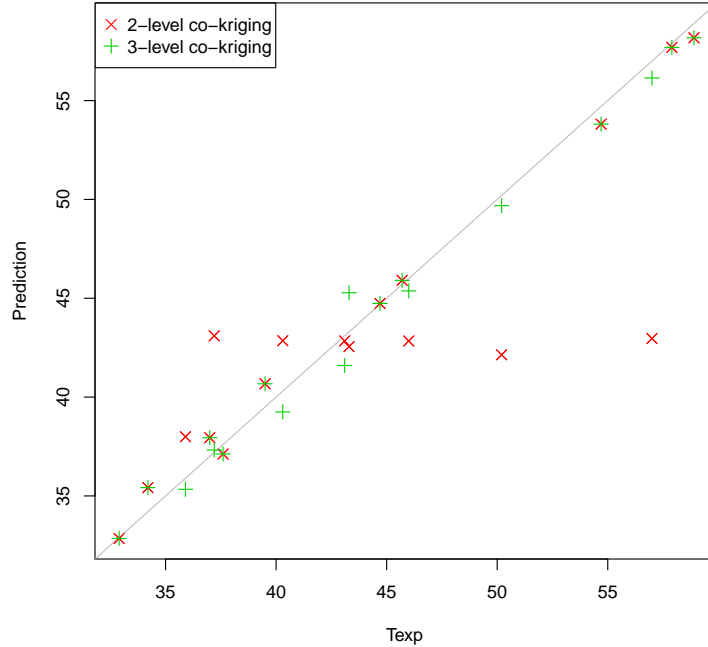


Figure 8: Predictions of the 2-level and the 3-level co-krigings for the fluidized-bed process. The 3-level co-kriging improves significantly the predictions of the 2-level one.

prediction means as the 2-level co-kriging at the 10 points (points 2, 5, 6, 7, 9, 11, 13, 16, 22, 24) where T_3 is known. These overlapped points mean that T_2 does not influence the surrogate model at these points. This follows from the Markov property introduced in Section 3, which implies that the prediction of T_{exp} is entirely determined by T_3 at these points. We also note that, in general, the 2-level co-kriging predictions - at points where T_3 is unknown - are not accurate and the 3-level co-kriging improves significantly the prediction means compared to the 2-level co-kriging. Table 8 compares the 2-level co-kriging with the 3-level co-kriging and summarizes some results about the quality of the predictions on the 18 validation points. Nonetheless, it is important to notice that, in the 3-level case, the output of the cheapest code T_2 is known at the 18 test points. This means that the results of this subsection show that the 3-level co-kriging prediction is more accurate than the 2-level co-kriging prediction at a point where the cheapest response T_2 is available. In the next subsection we show that the 3-level co-kriging prediction is more accurate than the 2-level one at a point where no response is available.

	Q_2	RMSE	MaxAE
2-level co-kriging	61.23 %	4.24	14.04
3-level co-kriging	98.71 %	0.89	1.98
	Average Std. dev.	Median Std. dev.	Maximal Std. dev.
2-level co-kriging	2.90	1.02	5.68
3-level co-kriging	0.90	1.02	1.04

Table 8: Example: fluidized-bed process. Comparison between 2-level co-kriging and 3-level co-kriging. Predictions are better in the 3-level case and the prediction variance seems well-evaluated since the RMSE and the average standard deviation are close.

Figure 9 shows the prediction errors of the 2-level co-kriging and the confidence interval at plus or minus twice the prediction standard deviation. The last 10 prediction errors and their confidence intervals are the same as those of the 3-level case since it corresponds to the points where T_3 is known. We see in Figure 9 that the confidence intervals are well predicted.

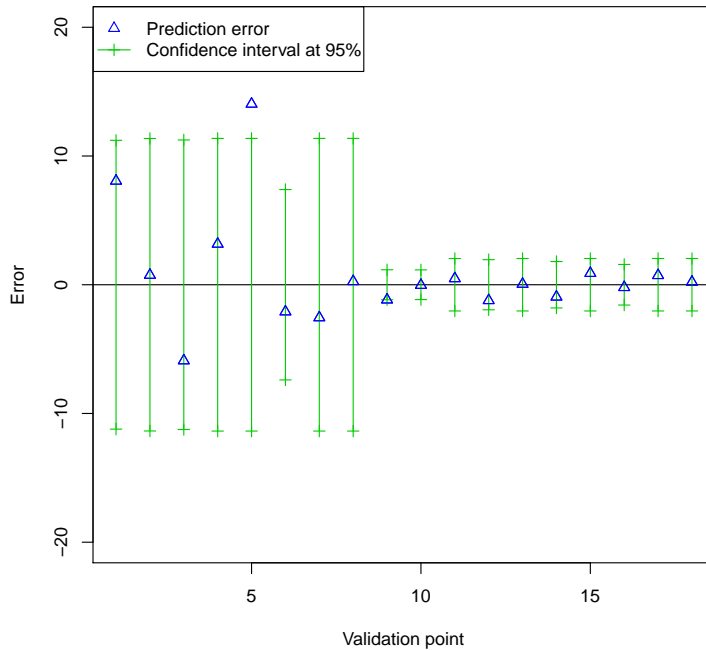


Figure 9: Prediction errors of the 2-level co-kriging and confidence intervals at plus or minus twice the standard deviation. We see a significant difference between the accuracy of the predictions means and their confidence intervals for the point where T_3 is unknown (the 8 first validation points) and for the ones where it is known (the last 10 validation points).

Furthermore, we see a significant difference between the accuracy of the prediction means and their confidence intervals for the point where T_3 is unknown (the 8 first validation points) and for the ones where it is known (the last 10 validation points).

11.3 3-level co-kriging prediction: predictions when code output is not available

In this subsection, we show that a multi-level co-kriging can significantly improve the prediction of a surrogate model at points where no response is available.

We have seen in Section 11.2 that the 3-level co-kriging improves significantly the 2-level co-kriging at points where T_3 is unknown and T_2 has been sampled. Nevertheless, to have a fair comparison between these two co-kriging models we can compare their accuracy at points where no response is available. We apply the Leave-One-Out Cross-Validation (LOO-CV) procedure at the 10 points where T_{exp} is known by using the formulas presented in Section 10. This means that we perform for each of these 10 points the following procedure:

1. The experimental and the two code outputs corresponding to the point are removed from the data set.
2. The 2-level co-kriging method and the 3-level co-kriging method are applied using the truncated data set in order to give a confidence interval for the experimental output at the point.

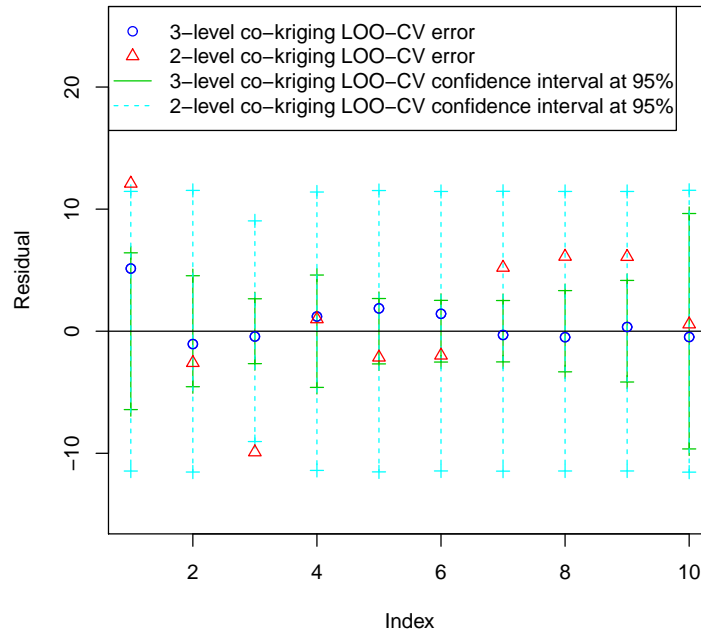


Figure 10: Leave-One-Out Cross-Validation predictive errors and variances of the 2-level and 3-level co-kriging. We see that the confidence intervals are accurate and the precision of the 3-level co-kriging is significantly better than the one of the 2-level co-kriging.

Figure 10 shows the result of the LOO-CV procedure for the 2-level and 3-level co-kriging. We see that the 3-level co-kriging is more accurate than the 2-level one. Indeed, the LOO-CV RMSE for the 2-level co-kriging is equal to 1.88 whereas it is equal to 1.09 for the 3-level co-kriging. This shows that the 3-level co-kriging provides better predictions also at points where no response is available. This highlights the strength of the proposed method and shows that a co-kriging method with more than 2 levels of code can be worthwhile.

12 Conclusion

We have presented a method for building kriging models using a hierarchy of codes with different levels of accuracy. This method allows us to improve a surrogate model built on a complex code using information from a cheap one. It is particularly useful when the complex code is very expensive. We see in our literature review that the first multi-level metamodel originally suggested is a first order auto-regressive model built with Gaussian processes. The AR(1) relation between two levels of code is natural and the building of the model is straightforward. Nevertheless, we have highlighted some key issues which makes it difficult to use this model in practical ways.

First, important parameters of the model, which are the adjustment coefficients between two successive levels of codes, were numerically estimated. We propose here an analytical estimation of these parameters with a Bayesian method. This method allows us to have information about the uncertainties of the estimations and above all, to easily use the AR(1) model and its generalization to the case of non-spatial stationarity. Furthermore, a strength of the proposed method is that it even works for a code with more than 2 levels since its implementation is such that the estimations of the parameters of a s -level co-kriging is equivalent as the ones of s independent krigings. It is important to highlight that this method is based on a joint estimation between the adjustment coefficient and the mean of the Gaussian process modelling the difference between two successive levels of code.

Second, we have seen that the variance of the predictive distribution of the AR(1) model could be underestimated. A natural approach to improve this estimation is a Bayesian modelling. We propose here a Bayesian co-kriging for 2 levels of code and to avoid computationally expensive implementation, we suggest another model than the one presented. This new model is based on a hierarchical specification of the parameters of the model. This allows us to have a Bayesian model including only two nested integrations without Markov chain Monte Carlo procedure.

Finally, for a non-Bayesian s -level co-kriging, we have proved that building a s -level co-kriging is equivalent to build s independent krigings. This result is very important since it solves one of the most important key issues of the co-kriging which is the inversion of the covariance matrix. A 3-level co-kriging example has been provided to show the efficiency of the presented method.

13 Acknowledgements

The author particularly thanks Professor Josselin Garnier for his constructive suggestions, helpful comments and fruitful guidance. He is also grateful to Dr. Claire Cannaméla for providing the data for example 1 and her interesting discussions.

A The case of ρ depending on x

A.1 Building a model with s levels of code

Let us consider s levels of code, if we note $\beta = (\beta_1^T, \dots, \beta_s^T)^T$, $\beta_\rho = (\beta_{\rho_1}^T, \dots, \beta_{\rho_{s-1}}^T)^T$, $\sigma^2 = (\sigma_1^2, \dots, \sigma_s^2)$ and $\theta = (\theta_1, \dots, \theta_s)$, we have:

$$\forall x \in Q \quad [Z_s(x)|\mathcal{Z} = z, \beta, \beta_\rho, \sigma^2, \theta] \sim \mathcal{N}(m_{Z_s}(x), s_{Z_s}^2(x))$$

where:

$$m_{Z_s}(x) = h'_s(x)^T \beta + t_s(x)^T V_s^{-1} (z - H_s \beta) \quad (58)$$

and:

$$s_{Z_s}^2(x) = \sigma_{Z_s}^2(x) - t_s(x)^T V_s^{-1} t_s(x) \quad (59)$$

Let us define the following notation:

$$\bigodot_{i=k}^l A_i = A_k \odot \dots \odot A_l$$

where \odot represents the matrix element-by-element product. Furthermore, let us denote by $\rho_t = \rho_t(D_t)$ the vector containing the values of $\rho_t(x)$, $x \in D_t$. $R_t = R_t(D_t)$ is the correlation matrix for D_t and $\rho_s(x) = 0$, $\forall s \leq 0$. The matrix V_s has the form:

$$V_s = \begin{pmatrix} V^{(1,1)} & \dots & V^{(1,s)} \\ \vdots & \ddots & \vdots \\ V^{(s,1)} & \dots & V^{(s,s)} \end{pmatrix} \quad (60)$$

The s diagonal blocks of size $n_t \times n_t$ are defined by:

$$V^{(t,t)} = \sigma_t^2 R_t(D_t) + \sigma_{t-1}^2 (\rho_{t-1}(D_t) \rho_{t-1}^T(D_t)) \odot R_{t-1}(D_t) + \dots + \sigma_1^2 \left(\bigodot_{i=1}^{t-1} \rho_i(D_t) \rho_i^T(D_t) \right) \odot R_1(D_t) \quad (61)$$

and the off-diagonal blocks of size $n_t \times n_{t'}$ are given by:

$$V^{(t,t')} = \left(\mathbf{1}_{n_t} \left(\bigodot_{i=t}^{t'-1} \rho_i(D_{t'}) \right)^T \right) \odot V^{(t,t)}(D_t, D_{t'}) \quad (62)$$

where $1 \leq t < t' \leq s$.

The vector $t_s(x)$ is such that $t_s(x) = (t_1^*(x, D_1)^T, \dots, t_s^*(x, D_s)^T)^T$, where:

$$t_t^*(x, D_t)^T = \rho_{t-1}^T(D_t) \odot t_{t-1}^*(x, D_t)^T + \left(\prod_{i=t}^{s-1} \rho_i(x) \right) \sigma_t^2 R_t(x, D_t) \quad (63)$$

where $1 < t \leq s$, $\left(\prod_{i=s}^{s-1} \rho_i(x) \right) = 1$ and $t_1^*(x, D_1)^T = \left(\prod_{i=1}^{s-1} \rho_i(x) \right) \sigma_1^2 R_1(x, D_1)$. Furthermore, if we define:

$$F_k(D_l) = \begin{pmatrix} f_k^T(x_1^{(l)}) \\ \vdots \\ f_k^T(x_{n_l}^{(l)}) \end{pmatrix} \quad 1 \leq k, l \leq s$$

it can be shown that $\forall t = 2, \dots, s$ the inverse of the matrix V_s has the form:

$$V_s^{-1} = \left(\begin{array}{c} V_{s-1}^{-1} + \left(\begin{array}{cc} 0 & 0 \\ 0 & (\rho_{s-1}(D_s)\rho_{s-1}^T(D_s)) \odot \frac{R_s^{-1}}{\sigma_s^2} \end{array} \right) \\ - \left(\begin{array}{cc} 0 & (\mathbf{1}_{n_s}\rho_{s-1}^T(D_s)) \odot \frac{R_s^{-1}}{\sigma_s^2} \end{array} \right) \end{array} \right) - \left(\begin{array}{c} 0 \\ (\rho_{s-1}(D_s)\mathbf{1}_{n_s}^T) \odot \frac{R_s^{-1}}{\sigma_s^2} \\ \frac{R_s^{-1}}{\sigma_s^2} \end{array} \right) \right) \quad (70)$$

$$V_1^{-1} = \frac{R_1^{-1}}{\sigma_1^2}$$

with V_{s-1}^{-1} an $(\sum_{i=1}^{s-1} n_i \times \sum_{i=1}^{s-1} n_i)$ matrix and R_s^{-1} an $(n_s \times n_s)$ matrix.

It can also be shown that:

$$t_s(x)^T V_s^{-1} = \left(\rho_{s-1}(x)t_{s-1}^T(x)V_{s-1}^{-1} - [0_{1 \times (\sum_{i=1}^{s-1} n_i - n_s)}, \rho_{s-1}^T(D_s) \odot R_s(\{x\}, D_s)R_s^{-1}], R_s(\{x\}, D_s)R_s^{-1} \right) \quad (71)$$

A.4 Bayesian prediction for a code with 2 levels

The equations for the Bayesian prediction when ρ depends on x can be directly derived from the Section 5 by replacing ρ with β_ρ and noting that the design matrix F is such that:

$$F = [F_\rho(D_2) \odot (z_1(D_2)\mathbf{1}_{p_\rho}^T) \quad F_2]$$

Finally, for the Bayesian prediction, we just have to adapt the integral (32) :

$$p(z_2(x)|z_1, z_2, \sigma_1^2, \sigma_2^2) = \int_{\mathbb{R}^{p_\rho+p_2}} p(z_2(x)|z_1, z_2, \beta_\rho, \beta_2, \sigma_1^2, \sigma_2^2) p(\beta_\rho, \beta_2|z_1, z_2, \sigma_1^2, \sigma_2^2) d\beta_\rho d\beta_2 \quad (72)$$

References

- [Floater & Iske (1996)] FLOATER, M.S. & ISKE, A. 1996 Multistep scattered data interpolation using compactly supported radial basis function *J. Comput. Appl. Math.* **73**, 1-15.
- [Santner, Williams & Notz (2003)] SANTNER, T. J., WILLIAMS, B. J. & NOTZ, W. I. 2003 *The Design and Analysis of Computer Experiments*. New York: Springer.
- [Rasmussen & Williams (2006)] RASMUSSEN, C. E. & WILLIAMS, C. K. I. 2006 *Gaussian Processes for Machine Learning*. the MIT Press.
- [Kennedy & O'Hagan (2000)] KENNEDY, M. C. & O'HAGAN, A. 2000 Predicting the output from a complex computer code when fast approximations are available. *Biometrika* **87**, 1-13.
- [Forrester, Sobester & Keane (2007)] FORRESTER, A. I. J., SOBESTER, A. & KEANE, A. J. 2007 Multi-fidelity optimization via surrogate modelling. *Proc. R. Soc. A* **463**, 3251-3269.
- [Jones, Schonlau & Welch (1998)] JONES, D. R., SCHONLAU, M. & WELCH, W. J. 1998 Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* **13**, 455-492.

- [Qian et al. (2006)] QIAN, Z., SEEPERSAD, C. C., JOSEPH, V. R., ALLEN, J. K. & JEFF WU, C. F. 2006 Building Surrogate Models Based on Detailed and Approximate Simulations. *Journal of Mechanical Design* **128**, 668-677.
- [Qian & Wu (2008)] QIAN, Z. & JEFF WU, C. F. 2008 Bayesian Hierarchical Modeling for Integrating Low-accuracy and High-accuracy Experiments. *Technometrics* **50**, 192-204.
- [Jeffreys (1961)] JEFFREYS, H. 1961 Theory of Probability. *Oxford University Press*, London.
- [Patterson & Thompson (1971)] PATTERSON, H. D. & THOMPSON, R. 1971 Recovery of interblock information when block sizes are unequal. *Biometrika* **58**, 545-554.
- [Cumming & Goldstein (2009)] CUMMING, J. A. & GOLDSTEIN, M. 2009 Small Sample Bayesian Designs for Complex High-Dimensional Models Based on Information Gained Using Fast Approximations. *Technometrics* **51**, 377-388.
- [Goldstein & Wooff (2007)] GOLDSTEIN, M., & WOUFF, D. A. 2007 *Bayes Linear Statistics: Theory and Methods*. Chichester, England: Wiley.
- [McKay, Beckman & Conover (1979)] MCKAY, M. D., BECKMAN, R. J. & CONOVER, W. J. 1979 A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239-245.
- [Morris & Mitchell (1995)] MORRIS, M. D. & MITCHELL, T. J. 1995 Exploratory designs for computer experiments. *J. Stat. Plan. Infer.* **43**, 381-402.
- [Owen (1992)] OWEN, A. 1992 Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica* **2**, 439-452.
- [Tang (1993)] TANG, B. 1993 Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association* **88**, 1392-1397.
- [Fang, Lin, Winker & Zhang (2000)] FANG, K. T., LIN, D. K. J., WINKER, P. & ZHANG, Y. 2000 Uniform design: Theory and application. *Technometrics* **42**, 237-248.
- [Qian, Ai & Wu (2009)] QIAN, Z., AI, M. & JEFF WU, C. F. 2009 Construction of nested space-filling designs. *The Annals of Statistics* **37**, 3616-3643.
- [Qian, Tang & Wu (2009)] QIAN, Z., TANG, B. & JEFF WU, C. F. 2009 Nested space-filling designs for computer experiments with two levels of accuracy. *Statistica Sinica* **19**, 287-300.
- [Gregoire, Souffland & Gauthier (2005)] GREGOIRE, O., SOUFFLAND, D. & GAUTHIER, S. 2005 A second-order turbulence model for gaseous mixtures induced by Richtmyer-Meshkov instability. *Journal of Turbulence*, Volume 6, Art No. N 29.
- [Lehmann & Casella (1998)] LEHMANN, E. & CASELLA, G. 1998 *Theory of Point Estimation*. Springer-Verlag, New York, revised edition.
- [Dewettinck, De Visscher, Deroo, Huyghebaert (1999)] DEWETTINCK, K., DE VISSCHER, A., DEROO, L. & HUYGHEBAERT, A. 1999 Modeling the steady-state thermodynamic operation point of top-spray fluidized bed processing. *Journal of Food Engineering* **39**, 131-143.

- [Marcotte (1995)] MARCOTTE, D. 1995 Generalized Cross-Validation for Covariance Model Selection. *Mathematical Geology* **27**, No. 5.
- [Dubrule (1983)] DUBRULE, O. 1983 Cross Validation in a Unique Neighborhood. *Mathematical Geology* **15**, No. 6.
- [Zhang & Wang (2009)] ZHANG, H. & WANG, Y. 2009 Kriging and cross-validation for massive spatial data. *Environmetrics* **21**, 290-304.
- [Harville (1997)] HARVILLE, D. A. 1997 *Matrix Algebra from Statistician's Perspective*. Springer-Verlag Inc.