# An adaptive kriging method for characterizing uncertainty in inverse problems

FU Shuai[12]
[1] *University Paris-Sud 11 & INRIA, Department of Mathematics*
*91405 Orsay, France*
*E-mail: fshvip@gmail.com*

Couplet Mathieu[2], Bousquet Nicolas[2]
[2] *EDF R&D, Department of Industrial Risk Management*
*6 quai Watier*
*78401 Chatou, France*
*E-mail: mathieu.couplet@edf.fr, nicolas.bousquet@edf.fr*

## 1 Industrial context of the model

Our present work is based on a simplified model which is related to the risk of dyke overflow during a flood. We want to predict the water level in this case, with a number of variables availables : the observed quantities are the water level at the dyke position $Z_c$, the speed of the river $V$ and the observed flow of river $Q$. The unobservable quantities are the value of Strickler coefficient $K_s$ as well as the river bed level at the dyke $Z_v$. In order to predict the future water level, we need to quantify the non-observed variables $K_s$ and $Z_v$.

In this flooding model, if we define

- the data vector $Y = (Z_c, V)^T \in \mathbb{R}^2$;
- the non-observed vector $X = (K_s, Z_v)^T \in \mathbb{R}^2$;
- the observed variable $d = Q \in \mathbb{R}^1$,

we can rewrite the measured observation $Y$ as a function of the non-observed vector $X$, adding an error $U$ :

$$Y = (Z_c, V)^T + U = H(K_s, Z_v; Q) + U.$$

In general, we consider the following model for the uncertainty treatment:

$$(1) \quad Y_i = H(X_i, d_i) + U_i, (1 \leq i \leq n)$$

with $n$ the sample size, and

- $(Y_i) \in \mathbb{R}^p$ denotes the data vectors;
- $H$ denotes a known function from $\mathbb{R}^{q+q_2}$ to $\mathbb{R}^p$ which is expensive in CPU-time consumption and often regarded as a "black box";
- $(X_i) \in \mathbb{R}^q$ denotes the non-observed random data which is assumed independent and identically distributed (i.i.d.);
- $(d_i) \in \mathbb{R}^{q_2}$ denotes the observed variables related to the experimental conditions;
- $(U_i) \in \mathbb{R}^p$ denotes the measurement errors which are assumed i.i.d. Besides, the variables $(X_i)$ and $(U_i)$ are assumed to be independent.

To solve this inverse problem we choose a Bayesian framework and we assume the following *priori* distributions for the random variables $X_i$ and $U_i$ :

$$\begin{aligned} X_i \,|\, m, C &\sim \mathcal{N}_q(m, C); \\ U_i &\sim \mathcal{N}_p(\mathbf{0}, R),\ (1 \leq i \leq n), \end{aligned}$$

which allows us to take into account the experts' *prior* knowledge as well as to reduce the problem of identifiability. Moreover, assuming a known $R$, the *prior* distributions of $m$ and $C$ can be defined as follows :

$$\begin{aligned} m \,|\, C &\sim \mathcal{N}_q(\mu, C/a),\ \text{with } a \text{ an hyperparameter to be specified;} \\ C &\sim \mathcal{IW}_q(\Lambda, \nu) \in \mathcal{M}^{q \times q}. \end{aligned}$$

where $\mathcal{IW}$ denotes a Inverse-Wishart distribution. Typically, we want to estimate the *posterior* distribution of $\theta = (m, C)$ knowing the data $\mathbf{Y} = (Y_1, \ldots, Y_n)$, which caracterizes the distribution of $X_i$. We choose a Gibbs sampling to approximate the Bayesian *posterior* distributions. In order to define a Gibbs sampler, noting $\mathbf{X} = (X_1, \ldots, X_n)$ and $\mathbf{Y} = (Y_1, \ldots, Y_n)$, we calculate the full conditional *posterior* distribution for each unknown quantity $(m, C, \mathbf{X})$ :

$$\begin{aligned} m \,|\, C, \mathbf{Y}, \mathbf{X}, \rho &\sim \mathcal{N}\Big(\frac{a}{n+a}\mu + \frac{n}{n+a}\overline{\mathbf{X}},\ \frac{C}{n+a}\Big) \\ C \,|\, m, \mathbf{Y}, \mathbf{X}, \rho &\sim \mathcal{IW}\Big(\Lambda + \sum_{i=1}^{n}(m - X_i)(m - X_i)' + a(m-\mu)(m-\mu)'.\, \nu + n + 1\Big) \\ \mathbf{X} \,|\, \mathbf{Y}, m, C, \rho &\propto \exp\Big\{ -\frac{1}{2}\sum_{i=1}^{n}\Big[(X_i - m)'C^{-1}(X_i - m) + (Y_i - H(X_i, d_i))'R^{-1}(Y_i - H(X_i, d_i))\Big]\Big\}. \end{aligned}$$

(2)

with $\rho$ the set of hyperparameters. As the last conditional distribution (2) of X is under a complicated and strange form, it is necessary to apply a numerical method for simulation, Metropolis-Hastings for example.

## 2 New kriging version of the model

In general, the CPU-time consuming for the function $H$ is very large (for the reason of complicated code of $H$). It is desirable to limit the number of calls to the function $H$. Here we propose a kriging approximation method where $\widehat{H}$. $H$ is regarded as the realisation of a Gaussian process $\mathcal{H}$. To apply this method, we choose a bounded hypercube $\Omega \subset \mathbb{R}^Q$ $(Q = q + q_2)$ and generate a set of $N_{\max}$ points $D = \{z_1, \ldots, z_{N_{\max}}\} \subset \Omega$ with $z_i = (x_i, d_i)$, applying a Latin Hypercube Sampling (LHS) - *maximin* strategy, called a *design*. Our approximation is restricted to $\Omega$ and the number of calls to $H$ is limited to $N_{\max}$, noted by $H_D = \{H(z_1), \ldots, H(z_{N_{\max}})\}$.

The predictor $\widehat{H}$ is calculated as a conditional experance for any point $z_0 \in \Omega$

$$\widehat{H}(z_0) = \mathbb{E}(\mathcal{H}(z_0)|\mathcal{H}_D = H_D),$$

and its conditional variance denoted by $\mathrm{MSE}(z_0)$ (Mean Squared Error) can be seen as a mesure of the prediction accuracy. Following this idea, each $Y_i$ can also be seen as a realisation of a new process

$\mathcal{Y}_i$ which corresponds with the Gaussian process $\mathcal{H}$, we rewrite the current model:

$$\begin{aligned} \mathcal{Y}_i &= \widehat{H}(X_i,d_i) + \underbrace{(\mathcal{H}-\widehat{H})(X_i,d_i)+U_i} \\ &= \widehat{H}(X_i,d_i)+V_i(X_i,d_i) \end{aligned} \tag{3}$$

with the new error term $V_i(X_i,d_i)$ combining two types of variance : the $R$ of the old error $U_i$ and the MSE of the kriging method. We regard especially the whole sample $\boldsymbol{\mathcal{Y}} = (\mathcal{Y}_1,\ldots,\mathcal{Y}_n)$, which permets us to consider the correlation between the predictions $\widehat{H}(Z_i)$ and $\widehat{H}(Z_j)$. Thus model (3) can be written (assuming that $p=2$ for simplicity):

$$\boldsymbol{\mathcal{Y}} = \begin{pmatrix} \mathcal{Y}_1^1 \\ \vdots \\ \mathcal{Y}_n^1 \\ \mathcal{Y}_1^2 \\ \vdots \\ \mathcal{Y}_n^2 \end{pmatrix} = \begin{pmatrix} \widehat{H}_1^1(Z_1) \\ \vdots \\ \widehat{H}_n^1(Z_n) \\ \widehat{H}_1^2(Z_1) \\ \vdots \\ \widehat{H}_n^2(Z_n) \end{pmatrix} + \begin{pmatrix} V_1^1(Z_1) \\ \vdots \\ V_n^1(Z_n) \\ V_1^2(Z_1) \\ \vdots \\ V_n^2(Z_n) \end{pmatrix} = \widehat{H}(\mathbf{Z}) + V(\mathbf{Z}). \tag{4}$$

We deduce its distribution given the variable $\mathbf{Z} = (Z_1,\ldots,Z_{N_{\max}})$ and the function values $H_D$ of the design:

$$\boldsymbol{\mathcal{Y}} \,|\, \mathbf{Z}, H_D \;\sim\; \mathcal{N}(\widehat{H}(\mathbf{Z}), \mathbf{R} + \mathrm{MSE}(\mathbf{Z})), \tag{5}$$

where $\mathbf{R}$ is a diagonal variance matrix:

$$\mathbf{R} \;=\; \left(\begin{array}{ccc|ccc} R^{11} & & & & & \\ & \ddots & & & \mathbf{0} & \\ & & R^{11} & & & \\ \hline & & & R^{22} & & \\ & \mathbf{0} & & & \ddots & \\ & & & & & R^{22} \end{array}\right), \qquad \begin{array}{l} \left.\vphantom{\begin{array}{c}1\\1\\1\end{array}}\right\} n \text{ times} \\ \left.\vphantom{\begin{array}{c}1\\1\\1\end{array}}\right\} n \text{ times} \end{array} \tag{6}$$

with $R^{ii}$ the $i-$th diagonal component of $R$; and $\mathrm{MSE}(\mathbf{Z})$ a block diagonal matrix composed by the $\mathrm{MSE}^i(\mathbf{Z})$ which is the variance-covariance matrix of $\mathcal{H}^i(\mathbf{Z})$:

$$\mathrm{MSE}(\mathbf{Z}) \;=\; \left(\begin{array}{c|c} \mathrm{MSE}^1(\mathbf{Z}) & \mathbf{0} \\ \hline \mathbf{0} & \mathrm{MSE}^2(\mathbf{Z}) \end{array}\right). \qquad \begin{array}{l} \left.\vphantom{\begin{array}{c}1\\1\end{array}}\right\} n \text{ times} \\ \left.\vphantom{\begin{array}{c}1\\1\end{array}}\right\} n \text{ times} \end{array}$$

The conditional distribution of the vector $\mathbf{X} \in \mathbb{R}^{p\times n}$ knowing $\boldsymbol{\mathcal{Y}}$ can be specified :

$$\begin{aligned} \mathbf{X} \,|\, \boldsymbol{\mathcal{Y}}, m, C, \rho, \mathbf{d}, H_D \;\propto\; & 1_\Omega \Big|\mathbf{R} + \mathrm{MSE}(\mathbf{Z})\Big|^{-\frac{1}{2}} \exp\Bigg\{ -\frac{1}{2}\sum_{i=1}^{n} \Big[(X_i-m)'C^{-1}(X_i-m)\Big] \\ & -\frac{1}{2}\Big(\big(\mathcal{Y}_1-\widehat{H}(Z_1)\big)',\ldots,\big(\mathcal{Y}_n-\widehat{H}(Z_n)\big)'\Big)\Big(\mathbf{R}+\mathrm{MSE}(\mathbf{Z})\Big)^{-1} \begin{pmatrix} \big(\mathcal{Y}_1-\widehat{H}(Z_1)\big) \\ \vdots \\ \big(\mathcal{Y}_n-\widehat{H}(Z_n)\big) \end{pmatrix} \Bigg\}. \end{aligned} \tag{7}$$

# 3 Hybrid MCMC Algorithm

We explicit the Gibbs algorithm as follows:

Given $(m^{[r]}, C^{[r]}, \mathbf{X}^{[r]})$ for $r = 0, 1, 2, \dots$, generate

1. $C^{[r+1]}|\cdots \sim \mathcal{IW}\Big(\Lambda + \sum_{i=1}^{n}(m^{[r]} - X_i^{[r]})(m^{[r]} - X_i^{[r]})' + a(m^{[r]} - \mu)(m^{[r]} - \mu)',\ \nu + n + 1\Big)$
2. $m^{[r+1]}|\cdots \sim \mathcal{N}\Big(\frac{a}{n+a}\mu + \frac{n}{n+a}\overline{\mathbf{X}^{[r]}},\ \frac{C^{[r+1]}}{n+a}\Big)$
3. $\mathbf{X}^{[r+1]}|\cdots \sim ?$

The simulation of $\mathbf{X}^{[r+1]}$ consists of $l$ iterations of Metropolis-Hastings algorithm. By consequence, our Gibbs sampling algorithm becomes a so-called *hybrid MCMC algorithm*, following Tierney (1994), which simultaneously utilizes both Gibbs sampling steps and Metropolis-Hastings steps. In detail,

- Let $\mathbf{X}_0 = (X_{1,0}, \dots, X_{n,0})' = \mathbf{X}^{[r]}$
- For $s = 1, \dots, l$, updating $\mathbf{X}^{[r]}$ component by component (for $i = 1, \dots, n$):
    1. Generate $\widetilde{X}_{i,s} \sim J(\cdot \mid X_{i,s-1})$ where $J$ is a proposal distribution
    2. Let

       $$\alpha(X_{i,s-1}, \widetilde{X}_{i,s}) \quad = \quad \min\Big(\frac{\pi_{\widehat{H}}(\widetilde{\mathbf{X}}_s \mid \boldsymbol{\mathcal{Y}}, \theta^{[r+1]}, \rho, \mathbf{d}, H_D)\, J(X_{i,s-1}|\widetilde{X}_{i,s})}{\pi_{\widehat{H}}(\mathbf{X}_{s-1} \mid \boldsymbol{\mathcal{Y}}, \theta^{[r+1]}, \rho, \mathbf{d}, H_D)\, J(\widetilde{X}_{i,s}|X_{i,s-1})}, 1\Big), \tag{8}$$

       where

       $$\begin{aligned} \widetilde{\mathbf{X}}_s &= \Big(X_{1,s},\, \dots,\, X_{i-1,s},\, \widetilde{X}_{i,s},\, X_{i+1,s-1},\, \dots,\, X_{n,s-1}\Big)' \\ \mathbf{X}_{s-1} &= \Big(X_{1,s},\, \dots,\, X_{i-1,s},\, X_{i,s-1},\, X_{i+1,s-1},\, \dots,\, X_{n,s-1}\Big)' \end{aligned}$$

    3. Accept

       $$X_{i,s} \quad = \quad \begin{cases} \widetilde{X}_{i,s} & \texttt{with probability } \alpha(X_{i,s-1}, \widetilde{X}_{i,s}), \\ X_{i,s-1} & \texttt{otherwise} \end{cases}$$

    4. Renovation $\mathbf{X}_s = \Big(X_{1,s},\, \dots,\, X_{i,s},\, X_{i+1,s-1},\, \dots,\, X_{n,s-1}\Big)'$
- $\mathbf{X}^{[r+1]} = \mathbf{X}_l$

In Step 2 of the MH algorithm, we consider a gaussian proposal distributions $J = \mathcal{N}\Big(m^{[r+1]}, C^{[r+1]}\Big)$.

# 4 Construction of an adaptive design

An important point for a kriging approximation is the choice of the design of experience (DoE). In order to fill the whole field, we use a classic strategy called Latin Hypercube Sampling (LHS) - *maximin*. But this general method is not always perfect, sometimes the design points are relatively far from the true value of $X$ and sometimes the MSE may explode at the edge of the experimental field. We propose to construct a design adaptively by providing the additional information sequentially. The substitution is as follows :

1. Fix $N_{\max}$ as calculation budget and a proportion $p$ of points.
2. Choose a hypercubic domain $\Omega$ where proxy is valid.
3. Build a design LHS-*maximin* $D$ with $p \times N_{\max}$ points in $\Omega$.
4. Decide whether the design is satisfactory (quality criterion). If it is, we call the kriging predictor $\hat{H}$ and we run the Gibbs algorithm (M-H).
5. If it isn't, we add the other $(1-p) \times N_{\max}$ points sequentially to the indicated area according to an adaptive procedure to improve the quality of design.

The adaptive scheme requires many choices: the experimental field $\Omega$; the proportion $p$ of points selected for the design LHS-*maximin*; a criterion to measure the quality of a design; the adaptive strategy of adding the points, etc.

**Small experience : with different number $N_{\max}$ of points** The number $N_{\max}$ of points of the design $D$ is very important. If we increase this number, we improve the accuracy of the kriging approximation. But on the other hand, the computation requires more time. In the following numerical experiments, we consider three different cases : $N_{\max} = 100, 300, 500$. The Figure 1 gives us an illustration : when we increase the points of the design $D$ from 100 to 200, there is an obvious decrease of the MSE.

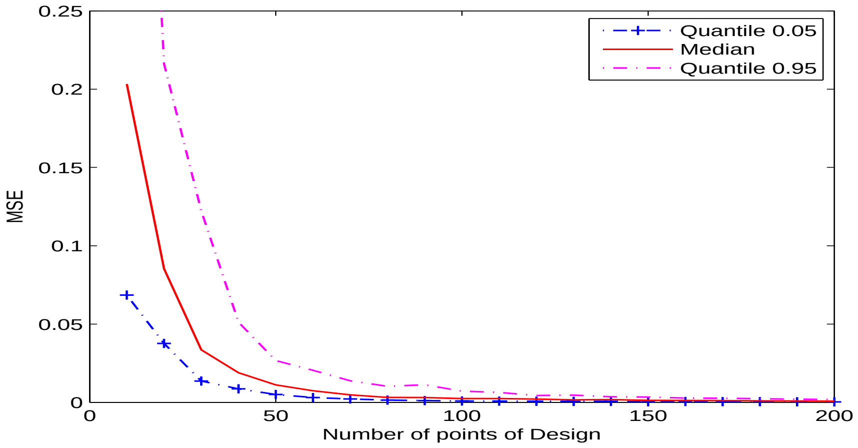

Figure 1: MSE with an increasing number $N_{\max}$ of the points of $D$

Now we consider three LHS-*maximin* designs with 100 points, 300 points and 500 points and we compare the *posterior* distributions of two parameters $m$ and $C$ by using these three kriging methods, based on a sample of size 1000, drawn from the simulated Monte Carlo chain after the burn-in period. We can notice a great difference when we apply the 100-point kriging.

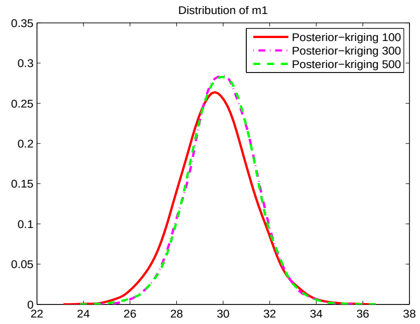

Figure 2: $m_1$

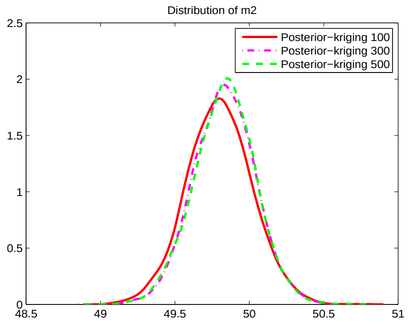

Figure 3: $m_2$

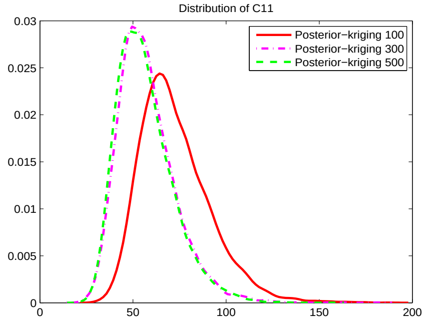

Figure 4: $C_{11}$

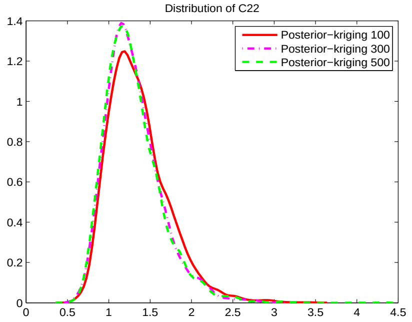

Figure 5: $C_{22}$

**How to choose adaptively the $N_{\max}$ points of design?** Here we give a procedure well detailed as follows :

1. Fix an experimental field $\Omega$ and generate a design $D'$ of $p \times N_{\max}$ ($p = 90\%$ for example) points in $\Omega$ according to the LHS - maximin strategy.
2. Divide the field $\Omega$ densely enough ($100 \times 100 \times 50$ respectively for $x_1$,$x_2$ and $d$ for example).
3. At each grid point $(x_1, x_2, d) = (x, d)$, we calculate its criterion value $\mathcal{C}^D(x, d)$ with two possible types :

   (a) weighted criterion :

   $$\max_{(x,d)\in E} \mathrm{MSE}(x,d)^{\alpha} \cdot \pi(x)^{1-\alpha}$$

   where

   $$\pi(x) \propto \left[1 + (x-\mu)' \left((1+\frac{1}{a}) \cdot \Lambda\right)^{-1} (x-\mu)\right]^{-\frac{\nu+1}{2}},$$

   and MSE$(x, d)$ can be obtained easily with the package DACE;

   (b) weighted criterion taking into account the $y_i$ (at iteration $k$) :

   $$\max_{(x,d)\in E} \mathrm{MSE}(x,d)^{\alpha} \cdot \sum_i \pi(x|y_i, \theta^{[k]})^{1-\alpha}$$

   where $\pi(x|y_i, \theta^{[k]})$ has to be updated at each iteration of MCMC.

4. Complete the design sequentially with $(1-p)\times N_{\max}$ additional points which maximise the criterion quantity. For the moment, we use only the first criterion by logarithm, which means that

$$\mathcal{C}^D(x,d) \quad = \quad \alpha \log(\text{MSE}(x,d)) \; - \; \frac{\nu+1}{2}(1-\alpha)\log\left[1+(x-\mu)'\left((1+\frac{1}{a})\cdot\Lambda\right)^{-1}(x-\mu)\right]$$

where $\alpha$ could be chosen between 0 and 1.

If we take $p = 90\%, N_{\max} = 100$, with different $\alpha$ we obtain the following graphs Figure 6, 7 and 8, where Figure 6 represents the design $D'$ with 90 points generated by LHS - Maximin plus 10 additional points with the biggest MSE value, and in Figure 7 and 8, we add 10 points according to the MSE value as well as the *prior* distribution of $X$ where the *prior* mean $\mu$ is all supposed $[40, 48]'$, but the weight $\alpha = 0.8$ for Figure 7 and $\alpha = 0.5$ for Figure 8. The strategy of adding points does allow us to diminish the kriging uncertainty. We can compare the final result : the *posterior* distributions of the mean parameter $m$ (Figure 9 and 10) in three cases and we use a 500-point-kriging result as a reference. We find that in our example, with $\alpha = 1$ (we only consider the MSE impact), the simulation results satisfy us the most.

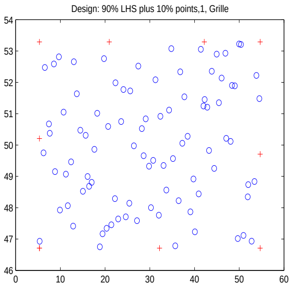

Figure 6: $\alpha = 1$

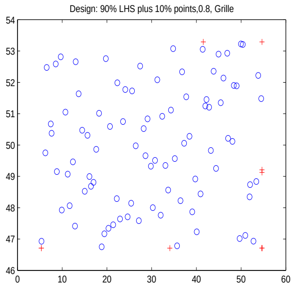

Figure 7: $\alpha = 0.8$

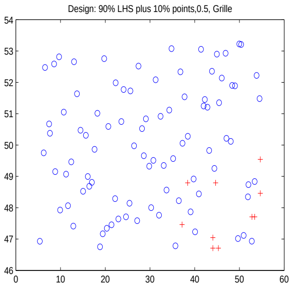

Figure 8: $\alpha = 0.5$

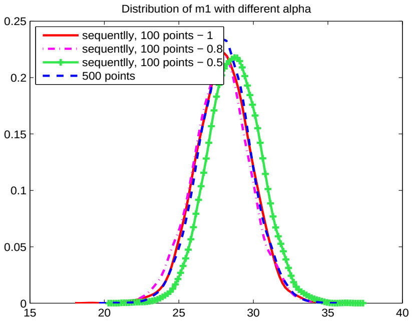

Figure 9: $m_1$

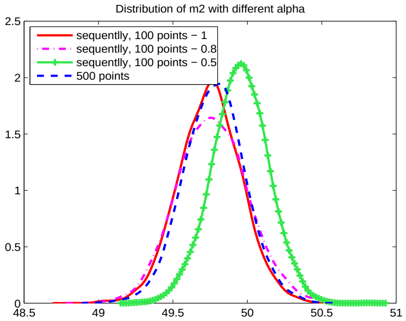

Figure 10: $m_2$

Why did we choose $p = 90\%$? In fact, in order to make a best choice of $p$, we repeat the experiment with different $p$ and we obtain the following graph (Figure 11). Obviously, $p = 80\% - 90\%$ seems to be a good solution.

**Remark :** When we add the $(1-p)\times N_{\max}$ points to $D'$, we have to distinguish their physical positions to avoid adding all the points in the same area. So we construct the DoE sequentially by adding one new point and recalculating the criterion $\mathcal{C}^D$ each time.

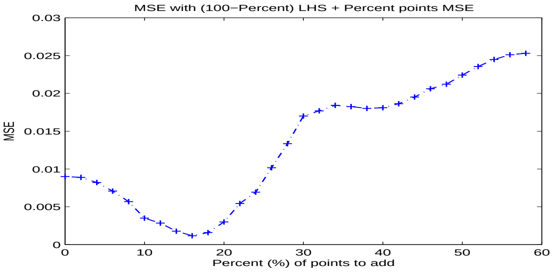

Figure 11: MSE with an increasing number of points to add

To illustrate the advantage of this adaptive idea, we compare the experimental results of the 100-point-adaptive-kriging methods by always using a 500-point-kriging method as a reference (Figure 12, 13, 14 and 15). There is a great improvement when we apply the adaptive kriging method with 100 points, especially for $m_2$ and $C_{22}$.

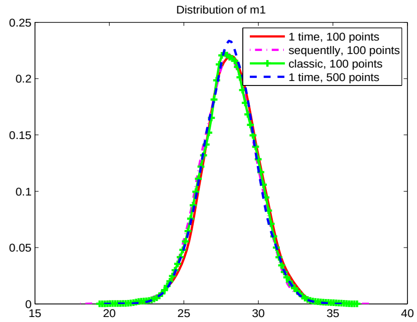

Figure 12: $m_1$

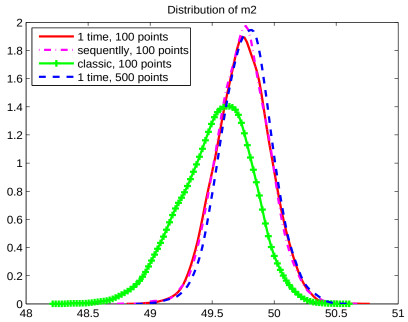

Figure 13: $m_2$

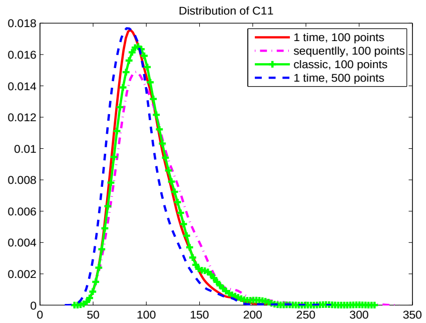

Figure 14: $C_{11}$

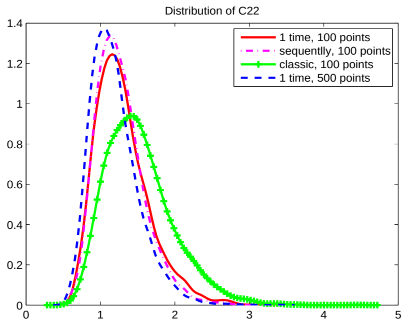

Figure 15: $C_{22}$

## REFERENCES (RÉFERENCES)

[1] Robert C. P. and Casella G. (2004). *Monte Carlo Statistical Methods*. Second Edition. New York: Springer.

[2] Gelman A., Carlin J. B., Stern H.S. and Rubin D.B. (2004). *Bayesian Data Analysis*. Second Edition. Chapman & Hall/Crc.

[3] Celeux G., Grimaud A., Lefebvre Y. and De Rocquigny E. (2009) Identifying variability in multivariate systems through linearised inverse methods. *Inverse Problems In Science & Engineering*, à aparaître.

[4] Barbillon P., Celeux G., Grimaud A., Lefebvre Y. and De Rocquigny E. (2009) Non linear methods for inverse statistical problems. *Rapport de research INRIA*.

[5] Bettinger R. (2009) Inversion d'un systme par krigeage. *Thesis, Nice-Sophia Antipolis University.*

[6] Dubourg V., Deheeger F. (2010) Une alternative  la substitution pour les mta-modles en analyse de fiabilit. *Journes Nationales de la Fiabilit, Toulouse.*

[7] Picheny V., Ginsbourger D., Roustant O., Haftka R.T., Kim N-H. Adaptive Designs of Experiments for Accurate Approximation of a Target Region.

[8] Williams B., Santner T. and Notz W. (2000) Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica.*

[9] Scheidt C. (2006) Analyse statistique d'expriences simules: Modlisation adaptative de rponses non-rgulires par krigeage et plans d'expriences. *Thesis, Louis Pasteur University, Strasbourg.*

[10] Leonardo S. Bastos and Anthony O'Hagan (2009) Diagnostics for Gaussian Process Emulators. *University of Sheffield*