

Multi-fidelity co-kriging models

Application to Sequential design

Loic Le Gratiet^{1,2}, Claire Cannamela³

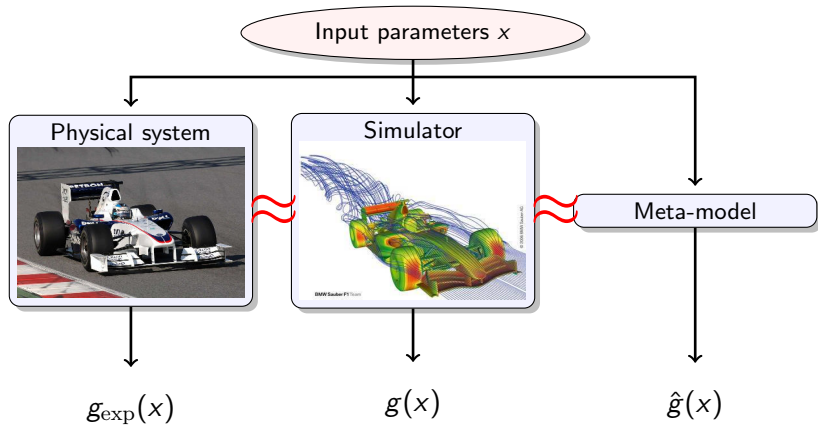
¹EDF R&D, Chatou, France

²UNS CNRS, 06900 Sophia Antipolis, France

³CEA, DAM, DIF, F-91297 Arpajon, France

ANR CHORUS
April 30, 2014

Context

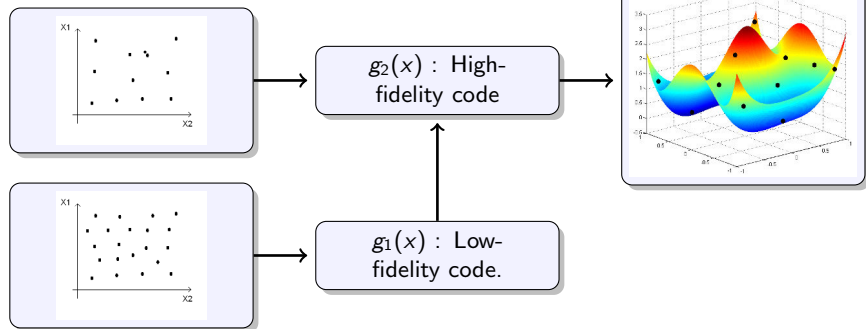


Motivations

- **Objective:** replace the output of a code, called $g_2(x)$, by a metamodel.

$$g_2(x) : x \in Q \subset \mathbb{R}^d \mapsto \mathbb{R}$$

- **Framework:** a coarse version g_1 of g_2 is available.



Principle: build a metamodel of $g_2(x)$ which integrates as well observations of the coarse code output. \rightarrow Multi-fidelity co-kriging model

Recursive formulation of the model

- ▶ **Multi-fidelity co-kriging model:** [Kennedy & O'Hagan (2000), Le Gratiet (2013), Le Gratiet (2014)]

$$\begin{cases} Z_2(x) = \rho Z_1^*(x) + \delta(x) \\ Z_1^*(x) \perp \delta(x) \end{cases}$$

where $Z_1^*(x) \sim [Z_1(x) | \mathbf{Z}_1 = \mathbf{g}_1, \beta_1, \sigma_1^2, \theta_1]$, with $\mathbf{g}_1 = \mathbf{g}_1(x), x \in \mathbf{D}_1$

and $Z_1(x) \sim \text{GP}(\mathbf{f}_1^t(x)\beta_1, \sigma_1^2 r_1(x, \tilde{x}; \theta_1))$, $\delta(x) \sim \text{GP}(\mathbf{f}_\delta^t(x)\beta_\delta, \sigma_\delta^2 r_\delta(x, \tilde{x}; \theta_\delta))$

- ▶ **Parameters estimation:**
 - ▶ $\theta_1, \theta_\delta, \sigma_1^2, \sigma_\delta^2$: maximum likelihood method
 - ▶ $\beta_1, \begin{pmatrix} \beta_\delta \\ \rho \end{pmatrix}$: analytical posterior distribution (Bayesian inference)

Recursive formulation of the model

- ▶ **Multi-fidelity co-kriging model:** [Kennedy & O'Hagan (2000), Le Gratiet (2013), Le Gratiet (2014)]

$$\begin{cases} Z_2(x) = \rho Z_1^*(x) + \delta(x) \\ Z_1^*(x) \perp \delta(x) \end{cases}$$

where $Z_1^*(x) \sim [Z_1(x) | \mathbf{Z}_1 = \mathbf{g}_1, \beta_1, \sigma_1^2, \theta_1]$, with $\mathbf{g}_1 = g_1(x), x \in \mathbf{D}_1$

and $Z_1(x) \sim \text{GP}(\mathbf{f}_1^t(x)\beta_1, \sigma_1^2 r_1(x, \tilde{x}; \theta_1))$, $\delta(x) \sim \text{GP}(\mathbf{f}_\delta^t(x)\beta_\delta, \sigma_\delta^2 r_\delta(x, \tilde{x}; \theta_\delta))$

- ▶ **Parameters estimation:**
 - ▶ $\theta_1, \theta_\delta, \sigma_1^2, \sigma_\delta^2$: maximum likelihood method
 - ▶ $\beta_1, \begin{pmatrix} \beta_\delta \\ \rho \end{pmatrix}$: analytical posterior distribution (Bayesian inference)
- ▶ Finally, $Z_2^*(x) \sim [Z_2(x) | \mathbf{Z}_2 = \mathbf{g}_2, \mathbf{Z}_1 = \mathbf{g}_1]$ with $\mathbf{g}_2 = g_2(x), x \in \mathbf{D}_2$

We suppose that $\mathbf{D}_2 \subset \mathbf{D}_1$ and $\theta_1, \theta_\delta, \sigma_1^2, \sigma_\delta^2$ are known.

Recursive formulation of the model

- ▶ **Multi-fidelity co-kriging model:** [Kennedy & O'Hagan (2000), Le Gratiet (2013), Le Gratiet (2014)]

$$\begin{cases} Z_2(x) = \rho Z_1^*(x) + \delta(x) \\ Z_1^*(x) \perp \delta(x) \end{cases}$$

where $Z_1^*(x) \sim [Z_1(x) | \mathbf{Z}_1 = \mathbf{g}_1, \beta_1, \sigma_1^2, \theta_1]$, with $\mathbf{g}_1 = g_1(x), x \in \mathbf{D}_1$

and $Z_1(x) \sim \text{GP}(\mathbf{f}_1^t(x)\beta_1, \sigma_1^2 r_1(x, \tilde{x}; \theta_1))$, $\delta(x) \sim \text{GP}(\mathbf{f}_\delta^t(x)\beta_\delta, \sigma_\delta^2 r_\delta(x, \tilde{x}; \theta_\delta))$

- ▶ **Parameters estimation:**
 - ▶ $\theta_1, \theta_\delta, \sigma_1^2, \sigma_\delta^2$: maximum likelihood method
 - ▶ $\beta_1, \begin{pmatrix} \beta_\delta \\ \rho \end{pmatrix}$: analytical posterior distribution (Bayesian inference)
- ▶ Finally, $Z_2^*(x) \sim [Z_2(x) | \mathbf{Z}_2 = \mathbf{g}_2, \mathbf{Z}_1 = \mathbf{g}_1]$ with $\mathbf{g}_2 = g_2(x), x \in \mathbf{D}_2$

We suppose that $\mathbf{D}_2 \subset \mathbf{D}_1$ and $\theta_1, \theta_\delta, \sigma_1^2, \sigma_\delta^2$ are known.

Predictive distribution

- ▶ In **Universal Cokriging**, the predictive distribution of $Z_2^*(x)$ is **not Gaussian**.

The predictive mean and variance can be **decomposed** as:

$$\begin{aligned}\mu_{Z_2}(x) &= \mathbb{E}[Z_2(x)|\mathbf{Z}_2 = \mathbf{g}_2, \mathbf{Z}_1 = \mathbf{g}_1] \\ &= \hat{\rho}\mu_{Z_1}(x) + \mu_\delta(x)\end{aligned}$$

$$\begin{aligned}\sigma_{Z_2}^2(x) &= \text{var}(Z_2(x)|\mathbf{Z}_2 = \mathbf{g}_2, \mathbf{Z}_1 = \mathbf{g}_1) \\ &= \hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x) + \sigma_\delta^2(x)\end{aligned}$$

- ▶ **Remarks:**

- ▶ in $\mu_{Z_2}(x)$: β and ρ are replaced by their posterior means.
- ▶ in $\sigma_{Z_2}^2(x)$: we infer from the posterior distributions of β and ρ .

Generalizations

- ▶ **Generalizations** for the AR(1) model:
 - ▶ $s > 2$ levels of code.
 - ▶ $\rho(x) = f'_\rho(x)\beta_\rho$.
 - ▶ Bayesian formulation.
 - ▶ Non-nested experimental design sets (see L. Le Gratiet thesis 2013).
- ▶ Extend the AR(1) approach (see F. Zertuche):

$$Z_2(x) = \psi(Z_1(x)) + \delta(x)$$

- ▶ Other Bayesian formulation with (see Qian and Wu 2008):
 - ▶ $\rho(x)$ a Gaussian process.
 - ▶ $z_1(x)$ is supposed as known.

Sequential design

- ▶ **Objective:** we want to minimize the following generalization error:

$$\text{IMSE} = \int_Q \sigma_{Z_2}^2(x) dx = \hat{\sigma}_\rho^2 \int_Q \sigma_{Z_1}^2(x) dx + \int_Q \sigma_\delta^2(x) dx$$

- ▶ **Sequential strategy:** we select a new point x_{n+1} such that :

$$x_{n+1} = \arg \max_x \sigma_{Z_2}^2(x)$$

Sequential design

- ▶ **Objective:** we want to minimize the following generalization error:

$$\text{IMSE} = \int_Q \sigma_{Z_2}^2(x) dx = \hat{\sigma}_\rho^2 \int_Q \sigma_{Z_1}^2(x) dx + \int_Q \sigma_\delta^2(x) dx$$

- ▶ **Sequential strategy:** we select a new point x_{n+1} such that :

$$x_{n+1} = \arg \max_x \sigma_{Z_2}^2(x)$$

- ▶ **Question :** which code level should be simulated ?

- ▶ What is the contribution of each code level to the model error?

$$\sigma_{Z_2}^2(x_{n+1}) = \hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) + \sigma_\delta^2(x_{n+1})$$

- ▶ What is computational cost of each code ?
- ▶ What is the expected reduction of the generalization error ?

Sequential design

- ▶ **Objective:** we want to minimize the following generalization error:

$$\text{IMSE} = \int_Q \sigma_{Z_2}^2(x) dx = \hat{\sigma}_\rho^2 \int_Q \sigma_{Z_1}^2(x) dx + \int_Q \sigma_\delta^2(x) dx$$

- ▶ **Sequential strategy:** we select a new point x_{n+1} such that :

$$x_{n+1} = \arg \max_x \sigma_{Z_2}^2(x)$$

- ▶ **Question :** which code level should be simulated ?

- ▶ What is the contribution of each code level to the model error?

$$\sigma_{Z_2}^2(x_{n+1}) = \hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) + \sigma_\delta^2(x_{n+1})$$

- ▶ What is computational cost of each code ?
- ▶ What is the expected reduction of the generalization error ?

Code level selection

- ▶ What is computational cost of each code ?
 - ▶ C : CPU time ration between $g_2(x)$ and $g_1(x)$.
 - ▶ 1 run of $g_1(x)$ and $g_2(x) \Leftrightarrow C + 1$ runs of $g_1(x)$ (i.e. $D_2 \subset D_1$)
- ▶ What is the expected reduction of the error?
 - ▶ Reduction of the generalization error for $Z_1(x)$:

$$\hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i$$

- ▶ Reduction of the generalization error for the bias $\delta(x)$:

$$\sigma_\delta^2(x_{n+1}) \prod_{i=1}^d \theta_\delta^i$$

Code level selection

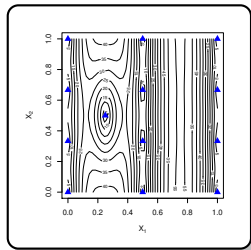
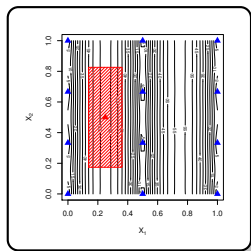
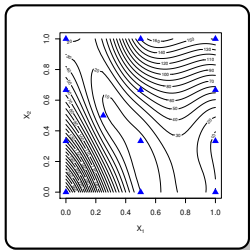
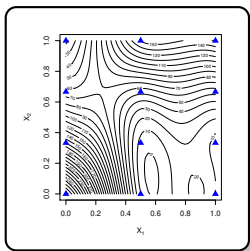
- ▶ What is computational cost of each code ?
 - ▶ C : CPU time ration between $g_2(x)$ and $g_1(x)$.
 - ▶ 1 run of $g_1(x)$ and $g_2(x) \Leftrightarrow C + 1$ runs of $g_1(x)$ (i.e. $D_2 \subset D_1$)
- ▶ What is the expected reduction of the error?
 - ▶ Reduction of the generalization error for $Z_1(x)$:

$$\hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i$$

- ▶ Reduction of the generalization error for the bias $\delta(x)$:

$$\sigma_\delta^2(x_{n+1}) \prod_{i=1}^d \theta_\delta^i$$

Illustration of the design criterion



Code level selection

- ▶ Expected error reduction for the code level 2 :

$$\hat{\sigma}_{\rho}^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i + \sigma_{\delta}^2(x_{n+1}) \prod_{i=1}^d \theta_{\delta}^i$$

- ▶ Expected error reduction for the code level 1 :

$$(C + 1) \hat{\sigma}_{\rho}^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i$$

Code level selection

- ▶ Expected error reduction for the code level 2 :

$$\hat{\sigma}_{\rho}^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i + \sigma_{\delta}^2(x_{n+1}) \prod_{i=1}^d \theta_{\delta}^i$$

- ▶ Expected error reduction for the code level 1 :

$$(C + 1) \hat{\sigma}_{\rho}^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i$$

- ▶ It is worth simulating $g_2(x)$ if :

$$(C + 1) \hat{\sigma}_{\rho}^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i < \hat{\sigma}_{\rho}^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i + \sigma_{\delta}^2(x_{n+1}) \prod_{i=1}^d \theta_{\delta}^i$$

i.e.

Code level selection

- ▶ Expected error reduction for the code level 2 :

$$\hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i + \sigma_\delta^2(x_{n+1}) \prod_{i=1}^d \theta_\delta^i$$

- ▶ Expected error reduction for the code level 1 :

$$(C + 1) \hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i$$

- ▶ It is worth simulating $g_2(x)$ if :

$$(C + 1) \hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i < \hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i + \sigma_\delta^2(x_{n+1}) \prod_{i=1}^d \theta_\delta^i$$

i.e.

$$\frac{\hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i}{\hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i + \sigma_\delta^2(x_{n+1}) \prod_{i=1}^d \theta_\delta^i} < \frac{1}{C + 1}$$

Code level selection

- ▶ Expected error reduction for the code level 2 :

$$\hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i + \sigma_\delta^2(x_{n+1}) \prod_{i=1}^d \theta_\delta^i$$

- ▶ Expected error reduction for the code level 1 :

$$(C + 1) \hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i$$

- ▶ It is worth simulating $g_2(x)$ if :

$$(C + 1) \hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i < \hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i + \sigma_\delta^2(x_{n+1}) \prod_{i=1}^d \theta_\delta^i$$

i.e.

$$\frac{\hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i}{\hat{\sigma}_\rho^2 \sigma_{Z_1}^2(x_{n+1}) \prod_{i=1}^d \theta_1^i + \sigma_\delta^2(x_{n+1}) \prod_{i=1}^d \theta_\delta^i} < \frac{1}{C + 1}$$

New strategy

- ▶ **Problem:** the following criterion does not take into account the real prediction error

$$x_{n+1} = \arg \max_x \sigma_{Z_2}^2(x)$$

- ▶ **New strategy:** to take into account the true prediction error we consider the following criterion (adjusted variance) :

$$x_{n+1} = \arg \max_x \hat{\sigma}_p^2 \sigma_{Z_1, UK}^2(x) \left(1 + \sum_{i=1}^{m_1} \frac{\epsilon_{100,1}^2(x_i^{(1)})}{\sigma_{100,1}^2(x_i^{(1)})} \mathbf{1}_{x \in V_{i,1}} \right) + \sigma_{\delta, UK}^2(x) \left(1 + \sum_{i=1}^{m_2} \frac{\epsilon_{100,\delta}^2(x_i^{(2)})}{\sigma_{100,\delta}^2(x_i^{(2)})} \mathbf{1}_{x \in V_{i,2}} \right)$$

where $V_{i,j}$ is the Voronoi cell associated to $x_i^{(j)}$, $j = 1, 2$, $i = 1, \dots, n_j$.

New strategy

- **Problem:** the following criterion does not take into account the real prediction error

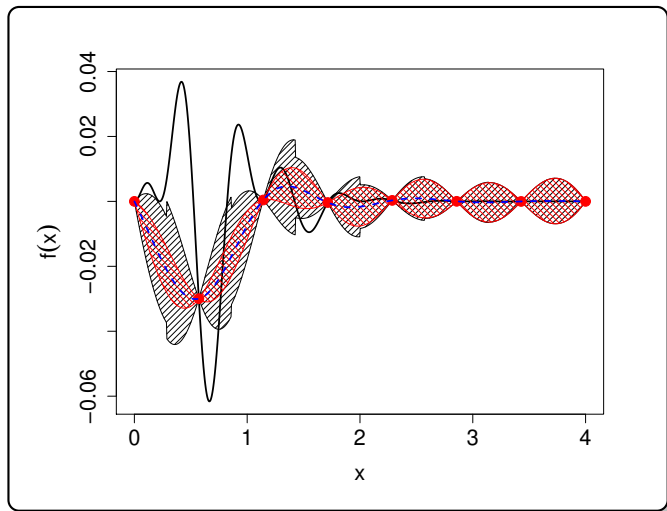
$$x_{n+1} = \arg \max_x \sigma_{Z_2}^2(x)$$

- **New strategy:** to take into account the true prediction error we consider the following criterion (adjusted variance) :

$$x_{n+1} = \arg \max_x \hat{\sigma}_\rho^2 \sigma_{Z_1, UK}^2(x) \left(1 + \sum_{i=1}^{n_1} \frac{\varepsilon_{LOO,1}^2(x_i^{(1)})}{\sigma_{LOO,1}^2(x_i^{(1)})} \mathbf{1}_{x \in V_{i,1}} \right) \\ + \sigma_{\delta, UK}^2(x) \left(1 + \sum_{i=1}^{n_2} \frac{\varepsilon_{LOO,\delta}^2(x_i^{(2)})}{\sigma_{LOO,\delta}^2(x_i^{(2)})} \mathbf{1}_{x \in V_{i,2}} \right)$$

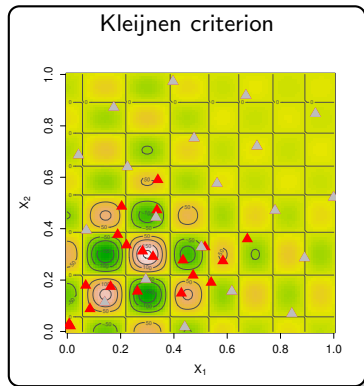
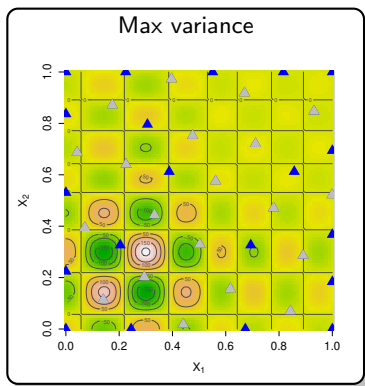
where $V_{i,j}$ is the Voronoi cell associated to $x_i^{(j)}$, $j = 1, 2$, $i = 1, \dots, n_j$.

Illustration of the new criterion



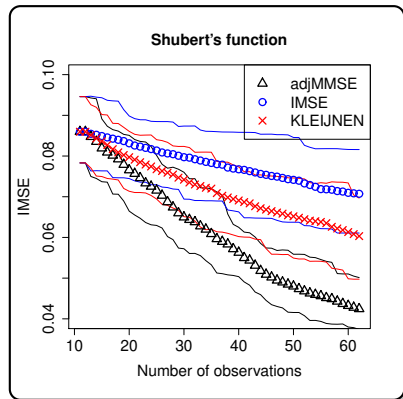
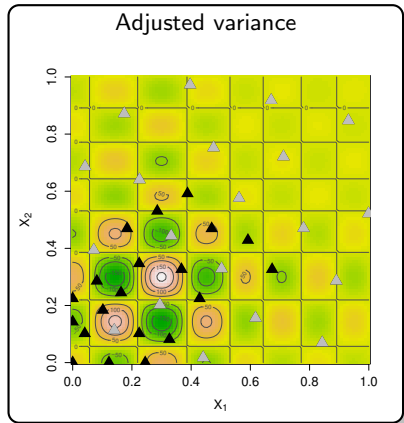
Academic example: the Schubert's function

$$g(x) = \left(\sum_{i=1}^5 i \cos \left((i+1)x^1 + i \right) \right) \left(\sum_{i=1}^5 i \cos \left((i+1)x^2 + i \right) \right)$$



Academic example: the Schubert's function

$$g(x) = \left(\sum_{i=1}^5 i \cos \left((i+1)x^1 + i \right) \right) \left(\sum_{i=1}^5 i \cos \left((i+1)x^2 + i \right) \right)$$



Application : spherical tank under internal pressure

- ▶ **High-fidelity code:** $g_2(x)$ is the von Mises stress at point 1, 2 or 3 provided by a finit elements code. $x = (P, R_{int}, T_{shell}, T_{cap}, E_{shell}, E_{cap}, \sigma_{y,shell}, \sigma_{y,cap})$ ($d = 8$)

P : internal pression.

R_{int} : tank internal radius.

T_{shell} : tank thickness.

T_{cap} : cap thickness.

E_{shell} : tank Young's modulus.

E_{cap} : cap Young's modulus.

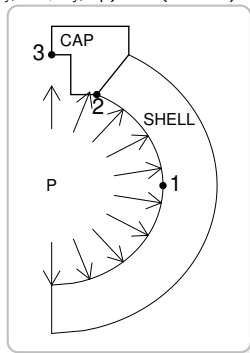
$\sigma_{y,shell}$: tank yield stress.

$\sigma_{y,cap}$: cap yield stress.

- ▶ **Low-fidelity code:**

g_1 is the 1D approximation of g_2 (perfectly spherical tank).

$$g_1(x) = \frac{3}{2} \frac{(R_{int} + T_{shell})^3}{(R_{int} + T_{shell})^3 - R_{int}^3} P$$



Application : spherical tank under internal pressure

- ▶ **High-fidelity code:** $g_2(x)$ is the von Mises stress at point 1, 2 or 3 provided by a finit elements code. $x = (P, R_{int}, T_{shell}, T_{cap}, E_{shell}, E_{cap}, \sigma_{y,shell}, \sigma_{y,cap})$ ($d = 8$)

P : internal pression.

R_{int} : tank internal radius.

T_{shell} : tank thickness.

T_{cap} : cap thickness.

E_{shell} : tank Young's modulus.

E_{cap} : cap Young's modulus.

$\sigma_{y,shell}$: tank yield stress.

$\sigma_{y,cap}$: cap yield stress.

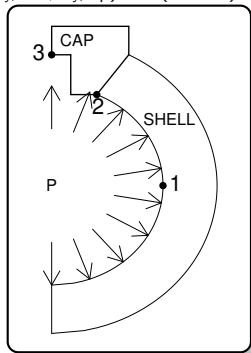
- ▶ **Low-fidelity code:**

g_1 is the 1D approximation of g_2 (perfectly spherical tank).

$$g_1(x) = \frac{3}{2} \frac{(R_{int} + T_{shell})^3}{(R_{int} + T_{shell})^3 - R_{int}^3} P$$

- ▶ **Co-kriging multi-fidelity model** built with $n_1 = 100$ and $n_2 = 20$.

The model efficiency Q_2 is estimated from a test set of 7000 points. $Q_2 \approx 86\%$.



Application : spherical tank under internal pressure

- ▶ **High-fidelity code:** $g_2(x)$ is the von Mises stress at point 1, 2 or 3 provided by a finit elements code. $x = (P, R_{int}, T_{shell}, T_{cap}, E_{shell}, E_{cap}, \sigma_{y,shell}, \sigma_{y,cap})$ ($d = 8$)

P : internal pression.

R_{int} : tank internal radius.

T_{shell} : tank thickness.

T_{cap} : cap thickness.

E_{shell} : tank Young's modulus.

E_{cap} : cap Young's modulus.

$\sigma_{y,shell}$: tank yield stress.

$\sigma_{y,cap}$: cap yield stress.

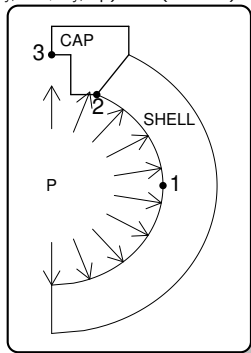
- ▶ **Low-fidelity code:**

g_1 is the 1D approximation of g_2 (perfectly spherical tank).

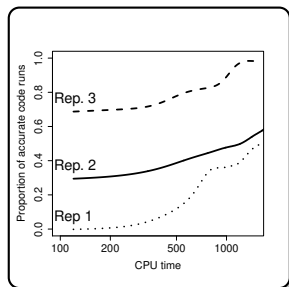
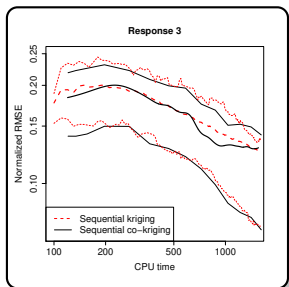
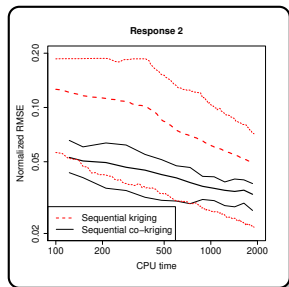
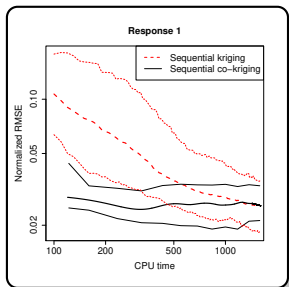
$$g_1(x) = \frac{3}{2} \frac{(R_{int} + T_{shell})^3}{(R_{int} + T_{shell})^3 - R_{int}^3} P$$

- ▶ **Co-kriging multi-fidelity model** built with $n_1 = 100$ and $n_2 = 20$.

The model efficiency Q_2 is estimated from a test set of 7000 points. $Q_2 \approx 86\%$.



Results





KENNEDY, M. C. & O'HAGAN, A. (2000) , Predicting the output from a complex computer code when fast approximations are available. *Biometrika* **87**, 1–13.



LE GRATIET, L. (2013) , Bayesian analysis of hierarchical multifidelity codes. *SIAM/ASA J. Uncertainty Quantification*: 1-1, pp. 244-269



LE GRATIET, L. & GARNIER, J. (2014), *Recursive co-kriging model for Design of Computer experiments with multiple levels of fidelity with an application to hydrodynamic*, *Int. J. Uncertainty Quantification*. DOI: 10.1615.



BATES, R.A., BUCK, R., RICCOMAGNO, E. & WYNN, H. (1996), Experimental design of computer experiments for large systems, *Journal of the Royal Statistical Society B*, *58* (1):77-94.



KLEIJNEN, J. & VAN BEERS, W. (2004), Application-driven sequential design for simulation experiments: Kriging metamodeling., *Journal of the Operational Research Society*, *55*:876-883.



LE GRATIET, L. & CANNAMELA CLAIRE (2014), Kriging-based sequential design strategies using fast cross-validation techniques with extensions to multi-fidelity computer codes, *to appear in TECHNOMETRICS*.

R CRAN package: [MuFiCokriging](#)