

Sequential designs for computer experiments

Pierre BARBILLON

AgroParisTech / INRA MIA UMR 518

CHORUS Workshop, 30/04/2014

Joint work with Guillaume Damblin (EDF)



Outline

- 1 Calibration context
 - Two kinds of data
 - Bayesian Calibration
 - Meta-modeling / emulator of the computer code
 - Calibration with emulator
- 2 Expected Improvement
 - Efficient Global Optimization
 - Calibration
- 3 Conclusion

Plan

1 Calibration context

■ Two kinds of data

- Bayesian Calibration
- Meta-modeling / emulator of the computer code
- Calibration with emulator

2 Expected Improvement

- Efficient Global Optimization
- Calibration

3 Conclusion

Field data

- Field data provided by physical experiments:

$$\mathbf{y}^F = y^F(\mathbf{x}_1), \dots, y^F(\mathbf{x}_n),$$

- n is small, $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{X}$ hard to set, sometimes uncontrollable, included in a small domain...

- Model:

$$y^F(\mathbf{x}_i) = \zeta(\mathbf{x}_i) + \epsilon(\mathbf{x}_i),$$

where

- $\zeta(\cdot)$ real physical process (unknown),
- $\epsilon(\mathbf{x}_i)$ often assumed i.i.d. $\mathcal{N}(0, \sigma^2)$,
- σ^2 sometimes treated as known...

Computer model / simulator

Computer experiments:

Computer model (simulator) $(\mathbf{x}^*, \theta) \mapsto f(\mathbf{x}^*, \theta) \in \mathbb{R}^s$ where

- **physical parameters:** $\mathbf{x}^* \in \mathbb{X} \subset \mathbb{R}^m$ observable and often controllable inputs
 - \mathbf{x}^* same meaning as in field data,
 - extrapolation if $\mathbf{x}^* > \max(\mathbf{x}_i)$ or $\mathbf{x}^* < \min(\mathbf{x}_i)$.
- **simulator parameters:** $\theta \in \Theta \subset \mathbb{R}^d$ non-observable parameters, required to run the simulator.
2 types:
 - “calibration parameters”: physical meaning but unknown, necessary to make the code mimic the reality,
 - “tuning parameters”: no physical interpretation.

f designed to mimic the unknown physical process $\zeta(\cdot)$ for a value of θ .

The simulator is often an **expensive black-box function**.

\Rightarrow limited number N_{run} of runs of the simulator.

Relationship between the simulator and the data

for $i = 1, \dots, n$,

- if the simulator sufficiently represents the physical system:

$$y_i^F = f(\mathbf{x}_i, \theta^*) + \epsilon(\mathbf{x}_i),$$

i.e. for the unknown value $\theta = \theta^* : f(\mathbf{x}, \theta^*) = \zeta(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{X}$,

- if the field observations are inconsistent with the simulations (irreducible model discrepancy):

$$y_i^F = f(\mathbf{x}_i, \theta^*) + \delta(\mathbf{x}_i) + \epsilon(\mathbf{x}_i).$$

$\delta(\cdot)$ models the difference between the simulator and the physical system:

$$\delta(\mathbf{x}) = \zeta(\mathbf{x}) - f(\mathbf{x}, \theta^*),$$

but

- What does θ^* mean ?
- A best fitting ?
- identifiability issues ?
- usually assumed to be smoother than the real physical process $\zeta(\cdot)$

Ref.: [Kennedy and O'Hagan \(2001\)](#), [Hidgon et al. \(2005\)](#)...

Plan

1 Calibration context

- Two kinds of data
- **Bayesian Calibration**
- Meta-modeling / emulator of the computer code
- Calibration with emulator

2 Expected Improvement

- Efficient Global Optimization
- Calibration

3 Conclusion

A calibration example

Hypotheses:

- The simulator represents sufficiently well the physical system:

$$y^F(\mathbf{x}_i) = f(\mathbf{x}_i, \boldsymbol{\theta}^*) + \epsilon_i, \quad i = 1, \dots, n.$$

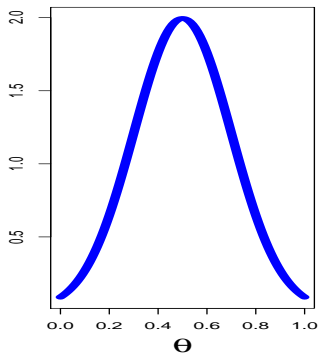
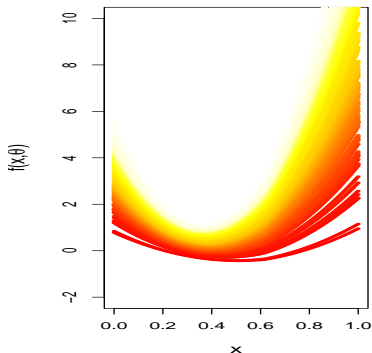
- But unknown $\boldsymbol{\theta}^*$.
- $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ i.i.d. with known σ^2 .
- $\sigma^2 = 0.3$
- $n = 6$,
- $\theta^* = 0.6$

A calibration example

Prior:

prior distribution on unknown θ : $\pi(\cdot)$
from expert judgment, past experiments...

Possible choice $\pi(\theta) = \mathcal{N}(\theta_0, \sigma_0^2) = \mathcal{N}(0.5, 0.04)$.



A calibration example

Data:

Couples $(\mathbf{x}_1, y_1^F), \dots, (\mathbf{x}_n, y_n^F)$ from physical experiments.

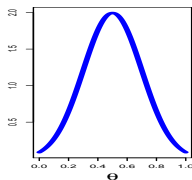
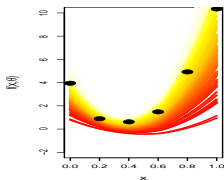
Posterior distribution:

$$\begin{aligned}\pi(\boldsymbol{\theta} | \mathbf{y}^F) &\propto l(\boldsymbol{\theta} | \mathbf{y}^F) \cdot \pi(\boldsymbol{\theta}) \\ &\propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i^F(\mathbf{x}_i) - f(\mathbf{x}_i, \boldsymbol{\theta}))^2 - \frac{1}{2\sigma_0^2} (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^2\right)\end{aligned}$$

- Analytical posterior if $\boldsymbol{\theta} \mapsto f(\mathbf{x}, \boldsymbol{\theta})$ is a linear map,
- Otherwise MH sampling to simulate according to the posterior distribution.

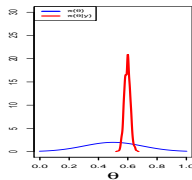
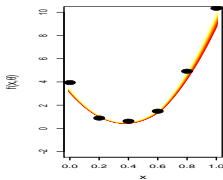
A calibration example

Prior with data:



↓ Metropolis-Hastings algorithm ↓

Posterior on θ :



More details on the MH algorithm

Initialisation:

θ^0 chosen.

Update:

iterations $t = 1, \dots,$

1 Proposal: $\tilde{\theta}^{t+1} = \theta^t + \mathcal{N}(0, \tau^2)$.

2 Compute

$$\alpha(\theta^t, \tilde{\theta}^{t+1}) = \frac{\pi(\tilde{\theta}^{t+1} | \mathbf{y}^F)}{\pi(\theta^t | \mathbf{y}^F)}$$

3 Acceptation:

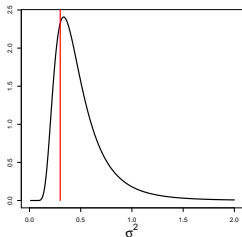
$$\theta^{t+1} = \begin{cases} \tilde{\theta}^{t+1} & \text{with probability } \alpha(\theta^t, \tilde{\theta}^{t+1}) \\ \theta^t & \text{otherwise.} \end{cases}$$

Note that the ratio $\alpha(\theta^t, \tilde{\theta}^{t+1})$ needs several computations of $f(\mathbf{x}, \theta)$ at each step since

$$\pi(\theta | \mathbf{y}^F) \propto \exp \left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y^F(\mathbf{x}_i) - f(\mathbf{x}_i, \theta))^2 - \frac{1}{2\sigma_0^2} (\theta - \theta_0)^2 \right).$$

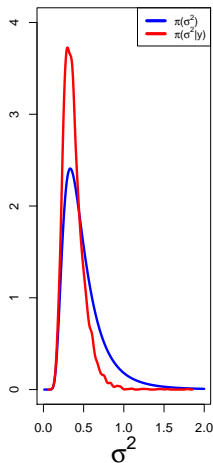
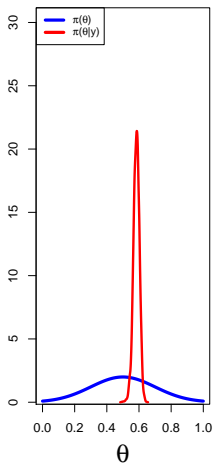
Unknown σ^2

- prior distribution on σ^2 : $\pi(\sigma^2) = \mathcal{IG}(5, 2)$



- Gibbs algorithm to simulate couples (θ, σ^2) from $\pi(\theta, \sigma^2 | \mathbf{y}^F)$. Iterate :
 - 1 MH algorithm to simulate θ_t from $\pi(\cdot | \mathbf{y}^F, \sigma_{t-1}^2)$,
 - 2 conditional simulation of σ_t^2 from $\pi(\cdot | \mathbf{y}^F, \theta_t)$.

Posterior distributions



Comparison

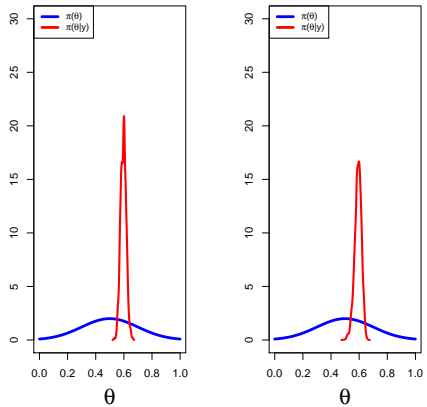


Figure : known σ^2 vs unknown σ^2

with a bad prior....

prior on θ : $\pi(\theta) = \mathcal{N}(0.2, 0.04)$ and $n = 12$ field data

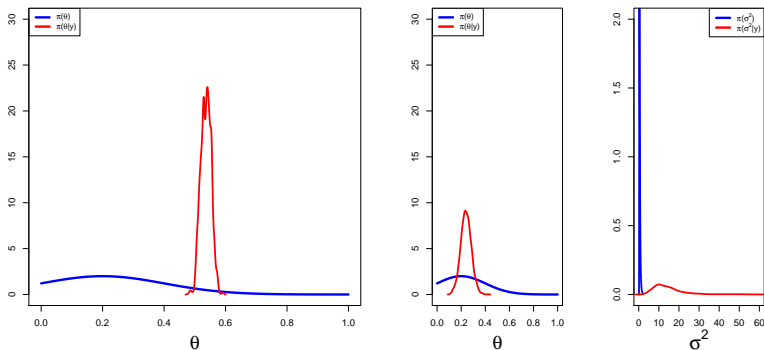


Figure : known σ^2 vs unknown σ^2

Plan

1 Calibration context

- Two kinds of data
- Bayesian Calibration
- **Meta-modeling / emulator of the computer code**
- Calibration with emulator

2 Expected Improvement

- Efficient Global Optimization
- Calibration

3 Conclusion

Expensive black-box computer code

- Run the simulator for a given (\mathbf{x}^*, θ) is time-consuming / expensive.
- The simulator is a black-box, no intrusive methods are possible.

⇒ Only few runs of the simulator are possible then we cannot apply algorithms (as in Bayesian calibration) which make a massive use of simulator runs.

Using an emulator / metamodel / coarse model / approximation of the simulator which is fast to compute, but:

- loss on precision of prediction,
- new uncertainty source: accuracy of the model approximation,
- taken into account.

Choosing a design of experiments

Choose N_{run} couples

$$(\mathbf{x}_j^*, \theta_j)$$

- space filling for x ,
- with respect to the prior distribution on θ ,
- $\mathbf{x}_j^* = \mathbf{x}_i$?

where the simulator is called.

Emulator using Gaussian Process:

- Very popular in computer experiments.
- integrated in a Bayesian framework: appears in the likelihood function and a prior on the parameters of the Gaussian process are chosen.
- model uncertainty coming from approximation of f .
- After the calibration step, used in prediction for a new point \mathbf{x} .

Meta-modeling: prior distribution on f

Sacks et al. (1989).

f realization of a Gaussian process F :

$\forall(\mathbf{x}^*, \boldsymbol{\theta}) \in E$,

$$F((\mathbf{x}^*, \boldsymbol{\theta})) = \sum_{k=1}^Q \beta_k h_k((\mathbf{x}^*, \boldsymbol{\theta})) + Z((\mathbf{x}^*, \boldsymbol{\theta})) = H((\mathbf{x}^*, \boldsymbol{\theta}))^T \boldsymbol{\beta} + Z((\mathbf{x}^*, \boldsymbol{\theta})),$$

où

- h_1, \dots, h_Q regression functions and $\boldsymbol{\beta}$ parameters vector,
- Z centered Gaussians process with covariance function:

$$\text{Cov}(Z((\mathbf{x}_1^*, \boldsymbol{\theta}_1)), Z((\mathbf{x}_2^*, \boldsymbol{\theta}_2))) = \sigma^2 K((\mathbf{x}_1^*, \boldsymbol{\theta}_1), (\mathbf{x}_2^*, \boldsymbol{\theta}_2)),$$

where K is correlation kernel.

Hypotheses

- $K((\mathbf{x}_1^*, \boldsymbol{\theta}_1), (\mathbf{x}_2^*, \boldsymbol{\theta}_2)) = \sigma_K^2 \exp(-\xi_{\mathbf{x}^*} \sum |\mathbf{x}_1^* - \mathbf{x}_2^*|^{\alpha} - \xi_{\boldsymbol{\theta}} \sum |\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2|^{\alpha})$
- parameters $\phi = (\boldsymbol{\beta}, \sigma^2, K \text{ parameters})$ assumed fixed (in practice, maximum likelihood estimators);

Meta-modeling: posterior

- $v_1 = f((\mathbf{x}^*, \theta)_1), \dots, v_{N_{run}} = f((\mathbf{x}^*, \theta)_{N_{run}})$ evaluations of f on a design $D_{N_{run}}$
- Process $F^{D_{N_{run}}}$: Conditioning F to $F((\mathbf{x}_1^*, \theta_1)) = v_1, \dots, F(\mathbf{x}_{N_{run}}^*, \theta_{N_{run}}) = v_{N_{run}}$.
 Gaussian Process with mean $m((\mathbf{x}^*, \theta))$ and covariance $C((\mathbf{x}^*, \theta), (\mathbf{x}^*, \theta)') \forall (\mathbf{x}^*, \theta), (\mathbf{x}^*, \theta)'$.

For all $(\mathbf{x}^*, \theta) \in E$,

- $m((\mathbf{x}^*, \theta))$ approximates $f((\mathbf{x}^*, \theta))$,
- $C((\mathbf{x}^*, \theta), (\mathbf{x}^*, \theta))$ uncertainty on this approximation.

For all $(\mathbf{x}_i^*, \theta_i) \in D_{N_{run}}$,

- $m(\mathbf{x}_i^*, \theta_i) = f(\mathbf{x}_i^*, \theta_i)$,
- $C((\mathbf{x}_i^*, \theta_i), (\mathbf{x}_i^*, \theta_i)) = 0$.

Gaussian process emulator: illustration

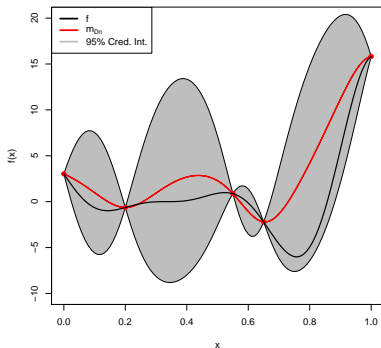


Figure : Posterior mean and pointwise credible interval

Plan

1 Calibration context

- Two kinds of data
- Bayesian Calibration
- Meta-modeling / emulator of the computer code
- Calibration with emulator

2 Expected Improvement

- Efficient Global Optimization
- Calibration

3 Conclusion

Likelihood with a Gaussian process hypothesis on f

- $\mathbf{z} = (\mathbf{y}_1^F, \dots, \mathbf{y}_n^F, f(\mathbf{x}_1^*, \boldsymbol{\theta}_1), \dots, f(\mathbf{x}_{N_{run}}^*, \boldsymbol{\theta}_{N_{run}}))$
- likelihood on \mathbf{z}

$$l(\boldsymbol{\theta}, \sigma^2 | \mathbf{z}) \propto |\Sigma_{\mathbf{z}}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^T \Sigma_{\mathbf{z}}^{-1}(\mathbf{z} - \boldsymbol{\mu})\right)$$

where

- $\boldsymbol{\mu}$ is the mean of the Gaussian process,
-

$$\Sigma_{\mathbf{z}} = \Sigma_f + \begin{pmatrix} \Sigma_y & 0 \\ 0 & 0 \end{pmatrix}$$

with $\Sigma_y = \sigma^2 I_n$ and Σ_f is obtained as the covariance matrix corresponding to the points: $(\mathbf{x}_1, \boldsymbol{\theta}), \dots, (\mathbf{x}_n, \boldsymbol{\theta}), (\mathbf{x}_1^*, \boldsymbol{\theta}_1), \dots, (\mathbf{x}_{N_{run}}^*, \boldsymbol{\theta}_{N_{run}})$.

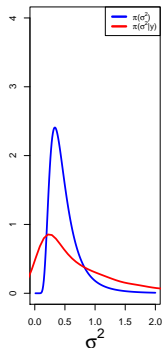
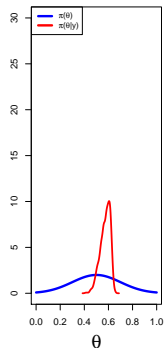
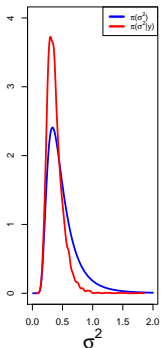
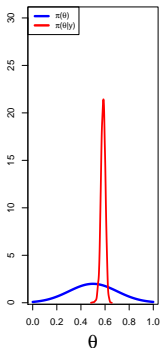
Dealing with GP parameters

- prior distribution on μ and covariance parameters [Hidgon et al. \(2005\)](#)
 ⇒ MCMC inference
- MLE estimators [Kennedy and O'Hagan \(2001\)](#)
 - treated as fixed,
 - only computer data $f(\mathbf{x}_1^*, \theta_1), \dots, f(\mathbf{x}_{N_{run}}^*, \theta_{N_{run}})$ are used ($n < N_{run}$) for MLE
 - likelihood $l(\theta, \sigma^2 | \mathbf{z})$:

$$l(\theta, \sigma^2 | \mathbf{z}) \propto |\tilde{\Sigma}_{\mathbf{y}^F}|^{-1/2} \exp \left(-\frac{1}{2} (\mathbf{y}^F - m(\mathbf{x}, \theta))^T \tilde{\Sigma}_{\mathbf{y}^F}^{-1} (\mathbf{y}^F - m(\mathbf{x}, \theta)) \right)$$

where

- $m(\cdot)$ is the mean of the GP conditioned to simulator data,
- $\tilde{\Sigma}_{\mathbf{y}^F} = \Sigma_{\mathbf{y}^F} + \tilde{\Sigma}_f = \sigma^2 I_n + \tilde{\Sigma}_f$ where $\tilde{\Sigma}_f$ is constructed with the covariance function C of the conditioned GP.



unlimited runs versus $N_{run} = 12$

Outline

- 1 Calibration context
 - Two kinds of data
 - Bayesian Calibration
 - Meta-modeling / emulator of the computer code
 - Calibration with emulator
- 2 **Expected Improvement**
 - **Efficient Global Optimization**
 - **Calibration**
- 3 Conclusion

Principle

- 1 Construct a first exploratory design: D_n s. t. $n \leq N$,
- 2 For $i = n + 1 \dots N$ do $D_i = D_{i-1} \cup \{\mathbf{x}_i\}$ where

$$\mathbf{x}_i \in \arg \max \text{Crit}(D_{i-1}, f).$$

$\text{Crit}(D_{i-1}, f)$ can be adapted to the applied goal (optimization, estimation of probability of rare event).

Plan

- 1 Calibration context
 - Two kinds of data
 - Bayesian Calibration
 - Meta-modeling / emulator of the computer code
 - Calibration with emulator
- 2 Expected Improvement
 - Efficient Global Optimization
 - Calibration
- 3 Conclusion

Expected Improvement criterion

- Goal: Find the global extremum (here minimum e.g.) of f ,
- Expected improvement criterion proposed by [Jones et al. \(1998\)](#):

$$EI_n(\mathbf{x}) = \mathbb{E}((\min_n - F(\mathbf{x}))^+ | F(D_n)),$$

where \min_n is the current minimum value:

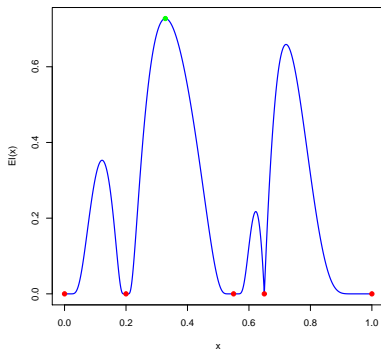
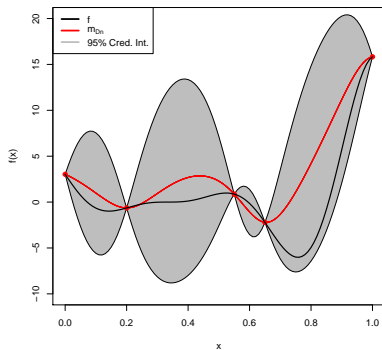
$$\min_n = \min_{1, \dots, n} f(\mathbf{x}_j)$$

- Closed-form computation:

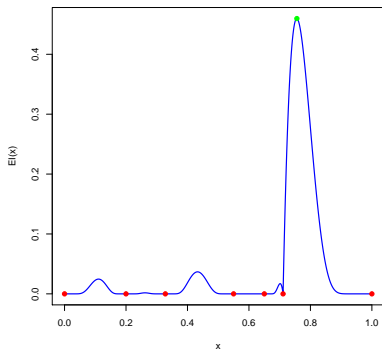
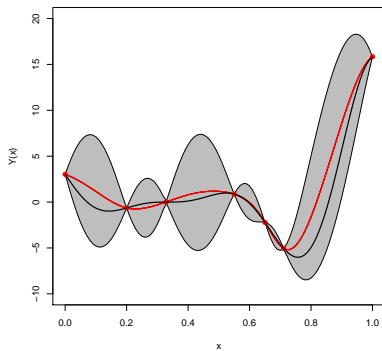
$$EI_n(\mathbf{x}) = (\min_n - m_{D_n}(\mathbf{x}))\Phi\left(\frac{\min_n - m_{D_n}(\mathbf{x})}{\sqrt{C_{D_n}(\mathbf{x}, \mathbf{x})}}\right) + \sqrt{C_{D_n}(\mathbf{x}, \mathbf{x})}\phi\left(\frac{\min_n - m_{D_n}(\mathbf{x})}{\sqrt{C_{D_n}(\mathbf{x}, \mathbf{x})}}\right)$$

where Φ and ϕ are respectively the cdf and the pdf of $\mathcal{N}(0, 1)$.

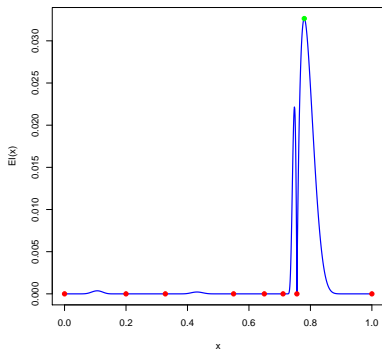
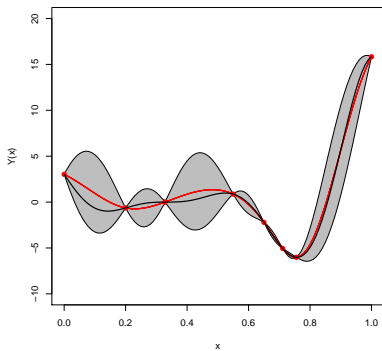
Example step 1



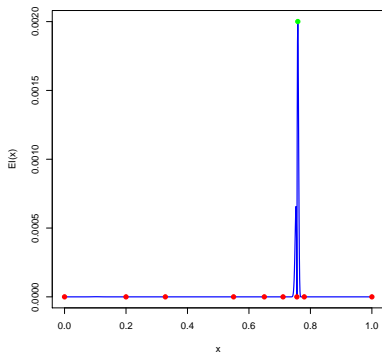
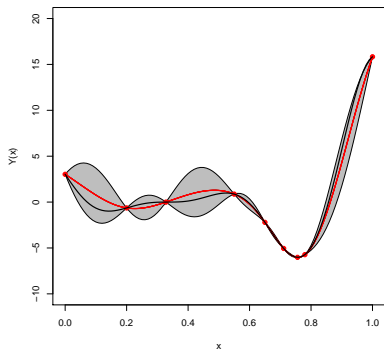
Example step 2



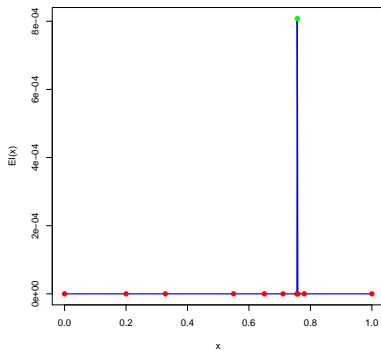
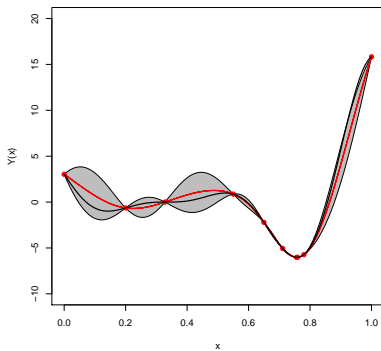
Example step 3



Example step 4



Example step 5

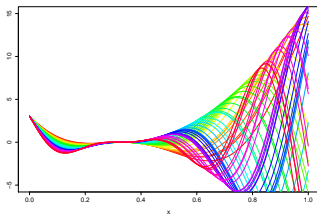


Plan

- 1 Calibration context
 - Two kinds of data
 - Bayesian Calibration
 - Meta-modeling / emulator of the computer code
 - Calibration with emulator
- 2 Expected Improvement
 - Efficient Global Optimization
 - Calibration
- 3 Conclusion

Example:

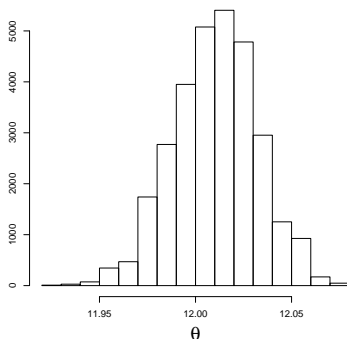
- $\theta = 12$,
- $(x_1, x_2, x_3) = (0.1, 0.3, 0.8)$,
- $f(x, \theta) = (6 \cdot x - 2)^2 \cdot \sin(\theta \cdot x - 4) + \epsilon$,
- $\epsilon_i \sim \mathcal{N}(0, 0.1^2)$ i.i.d.,
- prior $\theta \sim \mathcal{U}[5, 15]$,
- $y_i = f(x_i, \theta) + \epsilon_i$.



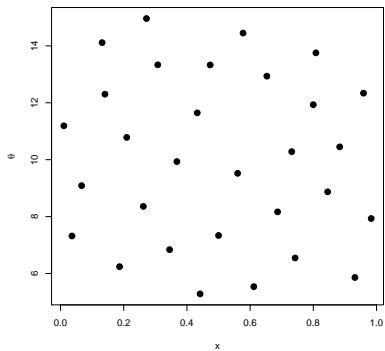
Motivation for adaptive designs in calibration

Quality of calibration (Bayesian or ML) is affected by choice in the numerical design.

- Calibration with unlimited runs of f

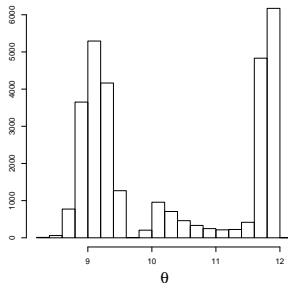
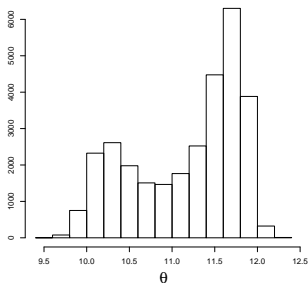


LHS maximin design



Motivation for adaptive designs in calibration

- Calibration with emulator built from a design with $N = 30$ calls to f



Likelihood for calibration

$l(\theta|\mathbf{z})$:

$$l(\theta|\mathbf{z}) \propto |\tilde{\Sigma}_{\mathbf{y}^F}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y}^F - m(\mathbf{x}, \theta))^T \tilde{\Sigma}_{\mathbf{y}^F}^{-1}(\mathbf{y}^F - m(\mathbf{x}, \theta))\right)$$

where

- \mathbf{y}^F is the vector of field data,
- $m(\cdot)$ is the mean of the GP conditioned to simulator data,
- $\tilde{\Sigma}_{\mathbf{y}^F} = \Sigma_{\mathbf{y}^F} + \tilde{\Sigma}_f = \sigma^2 I_n + \tilde{\Sigma}_f$ where $\tilde{\Sigma}_f$ is constructed with the covariance function C of the conditioned GP.

Optimization goal : maximize the likelihood \Rightarrow Expected Improvement for calibration.

EI

Maximize the likelihood $l(\theta|\mathbf{z})$ over $\theta \Leftrightarrow$ Minimize $MC(\theta) = \|\mathbf{y}^F - f(\mathbf{x}, \theta)\|^2$ over θ .

For given:

- field experiments $\mathbf{y}^F = y^F(\mathbf{x}_1), \dots, y^F(\mathbf{x}_n)$,
- D_k numerical design on $\mathbb{X} \times \Theta$ with M points,
- m_0 current minimal value of $MC(\theta)$.

EI criterion:

$$El_{D_k}(\theta) = \mathbb{E}_{D_k} ((m_0 - MC(\theta))^+),$$

to be minimised.

EI criterion is applied to a function of f .

EI computation

$$\begin{aligned}EI_{D_k}(\boldsymbol{\theta}) &= \int_{B(0, \sqrt{m_0})} (m_0 - MC(\boldsymbol{\theta})) dF_{D_M} \\ &= m_0 \cdot \mathbb{P}_{D_M}(MC(\boldsymbol{\theta}) \leq m_0) - \mathbb{E}_{D_M}(MC(\boldsymbol{\theta}) \mathbb{I}_{MC(\boldsymbol{\theta}) \leq m_0})\end{aligned}$$

- no close form computation,
- $\mathbb{P}_{D_M}(MC(\boldsymbol{\theta}) \leq m_0)$ is an upper bound and easier to compute,
- importance sampling may be used for the second term.

Algorithm

- 1 Build a first space-filling design D_0 on $\mathbb{X} \times \Theta$,
- 2 Find the maximum: $\tilde{\theta}_0$ of $l(\theta|\mathbf{z})$,
- 3 Evaluate $f(\mathbf{x}_1, \tilde{\theta}_0), \dots, f(\mathbf{x}_n, \tilde{\theta}_0)$.
- 4 Set $m_0 = MC(\tilde{\theta}_0)$,
- 5 for $k=1 \dots$, repeat
 - 1 Compute El_{D_k} on a grid on Θ ,
 - 2 $\tilde{\theta}_k = \arg \max_{\Theta} El_{D_k}(\theta)$,
 - 3 Evaluate $f(\mathbf{x}_1, \tilde{\theta}_k), \dots, f(\mathbf{x}_n, \tilde{\theta}_k)$

Bayesian calibration based on the adapted design

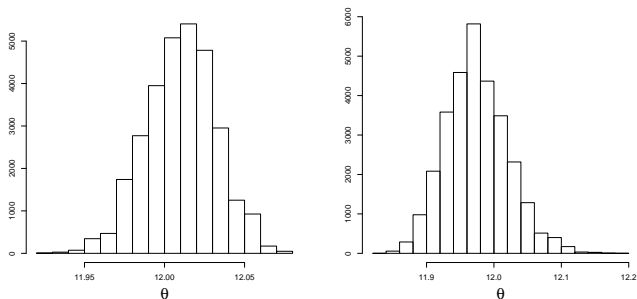


Figure : Bayesian calibration with unlimited runs vs Bayesian calibration with $N = 30$ chosen by EGO

Outline

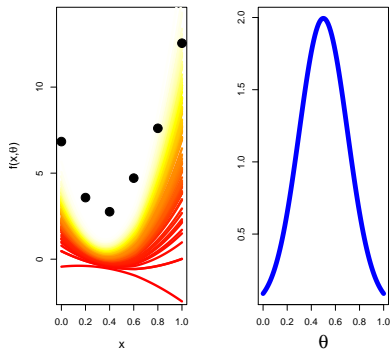
- 1 Calibration context
 - Two kinds of data
 - Bayesian Calibration
 - Meta-modeling / emulator of the computer code
 - Calibration with emulator
- 2 Expected Improvement
 - Efficient Global Optimization
 - Calibration
- 3 Conclusion

Conclusion

- Designs of numerical experiments adapted to calibration purpose,
- Robustness in calibration.
- Higher dimension questions, number of field experiments, dimension of θ ...
- New field experiments ?
- discrepancy issues ?

Model discrepancy

$$y_i^F = f(\mathbf{x}_i, \boldsymbol{\theta}^*) + \delta(\mathbf{x}_i) + \epsilon(\mathbf{x}_i).$$



No value of θ makes the simulator corresponding to the field data

References

Calibration of computer models

- Dave Hidgon et al., 2005. Combining Field Data and Computer Simulations for Calibration and Prediction. *SIAM* 26(2).
- Marc Kennedy and Anthony O'Hagan, 2001. Bayesian Calibration of Computer Models. *Journal of the Royal Statistical Society B* 68.
- Jenny Brynjarsdóttir and Anthony O'Hagan, 2013. *J. of Uncertainty Quantification*.

Optimization

- Donald Jones et al., 1998. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* 13(4).

Gaussian Process emulator:

- Thomas Santner et al., 2003. *The Design and Analysis of Computer Experiments*. Springer-Verlag.
- Kai-Tai Fang et al., 2006. *Design and Modeling for Computer Experiments*. Computer Science and Data Analysis. Chapman & Hall/CRC.

R

- packages Dice....