Introduction to Geostatistics - Metamodeling with Gaussian processes

Fabrice Gamboa
Bertrand Iooss

11/10/2015

Uncertainties management for cpu time consuming models

***p* input variables**

$X = (X_1, \ldots, X_p)$

Physical phenomena → **observed experiences**

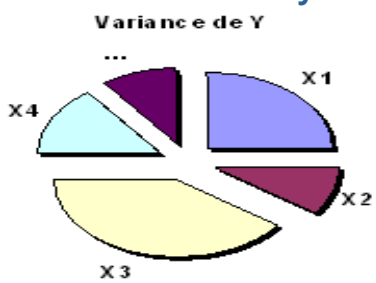
Computer code → $y(\mathbf{X})$ Time consuming → **simulated experiences**

Metamodel → $\hat{Y}(\mathbf{X})$ Negligible cost → **Predicted experiences**

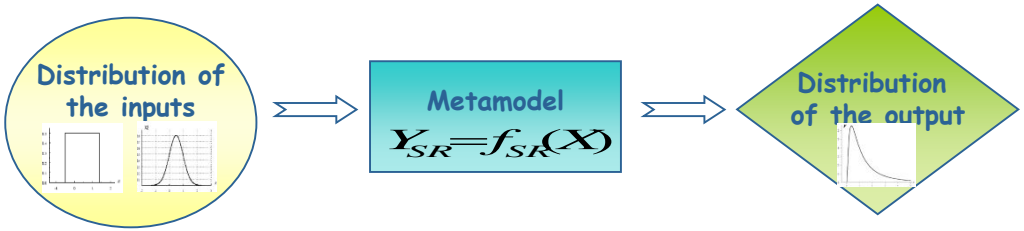
≈ ≈

Use of the metamodel :

- C':Sensitivity analysis



- C:Uncertainty propagation (via Monte Carlo methods)



- B':Calibration

Identification of input parameters values

↓

Adequation between observed and simulated experiences

Metamodel : definition

[Kleijnen 70's]

A metamodel is a mathematical function

- **which approximates the outputs of the model,**
- **with neglgible cpu cost,**
- **which allows to make new output predictions with a good accuracy**

- <u>Synonyms:</u>

  - Response surface
  - Simplified model
  - Emulator
  - Proxy model
  - Surrogate model

Metamodeling steps

Design of experiments:

Points to perform simulations

→

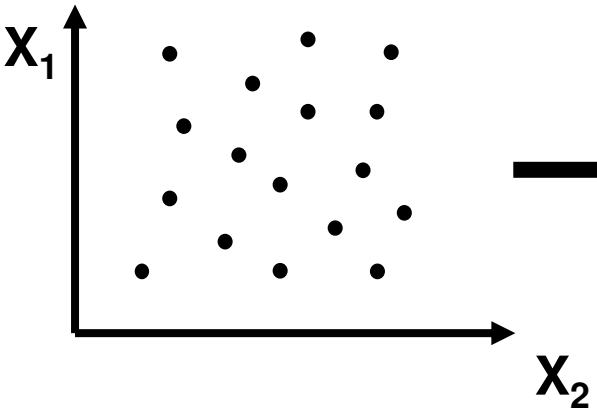
Simulation :

Performing simulations

→

Metamodeling :

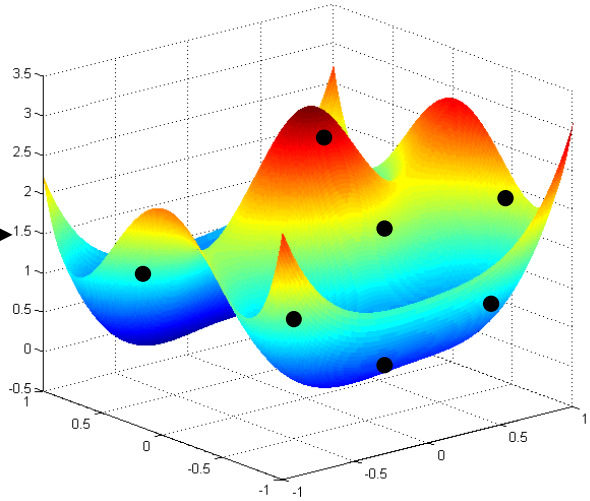
Approximation of the computer code



→

Computer code

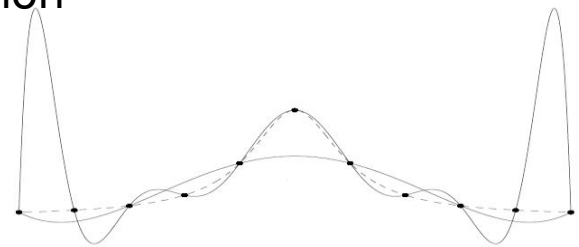
→

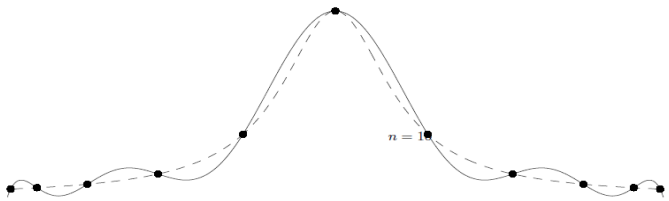


Differents types of metamodels

[Simpson et al. 2001]
[Storlie & Helton 2008]

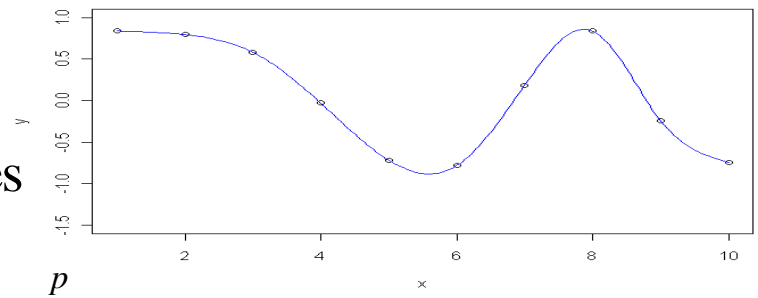
- Linear regression
- Polynomials





- Splines

$$\hat{G}(\mathbf{x}) = \sum_{k=1}^{K} \hat{\beta}_k B_k(\mathbf{x}) \text{ with } K \text{ the number of nodes}$$



- Additive models, GAM $\hat{G}(\mathbf{x}) = \sum_{i=1}^{p} s_i(x_i) + \sum_{i<j}^{p} s_{ij}(x_i, x_j) + \cdots$
- Regression trees $\hat{G}(\mathbf{x}) = \sum_{k=1}^{K} \hat{\beta}_k I_k(\mathbf{x})$
- Neural networks
- Chaos polynomials
- Support Vector Machines
- Kriging – Gaussian process

Kriging metamodel

Kriging *[Matheron 63]* for computer codes relies on the idea to interpolate the code outputs in dimension *p* *[Sacks et al. 89]* as a spatial cartography

Kriging (or Gaussian process) is interesting because:

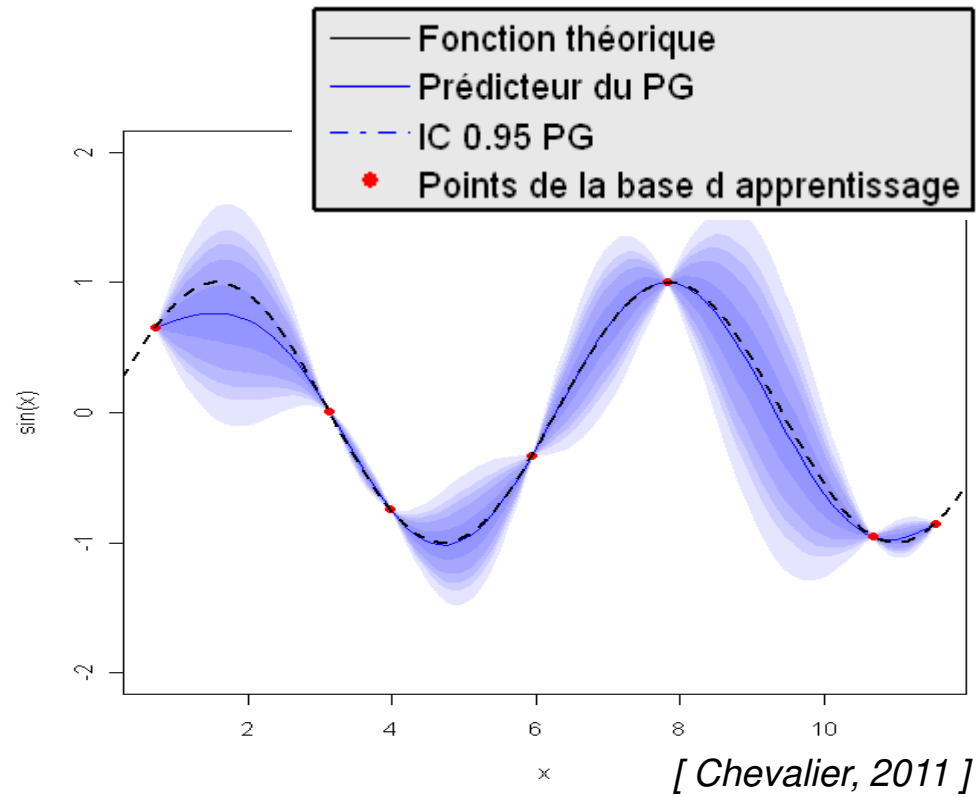
- **it interpolates the outputs,**
- **it gives predictor associated with confidence bands**

<u>Example in 1D :</u>

Theoretical function (p =1) :

$$Y = f(X) = \sin(X)$$

Simulation of N = 7 computation points



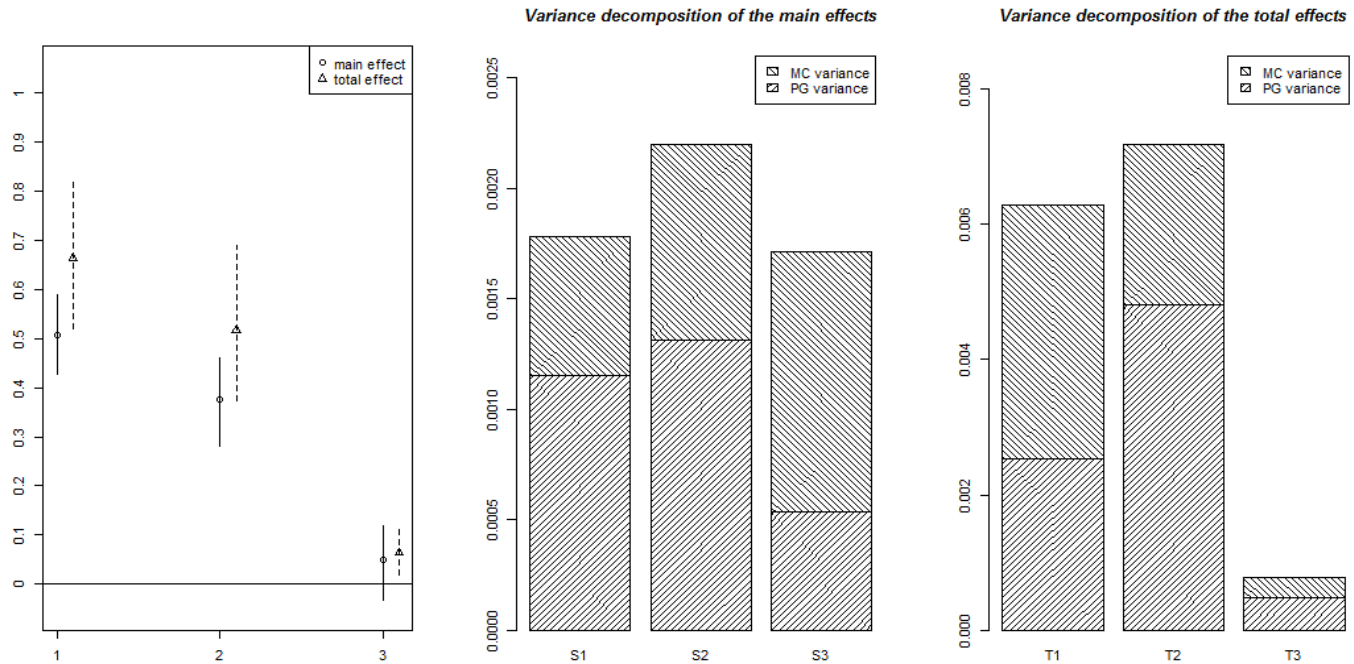
[Chevalier, 2011]

Example: Using metamodels to estimate Sobol' indices

$$\text{Main effects}: S_i = \frac{\mathrm{Var}[\mathrm{E}(Y|X_i)]}{\mathrm{Var}(Y)}; \text{Total effects}: S_{Ti} = 1 - \frac{\mathrm{Var}[\mathrm{E}(Y|X_{-i})]}{\mathrm{Var}(Y)}$$

<u>Many methods to estimate Sobol' indices:</u> Sampling (Monte Carlo, quasi-MC, spectral approaches), smoothing methods, metamodels, …

To reduce the cost (number of model evaluations), the kriging metamodel is efficient and allows to propagate the metamodel error on S_i and S_{Ti} estimates



Kriging in spatial statistics (Geostatistics)

Introduction to Geostatistics

Objectives : treatment of **numerical data** with **spatial support** (or temporal) with **uncertainty quantification**

Principal aspects:

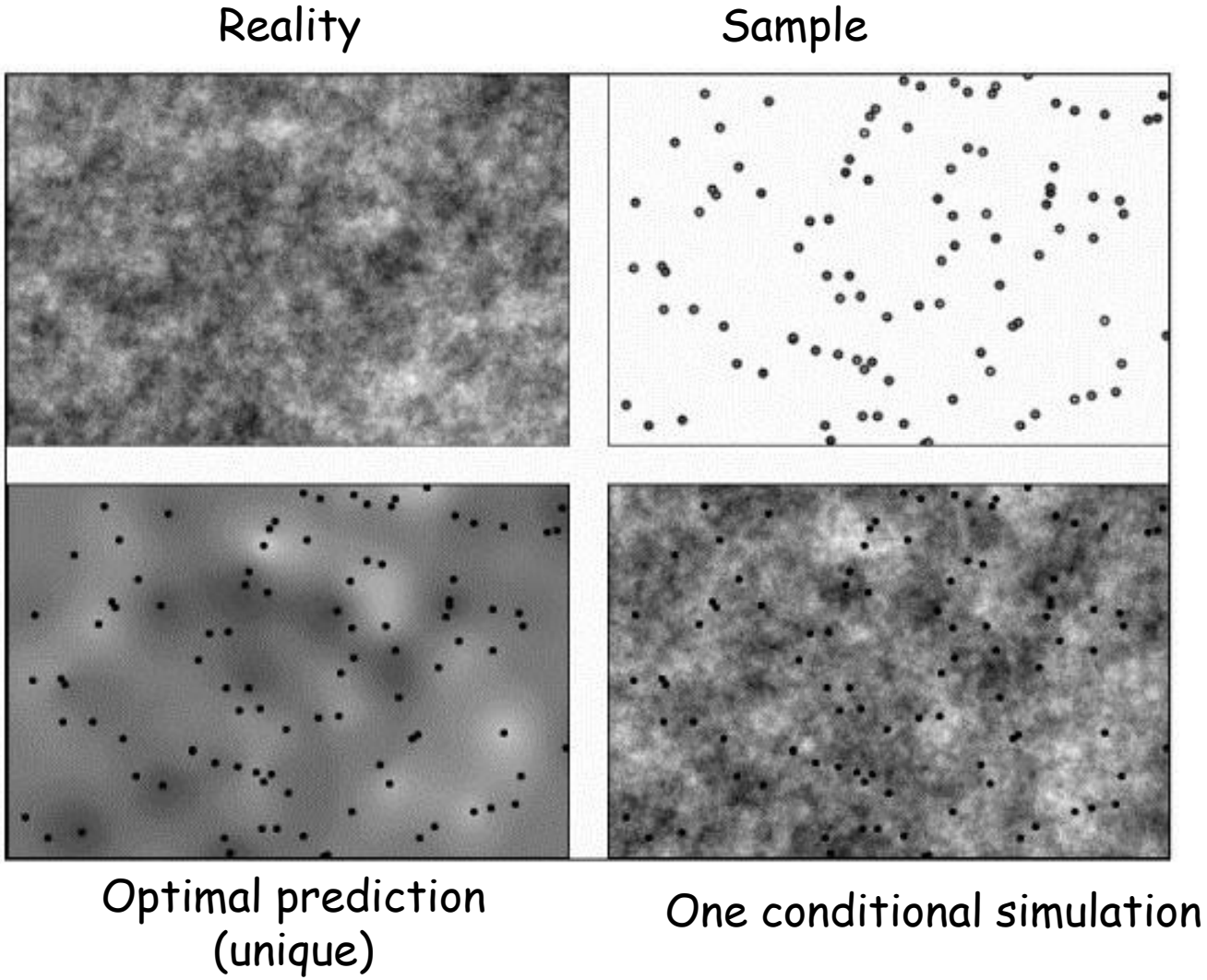
- Taking into account the spatial structure of data,
- Dimension 1, 2, 3, …,
- Irregular sampling,
- Integrating external information

2 types of methods:

- **Estimation** (prediction, …) at a given point

- **Simulations** reproducing the variability of the phenomenon

Example : porosity of a geological medium



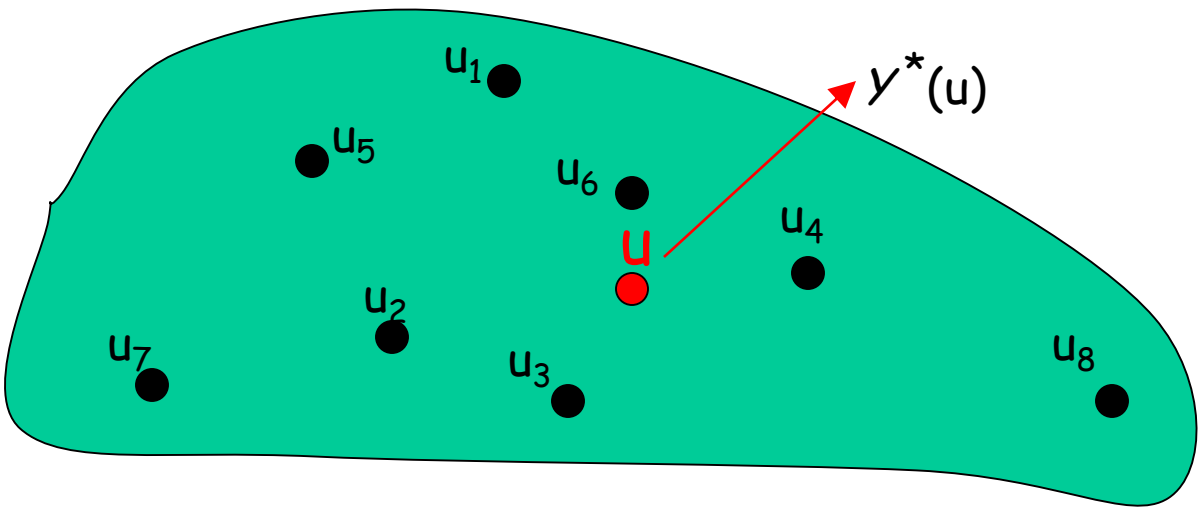
[Chilès]

Spatial statistics: kriging interpolation

Linear combination of N data:

$$Y^*(u) = \sum_{i=1}^{N} \lambda_i Y(u_i)$$

Kriging can take into account the data configuration, the distance between data and target, the spatial correlations and potential external information



Probabilistic framework

Estimation without bias: $E[\, Y^*(u) - Y(u) \,] = 0$

the mean of the errors is zero

Estimation $Y^*(u)$ is optimal: $Var[\, Y^*(u) - Y(u) \,]$ is minimal

the dispersion of the errors is reduced

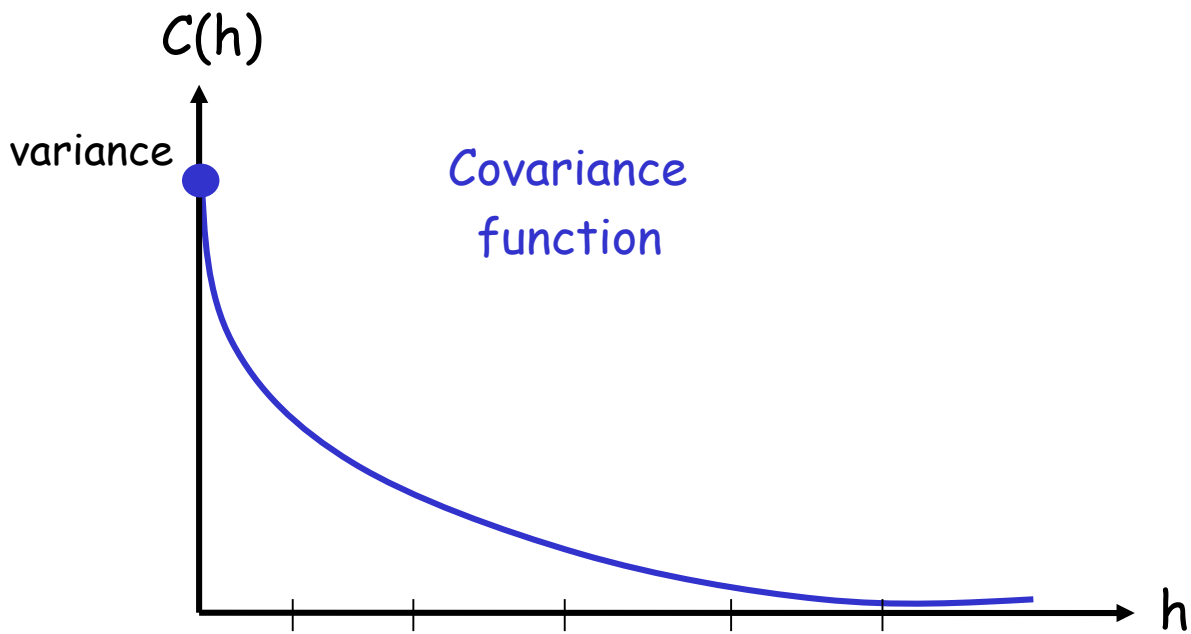
Stochastic model for Y(x)

The random field $Y(x)$, with $Y \in \Re$ and $x \in \Re^p$, is characterized by its mean and its covariance

$Y(x)$ is stationary of second order:

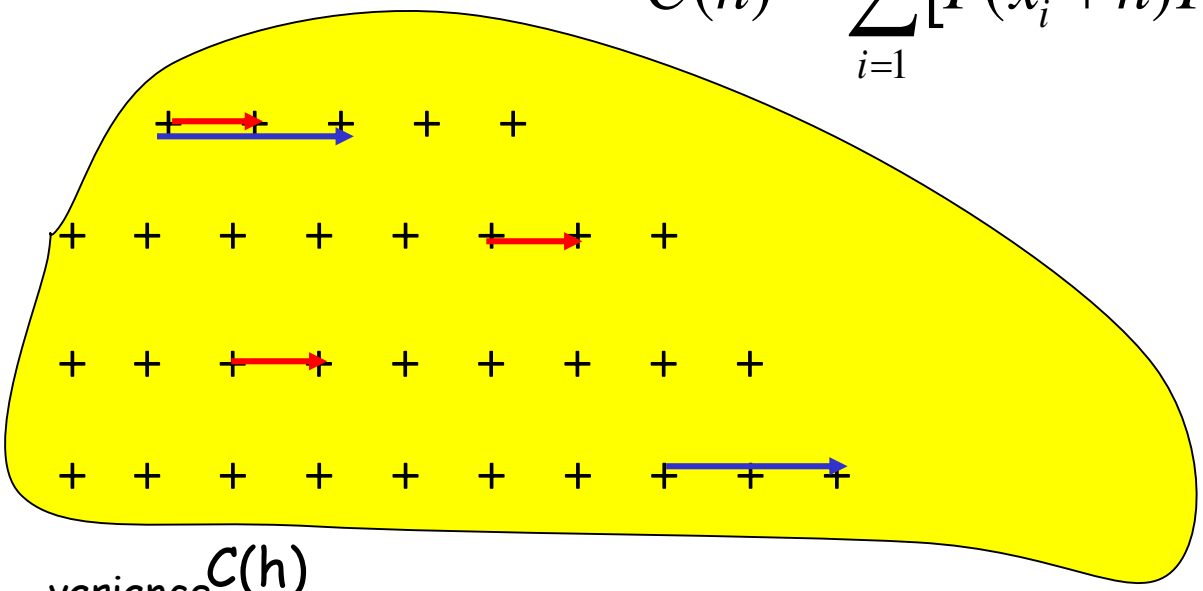
1. $E[Y(x)] = m$ does not depend on x
2. **Covariance:** $\mathrm{Cov}[Y(x), Y(x+h)] = \mathrm{E}\left[Y(x+h)Y(x)\right] - E\left[Y(x+h)\right]E\left[Y(x)\right] = C(h)$

does not depend on x



In practice

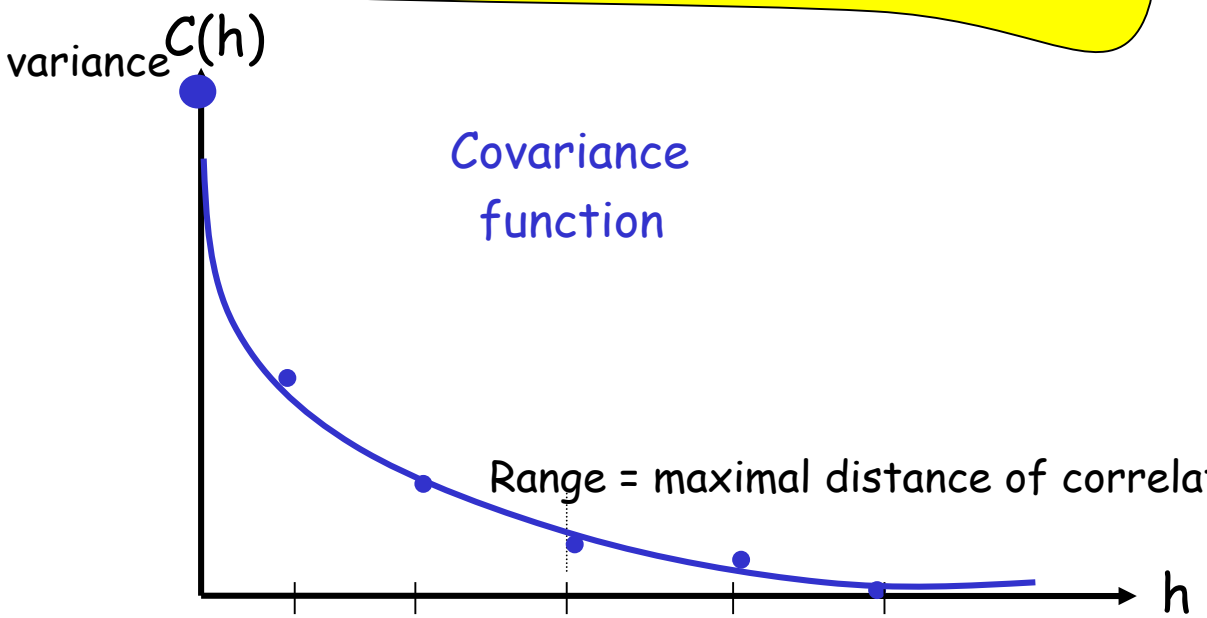
$$C(h) = \sum_{i=1}^{N(h)} \left[Y(x_i + h) Y(x_i) \right] - m^2$$



Var[Y(x)] = C(0)

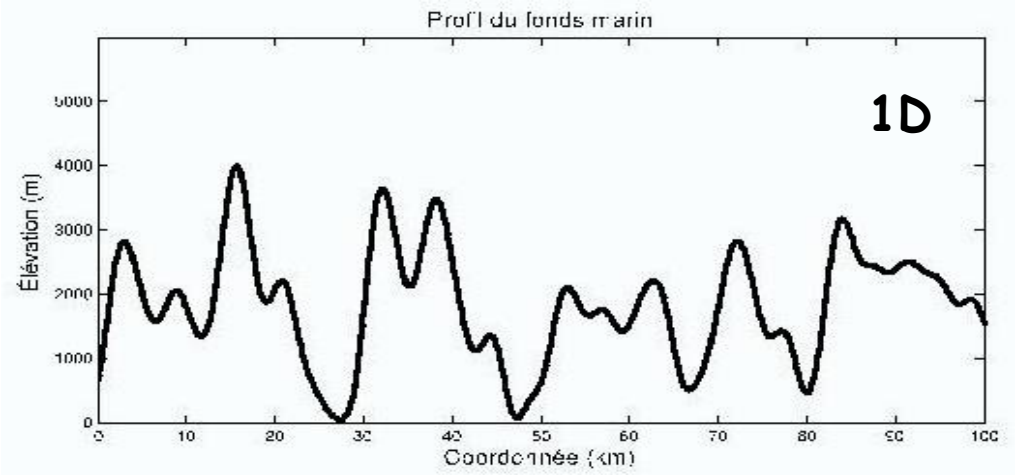
Y(x) → Y(x+h)

Y(x) → Y(x+2h)

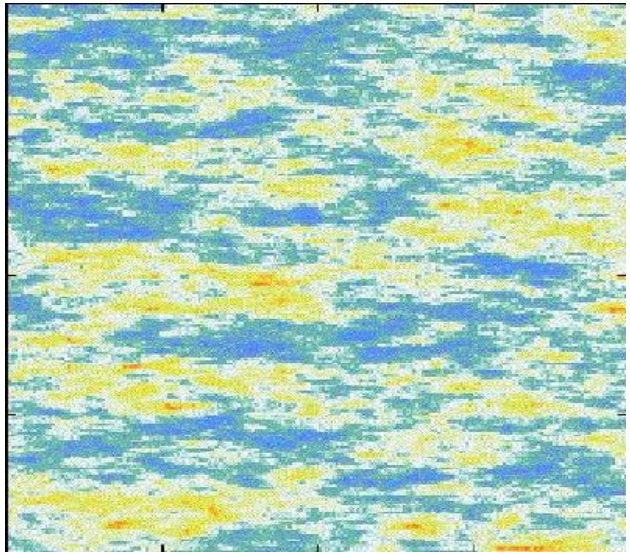


Range = maximal distance of correlation = correlation length

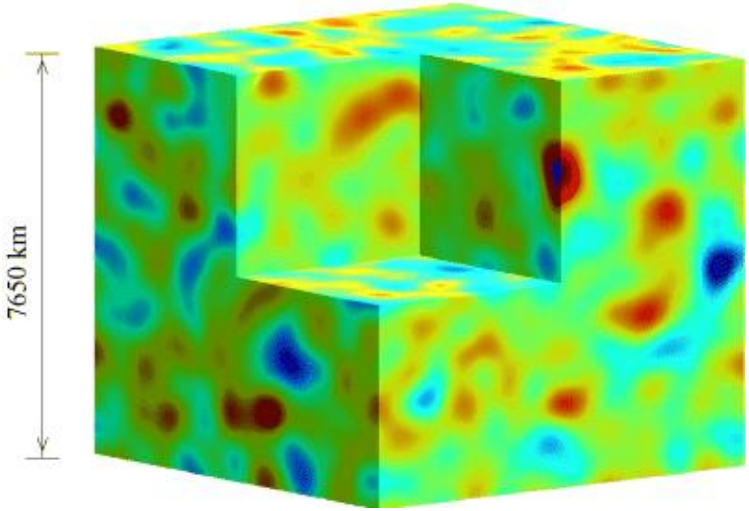
Examples of stochastic processes (Gaussian)



[from: Marcotte]



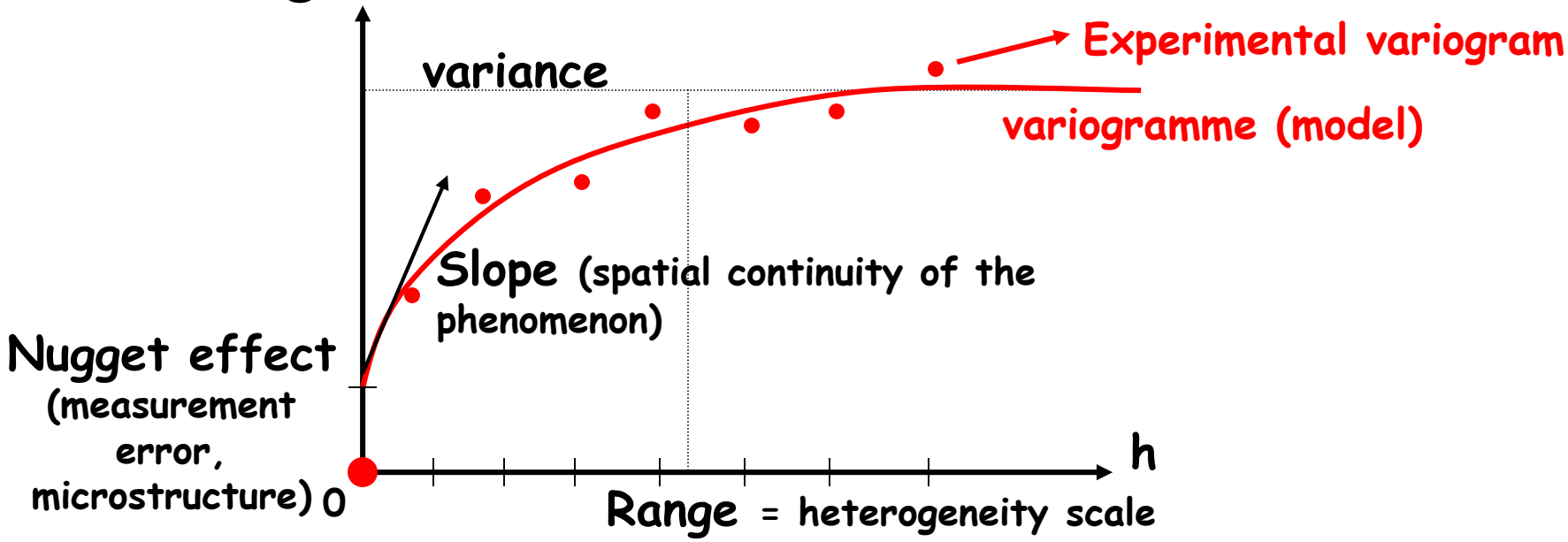
2D

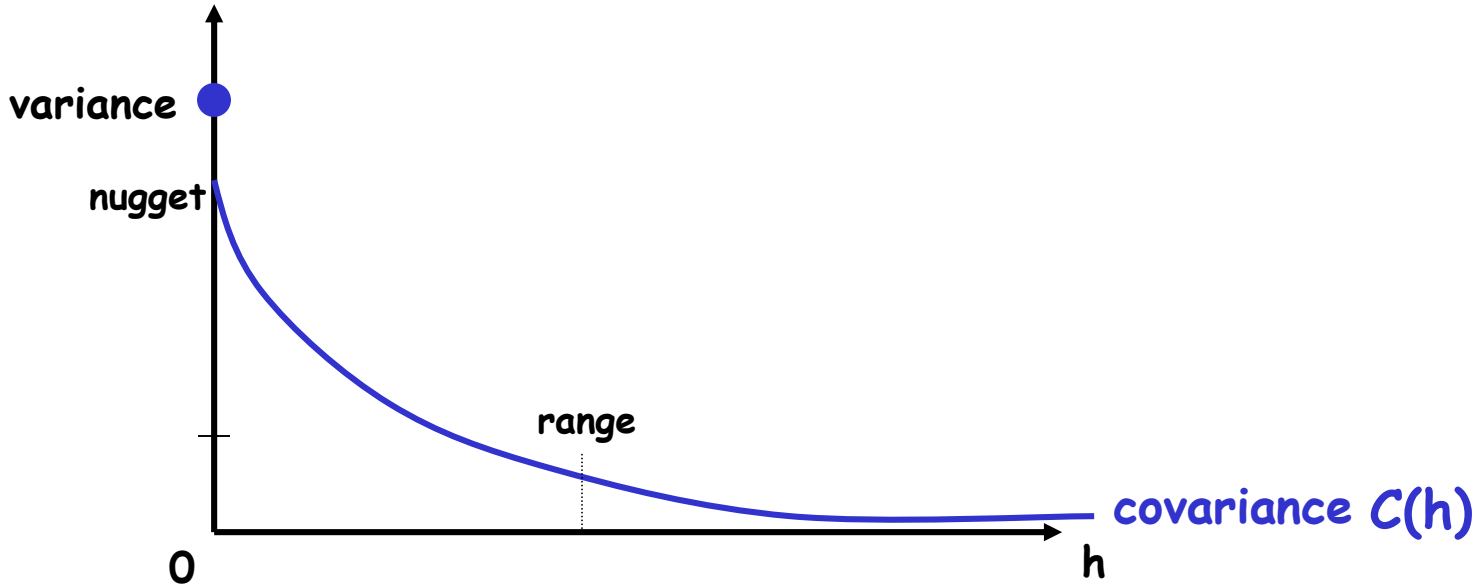


3D

[from: Baig, 2003]

The variogram





Simple kriging (known mean)

$$\boxed{Y^*(u) = \sum_{i=1}^{N} \lambda_i(u)\,[Y(u_i) - m] + m}$$

(m = known constant)

Min { E [$Y^*(u) - Y(u)$]^2 }
λ_i ➡ multiple linear regression by least squares

Best Linear Unbiased Predictor (BLUP)

Kriging weights $\lambda_i(u)$ for $Y(u_i)$ are obtained by:

$$\left\{ \sum_{j=1}^{N} \lambda_j(u)\, C(u_i - u_j) = C(u_i - u) \qquad \forall i = 1 \ldots N \right.$$

System of N linear equations with N unknowns which have an unique solution (for non singular covariance matrix)

Kriging variance (estimation error): $\sigma_K^2(u) = C(0) - \sum_{i=1}^{N} \lambda_i(u)\, C(u_i - u)$
does not depend on the Y values

=> Visualisation of regions with imprecise estimations
=> Put new observation points in these regions

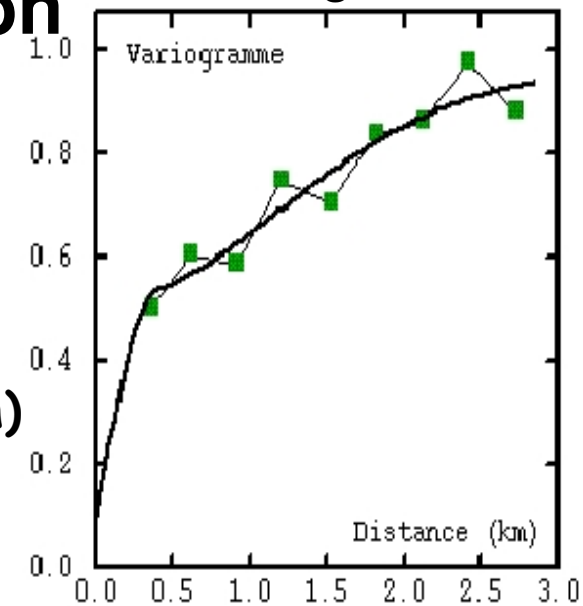
Example : cartography of air pollution

73 measures of benzene concentration (Rouen, France)

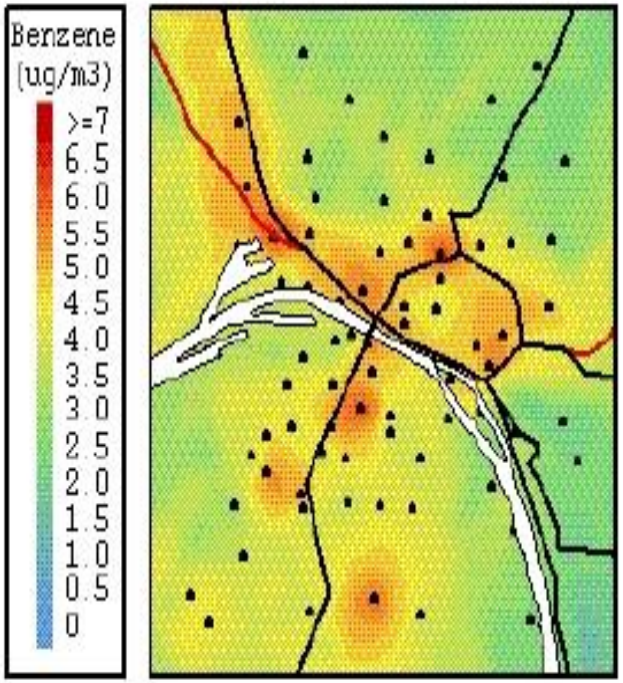
[from: Bobbia, Mietlicki & Roth, 2000]

$$\gamma(h) = C(0) - C(h)$$

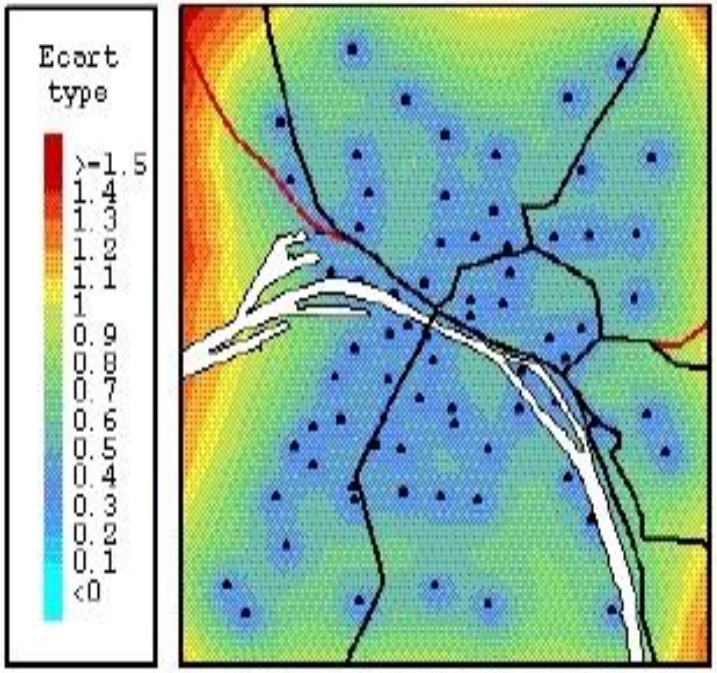
Variogram



Kriging mean



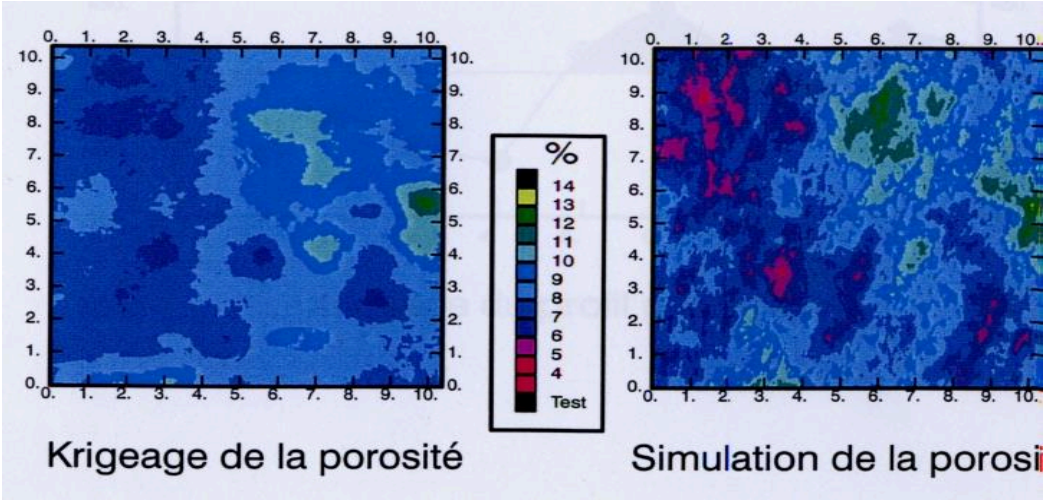
Kriging standard deviation



Simulations

- **Kriging give the optimal estimation** (unbiased, minimal error variance) of the variable at any point, from experimental data

- **A simulation represents a possible realization** of the real phenomenon
  It reproduces its true **variability** (distribution, variogram), with respecy to experimental data **(conditional simulation)**



Main goal of simulation : quantify the uncertainty via sampling (as Monte Carlo)

Numerous methods of random fields simulation (LU decomposition, turning bands method, spectral method, Karhunen-Loève, etc.)

Simulation of a Gaussian vector: Covariance decomposition

Problem:

Given the N x N covariance matrix C, simulate multivariate normal (mean 0) random vector X with covariance E[XX'] = C

Solution:

1) Decompose C = SS' where S is a triangular matrix
2) Generate a standard normal vector Z with covariance equal to unit matrix I
3) Let X = SZ

Proof:

Note that E[XX'] = E[SZZ'S'] = S E[ZZ'] S' = SS' = C

The actual method for finding such matrix S is not unique:

LU-decomposition, Cholesky or matrix square root

Conditional simulations

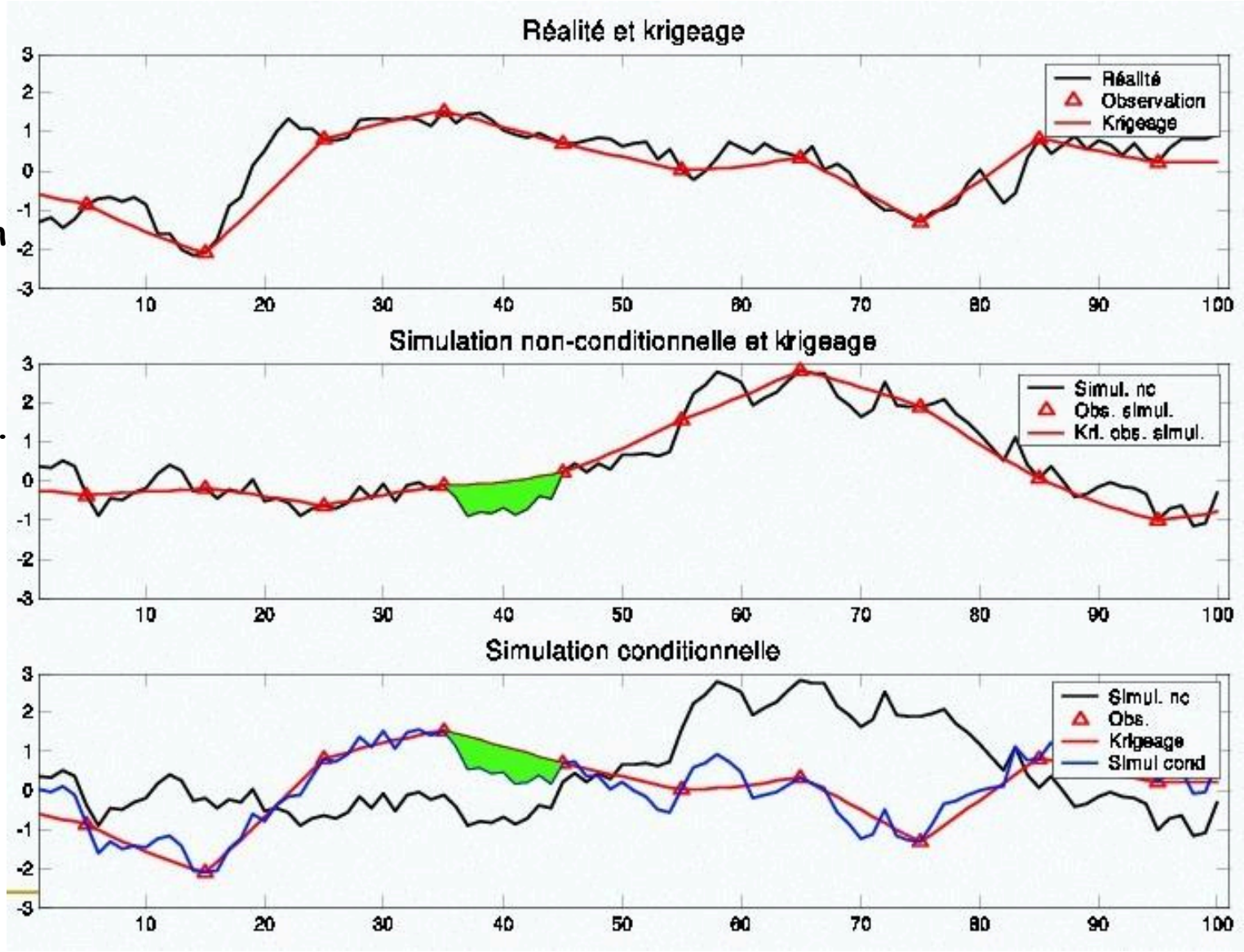
Kriging
of observation

+

Non cond. Simul.
-
Kriging of
non cond. simul

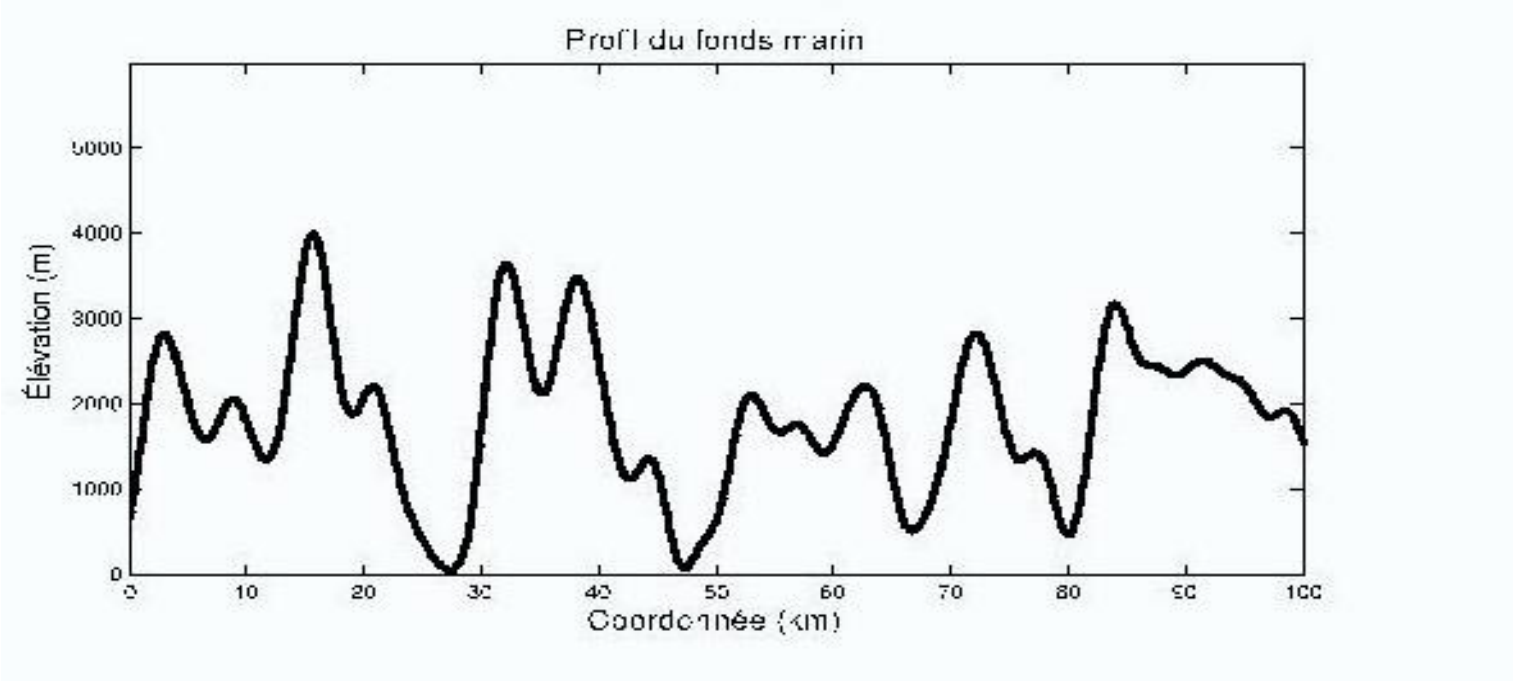
=

Conditional
simulation



[Marcotte, Cours EPM]

Example : profile of ocean bottom (1/5)



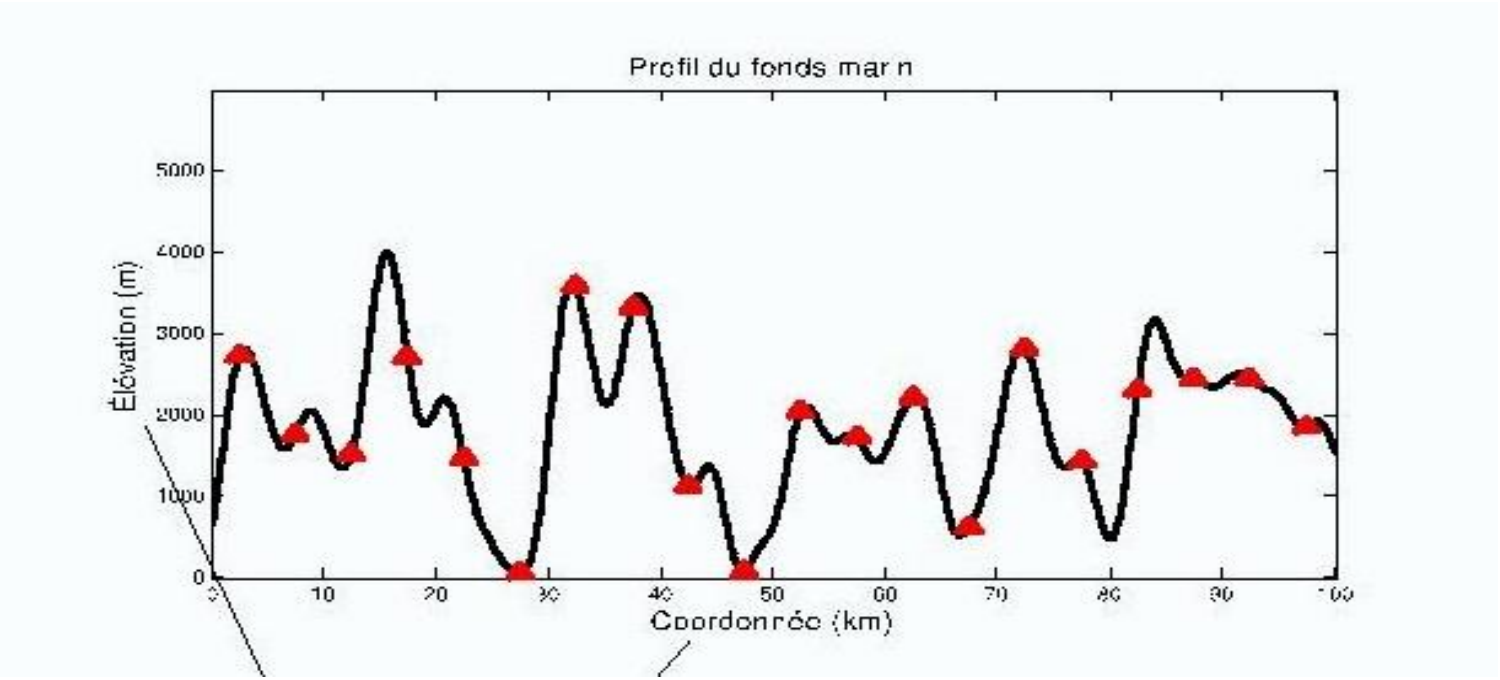
You have to put a cable on the ocean bottom

Question: what is the length of the cable?

[Marcotte, Cours EPM]

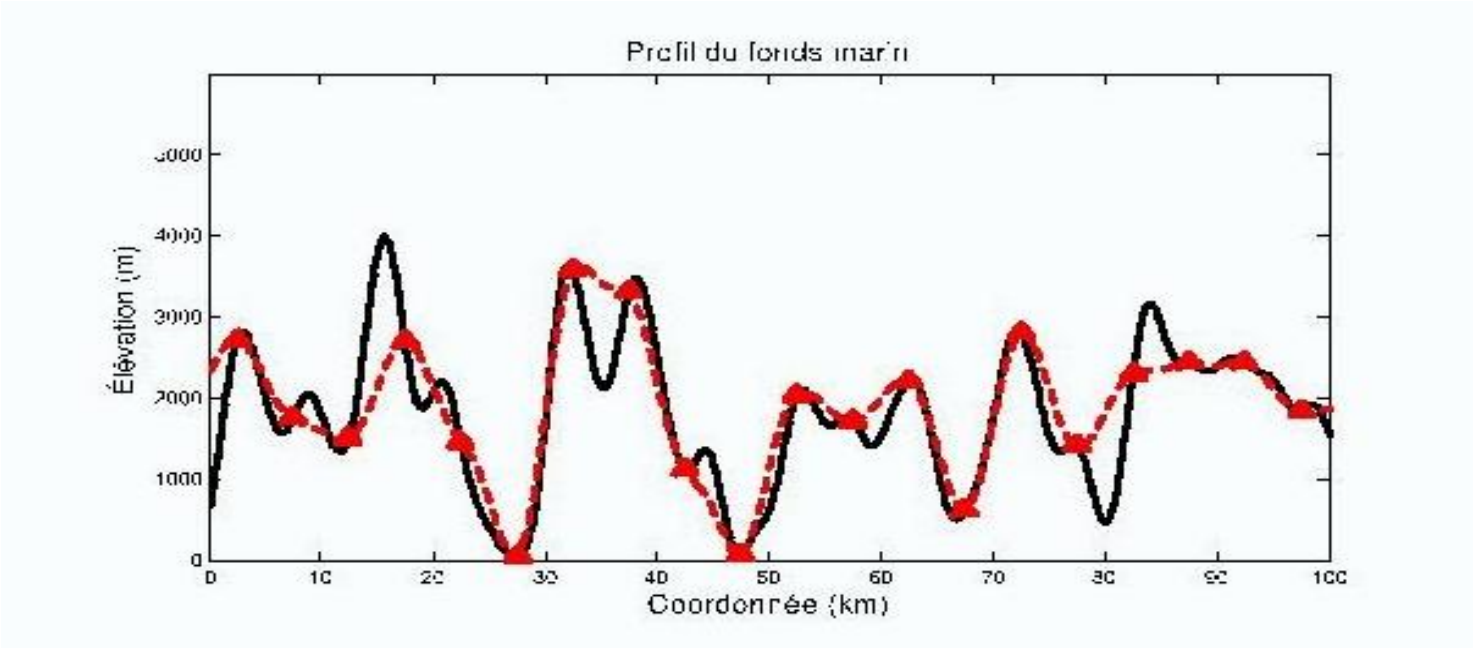
Example : profile of ocean bottom (2/5)

The exact depth is uniquely known at the observation points (survey)



Example : profile of ocean bottom (3/5)

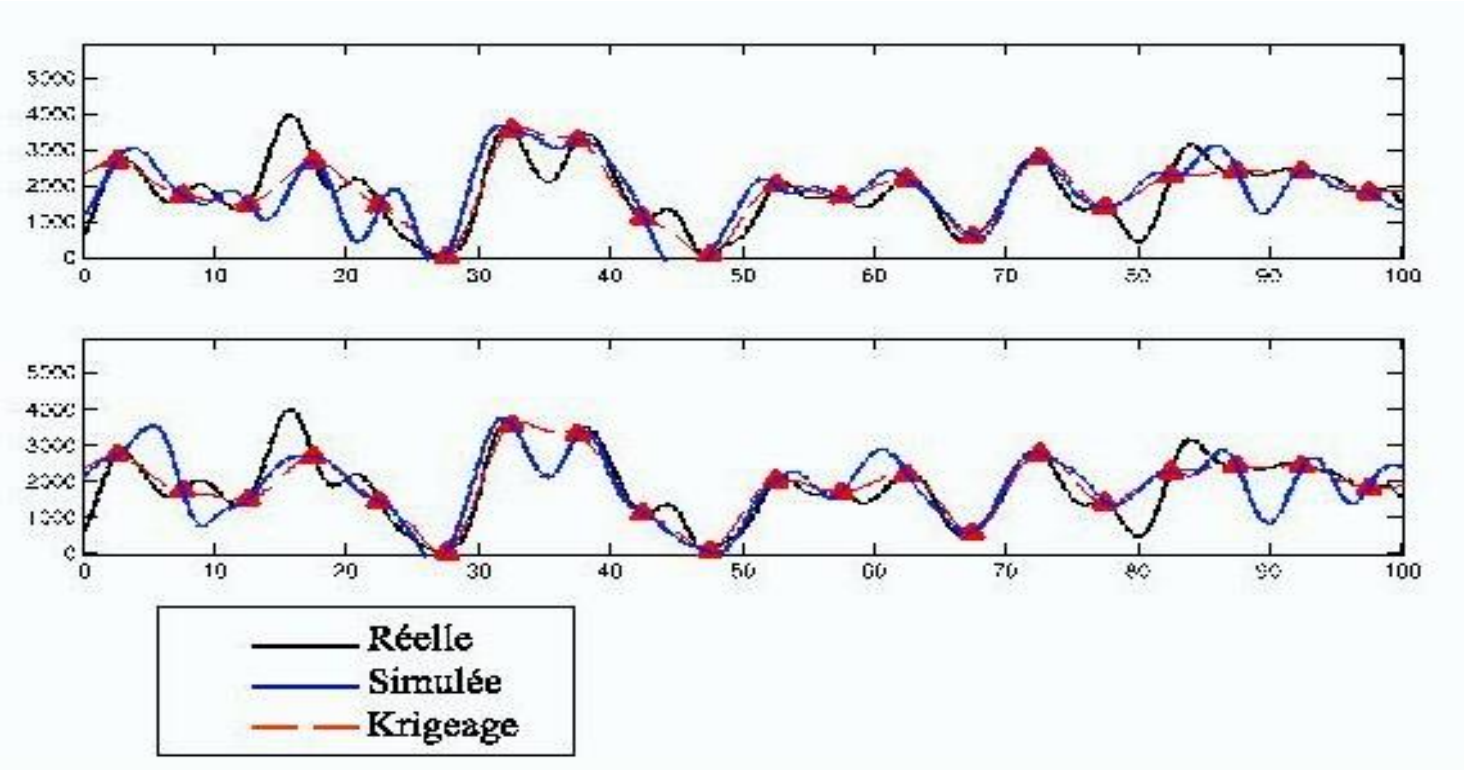
Kriging of the ocean depth



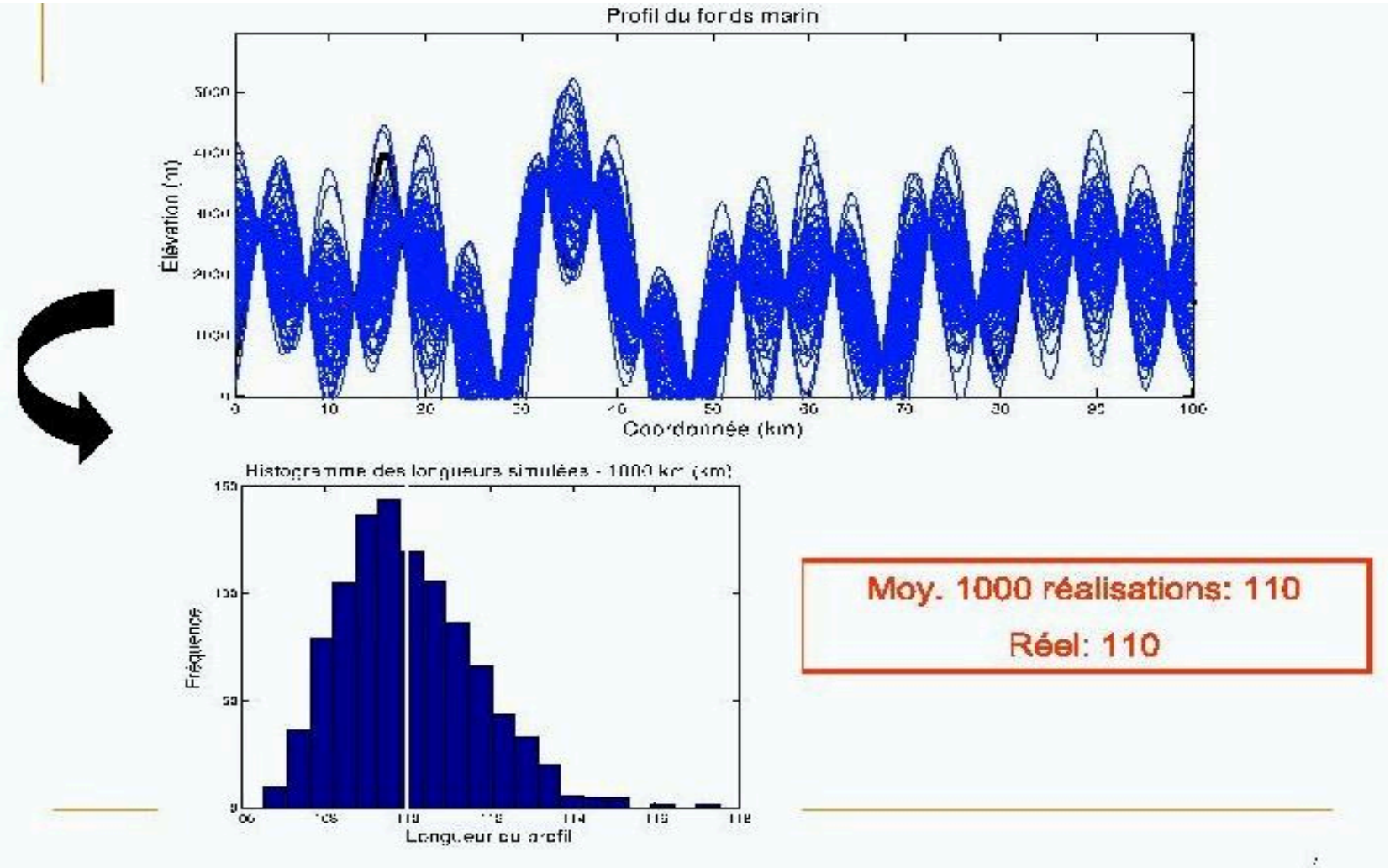
The true length is 110 km while kriging gives 104.6 km
=> some cable is missing

Example : profile of ocean bottom (4/5)

Another approach: the conditional simulations



Example : profile of ocean bottom (5/5)



The 95%-confidence interval from conditional simulations is [108.8,113.5]

Same problem for probability of failure estimation (non linear transfer fct)

Gaussian process metamodel

Gaussian process metamodel (1/2)

- Idea: Computer code results are interpolated with the kriging technique
- Necessary hypothesis: Gaussian process

- Definition:

$$Y(x) = \beta F(x) + Z(x)$$

Regression — stochastic part

Stochastic process Z with :

$$\begin{cases} E[Z(x)] = 0 \\ Cov(Z(x), Z(u)) = \sigma^2 R(x, u) \\ \text{where } \sigma^2 \text{ is the variance} \\ \text{and } R \text{ the correlation function} \\ Z \sim \mathcal{N}(0, \sigma^2 R) \end{cases}$$

- Parametric choices:
  - F : polynomial of degree 1 $\beta F(\mathrm{x}) = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$
  - R : stationary => covariance function

Example: Gaussian covariance $\mathrm{R(x,\ u)} = \mathrm{R(x - u)} = \exp\left(-\sum_{i=1}^{p} \theta_i \left| x_i - u_i \right|^2 \right)$

Anisotropy: θ_is are not equal (correlation length of each input variable)

Gaussian process metamodel (2/2)

- <u>Joint distribution :</u>
  - Gaussian process (Gp) model : $Y(x) = \beta F(x) + Z(x)$, $x \in \Re^p$
  - Learning sample (LS) of N simulations : (X_{LS}, Y_{LS})

$$X_{LS} = \left(x^{(1)}, \ldots, x^{(N)}\right), \; F_{LS} = F(X_{LS}), \; R_{LS} = \left(R\left(x^{(i)}, x^{(k)}\right)\right)_{i,k}$$
$$Y_{LS} \sim N(\beta F_{LS}, \sigma^2 R_{LS})$$

  - Conditional Gp metamodel :

$$\Longrightarrow \quad Y(x)_{|X_{LS}, Y_{LS}} \sim Gp$$

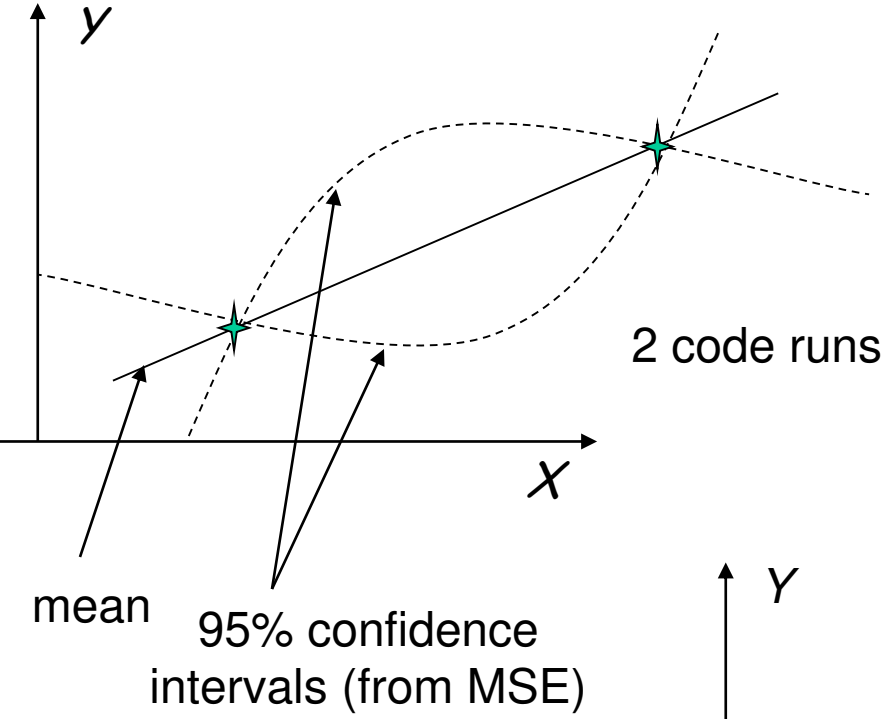
$$\text{Mean} : \hat{Y}(x) = \mathrm{E}\left[Y(x)_{|X_{LS}, Y_{LS}}\right] = \beta F(x) + r(x) R_{LS}^{-1}\left[Y_{LS} - \beta F_{LS}\right]$$

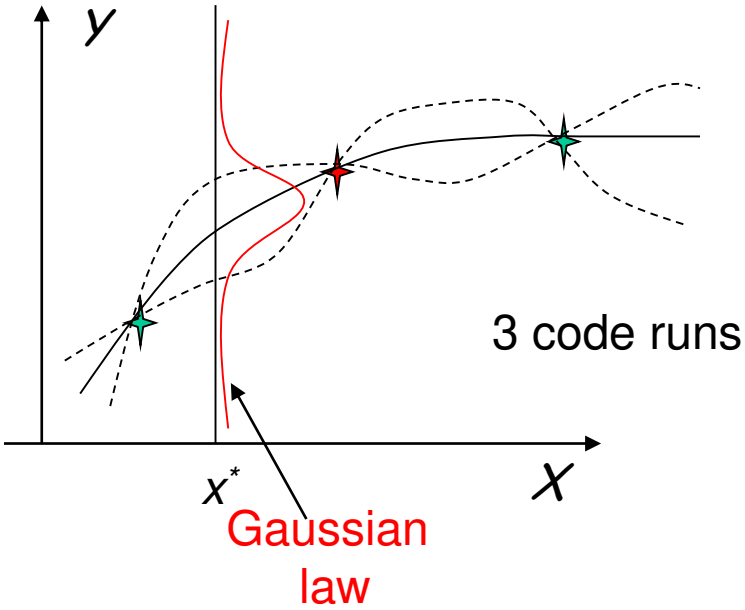
$$\text{with} \quad r(x) = \left[R\left(x^{(1)}, x\right), \ldots, R\left(x^{(N)}, x\right)\right]$$

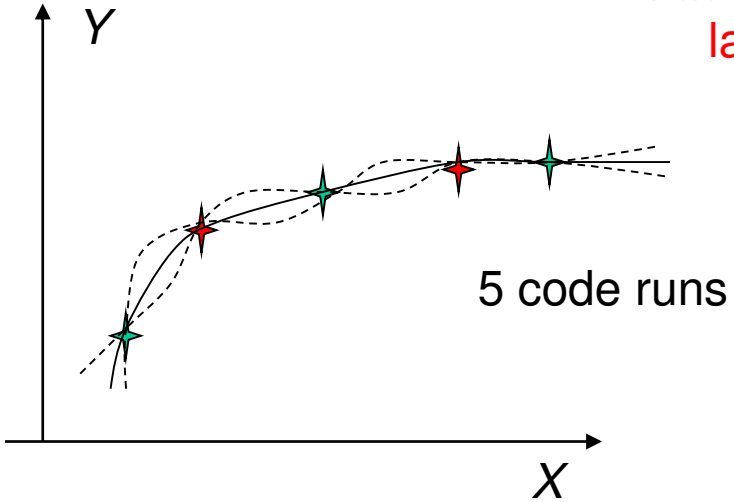
$$\text{Covariance} : \mathrm{Cov}\left(Y(u)_{|X_{LS}, Y_{LS}}, Y(v)_{|X_{LS}, Y_{LS}}\right) = \sigma^2\left(R(u,v) + {}^t r(u) R_{LS}^{-1} r(v)\right)$$

$$\Rightarrow \text{Variance} \Rightarrow \text{Mean Square Error (MSE)}$$

Illustration







Conclusion: Given a sufficient number of points, we obtain an accurate metamodel

Hyperparameters estimation

- **Maximum likelihood method**

  - Likelihood maximisation on the learning basis (X_s,Y_s):

$$\left(\beta^*,\theta^*,\sigma^*\right)=\underset{(\beta,\theta,\sigma)}{\mathrm{Argmax}}\ \ln L\left(Y_{LS},\beta,\theta,\sigma\right)$$

  - with

$$\ln L\left(Y_{LS},\beta,\theta,\sigma\right)=-\frac{N}{2}\ln\left(2\pi\sigma^2\right)-\frac{1}{2}\ln\left(\det R_{LS}\right)-\frac{1}{2}\sigma^2\,{}^t\left[Y_{LS}-\beta F_{LS}\right]R_{LS}^{-1}\left[Y_{LS}-\beta F_{LS}\right]$$

  - Joint estimation of β and σ :

$$\begin{cases}\beta^*=\left[{}^tF_{LS}R_{LS}^{-1}F_{LS}\right]^{-1}{}^tF_{LS}R_{LS}^{-1}Y_{LS}\\ \sigma^{2*}=\dfrac{1}{N}\,{}^t\left[Y_{LS}-\beta^*F_{LS}\right]R_{LS}^{-1}\left[Y_{LS}-\beta^*F_{LS}\right]\end{cases}$$

  - Estimation of correlation parameters θ :

$$\left(\theta^*\right)=\underset{\theta}{\mathrm{Argmin}}\ \ \psi\left(\theta\right)\quad\text{with}\quad\psi\left(\theta\right)=\left|R_{LS}\right|^{-1/N}\sigma^{2*}$$

Estimation and validation

$$R(\mathbf{u},\mathbf{v}) = R(\mathbf{u}-\mathbf{v}) = \exp\left(-\sum_{i=1}^{p}\theta_i|u_i - v_i|^2\right)$$

- **Hyperparameters $(\theta_i)_{i=1\ldots p}$ estimated by likelihood maximization**

Simplex method, **stochastic algorithms**

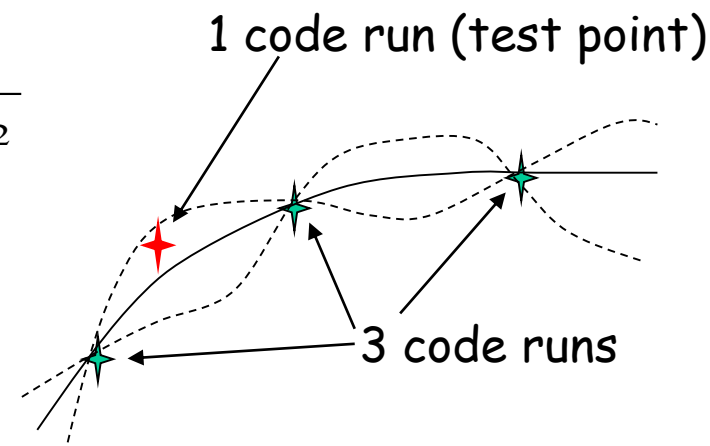
Problems in high dimensional context (p > 10), can be solved by sequential algorithms *[Marrel et al. 2008]*

- **Predictor validation:**

Predictivity coefficient

$$Q_2(Y,\hat{Y}) = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(\bar{Y} - Y_i\right)^2}$$

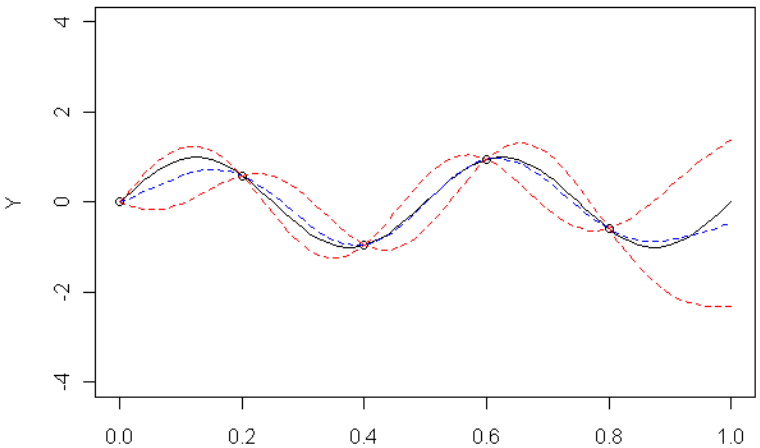
- Test sample
- or leave-one-out
- or *k*-fold cross validation



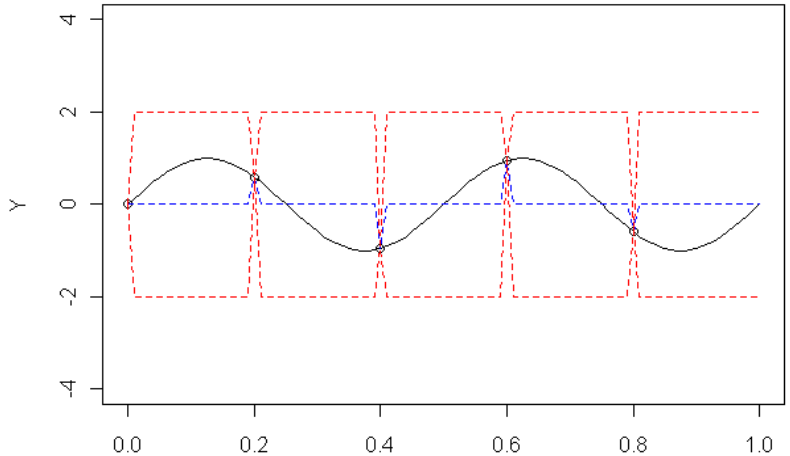
- **MSE validation:** Percentage of predicted values inside confidence bounds

Effects of the hyperparameters θ and σ

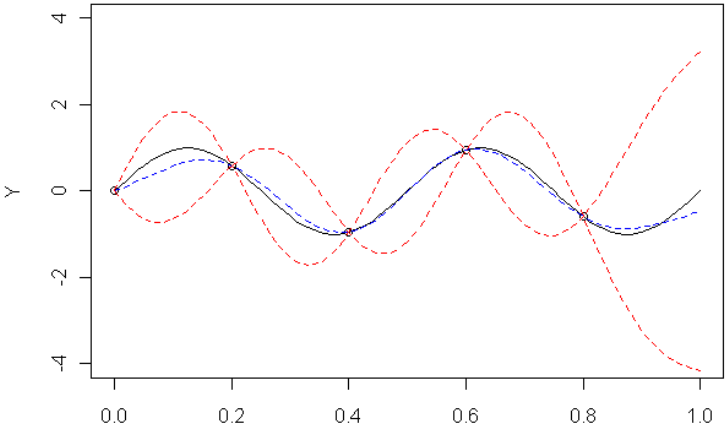
$$f(x) = \sin(4\pi x)$$



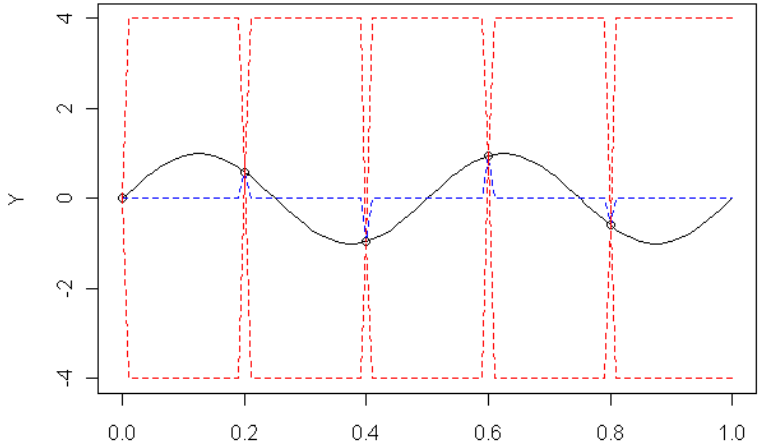
$$\sigma^2 = 1; \theta = 0.2$$



$$\sigma^2 = 1; \theta = 10^{-4}$$



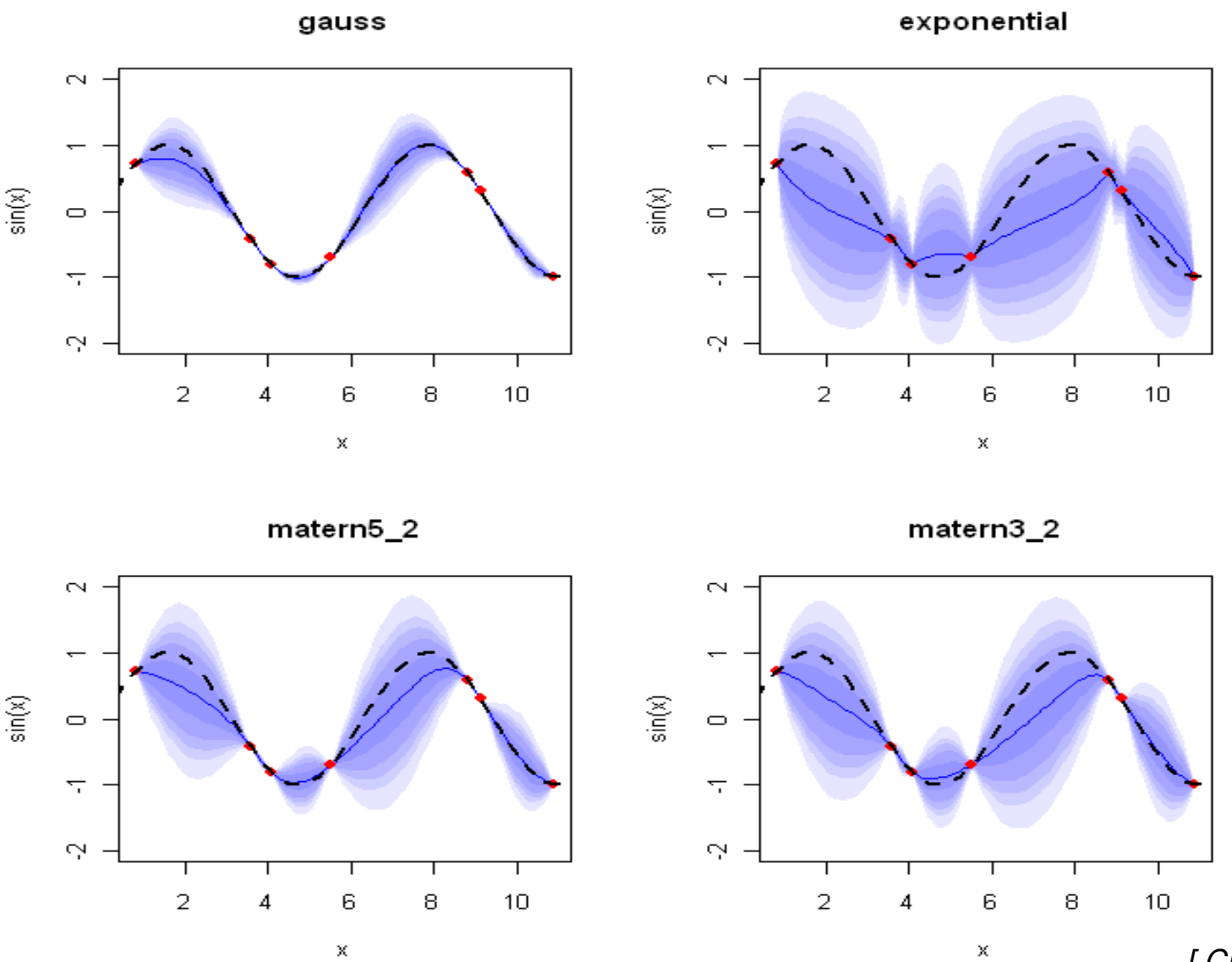
$$\sigma^2 = 4; \theta = 0.2$$



$$\sigma^2 = 4; \theta = 10^{-4}$$

[Le Gratiet, 2011]

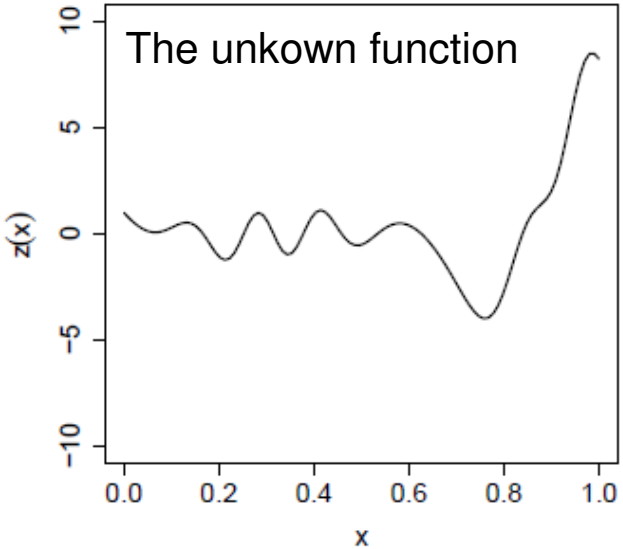
Effects of the covariance structure

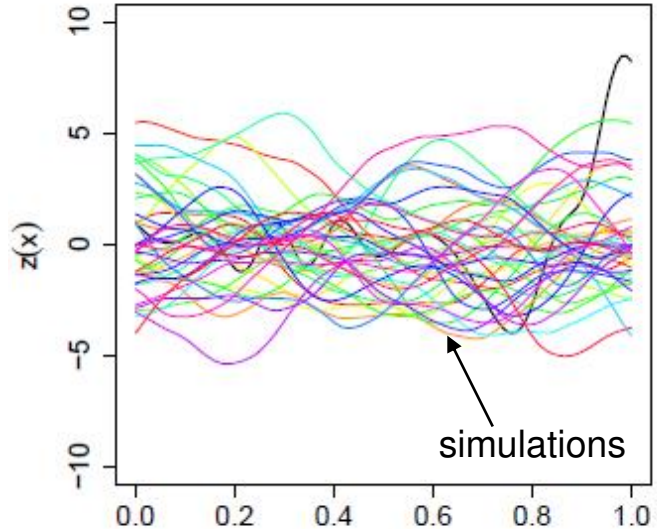


[Chevalier, 2011]

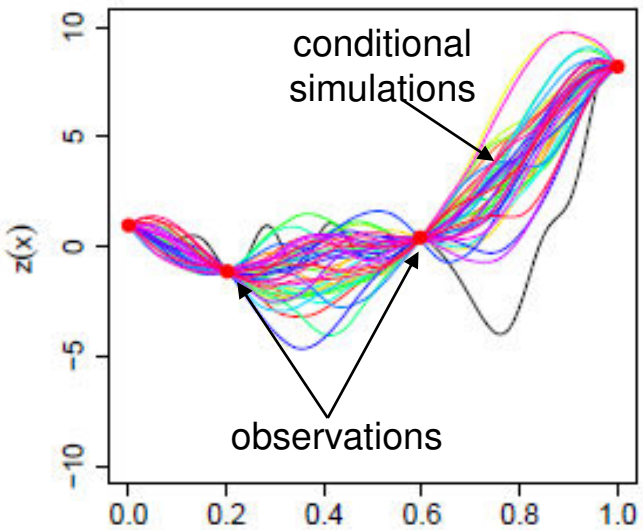
GP metamodel in summary

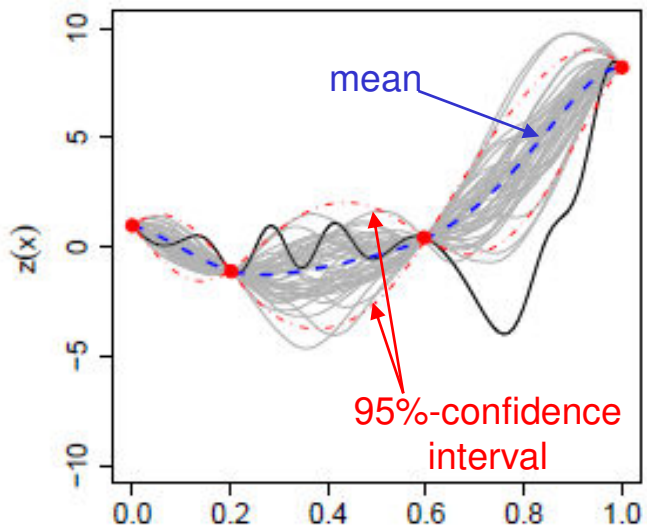
[Le Gratiet, 2014]





Main hypothesis: z(x) is the realization of a **Gaussian process** (defined by its mean and cov)





GP metamodel is given by the the **probability law of Z(x) conditionally to the observations**

Its mean gives the GP predictor
Its variance gives the error

Adaptive designs using Gaussian process metamodel

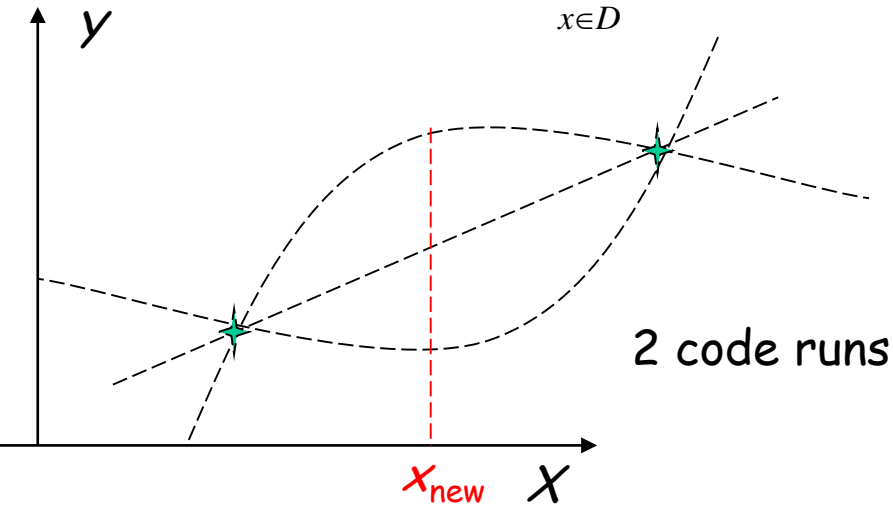
The best way to build Gp: model-based adaptive designs

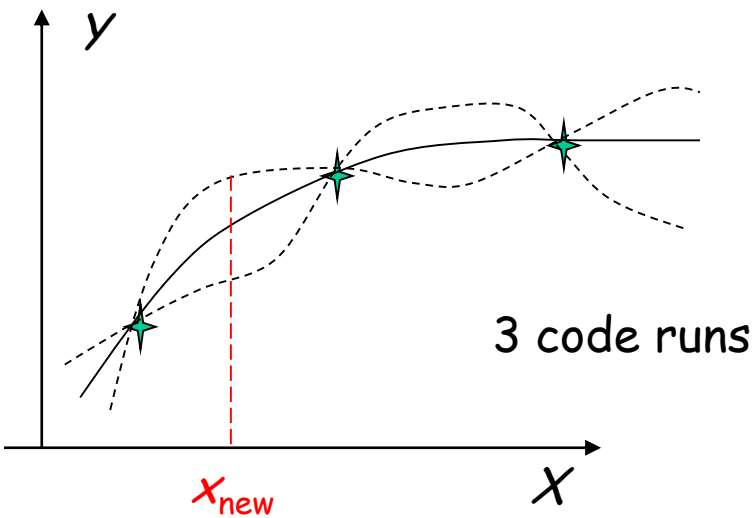
Example: criterion of the Gaussian process MSE (Mean Square Error)

$$MSE(x) = \sigma^2 + {}^t r(x) R_{LS}^{-1} r(x) + u(x)({}^t(\beta F_{LS}) R_{LS}^{-1} \beta F_{LS})^t u(x)$$

$$u(x) = \beta F(x) - {}^t k(x) R_{LS}^{-1} \beta F_{LS}$$

$$x_{\text{new}} = \arg\max_{x \in D} MSE(x)$$





<u>Remark:</u> other criteria are possible (e.g. focusing to active variables)

<u>Conclusion:</u> Model-based adaptive designs are the most efficient ones,
but are not always applicable
In practice, we need to initiate the process with a space-filling design

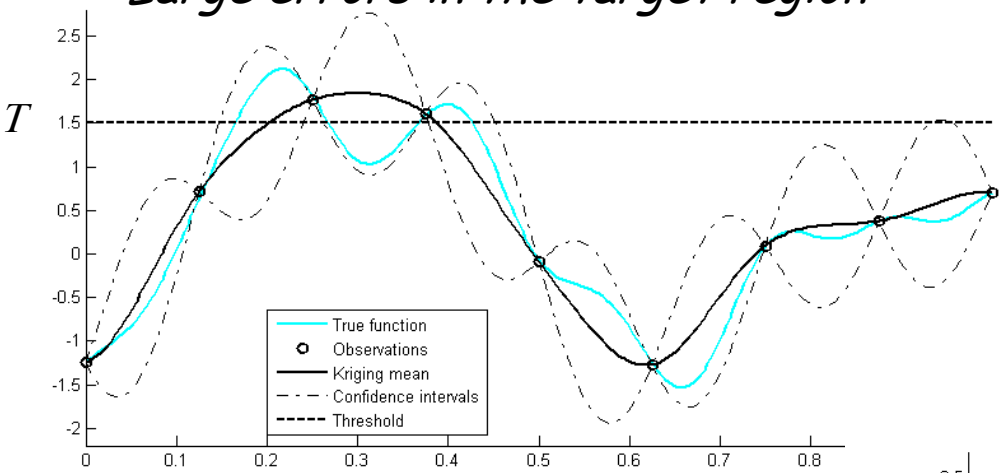
Estimation of rare events probability using kriging

[Bect et al. 2012]

Industrial problems: safety analysis with computer code (nuclear, transport, …)

<u>Problem:</u> find P_f = Prob [$f(X) > T$] with X = random inputs ; T = treshold

Reasonable variance everywhere
Large errors in the target region



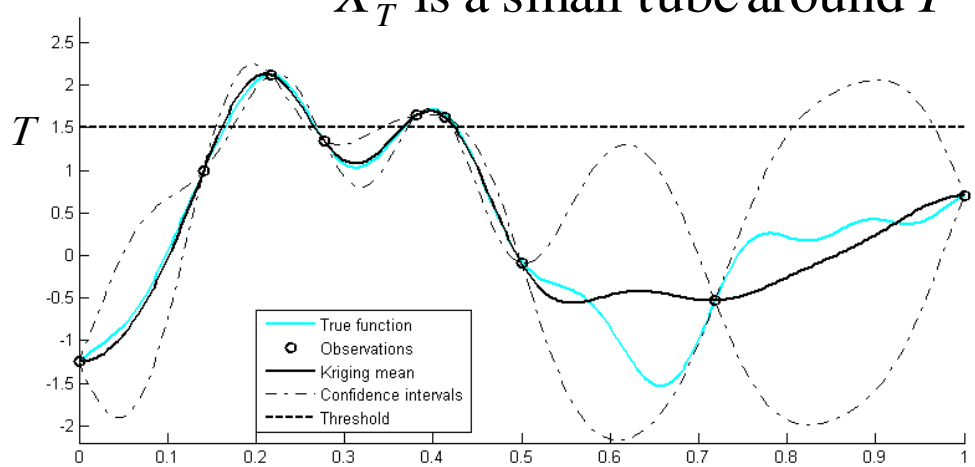
[from: Picheny et al. 2010]

☺ **New adaptive design**

$$X^* = \arg\min_X (IMSE_T)$$

$$IMSE_T = \int MSE(x) 1_{X_T}(x) dx$$

X_T is a small tube around T



Large variance in non-target region
Good accuracy in target region

Optimisation of a model output using kriging

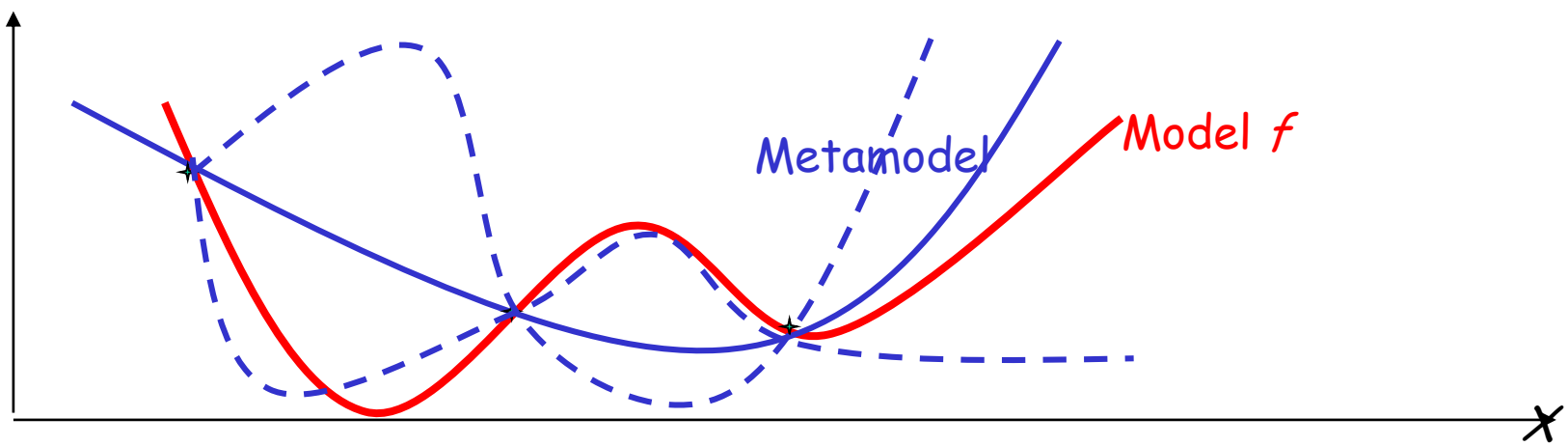
Industrial problems: conception with costly computer code (automobile, nuclear, aeronautics, …)

<u>Problem:</u> find the values of X which minimize the model output

$$X^* = \arg\min_{X \in D} f(X)$$

☠ **If *f* is costly, a natural solution would be to optimize a metamodel of *f* :** dangerous idea because the metamodel tends to smooth the true model

☺ Gp metamodel allows to take into account the metamodel error, and to define the expected improvement EI*(X)* for each $X \in D$

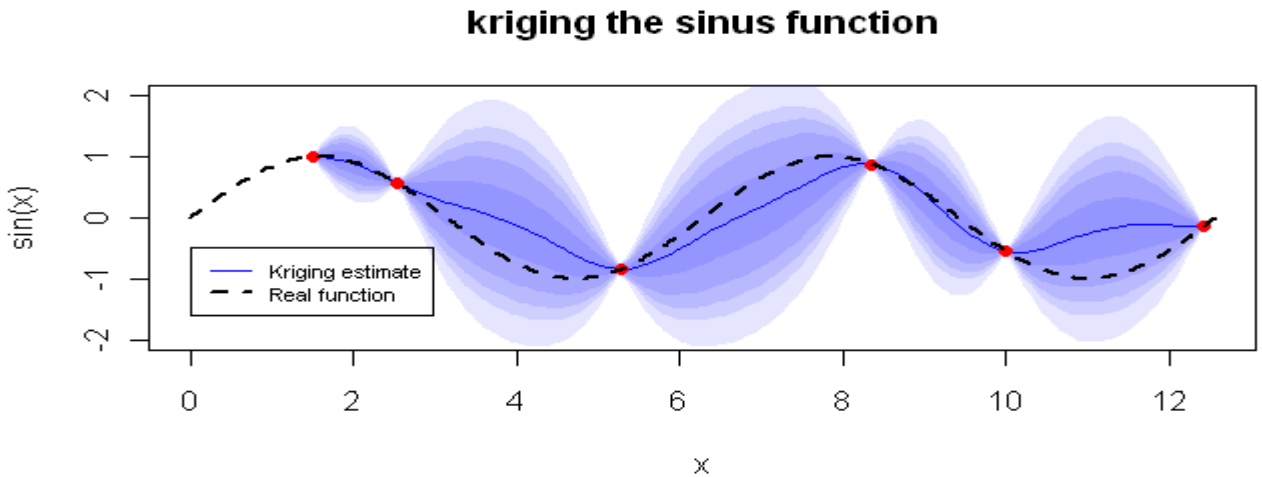


EI(x) = E[max(0 , observed minimum – f(x))]

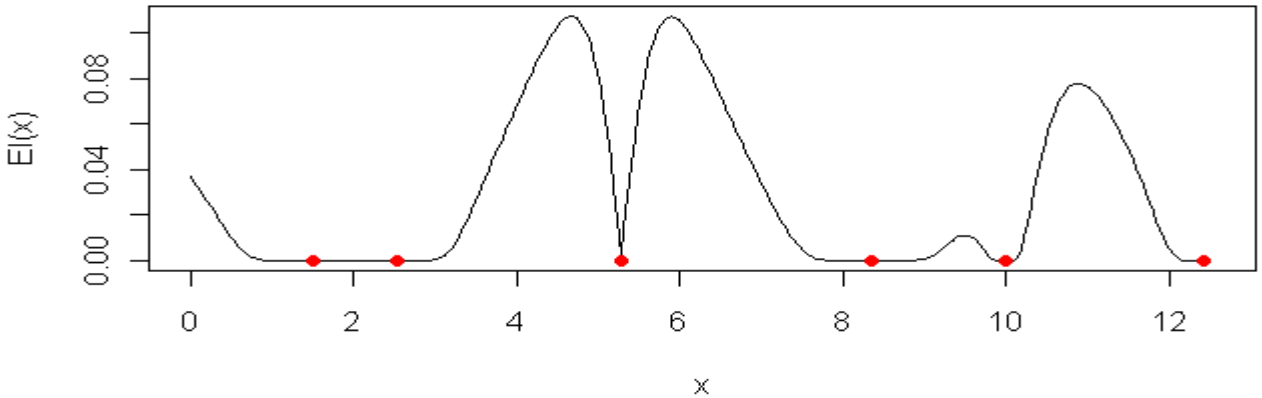
Adaptive design for optimization: EGO algorithm

[Jones et al. 1998]

EGO: step 0

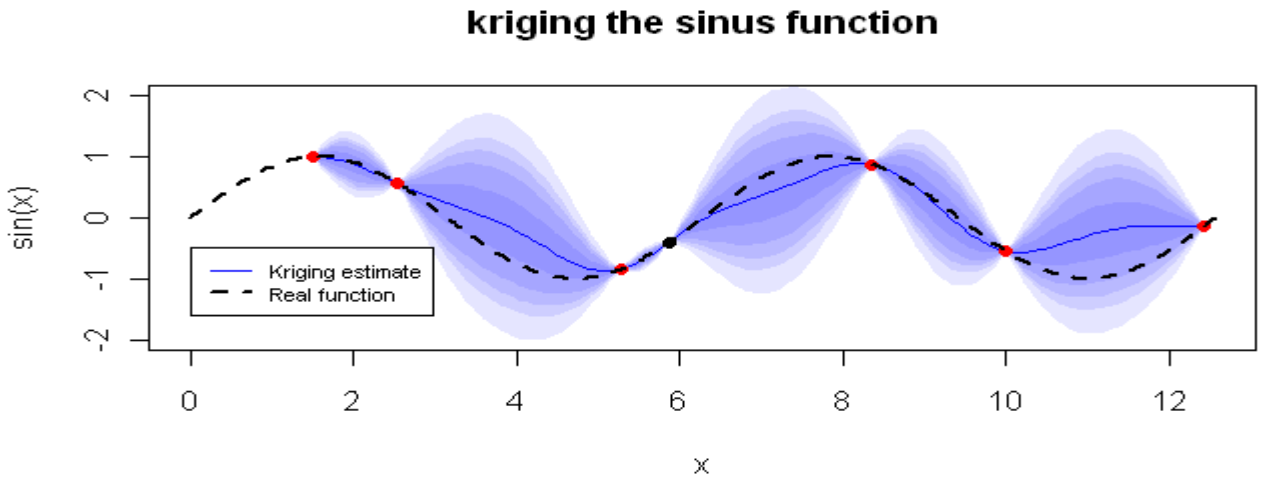


[Chevalier, 2011]

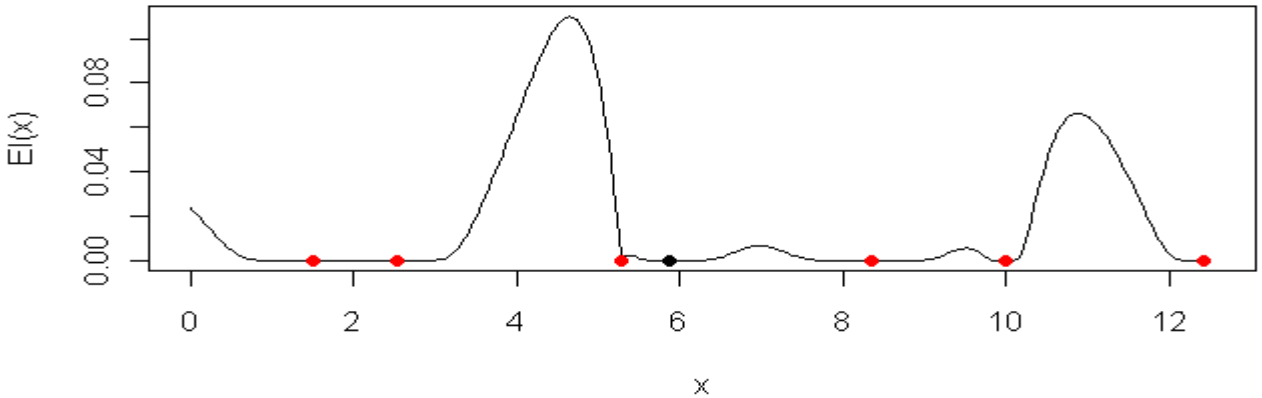


Adaptive design for optimization: EGO algorithm

EGO: step 1

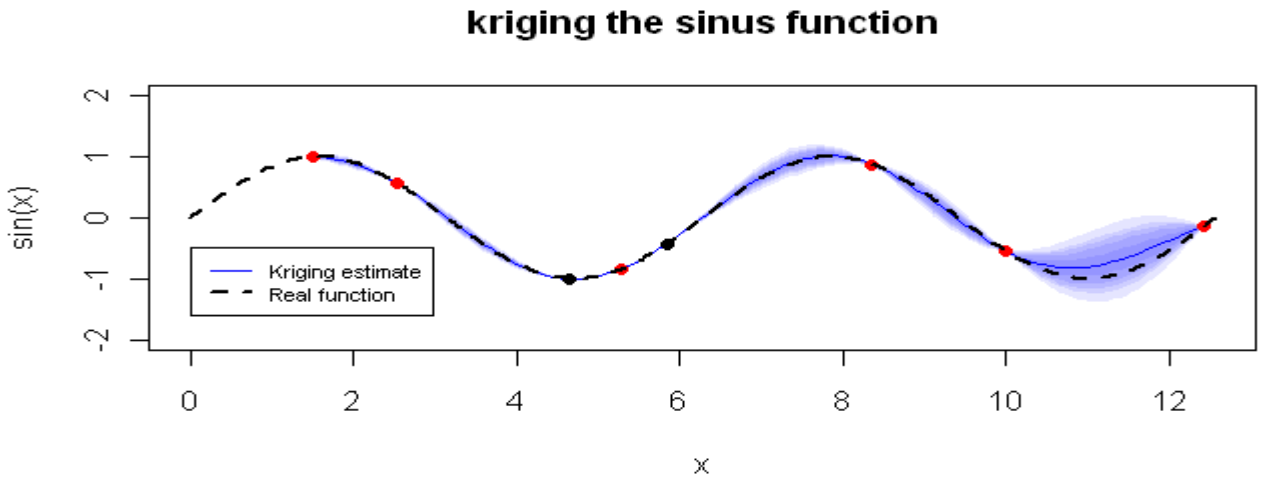


[Chevalier, 2011]

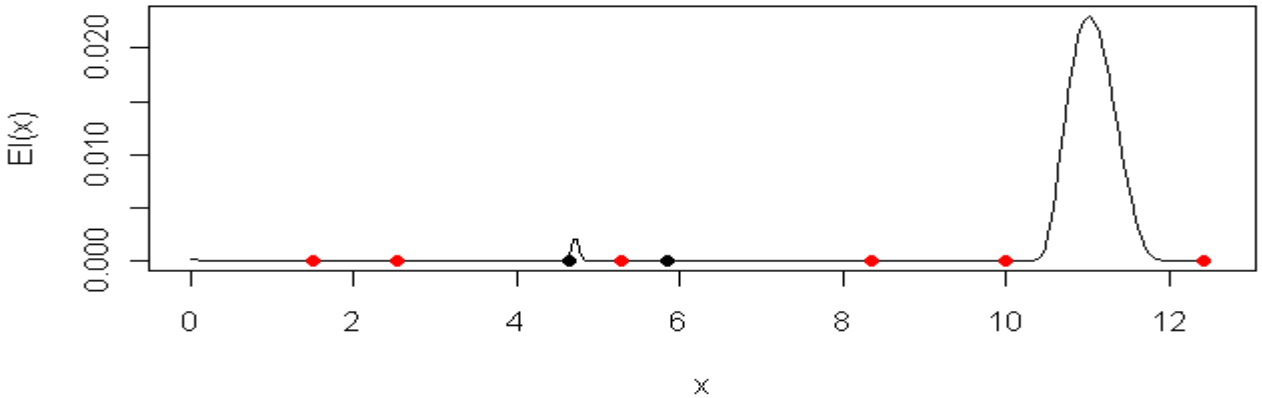


Adaptive design for optimization: EGO algorithm

EGO: step 2

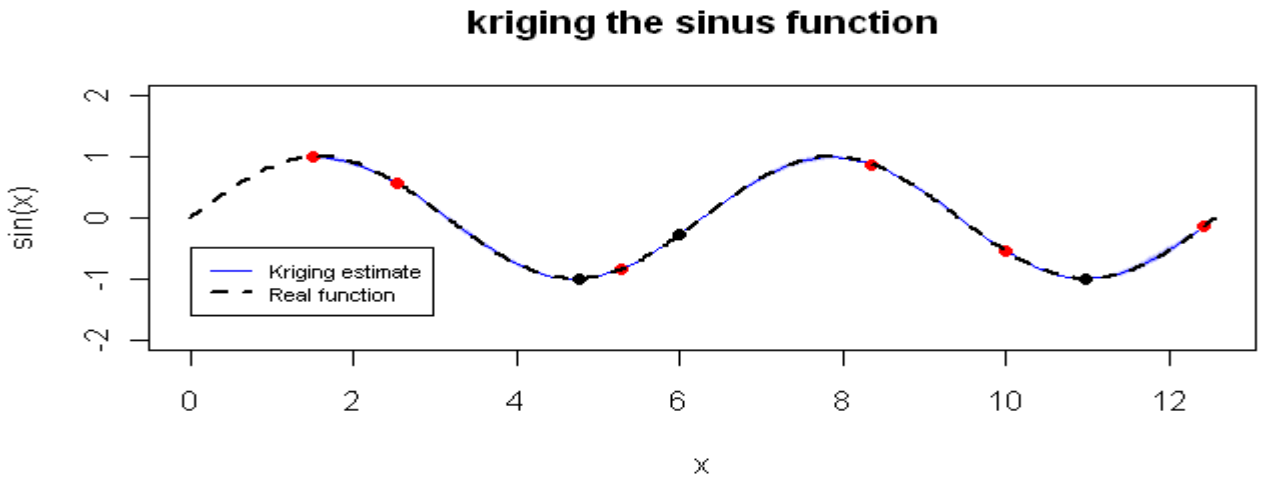


[Chevalier, 2011]

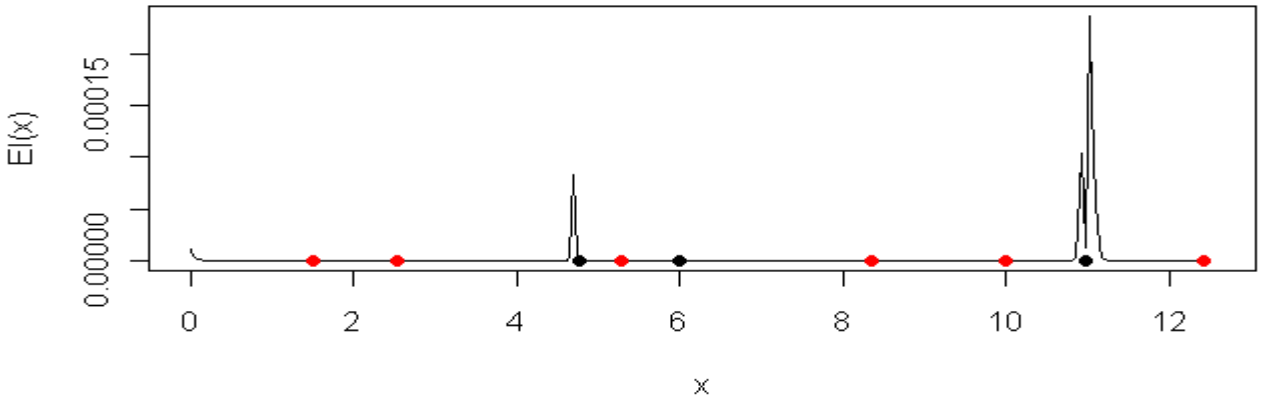


Adaptive design for optimization: EGO algorithm

EGO: step 3

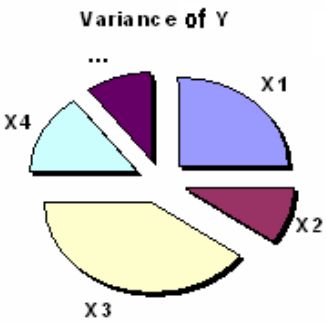


[Chevalier, 2011]



Sobol' indices estimation using Gaussian process metamodel

Sobol indices



- Definitions for a deterministic function $Y = f(X_1, \ldots, X_K)$

$$\text{First order}: \; S_i = \frac{V_i}{V} = \frac{\mathrm{Var}_{X_i}[\mathrm{E}(f(X_1, \ldots, X_K) | X_i)]}{V}$$

$$\text{Total}: \; S_{Ti} = \frac{V_{-i}}{V} = \frac{\mathrm{Var}[\mathrm{E}(f(X_1, \ldots, X_K) | X_{-i})]}{V}$$

Notation : $V = \mathrm{Var}(Y)$; $X_{-i} = [X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_K]$

Hypothesis: independent inputs

- Classical approach estimation: replace f (X) by $\hat{Y}$ (GP mean)

☺ + Computationally cheap (possible to perform analytical calculations)

☠ - Do not infer from the metamodel error

Sensitivity analysis with GP model : 2 analytical approaches

- Gp model conditionally to LS points :

$$\Longrightarrow Y(x,\omega)_{|X_{LS},Y_{LS}} \sim \mathrm{GP} \begin{cases} \hat{Y}(x) = E_{\Omega}\left[Y(x,\omega)_{|X_{LS},Y_{LS}}\right] \\ Cov_{\Omega}\left(Y(u,\omega)_{|X_{LS},Y_{LS}}, Y(v,\omega)_{|X_{LS},Y_{LS}}\right) \end{cases}$$

- Computation of Sobol indices :

From predictor formula *[Chen et al. 2005]*

$$\hat{Y}(X) = E_{\Omega}\left[Y(X,\omega)_{|X_{LS},Y_{LS}}\right]$$

$$\begin{cases} a(X_i) = E_{X_{-i}}\left[\hat{Y}(X)/X_i\right] \\ S_i = \dfrac{Var_{X_i}[E_{X_{-i}}(\hat{Y}(X)/X_i)]}{Var_X(\hat{Y})} \end{cases}$$

☠ **Do not infer from metamodel error**

From full GP model $Y(X,\omega)_{|X_{LS},Y_{LS}}$

$$\begin{cases} a(X_i,\omega) = E_{X_{-i}}\left[Y(X,\omega)_{|X_{LS},Y_{LS}}/X_i\right] \\ V_i(\omega) = Var_{X_i}\left[E_{X_{-i}}\left(Y(X,\omega)_{|X_{LS},Y_{LS}}/X_i\right)\right] \end{cases}$$

$a(X_i,\omega)$: stochastic process of X_i

V_i : random variables

$$\tilde{S}_i = \frac{V_i}{E_{\Omega}Var_X Y(X,\omega)_{|X_{LS},Y_{LS}}} \Rightarrow \mu_i = E_{\Omega}(\tilde{S}_i)$$

Analytical sensitivity indices from full GP model

[Oakley & O'Hagan 2004; Marrel et al. 2009]

- Sobol indices:

$$\mu_i = \frac{\mathrm{E}_\Omega\left(\mathrm{Var}_{X_i}\left[\mathrm{E}_{X_{-i}}\left(Y_{|X_{LS},Y_{LS}}(X,\omega)|X_i\right)\right]\right)}{\mathrm{E}_\Omega \mathrm{Var}_X Y(X,\omega)_{|X_{LS},Y_{LS}}}$$

$$\sigma_i^2 = \frac{\mathrm{Var}_\Omega\left(\mathrm{Var}_{X_i}\left[\mathrm{E}_{X_{-i}}\left(Y_{|X_{LS},Y_{LS}}(X,\omega)|X_i\right)\right]\right)}{\mathrm{E}_\Omega\left[\mathrm{Var}_X Y(X,\omega)_{|X_{LS},Y_{LS}}\right]^2}$$

☺ Infer from the metamodel error

- Computation:
  - Analytical calculations
  - Simple and double numerical integrals => **computationally expensive** ☠
- Other limits:
  - Additional hypothesis : GP covariance is product of one-dimensional covariance
  - This estimator is not the "true" expectation of the Sobol' index
  - Not possible to estimate the total Sobol' indices

GP-simulation based Sobol' indices (1/2)

[Le Gratiet et al. 2014]

$$S_i^{\omega} = \frac{\mathrm{Var}_{X_i}\left[\mathrm{E}_{X_{-i}}\left(Y_{|X_{LS},Y_{LS}}(X,\omega)|X_i\right)\right]}{\mathrm{Var}_X Y(X,\omega)_{|X_{LS},Y_{LS}}}$$ is a random variable

We want to build an unbiased Monte Carlo estimator $\hat{S}_{i,m}$ where *m* represents the number of Monte Carlo particles

and an estimator of the variance of $\hat{S}_{i,m}$

<u>Algorithm:</u>

1. Generate 2 *m* -size $\perp$ matrices (X_1, X_2) and compute the pick-freeze matrix X (the one which used in Sobol' indices estimation formula)

2. For k=1, …, nsim
   - Perform a conditional simulation y^*_k of the GP at points in X
   - Compute $\hat{S}_{i,m,k}$
   - Bootstrap procedure (*b* = 1, …, *B* replicas) gives $\hat{S}_{i,m,k,b}$

GP-simulation based Sobol' indices (2/2)

[Le Gratiet et al. 2014]

The estimator of Sobol' index: $$\hat{S}_{i,m} = \frac{1}{nsim \times B} \sum_{k=1}^{nsim} \sum_{b=1}^{B} \hat{S}_{i,m,k,b}$$

The variance of this estimator: $$\hat{\sigma}_i^2 = \frac{1}{nsim \times (B-1)} \sum_{k=1}^{nsim} \sum_{b=1}^{B} \left(\hat{S}_{i,m,k,b} - \hat{S}_{i,m} \right)$$

It integrates the metamodeling error + the Monte Carlo integration error

$$\hat{\sigma}_i^2(\mathrm{PG}) = \frac{1}{B} \sum_{b=1}^{B} \left[\frac{1}{nsim-1} \sum_{k=1}^{nsim} \left(\hat{S}_{i,m,k,b} - \overline{S}_{i,m,b} \right) \right]$$

$$\hat{\sigma}_i^2(\mathrm{MC}) = \frac{1}{nsim} \sum_{k=1}^{nsim} \left[\frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{S}_{i,m,k,b} - \overline{S}_{i,m,k} \right) \right]$$

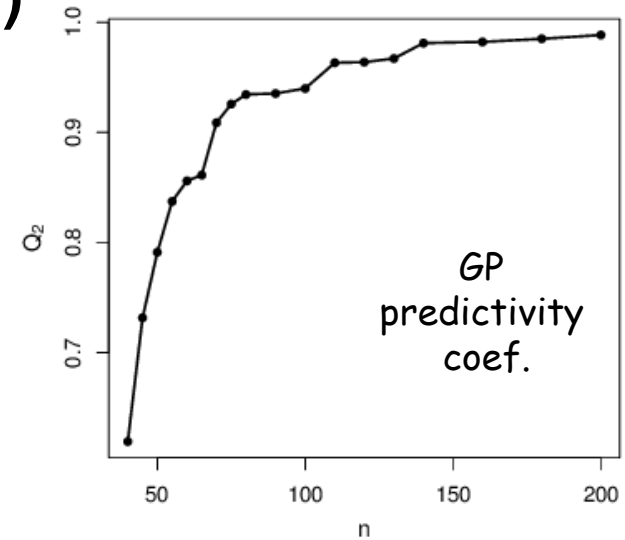
<u>Remark:</u> Same algorithm for total Sobol' indices

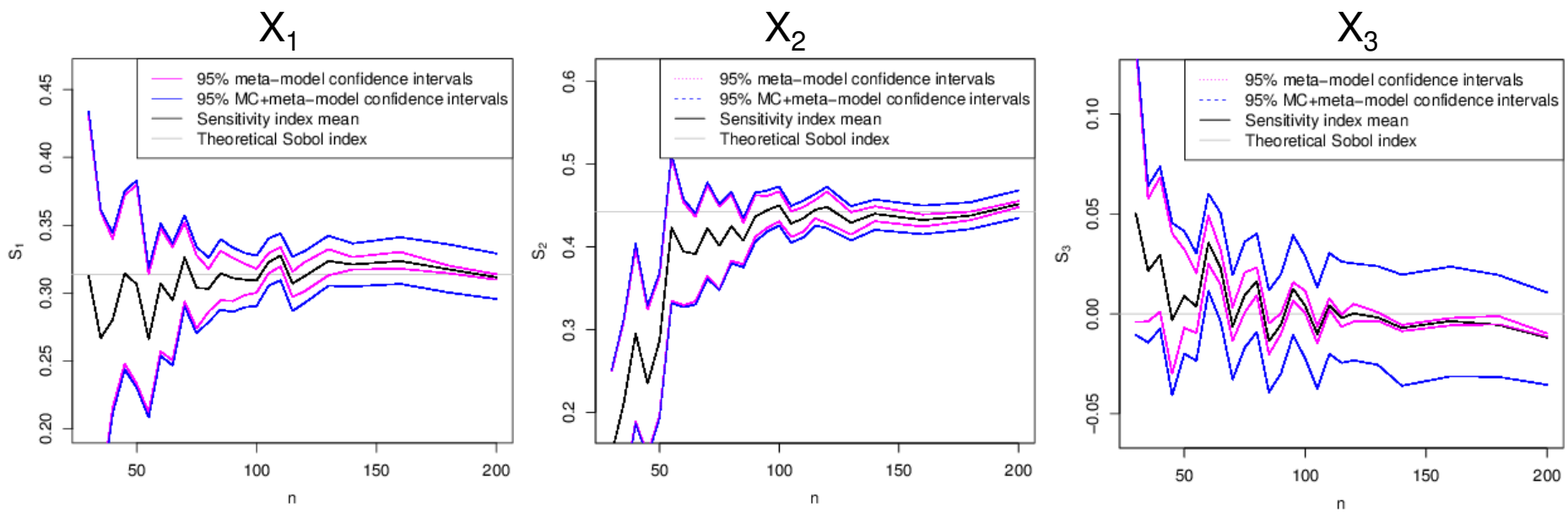
Example: Ishigami function (K = 3)

Learning sample: space filling design (n = 40 to 200)

Gaussian process with 5/2 Matérn cov.

m = 10000
$nsim$ = 500
B = 300

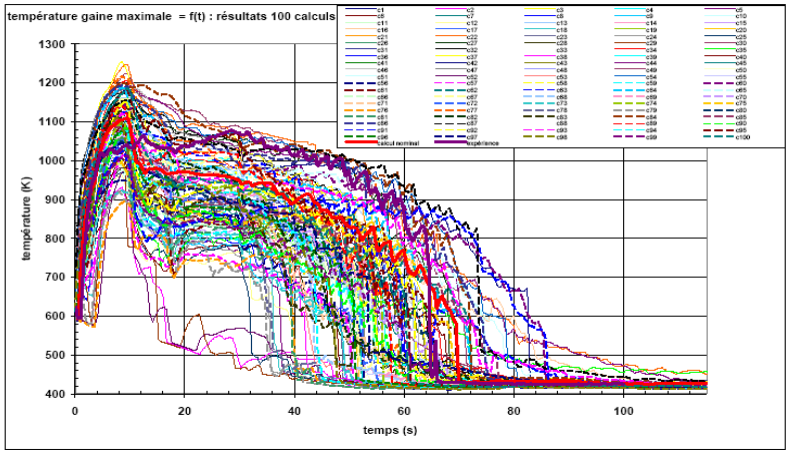




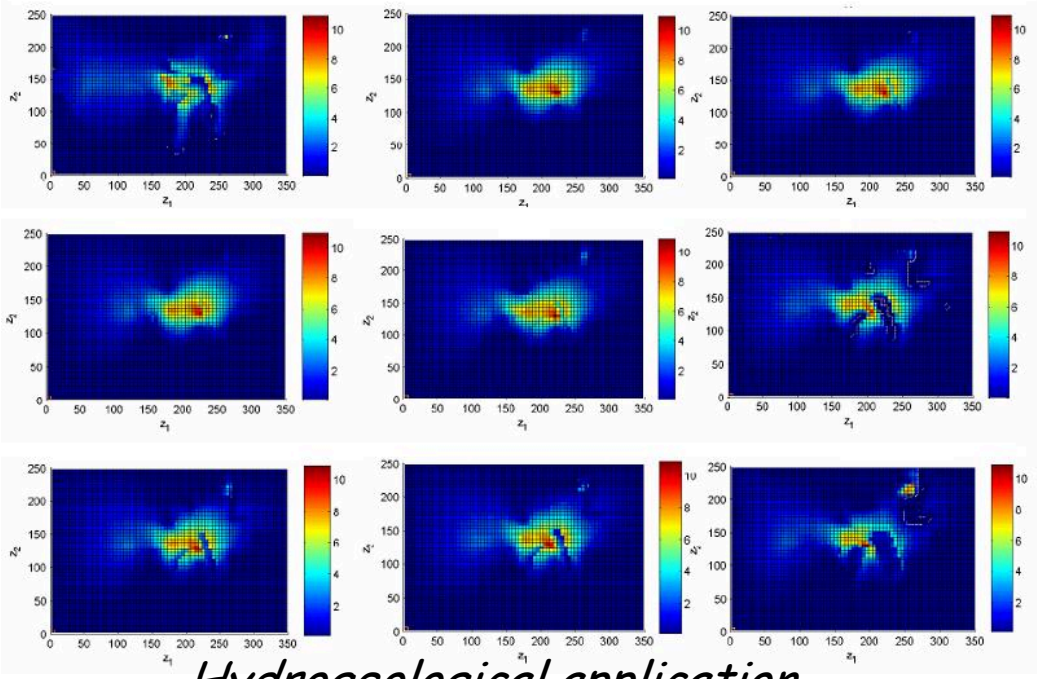
In practice, n is fixed and m can be calibrated in order to balance the error

Spatial output

Sensitivity analysis when model ouputs are functions



Thermal-hydraulic example



Hydrogeological application

<u>2 elementary cases</u>: sensitivity analysis on each scalar output (*q* pixels):

- Very small CPU time consuming model
- Linear regression model => use of standardized regression coef. (SRC)

<u>Difficult case:</u>

- Complex/Non linear model ➡ need of Sobol indices (for example)
- and CPU time expensive model ➡ need of metamodel

Sensitivity analysis for spatial outputs: Methodology

- Computer code $f(.)$: *[Marrel et al. 2011]*

  Input: $\mathbf{X} = (X_1, \ldots, X_p)$ random vector

  Output for input $\mathbf{x}^*$: $y = f(\mathbf{x}^*, \mathbf{z})$, $\mathbf{z} \in D_{\mathbf{z}} \subset \mathrm{R}^2$

In practice, $D_{\mathbf{z}}$ is discretized in q points (here: 64 x 64 = 4096 points)

$(\mathbf{X}, Y(\mathbf{X},\mathbf{z}))$ = input-output sample of size N

- **Decomposition of $Y(\mathbf{z})$ on a functional basis**, for example:
  - Principal component analysis PCA) gives $\perp$ functions fitted on the data,
  - A wavelet basis is well-suited if there are discontinuities

- **Modeling of the decomposition coefficients by a GP metamodel**

  Selection procedure of the most important coefficients

- **Prediction: $\mathbf{x}^*$** => prediction of coeff. => spatial output map reconstruction

Functional metamodel

Sensitivity analysis :

Spatial maps of sensitivity indices

Metamodel fitting: Methodology for spatial output (map)

- **Step 1 : Decomposition in basis functions of each map**

- **Step 2 : Kriging metamodeling of the main coefficients (the most variables) in fonction of X ; constant for the other coef.**

Computational challenge: we have to fit as many GP as main coefficients

- **Step 3 : Prediction for a new input $\mathbf{x}^*$**

  $\mathbf{x}^*$ => prediction of the coefficients => spatial output map reconstruction

- **Sensitivity analysis (spatial map of Sobol' indices)**

The estimation of Sobol indices require a large number of simulations

⇒ computation of the Sobol indices by the way of the metamodel

Computational challenge to manage thousand of calls of the functional metamodel

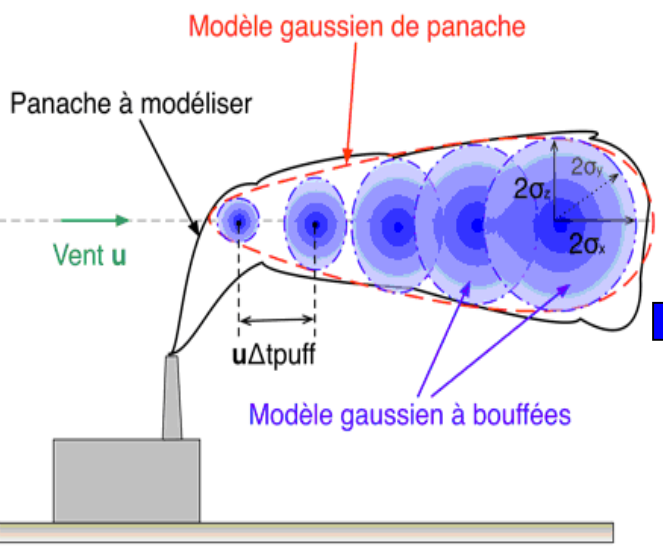
Description of CERES-MITHRA test case

ANR Costa Brava

Complex spatio-temporal dynamics analysis by model reduction and sensitivity analysis (2010-2014)

Assessment of the consequences on human health of radionuclide accidental or routine releases (atmospheric releases)

Modelling of radionuclide atmospheric dispersion based on Gaussian puff model



Developpement of an atmospheric dispersion code : CERES-MITHRA (C-M)
⇒ impact calculations relative to CEA facilities

[CEA application: Marrel & Perot, 2012]

Description of the scenario

C-M INPUTS

Release parameters

- 2 release locations (uncertain heights)
- Radionuclide quantity (uncertain)
- Deposition velocity (uncertain)
- Release duration (1h)
- Radionucleide: Cesium 137

Meteorological parameters

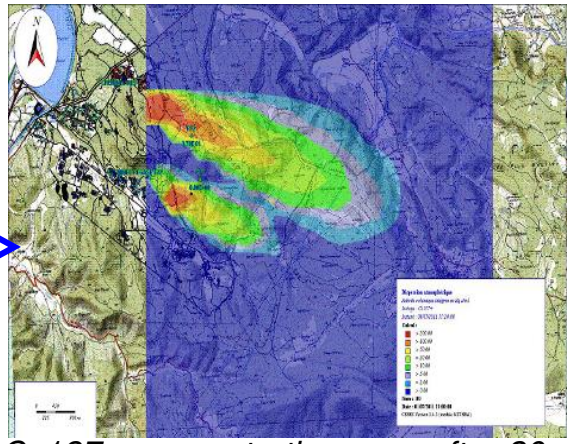
- Wind speed & direction (uncertain)
- Atmospheric stability
- Rain
- Temperature

C-M Simulator

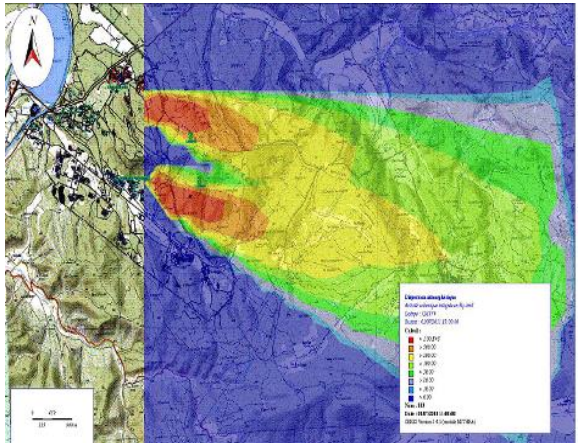
CPU time from 30s to 30min

C-M OUTPUTS

2D maps of time integrated activity concentrations (Bqsm^{-3}) for different instants (=> 2 hours)



Cs137+ concentration map after 20 minutes



Cs137+ concentration map at the end of the release

Objectives

Identify the influence of each uncertain parameters

Problems

- CPU time required for each simulation ⇒ limited number of available simulations
- Spatio-temporal outputs (very large dimension of model outputs)

Uncertainty quantification

➢ **Uncertain parameters** ($p = 6$)

Parameters		Reference Value	Variation Interval	Probability distribution
Release heigth (m)	A	15	[7.5 ; 22.5]	Uniform
	B	45	[22.5 ; 67.5]	Uniform
Deposition velocity ($m.s^{-1}$)	Cs137+	5.10^{-3}	[5.10^{-4} ; 5.10^{-2}]	Log-Uniform
Quantity released (Bq)	Cs137+	10^{9}	[10^{8} ; 10^{10}]	Log-Uniform
Wind direction (WD) (degree azimuth)		291	[249 ; 333]	Truncated AR(1) process
Wind speed (WS) ($m.s^{-1}$)		5.3	[0 ; 12.5]	Truncated AR(1) process

Experimental design and learning sample

Numerical experimental design :

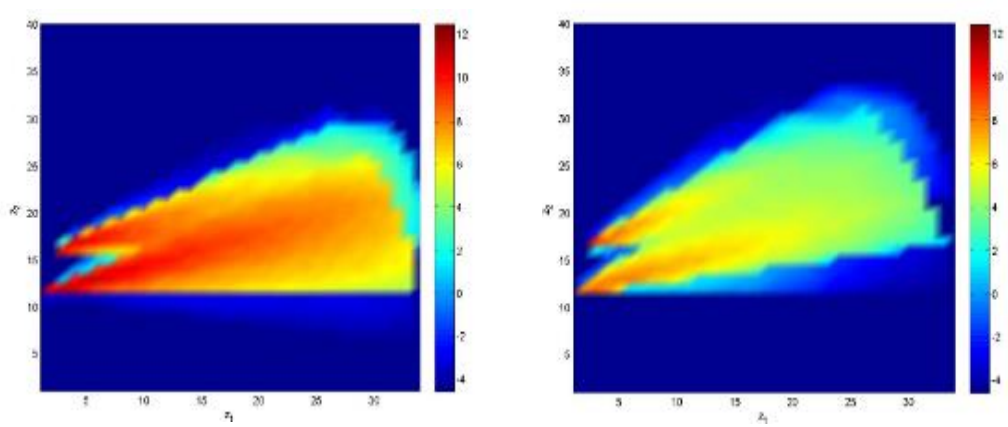
Space filling design in order to have a good coverage of the input space

=> Optimized Latin hypercube Sampling (LHS)

Number of C-M simulations : ***n = 200* simulations**

Compromise between CPU time and investigation of uncertain parameter domain

Simulations with C-M : Learning sample creation



Examples of Cs137+ integrated activity maps (logarithmic scale)

- **Notations**

$X_s = (x^{(i)})_{i=1,\cdots,n}$ learning sample

$Y^{(i)}(z) = y(x^{(i)}, z)_{i=1,\cdots n}$ C-M outputs

(*q = 4000 pixels*)

Approximation by a metamodel

Spatial metamodel : Proper Orthogonal Decomposition + Gaussian Process

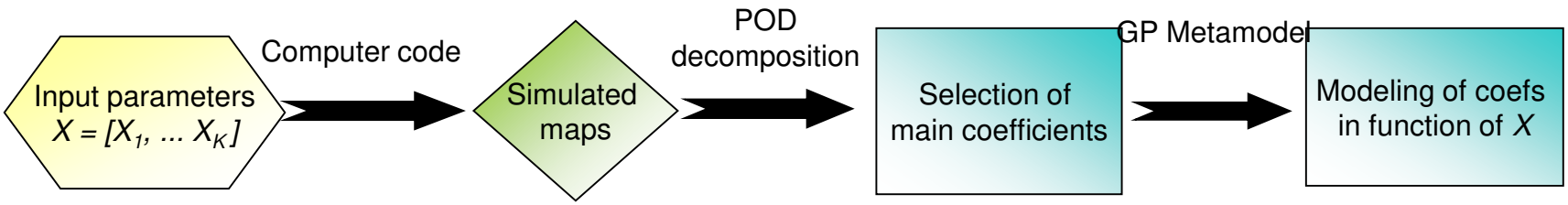
Step A : Spatial decomposition on a functional basis (Chatterjee [2000])

⇒ Selection of the *h* main coefficients of the decomposition

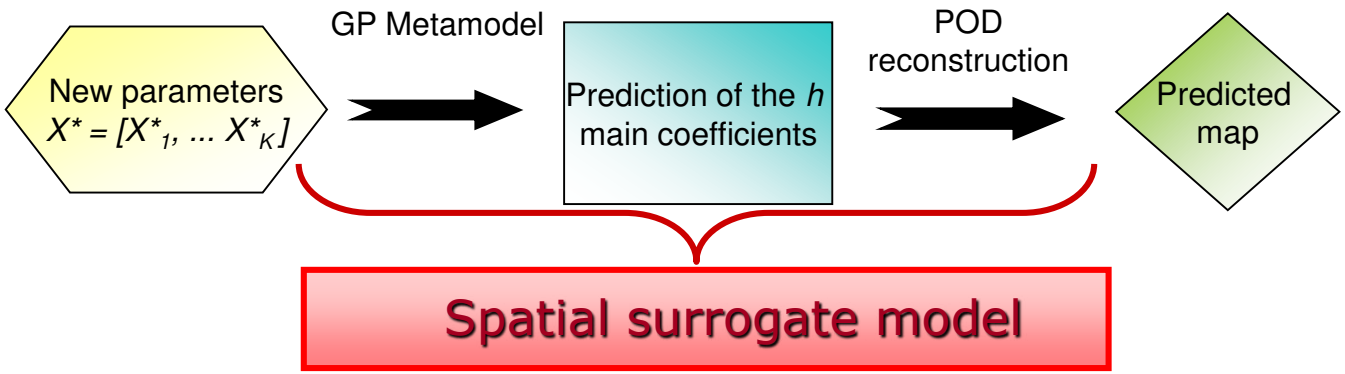
$$Y_h(X,z) = \mu(z) + \sum_{j=1}^{h} \alpha_j(X)\phi_j(z) \quad \text{with} \quad \alpha_j(X) = \int_D \left(Y(X,z) - \mu(z)\right)\phi_j(z)dz$$

Step B : Modeling of the *k* main PCA coefficients in function of X

⇒ Gaussian Process metamodel



Step C : Prediction for any new input x*



Basic principle of principal component analysis (PCA)

Transformation of dependent **<u>numerical</u>** variables to **independent variabl**es

→ information reduction

PCA: sequential search of new variables = **linear combinations** of Y_i with a **maximal dispersion** (variance or inertia) on each axis

$\mathbf{V} = {}^t\mathbf{Y_c}\mathbf{Y_c}$ variance-covariance matrix of $\mathbf{Y_c}$ (matrix of centered functions)

Diagonalisation of the variance-covariance matrix to obtain ordered eigen vectors (inertia axes)

Eigen values are principal component variances and give the total inertia (percentage of explained variance)

The matrix (N x q) of principal component is $\mathbf{H} = \mathbf{Y_c}\,\mathbf{L}$

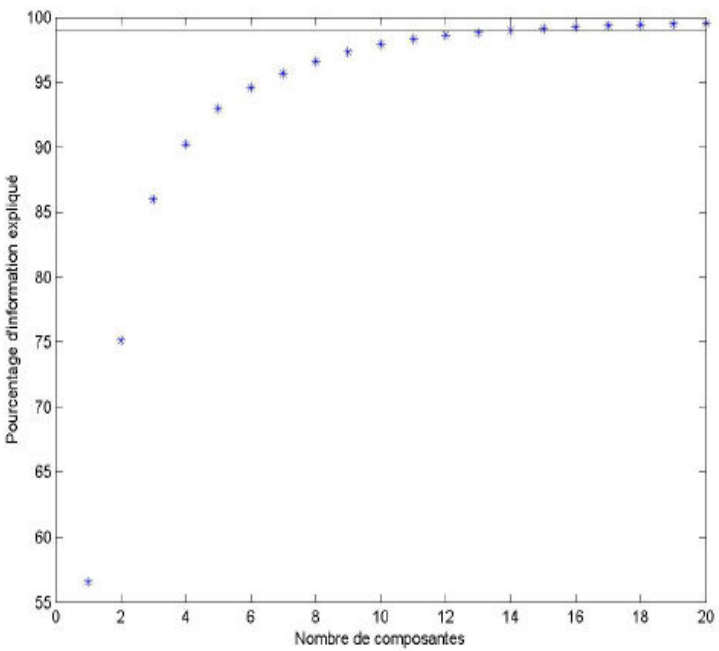
with L the eigen vector matrix

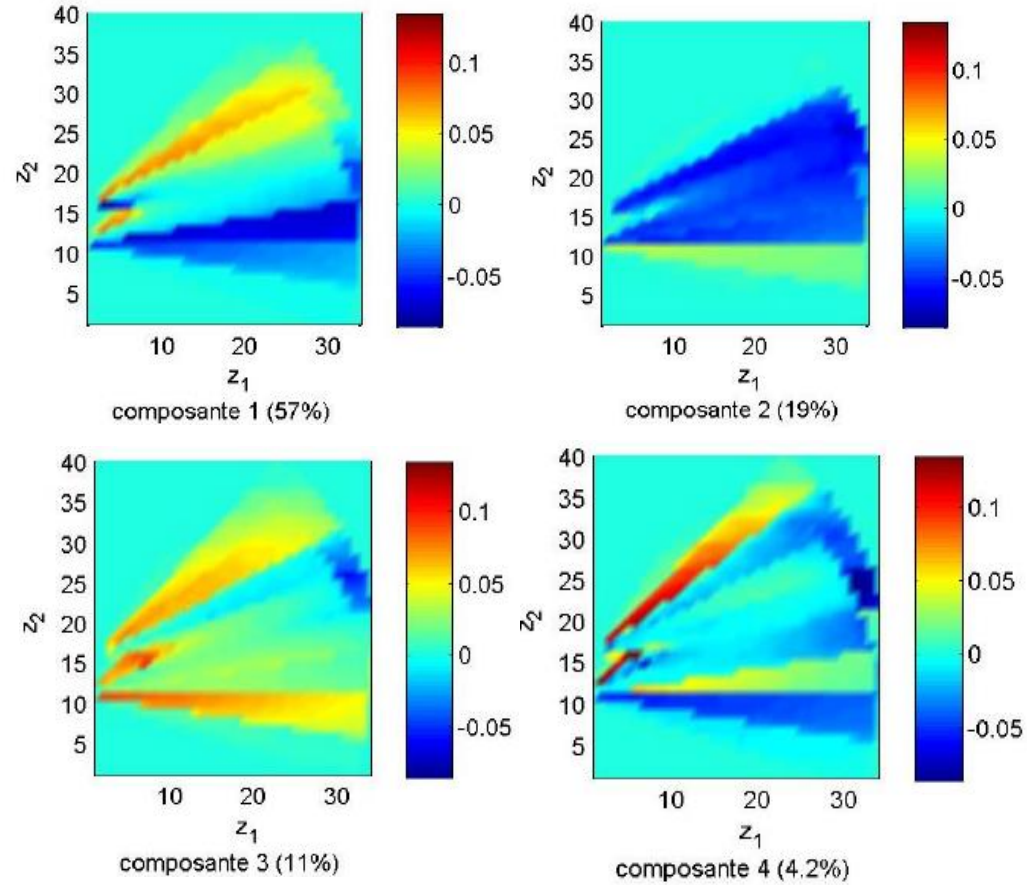
<u>Remark:</u> data adaptive basis while the wavelet bases are fixed

Selection of PCA coefficients

➢ **Application to C-M data & results :**

- ***h*** **= 15 selected PCA components = 99% of information explained**





⇒ **15 coefficients to be modelled by GP metamodels**

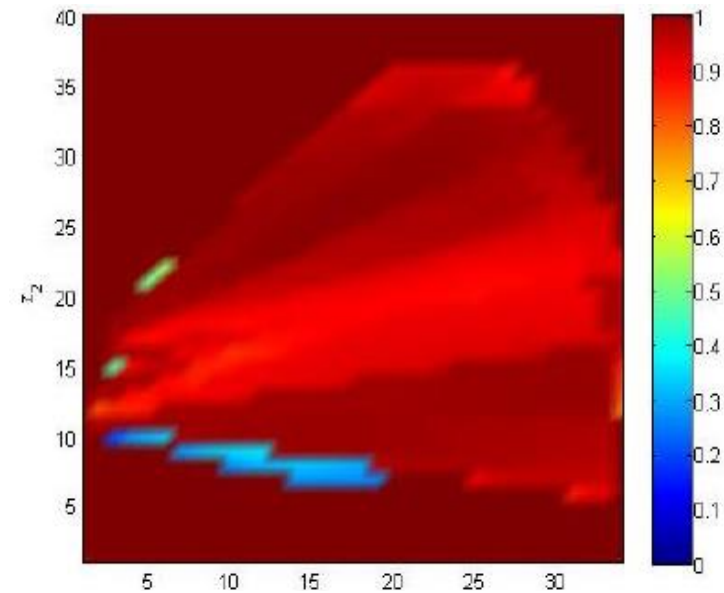
Validation of the spatial metamodel accuracy

❖ **Predictivity coefficient :**

Q² estimated by cross validation

$$Q^2(z) = 1 - \frac{\sum_{i=1}^{n}\left[\hat{Y}\left(x^{(i)}, z\right) - Y^{(i)}(z)\right]^2}{\sum_{i=1}^{n}\left[Y^{(i)}(z) - \frac{1}{n}\sum_{i=1}^{n} Y^{(i)}(z)\right]^2}$$

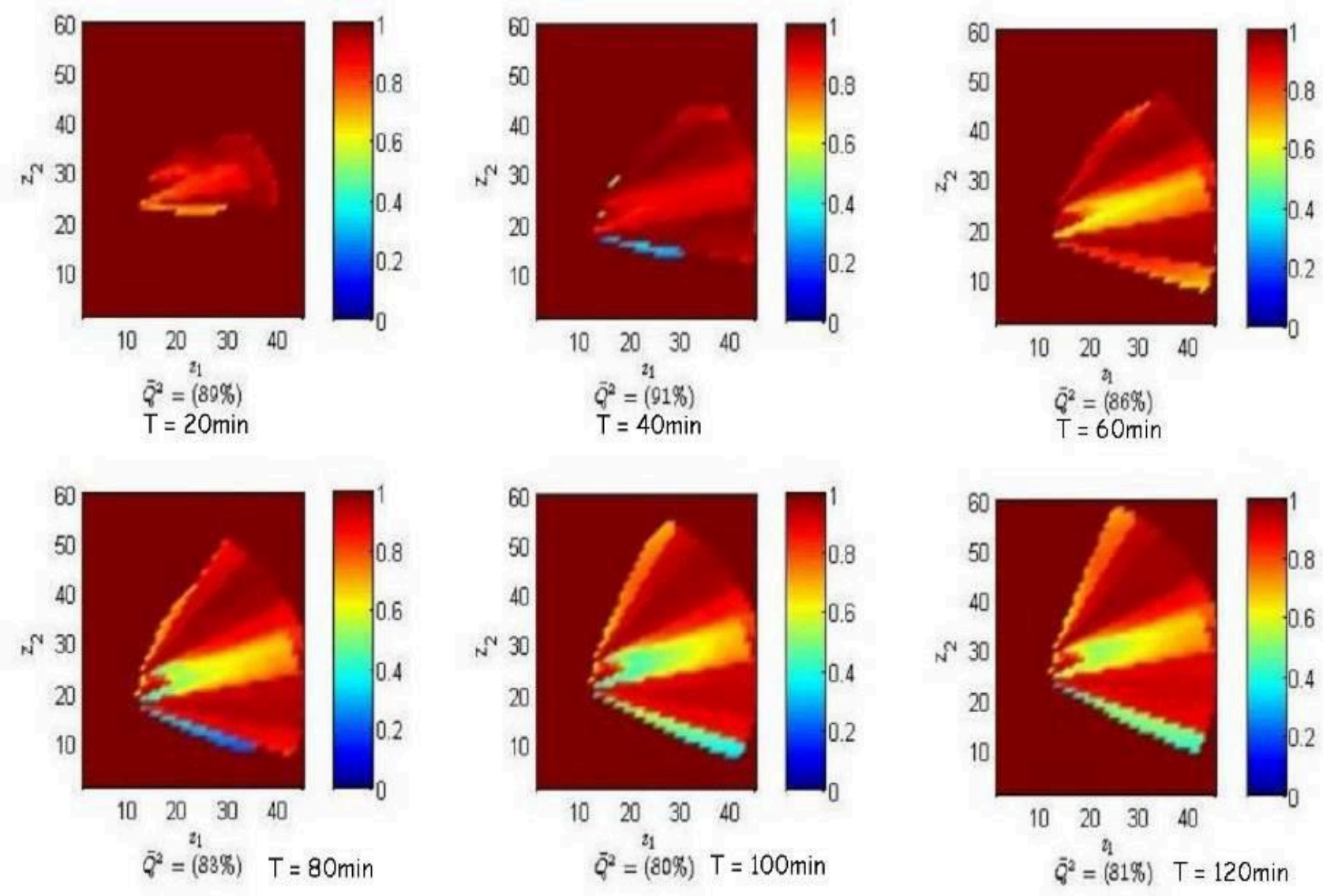
- **Average value of Q^2 on the map ≈ 92 %**



Q² map with 15 PCA components modelled with GP, for *t = 40min*

❖ **Global and local analysis of the metamodel predictivity**

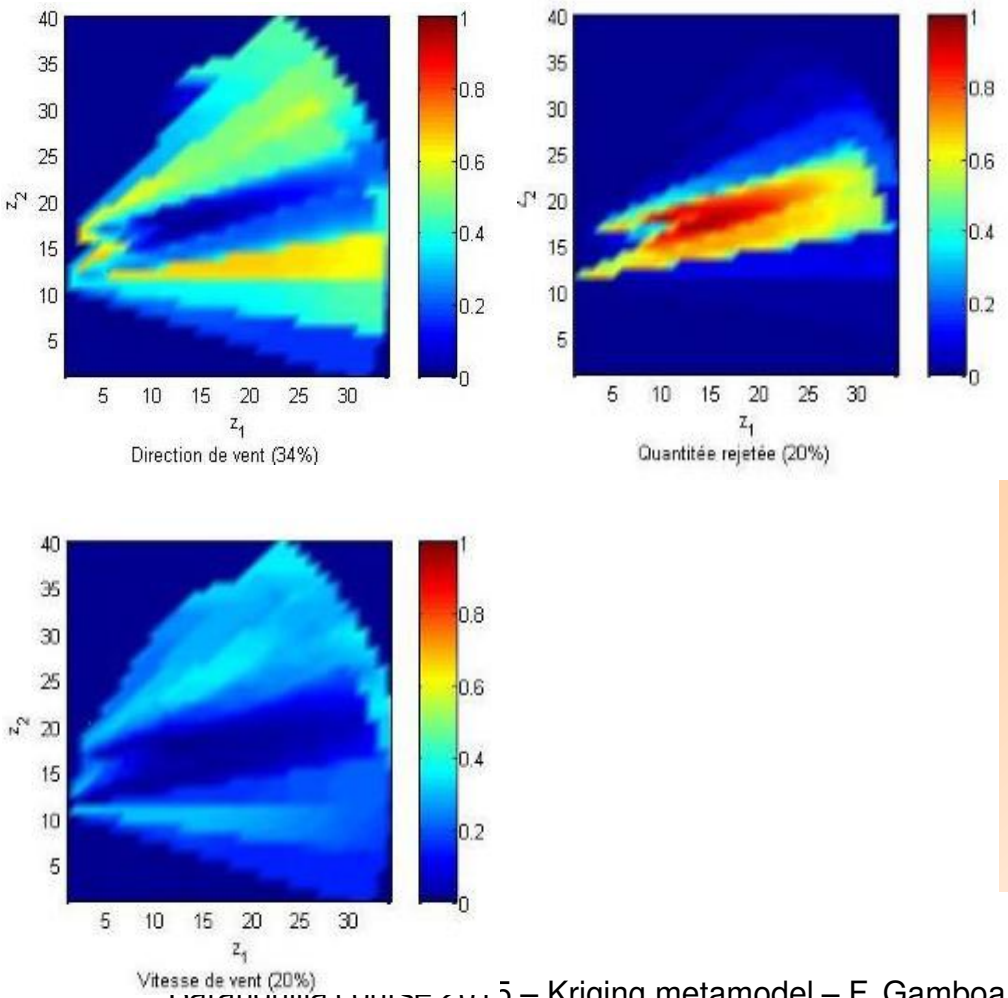
Metamodel quality for the various moments



Sensitivity Analysis

Use of the metamodel POD+GP to perform 10000 simulations required for Sobol indices estimations (RBD-FAST method)

First order Sobol indices for *t = 40min*



Input influence :

Predominant :

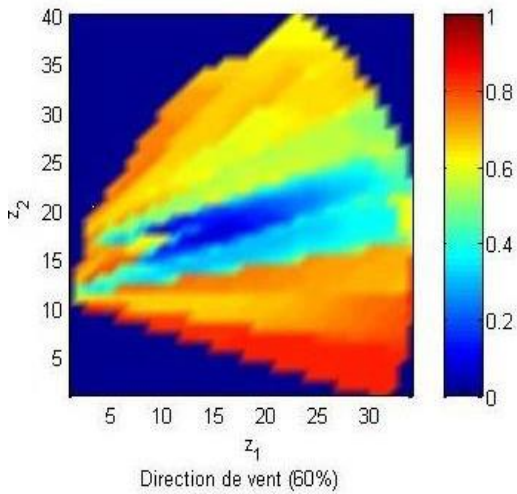
- **Wind direction**
- **Release quantity** : located in the center of the plume
- **Wind speed** : located at the edges of the plume

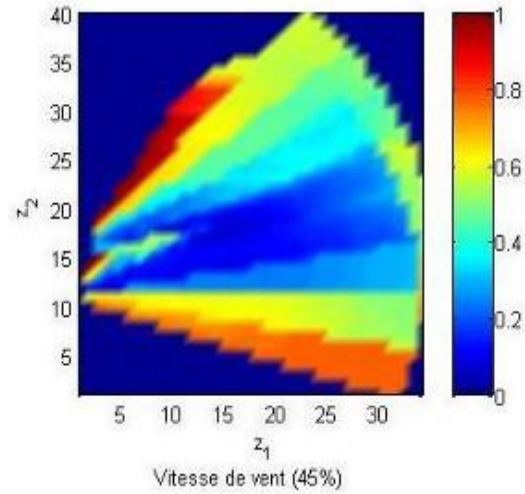
Negligible : deposition velocity, release heights

Sum of 1er order indices ≈ 75%

Interactions : ≈ 25% of model variablility

Total Sobol indices for t=40mn





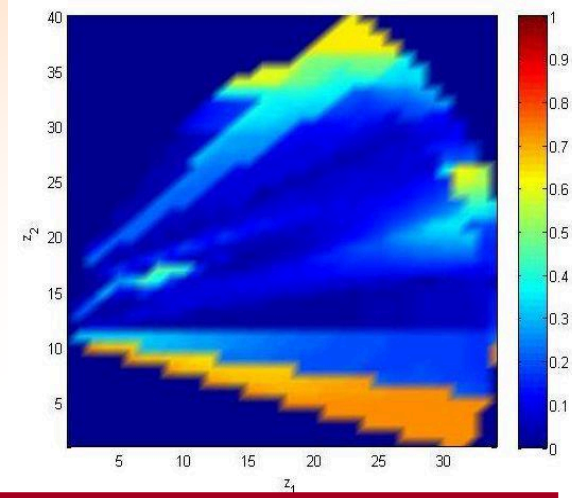
<u>Differences between 1st order and total indices :</u>

- An average of 25% of difference for wind direction and speed

=> Strong interaction Wind direction x Wind speed?

=> 2nd order indices:

(located on the edges of the dispersion plume)

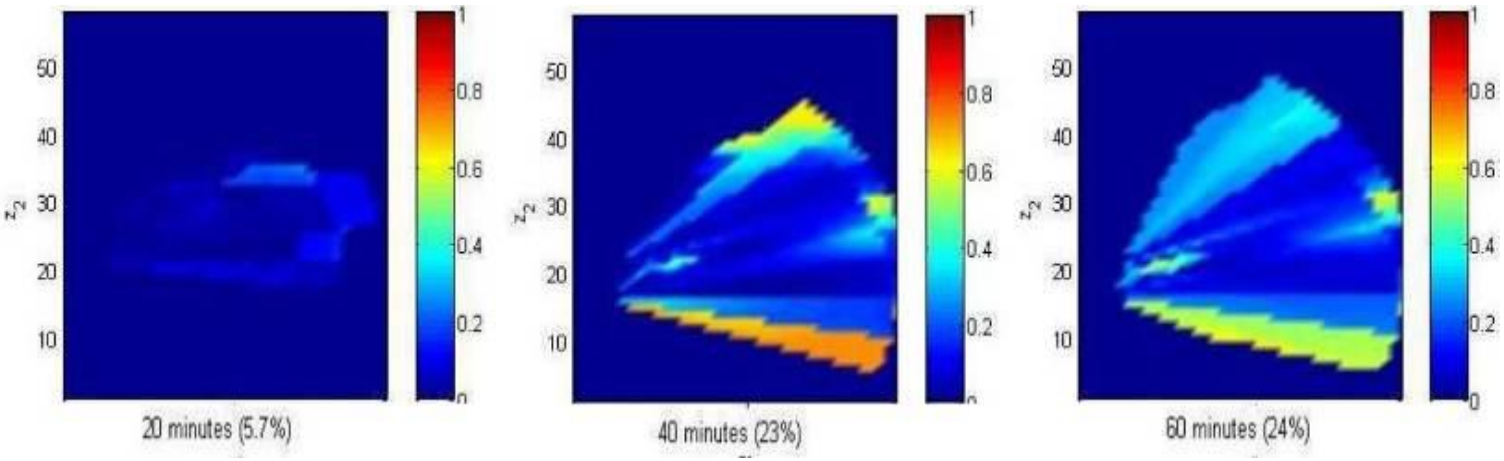


<u>CONCLUSION</u> : An average of 97% of the C-M output variance explained by :

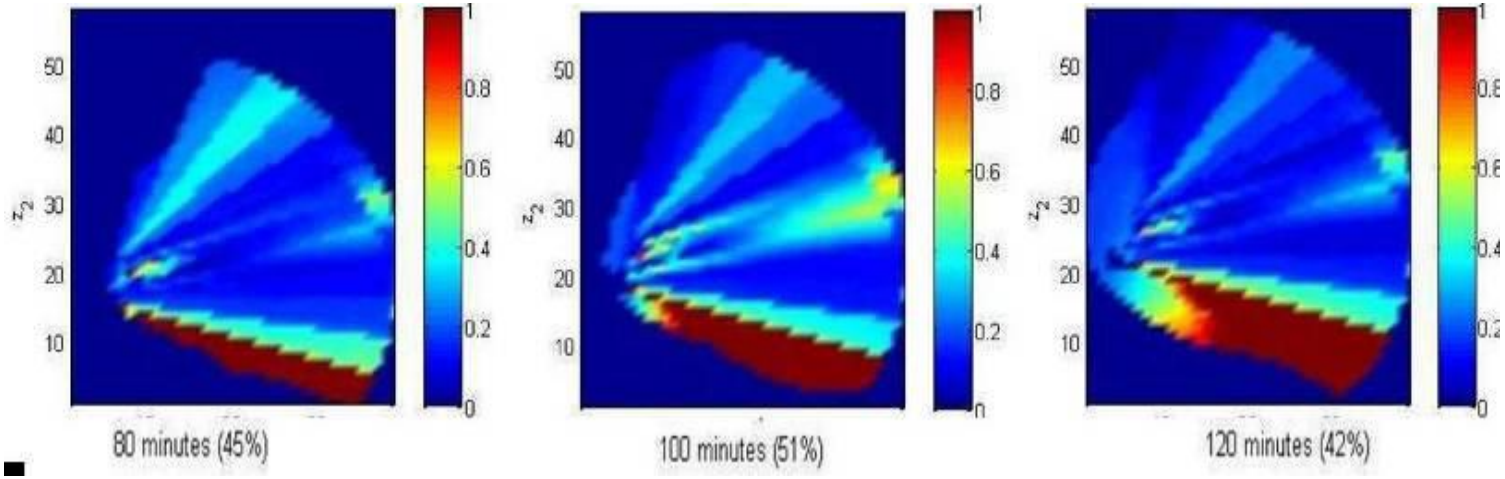
Wind Direction (34%) - Wind Direction x Wind Speed (23%)

Cs137+ released quantity (20%) - Wind Speed (20%)

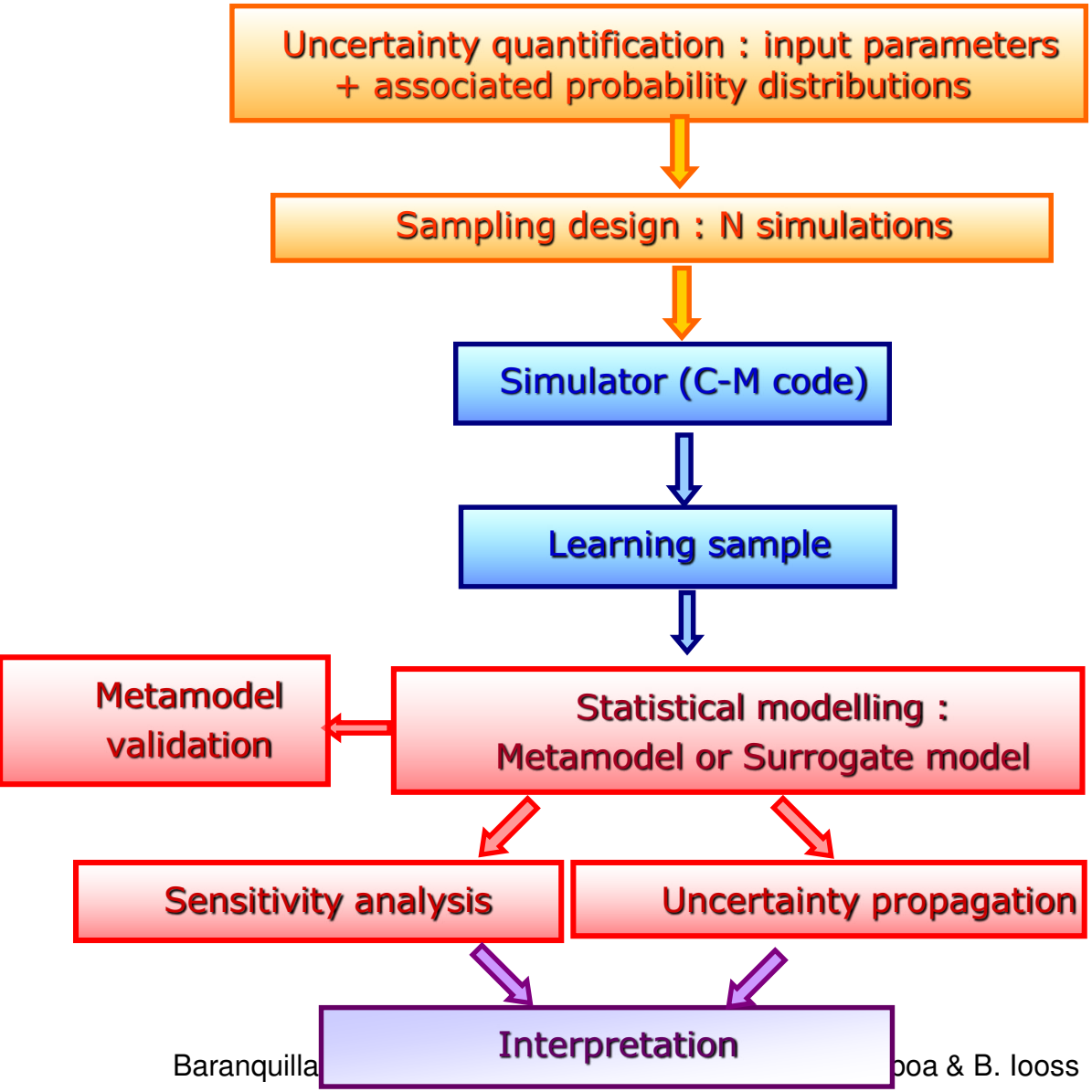
Time evolution of Sobol indices



Significant augmentation of interaction wind direction x wind speed influence:
Less than 10% at the release beginning and more than 40% at the end



Conclusion: proposed methodology



- **Uncertainty quantification :**

  Construction of a realistic simulator of weather conditions

- **Sampling design :**

  Latin Hypercube Sample (LHS) with optimal recovering properties

- **Metamodel :**

  Spatial decomposition with PCA

  +

  metamodelling of PCA coefficients with Gaussian processes

- **Sensitivity analysis :**

  Computation of Sobol indices

Conclusion

On the Gaussian process metamodel

- **Valuable tool when computer code is cpu-time expensive** (n ~ hundreds runs)

- GP model construction is possible for moderate dimensional cas ($p < 50$)

- Main advantage of GP: probabilistic metamodel which gives confidence bands in addition to a predictor

  Full interest in sensitivity analysis

- Fitting quality is dependent of the initial design

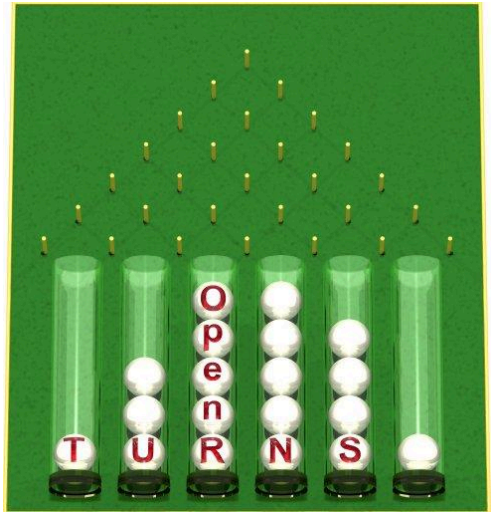
  GP model is well adapted to sequential and adaptative designs

- <u>Caveats:</u> it can require a large amount of effort during the fitting process and cases with more than 1000 points begin to be difficult (matrix inversion)

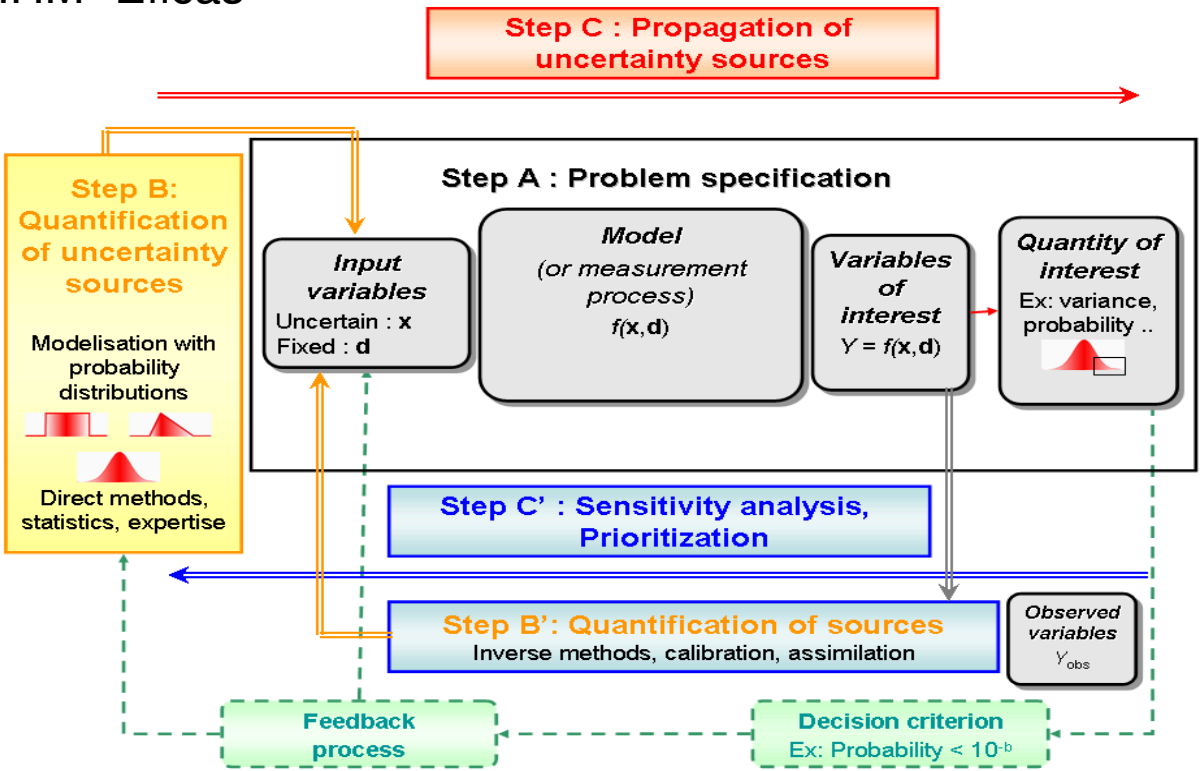
- Designs for specific objectives (optimization, quantile, probability, etc.)

- **Other hot topics: calibration and validation of computer codes**

Open TURNS – The software implementation of the uncertainty methodology



- TURNS : Treatments of Uncertainties, Risk'n Statistics
- Since 2005
- Open : Open source : LGPL (code), FDL (doc.)
  - http://www.openturns.org/
  - Environment : Linux, Windows
  - Languages : C++ (libraries), Python (command scripts)
  - IHM "Eficas"



Main features in OpenTURNS

- Use of non-intrusive techniques ➡ generic

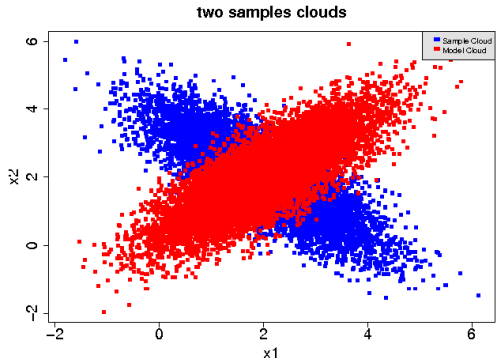
- Wrapping of external codes, considered as mathematical functions in OpenTurns

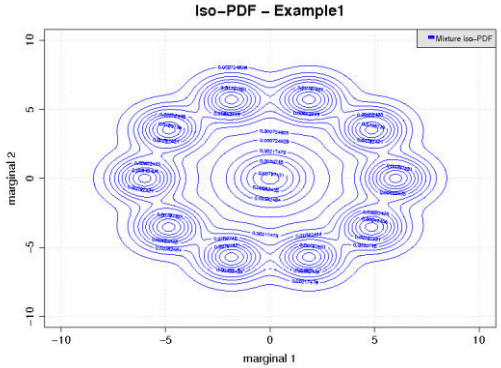
- Step B: modeling uncertainty of inputs
  - With data: parametric & non-parametric stats
  - Assessing experts' advice
  - Dependence: definition by marginals + copula

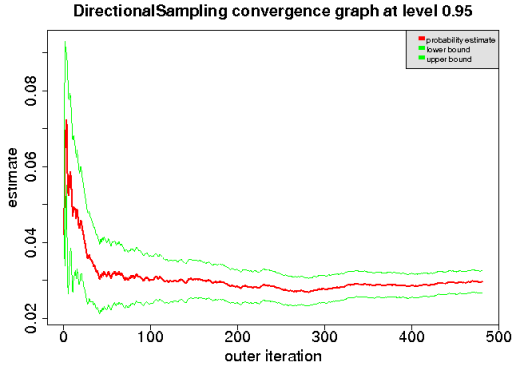
- Step C: Uncertainty propagation
  - Standard and advanced Monte Carlo techniques (Importance sampling, directional sampling …)
  - FORM-SORM method

- Metamodels: Polynomials, polynomial chaos expansion, Gaussian processw

- Step C': Sensitivity analysis - Linear & rank regression, Sobol, polynomial chaos, reliability importance factors







Bibliography

Books

- Chilès & Delfiner, Geostatistics, Wiley, 1999
- Fang, Li & Sudjianto, D*esign and modeling for computer experiments,* Chapman, 2006

Metamodels

Simpson, Peplinski, Koci & Allen, Metamodels for computer-based engineering design: Survey and recommendations, *Engineering with computers,* 17, 2001

Storlie, Swiler, Helton & Salaberry, Implementation and evaluation of nonparametric regression procesures for sensitivity analysis of computationally …, *RESS*, 94, 2009

Gaussian process

Koehler & Owen, Computer experiments, In Handbook of Statistics (Ghosh & Rao), 13, 1996

Marrel, Iooss, Van Dorpe & Volkova. An efficient methodology for modeling complex computer codes with Gaussian processes. *Computational Stat. and Data Analysis*, 52, 2008

Le Gratiet, Cannamela, Iooss. A Bayesian approach for global sensitivity analysis of (multifidelity) computer codes. *SIAM/ASA Journal of Uncertainty Quantification,* 2014