Sensitivity and Uncertainty Analysis for computer code experiments a first tour

Fabrice Gamboa-Bertrand Iooss

Institut de Mathématiques de Toulouse

18th-23th of March 2013

Agenda

The small story

Early 2000, IMT Toulouse begin to work with many Labs :

- **CEA Cadarache. N. Devictor, B. Iooss. Nuclear safety (Ph D A. Marrel-currently CEA-)**
- **ONERA-DOTA Palaiseau. G. Durand, A. Roblin. infrared profile of a plane (Ph D S. Varet)**
- **IFP Lyon. P. Duchène, F. Wahl-Univ Grenoble. A Antoniadis. Chemical cinetic problems (Ph D S. Da Veiga-Currently IFP-)**

⇒ Scientific meetings in Toulouse- Février 2006 and in Lyon en 2007 GDR CNRS borned

What are we dealing with?

Big computer codes= F **black box**

$$Y = F(X)$$

- Code inputs: X high dimension object (vectors or curves).
- Code outputs Y (scalar or vectorial).

X complex structure and/or uncertain

$\Rightarrow$ **seen as random**

STOCHASTIC APPROACH

Some questions on the general model

- **Sensitivity and uncertainty analysis= take informations on the joint distribution (X, Y)**
- **F too complicated. Design a reduced model= Estimate a response surface**
- **Optimise the run number= make an experimental design**

People working around this topic

- **GDR MASCOT NUM Annual meeting march 2012 : CEA Bruyères le Chatel** http://www.gdr-mascotnum.fr/
- **ANR project : OPUS EADS, CEA, EDF, ... (CEA)** http://www.opus-project.fr/
- **ANR project : COSTA BRAVA CEA, IFP, Univ Toulouse, Univ Grenoble** http://www.math.univ-toulouse.fr/COSTA_BRAVA/doku.php?id=index
- **SIAM conference : Uncertainty quantification 2th-5Th April 2012** http://www.siam.org/meetings/uq12

Motivation-Examples

Motivation-Examples

- **Main object : complicated computer simulation code**

Motivation-Examples

- Main object : complicated computer simulation code
- **Examples: Meteo, Oceanography, Complex physical or chemical process, Economics evolutions ....**

Motivation-Examples

- Main object : complicated computer simulation code
- Examples: Meteo, Oceanography, Complex physical or chemical process, Economics evolutions ....
- **Complexity:**

Motivation-Examples

- Main object : complicated computer simulation code
- Examples: Meteo, Oceanography, Complex physical or chemical process, Economics evolutions ....
- Complexity:
  - Big Code: many different numerical methods elaborated during a large time.

Motivation-Examples

- Main object : complicated computer simulation code
- Examples: Meteo, Oceanography, Complex physical or chemical process, Economics evolutions ....
- Complexity:
  - Big Code: many different numerical methods elaborated during a large time.
  - Many input: vectorial, functional, uncertain.

Motivation-Examples

- Main object : complicated computer simulation code
- Examples: Meteo, Oceanography, Complex physical or chemical process, Economics evolutions ....
- Complexity:
  - Big Code: many different numerical methods elaborated during a large time.
  - Many input: vectorial, functional, uncertain.
  - Many output: vectorial, functional.

Motivation-Examples

- Main object : complicated computer simulation code
- Examples: Meteo, Oceanography, Complex physical or chemical process, Economics evolutions ....
- Complexity:
  - Big Code: many different numerical methods elaborated during a large time.
  - Many input: vectorial, functional, uncertain.
  - Many output: vectorial, functional.
  - Expensive: from some minutes up to several days

Motivation-Examples

- Main object : complicated computer simulation code
- Examples: Meteo, Oceanography, Complex physical or chemical process, Economics evolutions ....
- Complexity:
  - Big Code: many different numerical methods elaborated during a large time.
  - Many input: vectorial, functional, uncertain.
  - Many output: vectorial, functional.
  - Expensive: from some minutes up to several days
  - Example functional code CERES from CEA ▸ CERES

Motivation-Examples

- Main object : complicated computer simulation code
- Examples: Meteo, Oceanography, Complex physical or chemical process, Economics evolutions ....
- Complexity:
  - Big Code: many different numerical methods elaborated during a large time.
  - Many input: vectorial, functional, uncertain.
  - Many output: vectorial, functional.
  - Expensive: from some minutes up to several days
  - Example functional code CERES from CEA ▸ CERES
- Need methods to enlight.

Frame: Black box

Frame: Black box

Today conference: vectorial input et scalar output

Y =output is a number and X =input is a vector of numbers

Frame: Black box

Today conference: vectorial input et scalar output

Y =output is a number and X =input is a vector of numbers

Black box model

Frame: Black box

Today conference: vectorial input et scalar output

Y =output is a number and X =input is a vector of numbers

Black box model

Frame: Black box

Today conference: vectorial input et scalar output

Y =output is a number and X =input is a vector of numbers

Black box model

Non linear regression model

$$Y = f(X).$$

The code is modeled as an abstract complicated function f

Gains of stochastic methods

Gains of stochastic methods

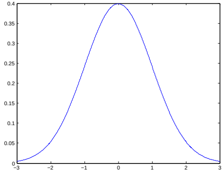
- **Take into account random characteristic of some components of X**

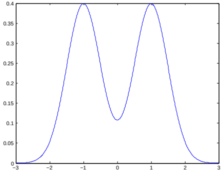
Gains of stochastic methods

- Take into account random characteristic of some components of X
  - Physical measures with error: Pressure, temperature...
  - unknown physical constants: wave in random media.

Gains of stochastic methods

- Take into account random characteristic of some components of X
  - Physical measures with error: Pressure, temperature...
  - unknown physical constants: wave in random media.
  - Allow to model multimodal distributions: double mode...





Gains of stochastic methods

- Take into account random characteristic of some components of X
  - Physical measures with error: Pressure, temperature...
  - unknown physical constants: wave in random media.
  - Allow to model multimodal distributions: double mode...
- **Parametric and non parametric estimation methods**

Presented techniques

Presented techniques

- **Sobol sensitivity analysis**

Presented techniques

- Sobol sensitivity analysis
  - Within the (random) components of the input (vector) X what are those having most influence on the output?
  - "influence" is quantified in terms of "variability" induced by this component.
  - Global analysis taking into account the whole distribution of the input.
- **Response surface methods (Reduced model)**

Presented techniques

- Sobol sensitivity analysis
  - Within the (random) components of the input (vector) X what are those having most influence on the output?
  - "influence" is quantified in terms of "variability" induced by this component.
  - Global analysis taking into account the whole distribution of the input.
- Response surface methods (Reduced model)
  - Replace the complicated code by a simple one easy to build from a short sample y cheap in CPU.
  - Goal: optimization, computation of a critical threshold, sensitivity analysis...
  - Discussed method: come from geostatistics (KRIGING).

Some links I

Some links I

- **Research/Developpement**

Some links I

- Research/Developpement
  - GDR CNRS MASCOT NUM http://www.gdr-mascotnum.fr/ ,
  - OPUS- ANR project big *open source* plateform including tools for codes.
  - COSTA BRAVA- ANR project functional input or output coupling random and deterministic methods.
- **Softwares**

Some links I

- Research/Developpement
  - GDR CNRS MASCOT NUM http://www.gdr-mascotnum.fr/ ,
  - OPUS- ANR project big *open source* plateform including tools for codes.
  - COSTA BRAVA- ANR project functional input or output coupling random and deterministic methods.
- Softwares
  - R package DICE (IRSN, EDF, Renault, ...). http://crocus.emse.fr/dice
  - MATLAB Kriging package: DACE http://www2.imm.dtu.dk/ hbn/dace/
  - Free software of O' Oakley and O' Hagan computation of sensitivity indices: GEM http://www.tonyohagan.co.uk/academic/GEM/index.html

Some links II

Some links II

- **Some references to begin with**

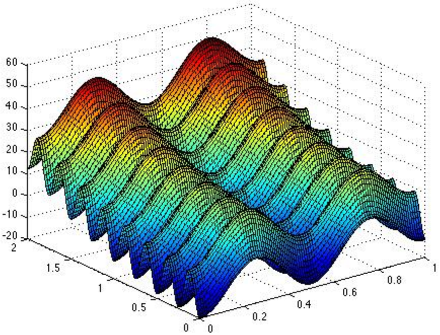
Some links II

- Some references to begin with
  - Linear and non linear regression: Azais, Antoniadis et al
  - Computer code experiments : Santner et al
  - Sensitivity analysis: Tarantolla et al, pioneering papers of Sobol, Antoniadis
  - Kriging: Stein, Cressie

A toy model

Rastrigin function

$$f(x) = f(x_1, x_2) = 8\|x\|^2 - 10(\cos(4\pi x_1) + \cos(8\pi x_2))$$

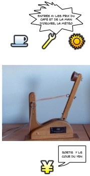


See `http://www.gdr-mascotnum.fr/doku.php?id=benchmarks`

Recall the goal

Model

$$Y = f(X).$$



- $X = (X_i)_{i=1...k}$ input vector
- Y output (real number).

Goal: Which of the components of X are more influent on Y?

Deterministic methods

Roughltly speaking are based on derivative of f:

Deterministic methods

Roughltly speaking are based on derivative of f:

- $\overline{x}$ being a point where the code is usually used

Deterministic methods

Roughltly speaking are based on derivative of f:

- $\overline{x}$ being a point where the code is usually used
- the influence of X_j is quantified using $(\frac{\partial f}{\partial X_j})(\overline{x})$.

Effective computation of the derivative

Deterministic methods

Roughltly speaking are based on derivative of f:

- $\overline{x}$ being a point where the code is usually used
- the influence of X_j is quantified using $(\frac{\partial f}{\partial X_j})(\overline{x})$.

Effective computation of the derivative

- Finite differences

$$(\frac{\partial f}{\partial X_j})(\overline{x}) \approx h^{-1}\left[f(\overline{x}_{j,h^+}) - f(\overline{x}_{j,h^-})\right]$$

Deterministic methods

Roughltly speaking are based on derivative of f:

- $\overline{x}$ being a point where the code is usually used
- the influence of X_j is quantified using $(\frac{\partial f}{\partial X_j})(\overline{x})$.

Effective computation of the derivative

- Finite differences

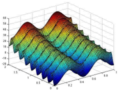
$$(\frac{\partial f}{\partial X_j})(\overline{x}) \approx h^{-1}\left[f(\overline{x}_{j,h^+}) - f(\overline{x}_{j,h^-})\right]$$

- Adjoint methods: the derivative is directly computed by the code (PDE models).

Deterministic methods-toy model

Rastrigin function

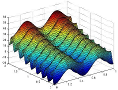
$$f(x) = f(x_1, x_2) = 8\|x\|^2 - 10(\cos(4\pi x_1) + \cos(8\pi x_2))$$

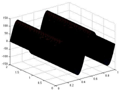


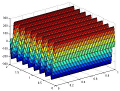
Deterministic methods-toy model

Rastrigin function

$$f(x) = f(x_1, x_2) = 8\|x\|^2 - 10(\cos(4\pi x_1) + \cos(8\pi x_2))$$



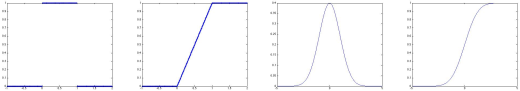


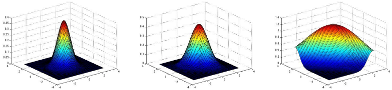


The derivative method is quite unstable.

A first insight in probability theory: Random variables

- Probability distribution
  - Z random variable on $\mathbb{R}$: most often with *density*. Repartition of Z is described by a function, ("mass function ").





  - Random vector on $\mathbb{R}^2$
  - Generalization: random vector on $\mathbb{R}^k$. Example multivariate (centered) Gaussian distribution with density

$$\frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{\det \Gamma}} \exp[\frac{1}{2} z^T \Gamma^{-1} z].$$

- Independence of random variables (Z_1, Z_2): observing Z_1 give no information on the distribution of Z_2.

A first insight in probability theory: Expectation, Variance

Z a random variable having distribution F.

- Expectation of a random variable: $\mathbb{E}(Z)$
  - Gravity center
  - Constant that explains the best the random variable.
  - Projection on constant *random* variables
- Variance of a random variable: $\mathrm{Var}(Z)$
  - Inertia moment
  - Magnitud of the fluctuactions around the mean
  - Squared norm of the random variable after having taken off the mean effect

Pythagora's Theorem

$$\mathbb{E}(Z^2) = \|Z\|^2 = \|\mathbb{E}(Z)\|^2 + \|Z - \mathbb{E}(Z)\|^2 = \mathbb{E}(Z)^2 + \mathrm{Var}(Z)$$

$$\mathrm{Var}(Z) = \mathbb{E}(Z^2) - \mathbb{E}(Z)^2.$$

Distribution examples : Expectation, Variance

- The most popular: Gaussian distribution (m, σ^2)
  - density on $\mathbb{R}$

$$g(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(z-m)^2}{2\sigma^2}\right].$$

  - Expectation

$$\mathbb{E}(Z) = \int_{-\infty}^{+\infty} z g(z) dz = \int_{-\infty}^{+\infty} \frac{z}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(z-m)^2}{2\sigma^2}\right] dz = m.$$

  - Variance

$$\mathrm{Var}(Z) = \mathbb{E}[(Z - \mathbb{E}(Z)^2)^2] = \mathbb{E}(Z^2) - [\mathbb{E}(Z)]^2 = \sigma^2.$$

- The most random: Uniform on $[z_{\text{min}}, z_{\text{max}}]$
  - density on $\mathbb{R}$

$$g(z) = \frac{\mathbf{1}_{[z_{\text{min}}, z_{\text{max}}]}(z)}{z_{\text{max}} - z_{\text{min}}}.$$

  - Expectation

$$\mathbb{E}(Z) = \int_{z_{\text{min}}}^{z_{\text{max}}} zg(z)dx = \int_{z_{\text{min}}}^{z_{\text{max}}} zdz = \frac{z_{\text{min}} + z_{\text{max}}}{2}$$

  - Variance

$$\text{Var}(Z) = \mathbb{E}[(Z - \mathbb{E}(Z)^2)^2] = \mathbb{E}(Z^2) - [\mathbb{E}(Z)]^2 = \frac{(z_{\text{max}} - z_{\text{min}})^2}{12}.$$

A first insight in probability theory: Conditional expectation

(Z_1, Z_2) a random vector

- Conditional expectation of Z_2 knowing Z_1): $\mathbb{E}(Z_2|Z_1)$
  - $Z_1 = z_1$ has been observed how one can predict the best Z_2?
  - What is the best function of Z_1 to explain Z_2?
  - Projection of Z_2 on functions of Z_1.

  Examples
  - $\mathbb{E}(Z_2|Z_1) = \mathbb{E}(Z_2)$ when (Z_1, Z_2) are independent random variables
  - $\mathbb{E}(Z_2|Z_1) = \rho Z_1$ for a centered Gaussian vector

- Some interesting facts for $\mathbb{E}(Z_2|Z_1)$
  - $\mathbb{E}[\mathbb{E}(Z_2|Z_1)] = \mathbb{E}[Z_2]$
  - $\mathbb{E}[\psi(Z_1)Z_2|Z_1] = \psi(Z_1)\mathbb{E}(Z_2|Z_1)$
  - Pythagora's Theorem

$$\mathbb{E}[Z_2^2] = \mathbb{E}[\mathbb{E}(Z_2|Z_1)^2] + \mathbb{E}[(Z_2 - \mathbb{E}(Z_2|Z_1))^2]$$

taking off $[\mathbb{E}(Z_2)]^2$

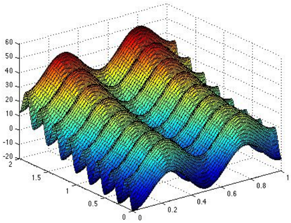
$$\text{Var}(Z_2) = \text{Var}[\mathbb{E}(Z_2|Z_1)] + \mathbb{E}[(Z_2 - \mathbb{E}(Z_2|Z_1))^2].$$

Of course, it is possible to generalize the notion of conditional expectation for a vector (Z_1 is a random vector).

Example toy model

- Rastrigin function

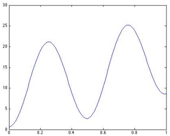
$$Y = f(X) = f(X_1, X_2) = 8\|X\|^2 - 10(\cos(4\pi X_1) + \cos(8\pi X_2))$$



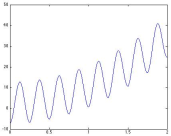
Example toy model

Assume that $X_1 \sim \mathcal{U}([0,1])$ et $X_2 \sim \mathcal{U}([0,2])$

$$\mathbb{E}(Y|X_1) = 8X_1^2 - 10\cos(4\pi X_1) + \frac{32}{3}$$



$$\mathbb{E}(Y|X_2) = 8X_2^2 - 10\cos(8\pi X_2) + \frac{8}{3}$$



An important example of vectorial conditioning

Centered Gaussian model

$$Z^T = (Z_1, Z_2)^T = (Z_1^1, \ldots Z_1^l, Z_2)$$

Gaussian vector with density

$$\frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{\det \Gamma}} \exp[\frac{1}{2} z^T \Gamma^{-1} z].$$

Γ is the covariance matrix of the random vector Z (assumed to be invertible):

$$\Gamma = \begin{pmatrix} \Gamma_{Z_1} & c^T_{Z_1,Z_2} \\ c_{Z_1,Z_2} & \sigma^2_{Z_2} \end{pmatrix}$$

- Γ_{Z_1} is the covariance matrix of the random vector Z_1,
- c_{Z_1,Z_2} is the covariance vector between Z_1 and Z_2 (row vector),
- $\sigma^2_{Z_2}$ is the variance of Z_2.

Centered Gaussian model

Theorem

$$\mathbb{E}(Z_2|Z_1) = c_{Z_1,Z_2}\Gamma_{Z_1}^{-1}Z_1,$$
$$\mathbb{E}[Z_2 - \mathbb{E}(Z_2|Z_1)]^2 = \sigma_{Z_2}^2 - c_{Z_1,Z_2}\Gamma_{Z_1}^{-1}c_{Z_1,Z_2}^T.$$

- Linear prediction,
- 1-d example $\mathbb{E}(Z_2|Z_1) = \rho Z_1$,
- Kalman filter=recursive formulation of the previous theorem

Sobol method

Model

$$Y = f(X).$$

We will quantify the *stochastic* influence of each input variables using previous projections:

Definition

Sobol indices for the output Y

- *First order indice for the input X_i*

$$S_i = \frac{Var(\mathbb{E}[Y|X_i])}{Var(Y)}$$

- *2nd order indice for the inputs $X_i, X_{j)}$*

$$S_{i,j} = \frac{Var(\mathbb{E}[Y|X_i, X_j])}{Var(Y)} - S_i - S_j$$

Sobol-Antoniadis (Hoeffding) Decomposition

Generalization: third order for the input X_i, X_j, X_l

$$S_{i,j,l} = \frac{\mathrm{Var}(\mathbb{E}[Y|X_i, X_j, X_l])}{\mathrm{Var}(Y)} - \sum_{i_1 < i_2 \in \{i,j,l\}} S_{i_1,i_2} + S_i + S_j + S_l$$

$S_{i,j,l}$ joint influence of X_i, X_j et X_k (marginal effects erased).

Theorem (Sobol-Antoniadis-Hoefding)

Assume that. $X_1, X_2, \ldots, X_k$ *are independent. then*

$$1 = \sum S_{ijl\ldots}$$

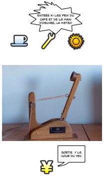
Sobol indices estimation

- Monte Carlo methods,
- Quasi Monte Carlo methods: FAST,
- Gaussian methods metamoddeling: Kriging O Oakley et al,
- Mathematical Statistics ANR COSTA BRAVA,

Recall the goal

Model

$$Y = f(X).$$



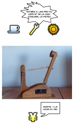
- $X = (X_i)_{i=1\dots k}$ is the input vector
- Y is the output (real number).

Goal: Build a function $\tilde{f}$ (cheap in terms of CPU) to emulate (approximate, estimate) f.

Several approaches

Model

$$Y = f(X).$$



Goal: Build a function $\tilde{f}$ (cheap in terms of CPU) to emulate (approximate, estimate) f.

- Approximation of f by a linear combination of given functions (e.g. Fourier, chaos or orthogonal polynomials,...),
- The same but non linear approximation (neural networks, non parametric statistics...),
- Discussed method: Bayesian approach using Gaussian processes (fields).

A short journey towards Gaussian fields

Gaussian vector $Z = (Z_i)_{i=1\ldots k}$: finite number of components
Random Gaussian field $Z = (Z_t)_{t \in T}$: *many* components as the elements of T ($T = \mathbb{Z}, \mathbb{R}, \mathbb{C}, \mathbb{R}^k$).
Gaussian vector: the probability density is

$$\frac{1}{(2\pi)^{\frac{k}{2}} \sqrt{\det \Gamma}} \exp[\frac{1}{2}(z - m)^T \Gamma^{-1} (z - m)].$$

The important parameters are:

- The mean (expectation) m vector of $\mathbb{R}^k$,
- The covariance matrix Γ ($\gamma_{i,j} = \text{cov}(Z_i, Z_j)$)

Random Gaussian field

Random Gaussian field: for any sample points $t_1, t_2, \dots t_p \in T$, the vector

$$Z := (Z_{t_i})_{i=1\dots p}$$

is a Gaussian vector. The important parameters are:

- The mean function $m(t) = \mathbb{E}(Z_t), \; t \in T$,
- The covariance function $\gamma(t, t') = \text{cov}(Z_t, Z_{t'}), t, t' \in T$

STATIONARY Gaussian field

STATIONARY Gaussian field: modeling an *unmoving dynamic* (in space or time) phenomena Translation on the parameters:

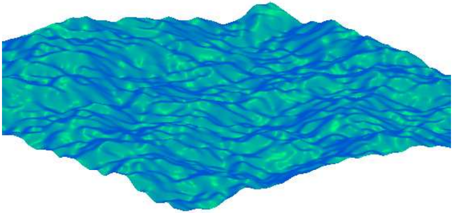
- The mean function is constant $m(t) = m, \ t \in T$,
- The covariance function only depends on $t - t'$
  $\gamma(t, t') = \text{cov}(Z_t, Z_{t'}) = r(t - t')$.

Classical frame

- Vanishing mean function $m(t) = 0, \ t \in T$,
- The covariance function $r(u)$ depends on some parameter θ. For example, assuming isotropy $r(u) = \exp(h\|u\|^\alpha) \ u \in T$. Here, the parameter is $\theta = (h, \alpha)$ $(h > 0, \alpha \geq 2)$.

Example modeling the sea

STATIONARY Gaussian process on $\mathbb{R}^2$ with an *ad hoc* covariance function (See the excellent book of Azais-Wchebor)



Kriging 0

Bayesian model in geostatistics

Kriging 0

Bayesian model in geostatistics

$$f(x) = \alpha_\theta(x) + Z_x(\nu)(x \in T)$$

Kriging 0

Bayesian model in geostatistics

$$f(x) = \alpha_\theta(x) + Z_x(\nu)(x \in T)$$

- θ and ν are unknown vectorial parameters

Kriging 0

Bayesian model in geostatistics

$$f(x) = \alpha_\theta(x) + Z_x(\nu)(x \in T)$$

- θ and ν are unknown vectorial parameters
- α_θ a simple mean function (*trend*): $\langle \theta, x \rangle$

Kriging 0

Bayesian model in geostatistics

$$f(x) = \alpha_\theta(x) + Z_x(\nu)(x \in T)$$

- θ and ν are unknown vectorial parameters
- α_θ a simple mean function (*trend*): $\langle \theta, x \rangle$
- $(Z_x)_{x \in T}$ centered stationary Gaussian field $r_\nu(x)$, $x \in T$

Kriging I

Main idea=THE COMPUTER CODE IS THE REALIZATION OF A GAUSSIAN FIELD TRAJECTORY

Kriging I

Main idea=THE COMPUTER CODE IS THE REALIZATION OF A GAUSSIAN FIELD TRAJECTORY

The model has been played *randomly*

$$Y(x) = f(x)(x \text{ deterministic} \in T)$$

Kriging I

Main idea=THE COMPUTER CODE IS THE REALIZATION OF A GAUSSIAN FIELD TRAJECTORY

The model has been played *randomly*

$$Y(x) = f(x)(x \text{ deterministic} \in T)$$

Kriging II

Bayesian model for the black box

Kriging II

Bayesian model for the black box

The model is running on a design $x_1, \dots x_N$ we have at hand $f(x_1), \dots, f(x_N)$.

Kriging II

Bayesian model for the black box

The model is running on a design $x_1, \ldots x_N$ we have at hand $f(x_1), \ldots, f(x_N)$. Model

$$Y(x) = \alpha_\theta(x) + Z_x(\nu)(x \in T)$$

Kriging II

Bayesian model for the black box

The model is running on a design $x_1, \ldots x_N$ we have at hand $f(x_1), \ldots, f(x_N)$. Model

$$Y(x) = \alpha_\theta(x) + Z_x(\nu)(x \in T)$$

- One uses $f(x_1), \ldots, f(x_N)$ to estimate the parameters θ et ν

Kriging II

Bayesian model for the black box

The model is running on a design $x_1, \dots x_N$ we have at hand $f(x_1), \dots, f(x_N)$. Model

$$Y(x) = \alpha_\theta(x) + Z_x(\nu)(x \in T)$$

- One uses $f(x_1), \dots, f(x_N)$ to estimate the parameters θ et ν
- Maximum likelihood method

Kriging II

Bayesian model for the black box

The model is running on a design $x_1, \ldots x_N$ we have at hand $f(x_1), \ldots, f(x_N)$. Model

$$Y(x) = \alpha_\theta(x) + Z_x(\nu)(x \in T)$$

- One uses $f(x_1), \ldots, f(x_N)$ to estimate the parameters θ et ν
- Maximum likelihood method
- *Roughly speaking :* least square fit of the parameters with weight functions depending on the parameters

Kriging III

Bayesian approach

Kriging III

Bayesian approach

- One have at hand $f(x_1), \ldots, f(x_N)$. Valeurs modeled by par $Y(x_1), \ldots Y(x_N)$.

Kriging III

Bayesian approach

- One have at hand $f(x_1), \dots, f(x_N)$. Valeurs modeled by par $Y(x_1), \dots Y(x_N)$.
- Parameters θ and ν has been previously *identified*

Gaussian emulator

$$\widehat{f}(x) = \widehat{Y}(x) = \mathbb{E}[Y(x) | Y(x_1), \dots Y(x_N)]$$

Kriging III

Bayesian approach

- One have at hand $f(x_1), \ldots, f(x_N)$. Valeurs modeled by par $Y(x_1), \ldots Y(x_N)$.
- Parameters θ and ν has been previously *identified*

Gaussian emulator

$$\widehat{f}(x) = \widehat{Y}(x) = \mathbb{E}[Y(x)|Y(x_1), \ldots Y(x_N)]$$

- Very simple formula

$$\widehat{Y}(x) \quad = \quad \alpha_\theta(x) + \mathbb{E}[Z_x|Z_{x_1}, \ldots Z_{x_N}]$$

Kriging III

Bayesian approach

- One have at hand $f(x_1), \ldots, f(x_N)$. Valeurs modeled by par $Y(x_1), \ldots Y(x_N)$.
- Parameters θ and ν has been previously *identified*

Gaussian emulator

$$\widehat{f}(x) = \widehat{Y}(x) = \mathbb{E}[Y(x)|Y(x_1), \ldots Y(x_N)]$$

- Very simple formula

$$\begin{aligned} \widehat{Y}(x) &= \alpha_\theta(x) + \mathbb{E}[Z_x|Z_{x_1}, \ldots Z_{x_N}] \\ &= \alpha_\theta(x) + c_x^T \Gamma_N^{-1} Z^N \end{aligned}$$

Kriging III

Bayesian approach

- One have at hand $f(x_1), \dots, f(x_N)$. Valeurs modeled by par $Y(x_1), \dots Y(x_N)$.
- Parameters θ and ν has been previously *identified*

Gaussian emulator

$$\widehat{f}(x) = \widehat{Y}(x) = \mathbb{E}[Y(x) | Y(x_1), \dots Y(x_N)]$$

- Very simple formula

$$\begin{aligned} \widehat{Y}(x) &= \alpha_\theta(x) + \mathbb{E}[Z_x | Z_{x_1}, \dots Z_{x_N}] \\ &= \alpha_\theta(x) + c_x^T \Gamma_N^{-1} Z^N \end{aligned}$$

- c_x covariance vector of Z_x and $Z_{x_1}, \dots, Z_{x_N}$,

Kriging III

Bayesian approach

- One have at hand $f(x_1), \dots, f(x_N)$. Valeurs modeled by par $Y(x_1), \dots Y(x_N)$.
- Parameters θ and ν has been previously *identified*

Gaussian emulator

$$\widehat{f}(x) = \widehat{Y}(x) = \mathbb{E}[Y(x) | Y(x_1), \dots Y(x_N)]$$

- Very simple formula

$$\begin{aligned} \widehat{Y}(x) &= \alpha_\theta(x) + \mathbb{E}[Z_x | Z_{x_1}, \dots Z_{x_N}] \\ &= \alpha_\theta(x) + c_x^T \Gamma_N^{-1} Z^N \end{aligned}$$

- c_x covariance vector of Z_x and $Z_{x_1}, \dots, Z_{x_N}$,
- Γ_N covariance matrix between $Z^N := (Z_{x_1}, \dots, Z_{x_N})^T$.

Kriging IV

Bayesian method in a functional space

Kriging IV

Bayesian method in a functional space

- Emulation method by linear regression

Kriging IV

Bayesian method in a functional space

- Emulation method by linear regression

$$\widehat{Y}(x) = \alpha_\theta(x) + c_x^T \Gamma_N^{-1} Z^N$$

Kriging IV

Bayesian method in a functional space

- Emulation method by linear regression

$$\widehat{Y}(x) = \alpha_\theta(x) + c_x^T \Gamma_N^{-1} Z^N$$

- Prediction error of Gaussian du model (if the parameter of the model are known)

$$\mathbb{E}[(Y(x) - \widehat{Y}(x))^2] = r_\nu(0) - c_x^T \Gamma_N^{-1} c_x$$

One example
from :http://www2.imm.dtu.dk/ hbn/dace/

One example

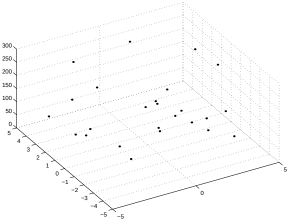
from :http://www2.imm.dtu.dk/ hbn/dace/

DTU

Introduction

Given $f : \mathbb{R}^n \mapsto \mathbb{R}$. May be a black-box (and "expensive") function.

Know values $y_i = f(s_i)$ at *design sites* $S = \{s_1, \ldots, s_m\}$. How does the function behave in between?



ISMP 2003 3

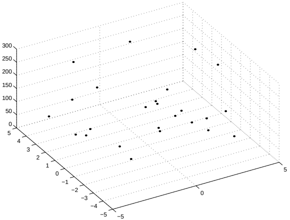
One example from :http://www2.imm.dtu.dk/~hbn/dace/

DTU

Introduction

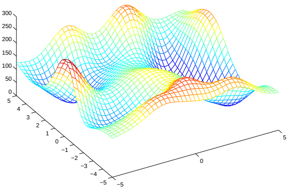
Given $f : \mathbb{R}^n \mapsto \mathbb{R}$. May be a black-box (and "expensive") function.

Know values $y_i = f(s_i)$ at *design sites* $S = \{s_1, \ldots, s_m\}$. How does the function behave in between?



DTU

Gauss model

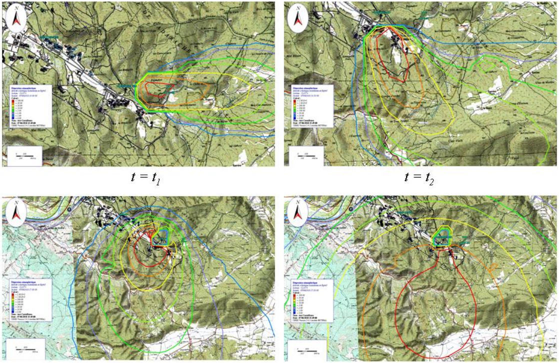


End

Gracias por su atencion
Thanks for your attention
Merci
Obrigado
Danke
Grazie

Le code CERES

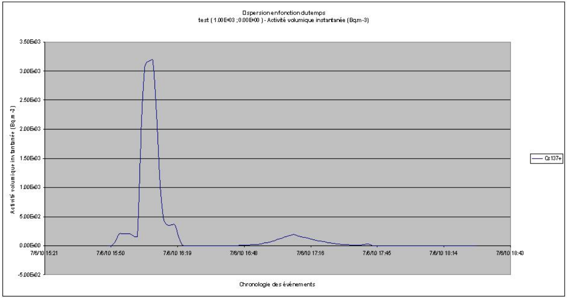
Evolution dans le temps de l'activité volumique instantanée du ^{137}Cs dans l'air



$t = t_1$ $t = t_2$ $t = t_3$ $t = t_4$

Le code CERES(bis)

Evolution dans le temps de l'activité volumique instantanée du ^{137}Cs en 1 point donné



back