Optimization with uncertainties
Methods from the OMD projects

Rodolphe Le Riche
CNRS and Ecole des Mines de Saint-Etienne

Uncertainty quantification for numerical model validation
summer school, CEA Cadarache, June 2011

Outline of the talk

1. Introduction to optimization : uses, challenges.

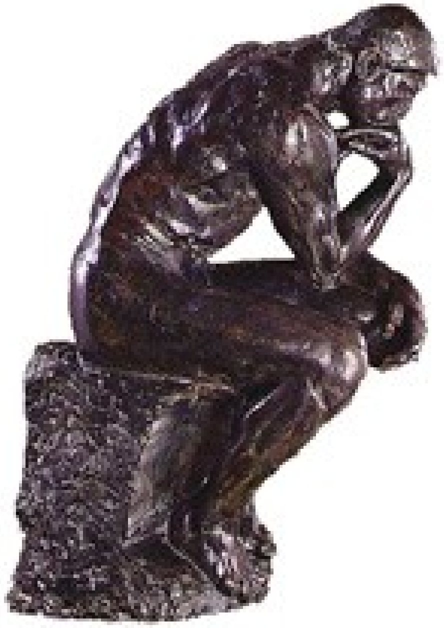
2. Formulations of optimization problems with uncertainties

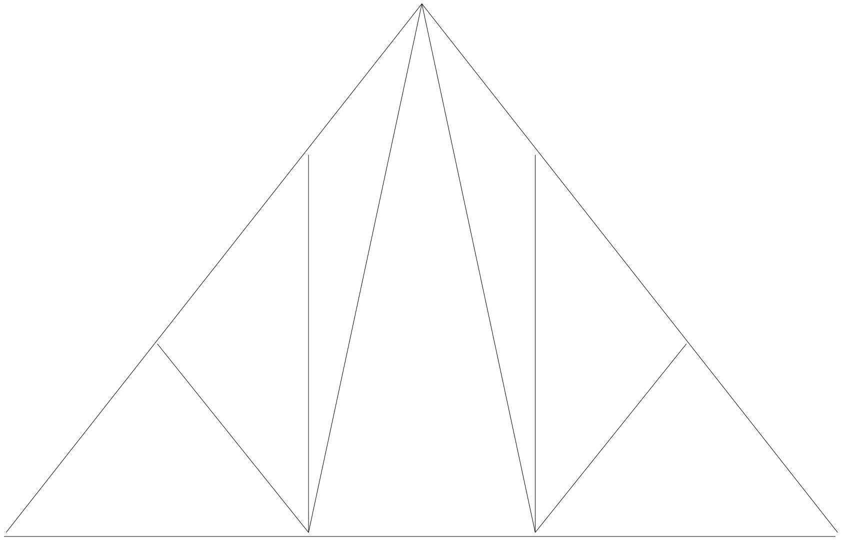
3. Noisy optimization

4. Kriging-based approaches (spatial statistics)

Goal of parametric numerical optimization

Ex : 15 bars truss, each bar chosen out of 10 profiles
→ 10^{15} possible trusses. How to choose ?
Choose the position of the joints (continuous)
→ How to search in $R^{+,15}$?





How to choose ? The modeling, formulation, optimization steps

1. Have a model or « simulator », y , (analytical, finite elements, coupled sub-models …) of the object you need to optimize.
2. Formulate the optimization problem

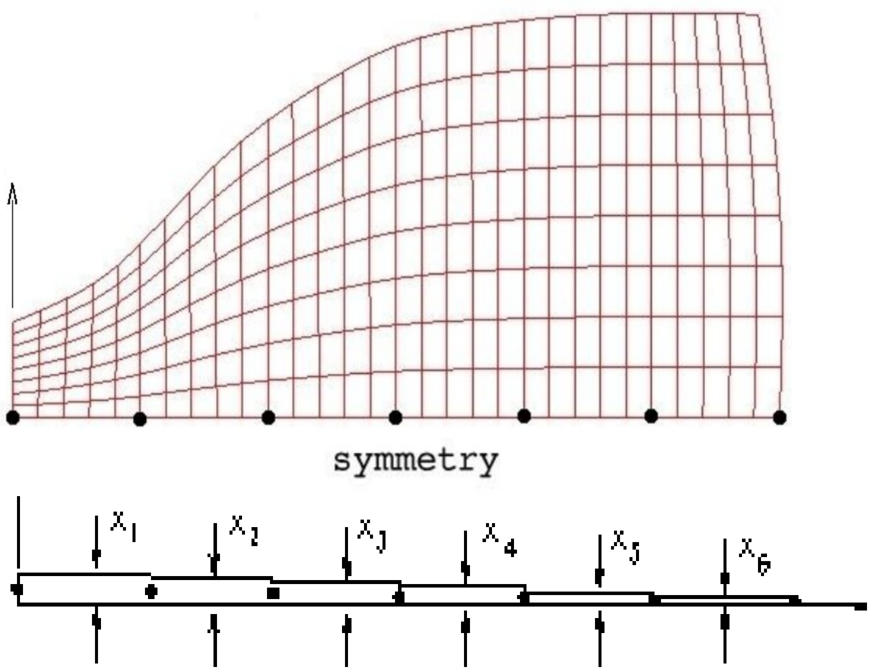
$$\min_{x \in S} f(y(x))$$
$$g(y(x)) \leqslant 0$$

x : optimization variables
f : objective functions
g : optimization constraints
f , g : optimization (performance) criteria

3. ***Try*** to solve the problem, either analytically (e.g., Karush Kuhn and Tucker conditions) or using optimization algorithms.
[4. Almost never right the first time : go back to 1]

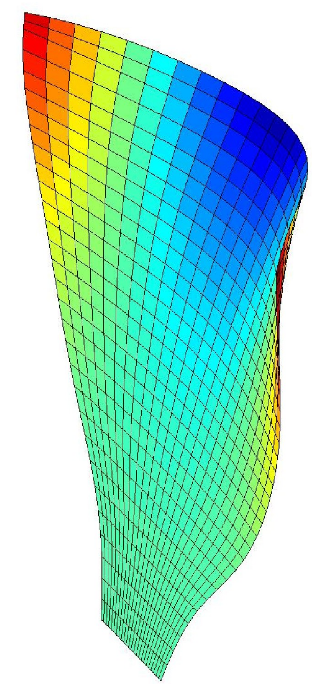
Application example (1) : structural design

Max_x forward power
such that total power < $power^{max}$



parameterization = choice of x

simulation model
y(x)

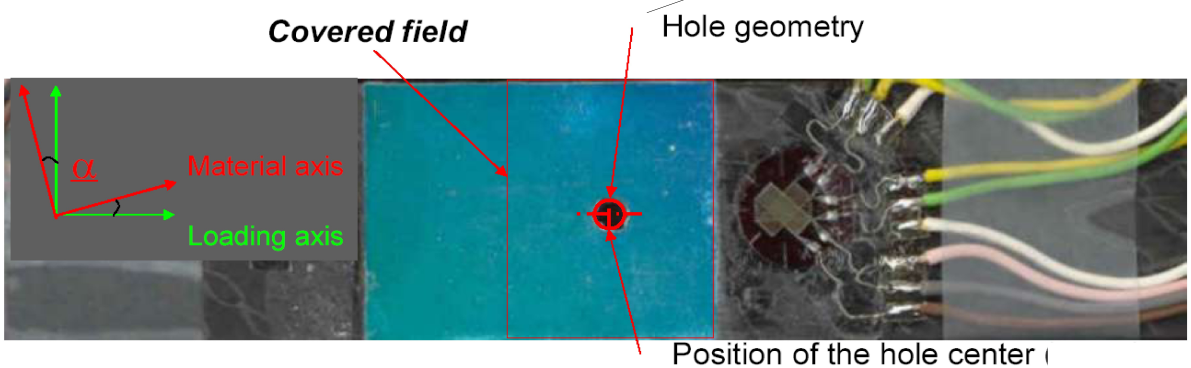


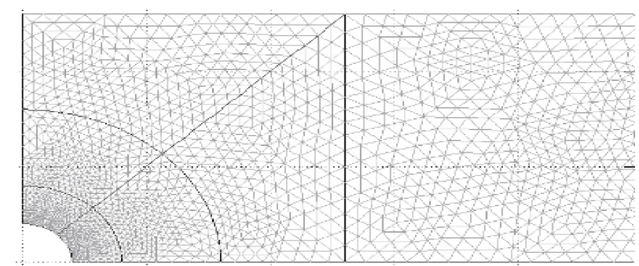
Luersen, M.A., Le Riche, R., Lemosse, D. and Le Maître, O.,
A computationally efficient approach to swimming monofin optimization, SMO 2006

Application example (2) : identification

x : material parameters

Min_x distance($y^{measured}$, $y(x)$)





Silva, G., Le Riche, R., Molimard, J., Vautrin, A. and Galerne, C., Identification of material properties using FEMU : Application to the open hole tensile test, J. of Appl. Mech. and Mat. 2007

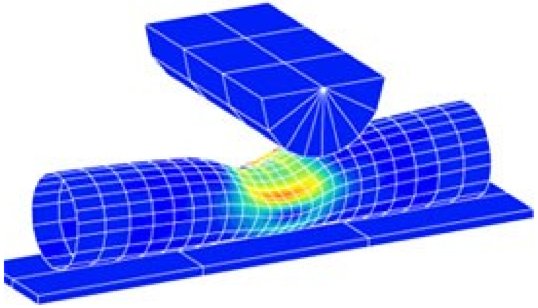
and similarly in supervised learning from data points (regression, classification, …) : x = model parameters, f = data representation or classification error (+ regularization).

A. Rakotomamonjy, R. Le Riche, D.Gualandris and Z. Harchaoui, A comparison of statistical learning approaches for engine torque estimation, Control Engineering Practice, 2008.

Application examples (3)

Optimal control

$x \equiv$ *rudder_angle(t)*
$f(x) \equiv$ time to goal

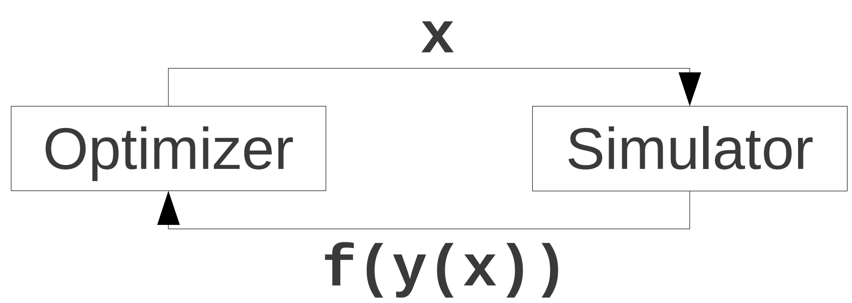


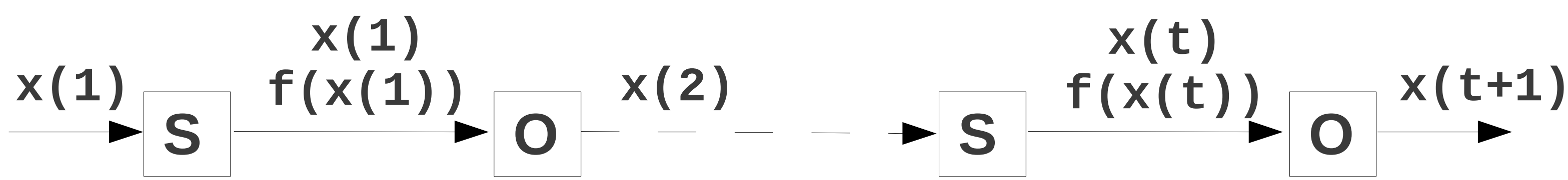
Modeling

in mechanics
$x \equiv$ nodes displacements
$f(x) \equiv$ total potential energy
$g(x) \equiv$ contact condition (non intrusion)

Optimization programs

An optimizer is an algorithm that iteratively proposes new x's based on past trials in order to approximate the solution to the optimization problem.





x(t+1) = Optimizer[x(1),f(x(1)) , … , x(t),f(x(t))]

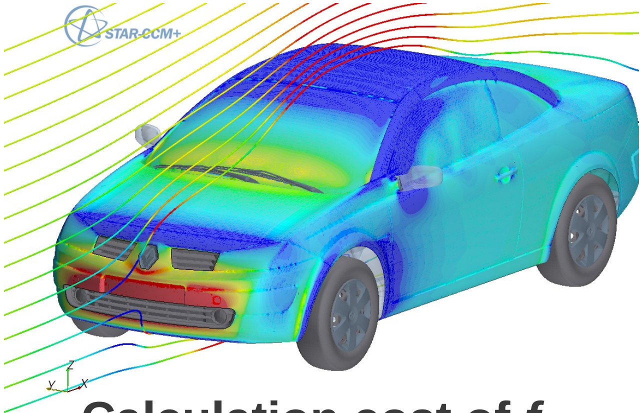
The ***cost*** of the optimization is the number of calls to the simulator y (usually = number of calls to f)

If relevant , $f \rightarrow f$ and g , on other slides too.

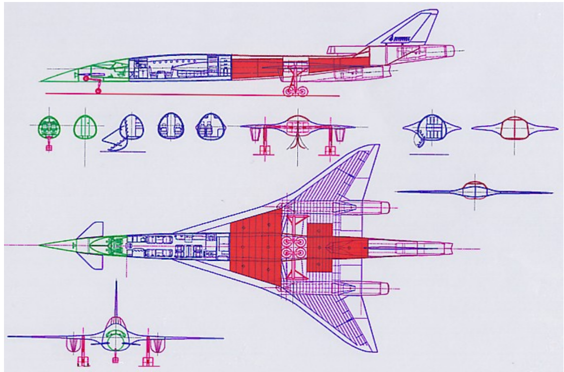
Make can make optimizing difficult

Goal : $min_{x \in S \subset \mathbb{R}^n} f(x)$

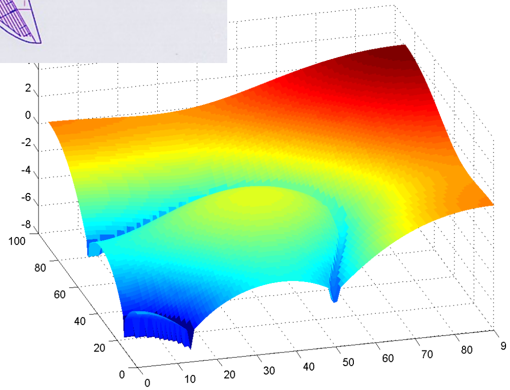
difficulties



Calculation cost of *f*

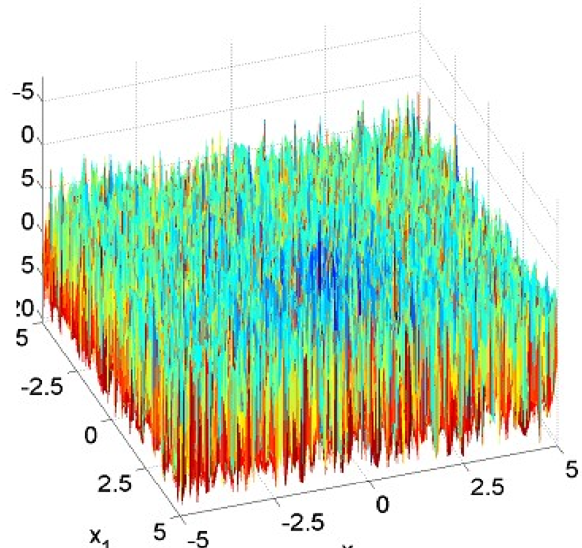


Number of variables, *n*



local optima
(expl. composites)

Ill conditioning



Noise

Outline of the talk

1. Introduction to optimization

2. Formulations of optimization problems with uncertainties

3. Noisy optimization

4. Kriging-based approaches

Formulations of optimization problems under uncertainties

G. Pujol, R. Le Riche, O. Roustant and X. Bay, *L'incertitude en conception: formalisation, estimation,* Chapter 3 of the book *Optimisation Multidisciplinaire en Mécaniques : réduction de modèles, robustesse, fiabilité, réalisations logicielles*, Hermes, 2009.

The double (x,U) parameterization

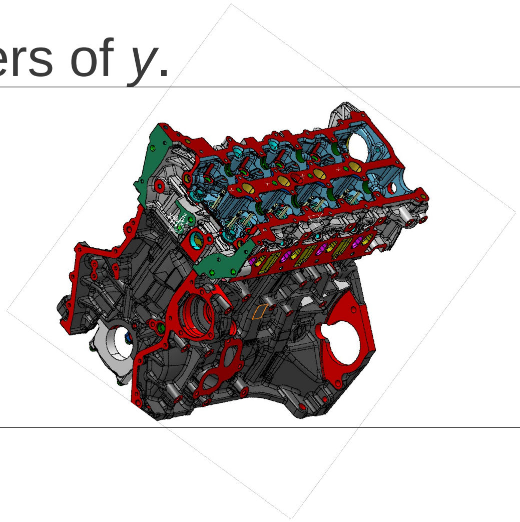
x is a vector of deterministic optimization (controlled) variables.
x in *S*, the search space.
We introduce *U*, a vector of uncertain (random) parameters that affect the simulator *y*.

$y(x) \rightarrow y(x,U)$, therefore $f(x) \rightarrow f(y(x,U)) = f(x,U)$
and $g(x) \rightarrow g(y(x,U)) = g(x,U)$

U used to describe

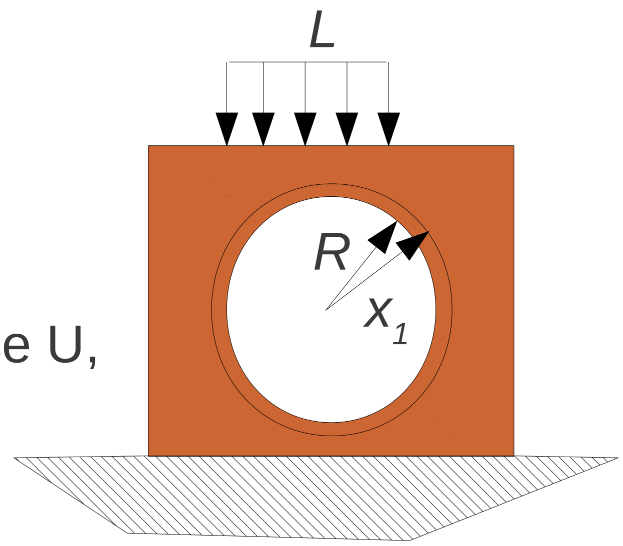
- noise (as in identification with noise measurement)
- model error (epistemic uncertainty)
- uncertainties on the values of some parameters of *y*.

Ex : a +/- 1mm dispersion in the manufacturing of a car cylinder head can degrade its performance (g CO_2/km) by +20% (worst case).



Formulation of optimization under uncertainty

The (x,U) parameterization is general



Two cases (which can be combined)

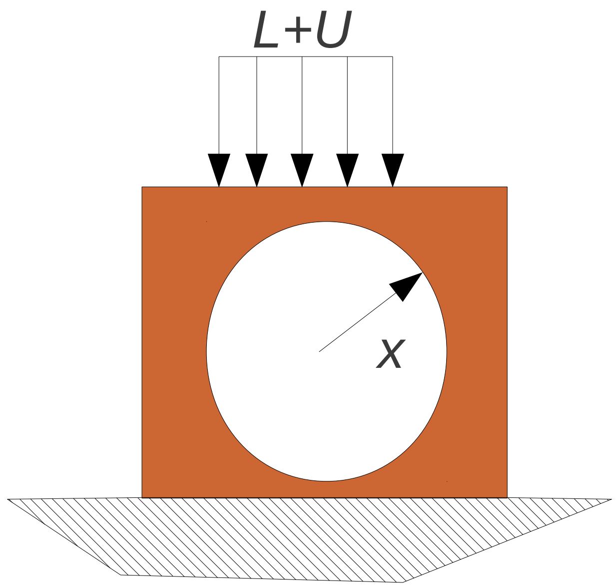
1. Noisy controlled variables

Expl : manufacturing tolerance U,

$R = x_1 + U$

$x = (\ E(R),\ VAR(R)\ )$

nominal value tolerance class



2. Noise exogenous to the optimization variables

Expl : U random part load added to load *L*, *x* is a geometric dimension.

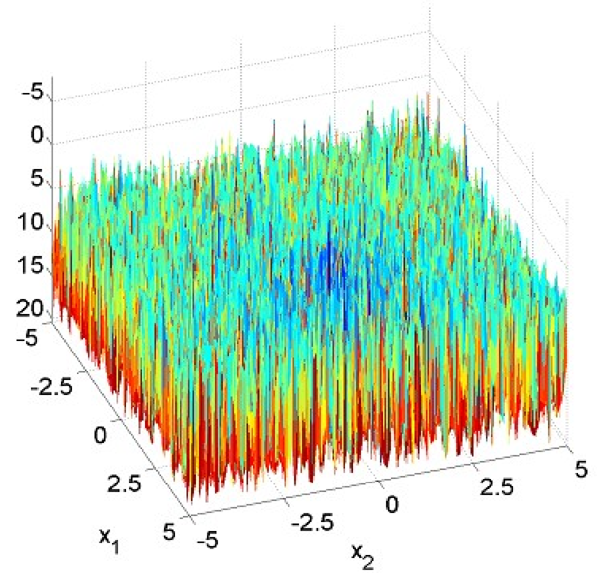
Expl : *y* finite element code, *f* volume of the structure, *g* upper bound on stresses.

Formulation of optimization under uncertainties
(1) the noisy case

Let's not do anything about the uncertainties, i.e., try to solve

$$\min_{x \in S} f(x, U)$$
$$g(x, U) \leqslant 0$$

$$U \text{ random}$$



It does not look good : gradients are not defined, what is the result of the optimization ?
But sometimes there is no other choice. Ex : *y* expensive simulator with uncontrolled random numbers inside (like a Monte Carlo statistical estimation, numerical errors, measured input).

Formulation of optimization under uncertainties
(2) an ideal series formulation

Replace the noisy optimization criteria by statistical measures

$G(x)$ is the random event "all constraints are satisfied" ,

$$G(x) = \bigcap_i \{ g_i(x, U) \leqslant 0 \}$$

$\min_{x \in S} q_\alpha^c(x)$ (conditional α-quantile)

such that $P(G(x)) \geqslant 1 - \varepsilon$

where $P\left(f(x, U) \leqslant q_\alpha^c(x) \mid G(x)\right) = \alpha$

$\varepsilon > 0$, small

Formulation of optimization under uncertainties

(3) simplified formulations often seen in practice

For bad reasons (joint probabilities ignored) or good ones (simple numerical methods, lack of data, organisation issues), quantiles are often replaced by averages and variances, conditioning is neglected, constraints are handled independently :

$$\min_{x \in S} q_\alpha(x) \quad \text{or} \quad \min_{x \in S} E\left(f(x,U)\right) \quad \text{and / or} \quad \min_{x \in S} V\left(f(x,U)\right)$$

$$\text{or} \quad \min_{x \in S} E\left(f(x,U)\right) + r\sqrt{V\left(f(x,U)\right)}$$

$$\text{where} \quad P\left(f(x,U) \leqslant q_\alpha\right) = \alpha \quad \text{and} \quad r > 0$$

$$\text{such that} \quad P\left(G(x)\right) \geqslant 1-\varepsilon \quad \text{or} \quad P\left(g_i(x) \leqslant 0\right) \geqslant 1-\varepsilon_i$$

where ε is the series system risk

and ε_i is the i th failure mode risk

Scope of the presentation

The field of optimization under uncertainties is extremely active.

From here on, the presentation focuses on methods for optimization under uncertainties developped in the neighborhood of the speaker i.e.,

the French national projects OMD and OMD2 (where OMD stands for Optimisation MultiDisciplinaire, MDO).

In other words, many useful contributions are not presented.

Related books :

• OMD book : *Multidisciplinary Design Optimization in Computational Mechanics*, P. Breitkopf and R. Filomeno Coehlo Eds., Wiley/ISTE, 2010
•A. Ben-Tal, L. El Ghaoui, A. Nemirowski, *Robust Optimization*, Princeton Univ. Press, 2009.
• R. E. Melchers, *Structural Reliability Analysis and Prediction*, Wiley, 1999.
• M. Lemaire, A. Chateauneuf, J.-C. Mitteau, *Structural Reliability*, Wiley, 2009.
• J. C. Spall, *Introduction to Stochastic Search and Optimization*, Wiley, 2003.
• A. J. Keane and P. B. Nair, *Computational Approaches for Aerospace Design: The Pursuit of Excellence*, Wiley, 2005.

Outline of the talk

1. Introduction to optimization

2. Formulations of optimization problems with uncertainties

3. Noisy optimization
- The general CMA-ES
- Improvements for noisy functions :
    - Mirrored sampling and sequential selection
    - Adding confidence to an ES

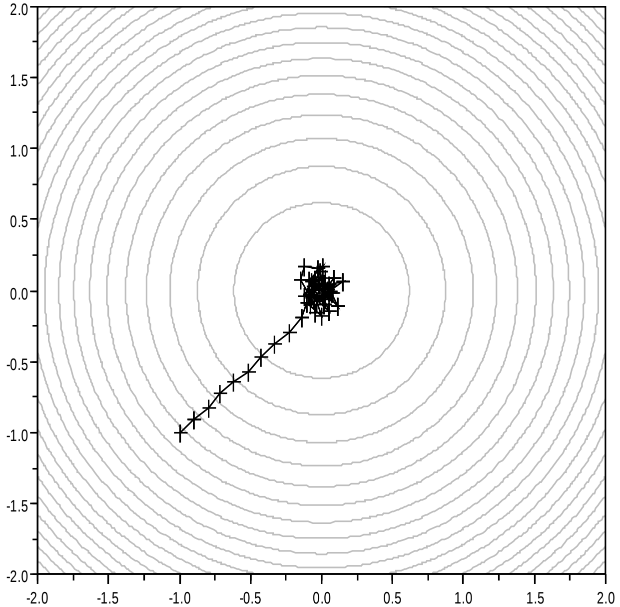
4. Kriging-based approaches

Continuous, unconstrained, noisy optimization

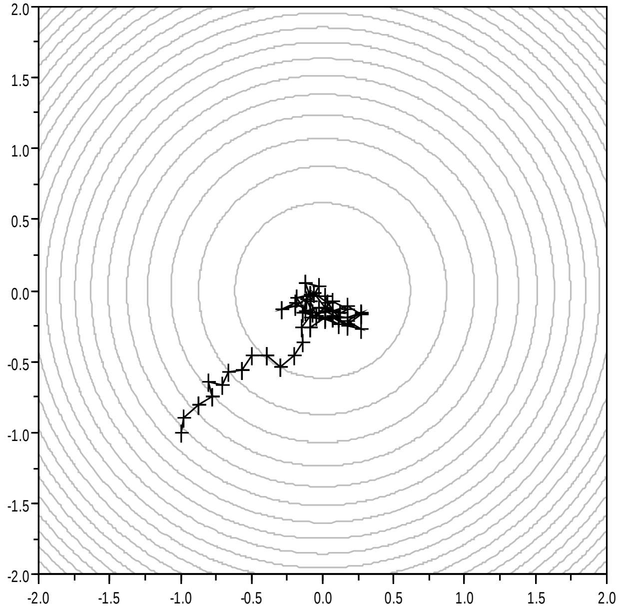
$$\min_{x \in \Re^n} f(x, U)$$ and no control over *U*, seen as noise.

Expl : convergence of a quasi-Newton method with finite differences.
A classical optimizer is sensitive to noise.

little noise



more noise



$$f(x) = \frac{1}{100} \sum_{i=1}^{100} \|x + u_i\|^2$$

$$u_i \sim N(0, I_2)$$

$$f(x) = \|x + u_i\|^2$$

Evolutionary algorithms

Taking search decisions in probability is a way to handle the noise corrupting observed *f* values

→ use a stochastic optimizer, an evolution strategy (ES). Assumptions : none.

« elitism »

A simple (1+1)-ES

Initializations : x, $f(x)$, m, C, t_{max}.

While $t < t_{max}$ do,

  Sample $N(m,C)$ --> x'

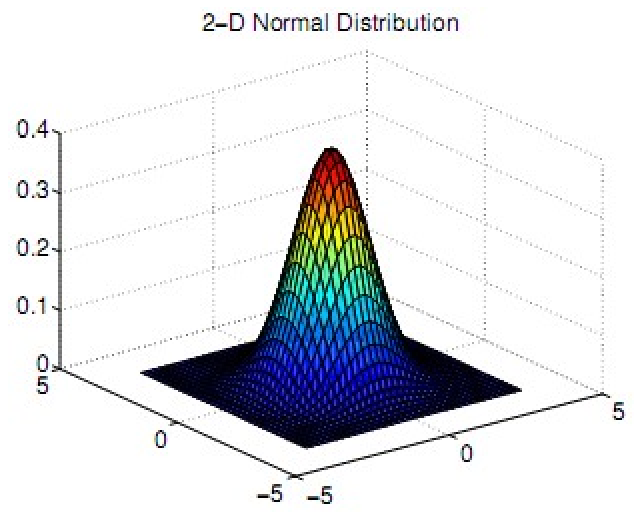
  Calculate $f(x')$, $t = t+1$

  If $f(x')<f(x)$, $x = x'$, $f(x) = f(x')$ Endif

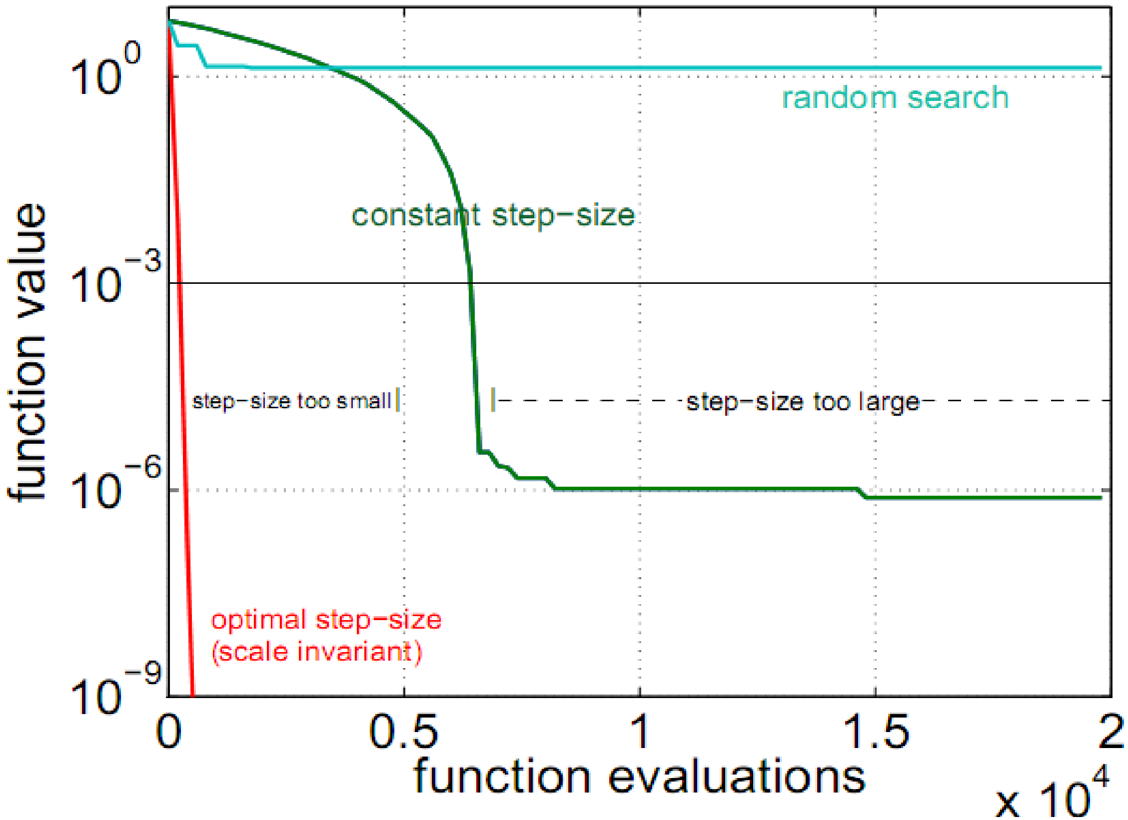
  Update m (*e.g.*, $m=x$) and C

End while

```
%(Scilab code)
x = m + grand(1,'mn',0,C)
```



Adapting the step size (C^2) is important



$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

in $[-0.2, 0.8]^n$
for $n = 10$

(A. Auger et N. Hansen, 2008)

Above isotropic ES(1+1) : $C = \sigma^2 \mathrm{I}$, σ is the step size.
With an optimal step size ($\approx \|x\| / n$) on the sphere function, performance degrades only in $O(n)$.

The population based CMA-ES

(N. Hansen et al., since 1996, now with A. Auger)

CMA-ES = *Covariance Matrix Adaptation Evolution Strategy* = optimization through sampling and updating of a multi-normal distribution.

A fully populated covariance matrix is build : pairwise variable interaction learned. Can adapt the step in any direction.

The state-of-the-art evolutionary / genetic optimizer for continuous variables.

flow-chart of CMA-ES

CMA-ES is a (non elitist) evolution strategy *ES*-(μ,λ) :

Initializations : m, C, t_{max}, μ , λ

While $t < t_{max}$ do,

Sample $N(m,C)$ --> $x^1,\ldots,x^\lambda$

Calculate $f(x^1),\ldots,f(x^\lambda)$, $t = t+\lambda$

Rank : $f(x^{1:\lambda}),\ldots,f(x^{\lambda:\lambda})$

Update m and C *with the* μ bests, $x^{1:\lambda},\ldots,x^{\mu:\lambda}$

End while

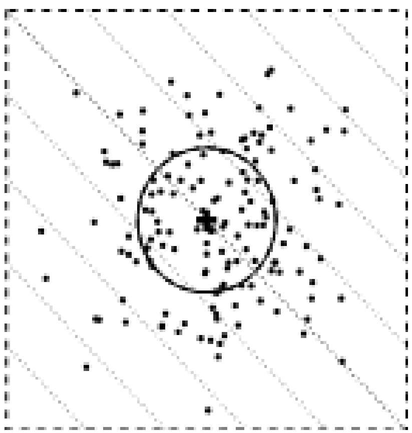
m *et* C are updated with

- the best **steps** (as opposed to points),
- a **time cumulation** of these best steps.

CMA-ES : adapting C^2 with good steps

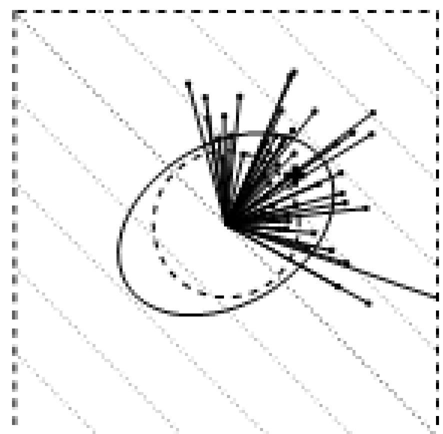
(A. Auger et N. Hansen, 2008)

Initialization : $m \in S$, $C = I$, $c_{cov} \approx 2/n^2$



sampling

$$x^i = m + y^i$$
$$y^i \propto N(0, C)$$
$$i = 1, \ldots, \lambda$$

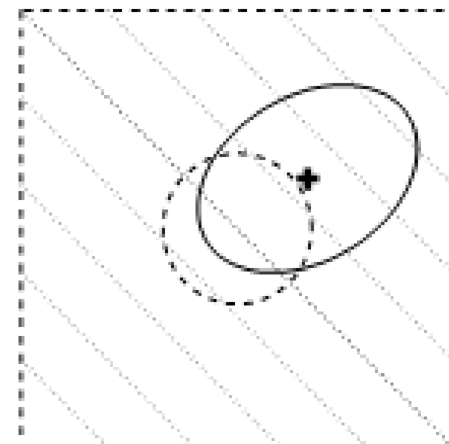


selection

$$y_w = \frac{1}{\mu} \sum_{i=1}^{\mu} y^{i:\lambda}$$

rank 1 C update

$$C \leftarrow (1 - c_{cov}) C + c_{cov} \mu \, y_w y_w^T$$



update m

$$m \leftarrow m + y_w$$

The state-of-the-art CMA-ES

(A. Auger and N. Hansen, *A restart CMA evolution strategy with increasing population size*, 2005)

Additional features :

- Steps weighting, $y_w = \sum_{i=1}^{\mu} w_i y^{i:\lambda}$
- Time cumulation of the steps.
- Simultaneous rank 1 and μ covariance adaptations.
- Use of a global scale factor, $C \rightarrow \sigma^2 C$.
- Restarts with increasing population sizes (unless it is the 2010 version with mirrored sampling and sequential selection, see later)

Has been used up to $n = 100$ continuous variables.

Outline of the talk

1. Introduction to optimization

2. Formulations of optimization problems with uncertainties

3. Noisy optimization
- The general CMA-ES
- Improvements for noisy functions :
  - Mirrored sampling and sequential selection
  - Adding confidence to an ES

4. Kriging-based approaches

Mirrored sampling and sequential selection (1)

D. Brockhoff, A. Auger, N. Hansen, D. V. Arnold, and T. Hohm. *Mirrored Sampling and Sequential Selection for Evolution Strategies,* PPSN XI, 2010

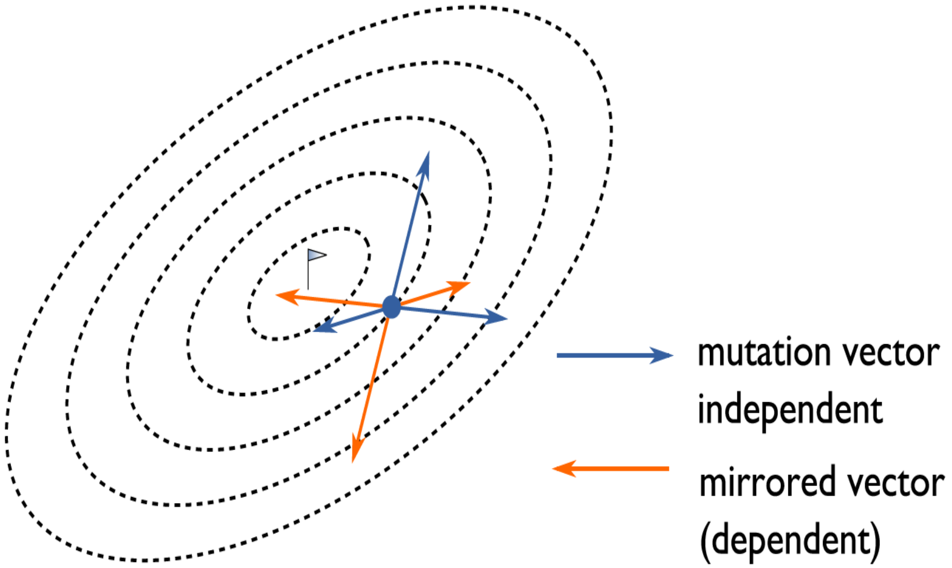
A. Auger, D. Brockhoff, N. Hansen, *Analysing the impact of mirrored sampling and sequential selection in elitist Evolution Strategies,* FOGA 2011

(1+1)-CMA-ES with restarts surprisingly good on some functions (including multimodal functions with local optima).

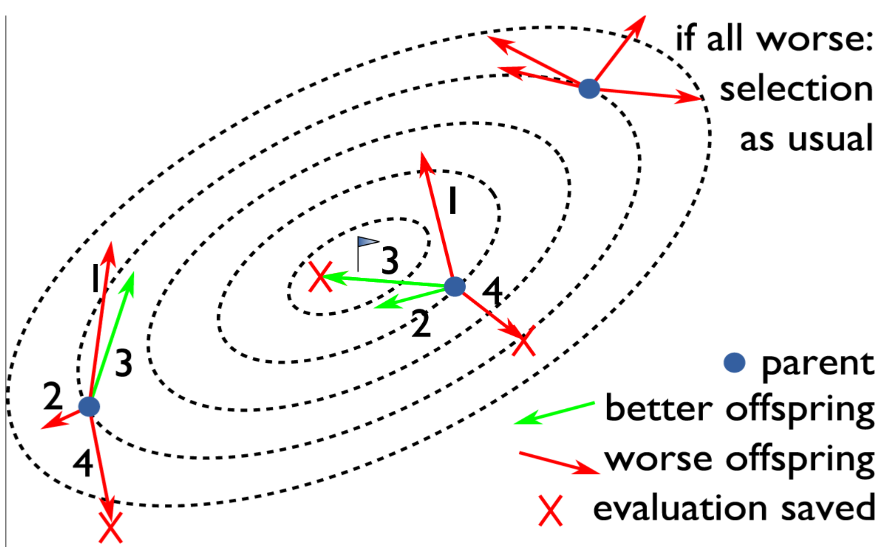
But « elitism » of (1+1)-ES bad for noisy functions : a lucky sample attracts the optimizer in a non-optimal region of the search space.

Question : how to design a fast local non-elistist ES ?

Mirrored sampling and sequential selection (2)



Derandomization via mirrored sampling : one random vector generates two offsprings.
Often good and bad in opposite directions.



Sequential selection : stop evaluation of new offsprings as soon as a solution better than the parent is found. Greedy !

Combine the two ideas : when an offspring is better than its parent, its symmetrical is worse (on convex level sets), and vice versa → evaluate in order $m+y^1$, $m-y^1$, $m+y^2$, $m-y^2$, … .

Mirrored sampling and sequential selection (3)

Results :

(1,4)-ES with mirroring and sequential selection faster than (1+1)-ES on sphere function.

Theoretical result: Convergence Rate ES (1+1)=0.202 ,
Convergence Rate (1,4ms)=0.223 .

Implementation within CMA-ES, tested in BBOB'2010* (Black Box Optimization Benchmarking)

Best performance among all algorithms tested so far on some functions of noisy testbed

* http://coco.gforge.inria.fr/bbob2010-downloads

Outline of the talk

1. Introduction to optimization

2. Formulations of optimization problems with uncertainties

3. Noisy optimization
 - The general CMA-ES
 - Improvements for noisy functions :
   - Mirrored sampling and sequential selection
   - Adding confidence to an ES

4. Kriging-based approaches

Adding confidence to an ES

D. Salazar, R. Le Riche, G. Pujol and X. Bay, *An empirical study of the use of confidence levels in RBDO with Monte Carlo simulations*, in Multidisciplinary Design Optimization in Computational Mechanics, Wiley/ISTE Pub., 2010.

Assumption : evaluations of f(x,u) can be repeated (even without control of u), $f(x,u^1)$, … , $f(x,u^s)$

Evolutionary optimizers are comparison based.
We now compare empirical averages of f, therefore solve

$$min_x E\left(f\left(x,U\right)\right)$$

Note : this can also be done with functions of *f* . For example, replace f(x,U) by its estimated *e*-th quantile (batching),

$$q\left(x,U\right)=f\left(x,U^{\lfloor e\times b\rfloor}\right) \quad , \quad f\left(x,U^1\right)\leqslant\ldots\leqslant f\left(x,U^b\right)$$

Hypothesis testing

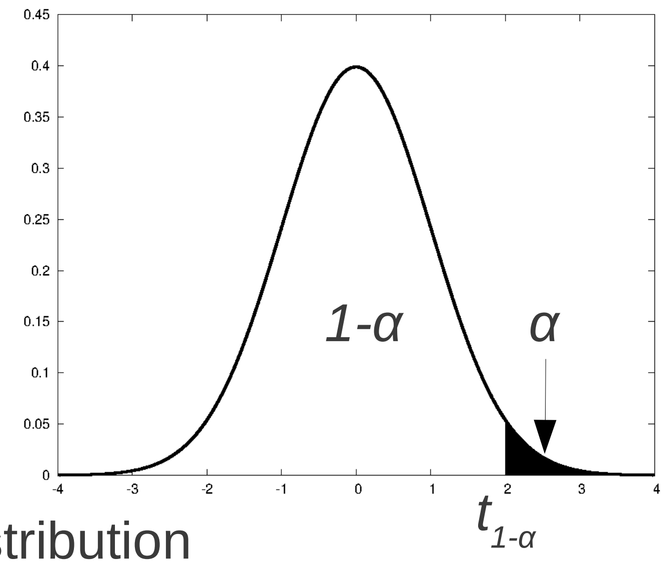
The decision to be made during the optimization, in the presence of noise, is (1 is current point , 2 the new point)

$$\frac{1}{s}\sum_{i=1}^{s} f\left(x^1, u_i\right) \quad \overset{?}{>, =, <} \quad \frac{1}{s}\sum_{j=1}^{s} f\left(x^2, u_j\right)$$

(same number of samples *s* to keep formula simple)

Test : *H0* the new point is better than the current one

$$HO \quad , \quad E\left(f\left(x_1, U\right)\right) \geqslant E\left(f\left(x_2, U\right)\right)$$
$$H1 \quad , \quad E\left(f\left(x_1, U\right)\right) < E\left(f\left(x_2, U\right)\right)$$



Statistic : Accept $H0$ if $\dfrac{M_2 - M_1}{\sqrt{V_1/s + V_2/s}} < t_{1-\alpha}$,

otherwise reject $H0$

where $t_{1-\alpha}$ is the $(1-\alpha)$'s quantile of a t-distribution

α is the error rate at which $H0$ is wrongly rejected,

M and *V* are the empirical averages and variances,

$$M_{1,2} = \frac{1}{s}\sum_{i=1}^{s} f\left(x^{1,2}, u_i\right) \quad , \quad V_{1,2} = \frac{1}{s-1}\sum_{i=1}^{s}\left(f\left(x^{1,2}, u_i\right) - M_{1,2}\right)^2$$

ES and hypothesis testing

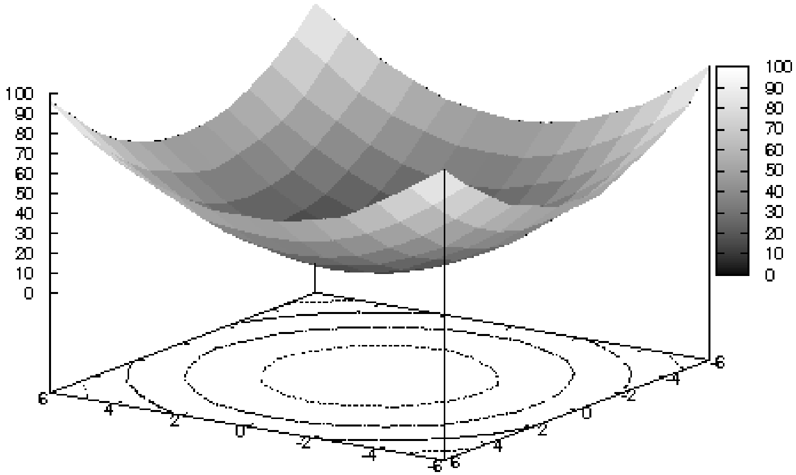
The simplest ES-(1+1) evolutionary optimizer improved by hypothesis testing.

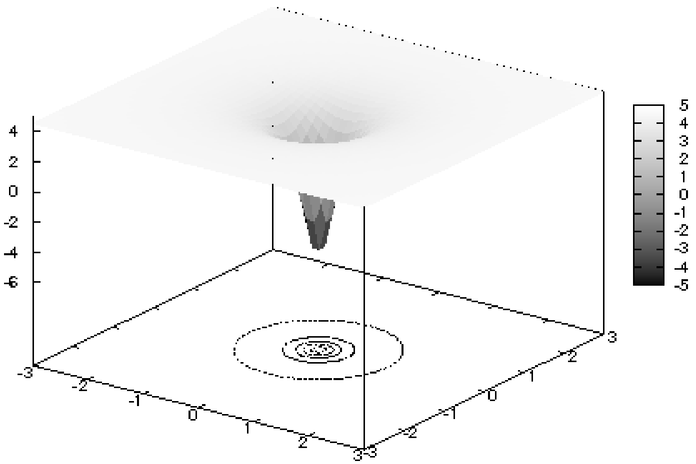
```
while cost < cost_max do
    x' = x(t) + σ N(0,I)
    calculate i.i.d. samples f(x',u^i) , i = 1,s
    cost = cost + s
    Hypothesis testing :
    H0 , Ef(x',U) ≤ Ef(x(t),U)  against  H1 , Ef(x',U) > Ef(x(t),U)
    Reject H0 with error α  ?
        Yes : x(t+1) = x(t)
        No :  x(t+1) = x'
    t = t+1
end
```

- α allows to change continuously the behavior of the optimizer from exploratory (low α) to conservative (high α).
- Three parameters, whose optimal values are coupled : σ , s , α

Test functions

On unimodal noisy functions : test convergence speed in noise, not globality of the search





$$F_Q^{\text{ideal}}(x) = q_{90}\left(f_Q(x, U)\right) = q_{90}\left(\|x+U\|^2\right)$$

$$F_H^{\text{ideal}}(x) = q_{90}\left(f_H(x, U)\right) = q_{90}\left(\frac{-1}{\|x\|^2+0.1}+U\right)$$

decreasing signal / noise ratio as x → x* (=0)

increasing signal / noise ratio as x → x* (=0)

Parameters of the tests

σ : the optimizer step size (=0.05 to 4)
nb **: number of batches (=*s*). High noise when *nb=2*, little noise when *nb=50*.**
α : first type error rate where H0 is « the new point is better than the current one ».

- ***α=0.1*** **: exploratory optimizer**
- ***α=0.5*** **: traditional optimizer (no hypothesis testing)**
- ***α=0.9*** **: conservative optimizer**

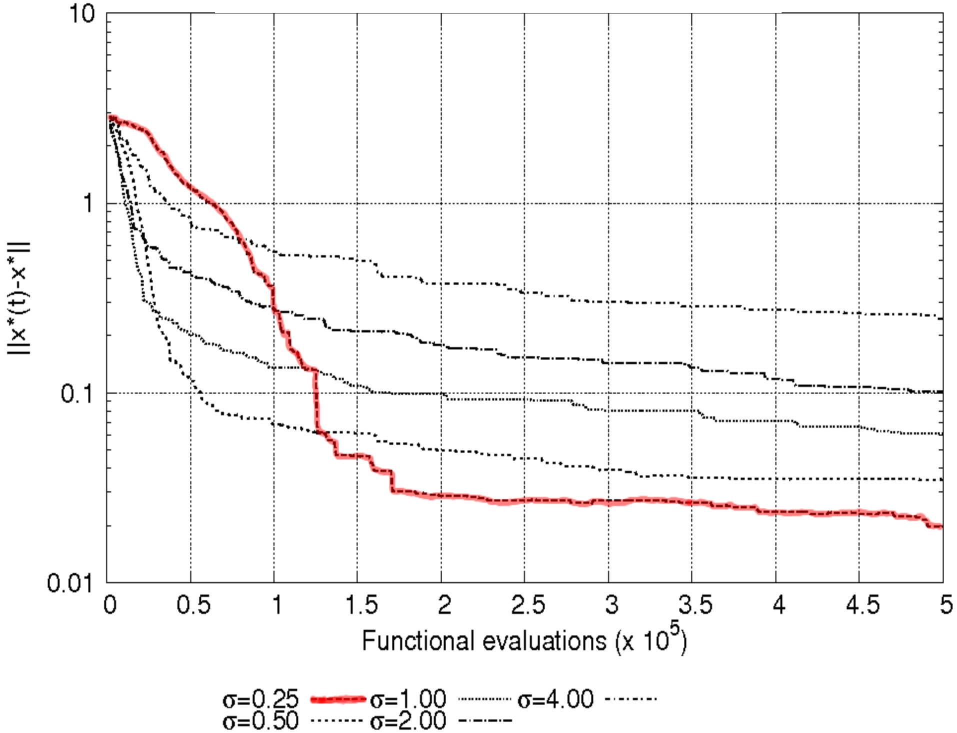
(but also,
n : number of variables, *n=2* or *10*. *n=2* here, results generalize.
Crude Monte Carlo (shown here) versus Latin Hypercube Sampling. Prefer LHS.
)

Each optimization is started from $\{2\}^n$, *500,000* calls to *f* long , and repeated 30 times.
b = 20 (fixed, smallest number for a Gaussian percentile)

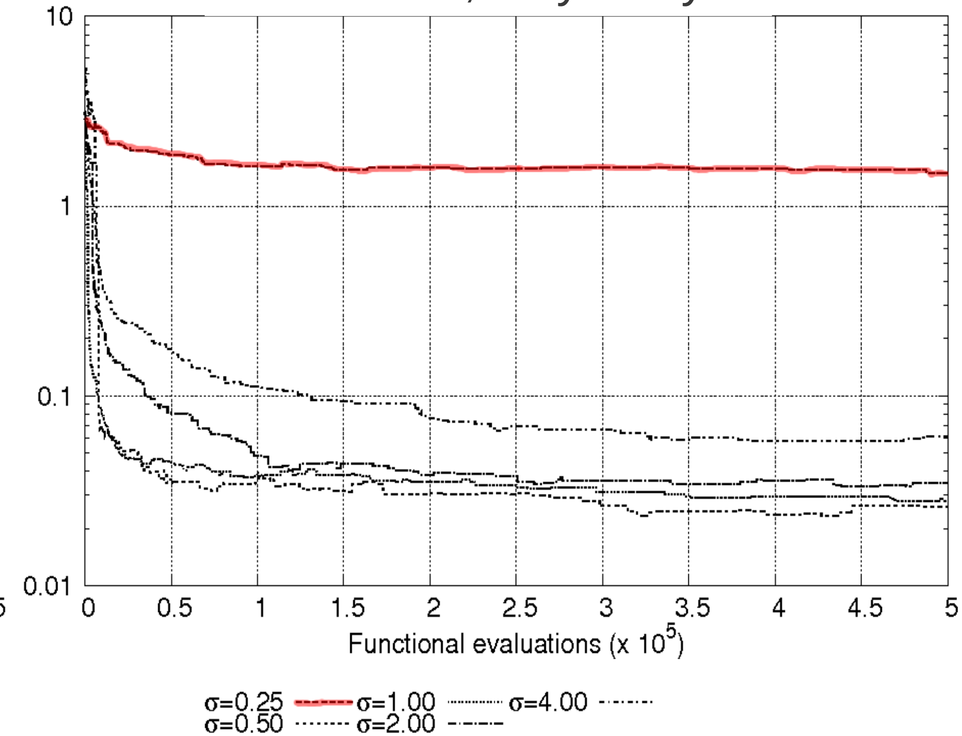
Number of MC simulations and step size

Expl. on F_H, traditional optimizer, the smallest step size ($\sigma = 0.25$) overlined in red.

nb = 50 : costly MC simulations, little noise



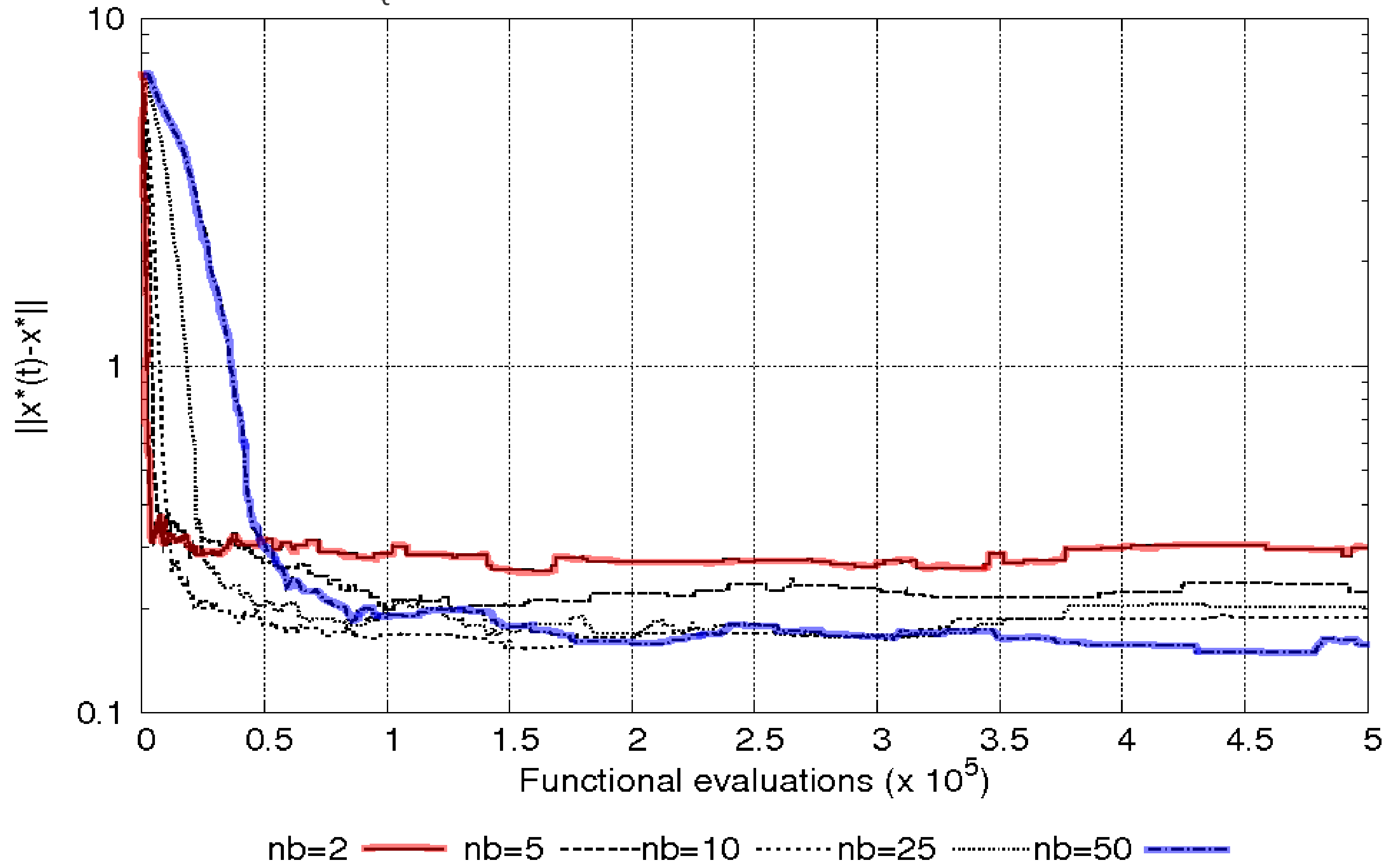
nb = 2 : cheap MC simulations, very noisy



Lower bound on step sizes needs to increase with the noise (decreasing *nb*) to prevent comparison errors

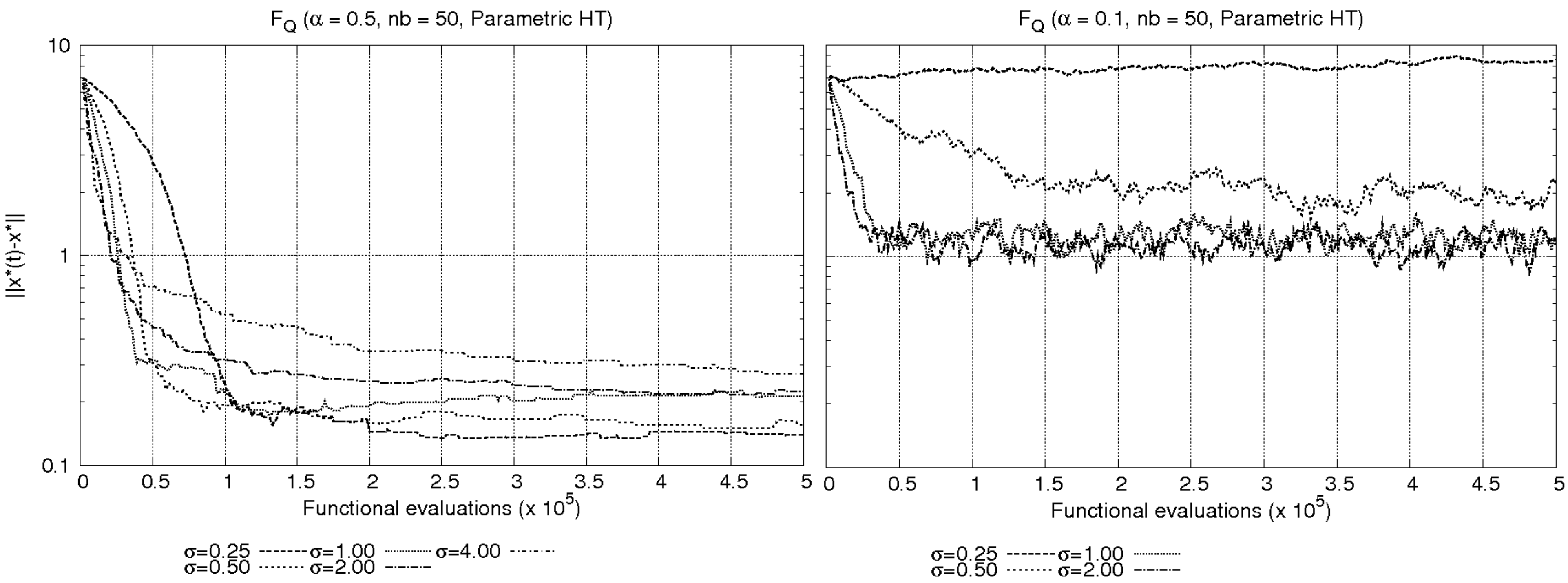
Number of MC simulations

Expl. on F_Q, traditional optimizer, step size $\sigma = 0.5$.



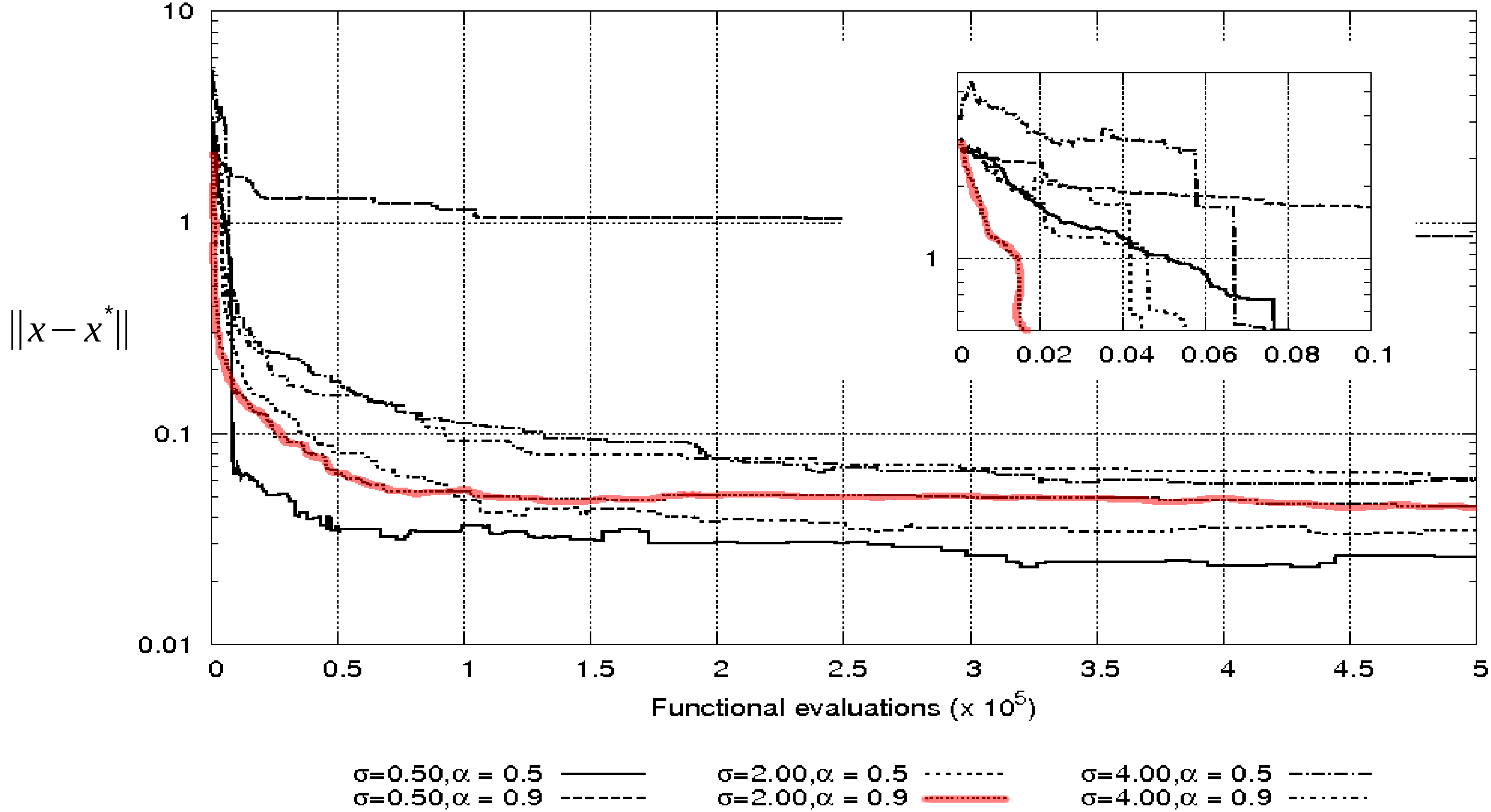
Low MC number at the beginning (*nb = 2*), increase *nb* later to converge accurately (observed on F_Q and F_H).

Traditional vs. exploratory optimizer



The exploratory optimizer (with HT, α=0.1) tends to diverge and never converges faster on FQ and FH
→ not useful on non-deceptive functions.

Conservative vs. traditional optimizer



Flat initial region (F_H) with high probability of being mislead (*nb = 2*)
→ the conservative optimizer is initially the best.
In all other tests made, traditional optimizer is better.

Noisy optimization – Summary

- min_x *f(x,U)*

- Use general stochastic (evolutionary) optimizers, which can be relatively robust to noise if properly tuned.
- Useful for optimizing statistical estimators which are noisy.

- No control over the U's
- No spatial statistics (i.e. in S or $S \times U$ spaces), pointwise approaches only.

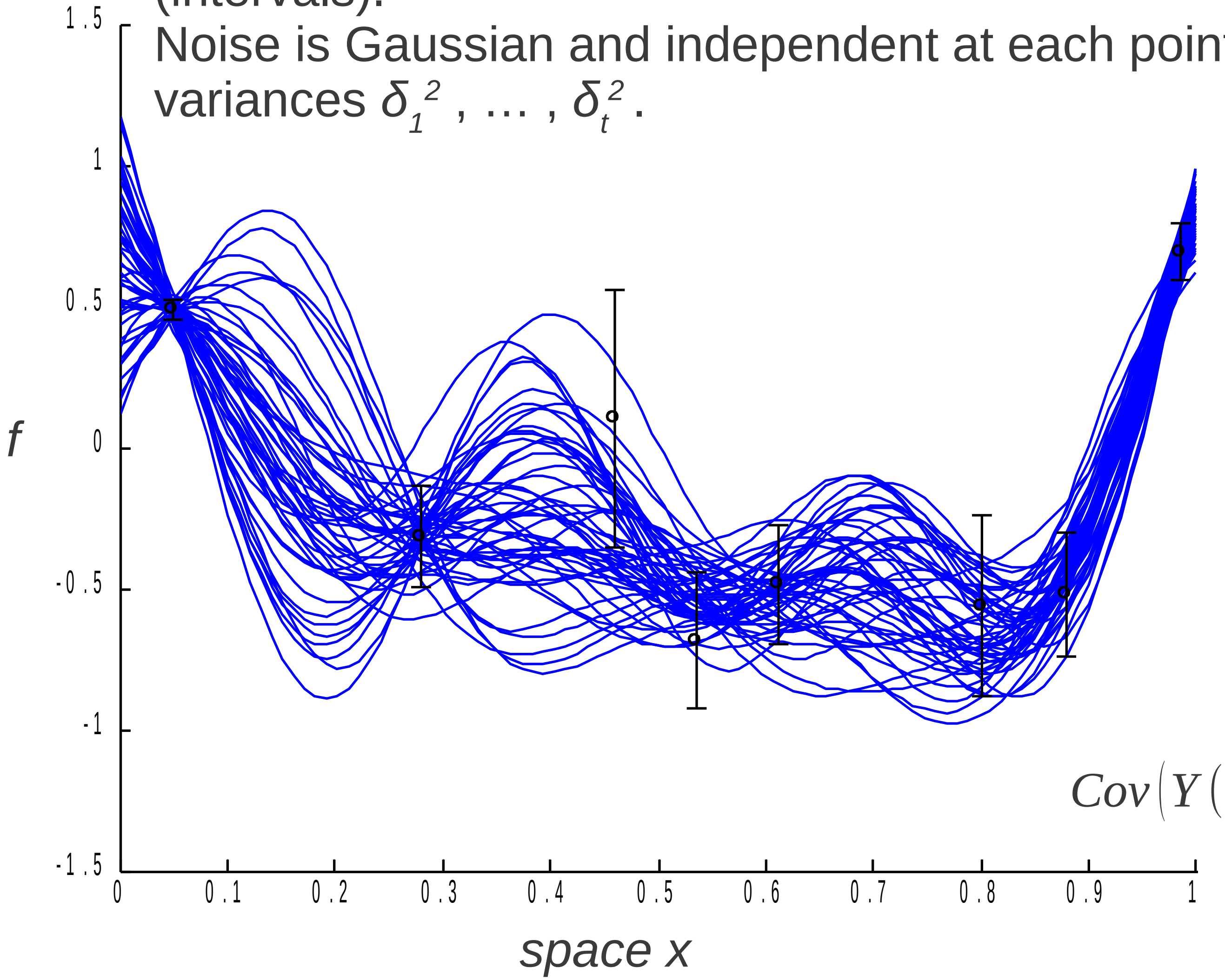
- Next : introduce spatial statistics to filter the noise → kriging based approaches.

Outline of the talk

1. Introduction to optimization
2. Formulations of optimization problems with uncertainties
3. Noisy optimization
4. Kriging-based approaches

Kriging : quick intro (1)

black circles : observed values , $f(x^1)$, … , $f(x^t)$, with heterogeneous noise (intervals).
Noise is Gaussian and independent at each point (nugget effect), variances δ_1^2 , … , δ_t^2 .



Assume : the blue curves are possible underlying true functions.
They are instances of stationary Gaussian processes *Y(x)* → fully characterized by their average *μ* and their covariance,

$$Cov\left(Y(x), Y(x')\right) = \text{Fct}\left(\text{dist}(x, x')\right)$$

Kriging : quick intro (2)

$f(x)$ represented by $Y^t(x) = [\, Y(x) \,|\, f(x^1), \ldots, f(x^t) \,]$

$Y^t(x) \sim N\left(m_K(x), s_K^2(x)\right)$ (simple kriging)

Kriging average : $m_K(x) = \mu + \boldsymbol{c}^T(x)\, \boldsymbol{C}_\Delta^{-1} (\boldsymbol{f} - \mu \boldsymbol{1})$

Kriging variance : $s_K^2(x) = \sigma^2 - \boldsymbol{c}^T(x)\, \boldsymbol{C}_\Delta^{-1}\, \boldsymbol{c}(x)$

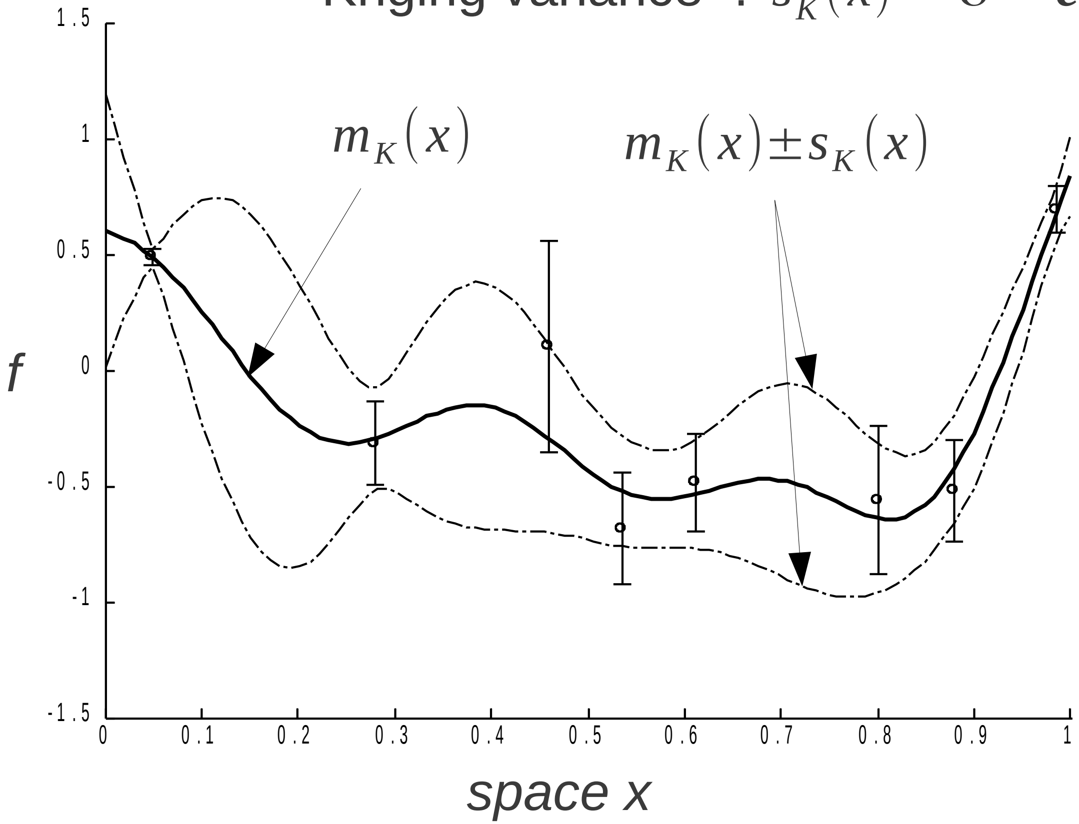
where

$\boldsymbol{c}(x) = \left[Cov(Y(x), Y(x^i)) \right]_{i=1,t}$

$\boldsymbol{C}_\Delta = \boldsymbol{C} + \boldsymbol{\Delta}$

$\boldsymbol{C} = \left[Cov(Y(x^i), Y(x^j)) \right]_{i,j}$

$\boldsymbol{\Delta} = diag[\delta_1^2 \,, \ldots ,\, \delta_t^2]$



Outline of the talk

1. Introduction to optimization
2. Formulations of optimization problems with uncertainties
3. Noisy optimization
4. Kriging-based approaches
   - No control on *U*
   - With control on *U*

No control on *U*

No control on *U*.

The variance of the observations $f(x^i)$, **Δ** , is

1. Estimated from the context
Expl : variance of a statistical estimator,

Average f : $V\left(\overline{f(x)}\right) = \frac{1}{s(s-1)} \sum_{i=1}^{s} \left(f(x, u_i) - \overline{f(x)}\right)^2$

Quantile of *f* : cf. Le Riche et al., *Gears design with shape uncertainties using Monte Carlo simulations and kriging*, SDM, AIAA-2009-2257.

2. Learned from data
By maximizing the likelihood of the data (for **Δ** and ***C*** parameters).
Cf. Roustant, O. et al., DiceKriging, DiceOptim : two R packages for the analysis of computer experiments by kriging based metamodeling and optimization, HAL, 2010

Kriging prediction minimization

The simplest approach.

For $t=1,t^{max}$ do,

Learn $Y^t(x)$ (m_K and s_K^2) from $f(x^1), \ldots, f(x^t)$

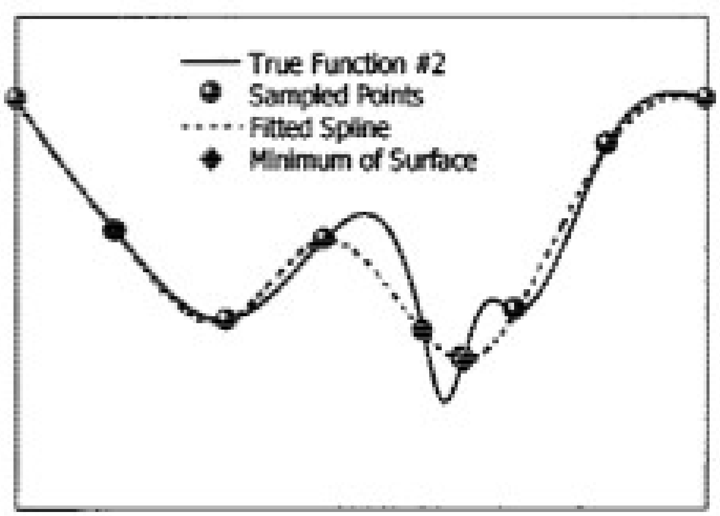
$x^{t+1} = \min_x m_K(x)$

Calculate $f(x^{t+1})$

$t = t+1$

End For

e.g., using CMA-ES because multimodal (Krisp toolbox in Scilab)



But it may fail : the minimizer of m_K is at a data point which is not even a local optimum.

D. Jones, A taxonomy of global optimization methods based on response surfaces, JOGO, 2001.

Expected Improvement criterion (1)

A sampling criterion for global optimization without noise :

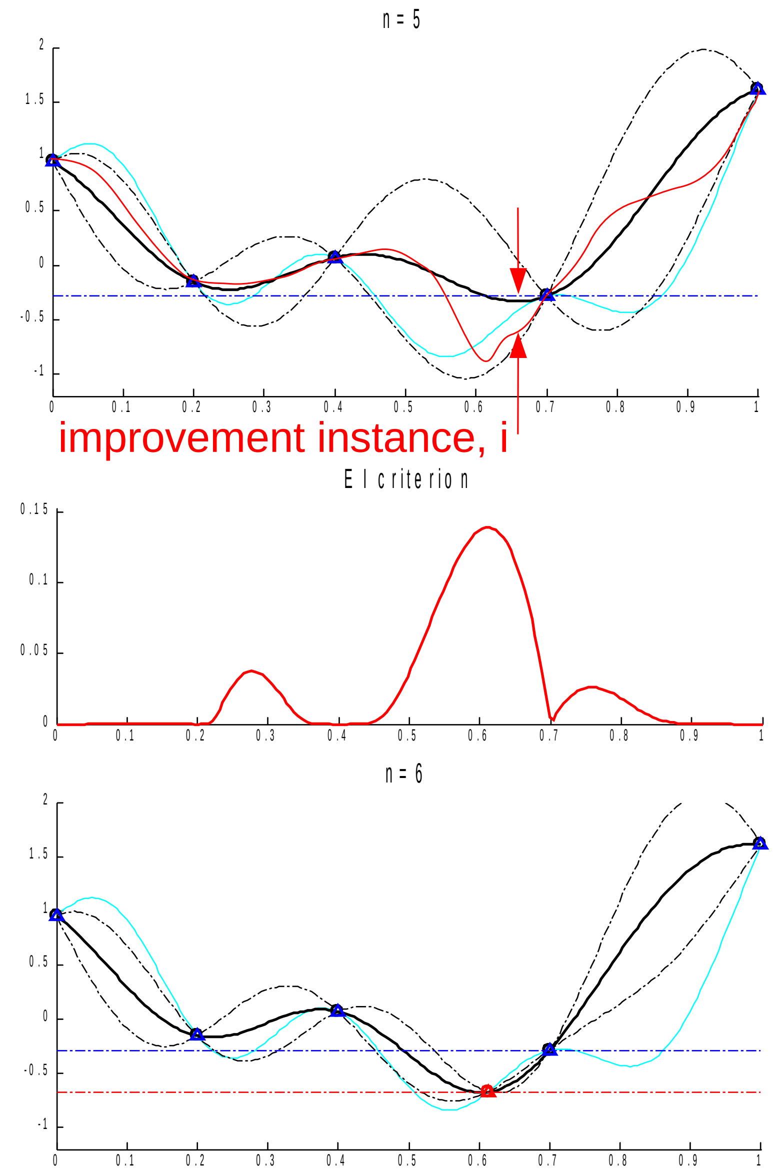
Improvement at x , $I(x) = max(y_{min} - Y(x), 0)$

The expected improvement, *EI(x)* , can be analytically calculated.

$$EI(x) = s(x)\left[\, a(x)\Phi\big(a(x)\big) + \phi\big(a(x)\big) \right] ,$$

$$a(x) = \frac{y_{min} - m(x)}{s(x)}$$

EI increases when m_K decreases and when s_K increases. *EI(x)* quantifies the exploration-exploitation compromise of global optimization.



Expected Improvement criterion

A sampling criterion for global optimization without noise :

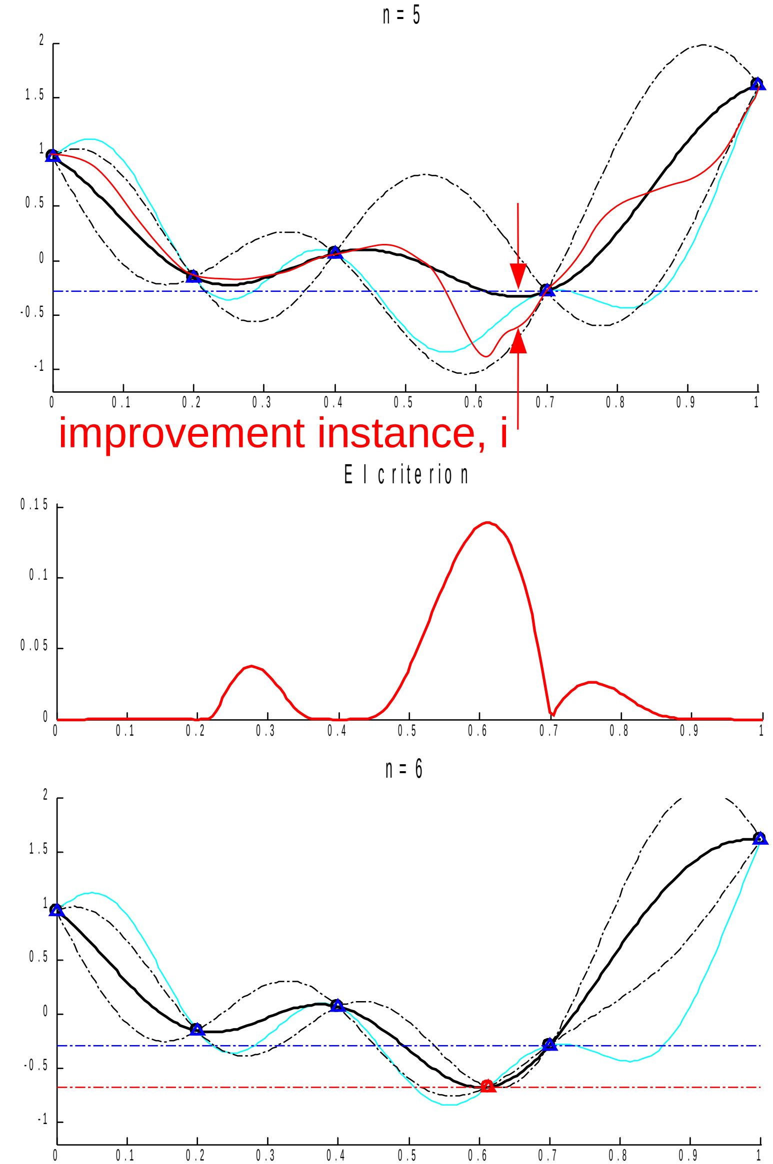
Improvement at x , $I(x) = max(y_{min} - Y(x), 0)$

The expected improvement, *EI(x)* , can be analytically calculated. *EI* increases when m_K decreases and when s_K increases. *EI(x)* quantifies the exploration-exploitation compromise of global optimization.

Next iterate : $x^{t+1} = max_x EI(x)$

EGO (Efficient Global Optimization) algorithm, D. Jones, 1998 .

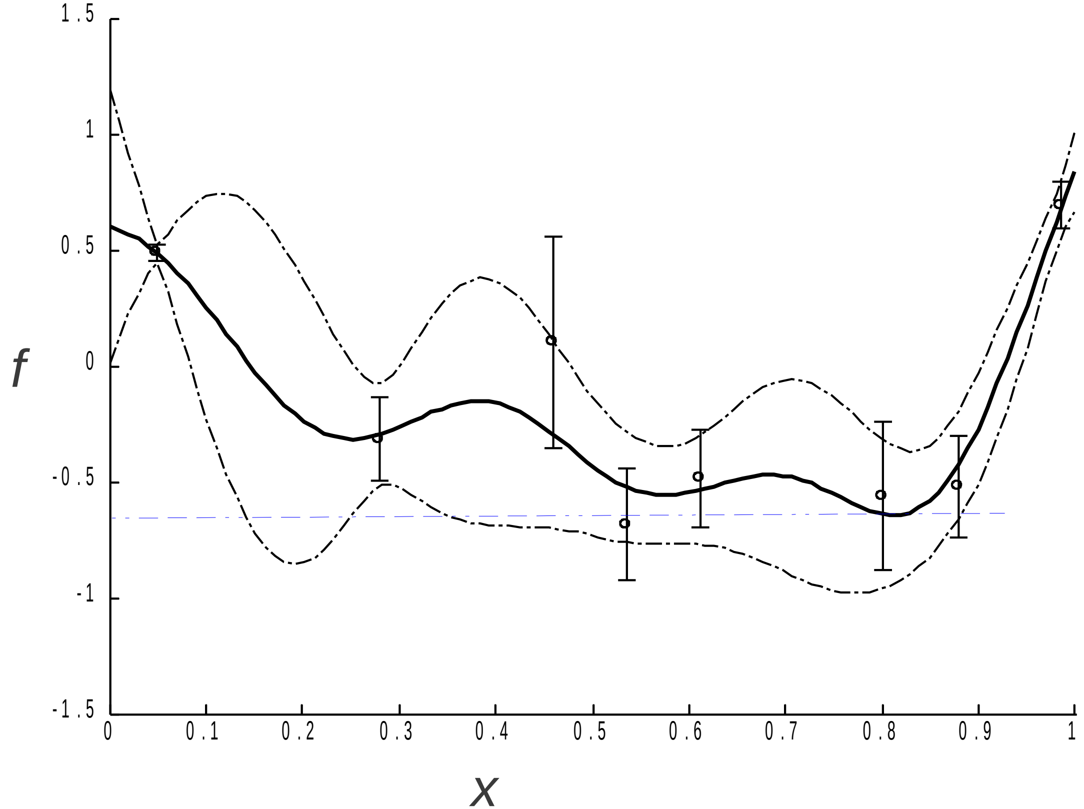
Not suitable for noisy functions because the noise can make y_{min} too small and create premature convergence around y_{min}.



EI for noisy functions

Solution 1 : Add nugget effect and replace y_{min} by the best observed mean (filters out noise in already sampled regions) :

$$EI^{\text{noisy}}(x) = E\left[max\left(\min_{i=1,t} m_K(x^i) - Y(x)\ ,\ 0 \right) \right]$$



Expected Quantile Improvement

V. Picheny, D. Ginsbourger, Y. Richet, *Optimization of noisy computer experiments with tunable precision*, Technometrics, 2011.

Solution 2 : Add nugget effect and use the expected quantile improvement. A conservative criterion (noise and spatial uncertainties are seen as risk rather than opportunities).

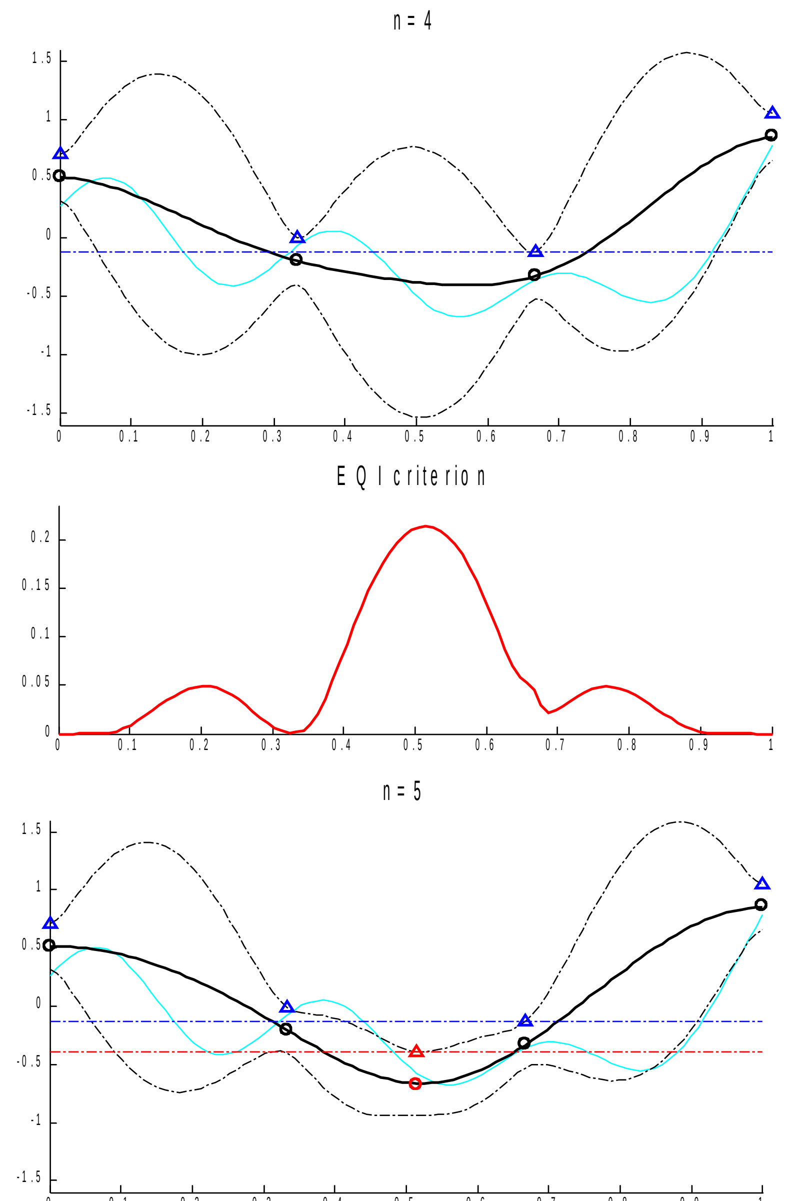
$$EQI(x) = E\left[max\left(q_{min} - Q^{t+1}(x)\ ,\ 0\right)\right]$$

$$q_{min} = \min_{i=1,t}\ m_K(x^i) + \alpha\, s_K(x^i)$$

$$Q^{t+1}(x) = m_K^{t+1}(x) + \alpha\, s_K^{t+1}(x)$$

$m_K^{t+1}(x)$ is a linear function of $Y(x)$

$\Rightarrow EQI(x)$ is known analytically



Related work

E. Vazquez, J. Villemonteix, M. Sidorkiewicz and E. Walter, *Global optimization based on noisy evaluations: an empirical study of two statistical approaches*, 6th Int. Conf. on Inverse Problems in Engineering, 2010.

J. Bect, *IAGO for global optimization with noisy evaluations,* workshop on noisy kriging-based optimization (NKO), Bern, 2010.

Outline of the talk

1. Introduction to optimization

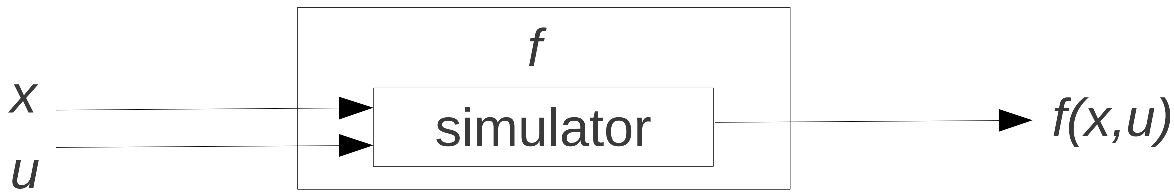
2. Formulations of optimization problems with uncertainties

3. Noisy optimization

4. Kriging-based approaches
   No control on U
   ➡ With control on U

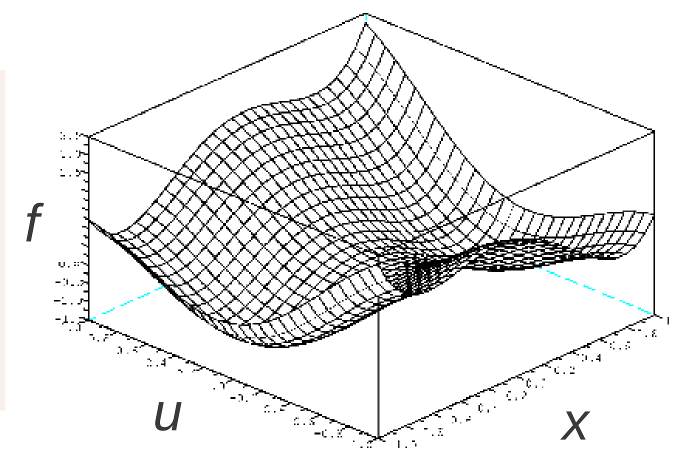
U control : uncertainty propagation

x and *u* can be chosen before calling the simulator and calculating the objective function. This is the general case.



Direct approaches to optimization with uncertainties have a double loop : propagate uncertainties on *U*, optimize on *x*.

- Optimization : loop on *x*
  - Estimation of the performance (average, std dev, percentile of *f(x,U)*) : loop on *u* , Monte Carlo



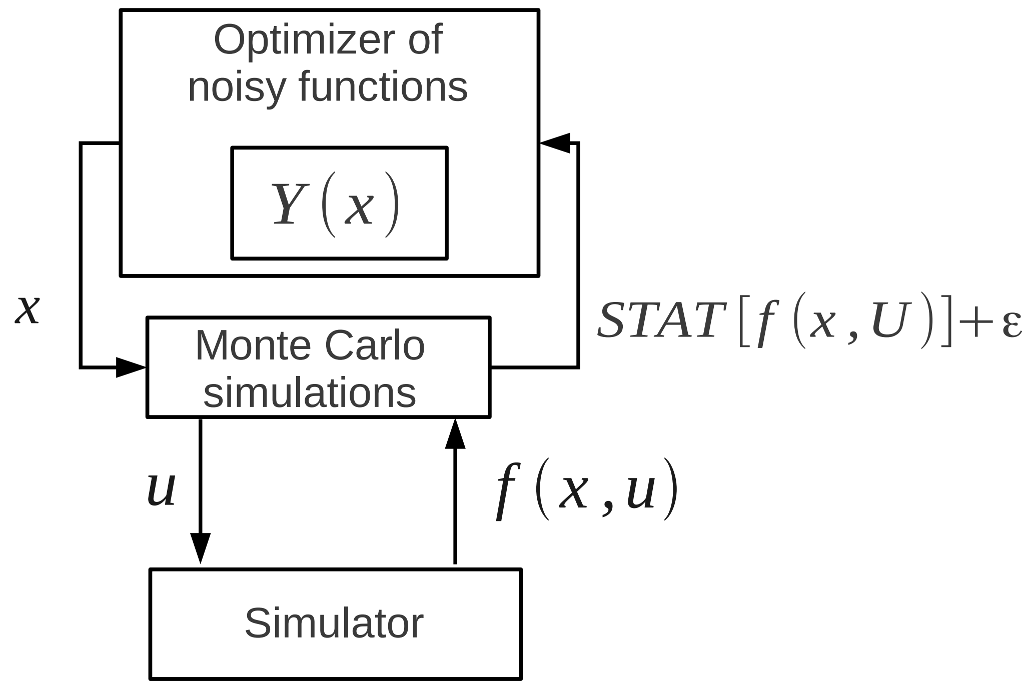
Such a double loop is very costly (more than only p
uncertainties or optimizing, which are already considered as costly) !

(x,u) surrogate based approach

Assumptions : *x* and *U* controlled

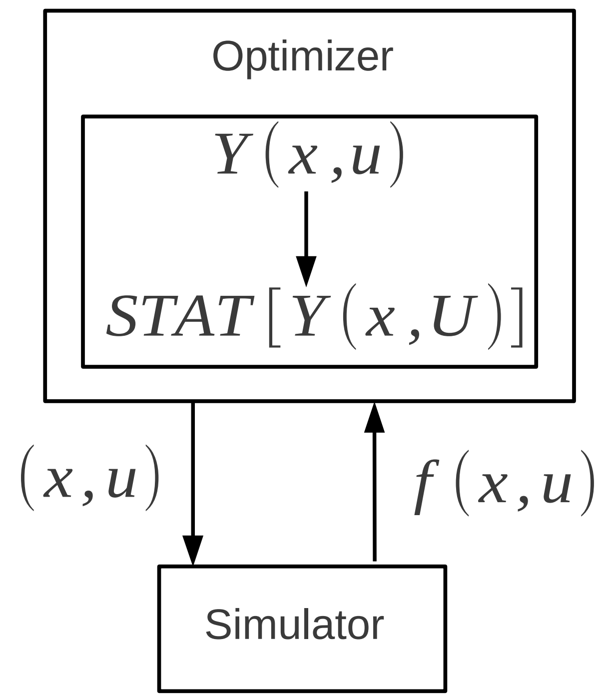
Y : surrogate model

Direct approach



Multiplicative cost of two loops involving *f*

(x,u) surrogate based approach



Only one loop of *f*

A general Monte Carlo - kriging algorithm

Hereafter is an example of a typical surrogate-based (here kriging) algorithm for optimizing any statistical measure of $f(x,u)$ (here the average).

MC – kriging algorithm

Create initial DOE (X^t,U^t) and evaluate f there ;
While stopping criterion is not met:

- Create kriging approximation Y^t in the joint (x,u) space from $f(X^t,U^t)$
- Estimate the value of the statistical objective function from Monte Carlo simulations on the kriging average $m_Y^{\,t}$.

  Expl : $\hat{f}(x^i) = \frac{1}{s}\sum_{k=1}^{s} m_K^t(x^i, u^k)$, where u^k i.i.d. from pdf of U
- Create kriging approximation Z^t in x space from $\left(x^i , \hat{f}(x^i)\right)_{i=1,t}$
- Maximize $EI_Z(x)$ to obtain the next simulation point $\rightarrow x^{t+1}$
  u^{t+1} sampled from pdf of U

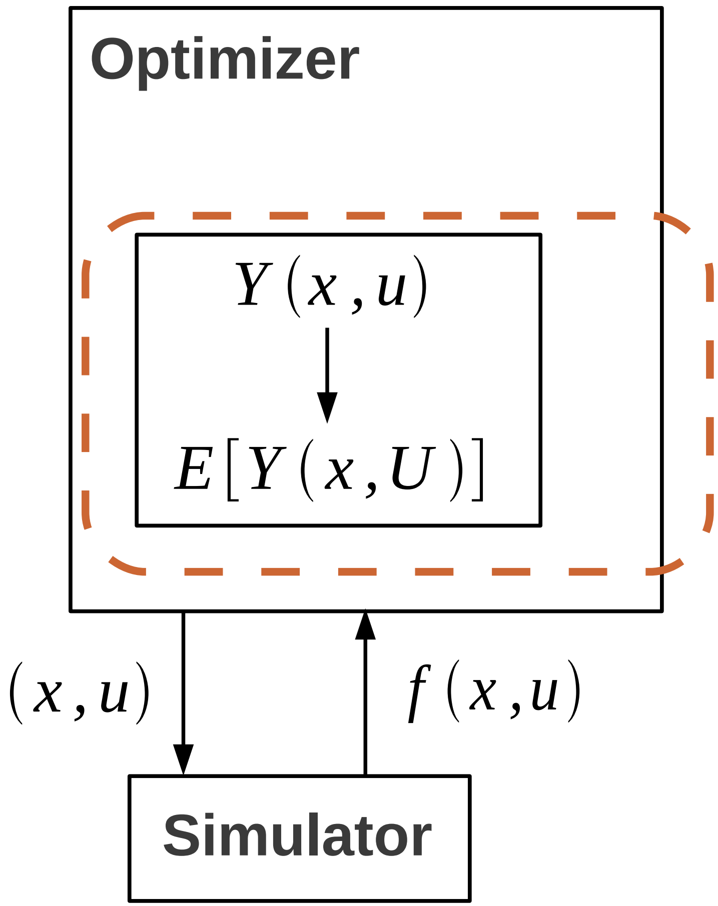
  only call to f !
- Calculate simulator response at the next point, $f(x^{t+1},u^{t+1})$.
  Update DOE and t

Simultaneous optimization and sampling (1)

J. Janusevskis and R. Le Riche, Simultaneous kriging-based sampling for optimization and uncertainty propagation, ROADEF 2011 and HAL report.

Assumptions : *x* and *U* controlled, *U* normal. Solve

Y : kriging model



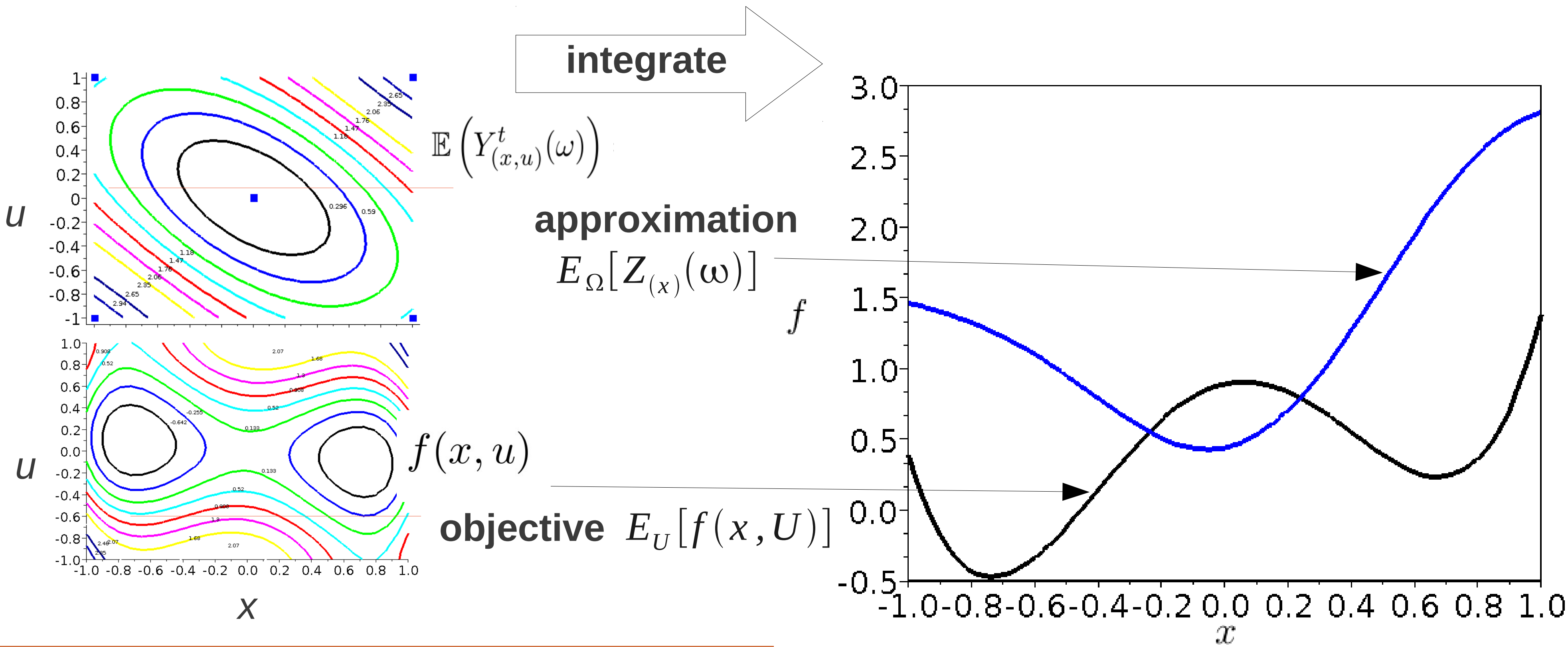
1. Building internal representation of the objective (mean performance) by «integrated» kriging.

Integrated kriging (1)

$\min_x \mathbb{E}_U[f(x,U)]$: **objective**

$Y^t_{(x,u)}(\omega)$: **kriging approximation to deterministic** $f(x,u)$

$Z^t_{(x)}(\omega) = \mathbb{E}_U[Y^t_{(x,U)}(\omega)]$: **integrated process approximation to** $\mathbb{E}_U[f(x,U)]$



Integrated kriging (2)

The integrated process over *U* is defined as

$$Z_{(x)}(\omega) = \mathbb{E}_U[Y^t_{(x,U)}(\omega)] = \int_{\mathbb{R}^m} Y^t_{(x,u)}(\omega) d_\mu(u)$$

$d\mu(u)$ -probability measure on *U*

Because it is a linear transformation of a Gaussian process, it is Gaussian, and fully described by its mean and covariance

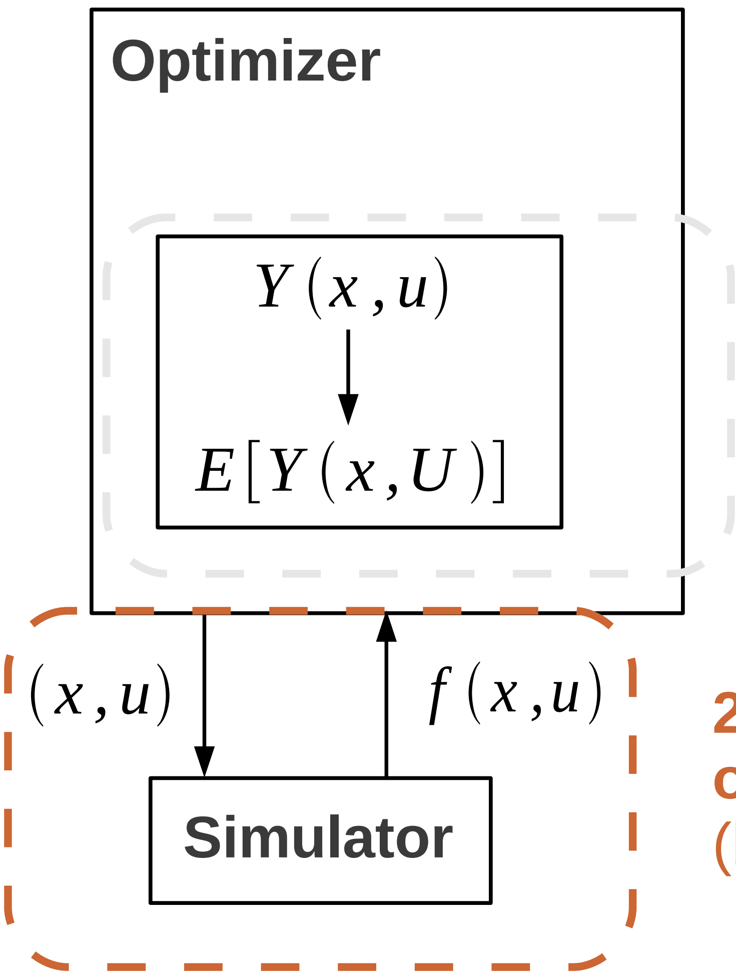
$$m_Z(x) = \int_{\mathbb{R}^m} m_Y(x,u) d\mu(u)$$

$$cov_Z(x;x') = \int_{\mathbb{R}^m} \int_{\mathbb{R}^m} cov_Y(x,u;x'u') d\mu(u) d\mu(u')$$

Analytical expressions of m_z and cov_z for Gaussian *U*'s are given in

J. Janusevskis, R. Le Riche. Simultaneous kriging-based sampling for optimization and uncertainty propagation, HAL report: hal-00506957

Simultaneous optimization and sampling (2)



1. Building internal representation of the objective (mean performance) by «projected» kriging.

2. Simultaneous sampling and optimization criterion for x and u
(both needed by the simulator to calculate f)

EI on the integrated process (1)

Z is a process approximating the objective function $\mathbb{E}_U[f(x,U)]$

Optimize with an Expected Improvement criterion,

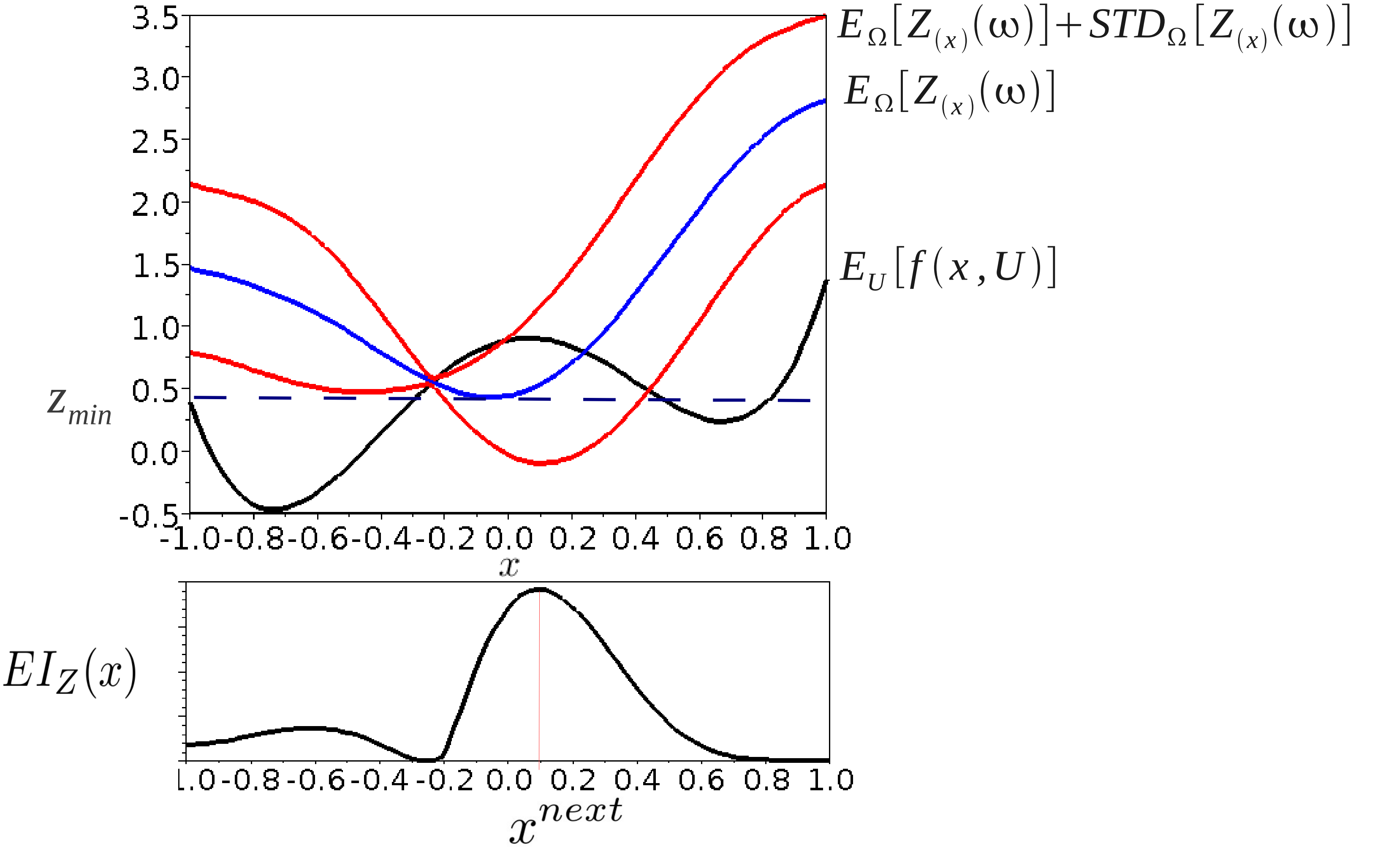
$$x^{next} = \arg\max_x EI_Z(x)$$

Optimize with an Expected Improvement criterion,

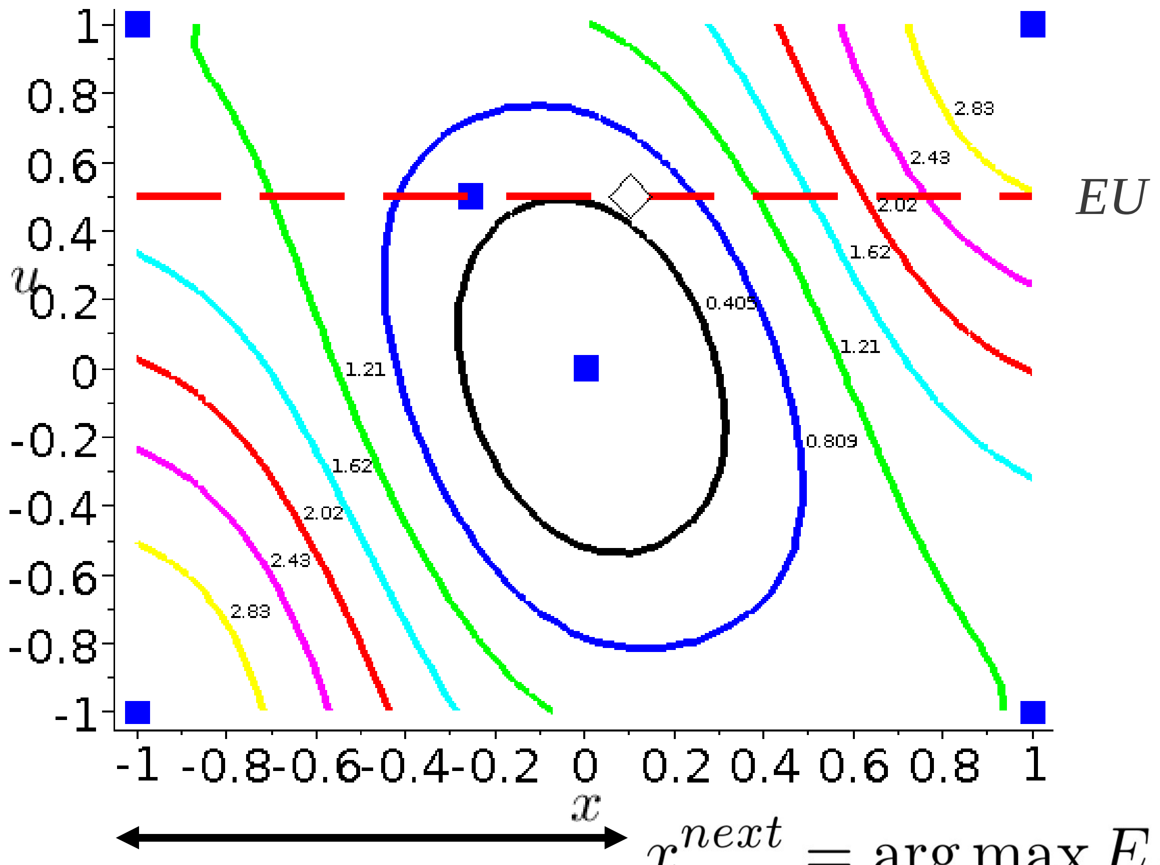
$I_Z(x) = max(z_{min} - Z(x), 0)$, but z_{min} not observed (in integrated space).

$\Rightarrow$ Define $z_{min} = \min_{x^1, \ldots, x^t} E(Z(x))$

EI on the integrated process (2)



EI on the integrated process (3)



$$x^{next} = \arg\max_x EI_Z^t(x)$$

x ok. What about *u* ? (which we need to call the simulator)

Simultaneous optimization and sampling : method

x^{next} gives a region of interest from an optimization of the expected f point of view.

One simulation will be run to improve our knowledge of this region of interest → one choice of (x,u).

Choose (x^{t+1},u^{t+1}) that provides the most information, i.e., which minimizes the variance of the integrated process at x^{next}

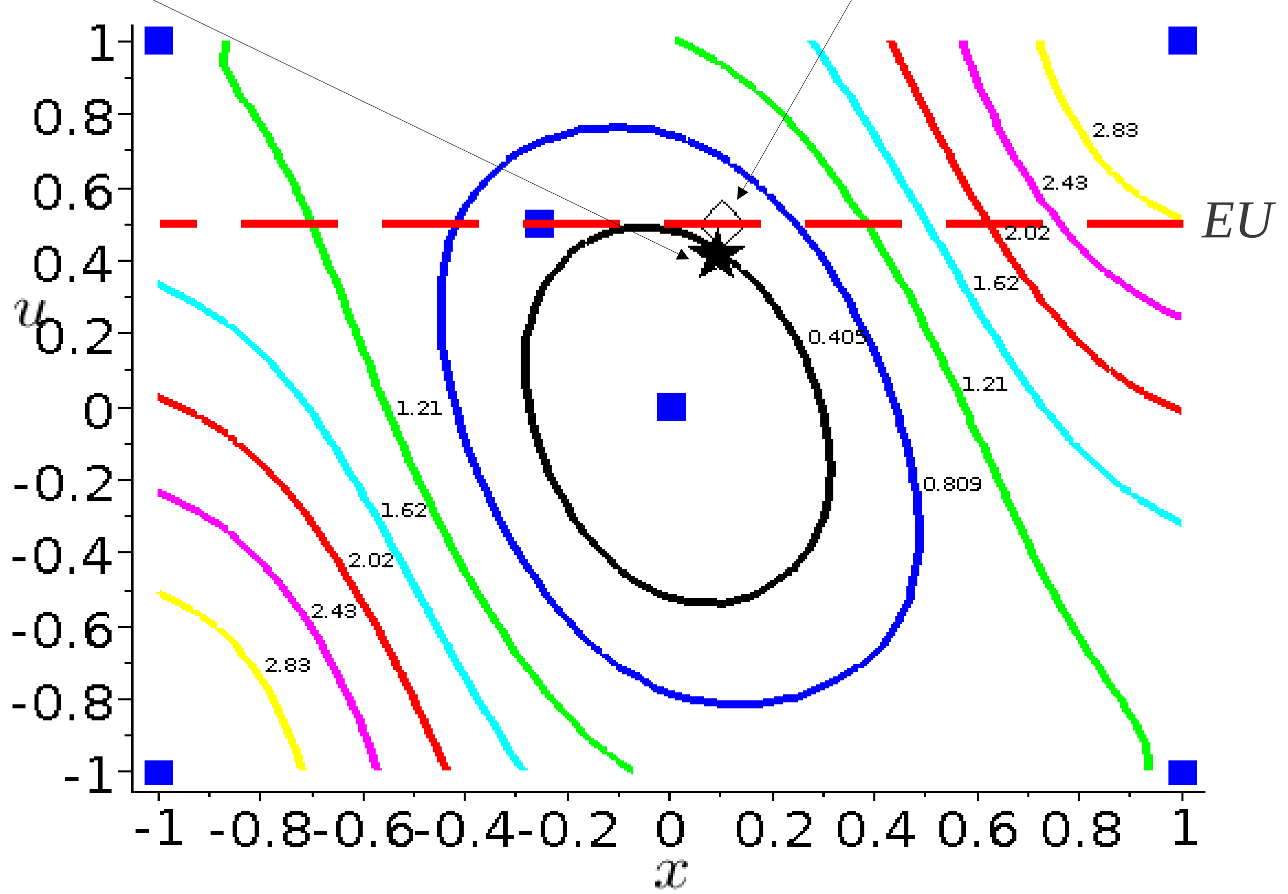
$$(x^{t+1}, u^{t+1}) = \arg\min_{x,u} \mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega)$$

(no calculation details, cf. article. Note that VAR of a Gaussian process does not depend on *f* values but only on *x*'s).

Simultaneous optimization and sampling : expl.

$$x^{next} = \arg\max_{x} EI_Z^t(x)$$

$$(x^{t+1}, u^{t+1}) = \arg\min_{x,u} \mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega)$$



Simultaneous optimization and sampling : algo

Create initial DOE in (x,u) space;

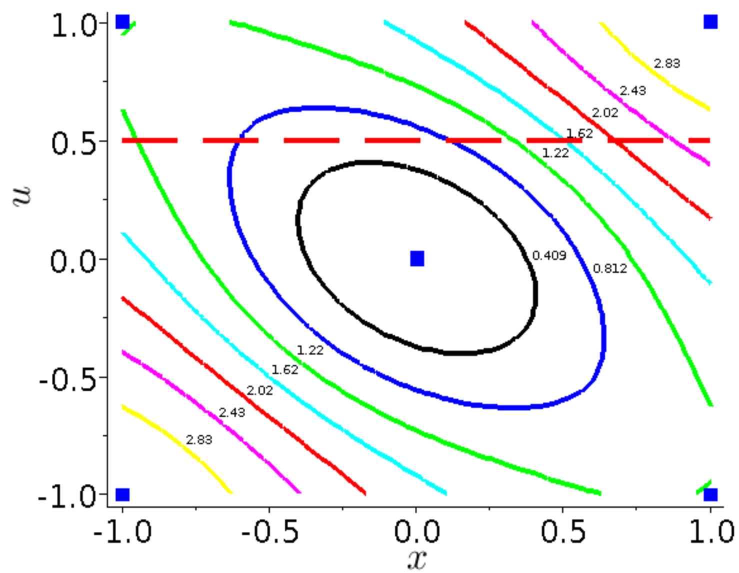
While stopping criterion is not met:

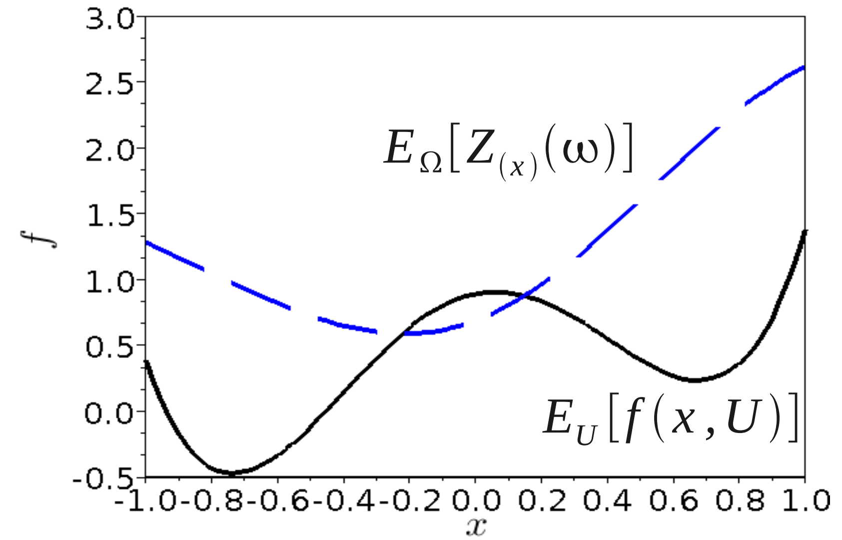
- **Create kriging approximation *Y* in the joint space** (x,u)
- **Calculate the covariance of *Z* from that of *Y***
- **Use EI of Z to choose** (x^{next})
- **Minimize** $VAR(Z(x^{next}))$ **to obtain the next point** (x^{t+1},u^{t+1}) **for simulation**
- **Calculate simulator response at the next point** $f(x^{t+1},u^{t+1})$

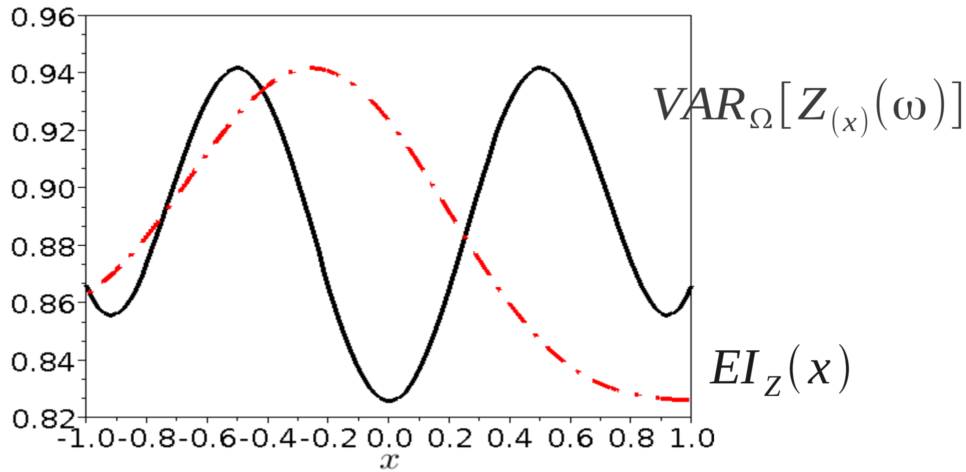
(4 sub-optimizations, solved with CMA-ES)

2D Expl, simultaneous optimization and sampling

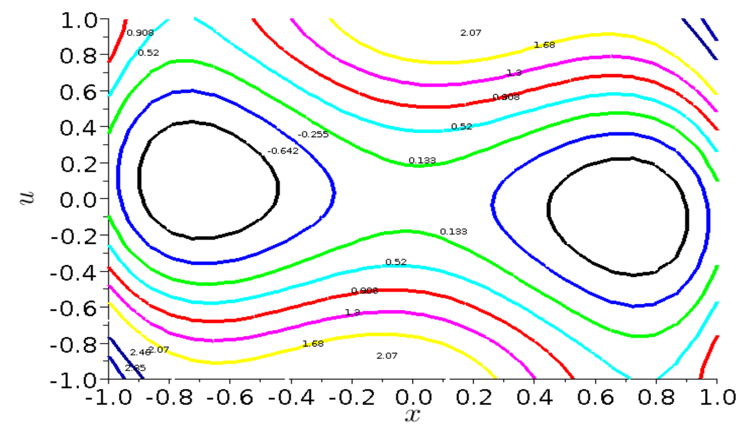
DOE and $E[Y(x,u)]$





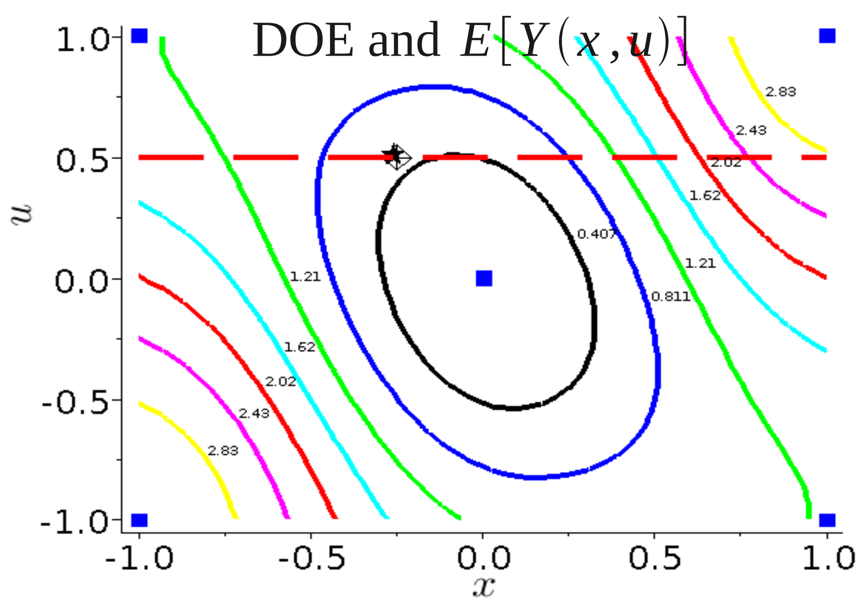


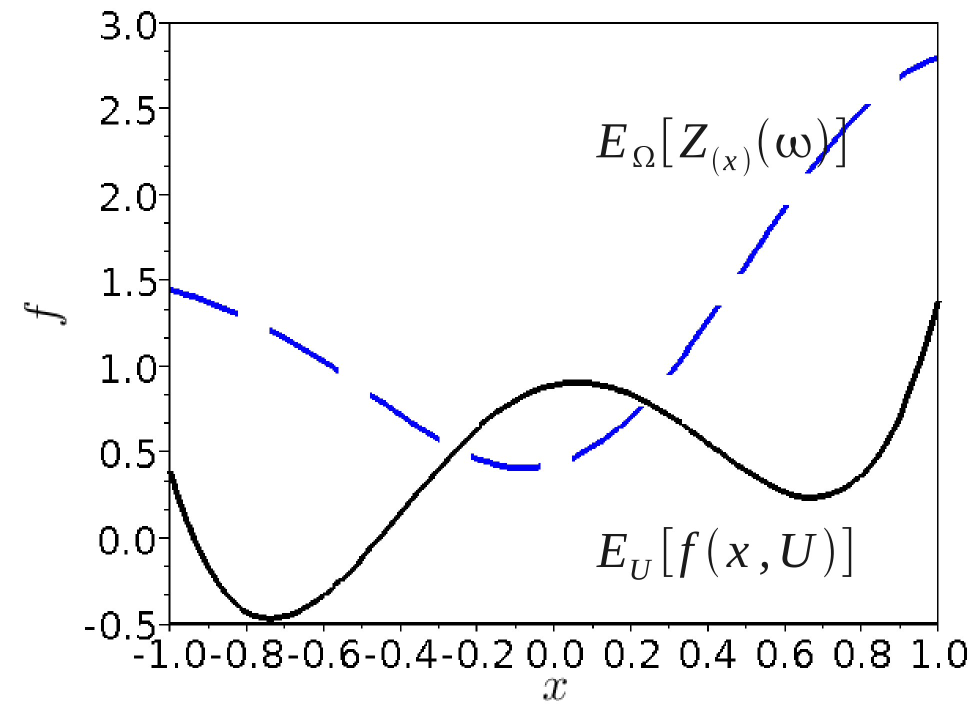
test function

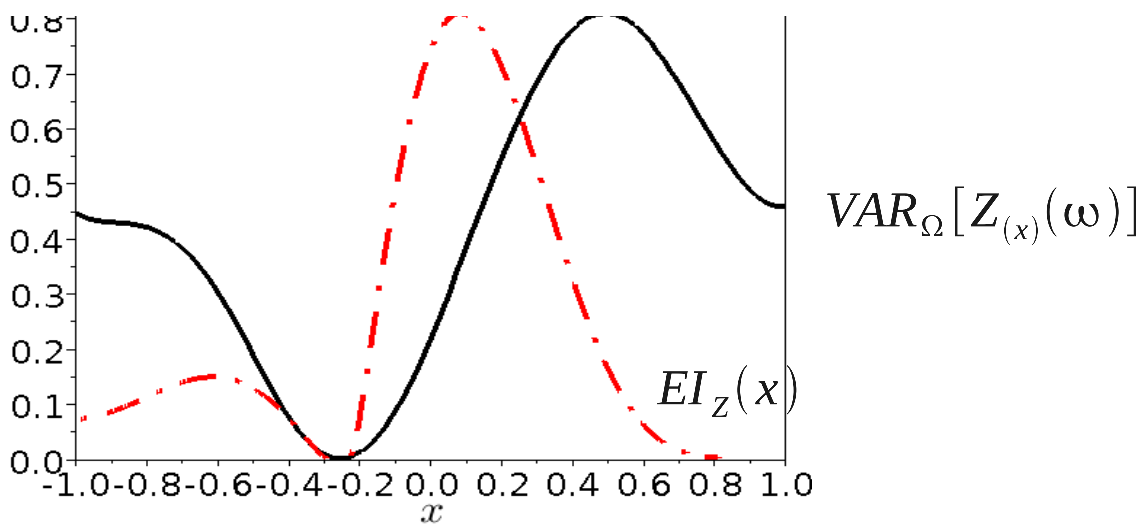


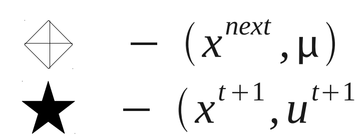
Kriging based optimization with uncertainties, U controlled

1st iteration

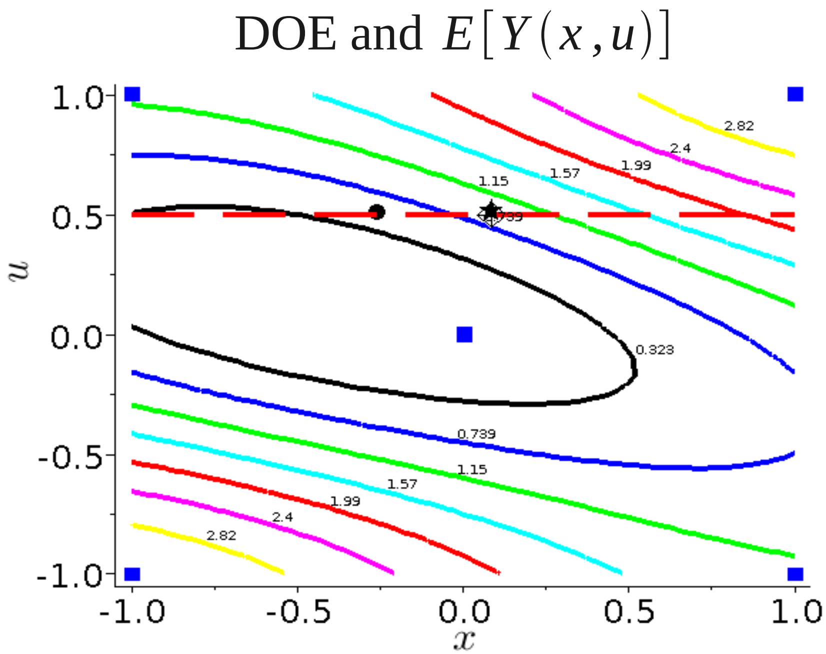


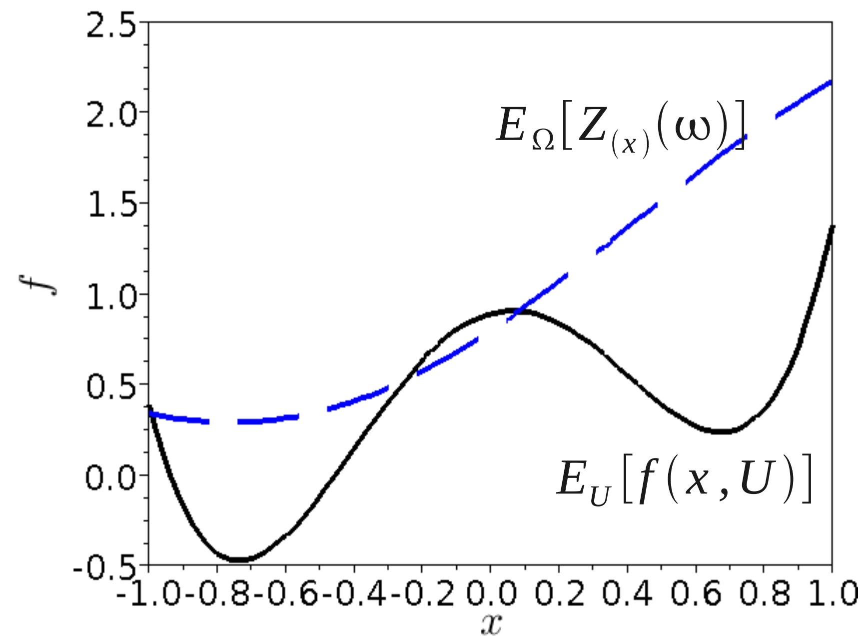


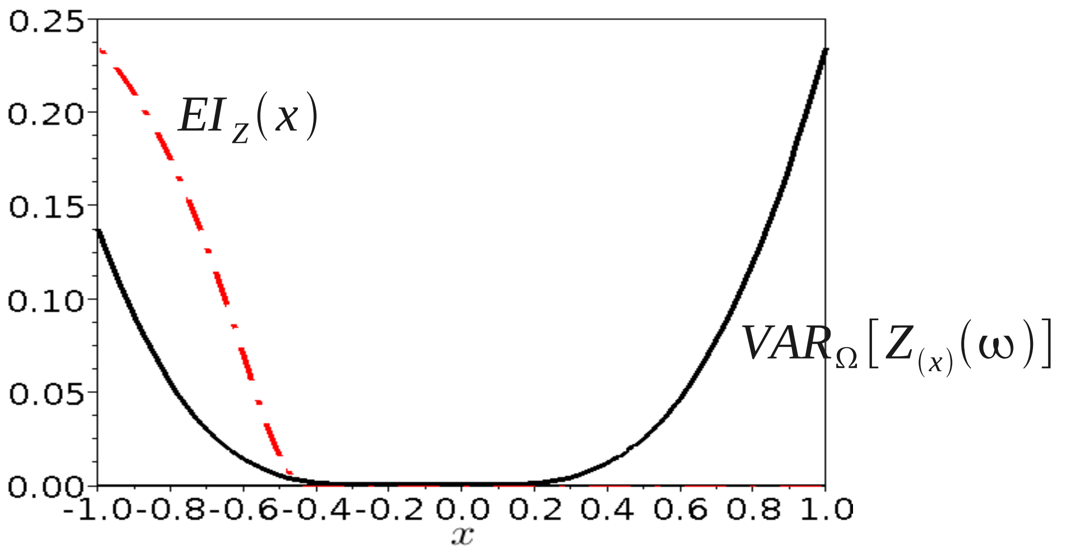




2nd iteration





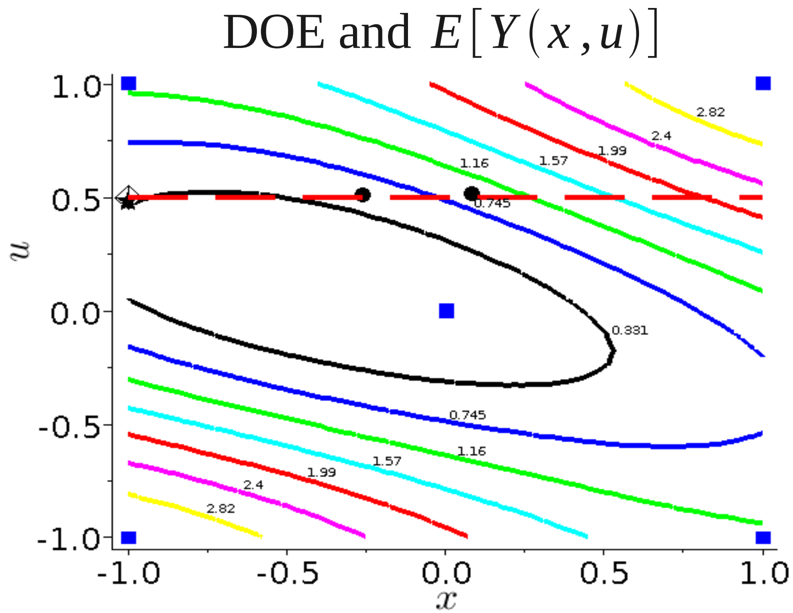


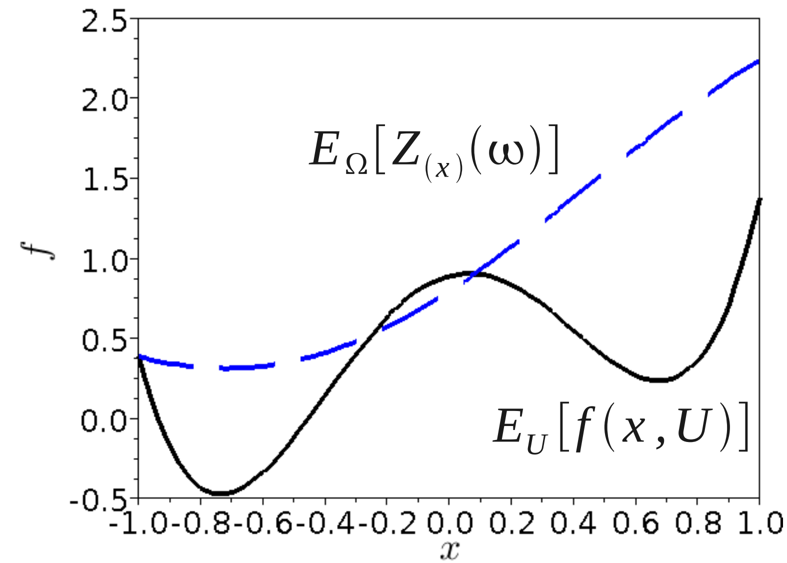
◇ – (x^{next}, μ)

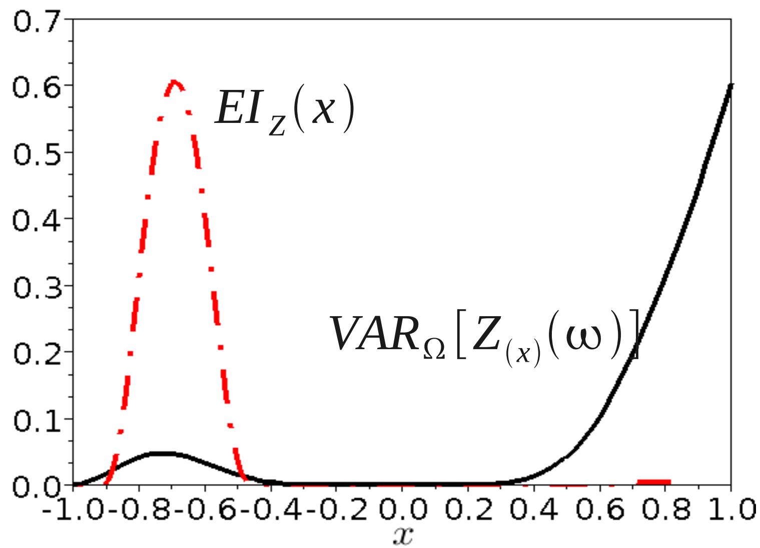
★ – (x^{t+1}, u^{t+1})

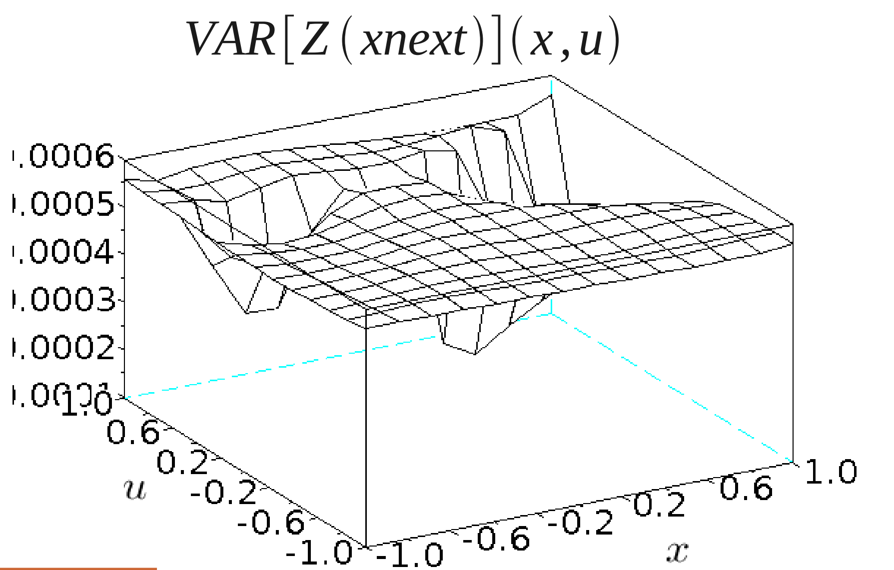
Kriging based optimization with uncertainties, U controlled

3rd iteration



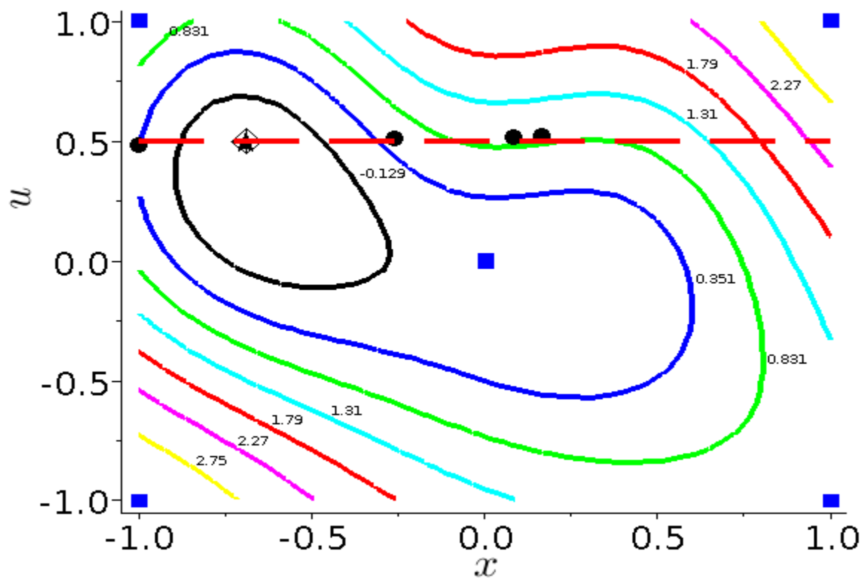


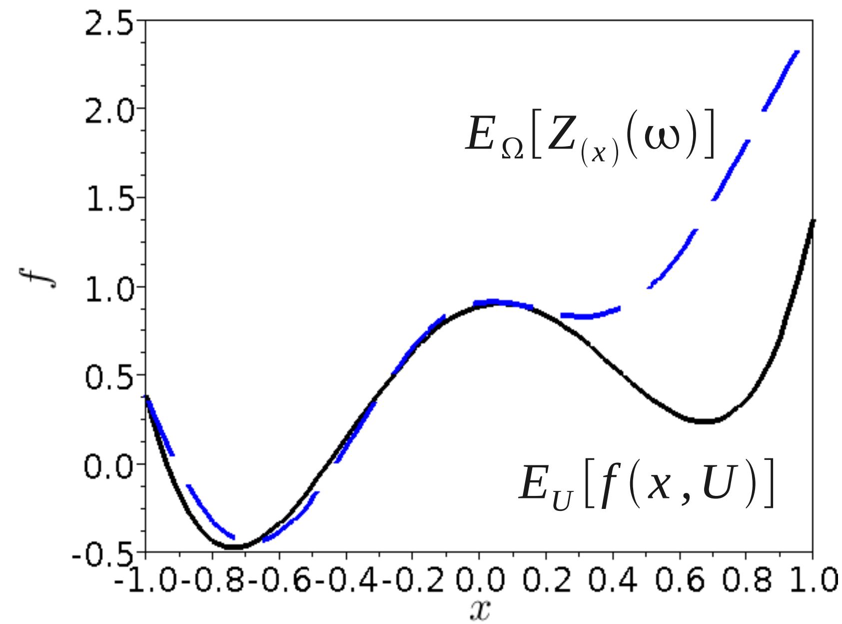


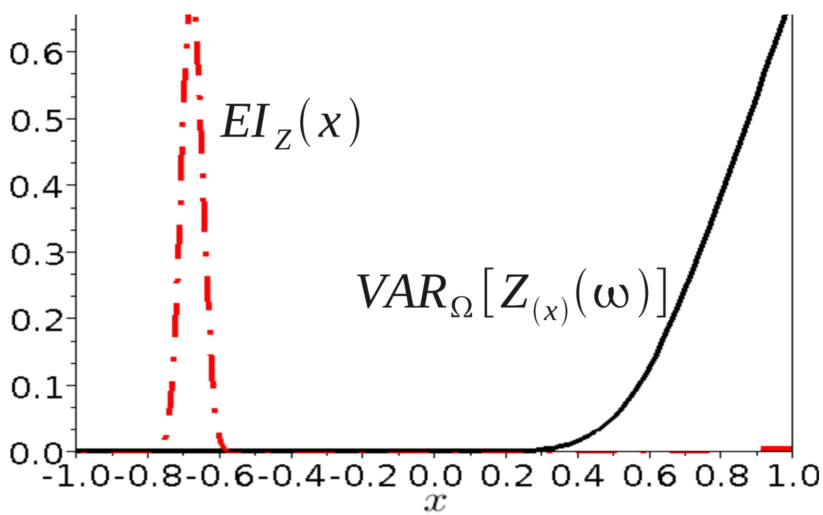


Kriging based optimization with uncertainties, U controlled
5th iteration

DOE and $E[Y(x,u)]$







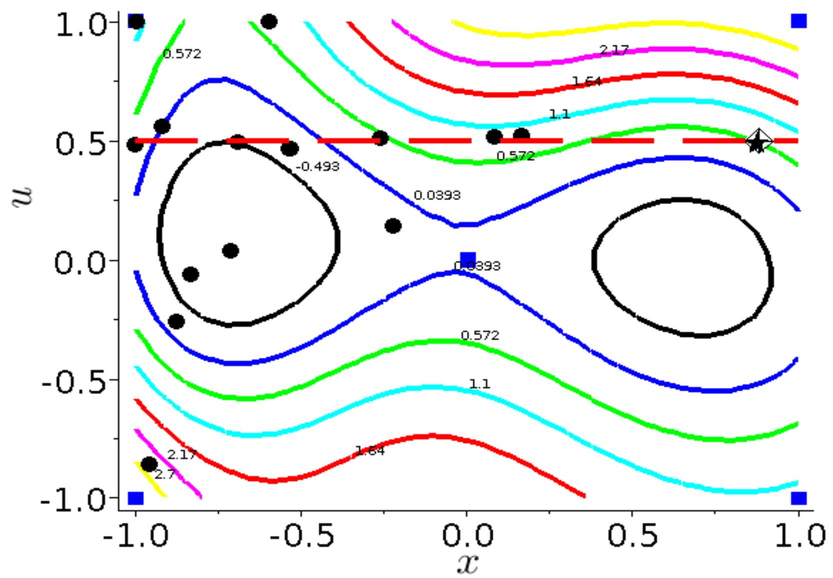
◇ – (x^{next}, μ)

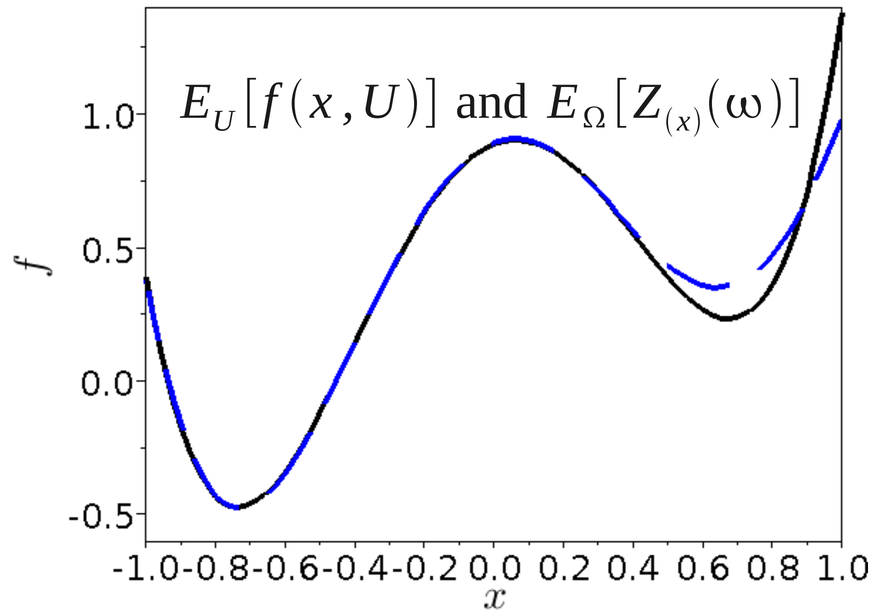
★ – (x^{t+1}, u^{t+1})

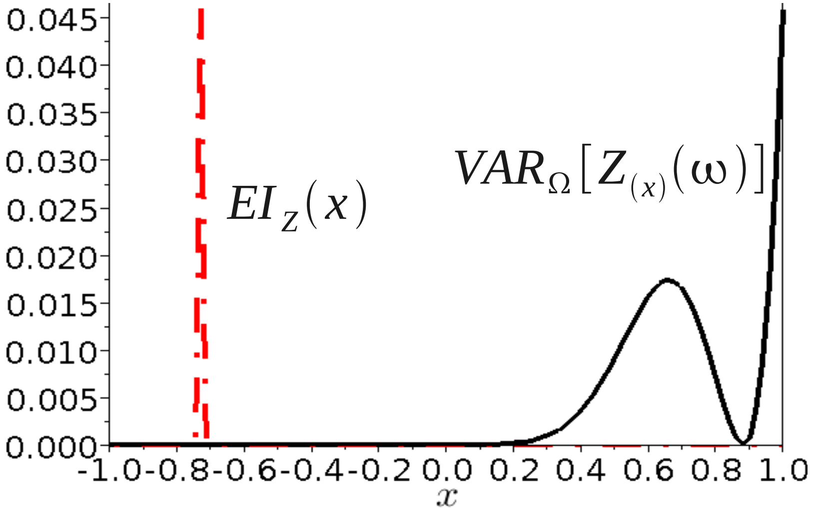
Kriging based optimization with uncertainties, U controlled

17th iteration

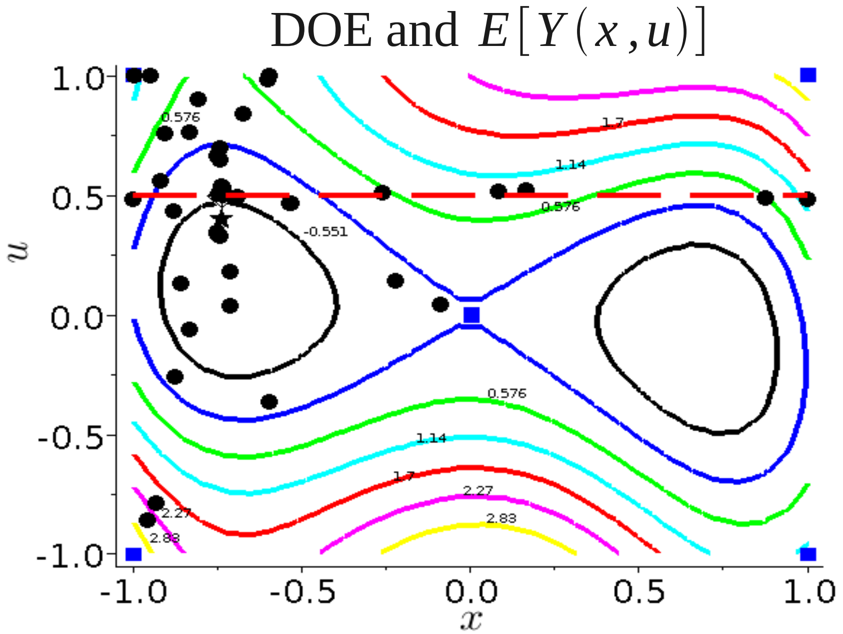
DOE and $E[Y(x,u)]$

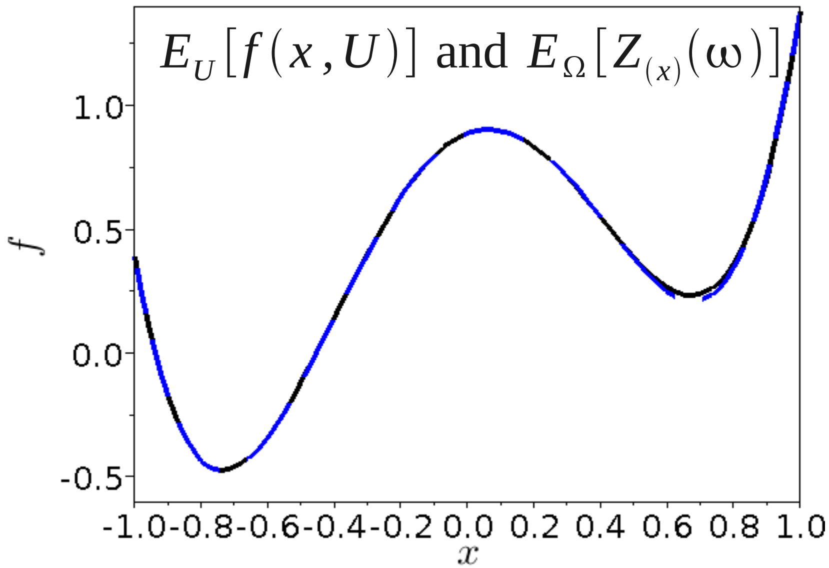


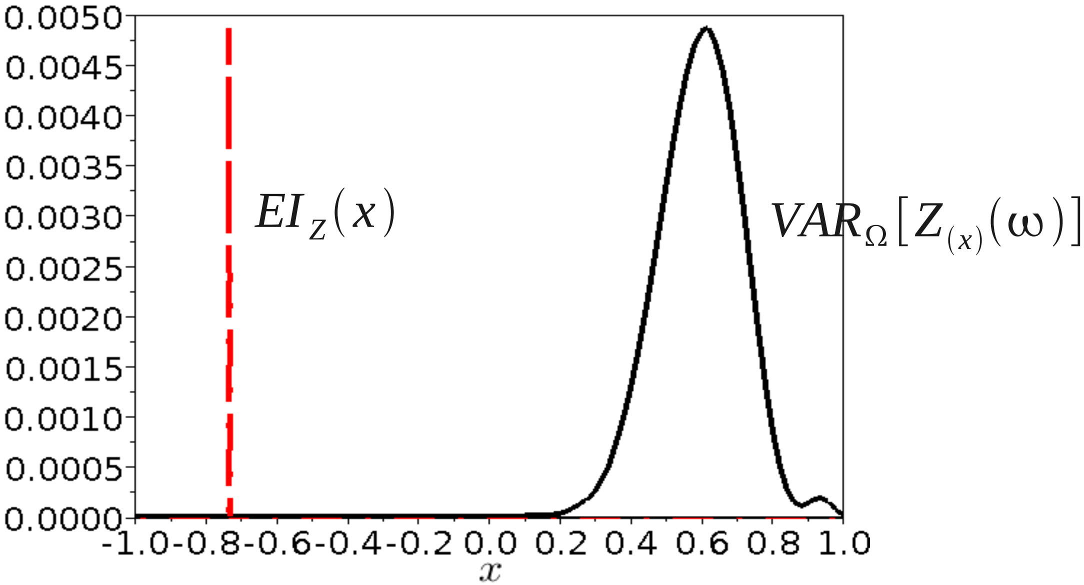




Kriging based optimization with uncertainties, U controlled
50th iteration







Comparison tests

Compare « simultaneous opt and sampling » method to

1. A direct MC based approach :
   EGO based on MC simulations in *f* with fixed number of runs, *s*.
   Kriging with homogenous nugget to filter noise.

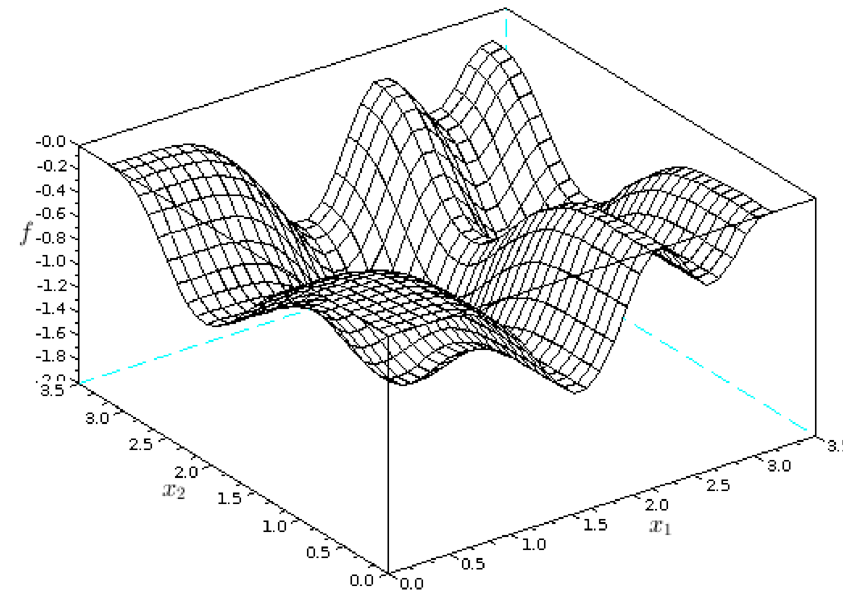
2. An MC-surrogate based approach :
   the MC-kriging algorithm.

Test functions

Test cases based on Michalewicz function

$$f(x) = -\sum_{i=1}^{n} \sin(x_i)\left[\sin\left(i x_i^2/\pi\right)\right]^2$$

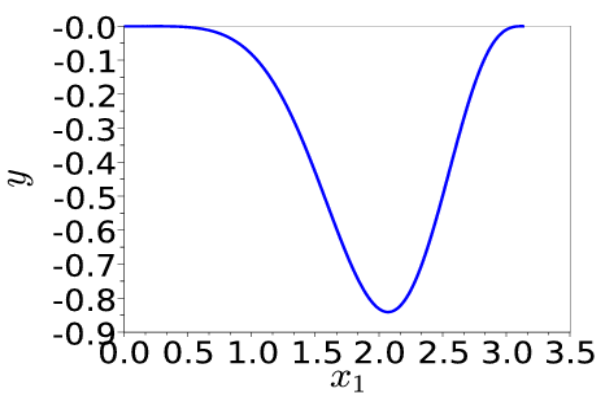
$$f(x,u) = f(x) + f(u)$$

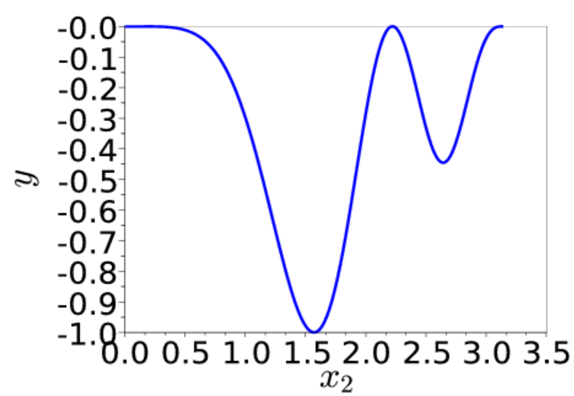


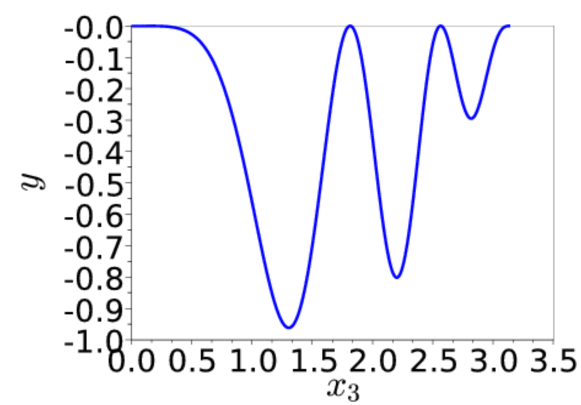
2D: $n_x = 1 \quad n_u = 1 \quad \mu = 1.5 \quad \sigma = 0.2$

4D: $n_x = 2 \quad n_u = 2 \quad \mu = [1.5,\ 2.1] \quad \sigma = [0.2,\ 0.2]$

6D: $n_x = 3 \quad n_u = 3 \quad \mu = [1.5,\ 2.1,\ 2] \quad \sigma = [0.2,\ 0.2,\ 0.3]$

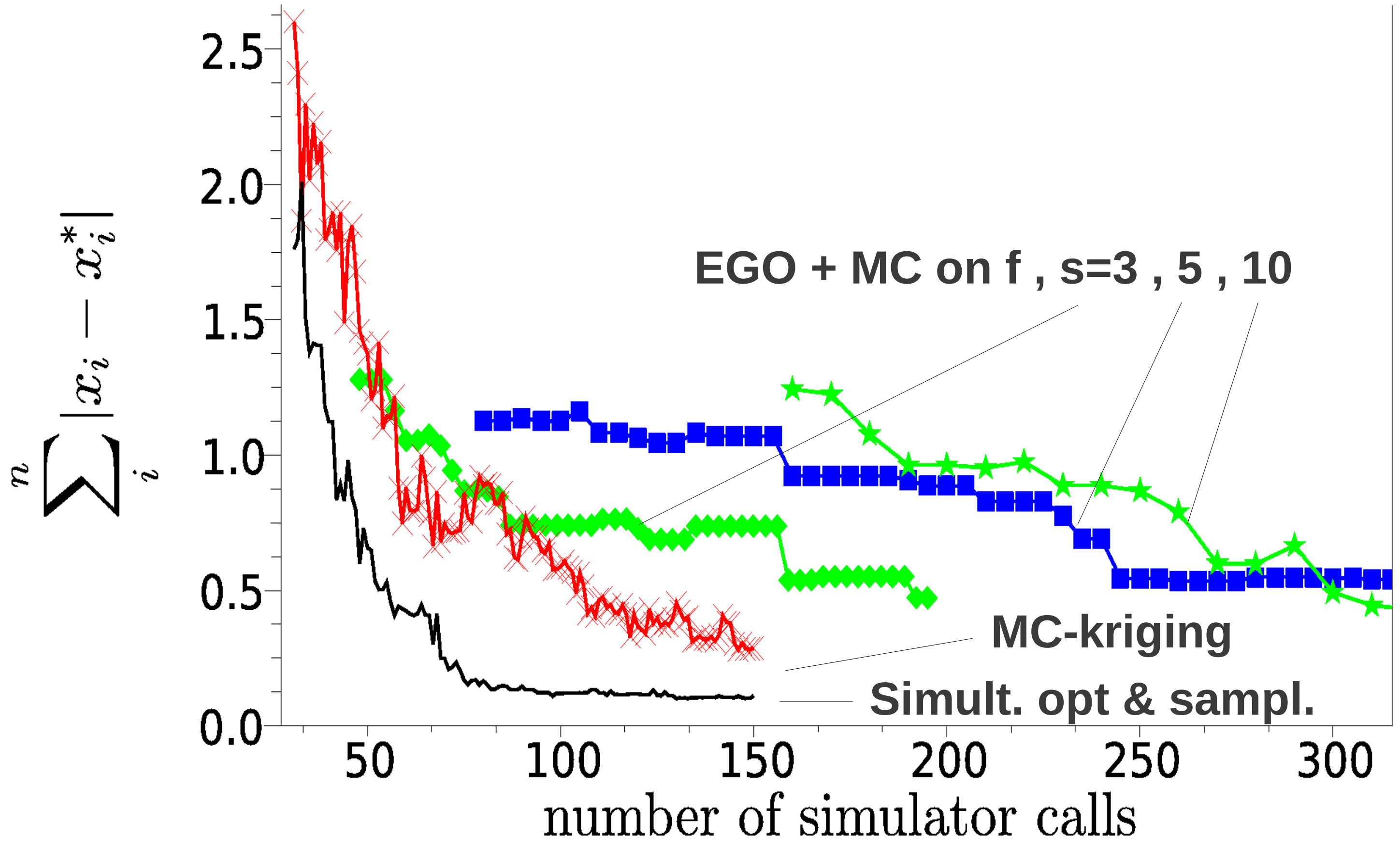






Test results

6D Michalewicz test case, n_x =3 , n_U =3 .
Initial DOE: RLHS , m=(n_x+n_U)*5 = (3+3)*5 = 30;
10 runs for every method.



Concluding remarks (1)

Today's story was :

• Optimization → difficult in the presence of noise → formulation of optimization in the presence of uncertainties → noisy optimization → methods without U control → methods with U control.

• There was an increasing degree of sophistication, and a decreasing degree of generality.

Each method has its application domain :

• Stochastic optimizers robust to noise cannot be directly applied to an expensive (simulation based) objective function. An intermediate surrogate is needed.

• Vice versa, kriging based method involve large side calculations : they are interesting only for expensive *f*'s.

• The applicability of kriging based methods to high dimensional spaces is a topic for further research.

Concluding remarks (2)

The following methods within OMD projects were not discussed :

Method of moments

• R. Duvigneau, M. Martinelli, P. Chandrashekarappa, *Uncertainty Quantification for Robust Design*, Multidisciplinary Design Optimization in Computational Mechanics, Wiley, 2010.

FORM / SORM, optimal safety factor methods for reliability (constraints with uncertainties)

• G. Kharmanda, A. El-Hami, E. Souza de Cursi, *Reliability-Based Design Optimization*, Multidisciplinary Design Optimization in Computational Mechanics, Wiley, 2010.

• D. Villanueva, R. Le Riche, G. Picard, G., R.T. Haftka and B. Sankar, *Decomposition of System Level Reliability-Based Design Optimization to Reduce the Number of Simulations*, ASME 2011 conf.(IDETC).

(and of course a lot of the large litterature on the subject could not be covered).