Simultaneous Kriging-based sampling for optimization and uncertainty propagation

23.11. NKO workshop, Bern

Janis Janusevskis, Rodolphe Le Riche

Ecole des Mines de Saint-Etienne (EMSE)

Outline

1. Introduction
2. Projected Gaussian process
3. Sampling criterion based on projected process
4. Demonstration on analytical test case
5. Comparisons with MC approach on analytical test case
6. Application to 2D test case of conditioner tube

EGO for optimization of deterministic simulator

EGO – Efficient Global Optimization Donald R.Jones, et al.(1998).

Based on Kriging's metamodel (GP regression) of deterministic simulator.

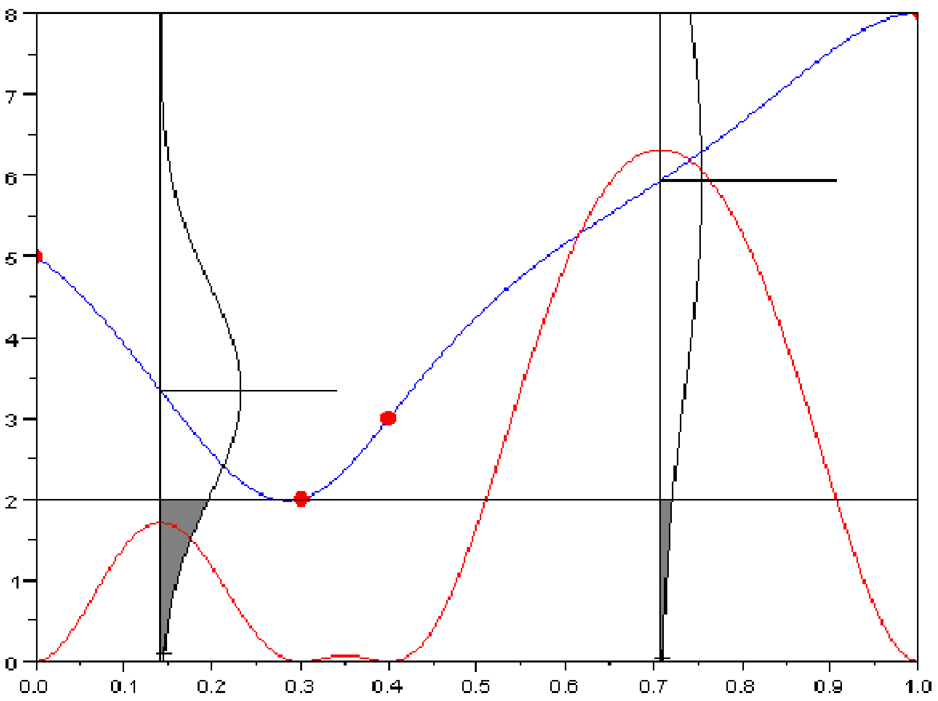
Possible to estimate predicted mean and variance at any point **x**.

Chooses next point by maximizing Expected Improvement (mixes local and global search).

$$\min_x f(x)$$

$$EI(x) = E\left[\max\left(f_{best}^t - Y^t(x), 0\right)\right]$$

$$x^{t+1} = \arg\max EI(x)$$



Optimization under uncertainty

Search for designs whose quality degrades in a controlled way due to uncertainties in parameters:

- Manufacturing errors.
- Material properties.
- Dynamic environment or noisy measurements.
- Model errors.

Stochastic response based on deterministic simulator: (x,u) parameters

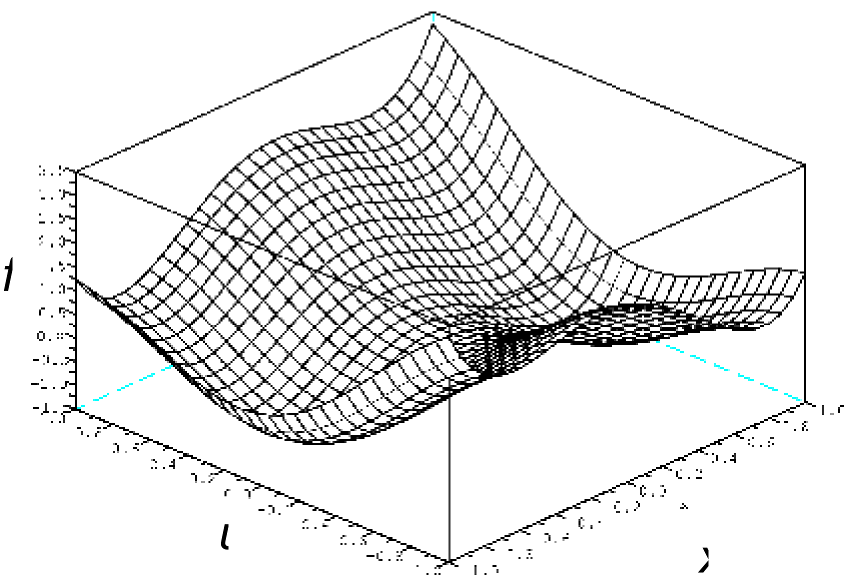
$$f(x,u) \longrightarrow f(x,U)$$

probability density of $U \sim f_{N(\mu,\Sigma)}$

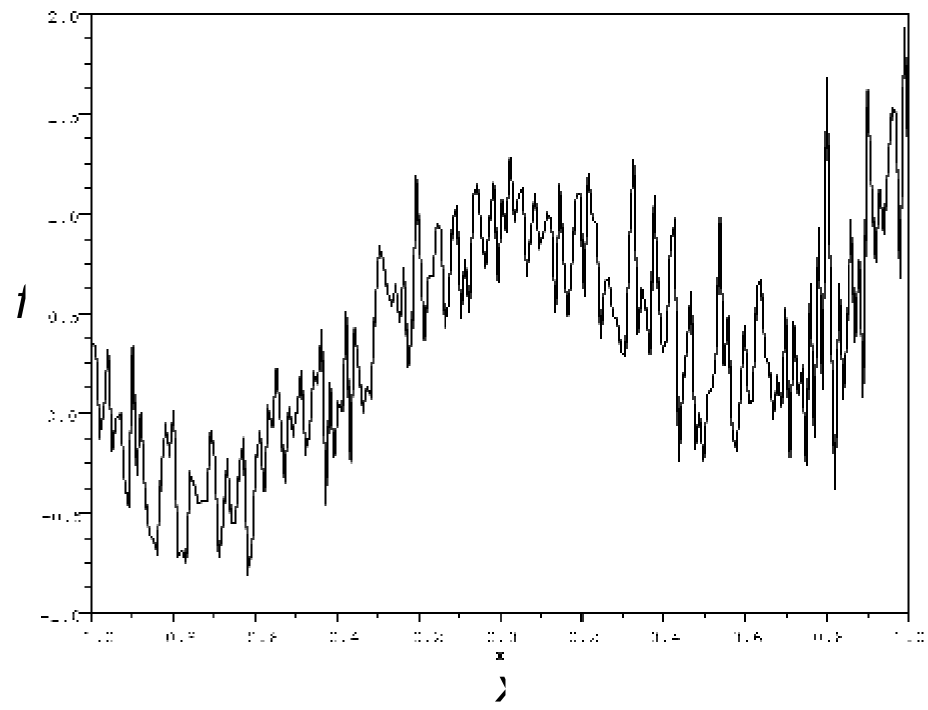
x – deterministic input variables / parameters

U – random input variables / parameters

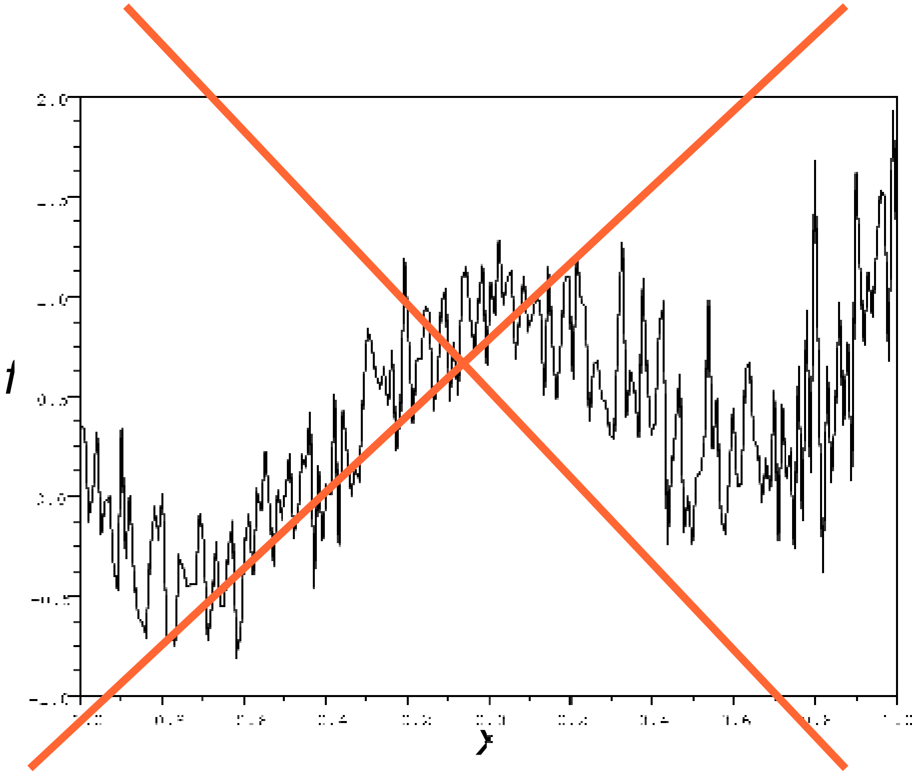
simulator (FE,CFD, etc.)



instance



We do not optimize a noisy function



part of the traditional answer to noisy optimization problems is to reformulate as

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

How to calculate $\mathbb{E}_U[f(x,U)]$?

Traditional Monte Carlo approach :

$$E_U[f(x,U)] \approx \frac{1}{M}\left(\sum_{i=1}^{M} f(x,u^i)\right)$$

(but M calls to the simulator, and still noisy)

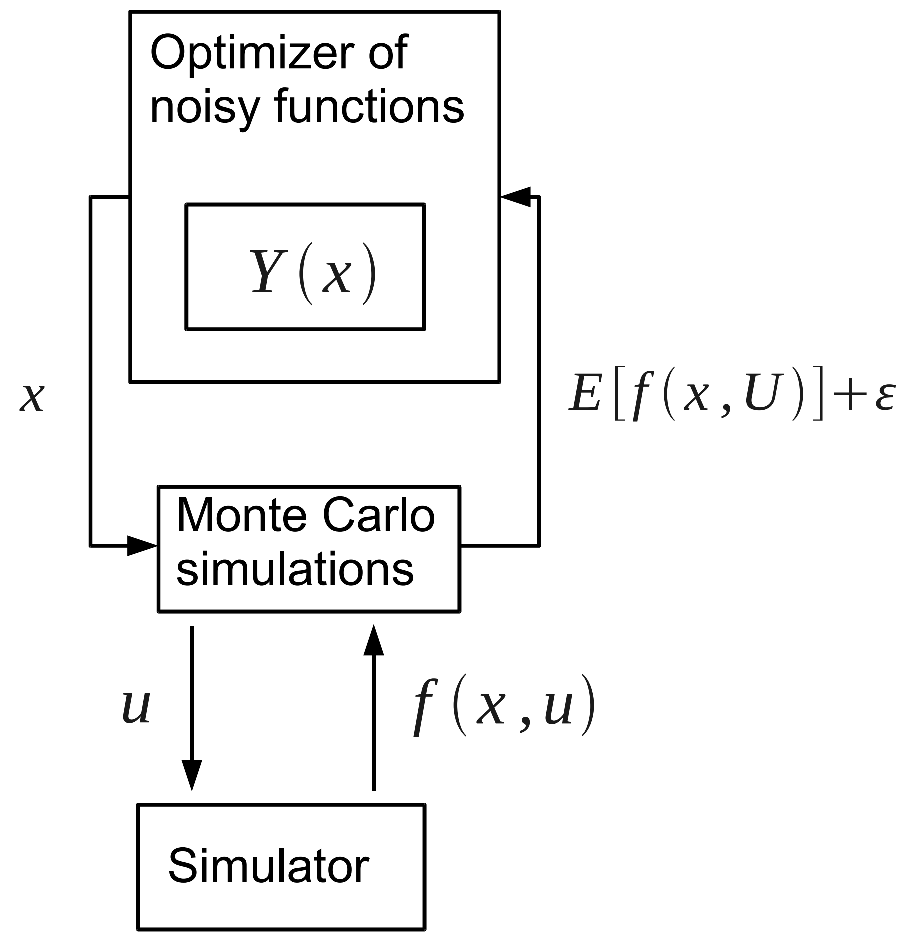
Our approach : use kriging

we will see how later ...

How to calculate $\mathbb{E}_U[f(x,U)]$ and optimize it on x's ?

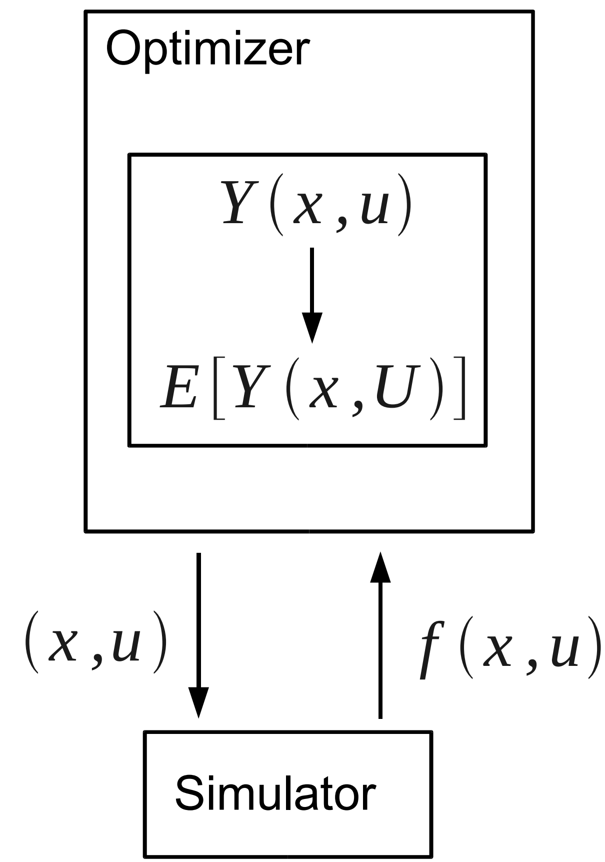
Y - approximation (kriging) model of the objective

Direct approach



Multiplicative cost of two loops

Proposed approach



Only one loop

Simultaneous Kriging-based sampling for optimization and uncertainty propagation

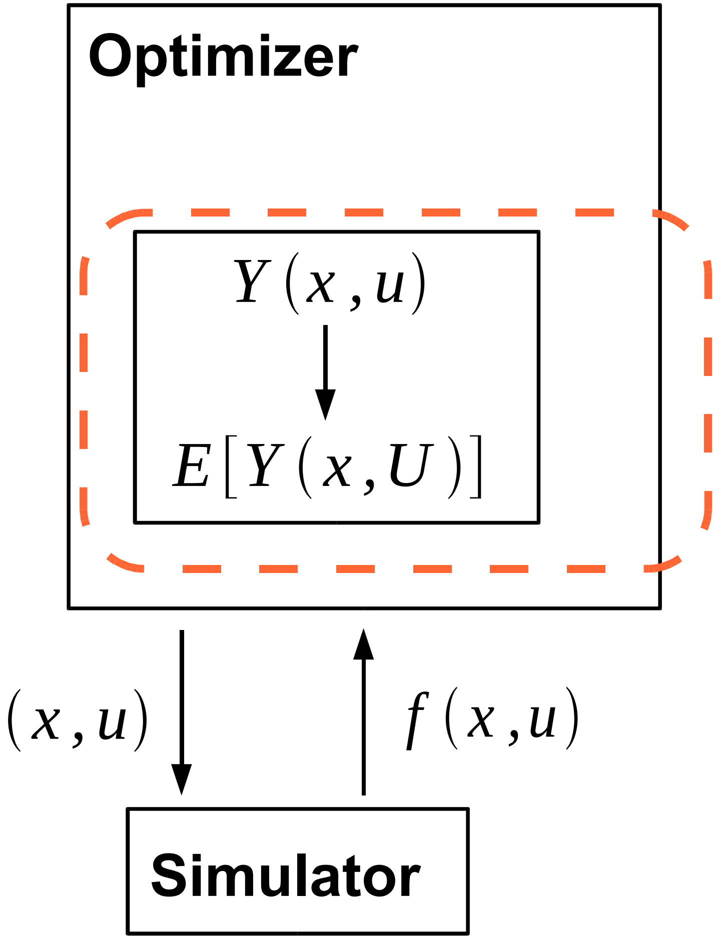
So, we propose :

1. an analytical way to estimate the mean objective function.

2. a strategy to choose x and u together to minimize this mean.

Let's go a little further into the details ...

Simultaneous Kriging-based sampling for optimization and uncertainty propagation

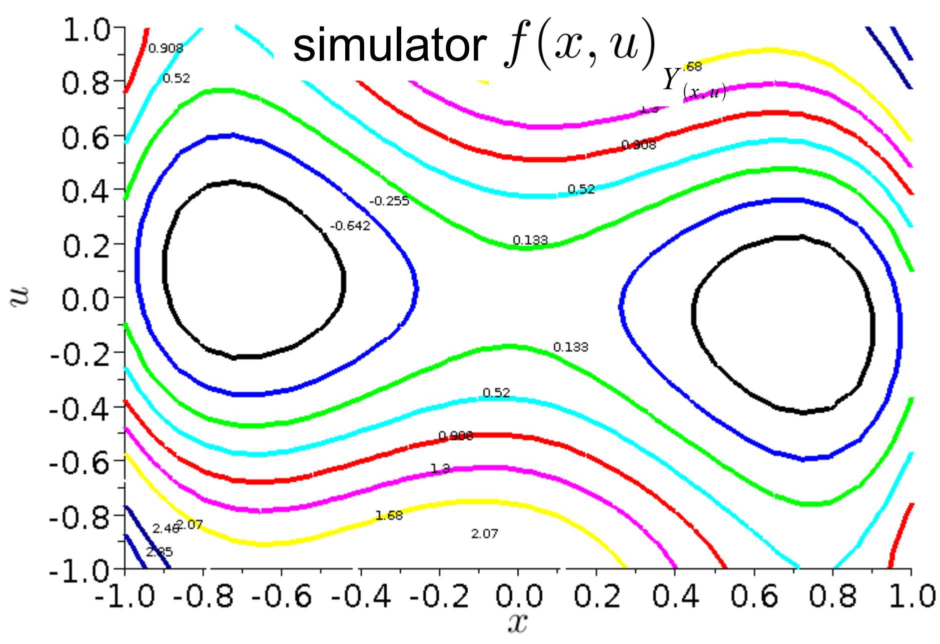


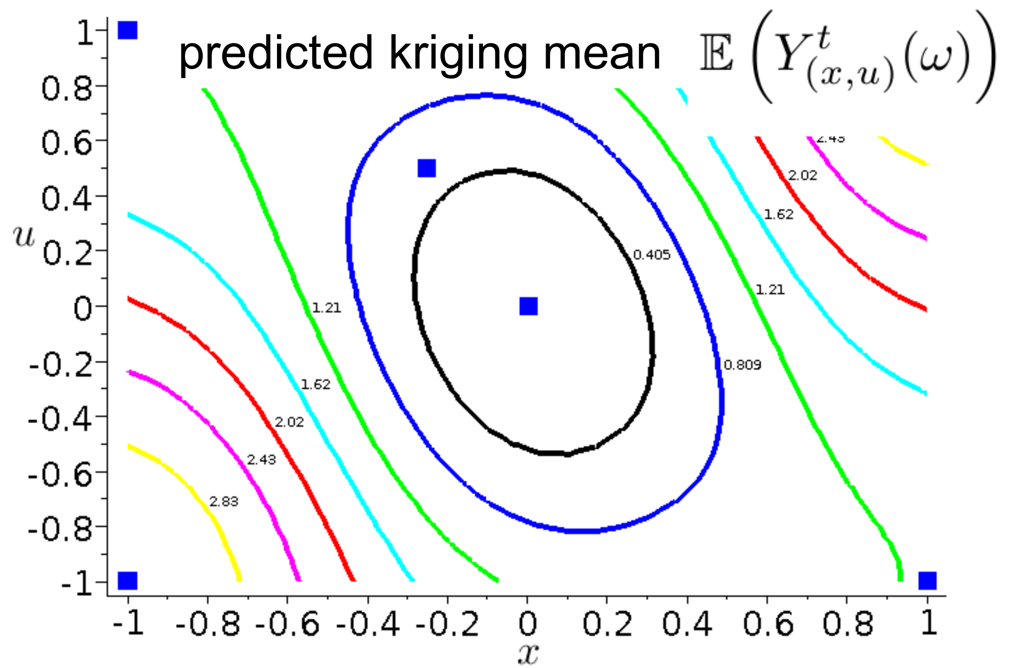
1. Building internal representation of the objective
(mean performance) by «projected» kriging.

Optimization using projected process (1/3)

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

$Y(x,u)$: **kriging approximation of deterministic** $f(x,u)$



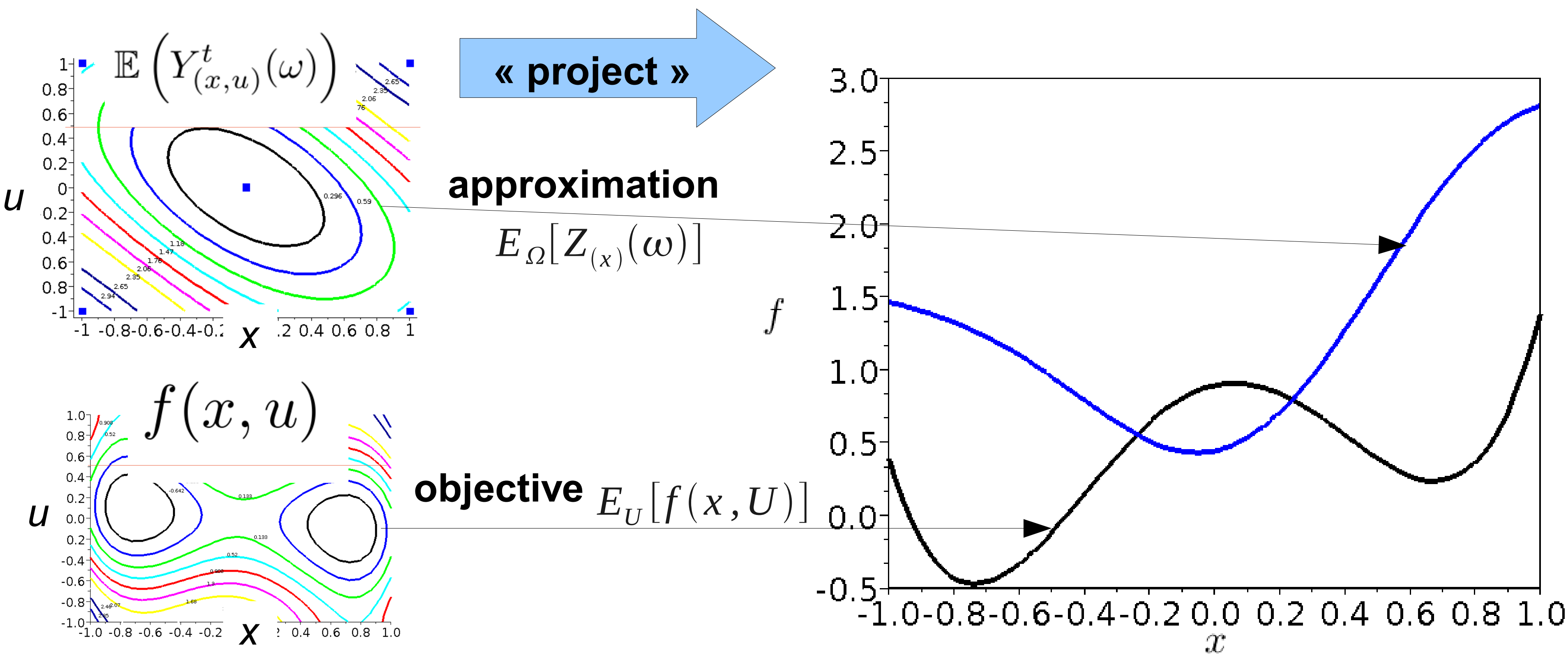


Optimization using projected process (2/3)

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

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

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



Projected process

Create new **Gaussian** process:

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

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

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

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

Conditional mean and covariance

$$\mathbb{E}\left(Z^t_{(\mathbf{X}_*)}(\omega)\right) = \int_{\mathbb{R}^m} m_Y(\mathbf{X}_*, u)d\mu(u)$$

$$+ \int_{\mathbb{R}^m} cov_Y(\mathbf{X}^*, u; \mathbf{X}, \mathbf{U}) cov_Y(\mathbf{X}, \mathbf{U}; \mathbf{X}, \mathbf{U})^{-1}(Y - m_Y(\mathbf{X}, \mathbf{U}))d\mu(u)$$

$$\mathbb{COV}\left(Z^t_{(\mathbf{X}_*)}(\omega)\right) = \int_{\mathbb{R}^m}\int_{\mathbb{R}^m} cov_Y(\mathbf{X}_*, u, \mathbf{X}_*, u')d\mu(u)d\mu(u')$$

$$+ \int_{\mathbb{R}^m} cov_Y(\mathbf{X}^*, u; \mathbf{X}, \mathbf{U})d\mu(u)\ cov_Y(\mathbf{X}, \mathbf{U}; \mathbf{X}, \mathbf{U})^{-1} \int_{\mathbb{R}^m} cov_Y(\mathbf{X}, \mathbf{U}, \mathbf{X}^*, u)d\mu(u)$$

For Gaussian covariance and noise it is possible to obtain analytical solutions.

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

Solutions for Gaussian kernel and Normal distribution of *U*

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

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

$$cov_Y(x,u;x',u') = \prod_{i=1}^{n} cov_Y(x_i;x_i')\prod_{i=1}^{m} cov_Y(u_i;u_i')$$

$$f_{N(\mu,\Sigma)}$$

$$cov_Y(u_i;u_i') = exp\left(-\frac{1}{2}\left(\frac{u_i-u_i'}{\theta_i}\right)^2\right)$$

$$D = diag(\theta_i^2)$$

$$I(x,u,x') = \prod_{i=1}^{n} cov_Y(x_i;x_i')\ \frac{|D|^{1/2}}{|\Sigma+D|^{1/2}} exp\left[-\frac{1}{2}(u-\mu)^T(\Sigma+D)^{-1}(u-\mu)\right]$$

$$I(x,x') = \prod_{i=1}^{n} cov_Y(x_i;x_i')\ \frac{|D|^{1/2}}{|2\Sigma+D|^{1/2}}$$

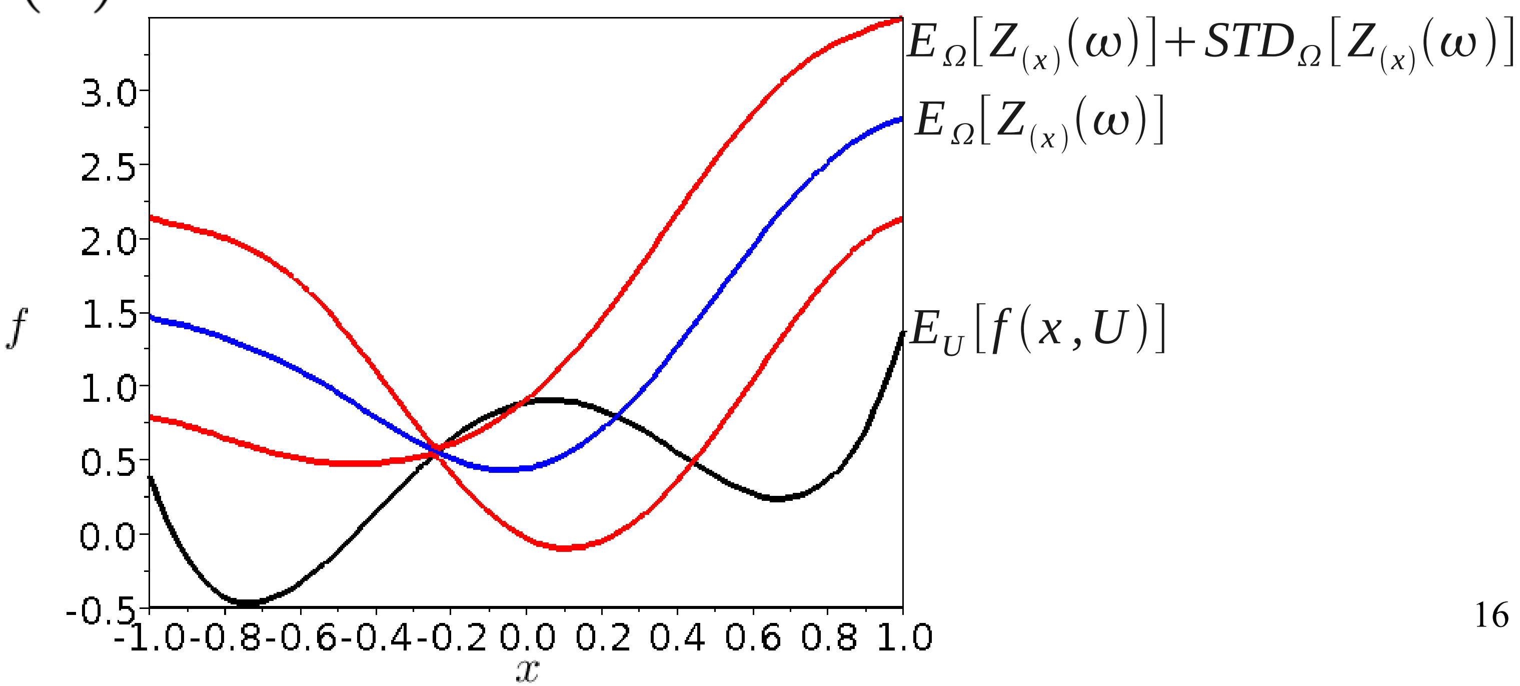
Optimization using projected process (3/3)

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

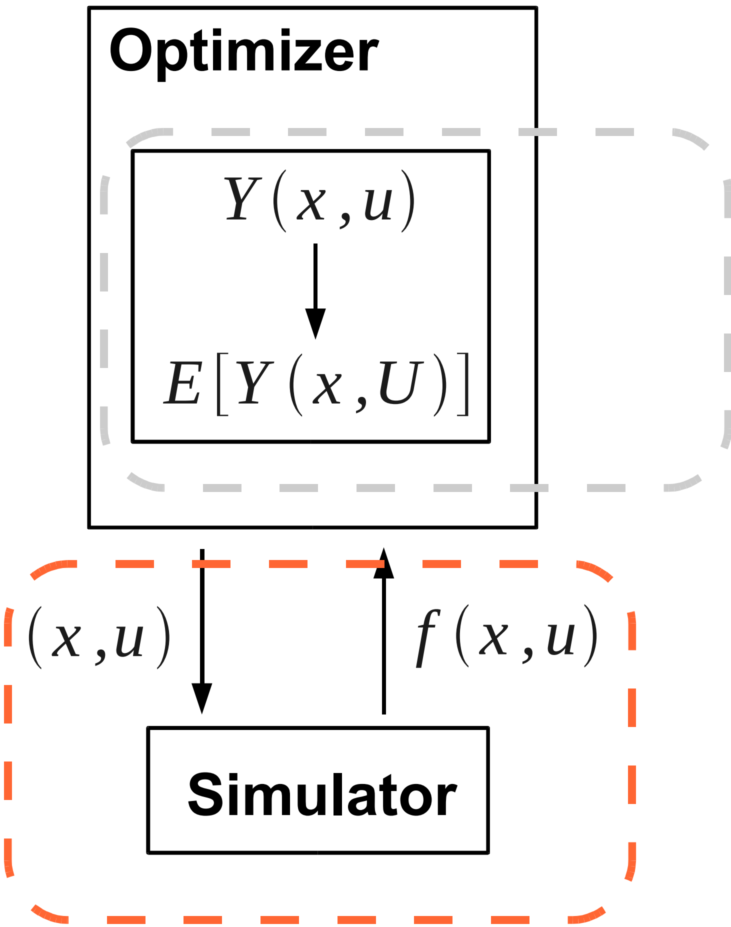
$$Y^t_{(x,u)}(\omega) = [Y_($$: **kriging approximation to deterministic** $f(x,u)$

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

$$\min_x Z^t_{(x)}$$: **use projected process to sample new points for optimization of the objective**



Simultaneous Kriging-based sampling for optimization and uncertainty propagation

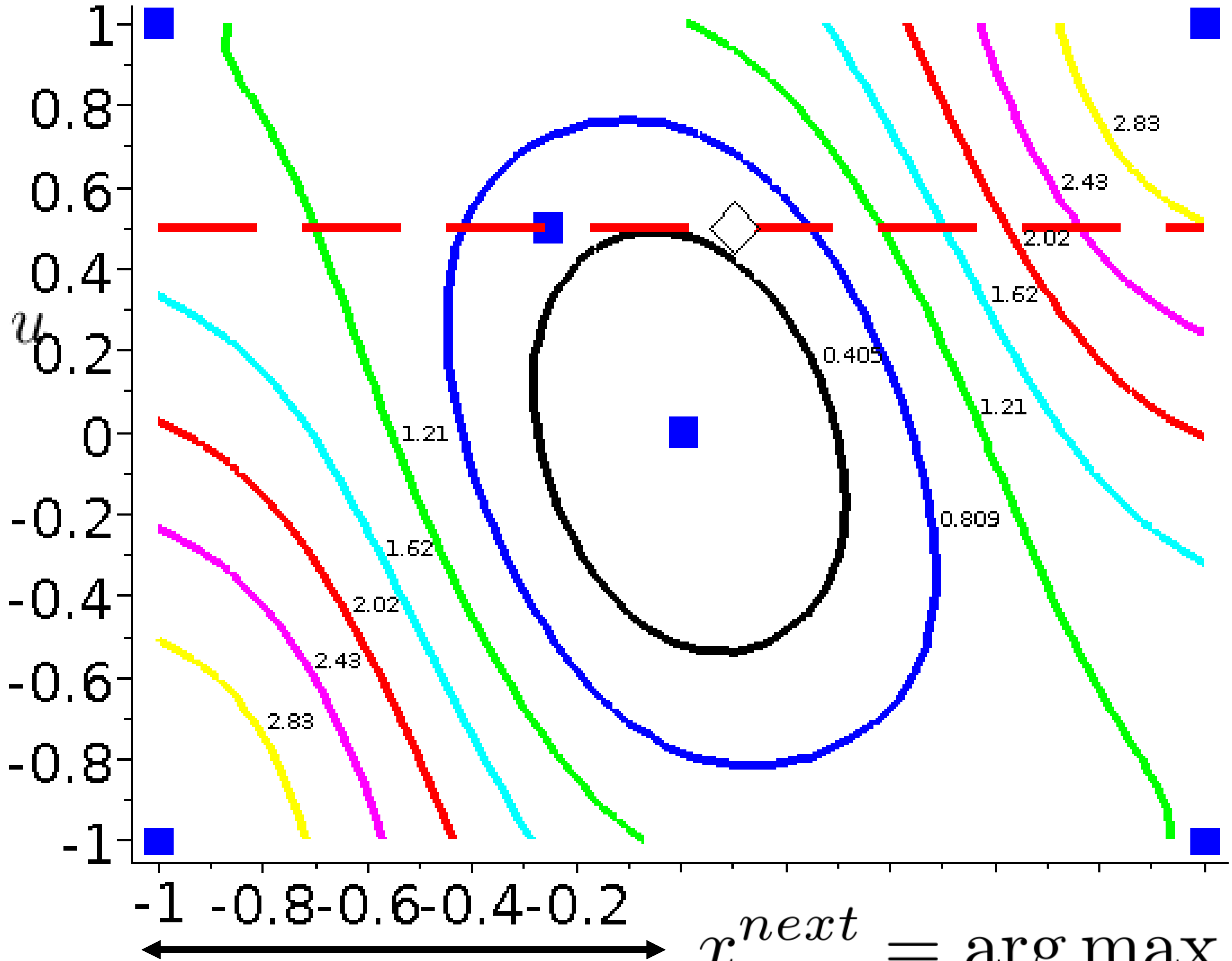


1. Building internal representation of the objective (mean performance) by «projected» kriging.

2. Simultaneous sampling criterion for x and u.

To run the simulator we need BOTH: x AND u

Sampling criterion in (*x*,*u*) space (1/2)

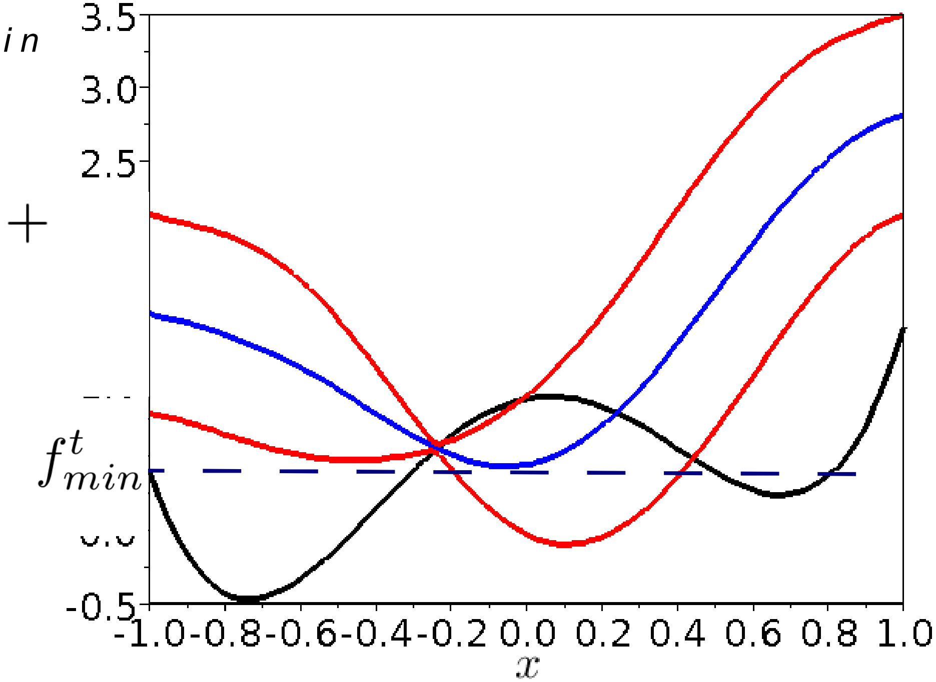


$$x^{next} = \arg\max_x EI_Z^t(x)$$

Sampling criterion for optimization

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

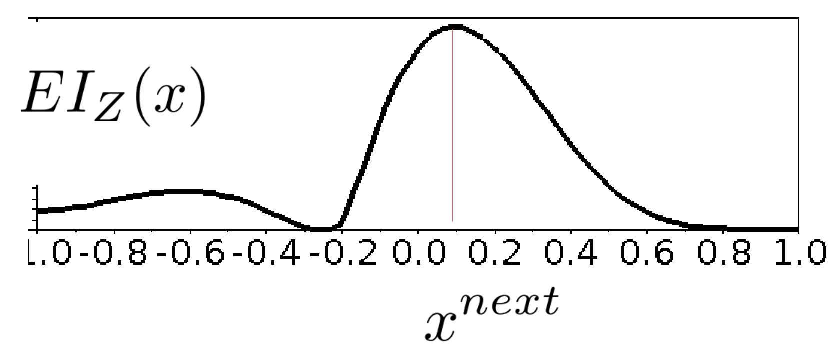
in the manner of EGO with a specific definition of f_{min}



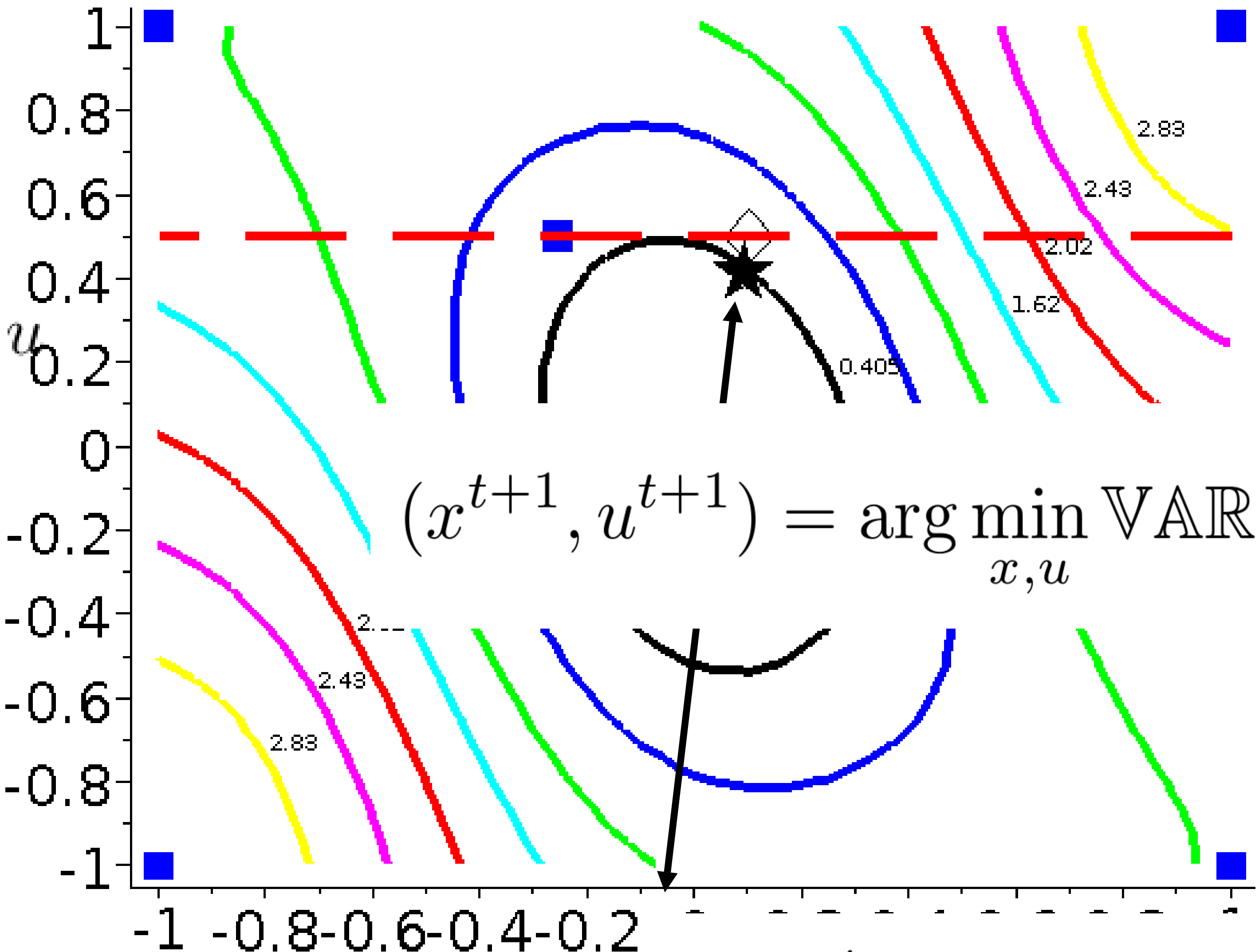
$$\max_x EI_Z(x) = \mathbb{E}\left[f^t_{min} - Z^t_{(x)}(\omega)\right]^+$$

$$f^t_{min} = \min_x \mathbb{E}\left[\, Z^t_{(x)}(\omega)\right]$$

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



Sampling criterion in (*x*,*u*) space (2/2)



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

$$x^{next} = \arg\max_{x} EI^t_Z(x)$$

Sampling criterion in (*x*,*u*) space : summary

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

   x^{next} **is an interesting point to minimize objective (maximizing EI of Z).**

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

   (x^{t+1}, u^{t+1}) **is the point that provides the most information about Z at x^{next}.**

!!! x^{t+1} and x^{next} can be different.

!!! Var Z^{t+1} can be calculated without knowing $f(x^{t+1},u^{t+1})$ (keeping kriging hyperparameters constant)

!!! it simultaneously defines a sampling criterion in the *u* space (as opposed to two loop MC based approaches).

Complete algorithm

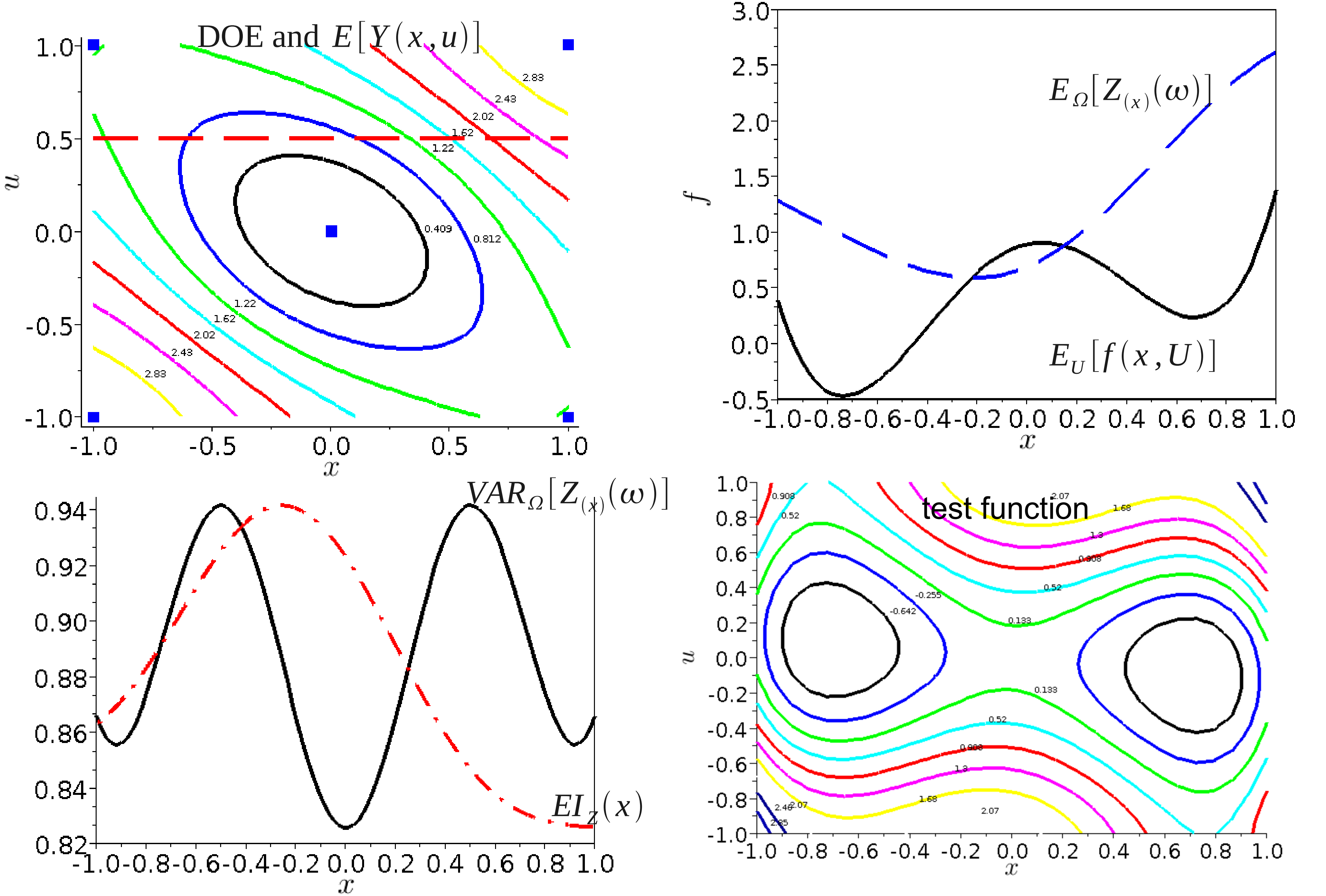
Create initial DOE in (x,u) space;

While stoping criterion is not met:

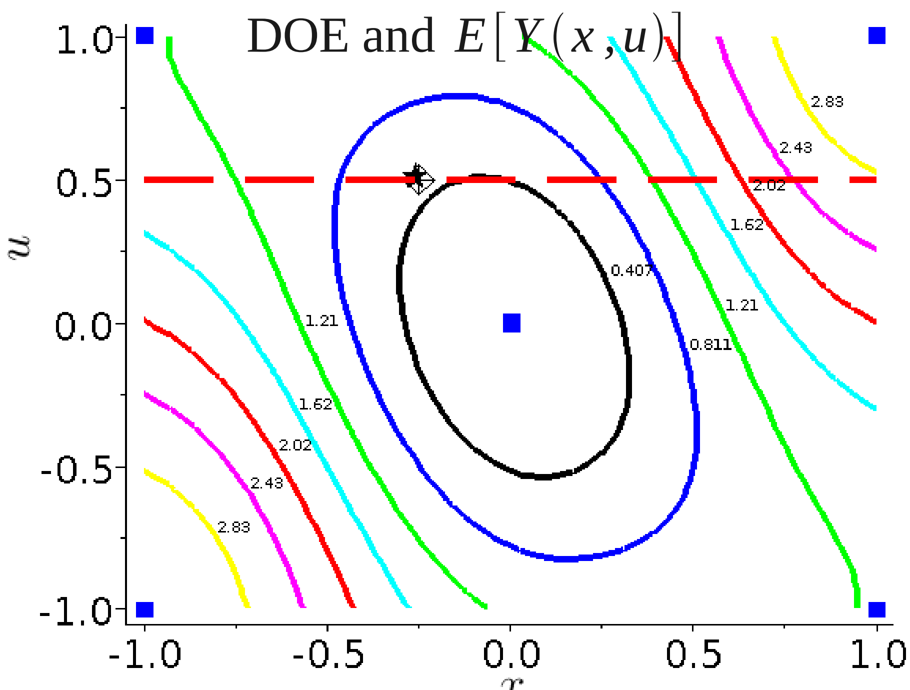
- **Create kriging approximation *Y* in the joint space** (x, u)
- **Using covariance information of *Y* to obtain approximation *Z* of the objective in the deterministic space** (x)
- **Use EI of Z to choose** (x^{next})
- **Minimize** $VAR(Z(x^{next}))$ **to obtain the next point** (x^{t+1}, u^{t+1})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

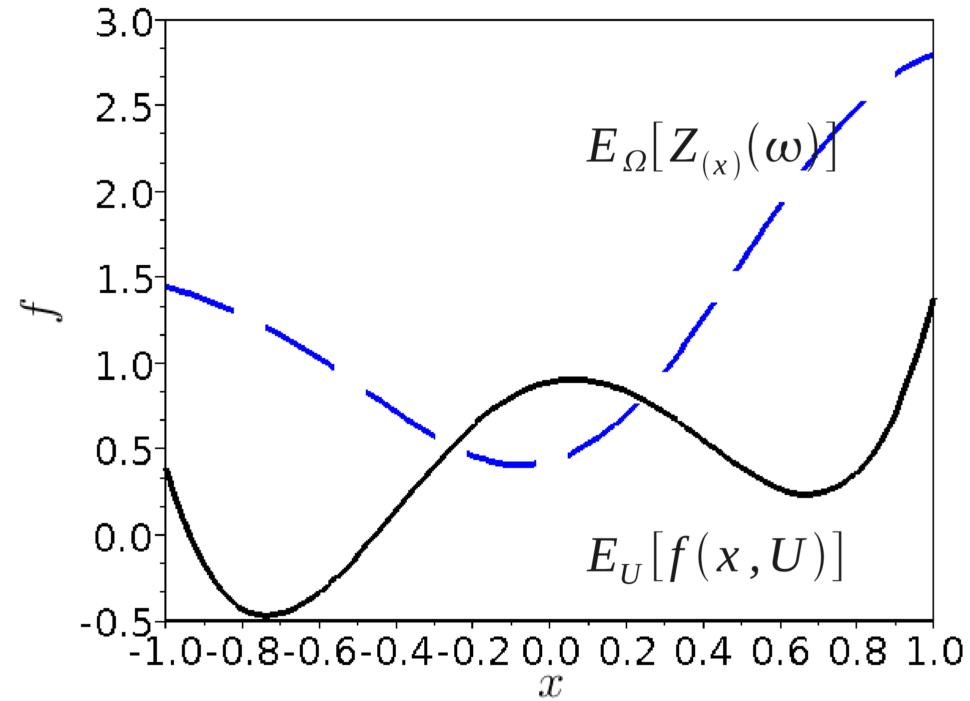
(4 sub-optimizations, solved with CMA-ES, implementation in Scilab)

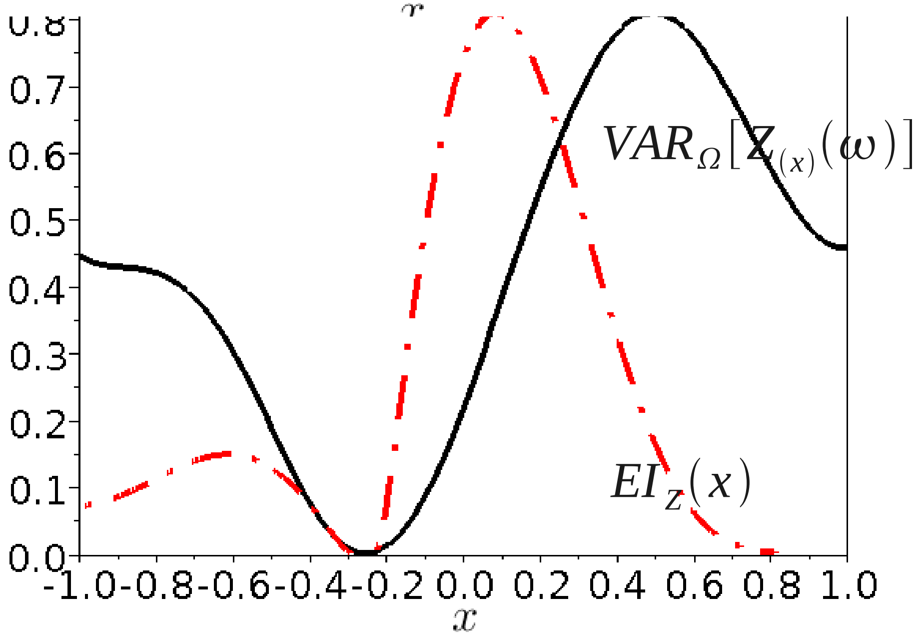
2D illustration of the method

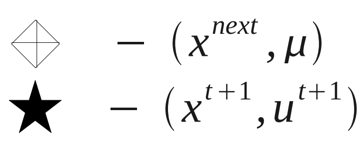


1st iteration

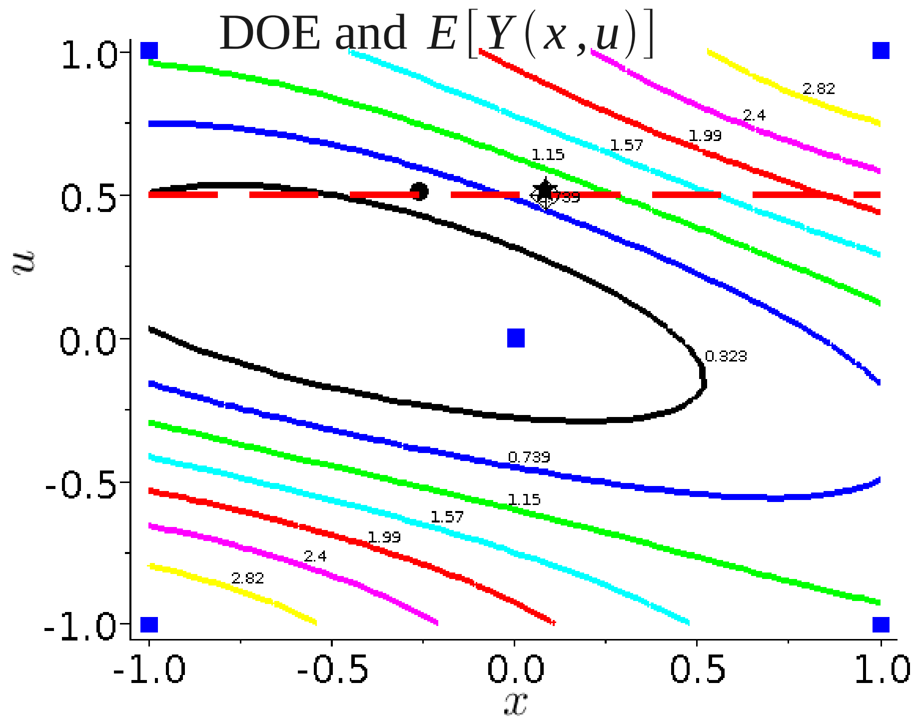


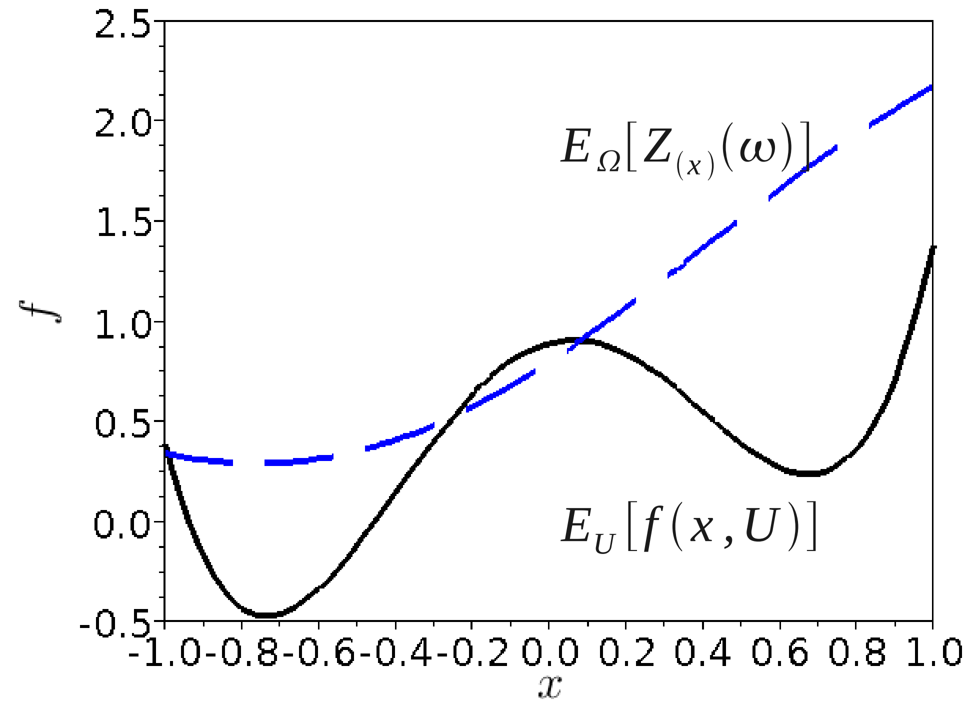


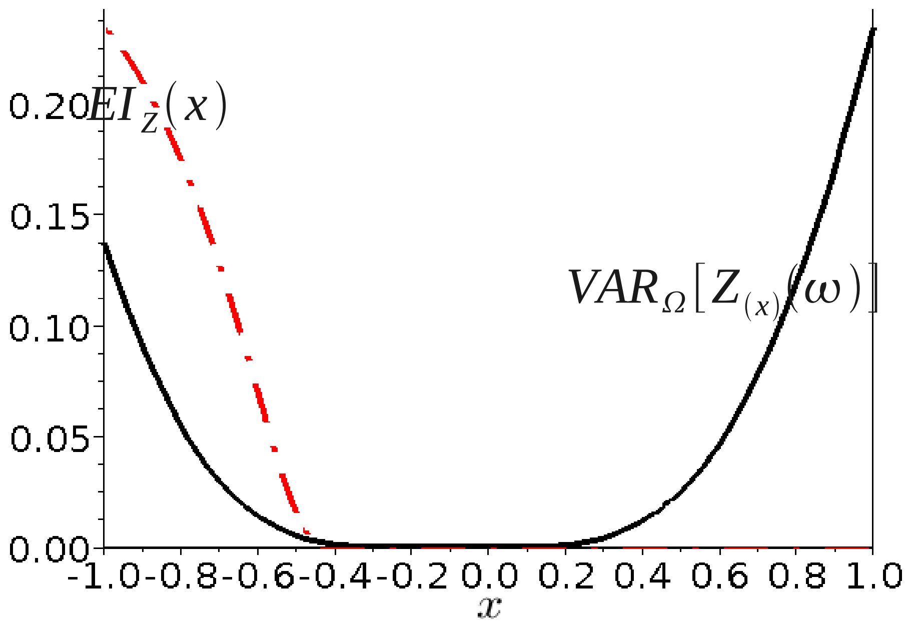


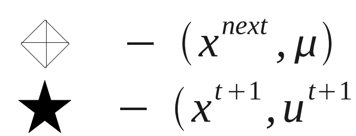


2nd Iteration

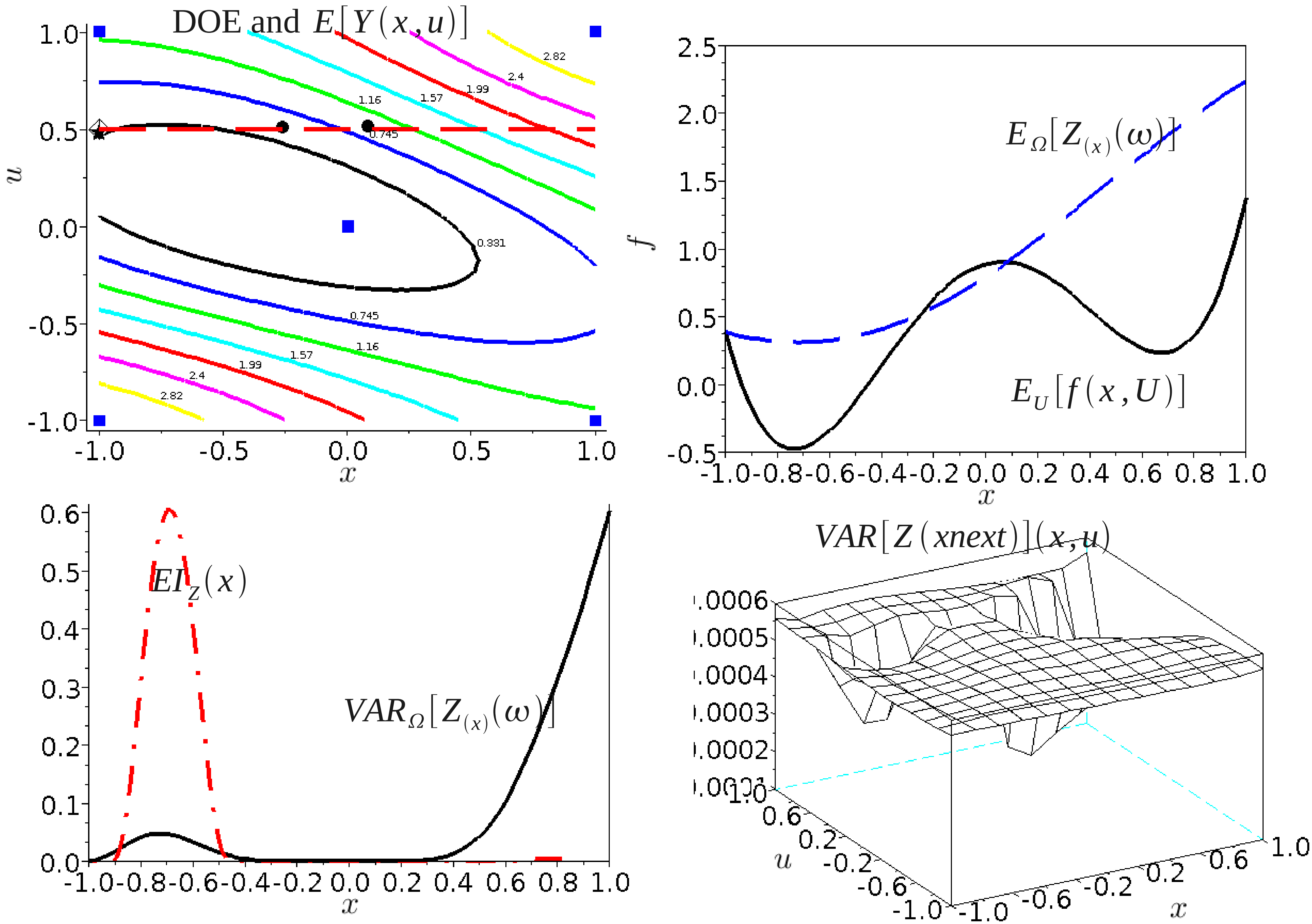




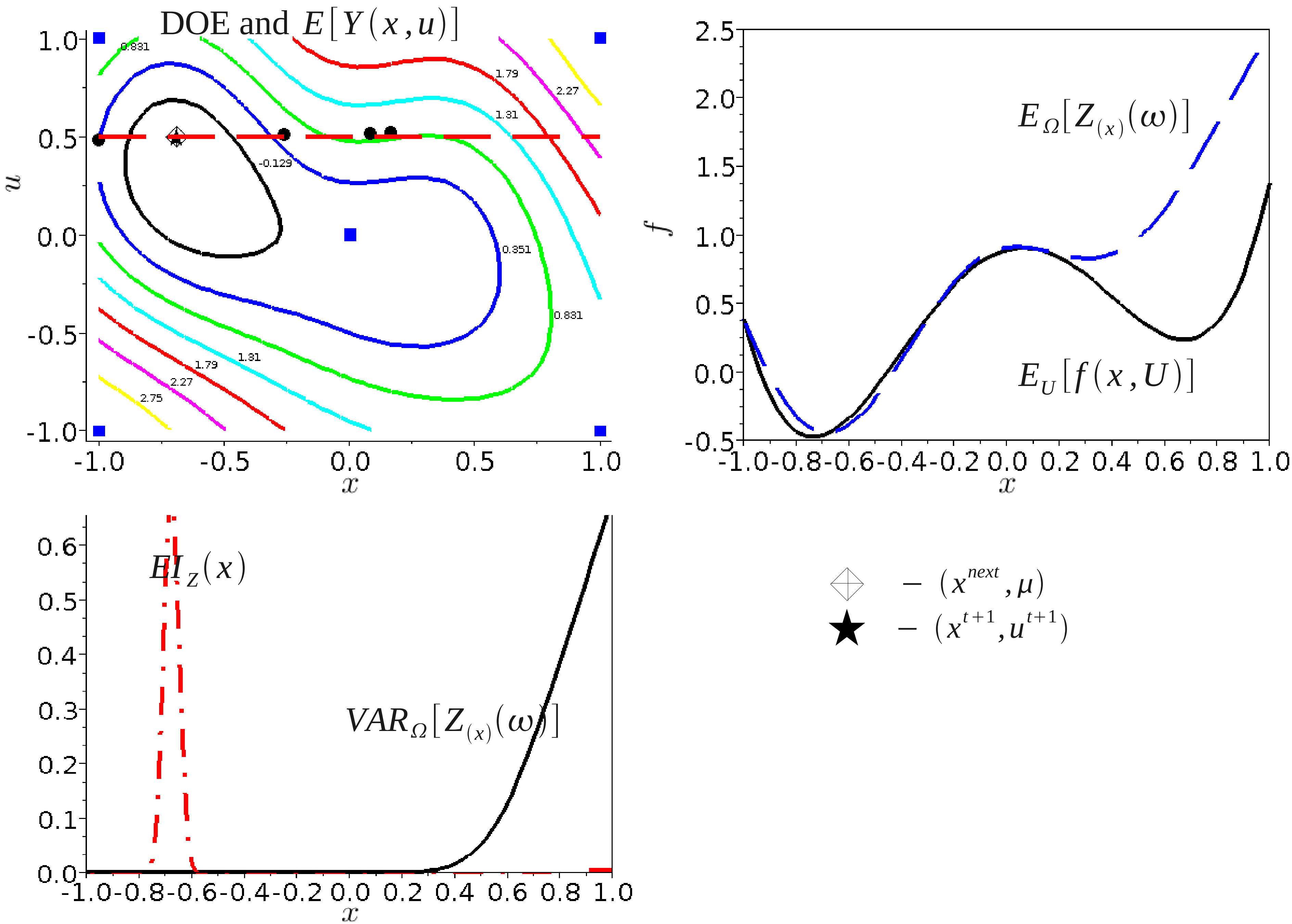




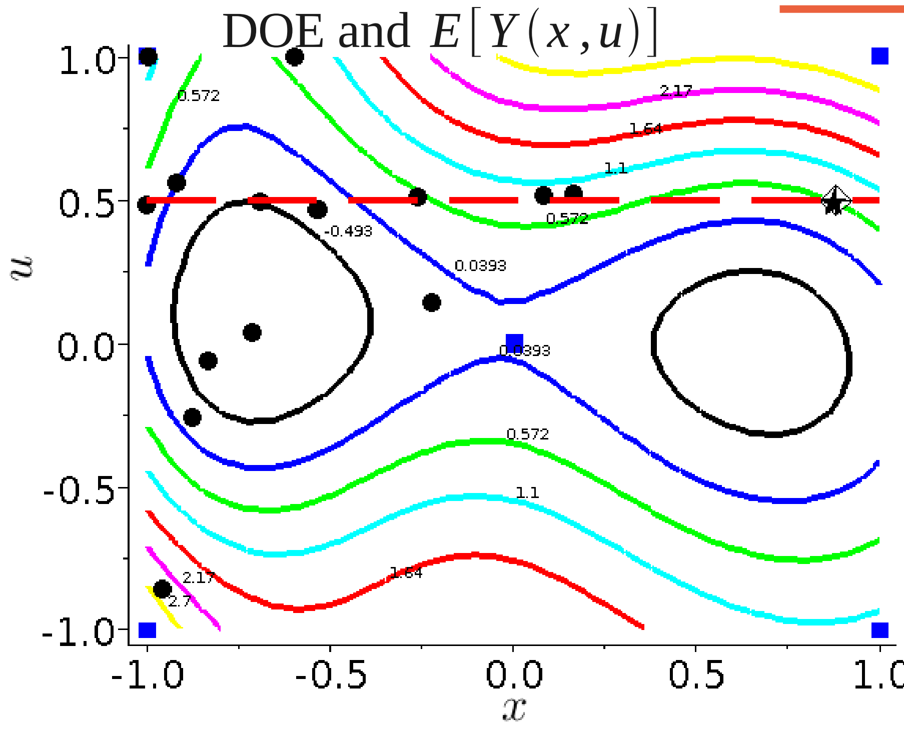
3rd Iteration

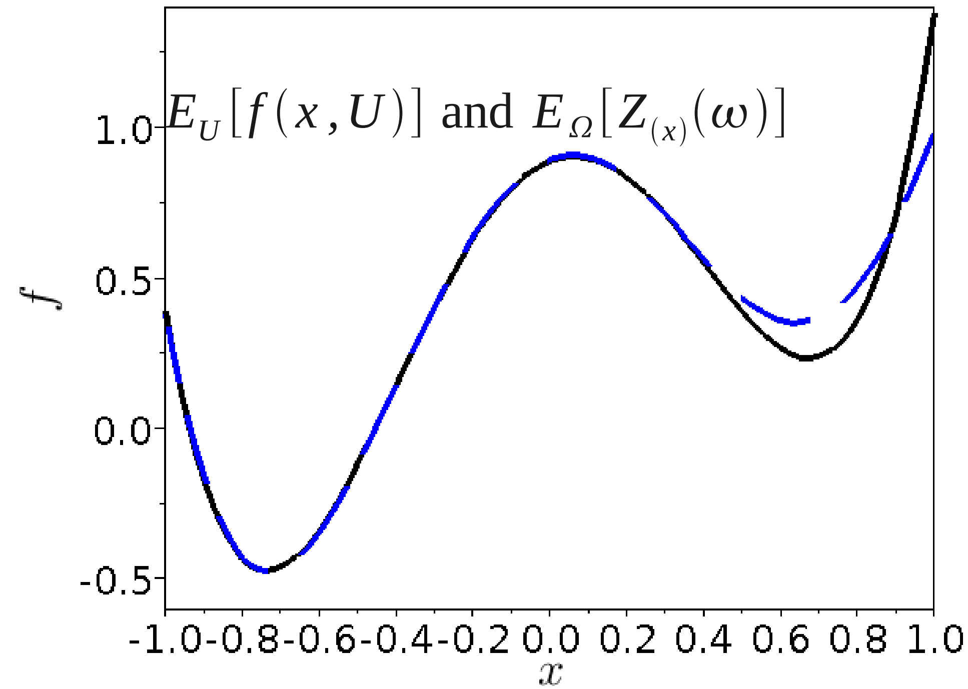


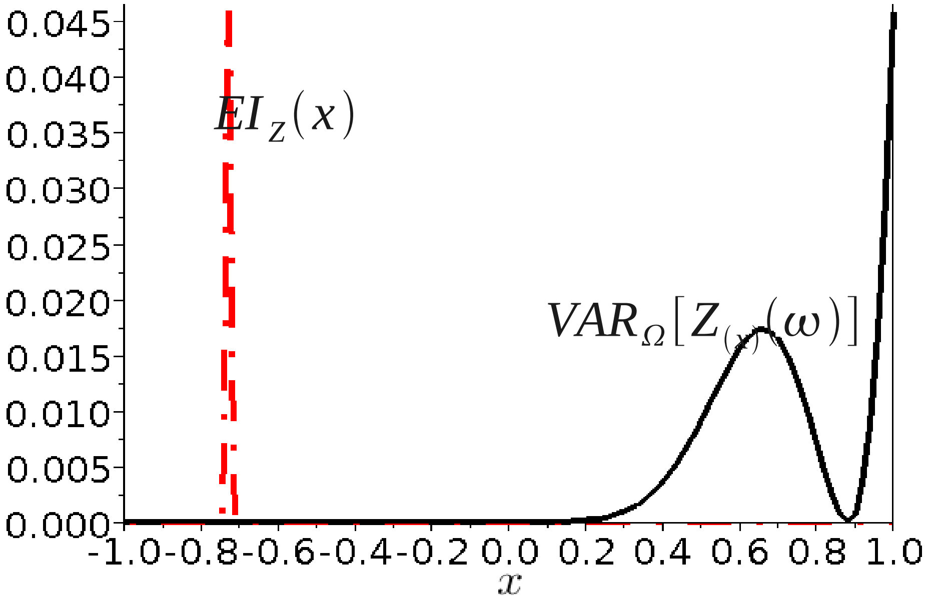
5^{th} Iteration



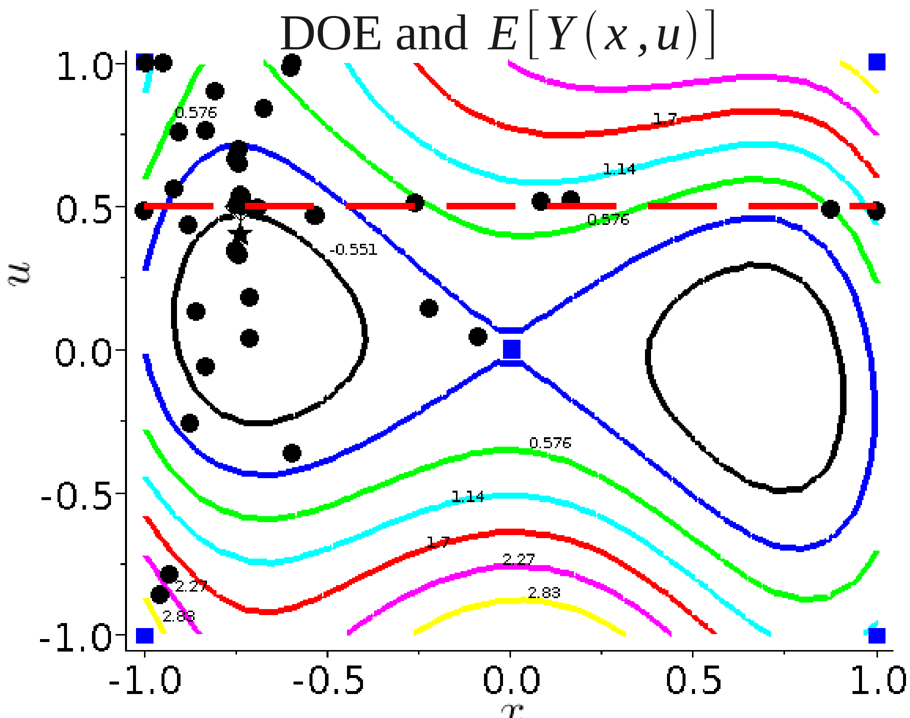
17^{th} Iteration

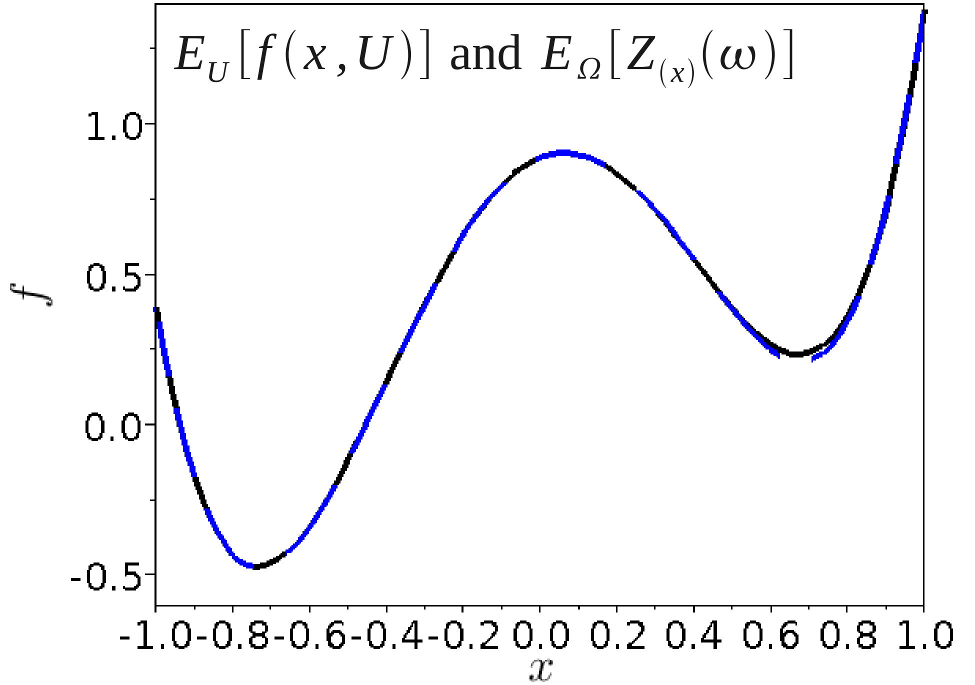


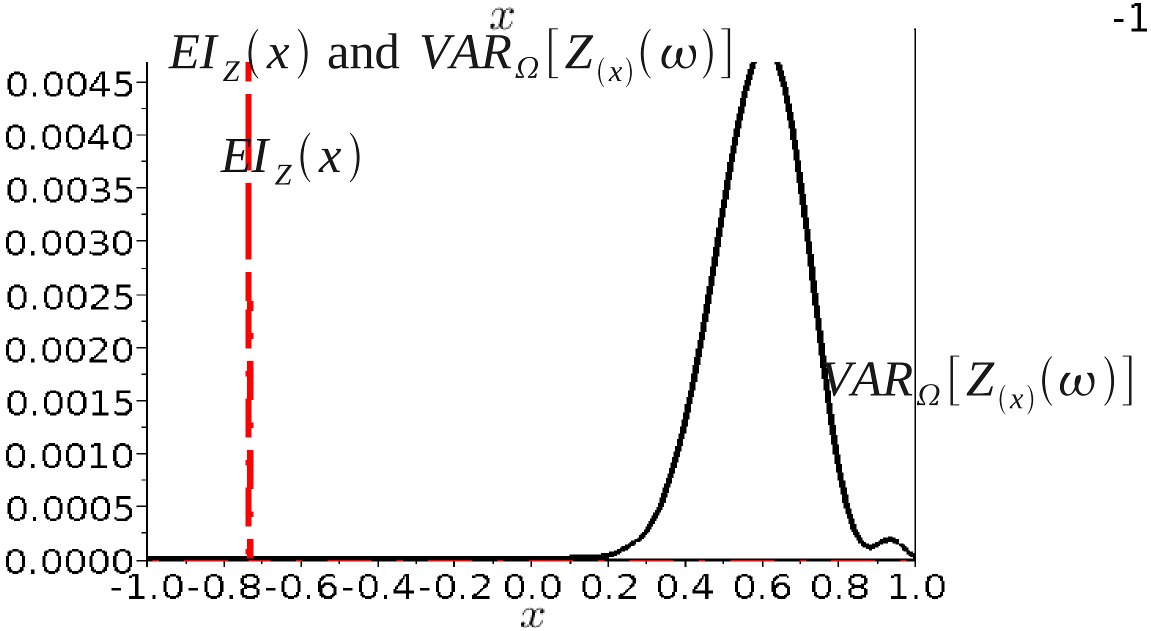




50^{th} Iteration







Comparison with MC based optimizer

Compare to:

1. EGO based on MC simulations with fixed number of runs.
   Kriging is used to filter homogenous noise.
   Use best observation for fmin.

$$I_x^t(\omega) = \left[f_{min}^t - Y_{(x)}^t(\omega) \right]^+$$

2. Simplified method where

$$x^{t+1} = x^{next}$$

$$u^{t+1} \sim \mathcal{N}(\mu, \sigma^2)$$

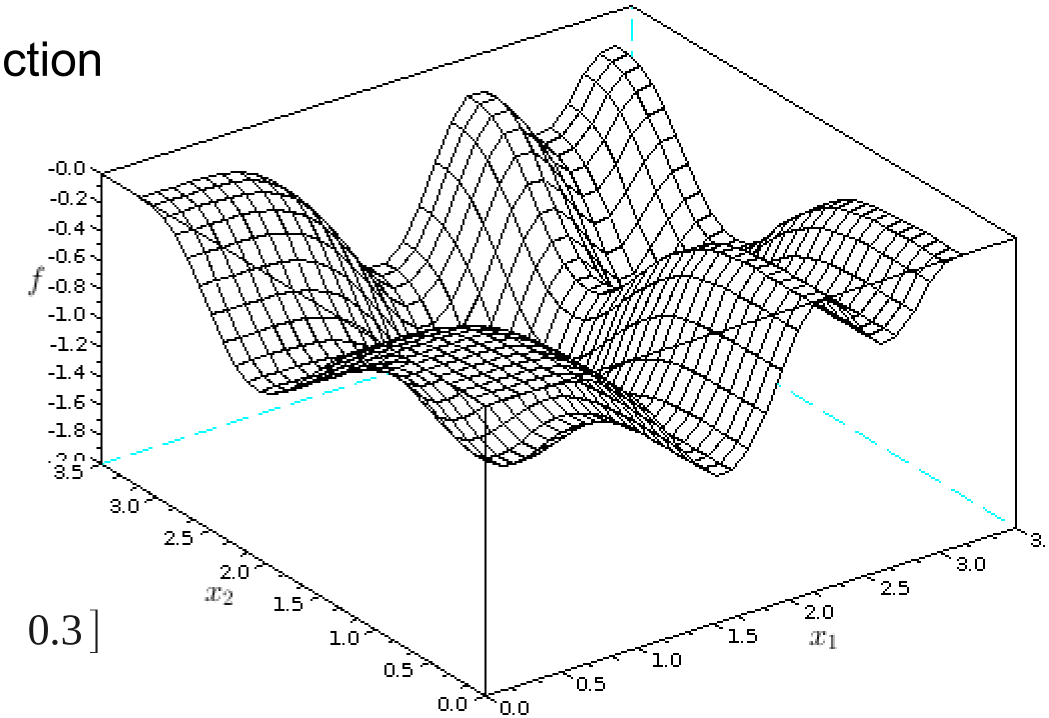
Initial DOE: RLHS (m=d*5) (for 1D case m=3);
10 runs for every test case.
|*x*-*x**| vs number of calls, where *x** is the optimum

Test case and comparison strategy

Test cases based on Michalewicz function

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

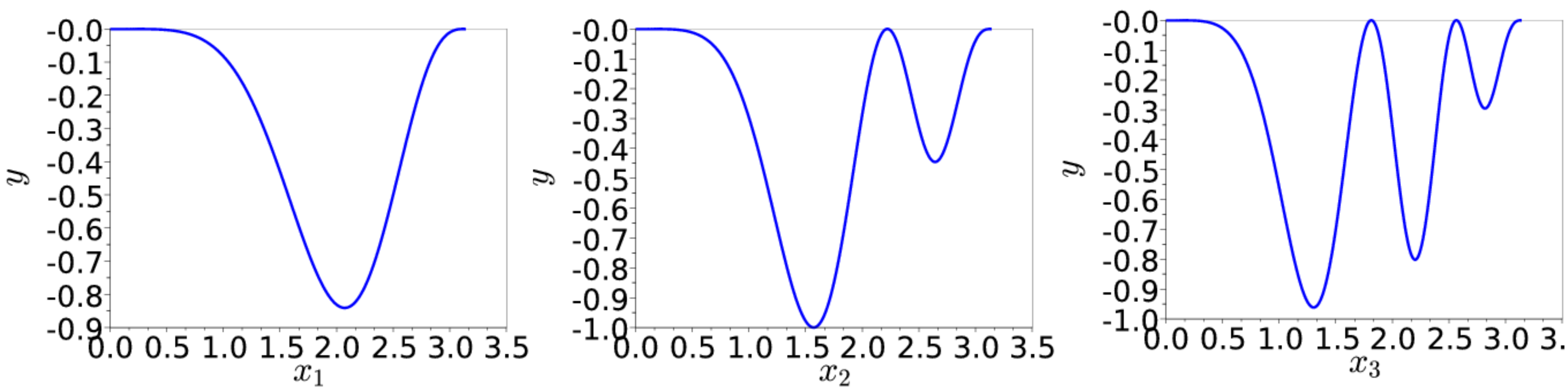
$$f(x,u) = f(x) + f(u)$$



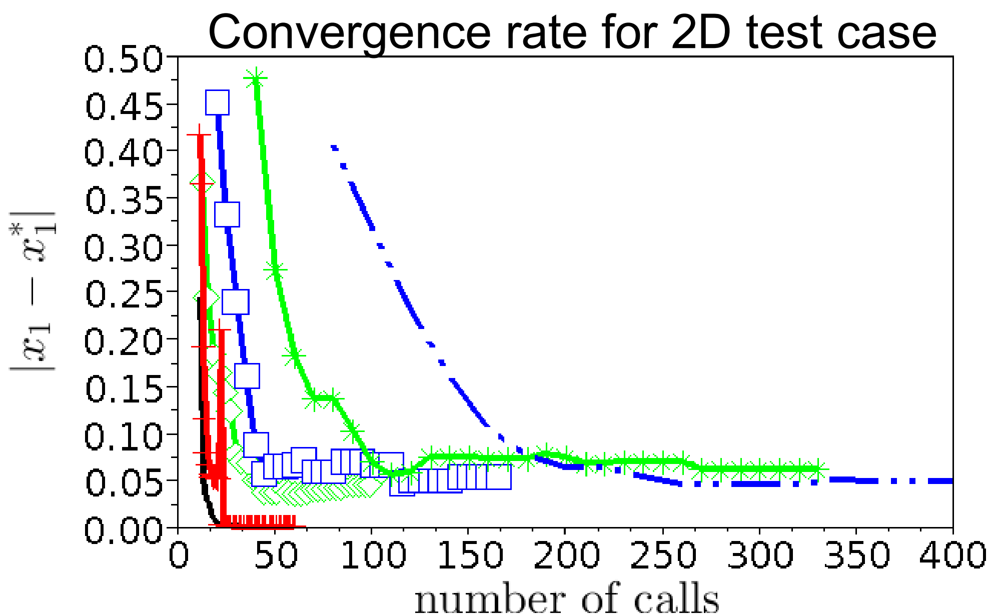
2D: $d_x=1$ $d_u=1$ $\mu=1.5$ $\sigma=0.2$

4D: $d_x=2$ $d_u=2$ $\mu=[1.5,\ 2.1]$ $\sigma=[0.2,\ 0.2]$

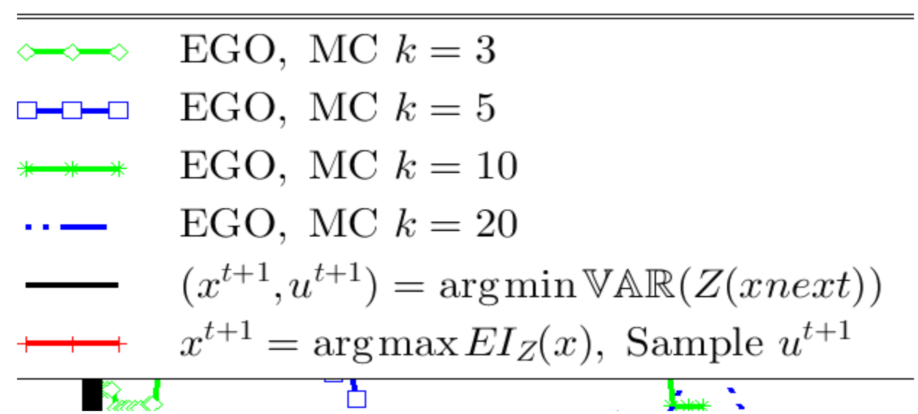
6D: $d_x=3$ $d_u=3$ $\mu=[1.5,\ 2.1,\ 2]$ $\sigma=[0.2,\ 0.2,\ 0.3]$

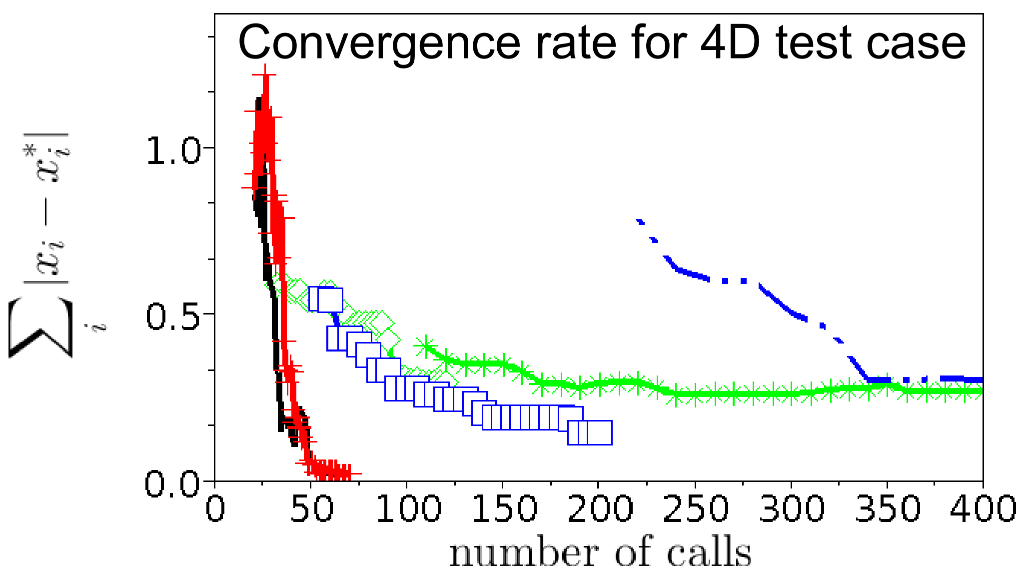


Comparison with the direct approach (EGO +MC)



$$d_x = 1 \quad d_u = 1 \quad \mu = [1.5] \quad \sigma = [0.2]$$



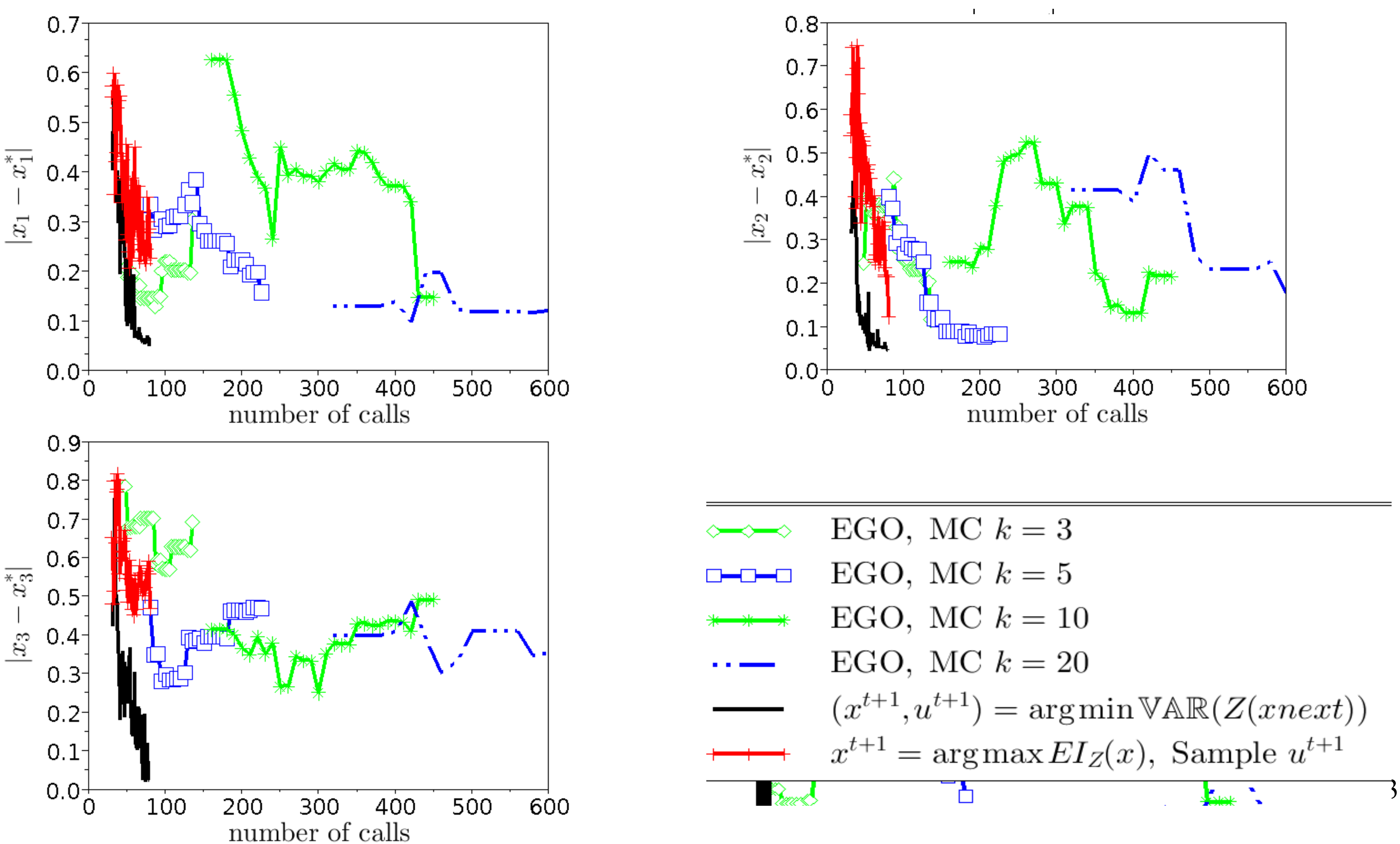


$$d_x = 2 \quad d_u = 2 \quad \mu = [1.5, \; 2.1]$$
$$\sigma = [0.2, \; 0.2]$$

Comparison with the direct approach (EGO +MC)

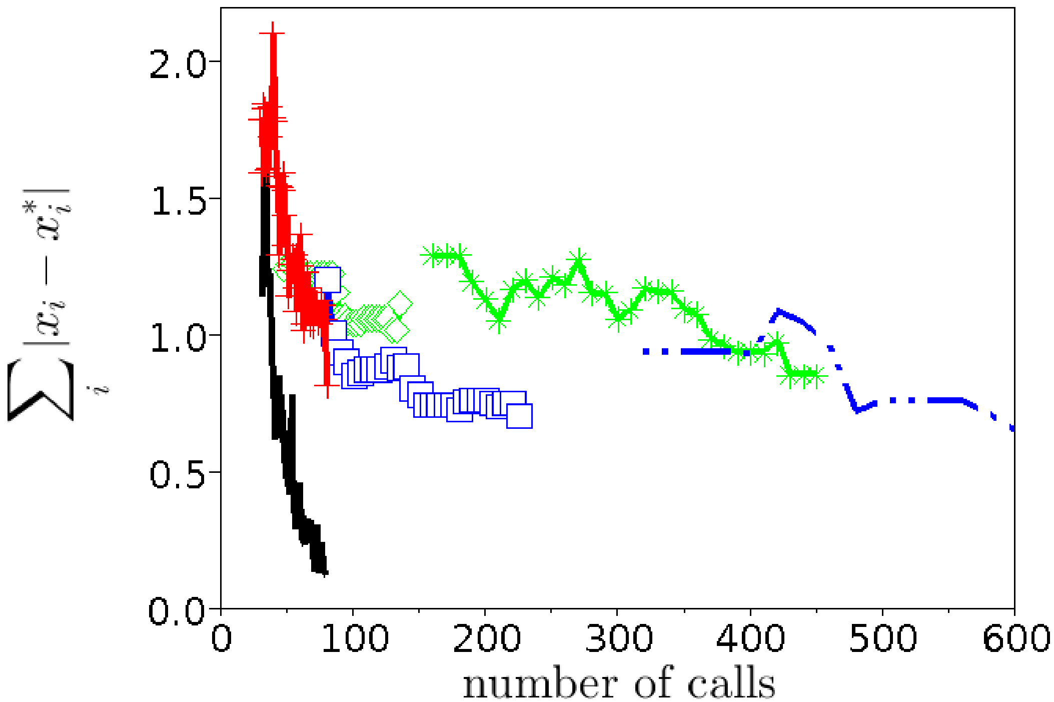
$$d_x = 3 \quad d_u = 3 \quad \mu = [1.5, \ 2.1, \ 2] \quad \sigma = [0.2, \ 0.2, \ 0.3]$$

Component wise convergence rates for 6D test case

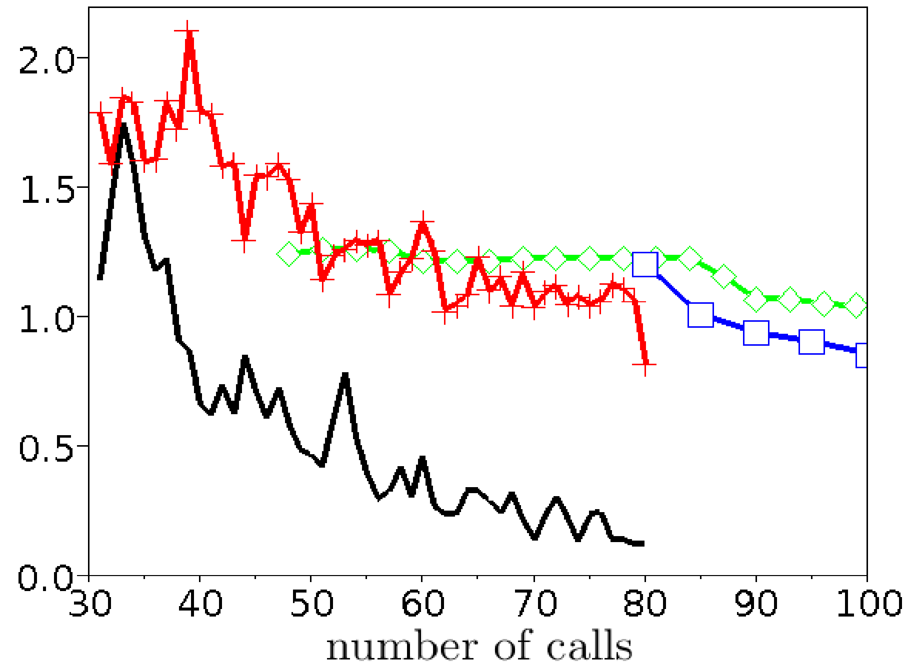


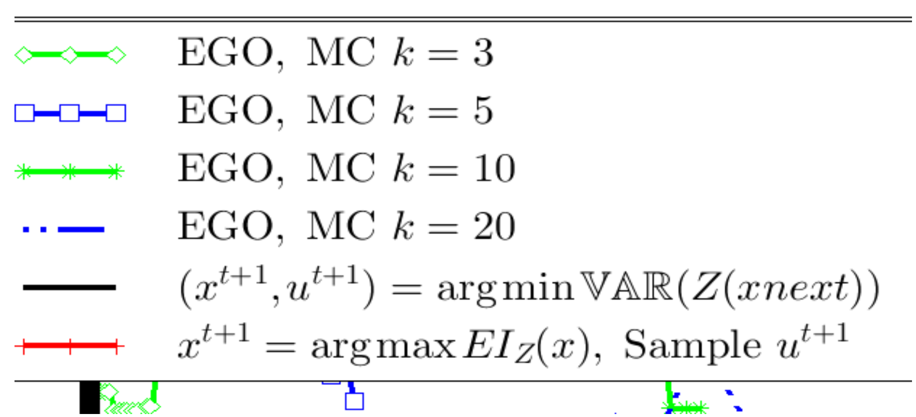
Comparison with the direct approach (EGO +MC)

Convergence rate for 6D test case

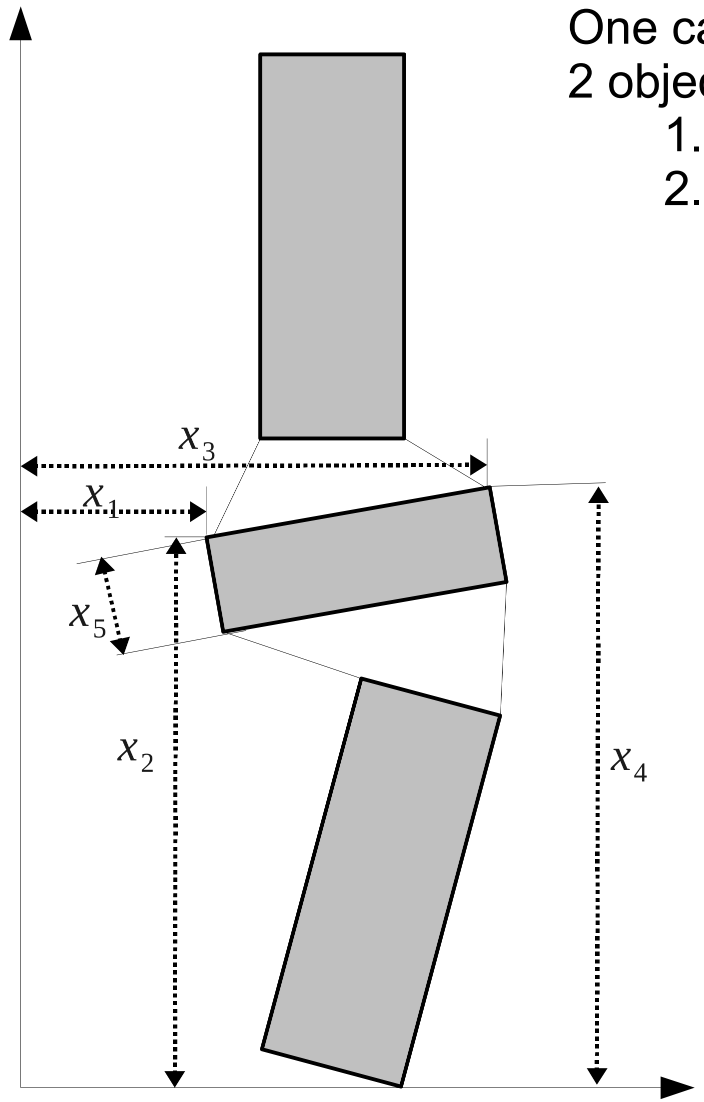


Convergence rate for 6D test case (zoomed in)



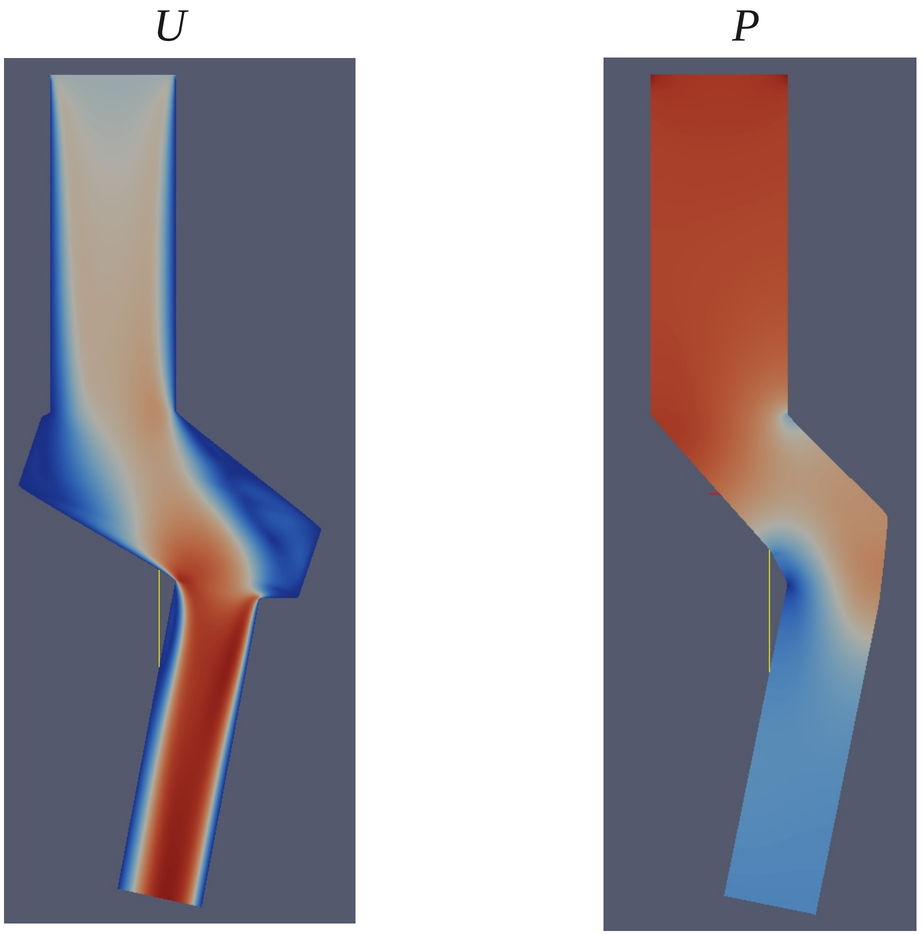


Application to ANR/OMD2 test case

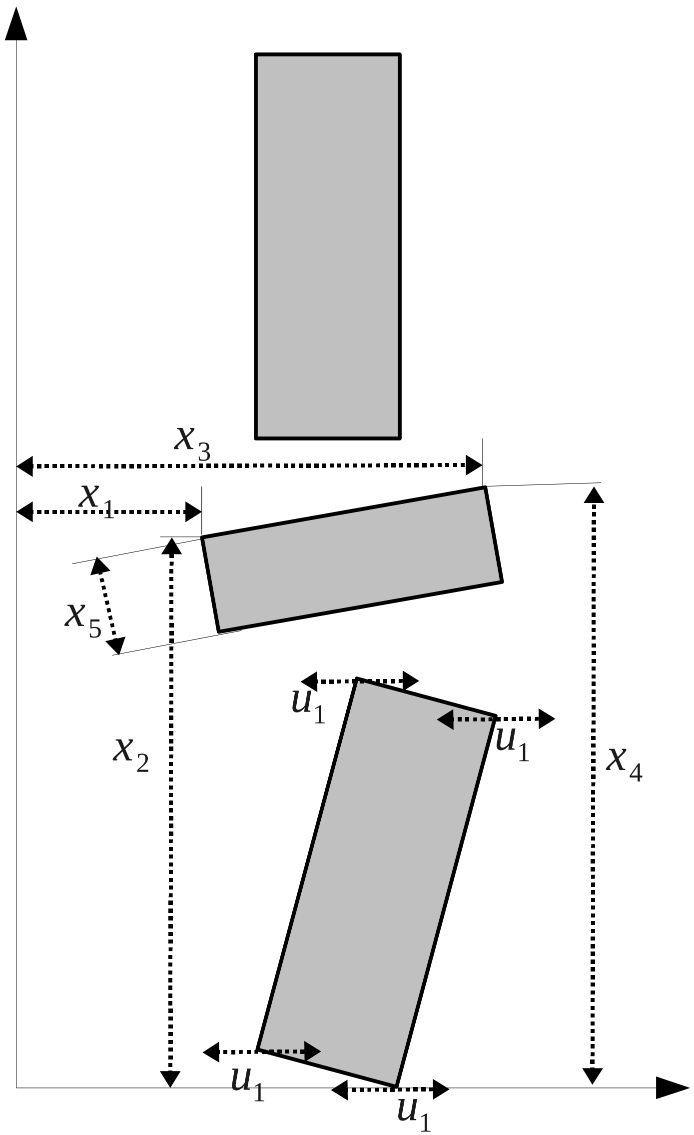


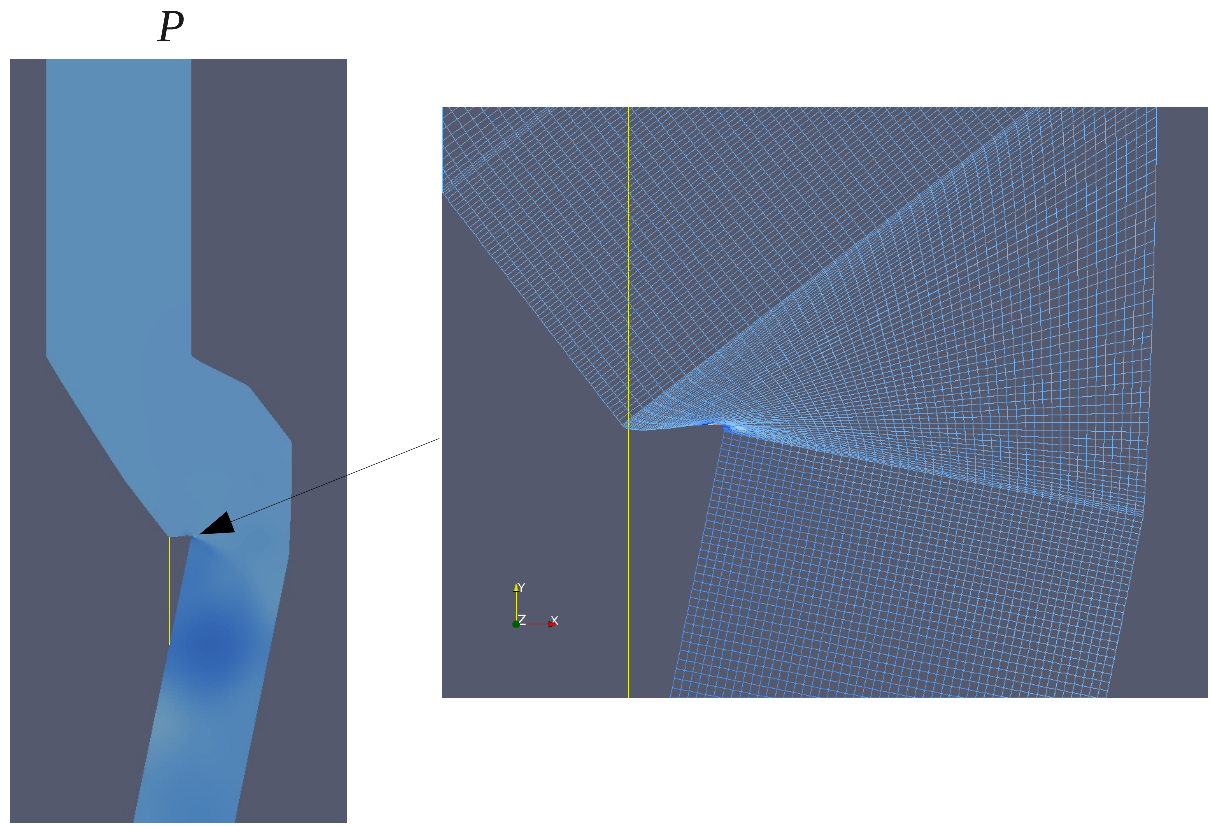
2D conditioner pipe CFD model in OpenFoam;
5 shape parameters;
One callculation up to 5 min.
2 objectives:

1. maximize uniformitiy of the flow velocity at the output.
2. minimize pressure drop between input and output.



2D shape optimization with uncertainties





Advantages and disadvantages

Advantages :

- Efficient ! (for expensive simulators in less than 10 dimensions)
- Do not have to choose number of the MC simulations.
- Obtains kriging model of the simulator and expectation, may be used for other purposes (variance of f, sensitivity analysis, ...).

Disadvantages :

•The performance is dependent on the kriging's ability to capture the underlying simulator in the joint space (future research: unstationary, non continuous simulators → need for other kernels).

• Suitable for about less than 10 dimensions. Maybe less advantageous w.r.t. MC approaches when number of *u* dimensions increases.

• Involves 4 suboptimization tasks (suitable only for costly simulators).