Learning and optimization of noisy systems using Gaussian processes

Victor Picheny, CERFACS, Toulouse

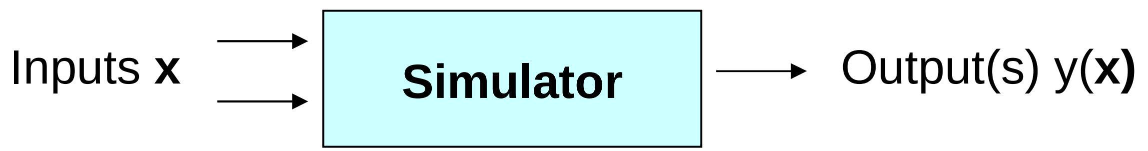
Joint work with:

D. Ginsbourger, University of Bern

Y. Richet, IRSN

Context: study of “expensive” computer experiments

- Examples
  - Finite elements: structure, fluids…
  - Reliability assessment



- Objectives: optimization / inversion / reliability analysis

- Number of runs very limited

- Popular solution: use of **meta-models**

Some contributions

- Adaptive designs of experiments for reliability analysis
- Optimal designs for kriging
- Space-time models for approximating partially converged simulations
- **Noisy optimization**

The optimization problem considered

- Single objective, unconstrained:

$$\min_{x \, in \, D} y(x)$$

- Dimension: 1 to 20
- Black-box approach (no derivatives)
- Expensive = 20 to 1000 runs
- Noisy: $y_{obs} = y(x_{obs}) + \varepsilon_{obs}$

Assumptions:

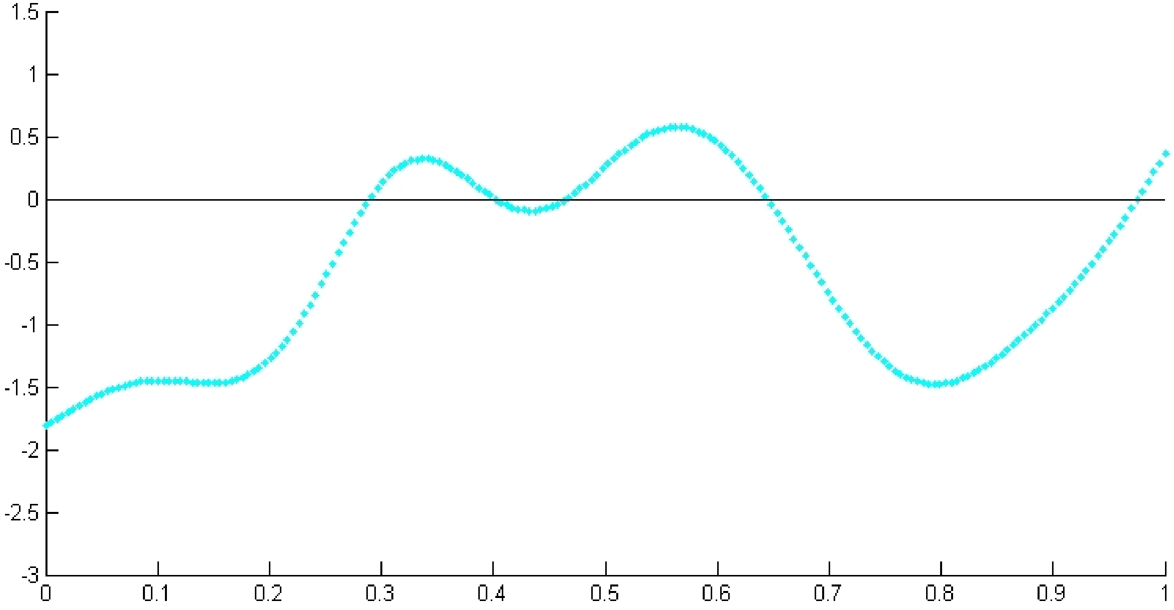
- Independent, centered, Gaussian noise
- Known noise variance
- Repeatable experiments

Outline

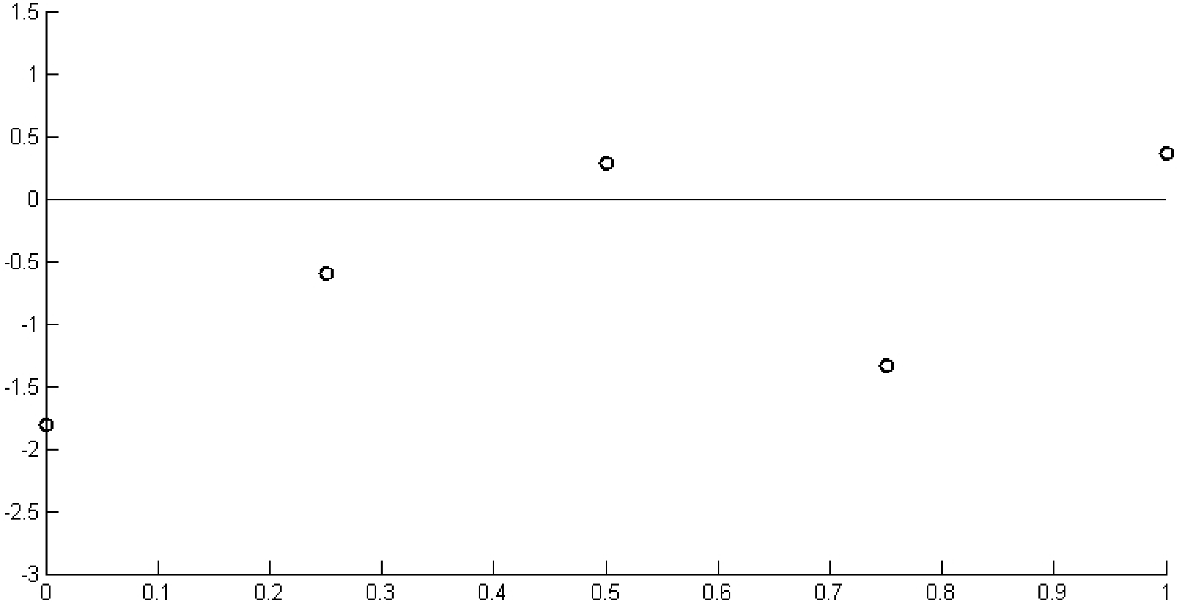
- Gaussian processes and optimization
- An infill criterion adapted to noisy problems
- Taking advantage of tunable precision
- One-dimensional example
- Application to a nuclear engineering problem

I- Basics of Gaussian process based optimization

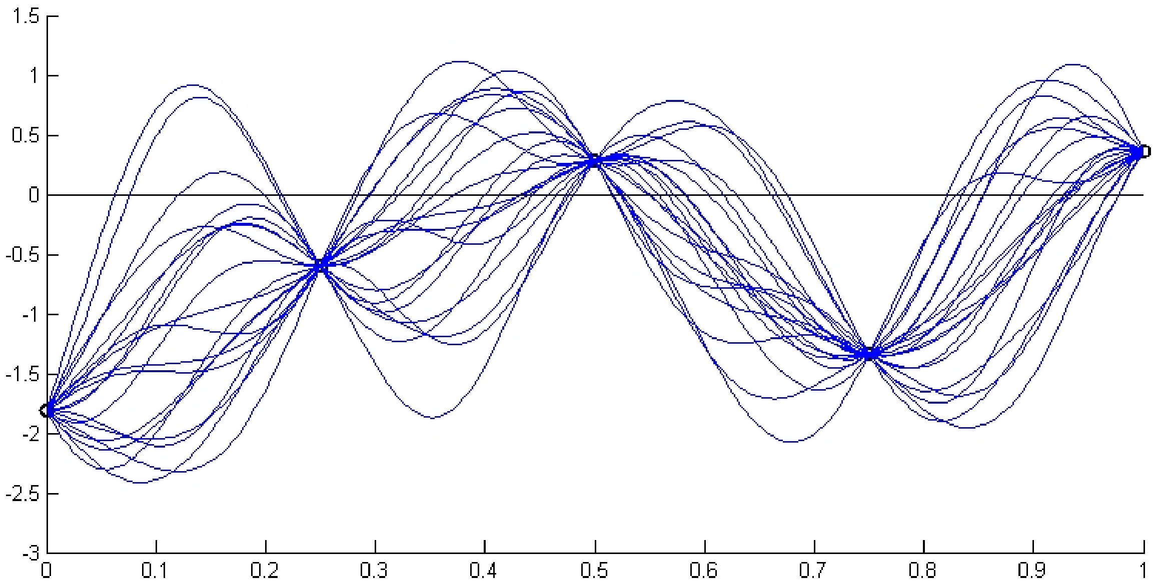
Introduction to kriging



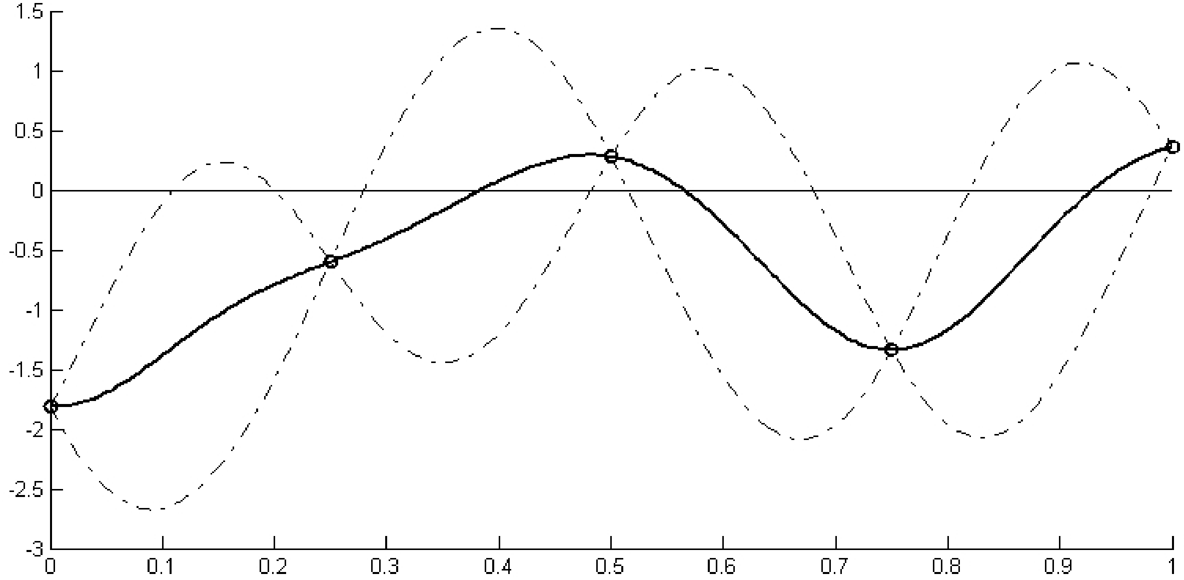
Introduction to kriging



Introduction to kriging

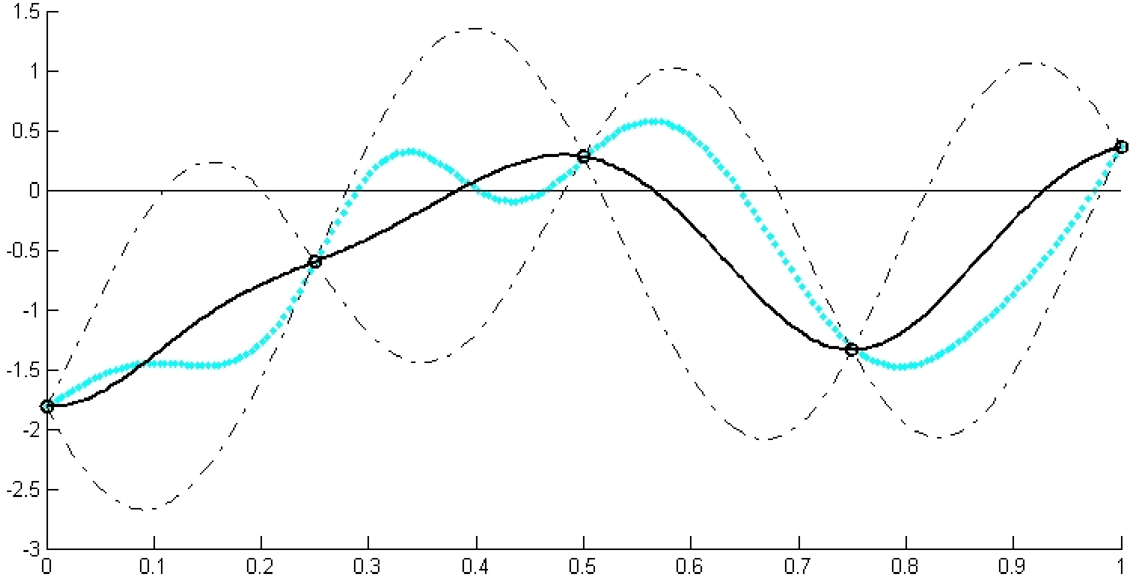


Introduction to kriging



Introduction to kriging

Probabilistic metamodel: associates a distribution to a prediction point instead of a scalar: $F(x) \sim N\left(m_K(x), s_K^2(x)\right)$



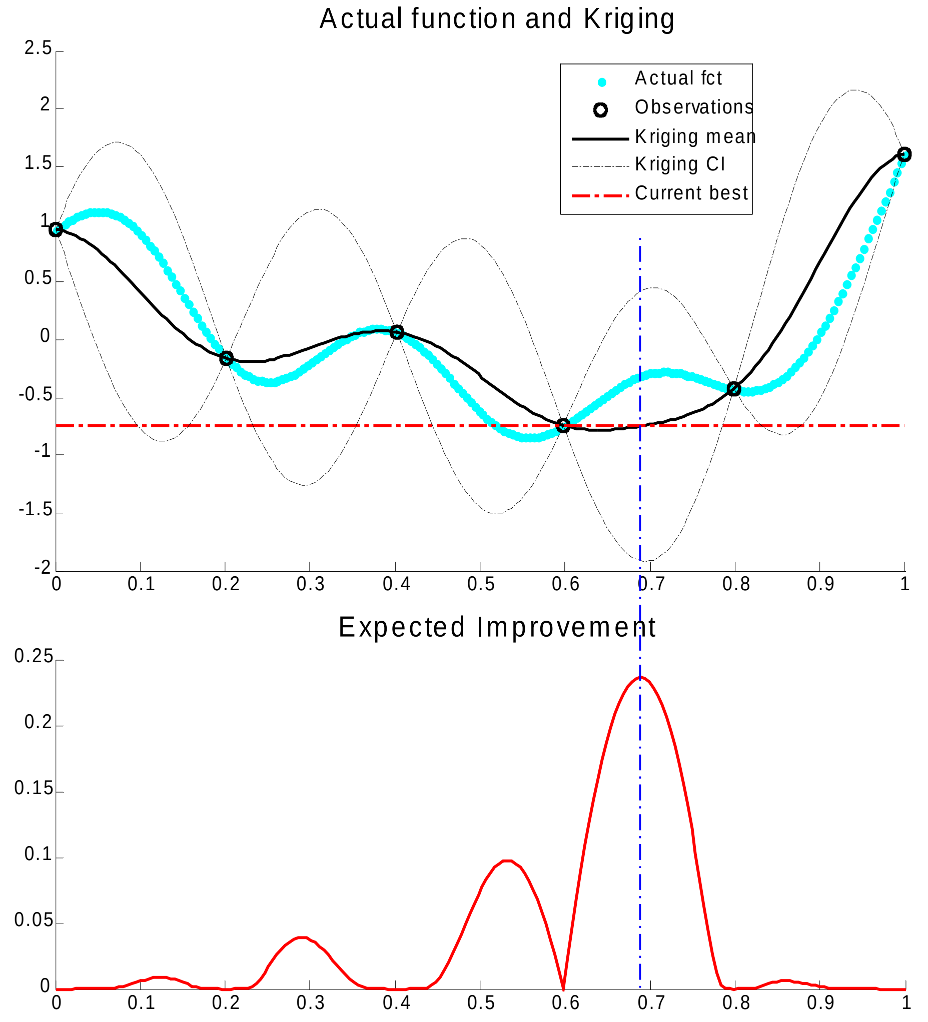
$m_K(x)$: kriging mean

$s_K^2(x)$: prediction variance

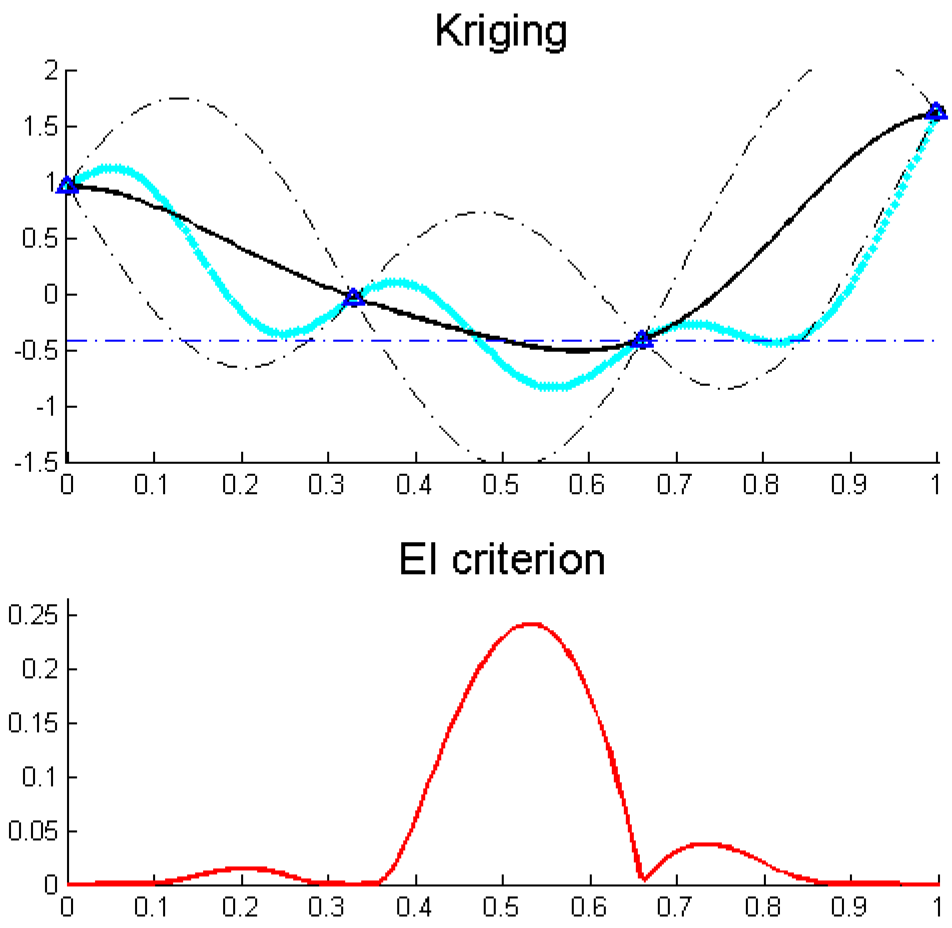
Very "rich" information for optimization!

Optimization with Kriging: the EGO algorithm

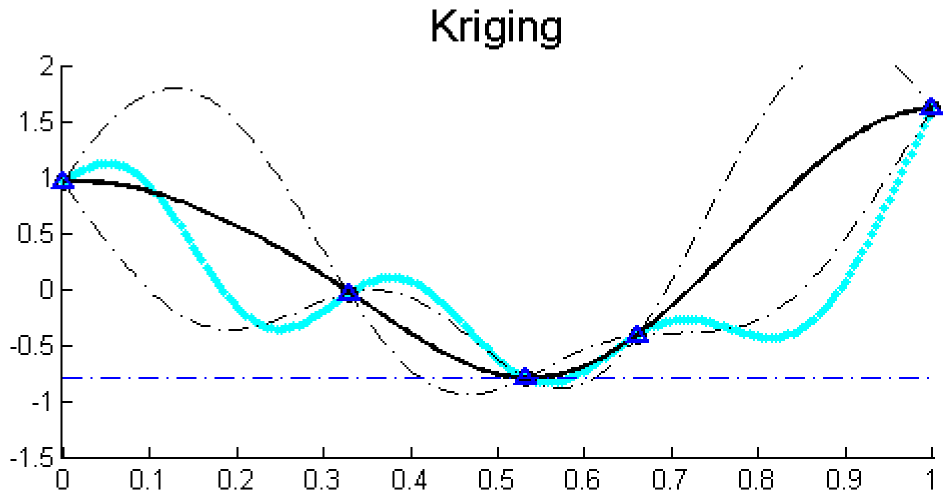
- Use of the kriging **prediction variance** to choose experiments
- Trade-off between uncertainty and potential
- At each step: maximization of the **expected improvement** (EI): $EI = \mathrm{E}(y_{min} - y_{new})^{+}$
- Noise-free only!

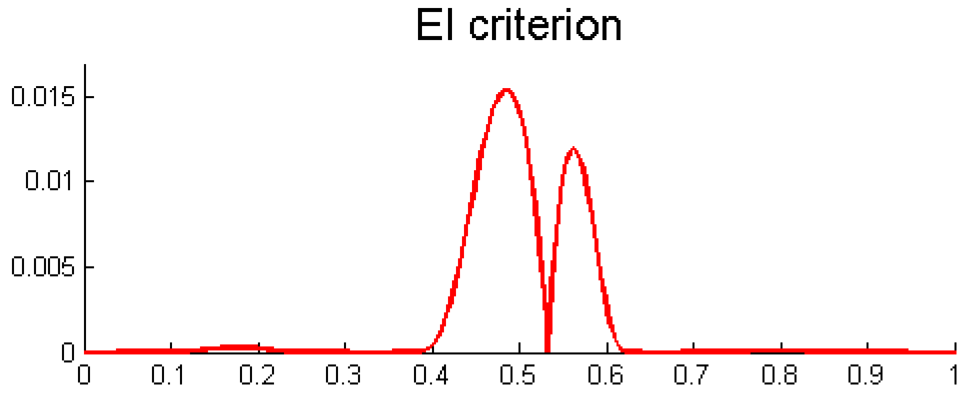


Example: initial state with 4 observations

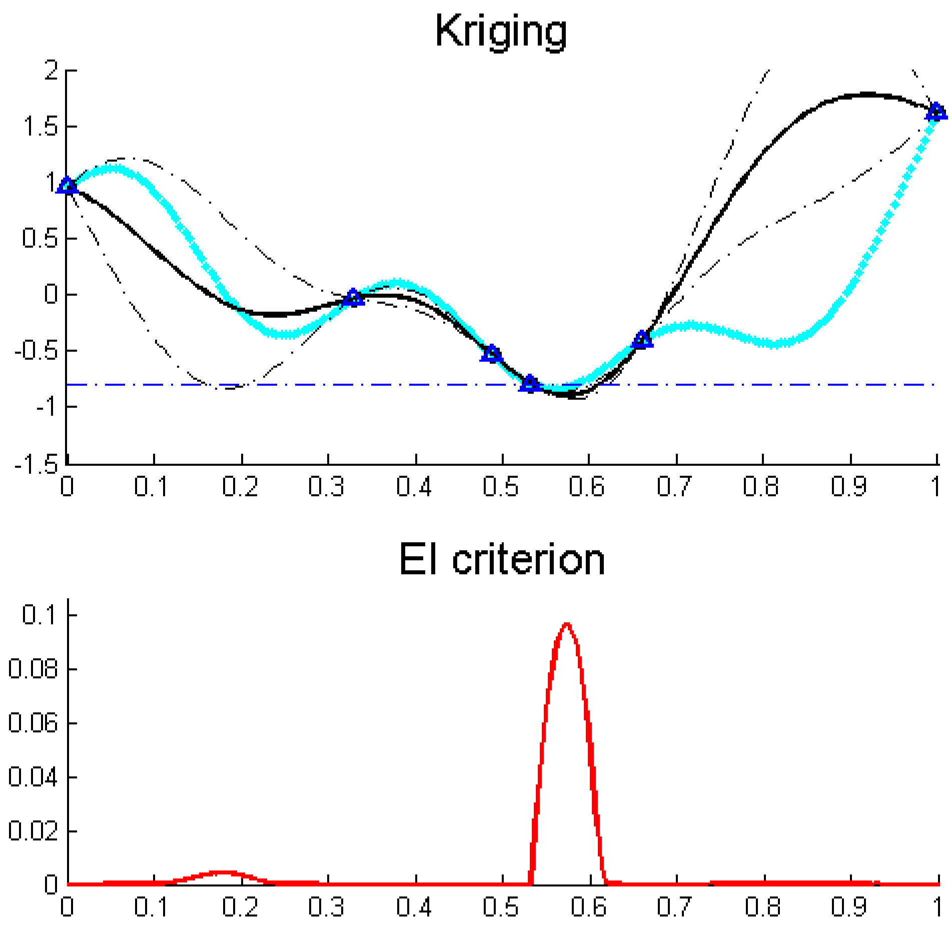


Example: 5 observations

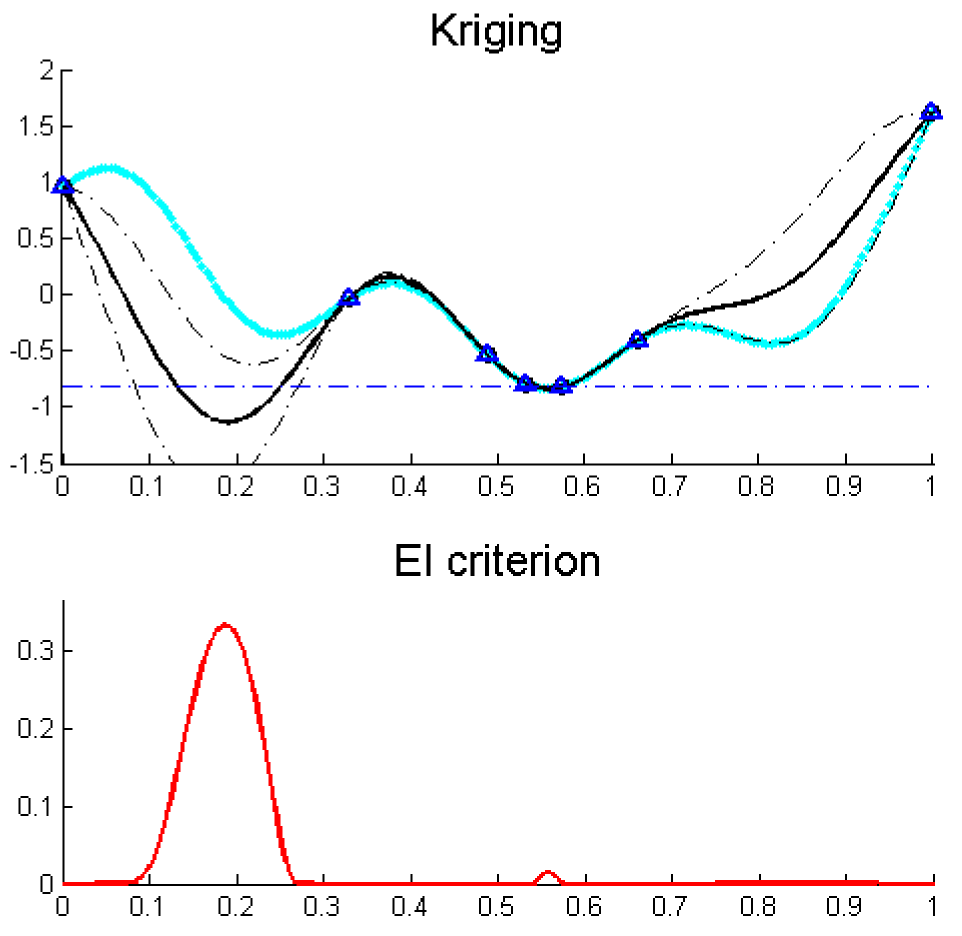




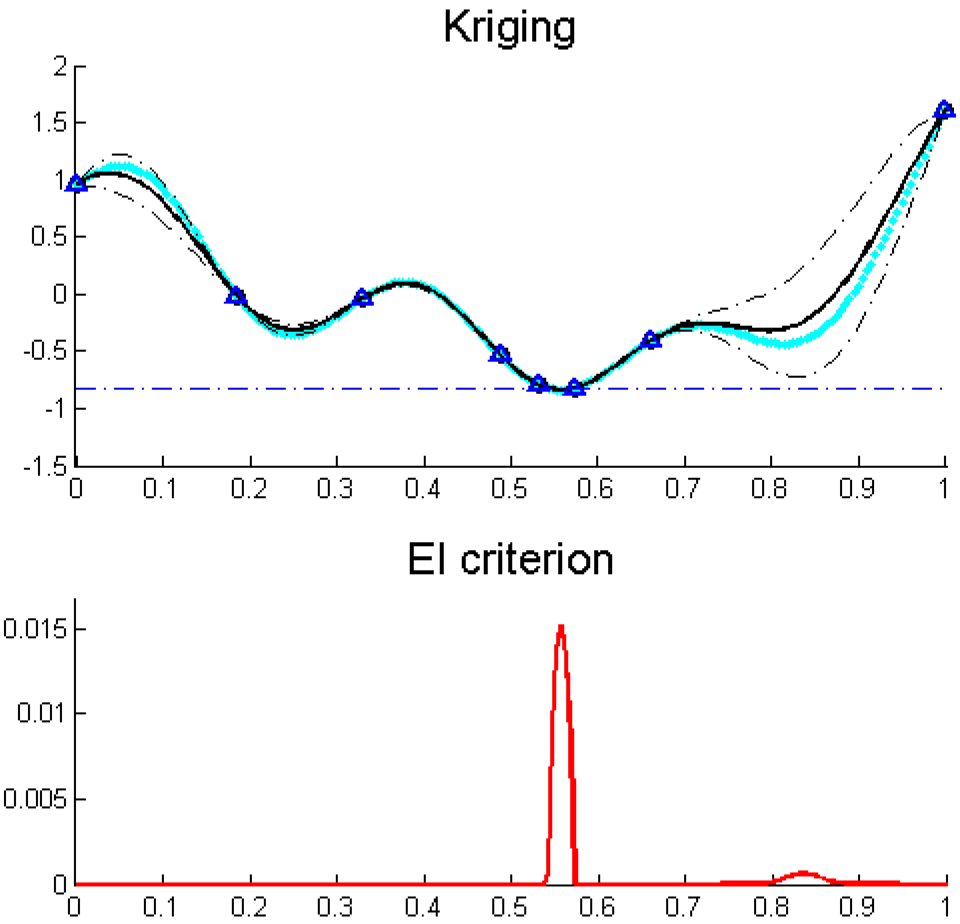
Example: 6 observations



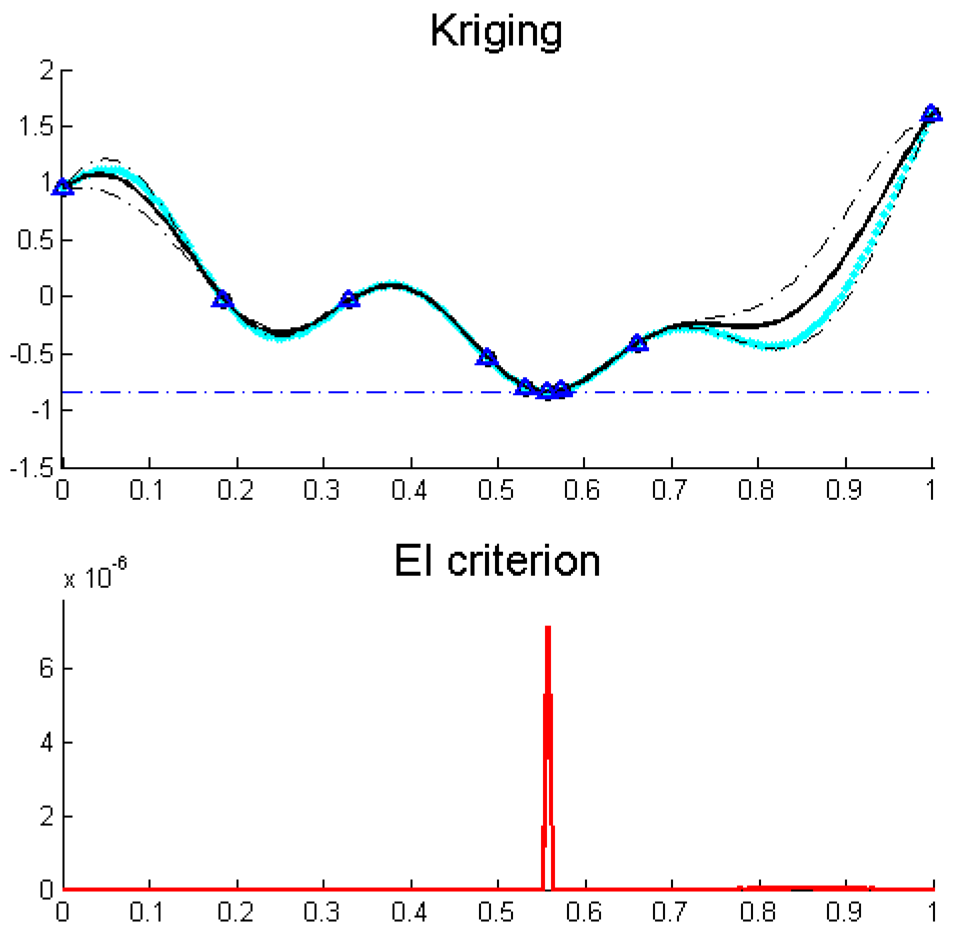
Example: 7 observations



Example: 8 observations



Example: final state: 9 observations



II- Adapting the Expected Improvement to noisy observations

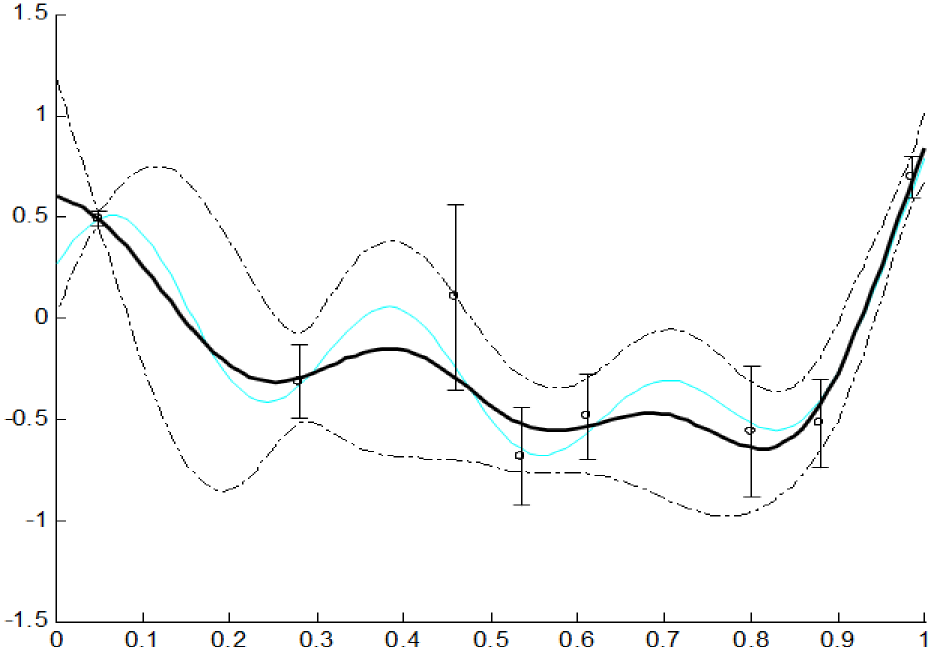
Kriging is well-suited to noisy observations

- Requires a small change in the equations:

$$m_n(\mathbf{x}) = \mathbf{c}_n(\mathbf{x})\mathbf{C}_n^{-1}\tilde{\mathbf{Y}}_n$$

$$s_n^2(\mathbf{x}) = \sigma^2 - \mathbf{c}_n(\mathbf{x})\mathbf{C}_n^{-1}\mathbf{c}_n^T(\mathbf{x})$$

With: $$\mathbf{C}_n = \begin{bmatrix} c(\mathbf{x}_1,\mathbf{x}_1)+\tau_1^2 & \cdots & c(\mathbf{x}_n,\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ c(\mathbf{x}_1,\mathbf{x}_n) & \cdots & c(\mathbf{x}_n,\mathbf{x}_n)+\tau_n^2 \end{bmatrix}$$



Limitation of EI with noise

- Two problems with classical EI:

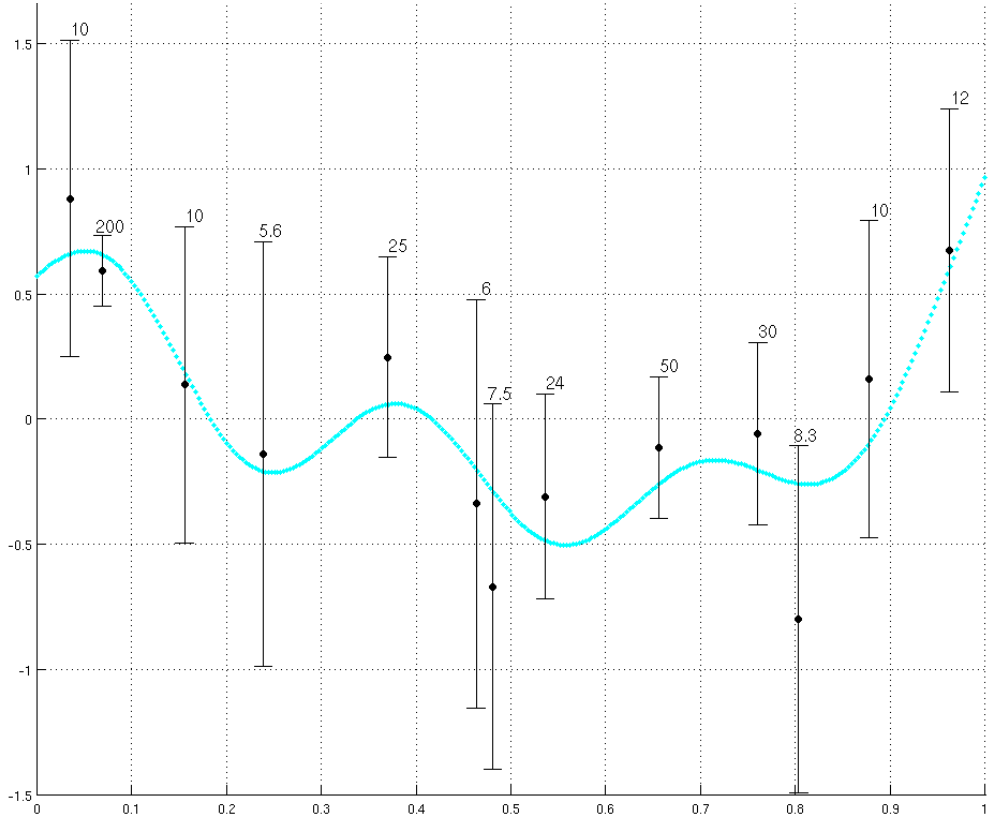
$$EI(\mathbf{x}) = E\left[\left(\underbrace{\min\left(Y(\mathbf{X}_n)\right)}_{unknown} - \underbrace{Y(\mathbf{x})}_{unreachable}\right)^+\right]$$

1. Current minimum is **unknown** due to noise
2. Future observation will also be observed in noise!

➡ We need to rethink the notion of "improvement"

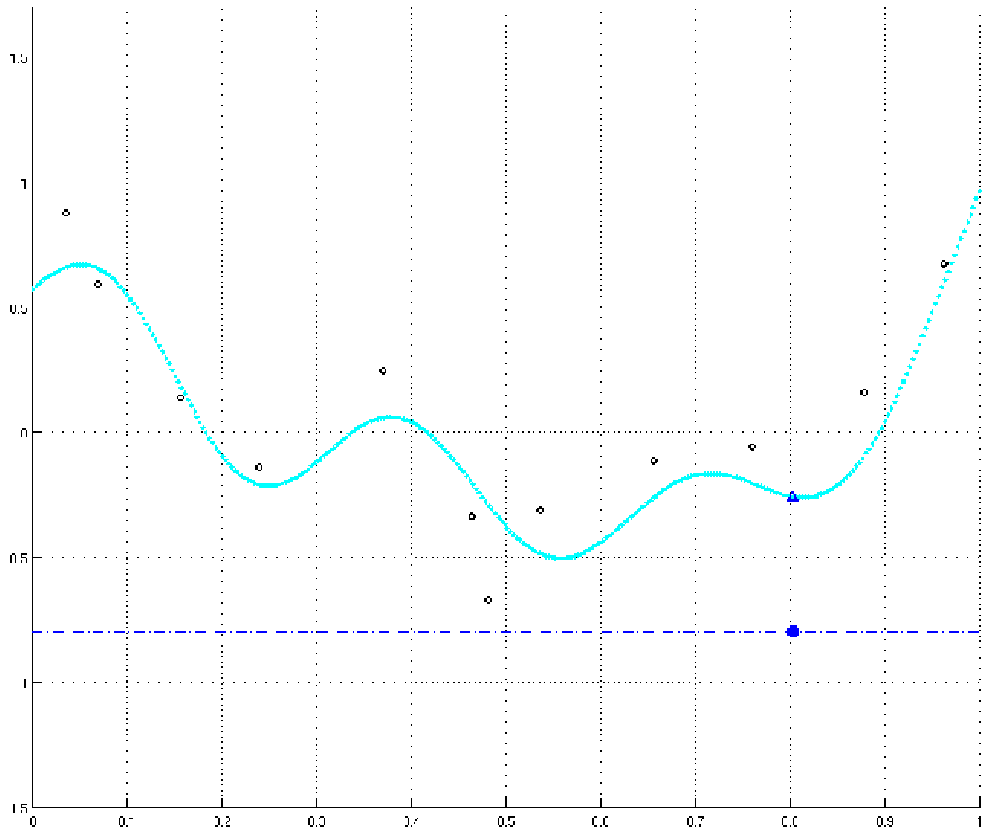
Decision-making with noisy observations

- With noise, relation-order is not maintained
  - ➡ Which design is best?



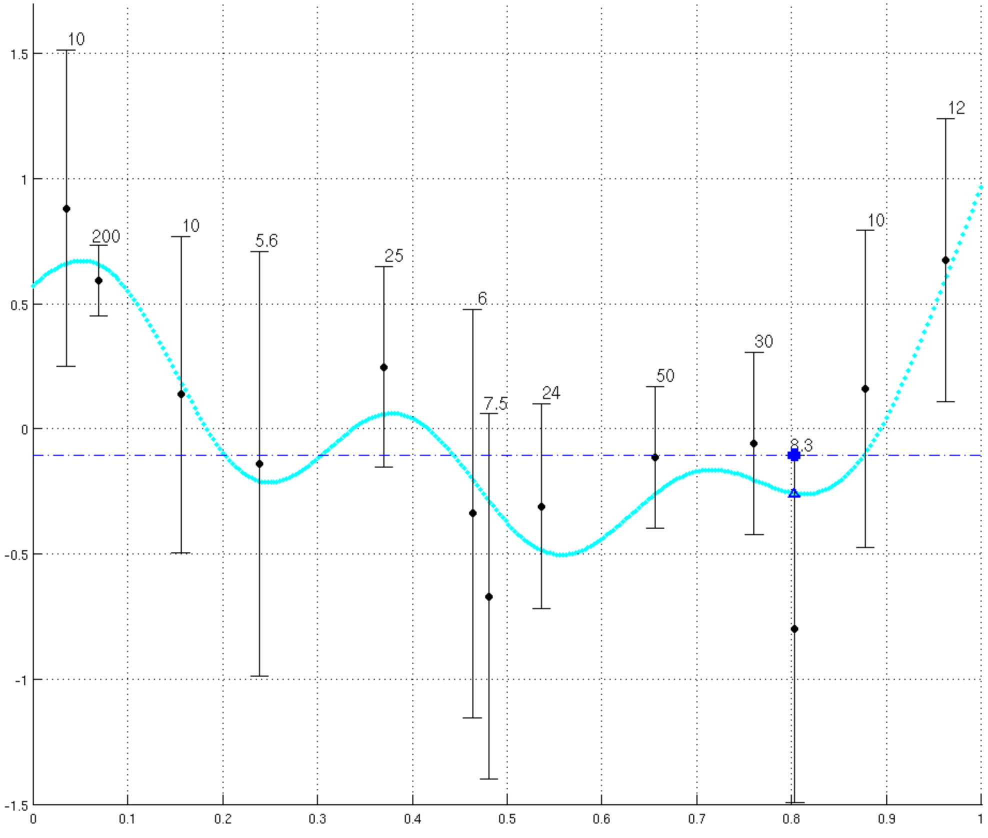
Decision-making with noisy observations

- Option 1: choose the best observation



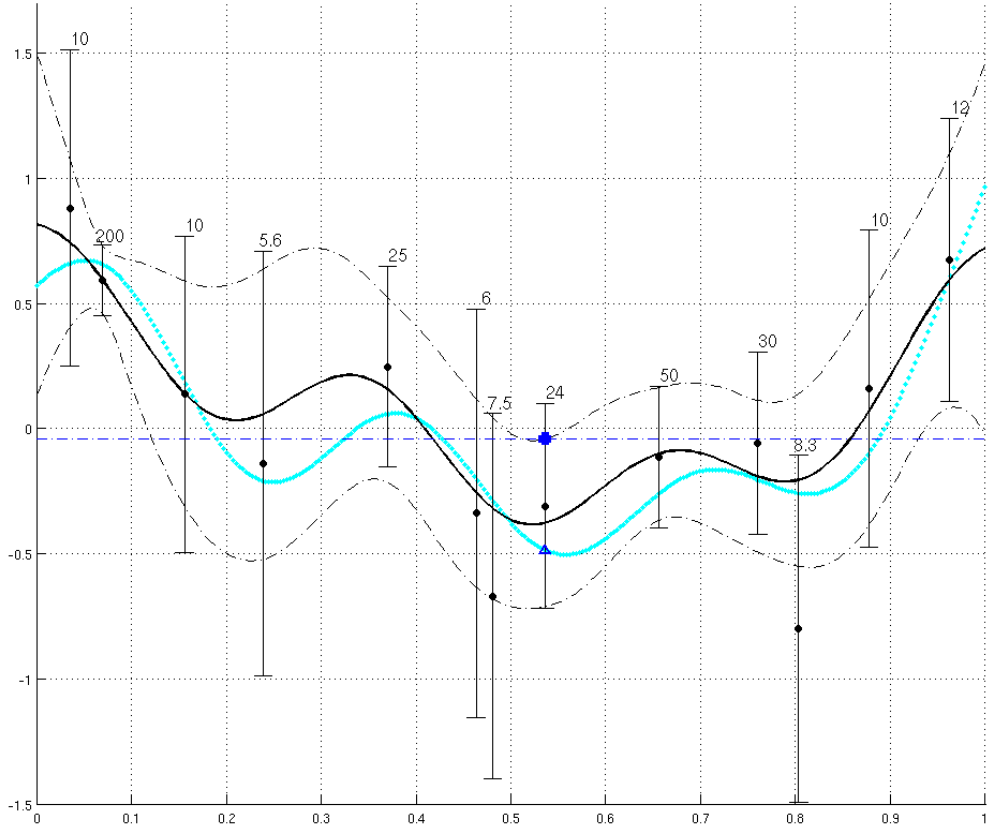
Decision-making with noisy observations

- Option 2: take the noise into account



Decision-making with noisy observations

- Option 3: use a metamodel to filter the noise
  - ➡ safe choice: **Kriging quantile**



Decision-making and “improvement”

- With noise-free function:
  - Best design = minimal observation
  - Improvement = reduction of the best observation value

- With noisy observations:
  - Best design = minimal quantile
  - Improvement = reduction of best quantile value
  - Expected Quantile Improvement:

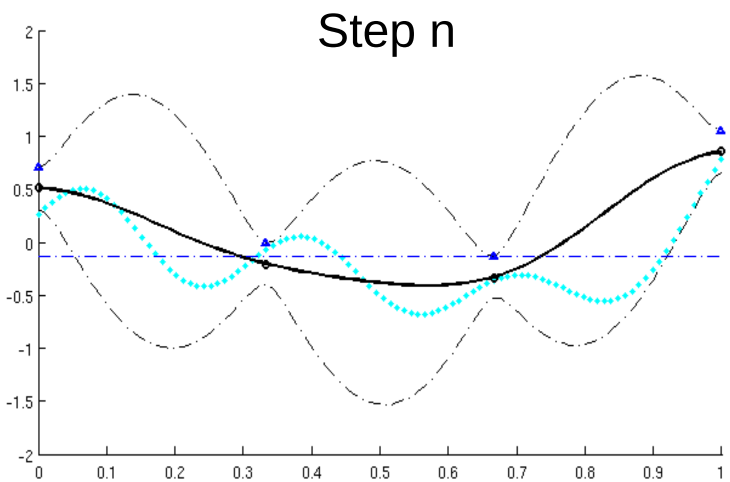
$$q_{min} = \min_{i=1\ldots n} \quad m_n(x_i) + \alpha \times s_n(x_i)$$

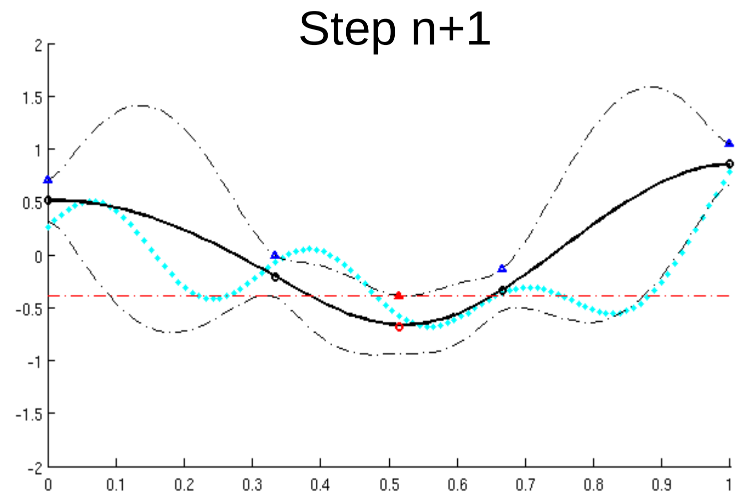
$$EQI(\mathbf{x}) = E\left[\left(q_{min} - q_{new}(\mathbf{x})\right)^+\right]$$

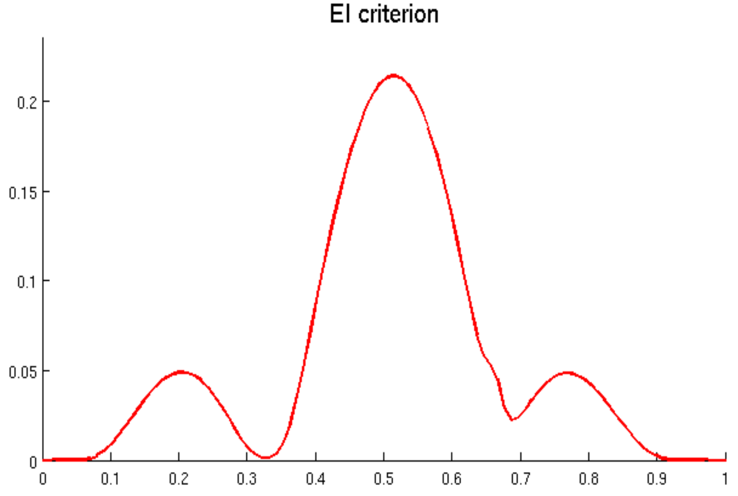
- *EQI* is **analytically tractable** (using Gaussian Process conditioning… and cumbersome calculations)

- Depends on past and future noise

Illustration

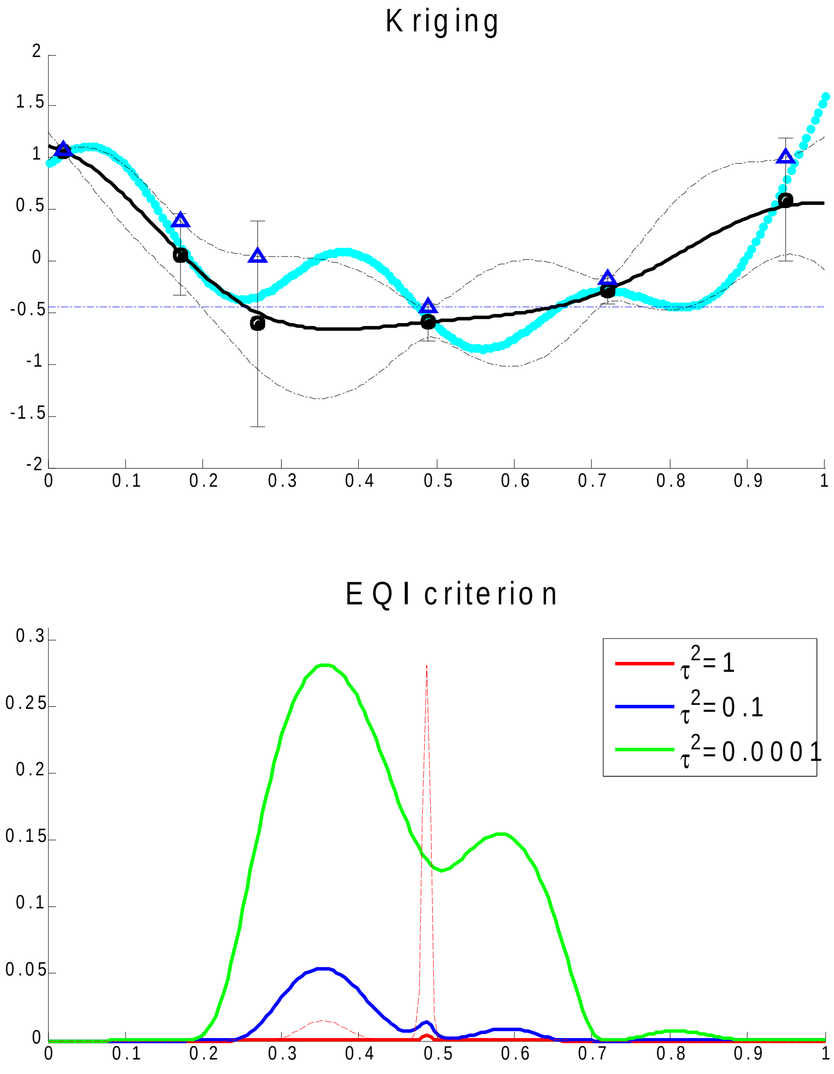






- Future noise variance is 0.02
- Actual improvement here is 0.26.

Influence of future noise level



- Criterion computed for several noise levels of the new observation
- With small noise: equal to classical EI
- With large noise:
  - New observation does not change the Kriging
  - EQI is maximum at data points

Conclusion of part II

- The EQI criterion
  - Allows rigorous treatment of noise
  - Is analytical
  - Reflects the final user decision
  - Depends on future noise
- The values of quantile level and future noise affect greatly the shape of the EQI
- Open question: one-step quantile improvement vs. global quantile minimization

III- Taking advantage of tunable precision (and replications)

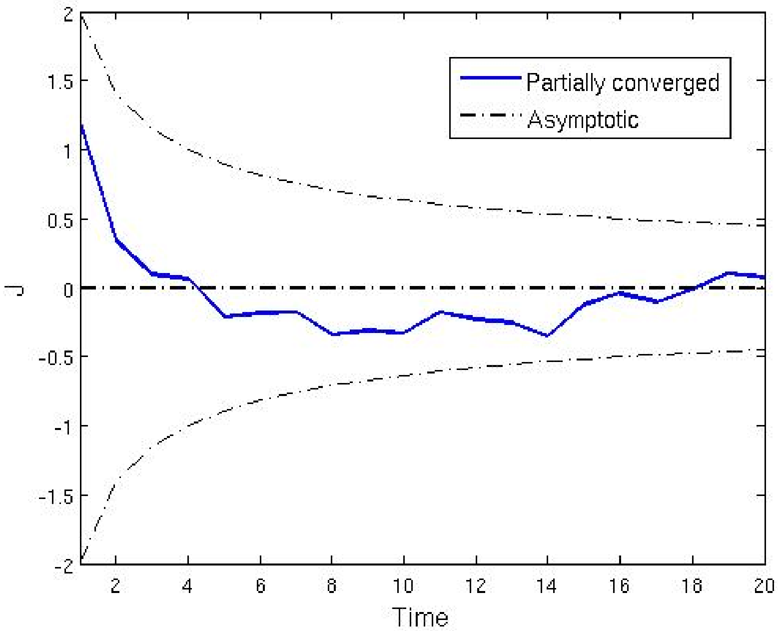
What is tunable precision?

- Many simulators depends on parameters that tune the precision
- Two examples:
  - Partially converged simulations (solver number of steps)
  - Monte-Carlo simulators (sampling size)
  - (Repeated experiments)
- Each observation is a trade-off between rapidity and accuracy
- Objective: use the tunable precision as an additional degree of freedom

Assumptions

- Noise variance decreases with computational time
- Relation between variance and time is assumed to be known
- The Monte-Carlo case:

$$\begin{cases} y_i = y(x_i) + \varepsilon_i \\ \varepsilon_i \sim N\left(0, \tau^2(x_i, t_i)\right) \\ \tau^2(x_i, t) = \dfrac{y(x_i)}{t} \end{cases}$$



- Response convergence is **tractable** ***on-line***

Key concepts and objectives

- On-line allocation
  - Allocate computational time adapted to each design
  - Detect when adding computational time will not provide valuable information
  - Allows early stop / accurate simulations
- Finite time strategy
  - Computational time is limited by resources and simulator complexity
  - Our trade-off is necessarily driven by this limitation

Using EQI for online allocation

- EQI allows us to choose the next experiment given a future noise variance
- EQI can be updated on-line
  - By updating kriging with current observation value
  - By updating future noise variance *(see next slide)*
- EI measures by how much we can improve our decision
- Proposition: use it as a *point-switching* criterion
  - EQI decreases when observation becomes accurate
  - If the design is 'better than expected': EQI increases
  - If the design is 'worse than expected': EQI decreases faster
- To avoid too many EQI optimization: switch point when: $EQI < EQI_{init}/2$

Choice of the noise level for on-line allocation

- Natural idea: evaluate the interest of a **single time step**

  ⟹ EQI would show by how much we expect to decrease the quantile with one time step

- Problem: EQI would be ≈ zero everywhere

- Proposition: use the value of the **smallest noise achievable**
  - Noise can be bounded by the user (solver tolerance)
  - Noise is always bounded by the computational resource

    ⟹ EQI shows the ultimate gain achievable by this observation

Consequences

- The ‘smallest noise achievable’
  - depends on the computational resource
  - increases during the optimization

- The algorithm behaves differently at the beginning and the end of the optimization
  - Beginning: enhances exploration
  - End: avoids visiting new sites

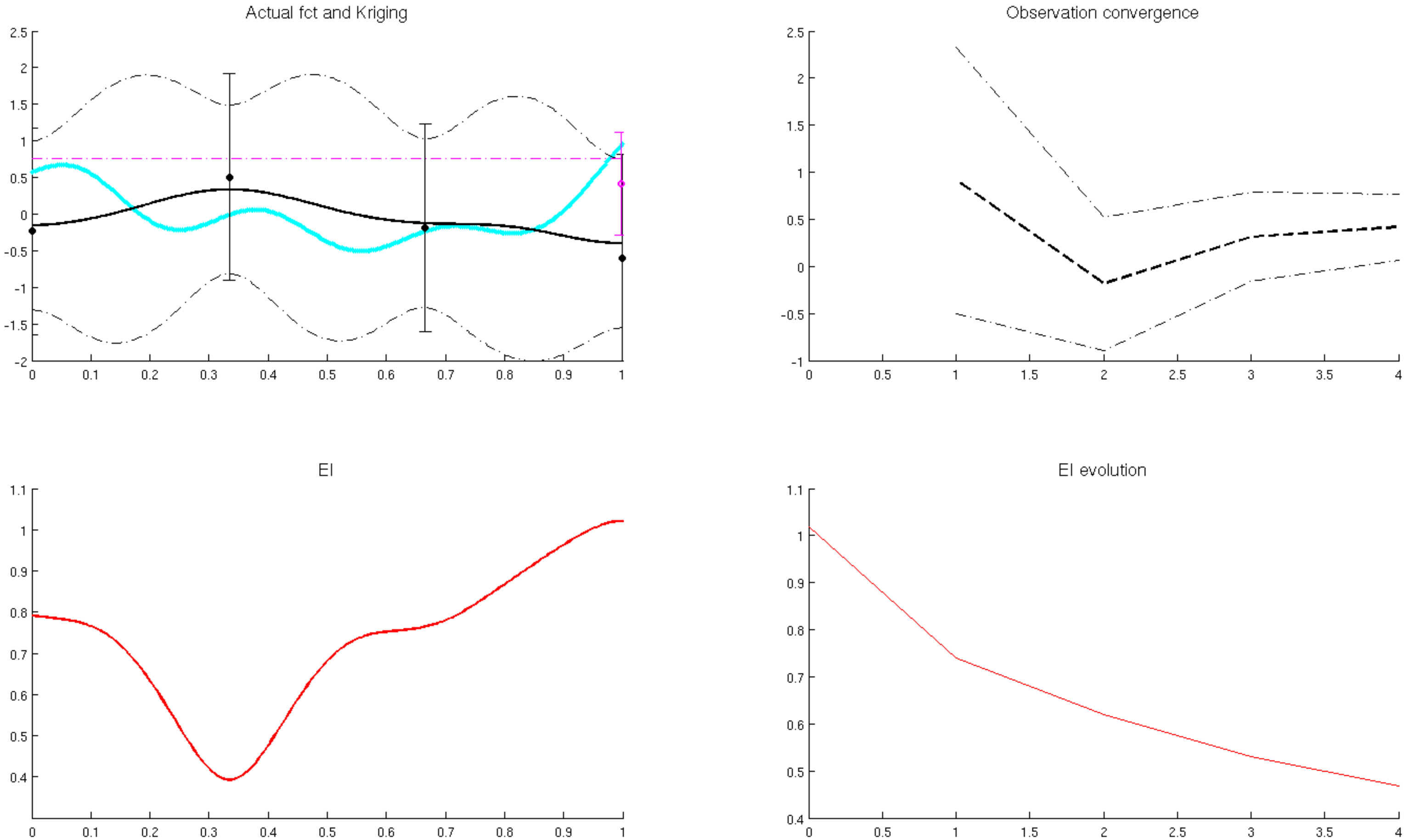
- **The strategy takes into account the limited computational resource**

1D example

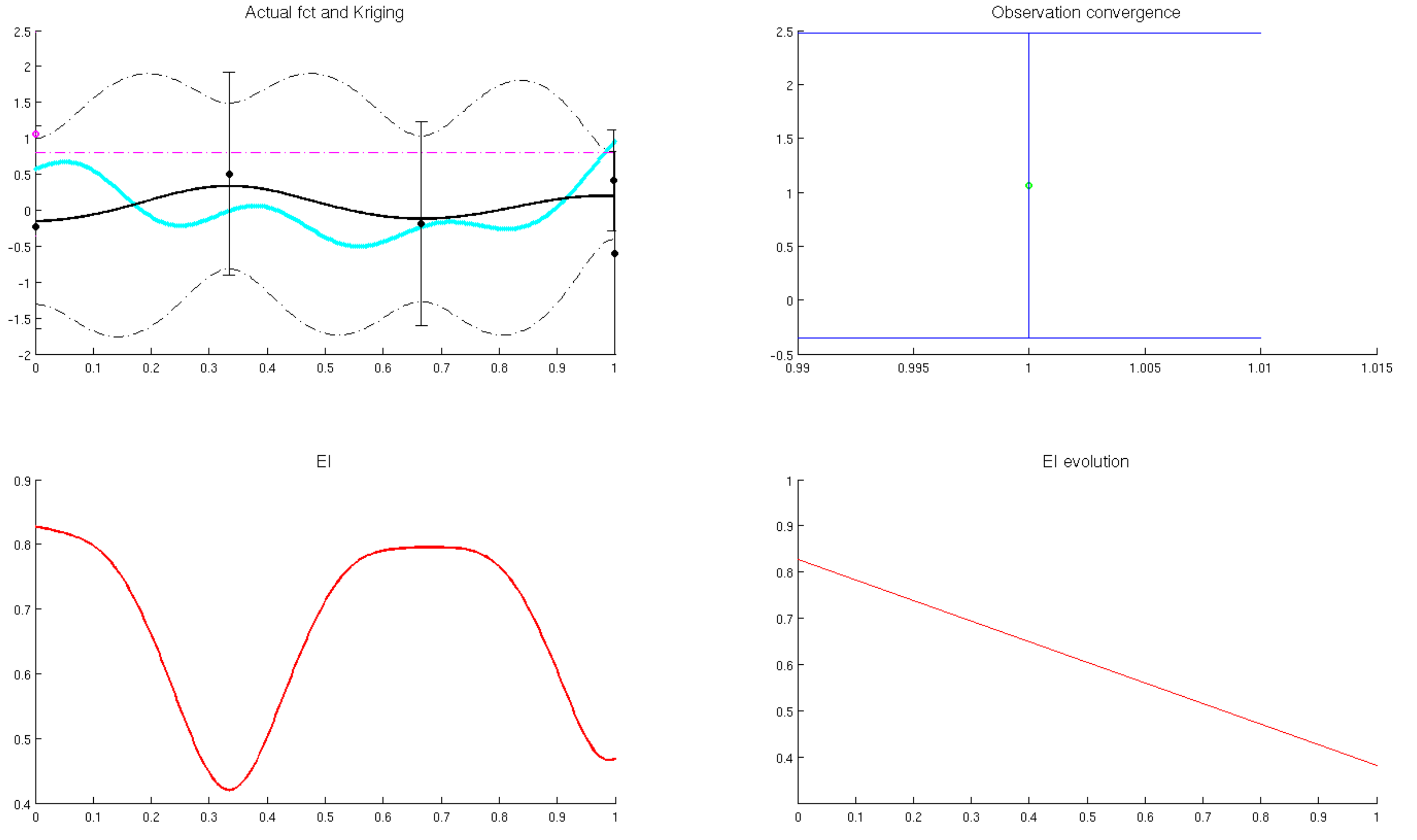
- 1D function
- Normally distributed error
- var(ε) = 0.5 / t
- Total time T = 100
- Time is divided in 100 steps

- We distinguish here:
  - Algorithm iterations
  - Time steps

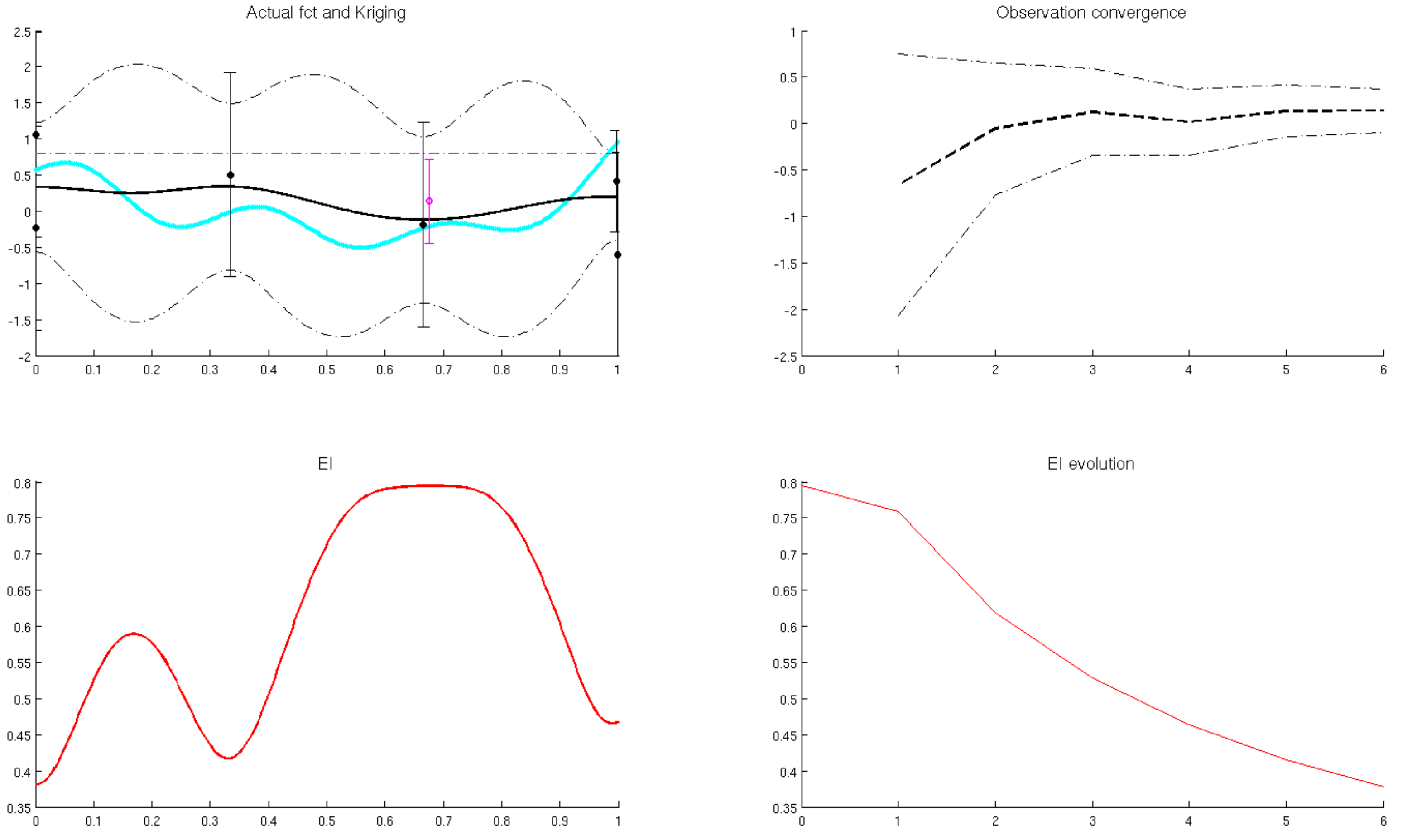
Iteration 1: 4 steps used / 92 remaining



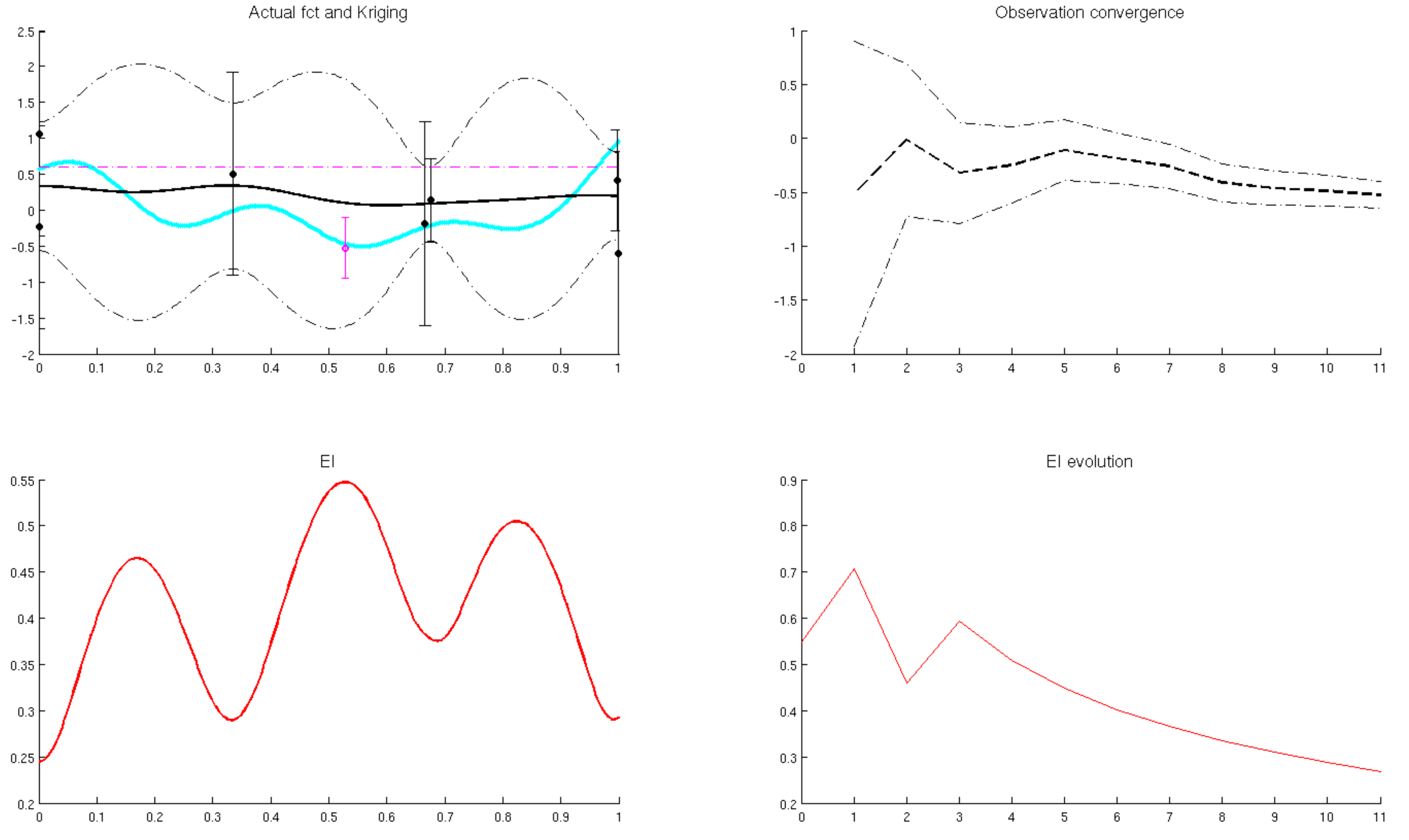
Iteration 2: 1 step used / 91 remaining



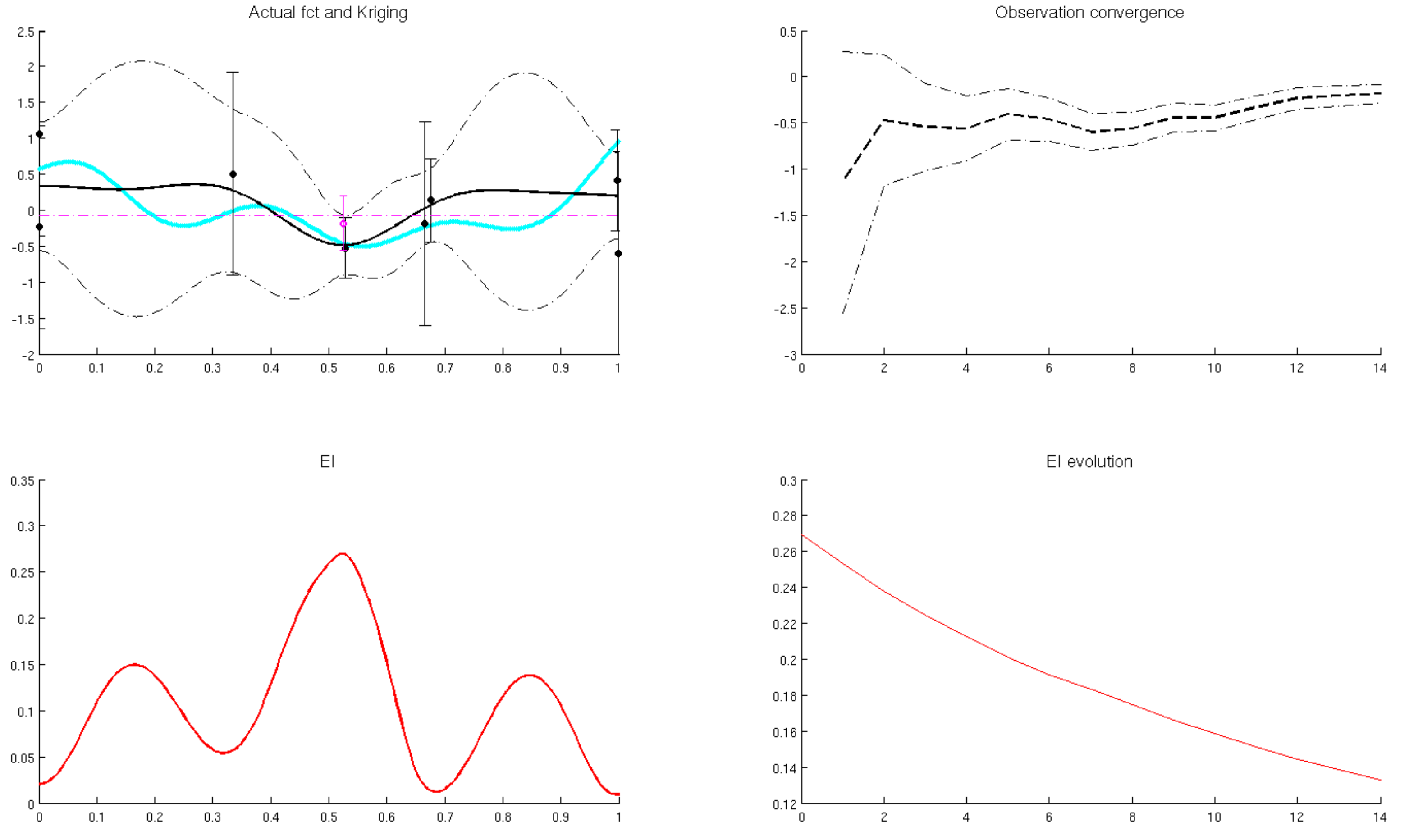
Iteration 3: 6 steps used / 85 remaining



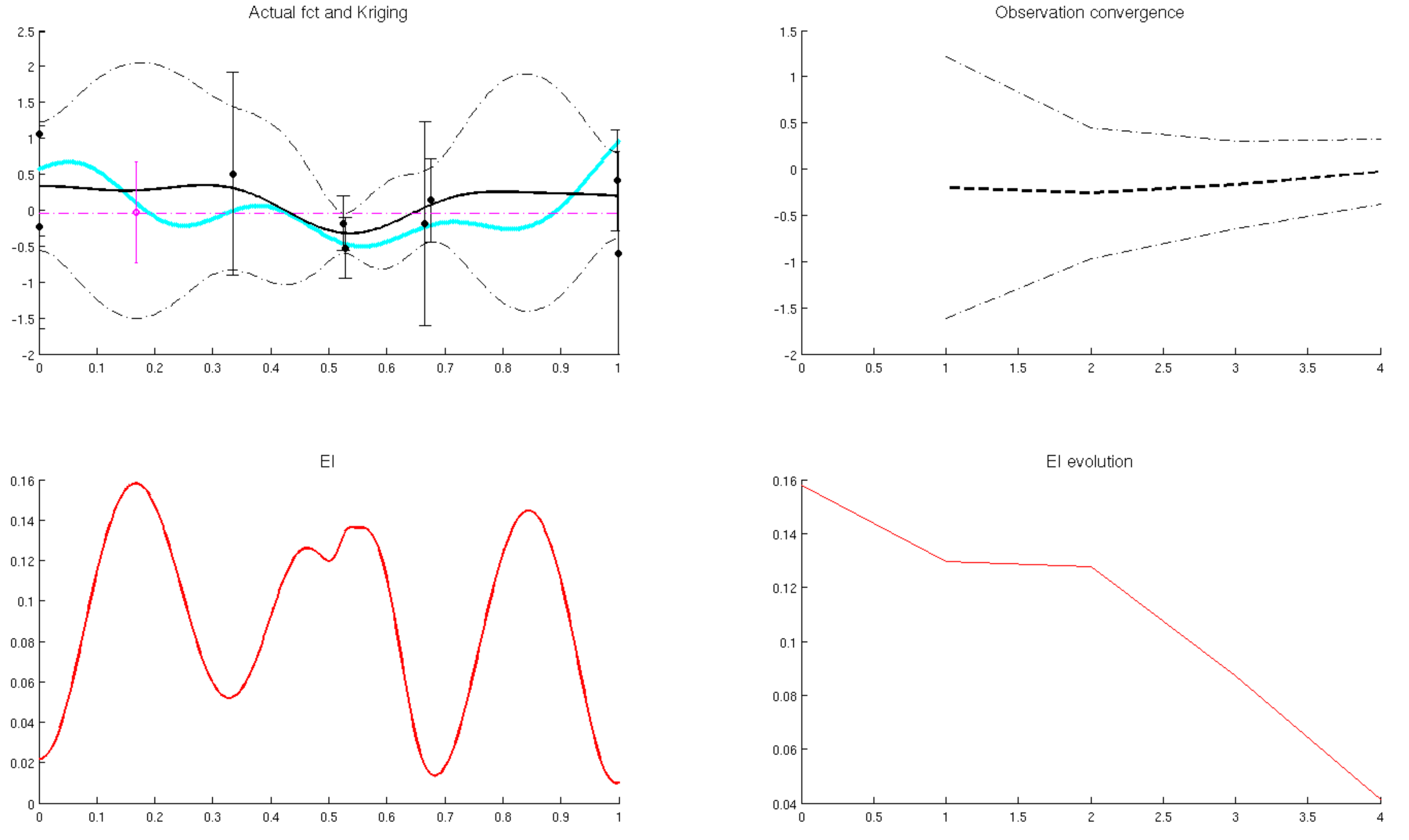
Iteration 4: 11 steps used / 74 remaining



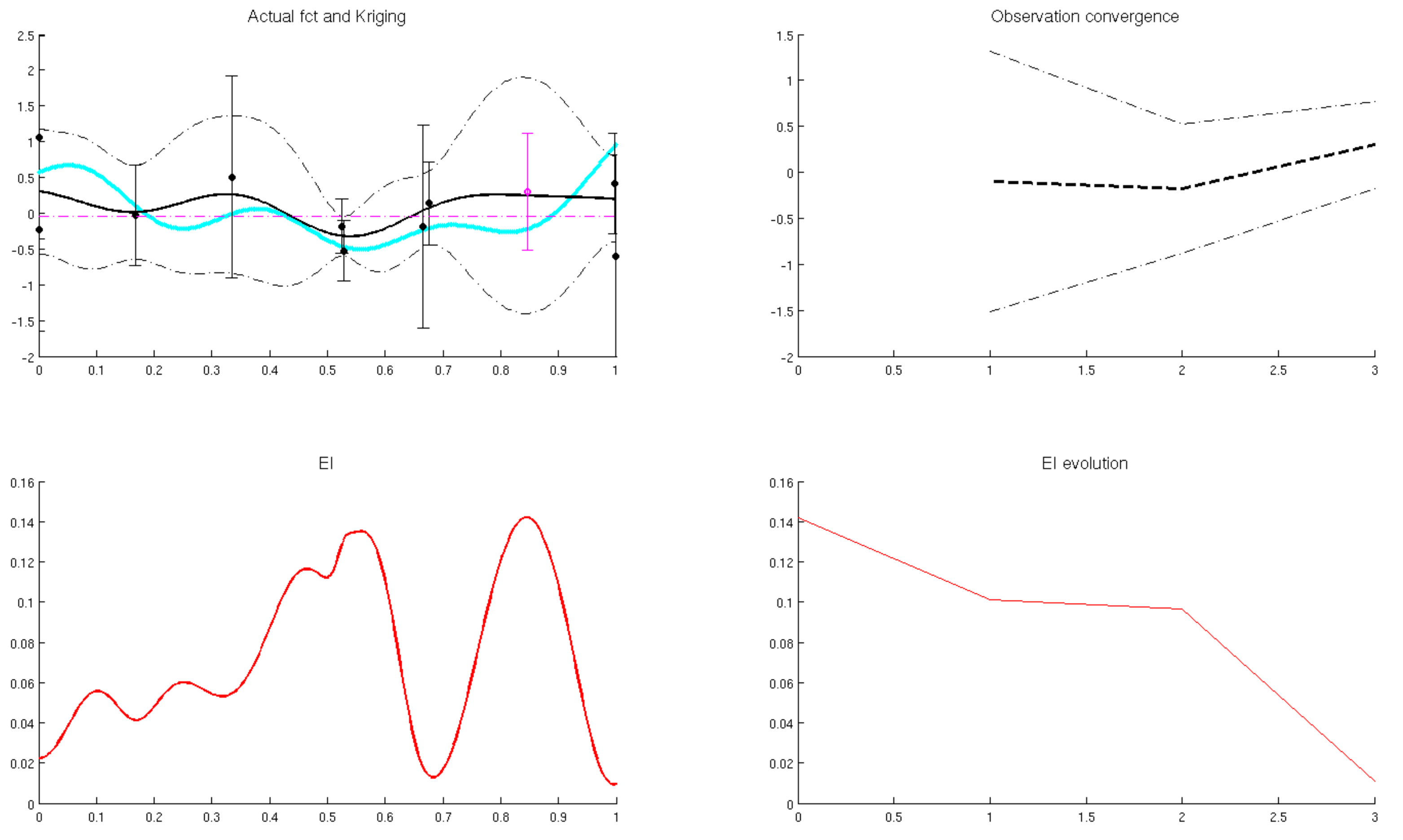
Iteration 5: 14 steps used / 60 remaining



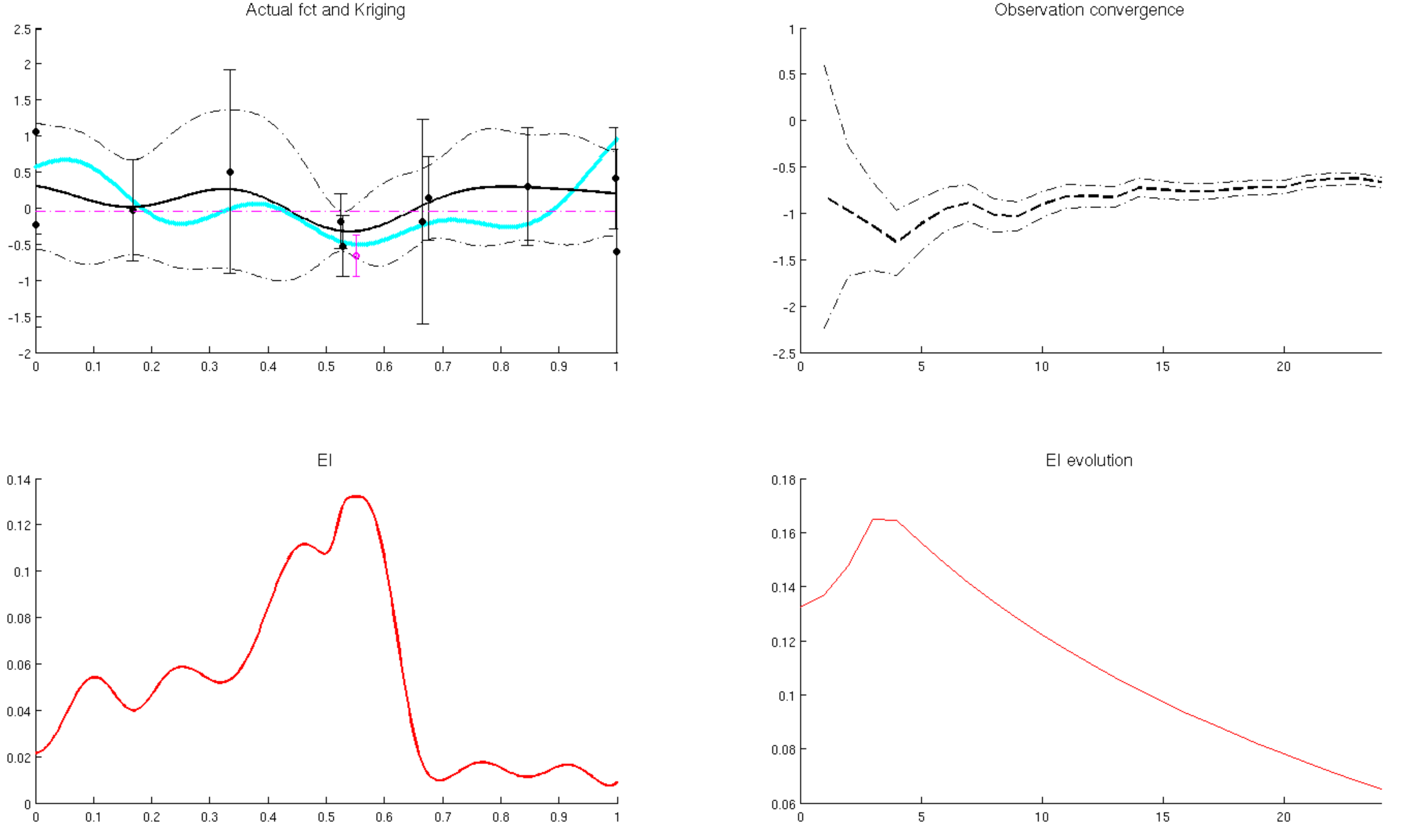
Iteration 6: 4 steps used / 56 remaining



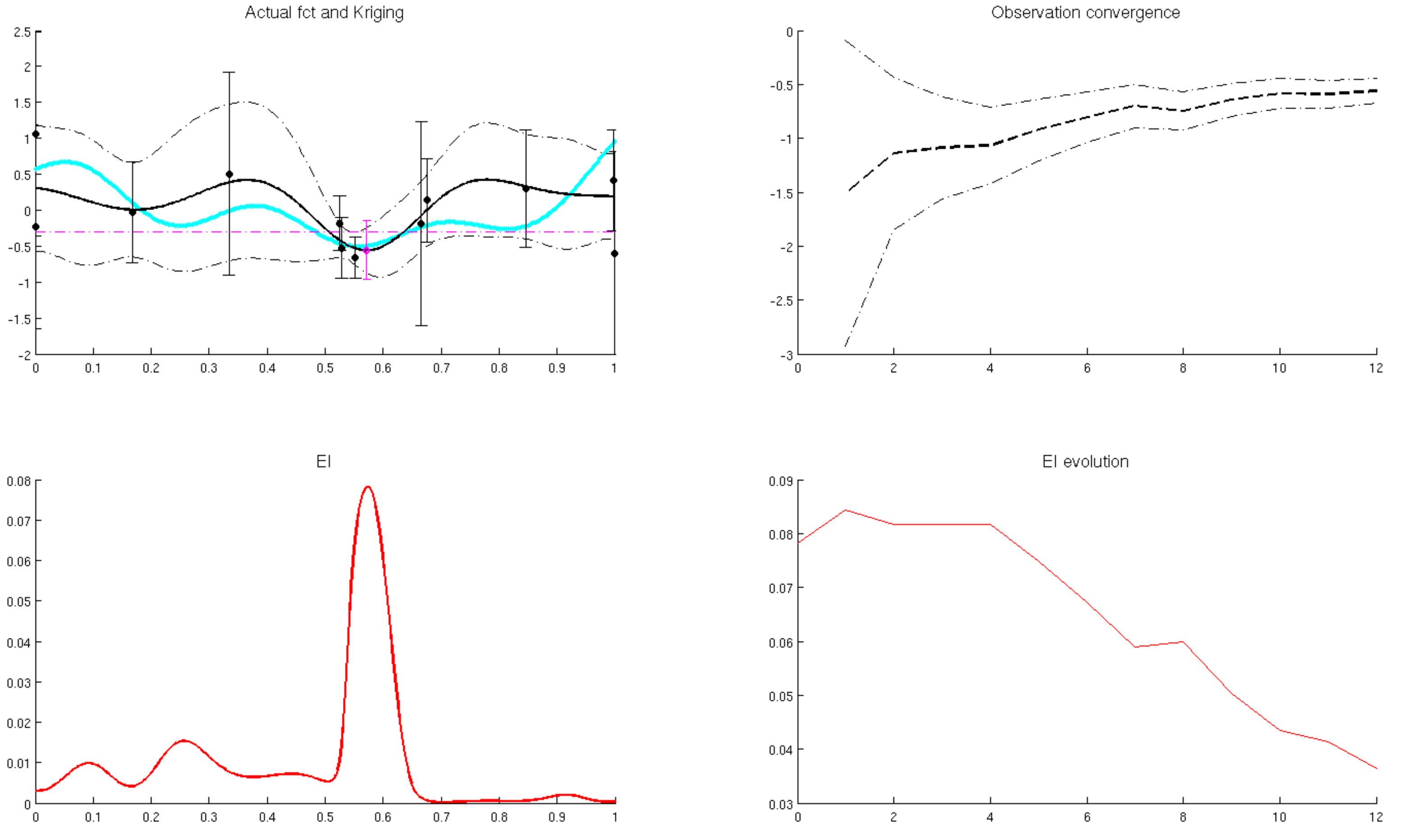
Iteration 7: 3 steps used / 53 remaining



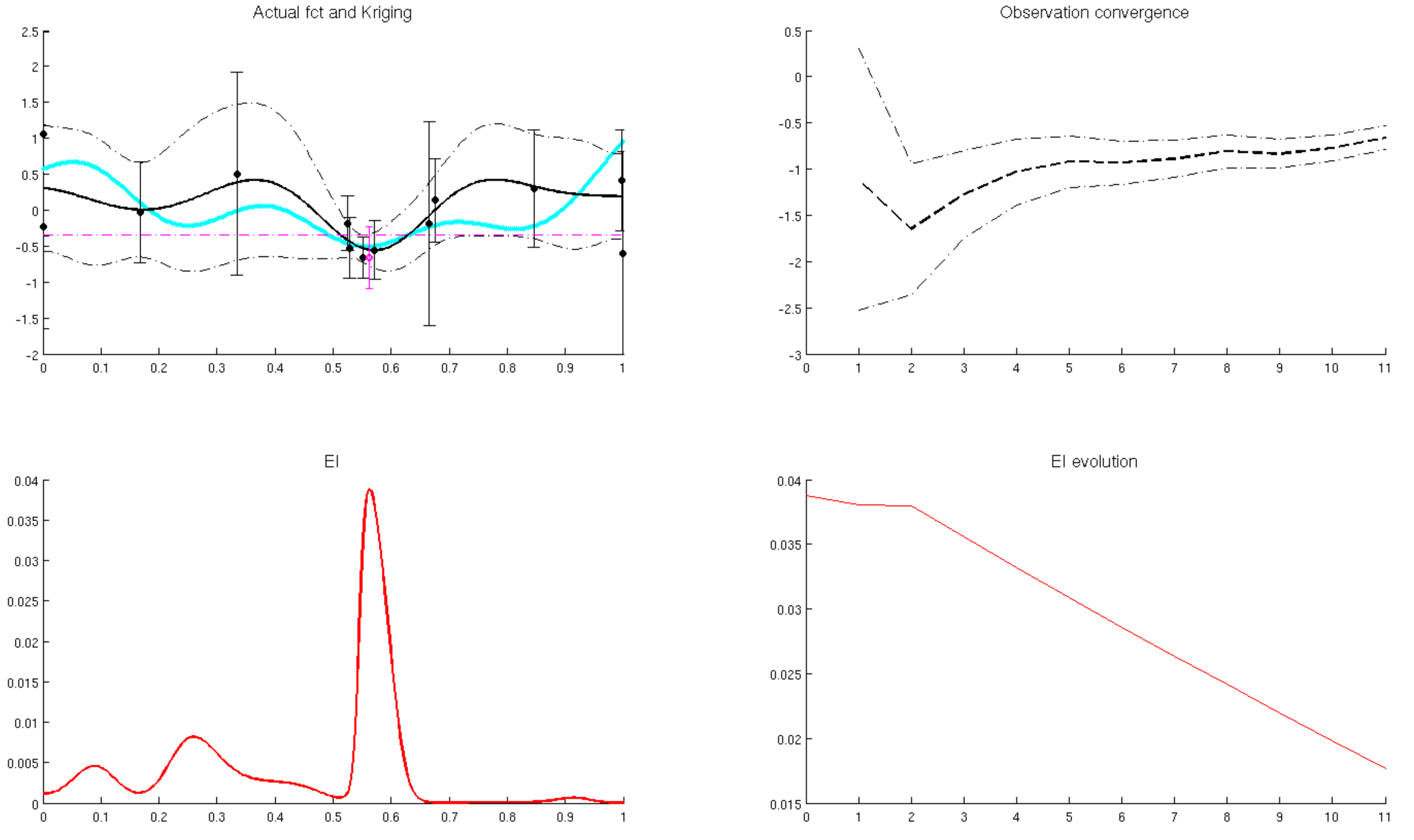
Iteration 8: 22 steps used / 29 remaining



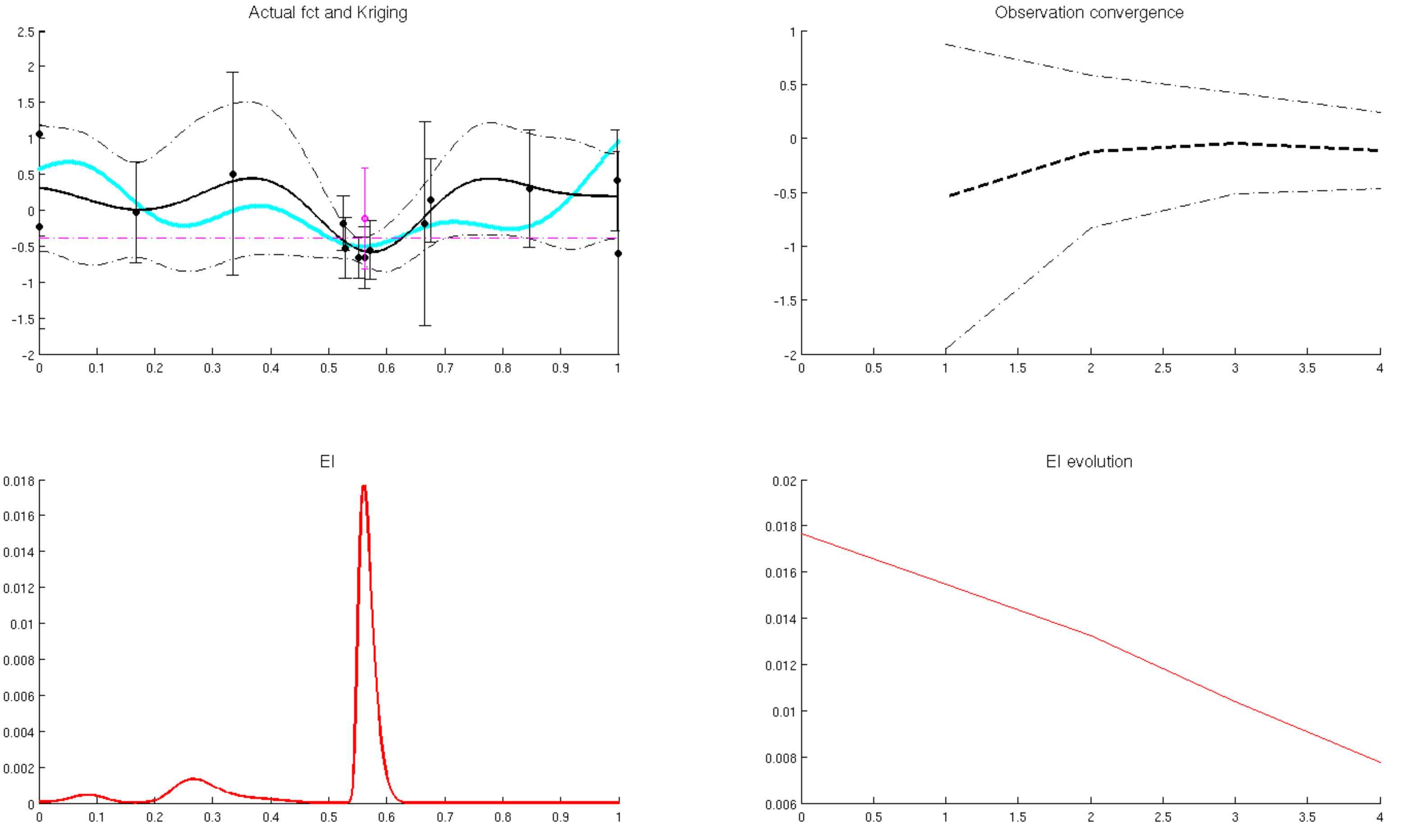
Iteration 9: 12 steps used / 17 remaining



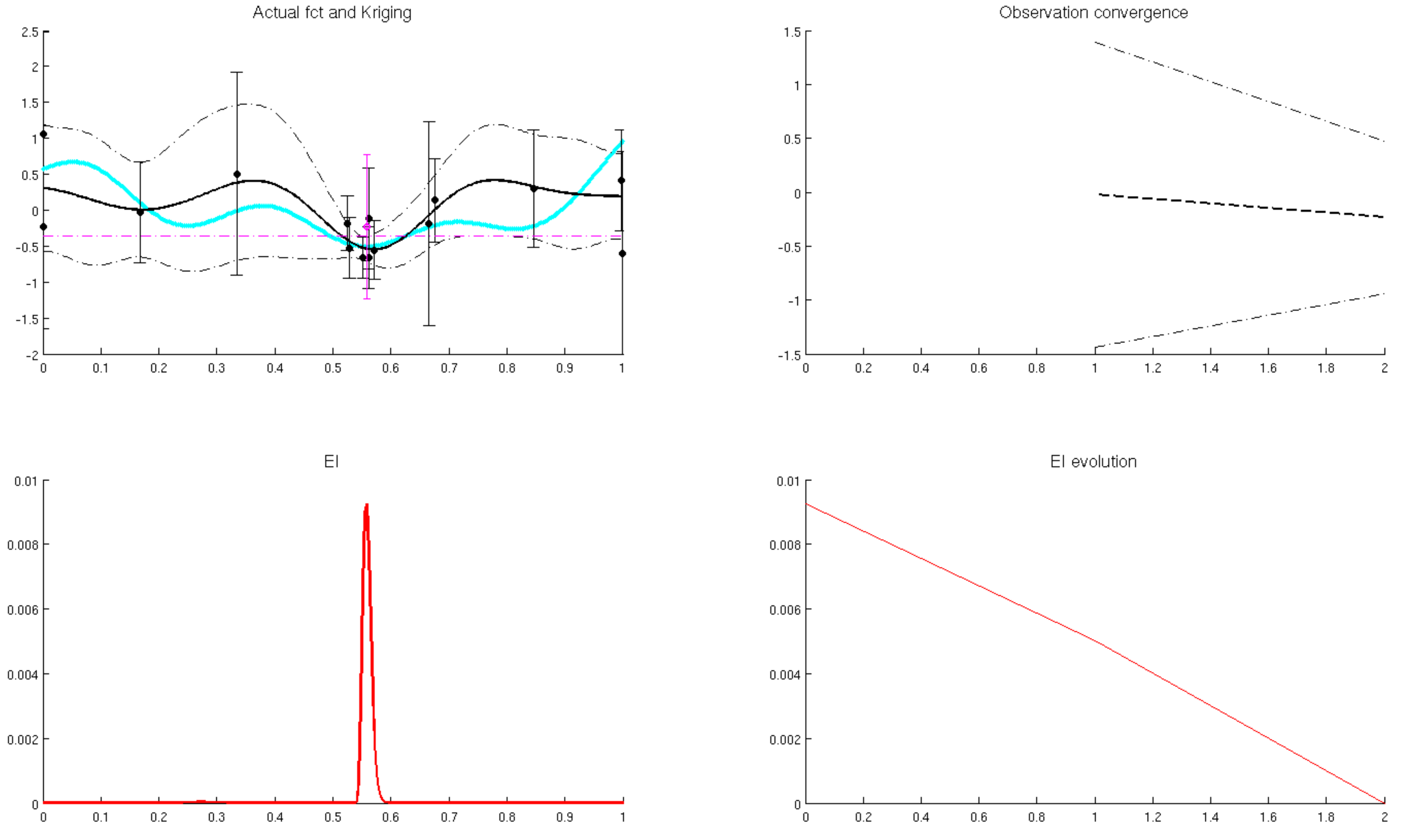
Iteration 10: 11 steps used / 6 remaining



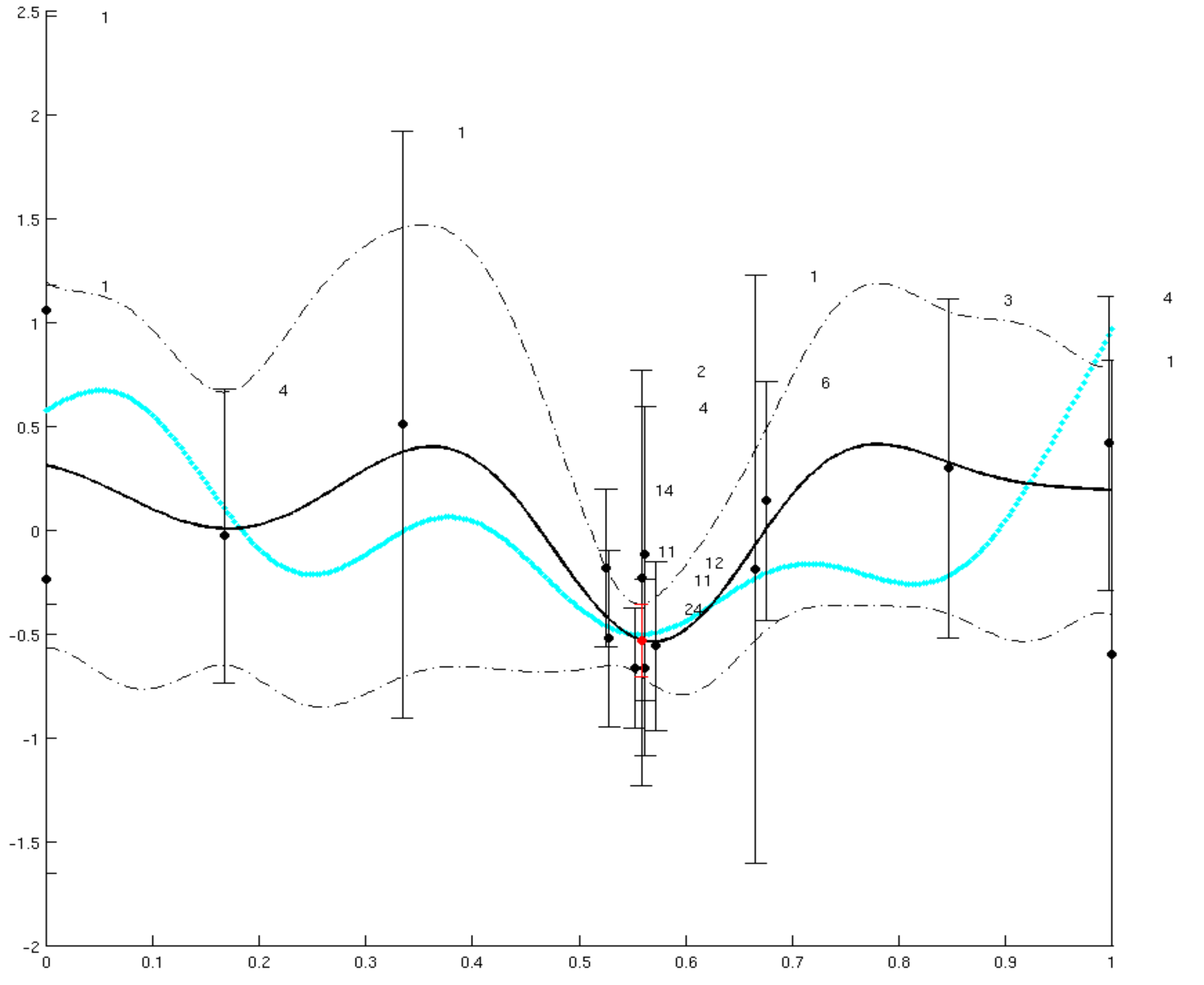
Iteration 11: 4 steps used / 2 remaining



Iteration 12: 2 steps used / 0 remaining



Iteration 12: final design



Concluding comments

- Main ideas:
  - Use of metamodel for final decision
  - Sampling criterion adapted to the decision
  - On-line resource allocation for improved efficiency
  - Finite resource strategy

- Requirements
  - Prior knowledge (or learning) of error variance
  - Response monitoring

Ongoing and future work

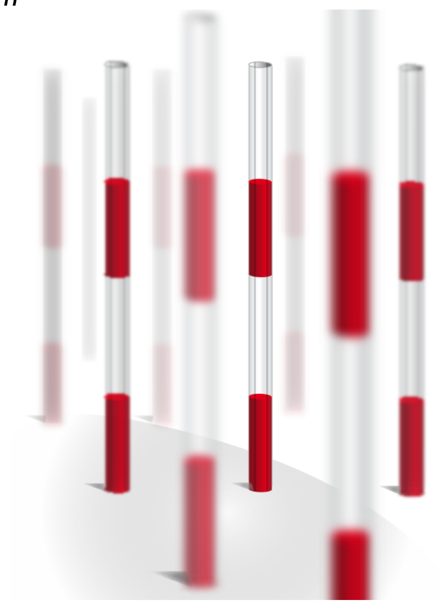
- Implementation in an R toolbox (*DiceOptim*)
- Comparison of all existing kriging-based methods
- Application to partial convergence

- Parallelization
- Integration of complementary heuristics (*racing*)
- Applications: CFD, Geosciences, …

Appendix 1: Application to a nuclear engineering problem

Nuclear criticality safety assessment

- Physical system: (interim) storage of fissile material (P_uO_2 powder in tubes)
- Safety measurement: neutron multiplication factor k_{eff}:
  - $k_{eff} > 1$: increasing neutrons productions
  - $k_{eff} = 1$: stage neutrons populations
  - $k_{eff} < 1$: required for storage

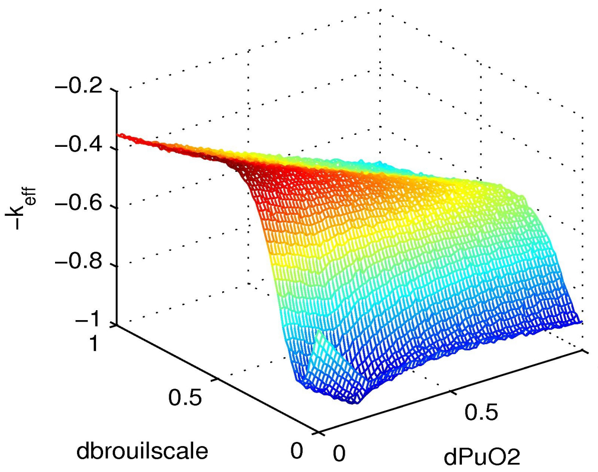


- Optimization problem: search for worst case of physical configurations
- k_{eff} computed using the MORET simulator:
  - based on MCMC
  - Sample size can be chosen by user
  - Known variance, inversely proportional to sample size

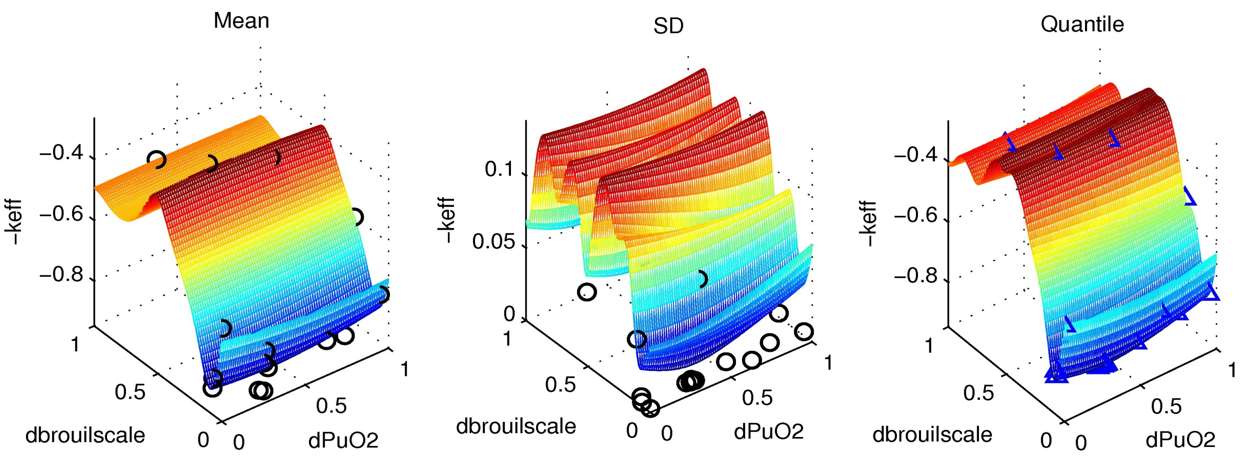
The benchmark problem

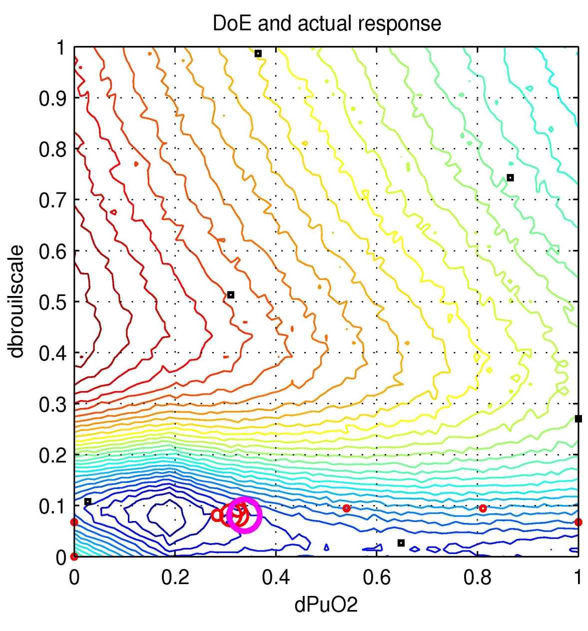
- Two parameters
  - Density of fissile powder
  - Density of water between storage tubes
- Computational time
  - One time step = 4000 particles = 30s
  - For one time step: $\tau = 5 \cdot 10^{-2}$
  - For 200 time steps: $\tau = 4 \cdot 10^{-3}$

Actual response:



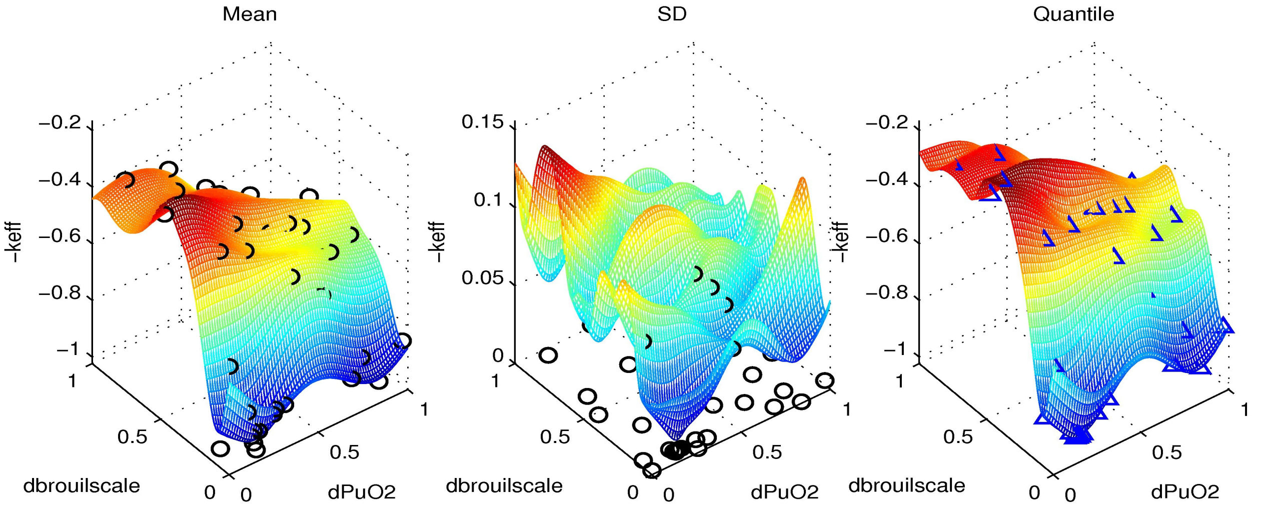
Results for a computational budget T=30

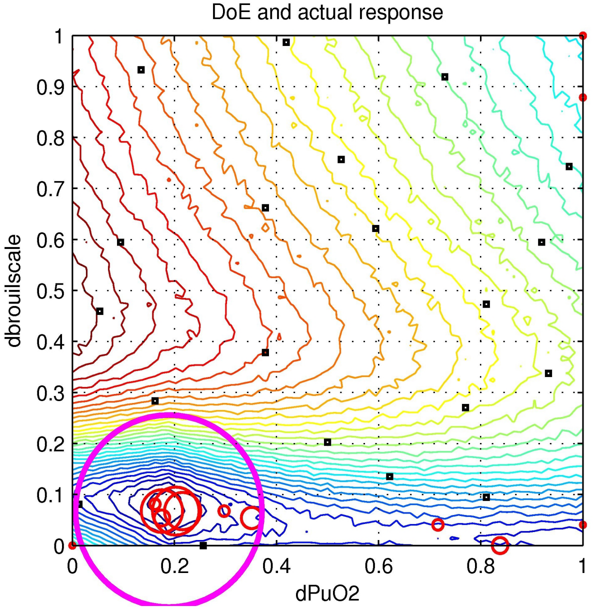




- Final design: 6 + 10 observations
- Poor kriging model
- Region of actual minimum identified
- Actual minimum missed

Results for a computational budget T=100





- Final design: 20 + 14 observations
- Locally accurate kriging model
- Actual minimum found
- 36% budget allocated to best design

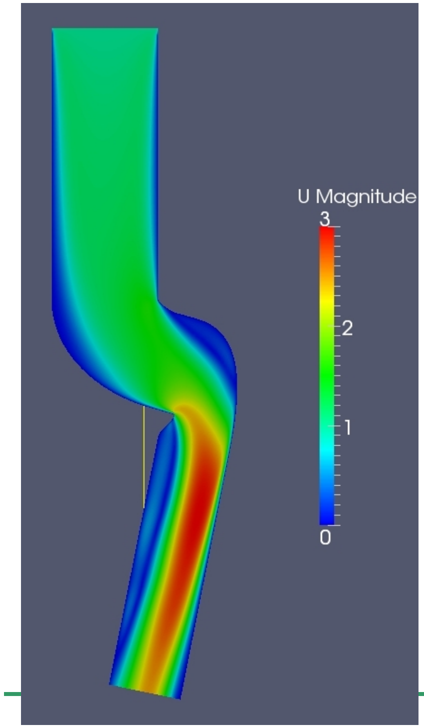
Appendix 2: partial convergence

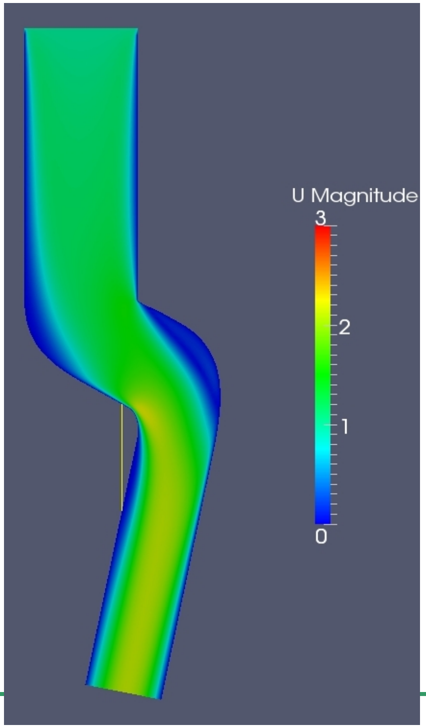
Partial convergence

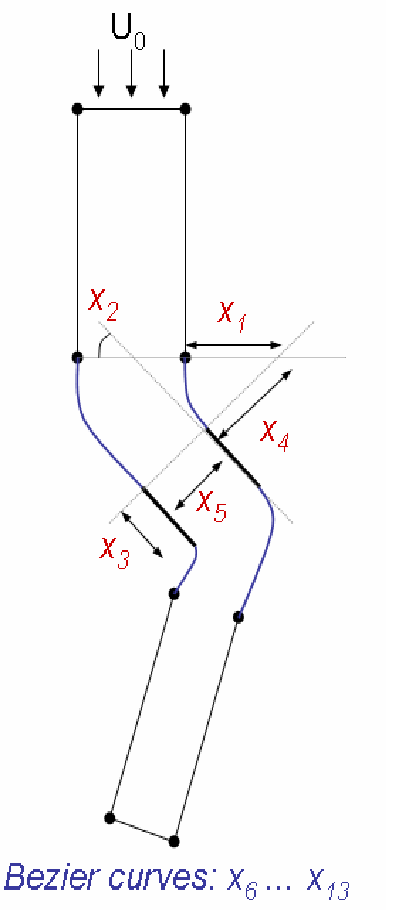
- Principle
  - CFD code: relies on an internal solver (Newton-Raphson)
  - Idea: stop calculations before convergence
  - Faster response, less accurate
- Potential assets
  - Almost non-instrusive
  - « Free » multi-fidelity
  - Single simulator
  - Continuum of fidelities available

Example: pipe flow problem

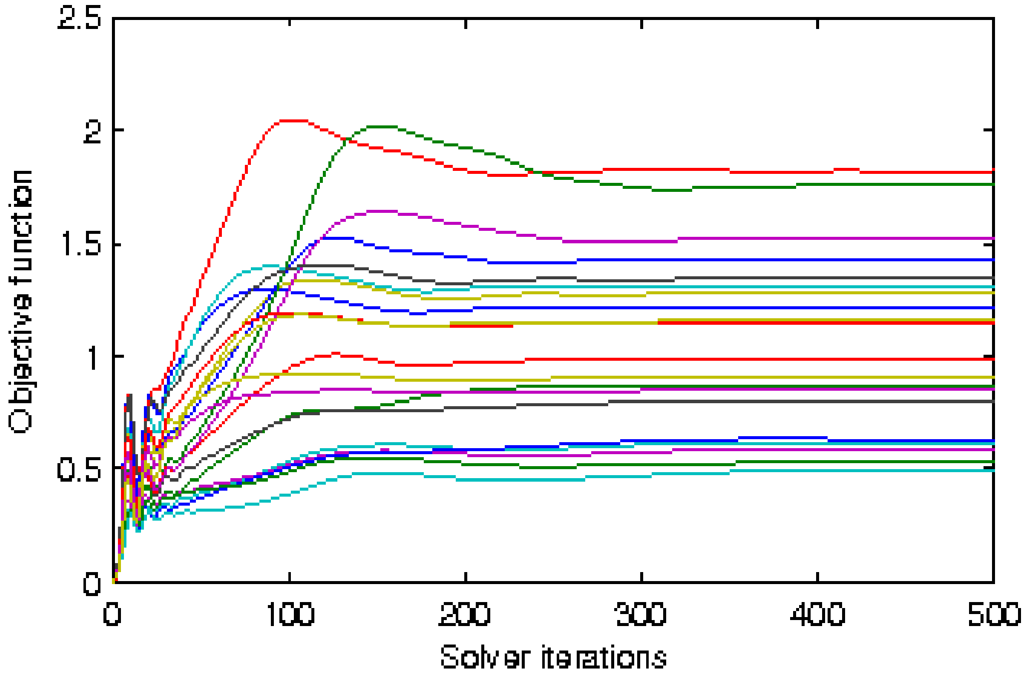
- 13 parameters
- OpenFOAM model: convergence in 500 steps
- Objective function: flow standard deviation







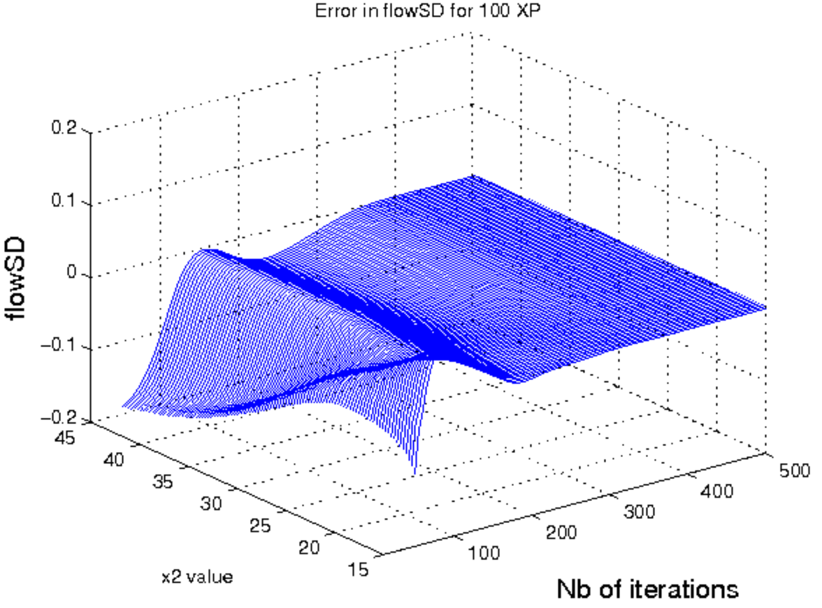
Objective function convergence at 20 designs



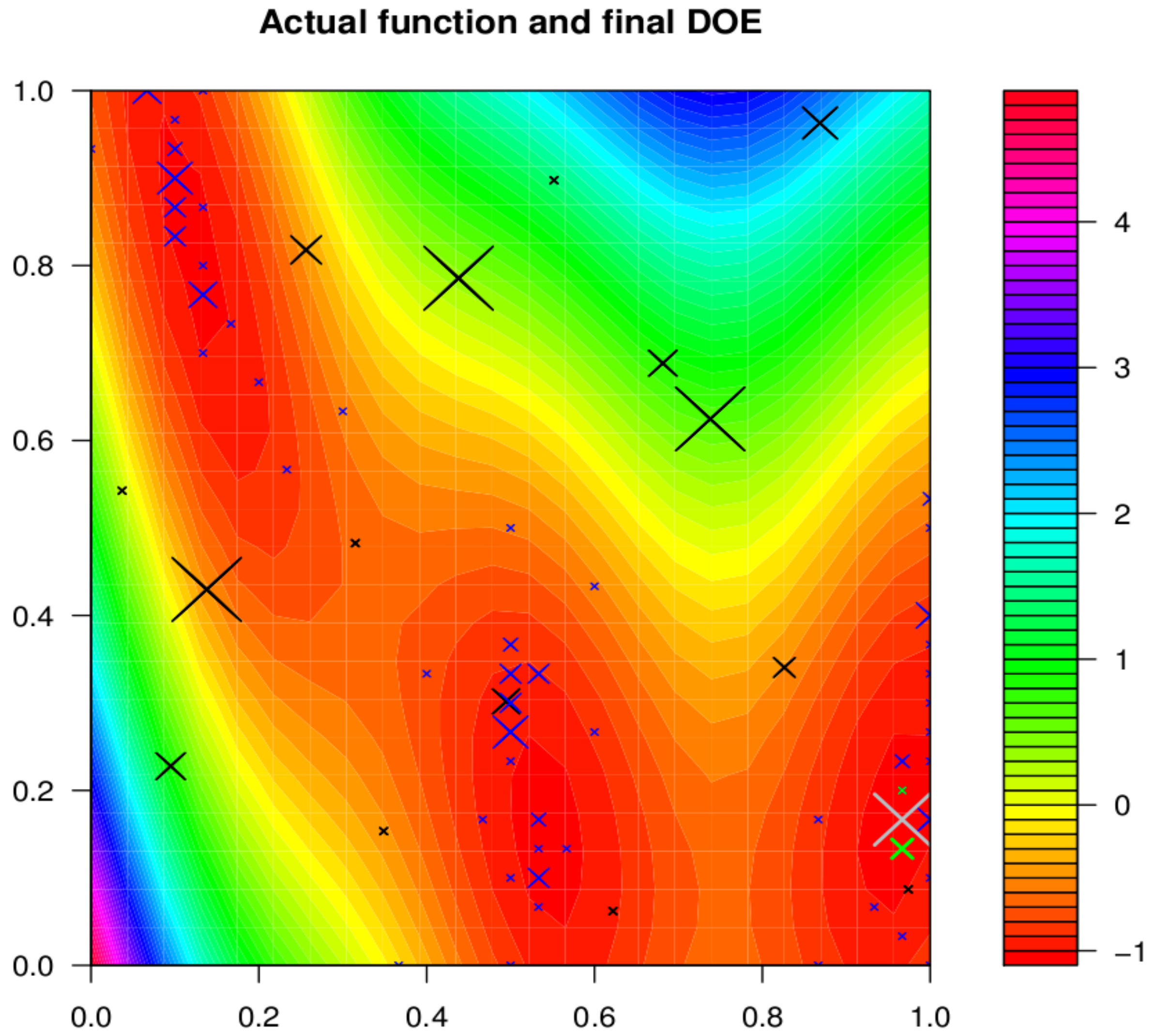
Useful information is obtained before full convergence!

A Space-time Gaussian process model

- Stochastic modeling of convergence error
- Intrinsic properties integrated in the covariance function
- Requires a learning stage
- Allows the use of EQI



Optimisation results on a toy problem



Appendix 3: adaptive designs

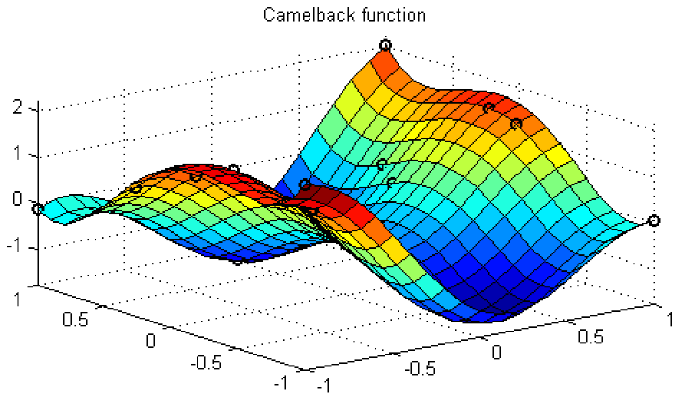
Adaptive design of experiments (1/3)

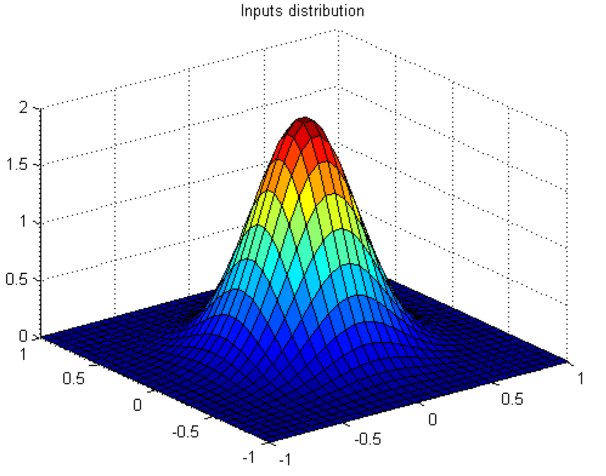
- Objective: accurate approximation when $f(x)=T$
- Constrained optimization / Probability of exceeding threshold / Inversion
- Proposition: new criterion for choosing sequentially experiments
  - Automated trade-off between uncertainty reduction and exploration of critical regions

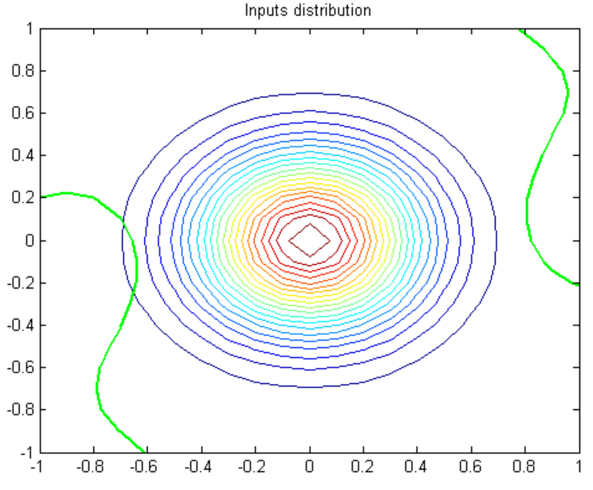
Adaptive design of experiments (2/3)

Fonction exacte

Distribution des paramètres d'entrée (Bigaussienne)



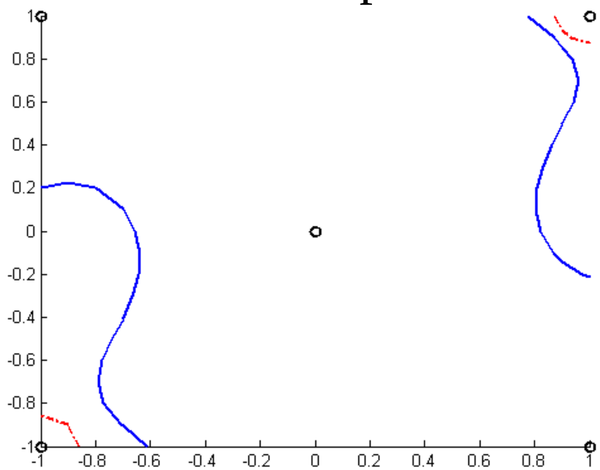




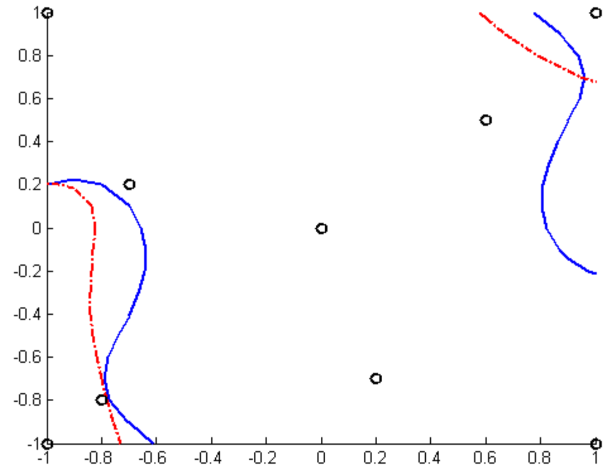
- Seuil : T = 1.3
- 2 régions de défaillance
- 'Budget' : 16 observations

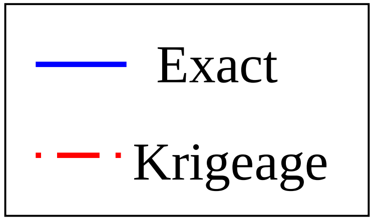
Adaptive design of experiments (3/3)

Initial: 5 points

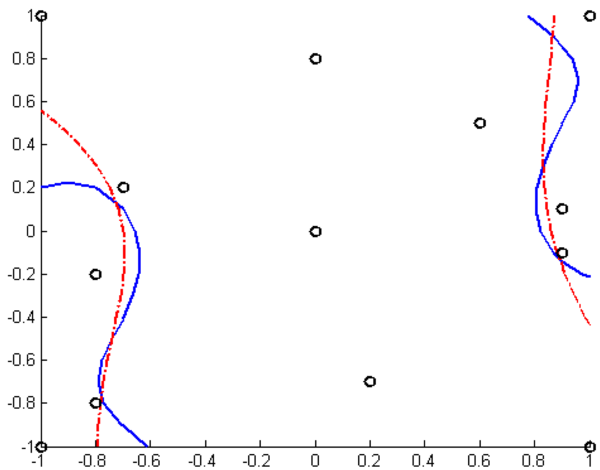


Après 4 itérations

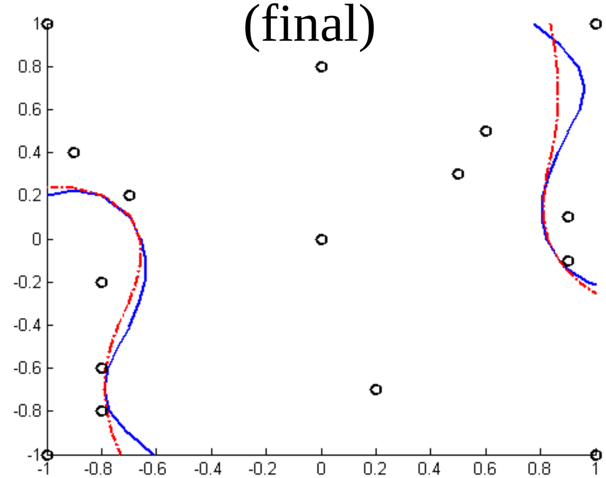




Après 8 itérations



Après 11 itérations (final)



Appendix 4: misc.

Future quantile distribution

- Kriging seen from step n+1 (observation added at $\mathbf{x}_{n+1}$):

$$M_{n+1}(\mathbf{x}) = \mathbf{c}_{n+1}(\mathbf{x})\mathbf{C}_{n+1}^{-1}\begin{bmatrix}\tilde{\mathbf{Y}}_n \\ \tilde{Y}(\mathbf{x}_{n+1})\end{bmatrix}$$

$$s_{n+1}^2(\mathbf{x}) = \sigma^2 - \mathbf{c}_{n+1}(\mathbf{x})\mathbf{C}_{n+1}^{-1}\mathbf{c}_{n+1}^T(\mathbf{x})$$

- At step n, everything is known except $\tilde{Y}(\mathbf{x}_{n+1})$
- We can provide a distribution using the kriging at step n:

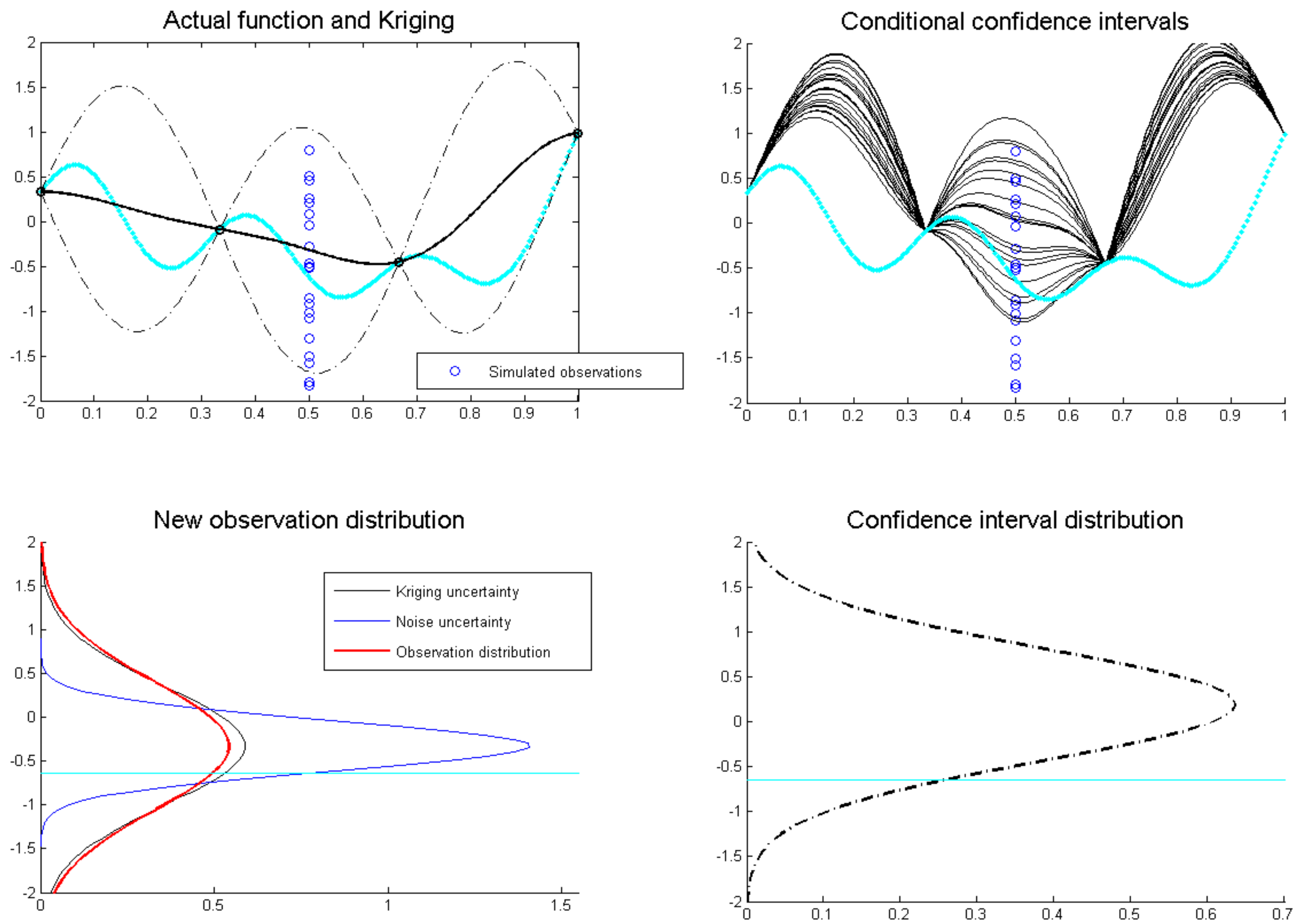
$$\tilde{Y}(\mathbf{x}_{n+1}) \sim N\left(m_n(\mathbf{x}_{n+1}), \underbrace{s_n^2(\mathbf{x}_{n+1})}_{\text{Kriging uncertainty}} + \underbrace{\tau^2}_{\text{noise}}\right)$$

- By linearity, future quantile follows a normal distribution:

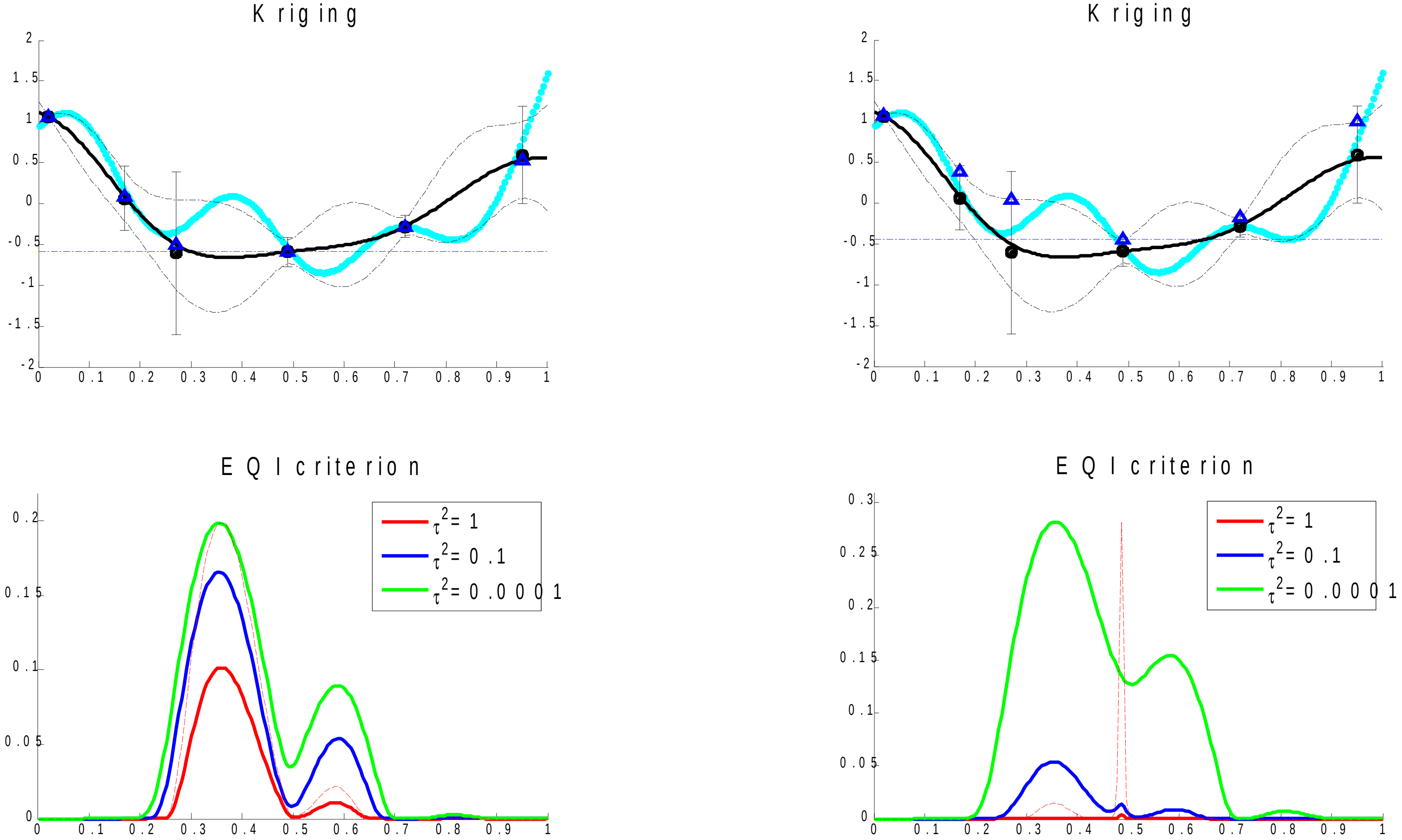
$$Q^{n+1}(\mathbf{x}) = M_{n+1}(\mathbf{x}) + \alpha s_{n+1}^2(\mathbf{x})$$

- Finally: $$EQI(\mathbf{x}) = E\left[\left(q_{min} - Q^{n+1}(\mathbf{x})\right)^+\right] \quad \text{with} \quad \mathbf{x}_{n+1} = \mathbf{x}$$

Illustration: kriging quantile distributions



Comparison of EQI for alpha=0 (left) and alpha=1.64 (right)



Algorithm overview

Initialization
- Define computational budget *T*
- Generate initial DoE
- Build metamodel

While *T* > 0

Select experiment
Choose new design that maximizes $EQI_{init} = EQI[\tau^2(T)]$

On-line allocation
While $EQI[\tau^2(T)] > EQI_{init}/2$
- Add one time step, update observation
- Update metamodel
- Update $T = T - t_{step}$
- Update EI

Choose final design based on Kriging quantile

EQI update

- Equivalent observation:
  - Two observations: $\widetilde{y}_{i,1}$ and $\widetilde{y}_{i,2}$ $\quad \tau_{i,1}^2$ and $\tau_{i,2}^2$
  - Equivalent to: $\widetilde{y}_{i,eq} = \left(\tau_{i,1}^2 + \tau_{i,2}^2\right)\left(\tau_{i,1}^{-2}\widetilde{y}_{i,1} + \tau_{i,2}^{-2}\widetilde{y}_{i,2}\right)$

$$\frac{1}{\tau_{i,eq}^2} := \frac{1}{\tau_{i,1}^2} + \frac{1}{\tau_{i,2}^2} \Longrightarrow \tau_{i,eq}^2 = \frac{\tau_{i,1}^2\tau_{i,2}^2}{\tau_{i,1}^2 + \tau_{i,2}^2}$$

- Noise of equivalent observation for going from step *j* to b_i:

$$\tau_i^2[j \to b_i] := \frac{\tau_i^2[j]\tau_i^2[b_i]}{\tau_i^2[j] - \tau_i^2[b_i]} = \frac{\tau^2(j \times t_e)\tau^2(T_i)}{\tau^2(j \times t_e) - \tau^2(T_i)} =: \tau^2(j \times t_e \to T_i)$$