Bayesian Optimization via Simulation with Pairwise Sampling and Correlated Prior Beliefs

Jing Xie
Peter I. Frazier
Stephen E. Chick

Operations Research & Information Engineering, Cornell University
Technology & Operations Management Area, INSEAD

Oct. 8, 2013
2013 INFORMS Annual Meeting, Minneapolis

Discrete Optimization via Simulation (DOvS)

- A is some discrete set.
- Objective function $f : A \mapsto \mathbb{R}$.
- Our goal is to solve

$$\max_{x \in A} f(x).$$

Discrete Optimization via Simulation (DOvS)

- A is some discrete set.
- Objective function $f : A \mapsto \mathbb{R}$.
- Our goal is to solve

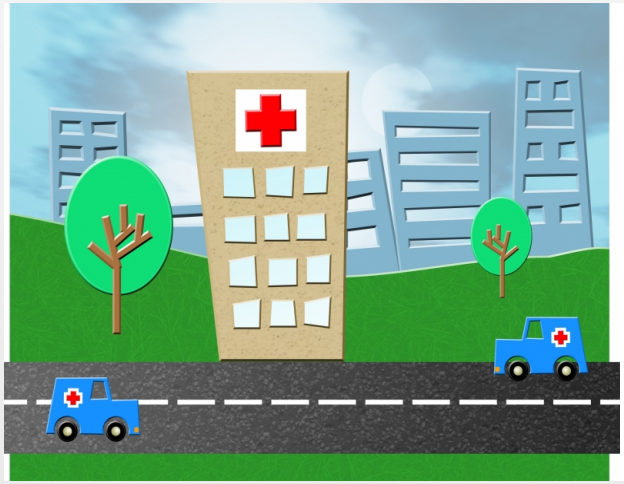
$$\max_{x \in A} f(x).$$

- We cannot evaluate $f(x)$ directly.
- We have a stochastic simulator that can evaluate $f(x)$ with noise.
- It gives us $g(x, \omega) = f(x) + \epsilon(x, \omega)$, where $\mathbb{E}[g(x, \omega)] = f(x)$.

Example Applications using Discrete-Event Simulations

A may correspond to a $\mathbb{Z}^d$ lattice:

- staffing levels in a hospital
- inventory policies for a supply chain
- admission controls in a call center



A can also correspond to to a set of combinatorial structures:

- configurations of an assembly line

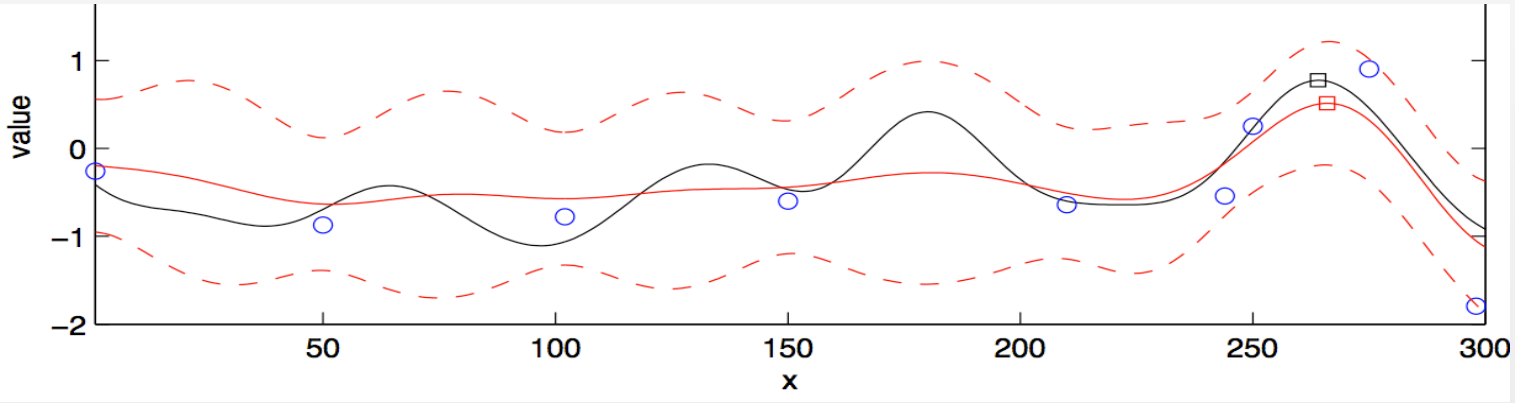
We Employ a Bayesian Value-of-Information Approach

We Employ a Bayesian Value-of-Information Approach

- Given the function evaluations obtained, use Bayesian statistics to get:
  - estimates of $f(x)$ over set A.
  - uncertainties in these estimates.

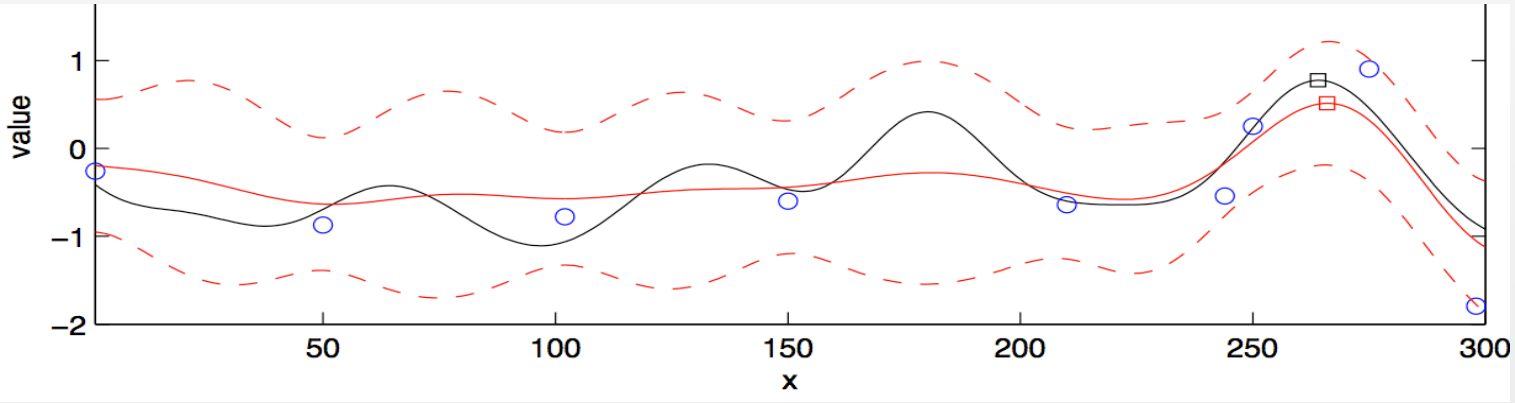
We Employ a Bayesian Value-of-Information Approach

- Given the function evaluations obtained, use Bayesian statistics to get:
  - estimates of $f(x)$ over set A.
  - uncertainties in these estimates.
- A common method: (discrete) Gaussian Process (GP) regression:



We Employ a Bayesian Value-of-Information Approach

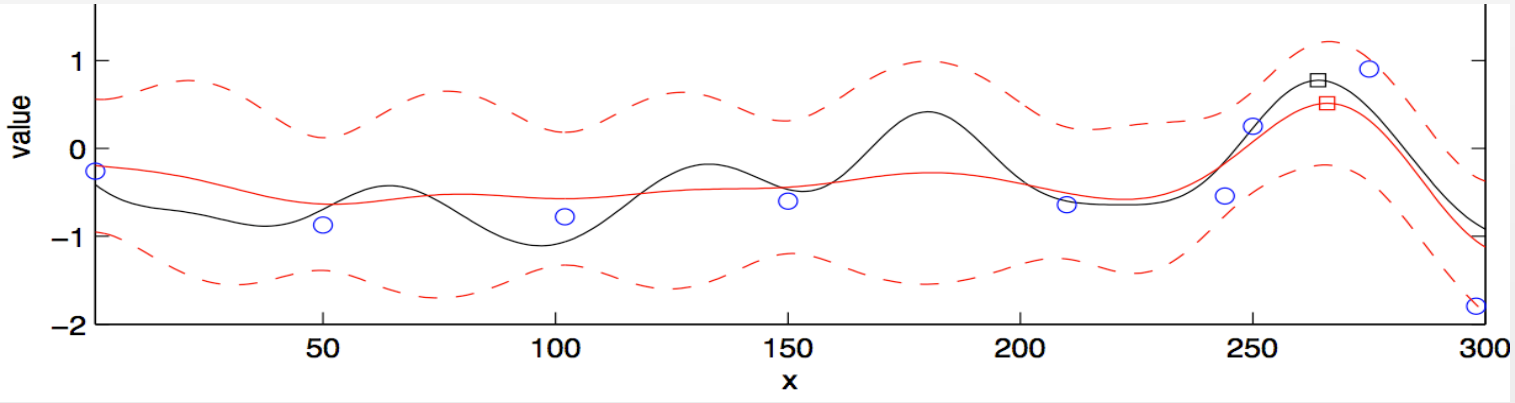
- Given the function evaluations obtained, use Bayesian statistics to get:
  - estimates of $f(x)$ over set A.
  - uncertainties in these estimates.
- A common method: (discrete) Gaussian Process (GP) regression:



- Use these estimates and uncertainties to
  - quantify the contribution of possible future evaluations.
  - decide where to evaluate next.

We Employ a Bayesian Value-of-Information Approach

- Given the function evaluations obtained, use Bayesian statistics to get:
  - estimates of $f(x)$ over set A.
  - uncertainties in these estimates.
- A common method: (discrete) Gaussian Process (GP) regression:



- Use these estimates and uncertainties to
  - quantify the contribution of possible future evaluations.
  - decide where to evaluate next.

VOI: *Information is valued according to the expected improvement it produces in some decision to be made later.* (Raiffa & Schlaifer 1961)

Our Problem Settings

- We consider DOvS over a finite set of alternatives $1, 2, \ldots, k$.

Our Problem Settings

- We consider DOvS over a finite set of alternatives $1, 2, \ldots, k$.
- Our simulator allows **correlated** output through common random numbers (CRN).

Our Problem Settings

- We consider DOvS over a finite set of alternatives $1, 2, \ldots, k$.
- Our simulator allows **correlated** output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector $\theta = (\theta_1, \ldots, \theta_k)$,
  - covariance matrix Λ (assumed known, but can be relaxed).

Our Problem Settings

- We consider DOvS over a finite set of alternatives $1, 2, \ldots, k$.
- Our simulator allows **correlated** output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector $\theta = (\theta_1, \ldots, \theta_k)$,
  - covariance matrix Λ (assumed known, but can be relaxed).
- GOAL: find $x^* = \text{argmax}_x\{\theta_x\}$.

Our Problem Settings

- We consider DOvS over a finite set of alternatives $1, 2, \ldots, k$.
- Our simulator allows **correlated** output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector $\theta = (\theta_1, \ldots, \theta_k)$,
  - covariance matrix Λ (assumed known, but can be relaxed).
- GOAL: find $x^* = \text{argmax}_x\{\theta_x\}$.
- We believe that the means of the alternatives are **correlated**, that is, similar alternatives often have similar performance.

Our Problem Settings

- We consider DOvS over a finite set of alternatives $1, 2, \ldots, k$.
- Our simulator allows **correlated** output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector $\theta = (\theta_1, \ldots, \theta_k)$,
  - covariance matrix Λ (assumed known, but can be relaxed).
- GOAL: find $x^* = \text{argmax}_x\{\theta_x\}$.
- We believe that the means of the alternatives are **correlated**, that is, similar alternatives often have similar performance.

  We model this belief by a multivariate normal prior on θ, $\theta \sim \mathcal{N}(\mu_0, \Sigma_0)$.

Our Problem Settings

- We consider DOvS over a finite set of alternatives $1, 2, \ldots, k$.
- Our simulator allows **correlated** output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector $\theta = (\theta_1, \ldots, \theta_k)$,
  - covariance matrix Λ (assumed known, but can be relaxed).
- GOAL: find $x^* = \text{argmax}_x\{\theta_x\}$.
- We believe that the means of the alternatives are **correlated**, that is, similar alternatives often have similar performance.

  We model this belief by a multivariate normal prior on θ, $\theta \sim \mathcal{N}(\mu_0, \Sigma_0)$.
- We calculate VOI to decide which subset of alternatives to sample.

Related Studies

- Jones, Schonlau and Welch 1998
- Chick and Inoue 2001
- Ginsbourger,Riche and Carraro2007
- Brochu, Cora and Freitas 2009
- Frazier, Powell and Dayanik 2009
- Negoescu, Frazier and Powell 2011
- Scott, Frazier, and Powell 2011
- Clark and Yang 1986
- Yang and Nelson 1991
- Nelson and Matejcik 1995
- Nakayama 2000
- Kim 2005
- Fu, Hu, Chen, and Xiong 2007
- Goldsman,Marshall,Kim and Nelson2000

Related Studies

- Jones, Schonlau and Welch 1998
- Chick and Inoue 2001
- Ginsbourger,Riche and Carraro2007
- Brochu, Cora and Freitas 2009
- Frazier, Powell and Dayanik 2009
- Negoescu, Frazier and Powell 2011
- Scott, Frazier, and Powell 2011
- Clark and Yang 1986
- Yang and Nelson 1991
- Nelson and Matejcik 1995
- Nakayama 2000
- Kim 2005
- Fu, Hu, Chen, and Xiong 2007
- Goldsman,Marshall,Kim and Nelson2000

Novelty of Our Work:

- Correlation types: BOTH

Related Studies

- Jones, Schonlau and Welch 1998
- Chick and Inoue 2001
- Ginsbourger,Riche and Carraro2007
- Brochu, Cora and Freitas 2009
- Frazier, Powell and Dayanik 2009
- Negoescu, Frazier and Powell 2011
- Scott, Frazier, and Powell 2011
- Clark and Yang 1986
- Yang and Nelson 1991
- Nelson and Matejcik 1995
- Nakayama 2000
- Kim 2005
- Fu, Hu, Chen, and Xiong 2007
- Goldsman,Marshall,Kim and Nelson2000

Novelty of Our Work:

- Correlation types: BOTH
- Sampling plan: PAIRS

Related Studies

- Jones, Schonlau and Welch 1998
- Chick and Inoue 2001
- Ginsbourger,Riche and Carraro2007
- Brochu, Cora and Freitas 2009
- Frazier, Powell and Dayanik 2009
- Negoescu, Frazier and Powell 2011
- Scott, Frazier, and Powell 2011
- Clark and Yang 1986
- Yang and Nelson 1991
- Nelson and Matejcik 1995
- Nakayama 2000
- Kim 2005
- Fu, Hu, Chen, and Xiong 2007
- Goldsman,Marshall,Kim and Nelson2000

Novelty of Our Work:

- Correlation types: BOTH
- Sampling plan: PAIRS
- Problem Scale: LARGE

Generic Sampling Algorithm

Choose one or several alternatives to evaluate at time 1.

At each time $n = 1, 2, \ldots$ (while the stopping criterion is not met):

- Calculate the Bayesian posterior distribution on θ, which is $\mathcal{N}(\mu_n, \Sigma_n)$.

Generic Sampling Algorithm

Choose one or several alternatives to evaluate at time 1.

At each time $n = 1, 2, \ldots$ (while the stopping criterion is not met):

- Calculate the Bayesian posterior distribution on θ, which is $\mathcal{N}(\mu_n, \Sigma_n)$.
- Use this posterior to calculate the potential value of sampling certain subsets of alternatives using CRN at time $n + 1$.
- Choose a subset of alternatives X_{n+1} to sample at time $n + 1$ using CRN.

Generic Sampling Algorithm

Choose one or several alternatives to evaluate at time 1.

At each time $n = 1, 2, \ldots$ (while the stopping criterion is not met):

- Calculate the Bayesian posterior distribution on θ, which is $\mathcal{N}(\mu_n, \Sigma_n)$.
- Use this posterior to calculate the potential value of sampling certain subsets of alternatives using CRN at time $n + 1$.
- Choose a subset of alternatives X_{n+1} to sample at time $n + 1$ using CRN.

The stopping criterion is often "stop after N samples", or an adaptive rule.

Generic Sampling Algorithm

Choose one or several alternatives to evaluate at time 1.

At each time $n = 1, 2, \ldots$ (while the stopping criterion is not met):

- Calculate the Bayesian posterior distribution on θ, which is $\mathcal{N}(\mu_n, \Sigma_n)$.
- Use this posterior to calculate the potential value of sampling certain subsets of alternatives using CRN at time $n + 1$.
- Choose a subset of alternatives X_{n+1} to sample at time $n + 1$ using CRN.

The stopping criterion is often "stop after N samples", or an adaptive rule.

Upon stopping, report the alternative with the best estimated value as the "implementation decision".

Value of Information

At time n, the VOI of evaluating subset X at time $n+1$ is

$$V_n(X) = \mathbb{E}_n \left[\max_x \mu_{n+1,x} \;\middle|\; X_{n+1} = X \right] - \max_x \mu_{n,x}.$$

Value of Information

At time n, the VOI of evaluating subset X at time $n+1$ is

$$V_n(X) = \mathbb{E}_n \left[\max_x \mu_{n+1,x} \;\middle|\; X_{n+1} = X \right] - \max_x \mu_{n,x}.$$

- $\mu_{n,x}$ is the expected value of θ_x given what we know at time n.

Value of Information

At time n, the VOI of evaluating subset X at time $n+1$ is

$$V_n(X) = \mathbb{E}_n \left[\max_x \mu_{n+1,x} \;\middle|\; X_{n+1} = X \right] - \max_x \mu_{n,x}.$$

- $\mu_{n,x}$ is the expected value of θ_x given what we know at time n.
- $\max_x \mu_{n,x}$ is the best we can do given what we know at time n.

Value of Information

At time n, the VOI of evaluating subset X at time $n+1$ is

$$V_n(X) = \mathbb{E}_n \left[\max_x \mu_{n+1,x} \,\middle|\, X_{n+1} = X \right] - \max_x \mu_{n,x}.$$

- $\mu_{n,x}$ is the expected value of θ_x given what we know at time n.
- $\max_x \mu_{n,x}$ is the best we can do given what we know at time n.
- $\max_x \mu_{n+1,x}$ is the best we will be able to do given what we know at time n and what we learn from the measurements at time $n+1$.

Value of Information

At time n, the VOI of evaluating subset X at time $n+1$ is

$$V_n(X) = \mathbb{E}_n \left[\max_x \mu_{n+1,x} \,\middle|\, X_{n+1} = X \right] - \max_x \mu_{n,x}.$$

- $\mu_{n,x}$ is the expected value of θ_x given what we know at time n.
- $\max_x \mu_{n,x}$ is the best we can do given what we know at time n.
- $\max_x \mu_{n+1,x}$ is the best we will be able to do given what we know at time n and what we learn from the measurements at time $n+1$.

$V_n(X)$ is the expected improvement that evaluating X can produce in the best estimated overall value from time n to time $n+1$.

Value of Information

At time n, the VOI of evaluating subset X at time $n+1$ is

$$V_n(X) = \mathbb{E}_n\left[\max_x \mu_{n+1,x} \,\middle|\, X_{n+1} = X\right] - \max_x \mu_{n,x}.$$

- $\mu_{n,x}$ is the expected value of θ_x given what we know at time n.
- $\max_x \mu_{n,x}$ is the best we can do given what we know at time n.
- $\max_x \mu_{n+1,x}$ is the best we will be able to do given what we know at time n and what we learn from the measurements at time $n+1$.

$V_n(X)$ is the expected improvement that evaluating X can produce in the best estimated overall value from time n to time $n+1$.

Can be computed analytically (using algorithm 1 in Frazier et al. 2009) when

$$X = x \text{ (a singleton)} \quad \text{OR} \quad X = x^{(1)} - x^{(2)} \text{ (difference between a pair)}$$

Knowledge Gradient Based Algorithms

At time n: the *Knowledge Gradient (KG) factor* of sampling X next, is

$$\nu_n^{KG}(X) = V_n(X)/c(X),$$

where $c(X)$ is the computation cost for sampling X.

Knowledge Gradient Based Algorithms

At time n: the *Knowledge Gradient (KG) factor* of sampling X next, is

$$\nu_n^{KG}(X) = V_n(X)/c(X),$$

where $c(X)$ is the computation cost for sampling X.

Algorithms for chosing X_{n+1}:

- KG: sample the alternative x with the largest KG factor.

  — Introduced in Gupta and Miescke 1996, Frazier and Powell 2008, and discussed further in Frazier et al. 2008, Chick and Frazier 2009.

Knowledge Gradient Based Algorithms

At time n: the *Knowledge Gradient (KG) factor* of sampling X next, is

$$\nu_n^{KG}(X) = V_n(X)/c(X),$$

where $c(X)$ is the computation cost for sampling X.

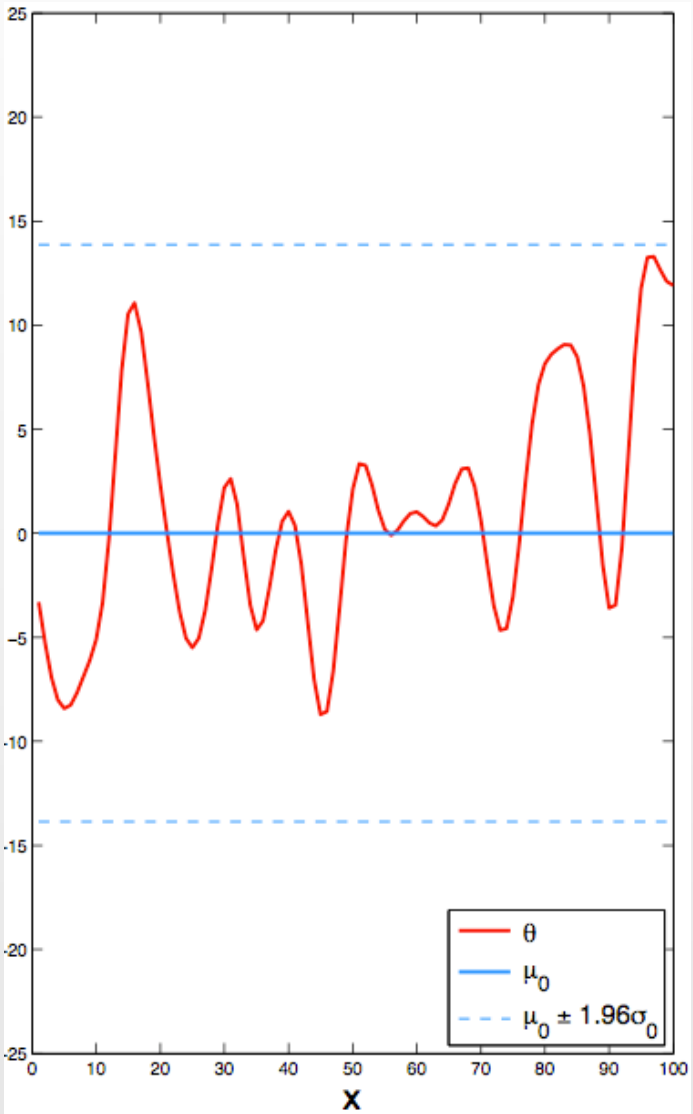
Algorithms for chosing X_{n+1}:

- KG: sample the alternative x with the largest KG factor.

  — Introduced in Gupta and Miescke 1996, Frazier and Powell 2008, and discussed further in Frazier et al. 2008, Chick and Frazier 2009.

- **KG²**: check the KG factors of all singletons x and pairs (x^1, x^2), and sample the one(s) with the largest factor.

Illustration of the KG Algorithm



$$k = 100.$$

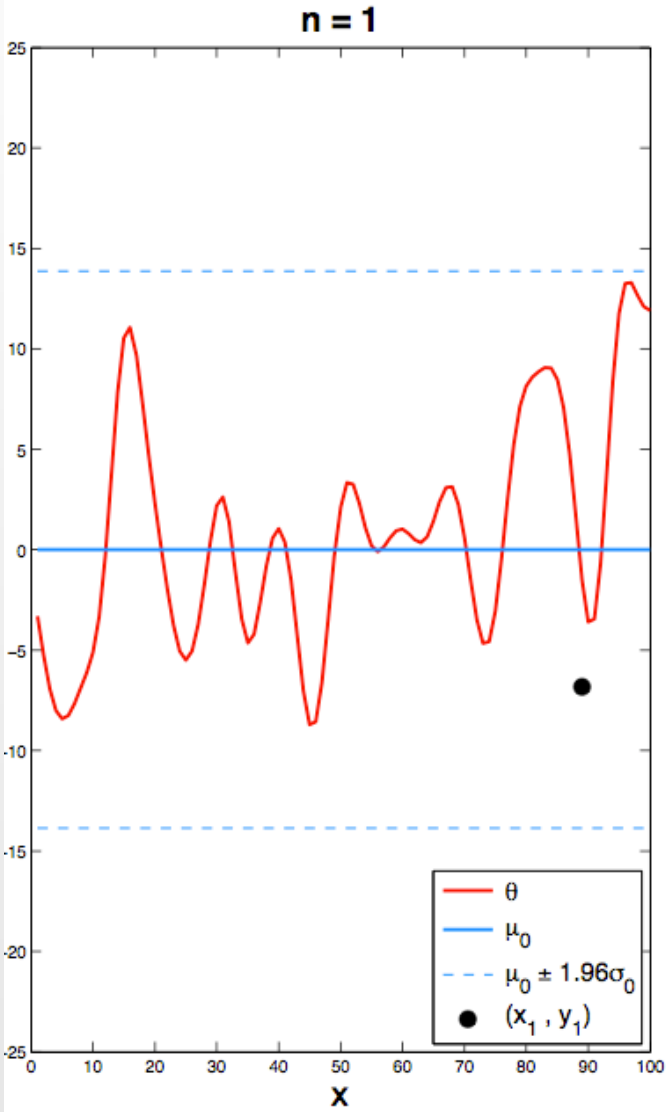
$$\Lambda(i,i) = 50,$$

$$\Lambda(i,j) = 25, \text{ for } i,j = 1,\ldots,100.$$

$$\mu_0 = \vec{0},$$

$$\Sigma_0(i,j) = 100 \exp\left[-\frac{(i-j)^2}{50}\right].$$

Illustration of the KG Algorithm



$$k = 100.$$

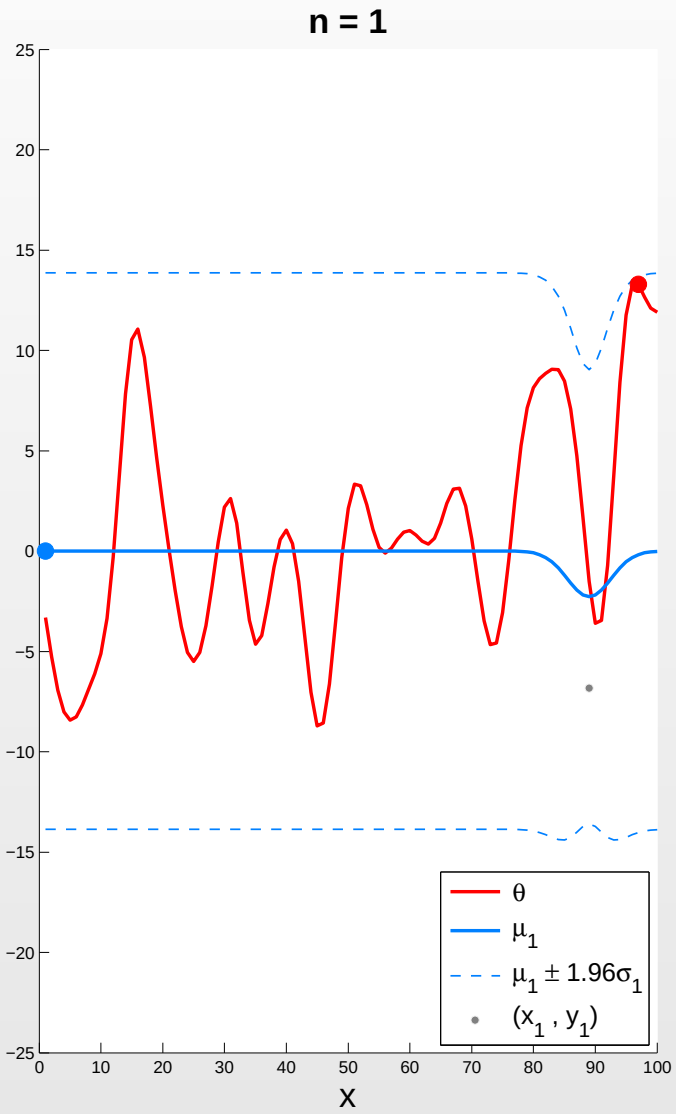
$$\Lambda(i,i) = 50,$$

$$\Lambda(i,j) = 25, \text{ for } i,j = 1,\ldots,100.$$

$$\mu_0 = \vec{0},$$

$$\Sigma_0(i,j) = 100 \exp\left[-\frac{(i-j)^2}{50}\right].$$

Illustration of the KG Algorithm



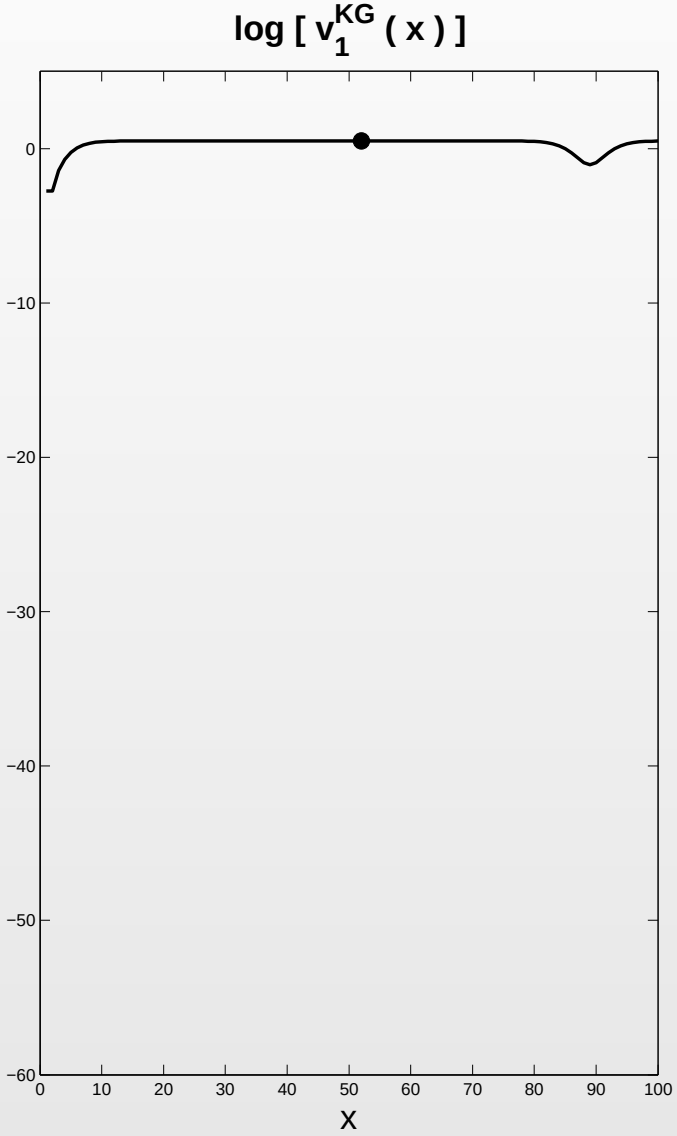
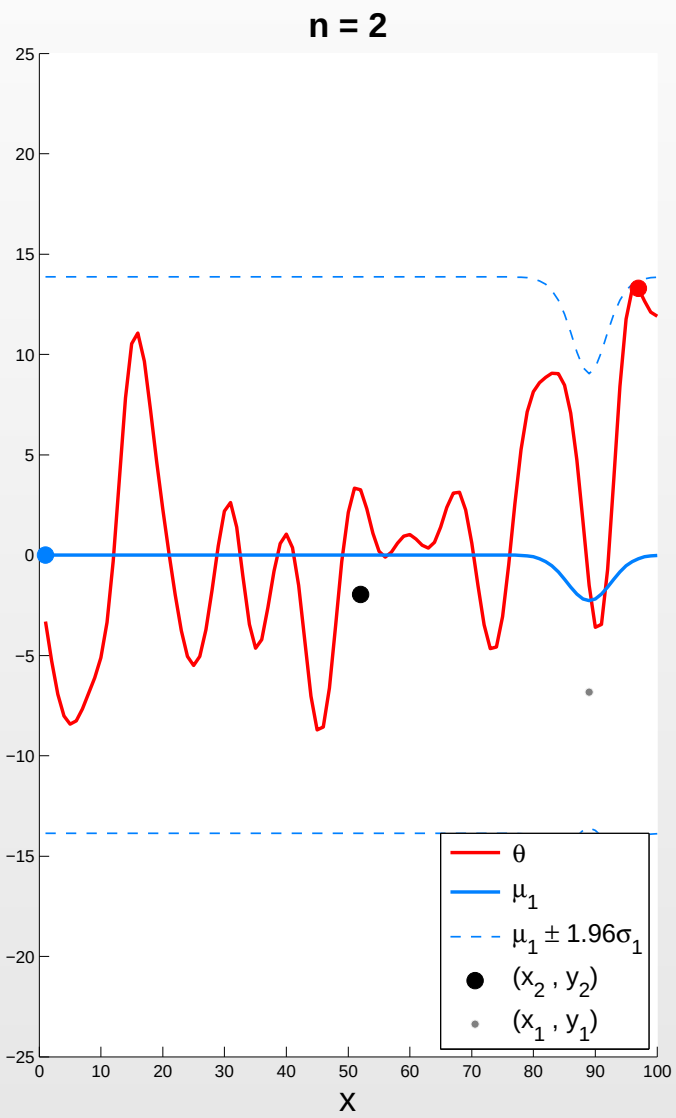


Illustration of the KG Algorithm



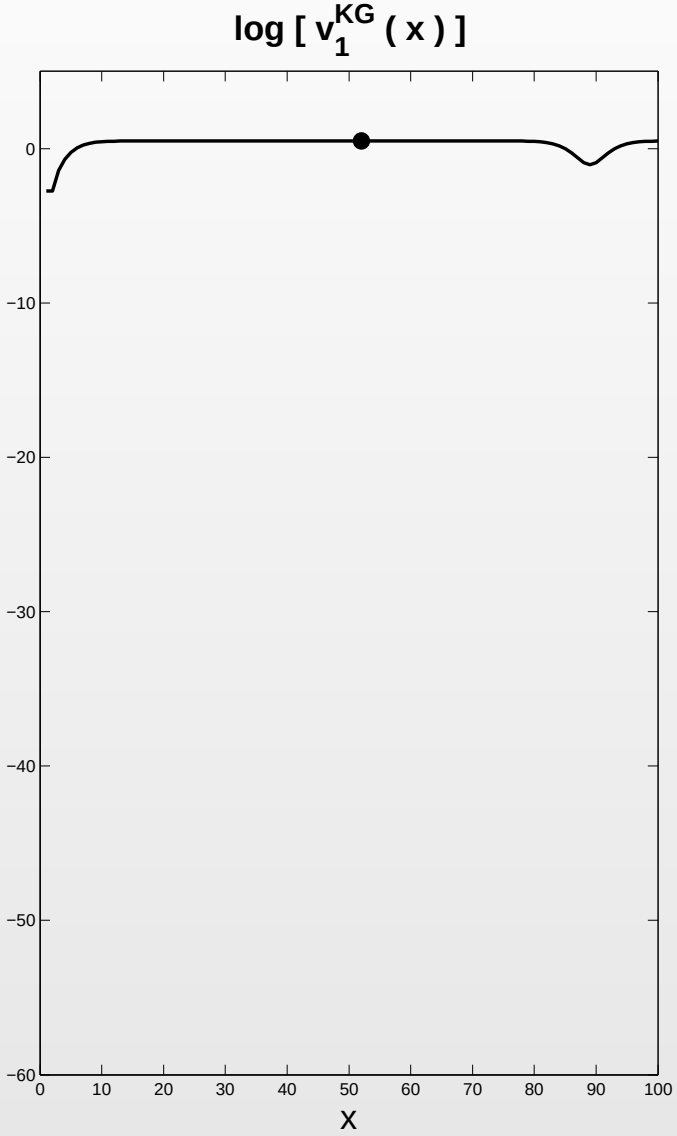
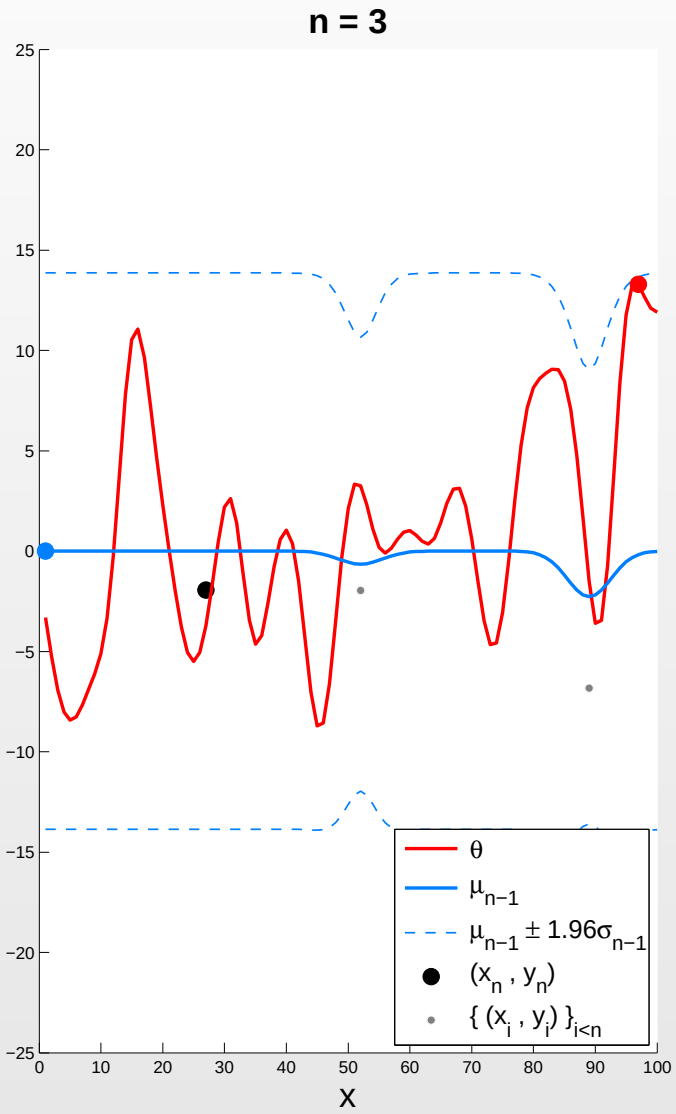


Illustration of the KG Algorithm



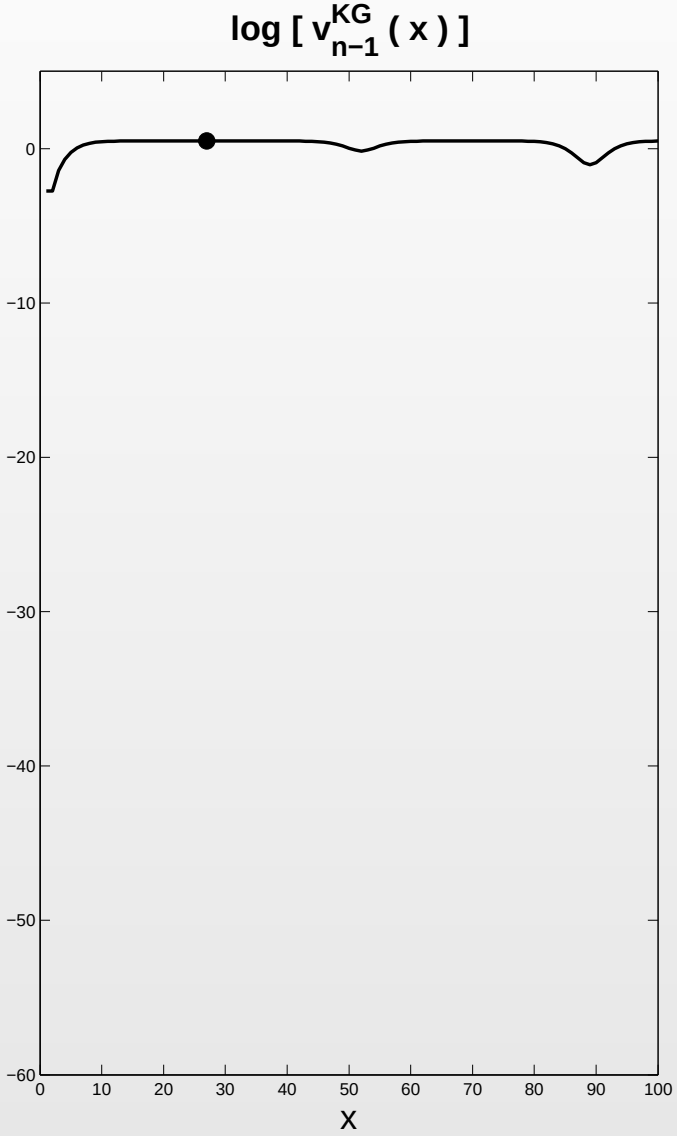
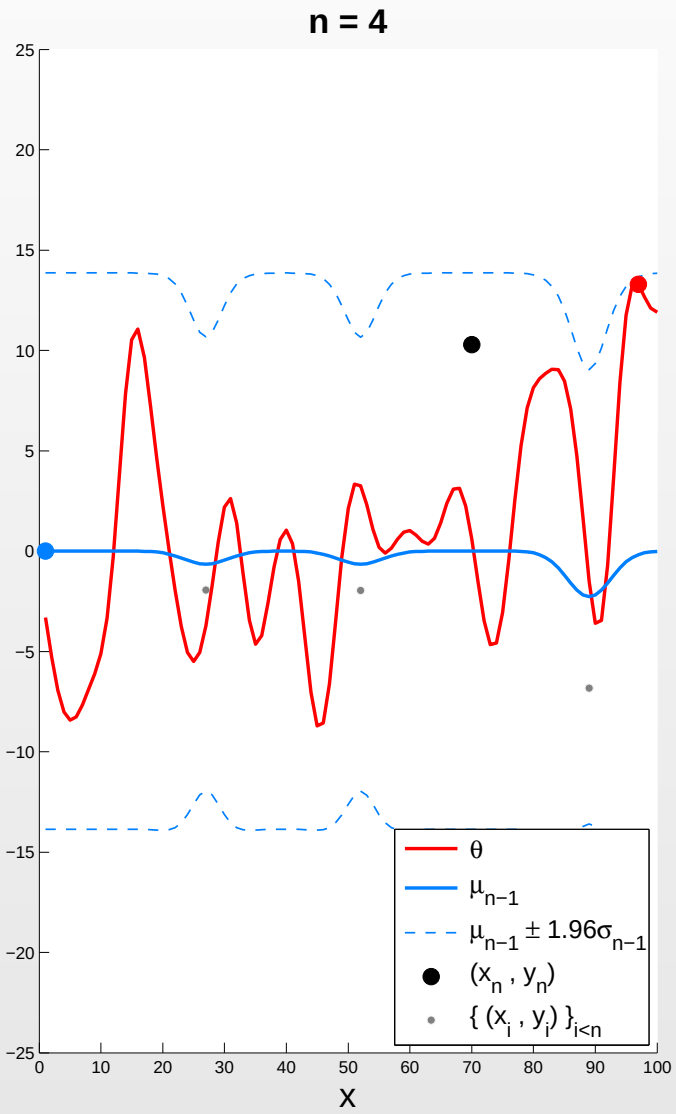


Illustration of the KG Algorithm



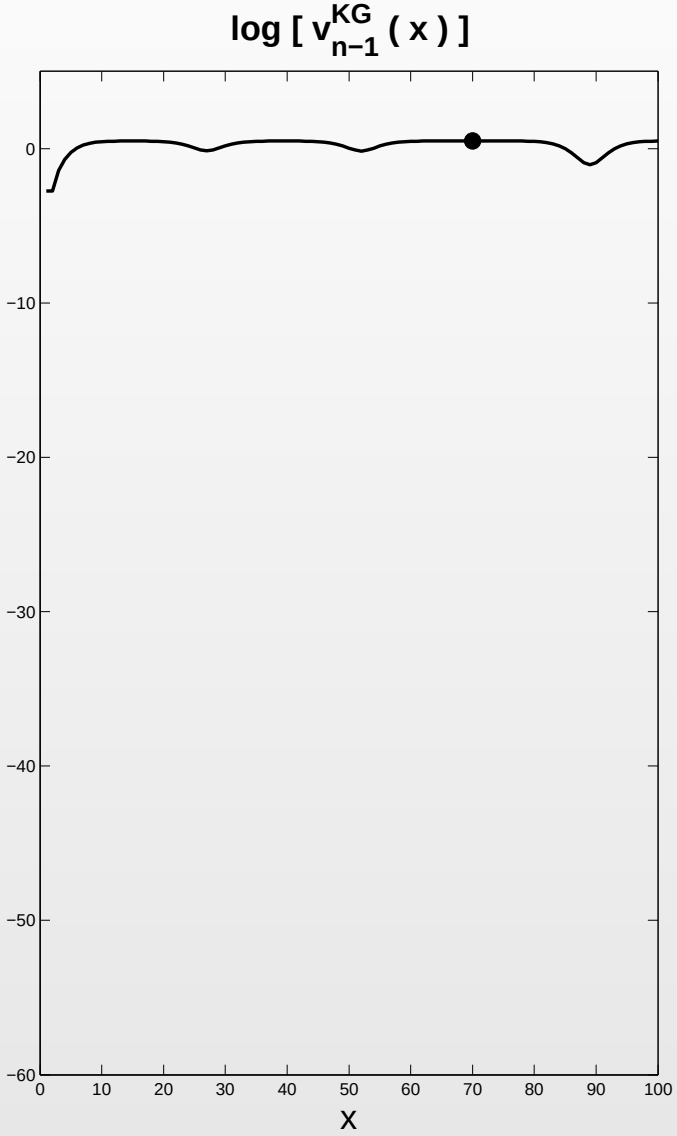
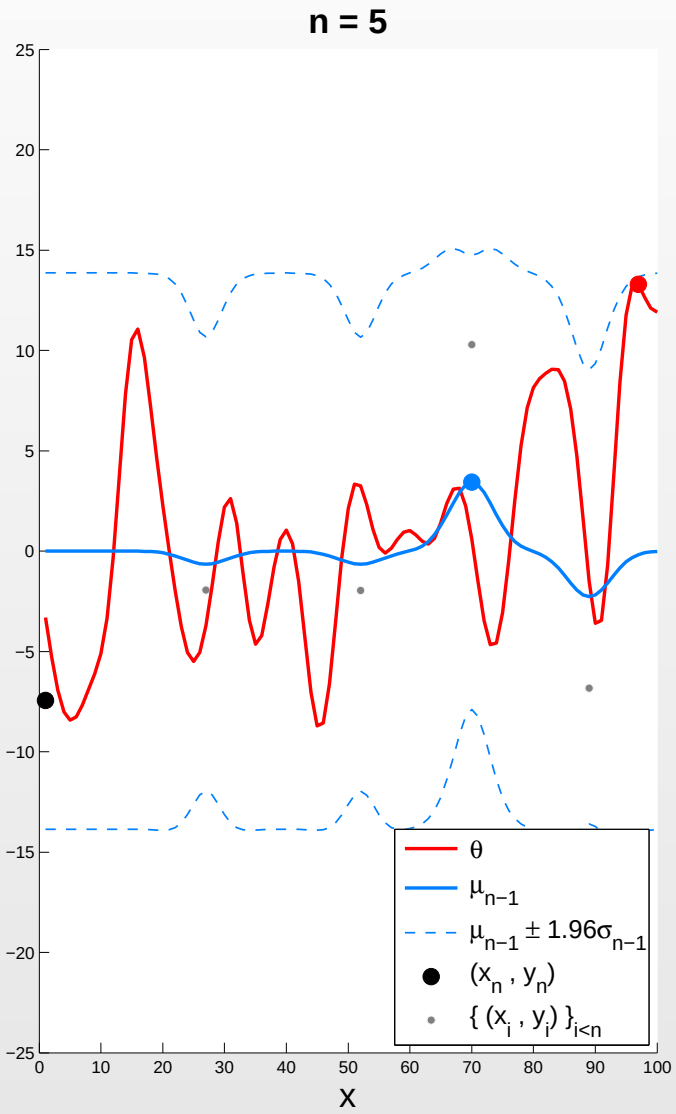


Illustration of the KG Algorithm



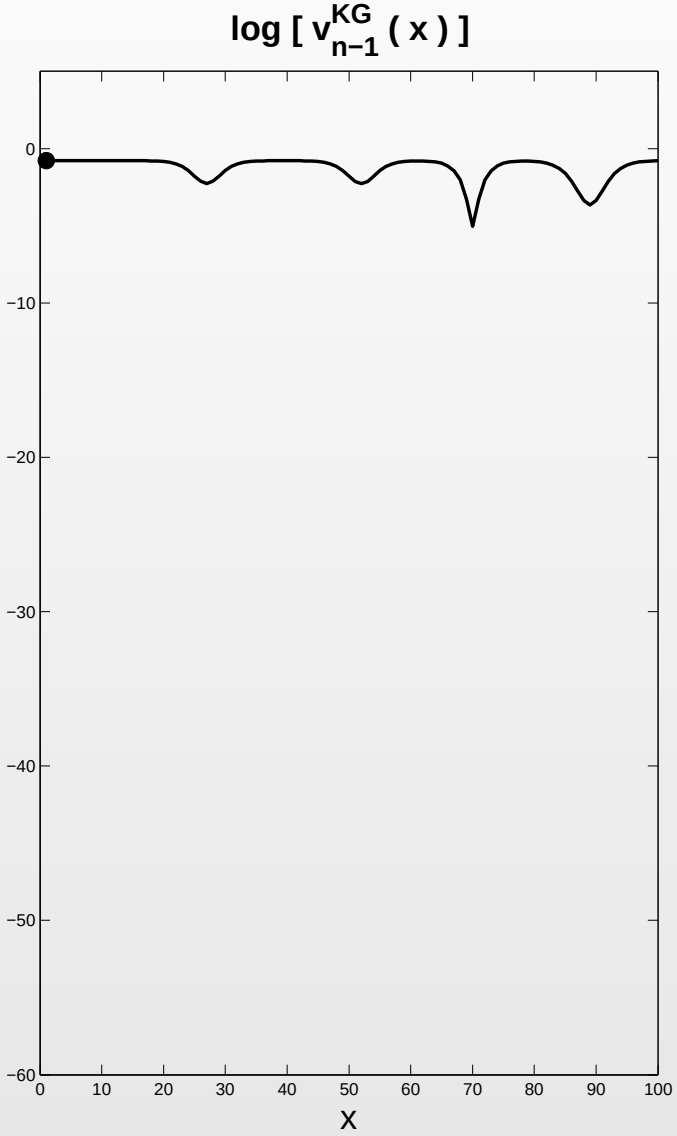
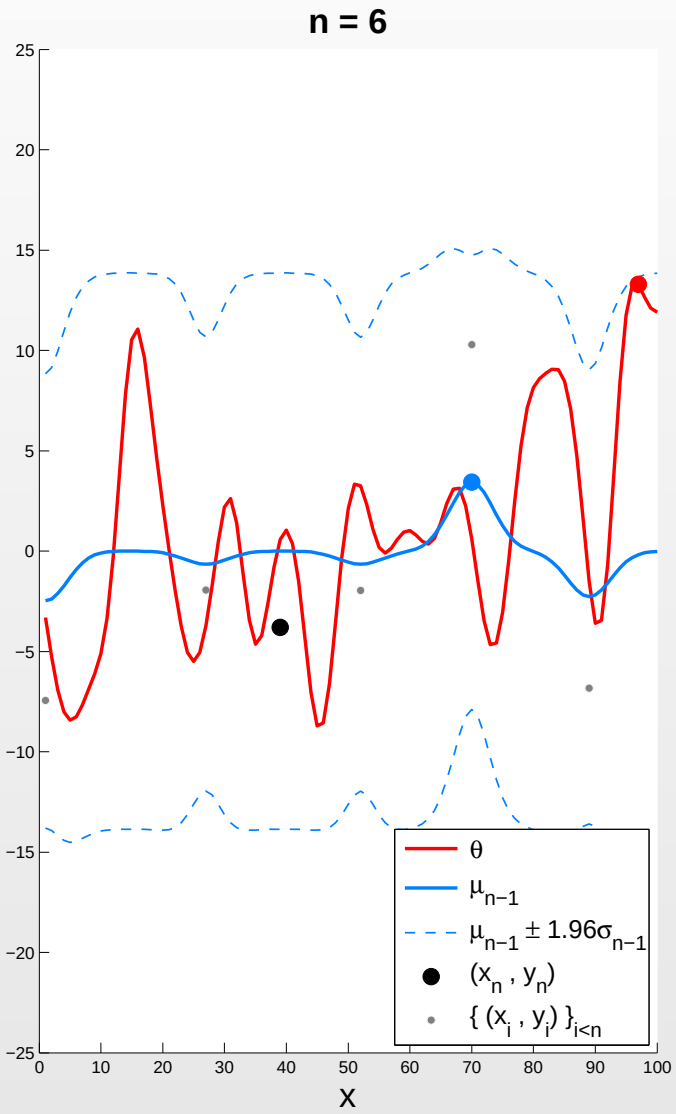


Illustration of the KG Algorithm



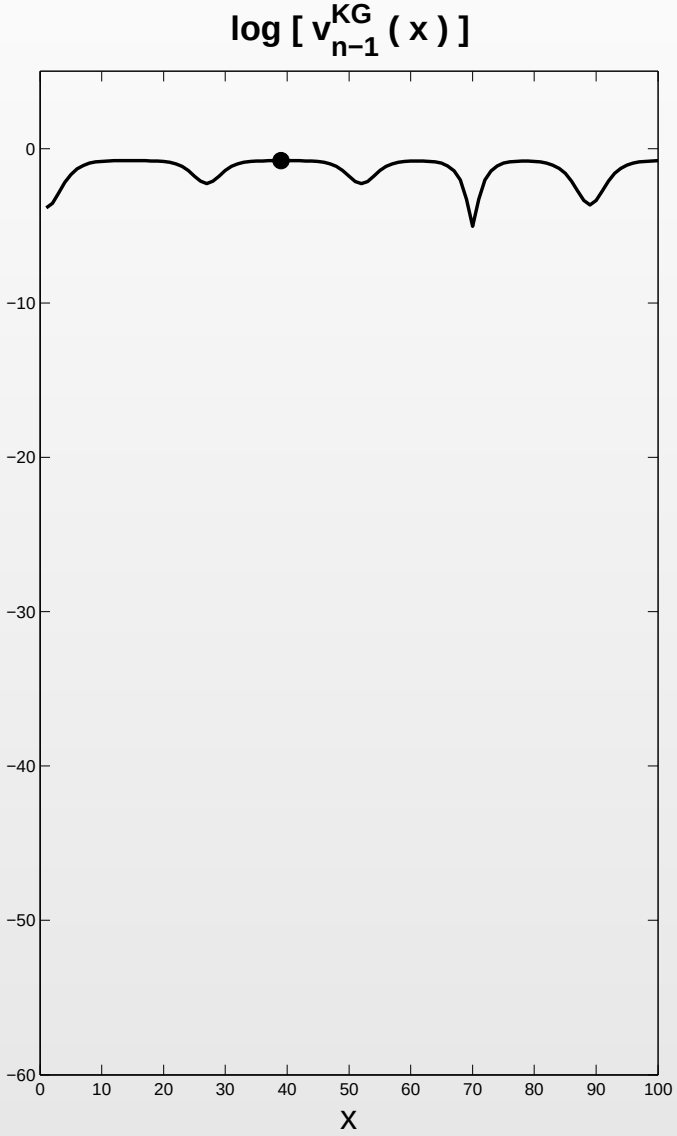
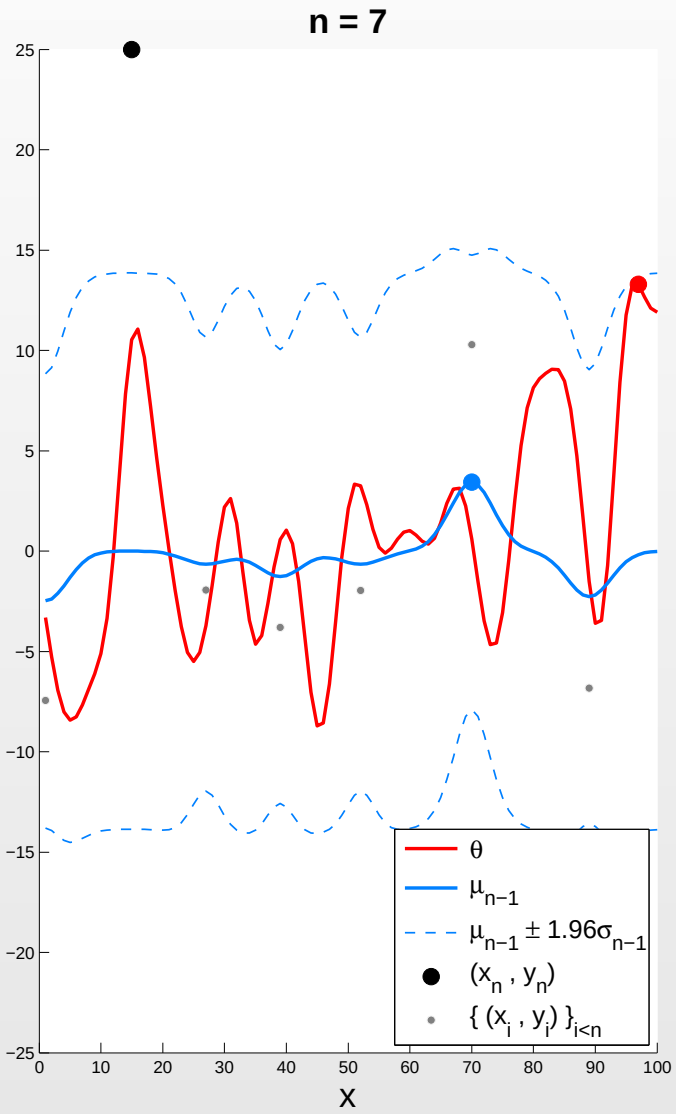


Illustration of the KG Algorithm



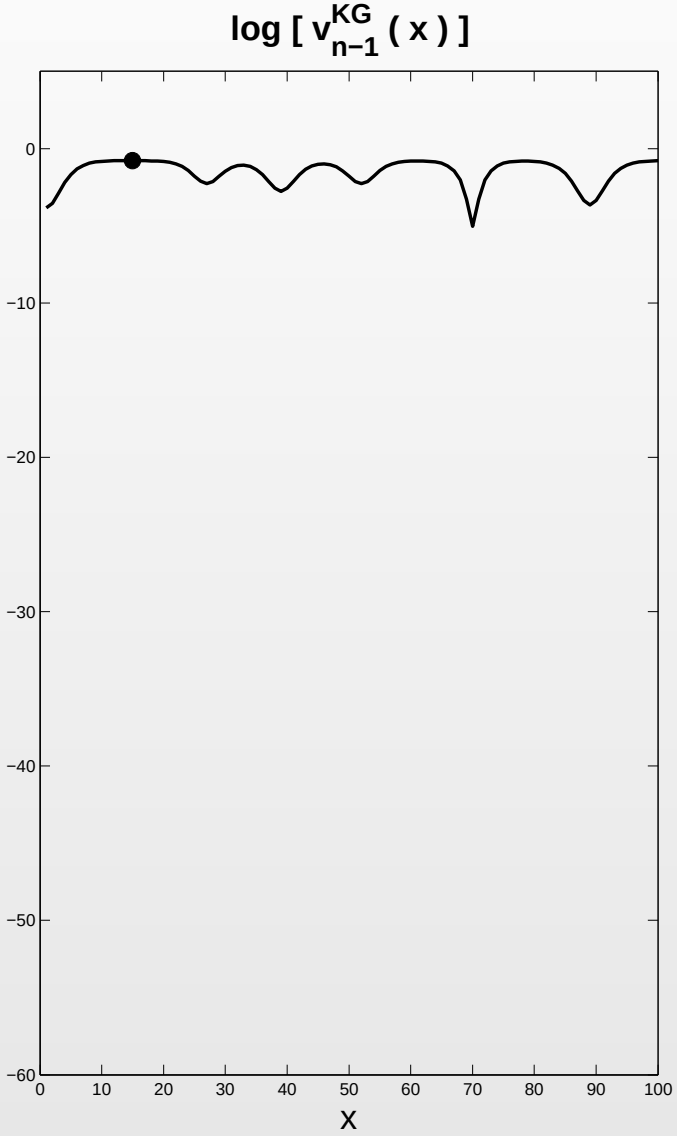
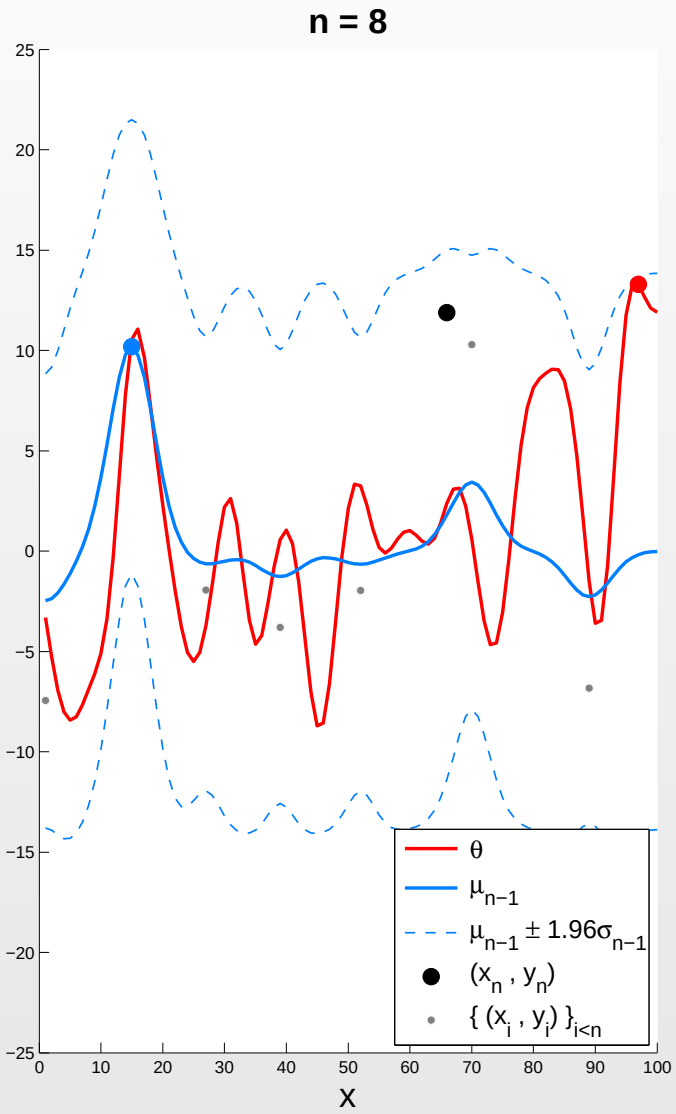


Illustration of the KG Algorithm



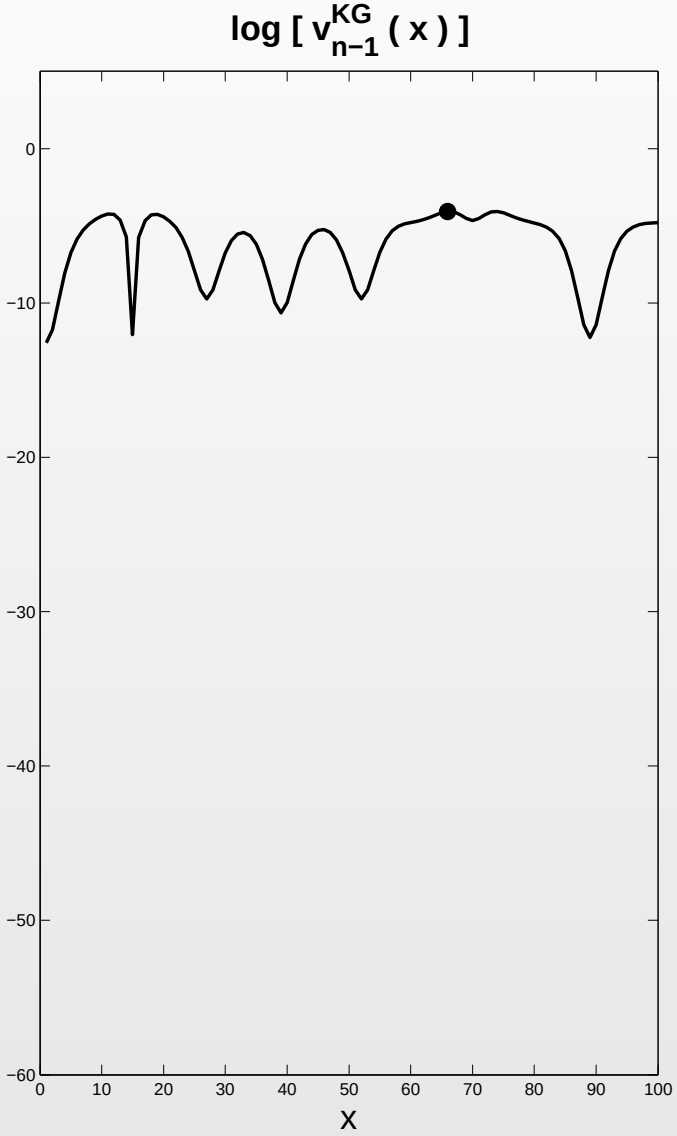
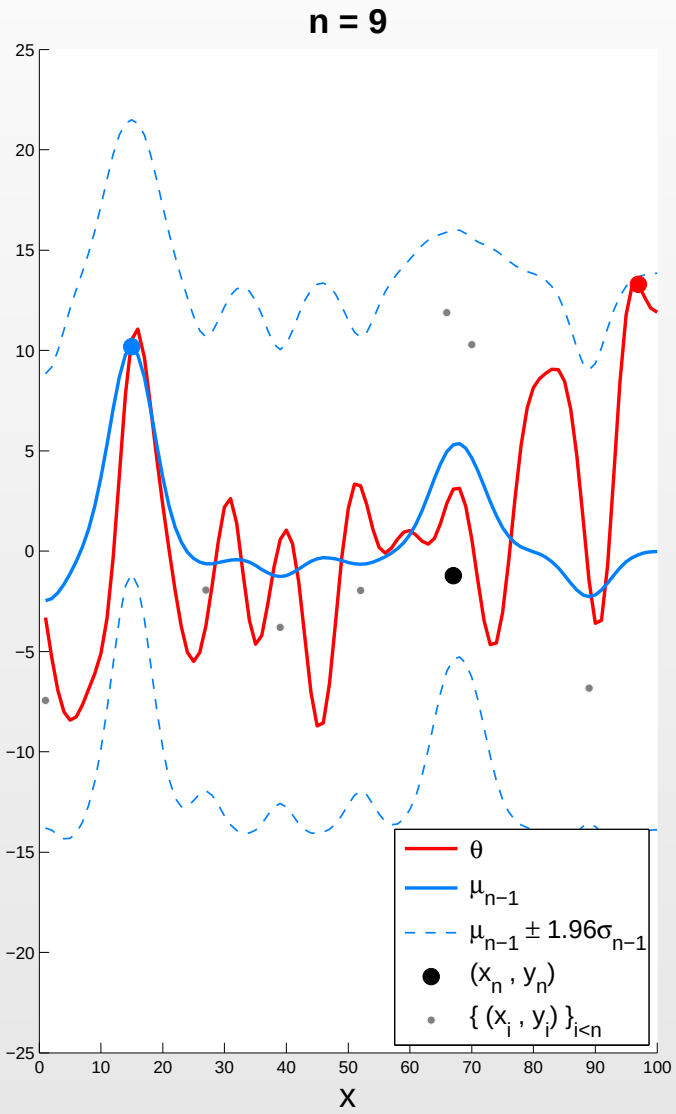


Illustration of the KG Algorithm



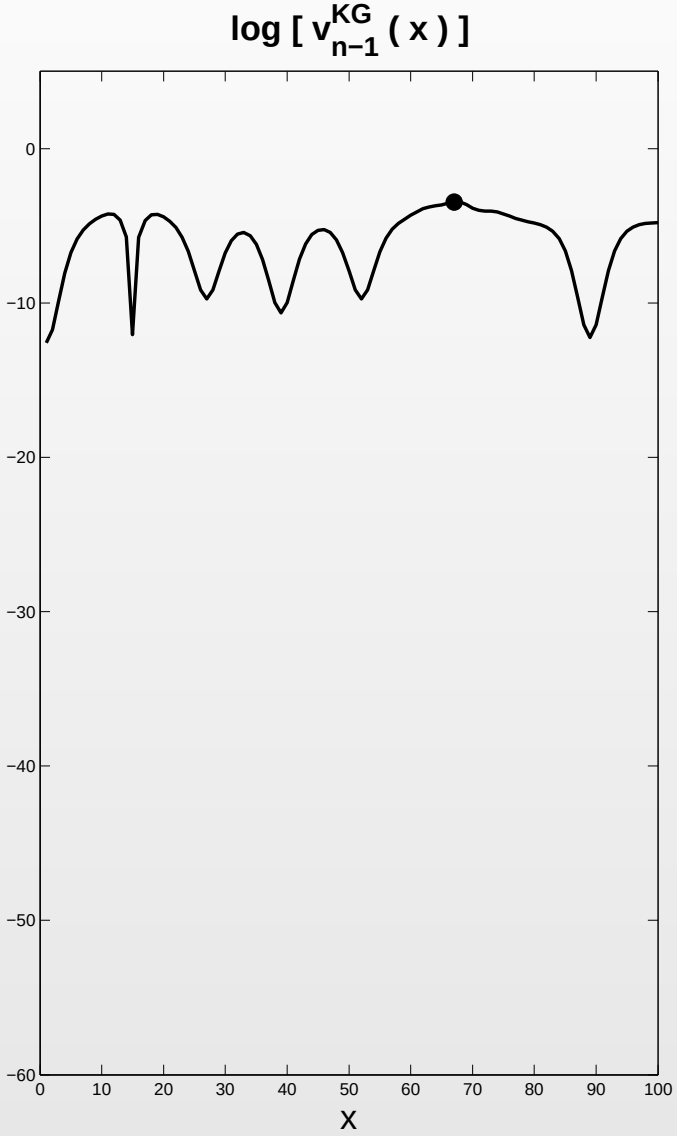
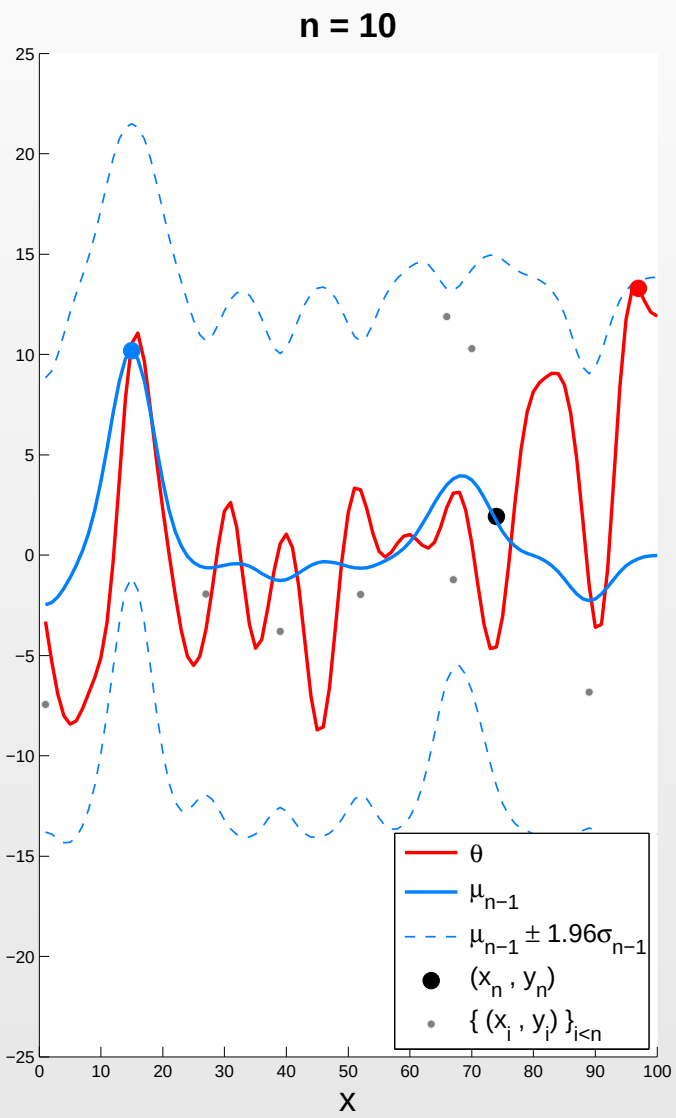


Illustration of the KG Algorithm



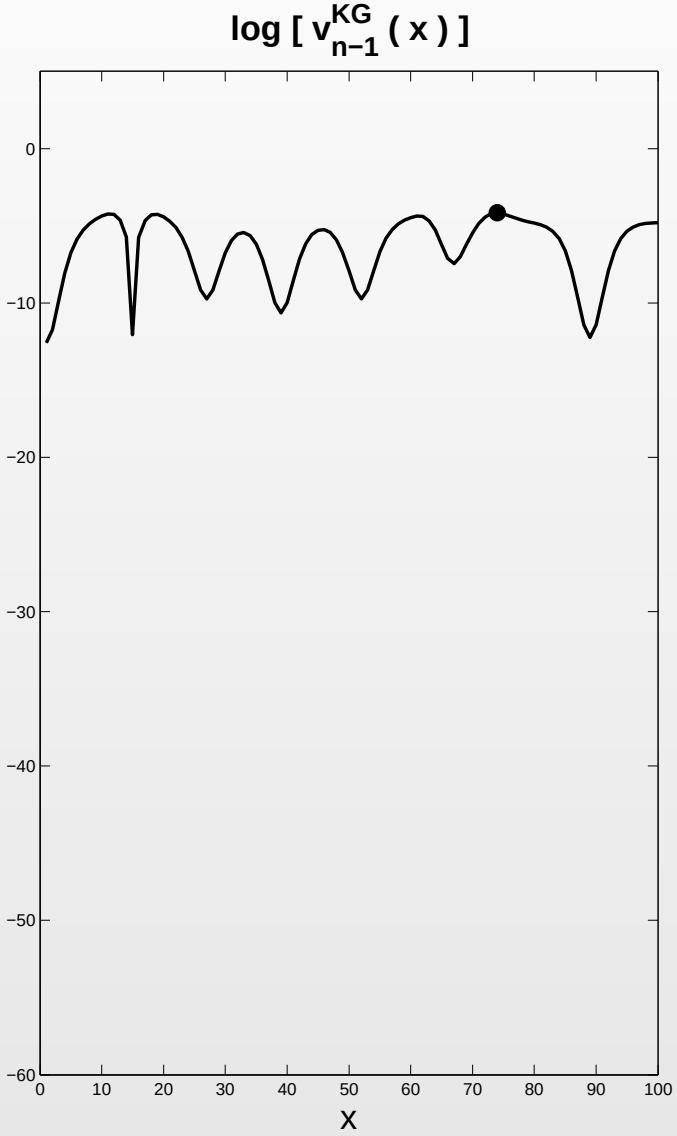
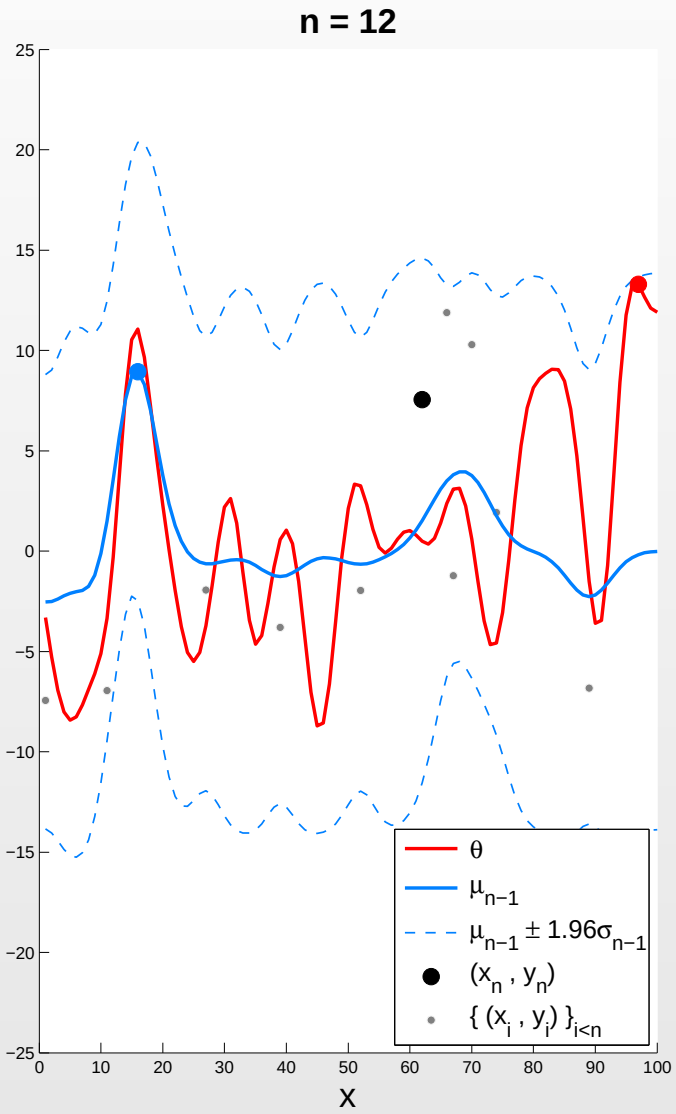


Illustration of the KG Algorithm



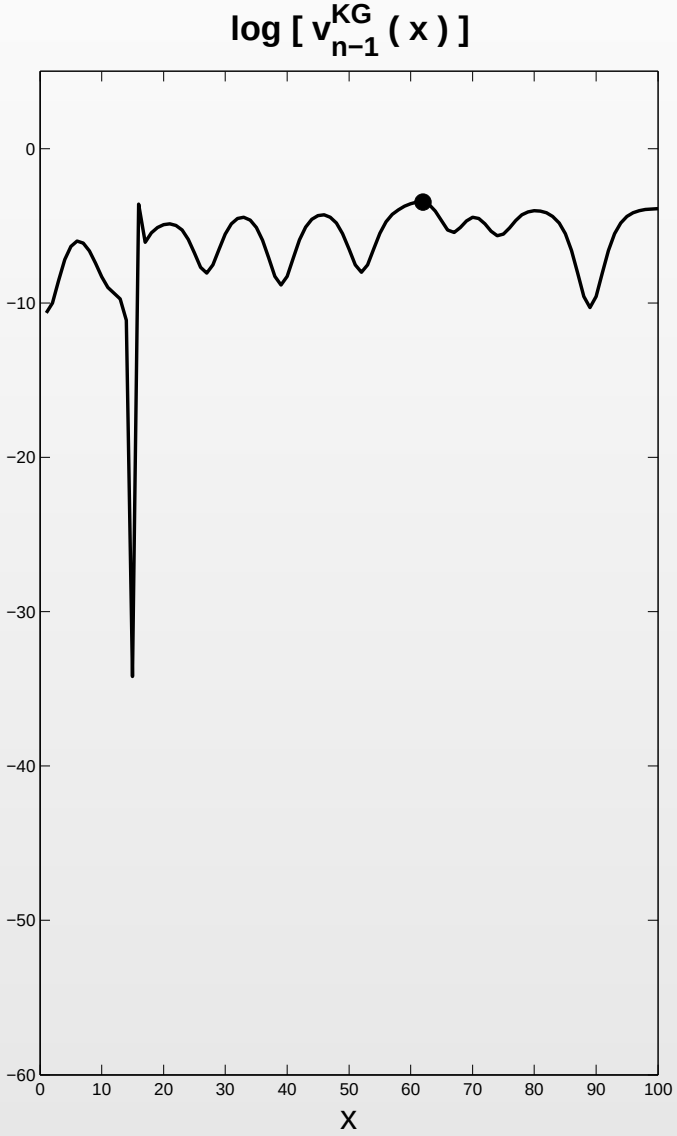
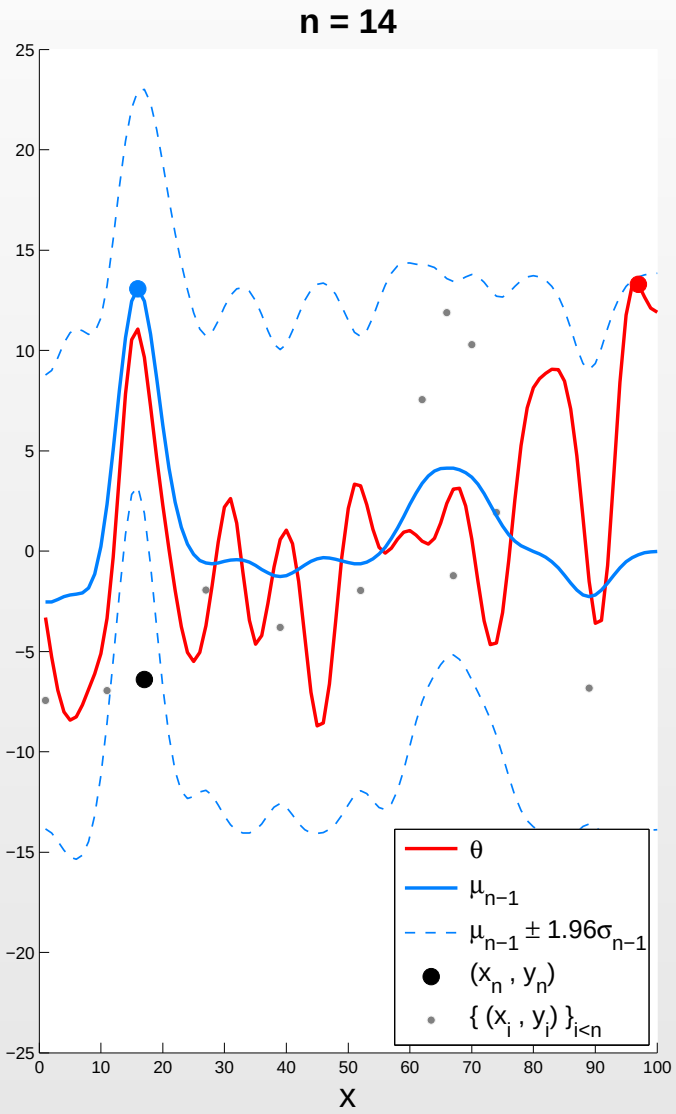


Illustration of the KG Algorithm



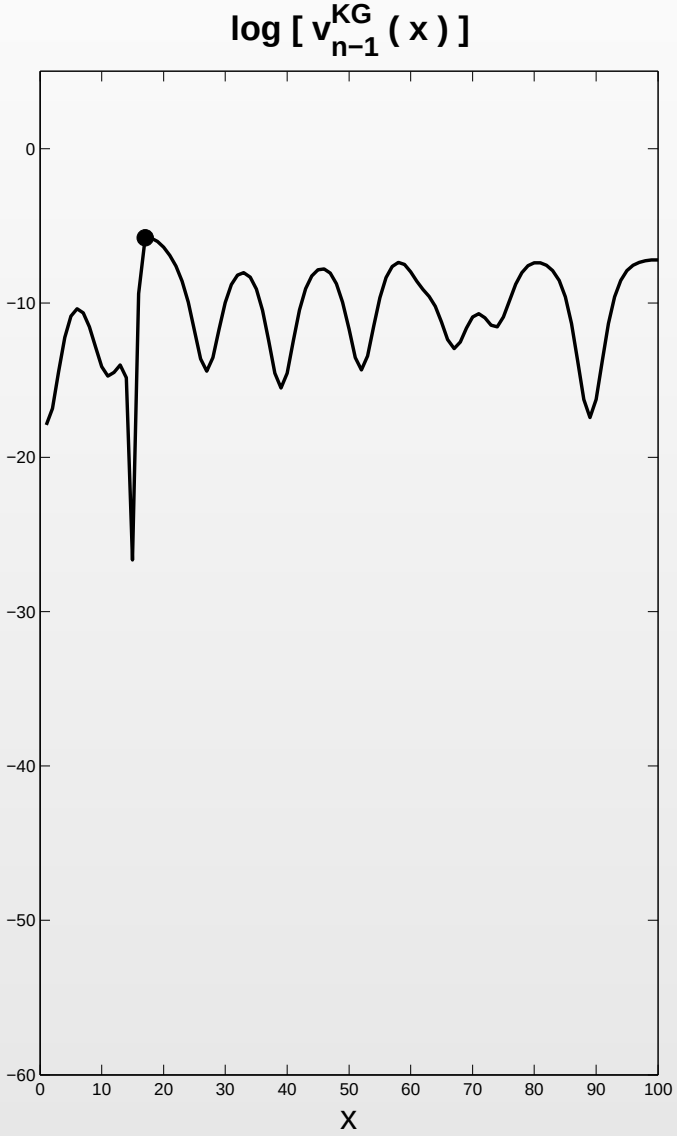
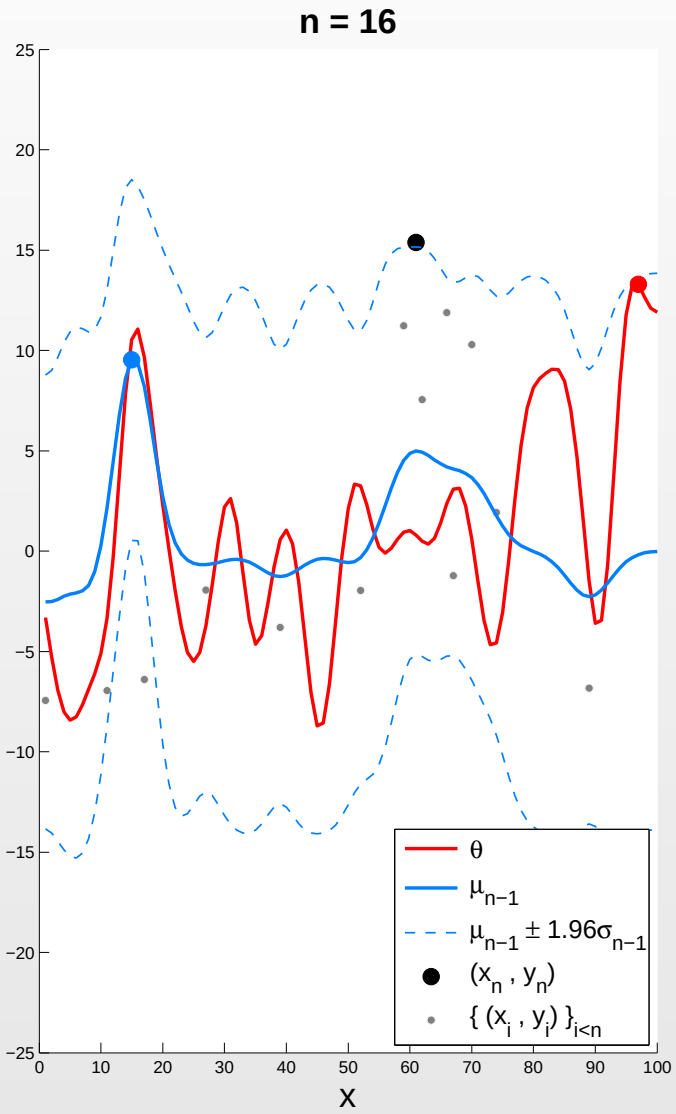


Illustration of the KG Algorithm



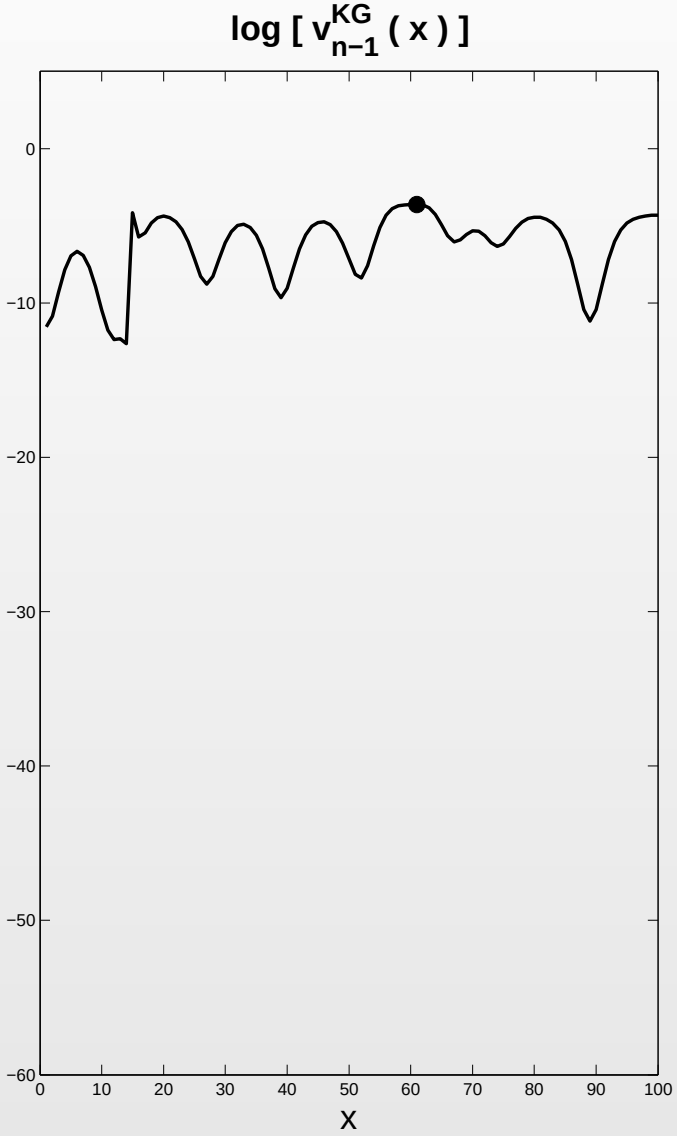
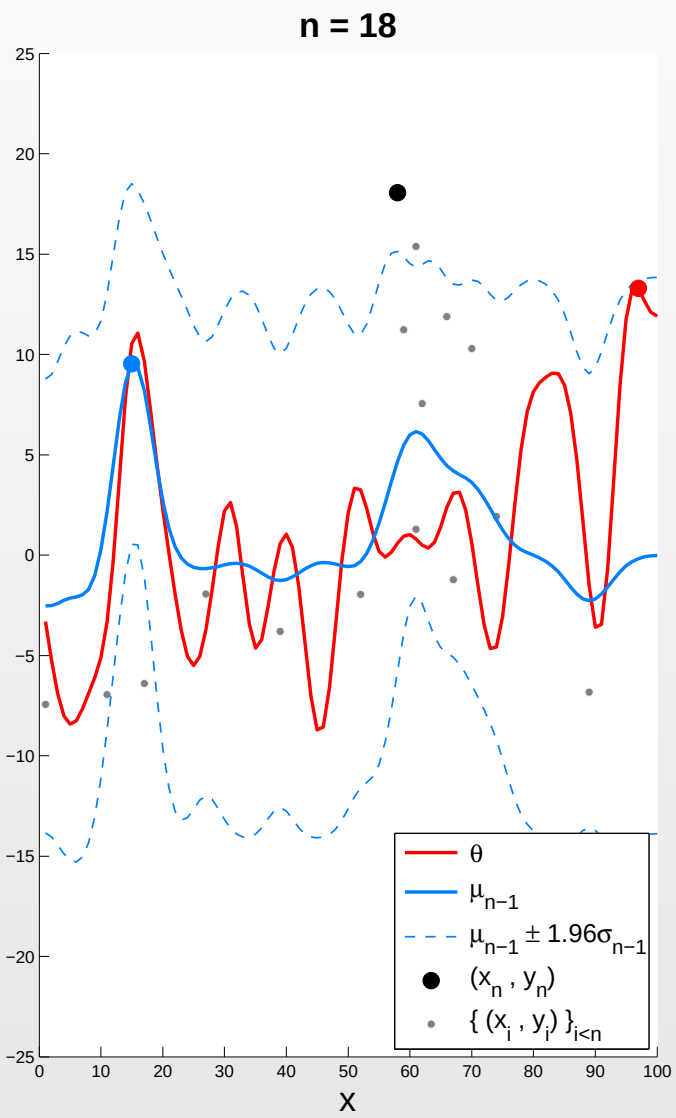


Illustration of the KG Algorithm



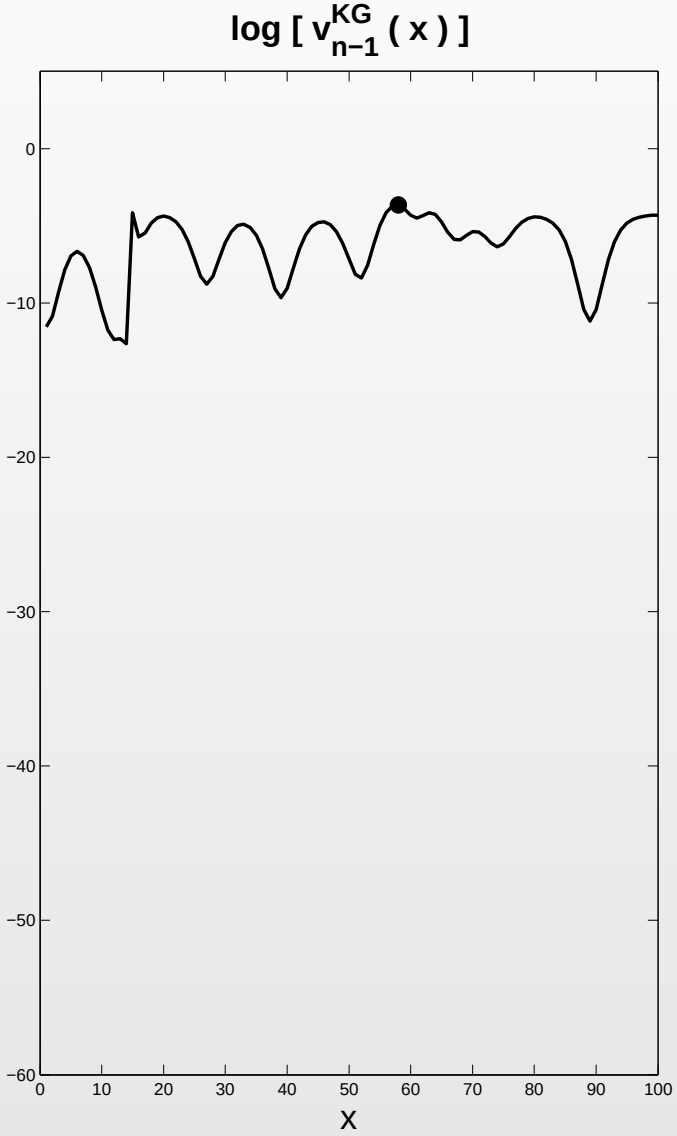
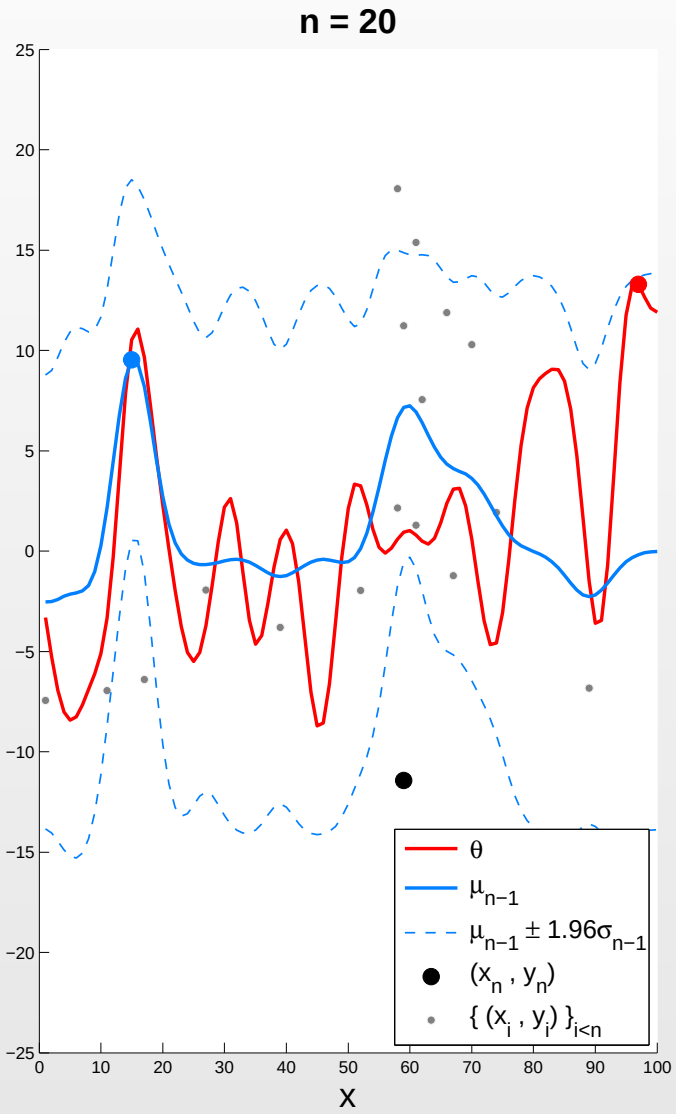


Illustration of the KG Algorithm



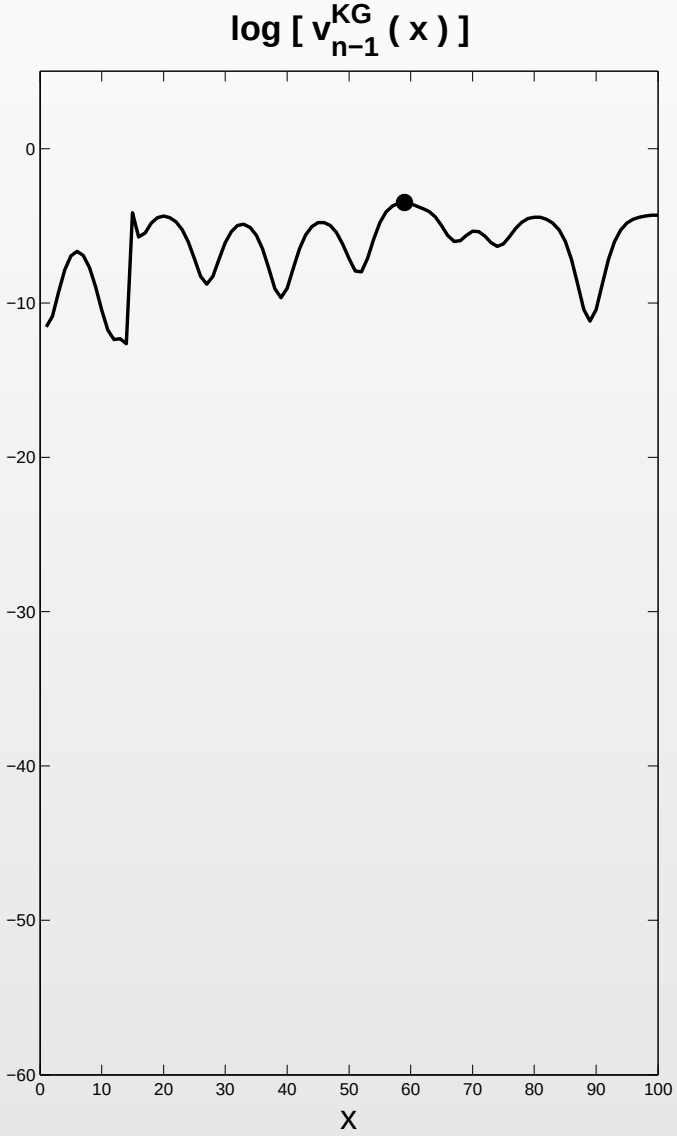
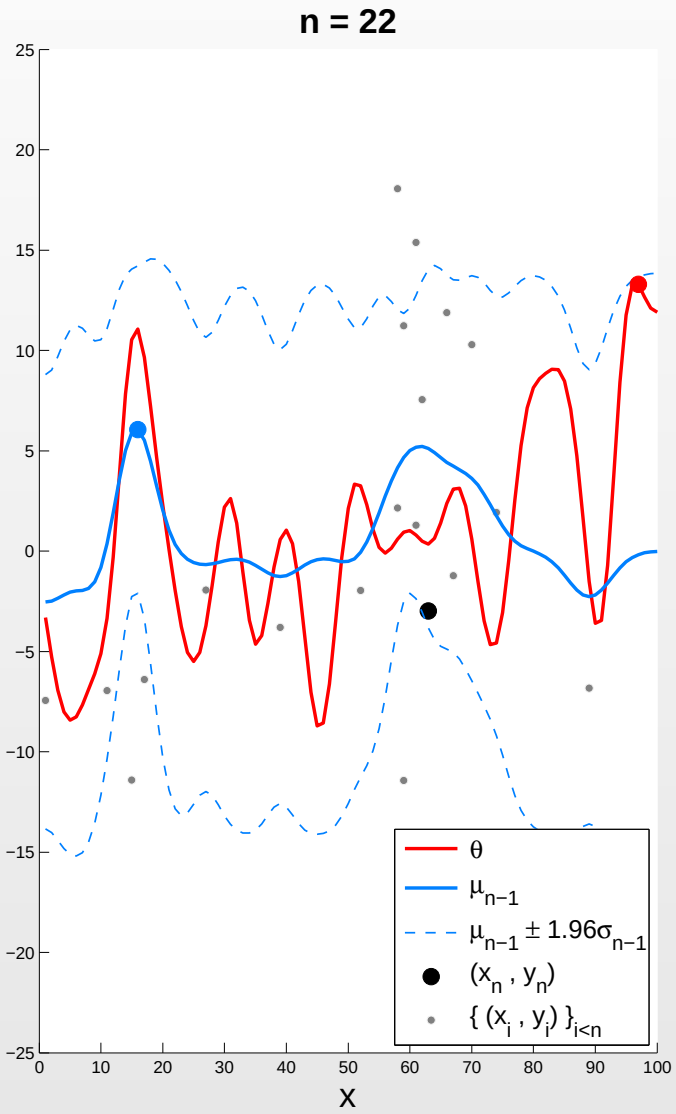


Illustration of the KG Algorithm



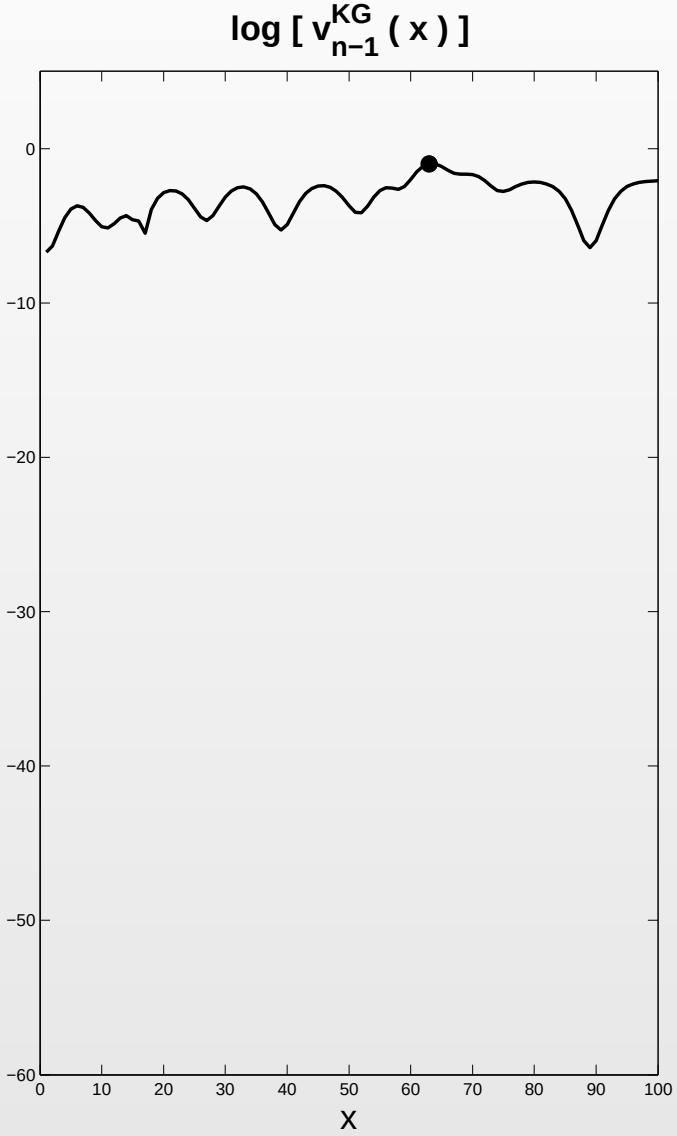
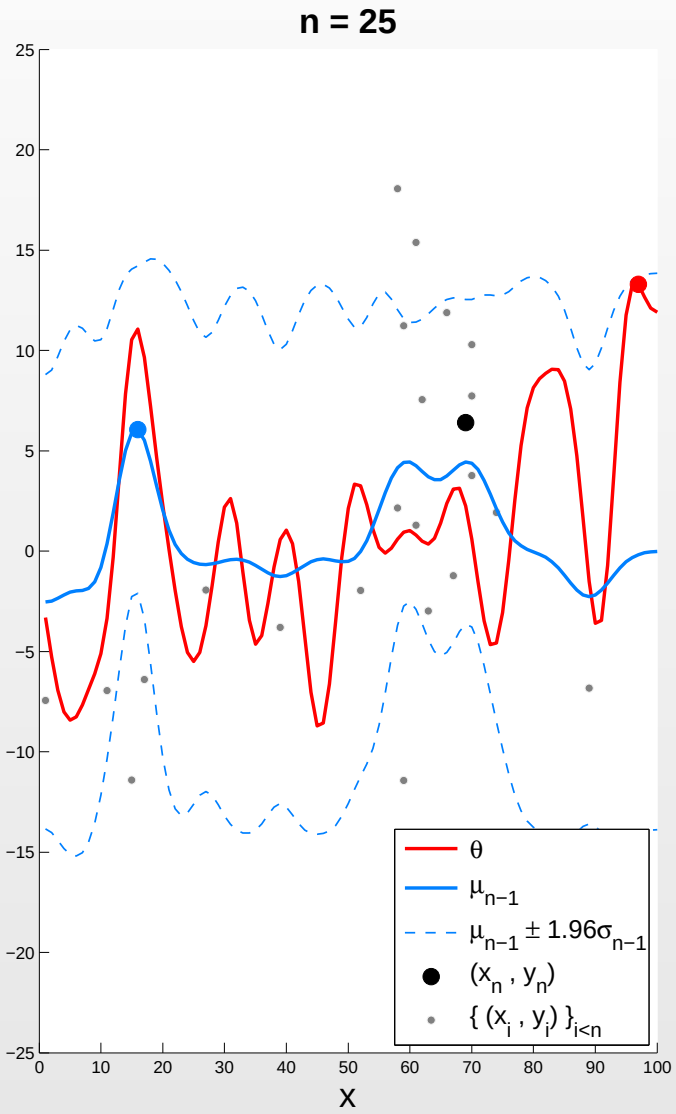


Illustration of the KG Algorithm



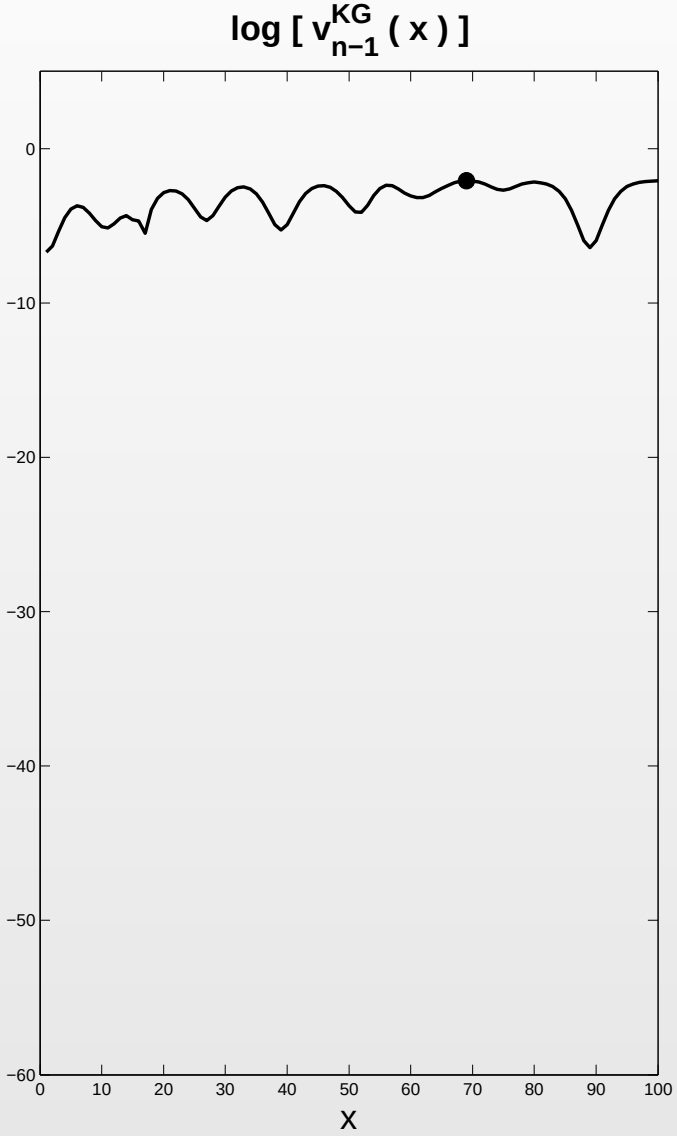
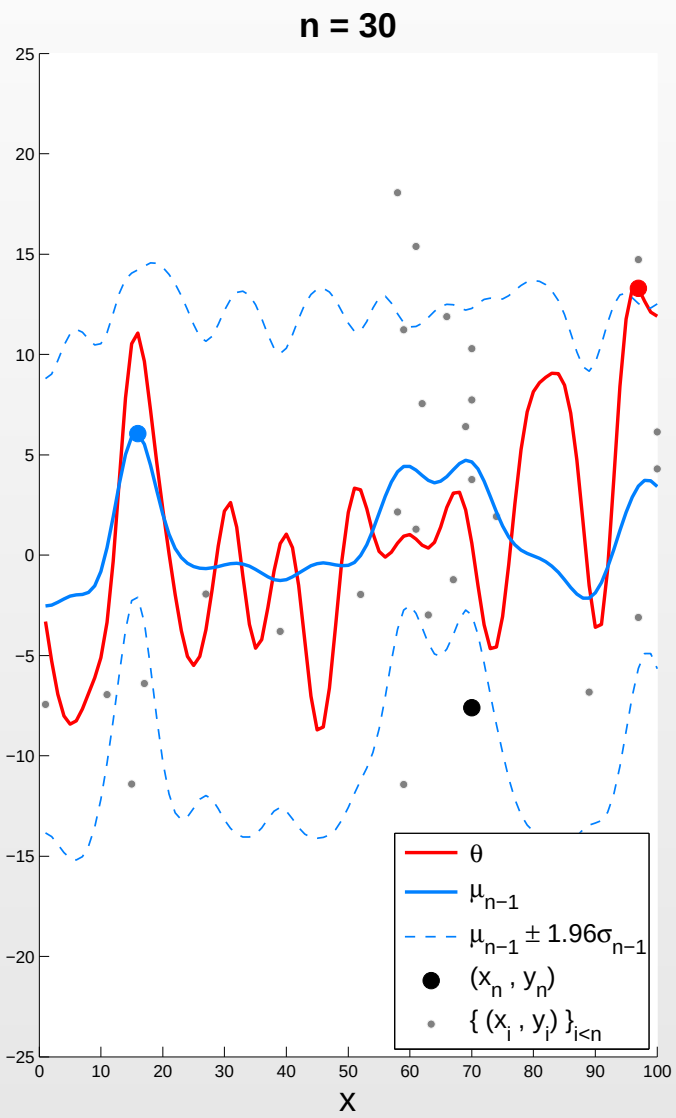


Illustration of the KG Algorithm



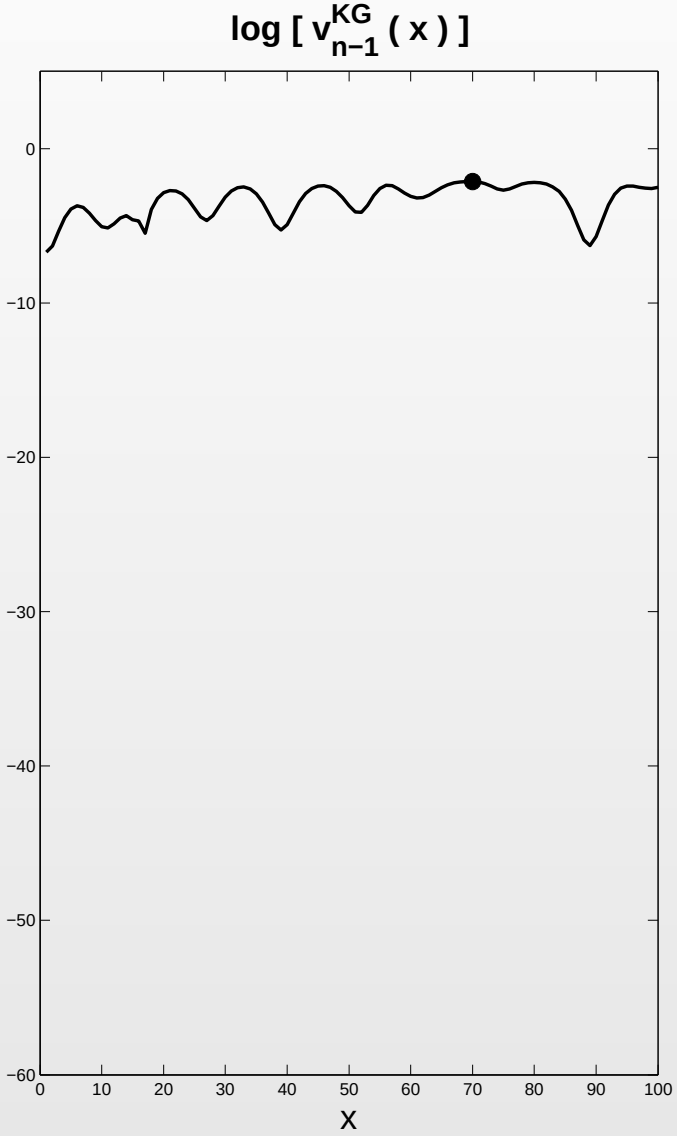
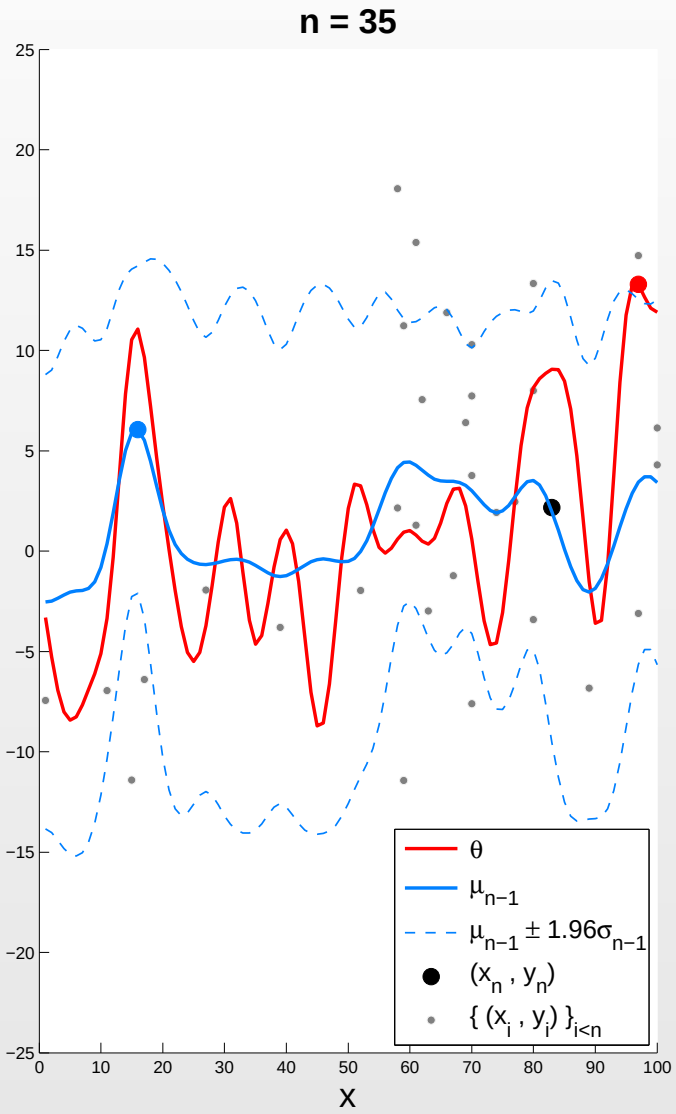


Illustration of the KG Algorithm



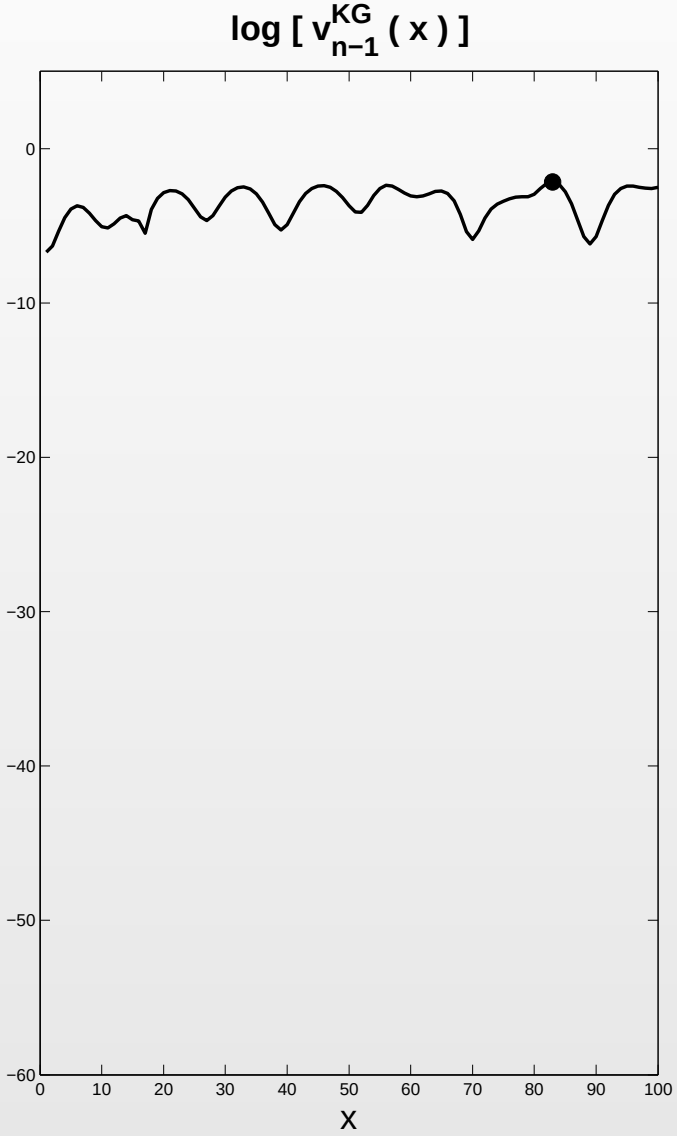
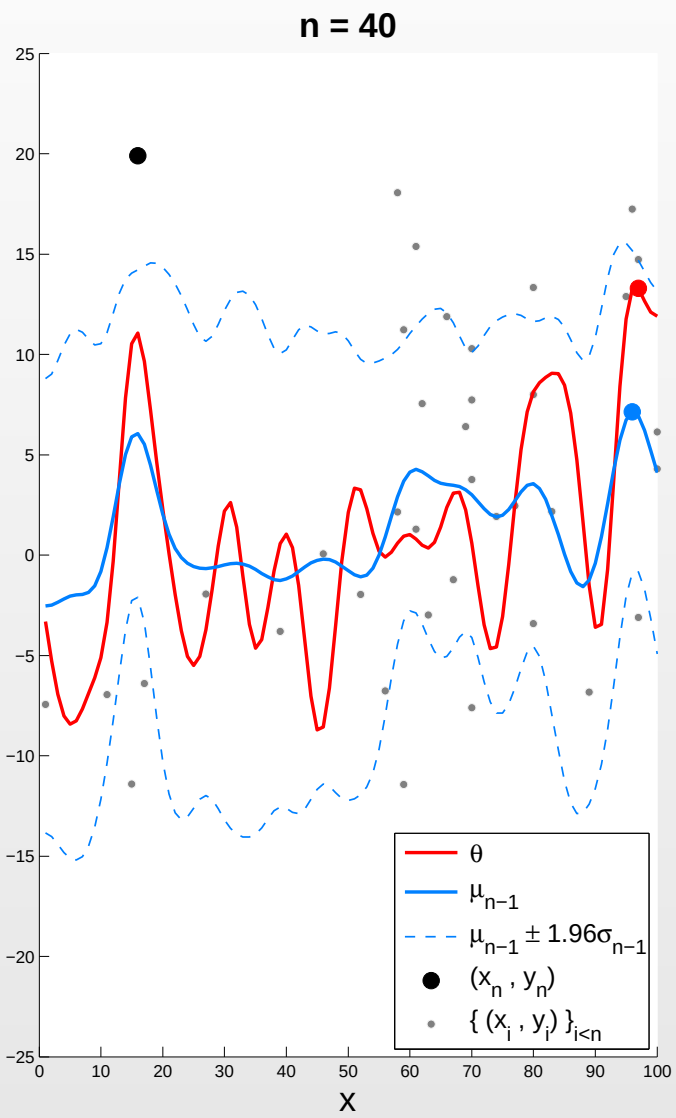


Illustration of the KG Algorithm



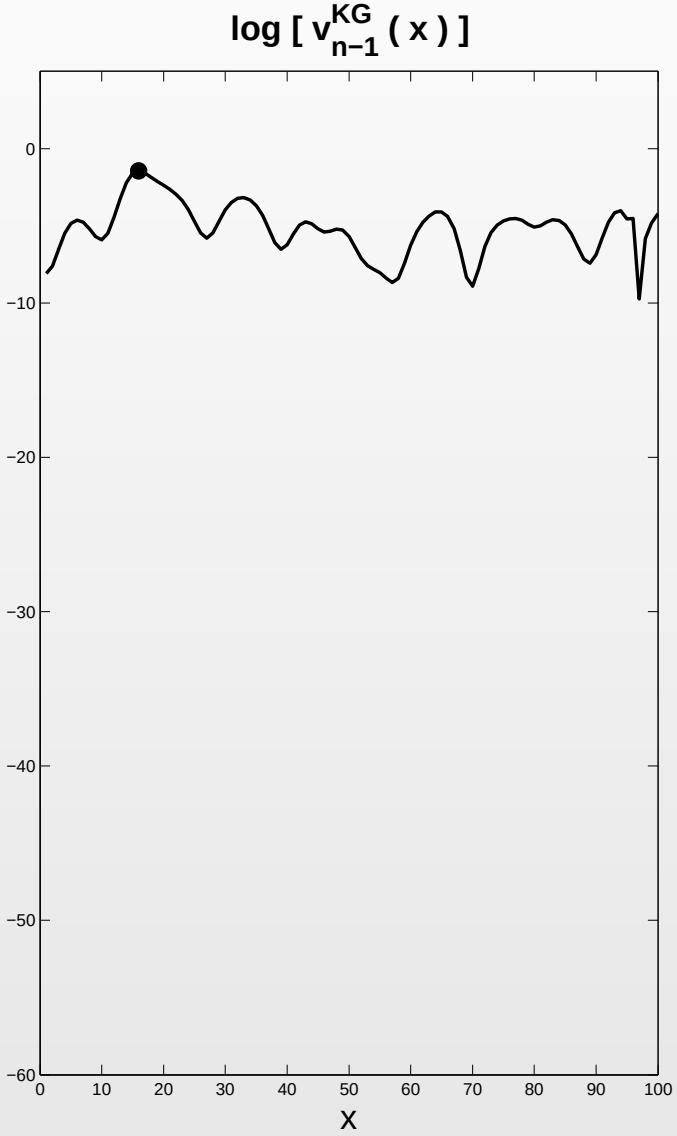
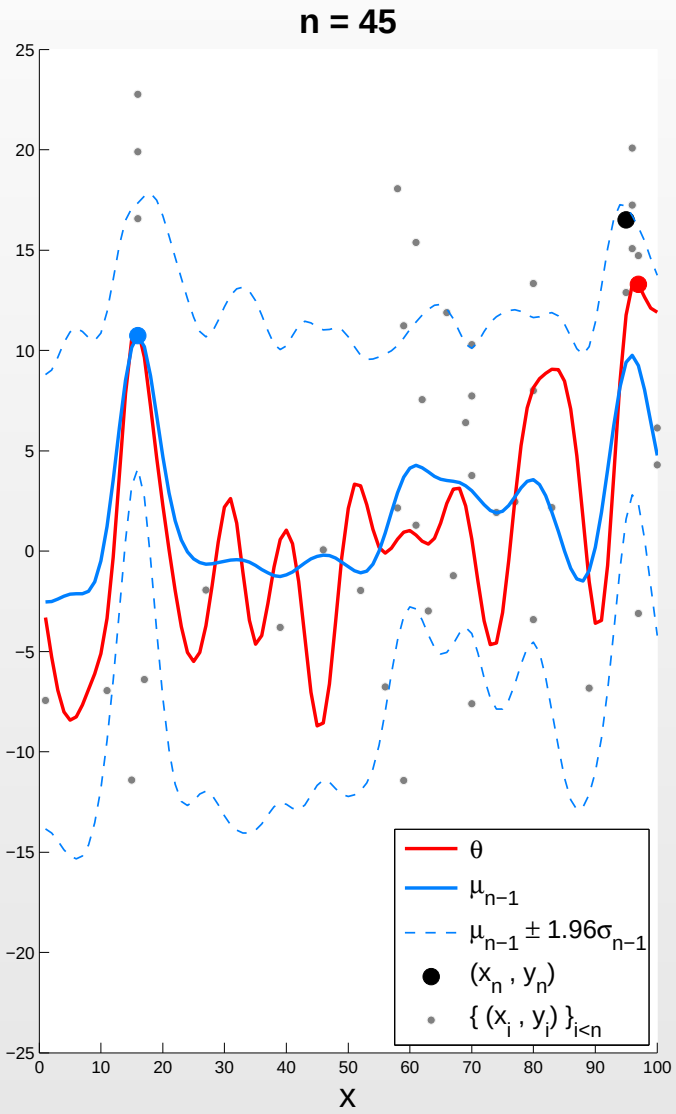


Illustration of the KG Algorithm



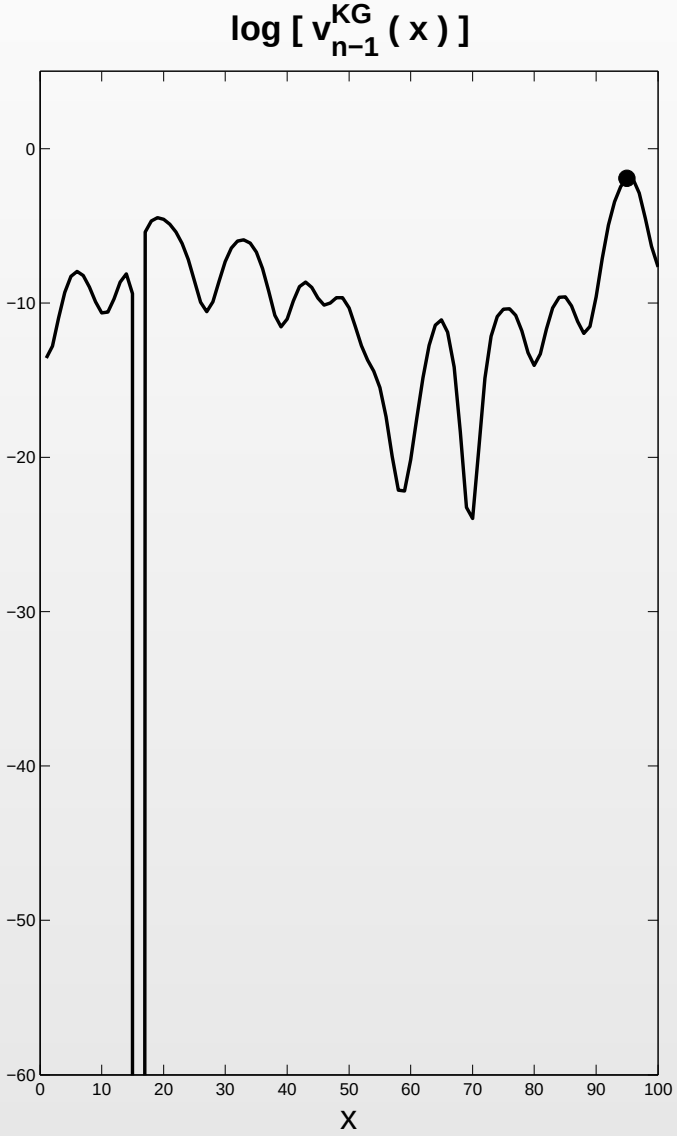
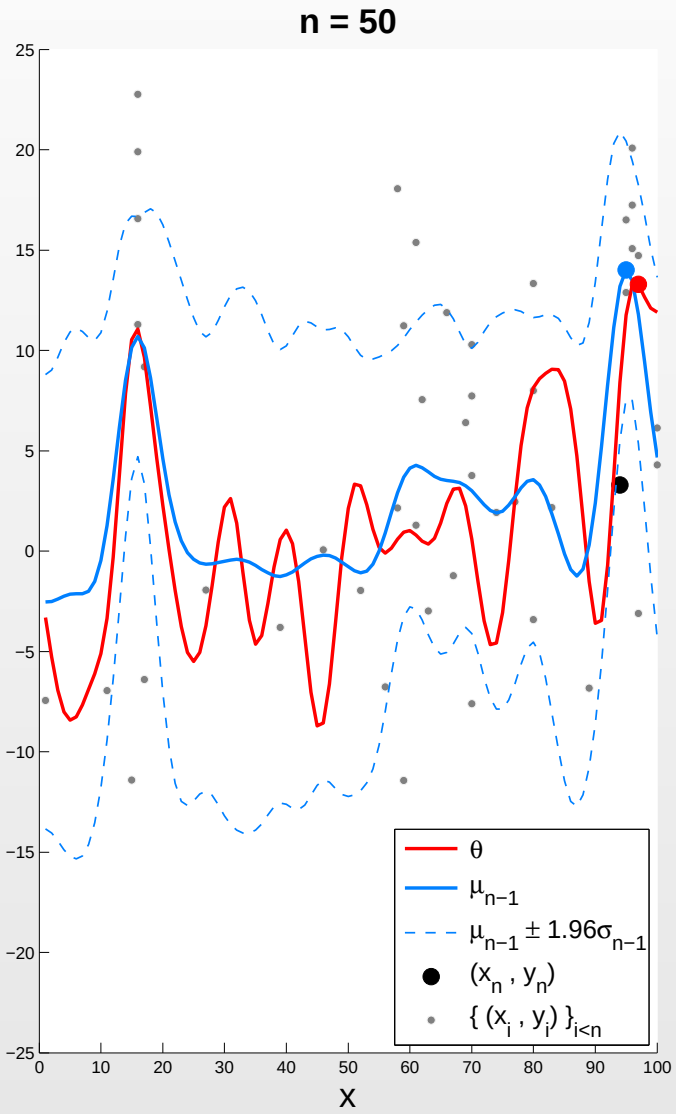


Illustration of the KG Algorithm



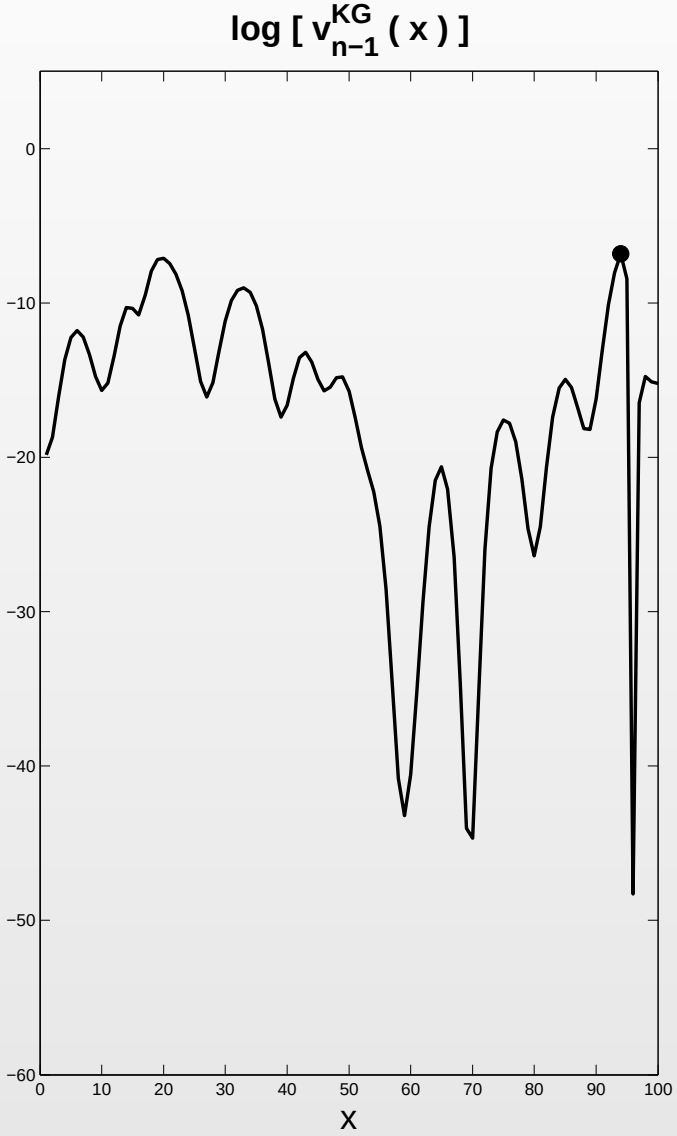
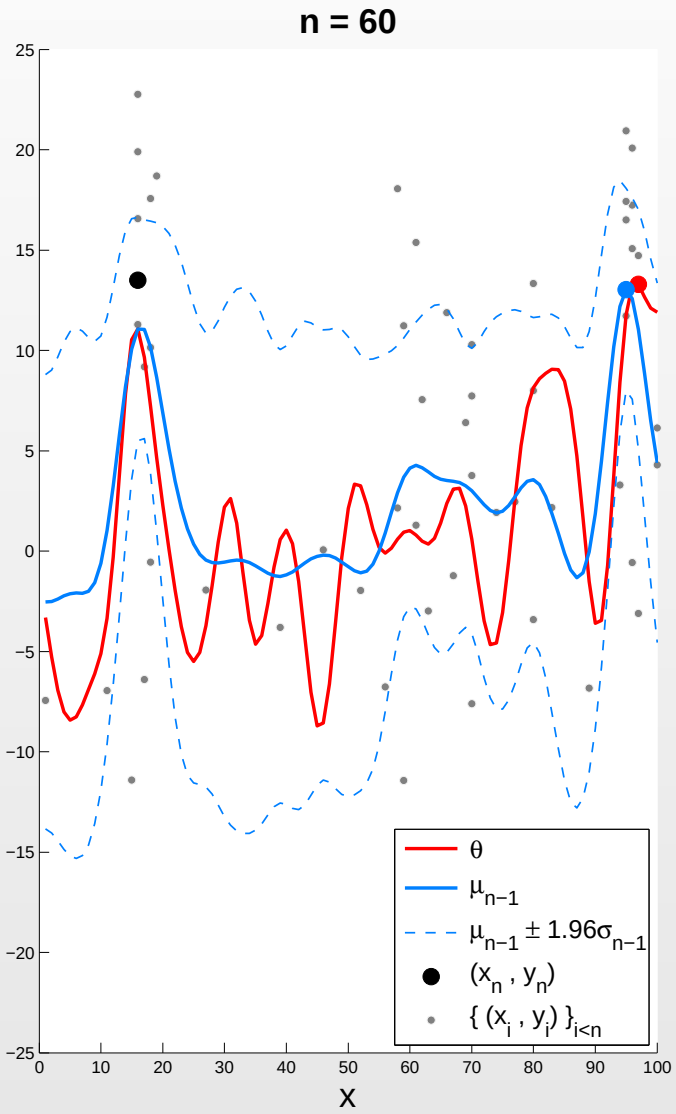


Illustration of the KG Algorithm



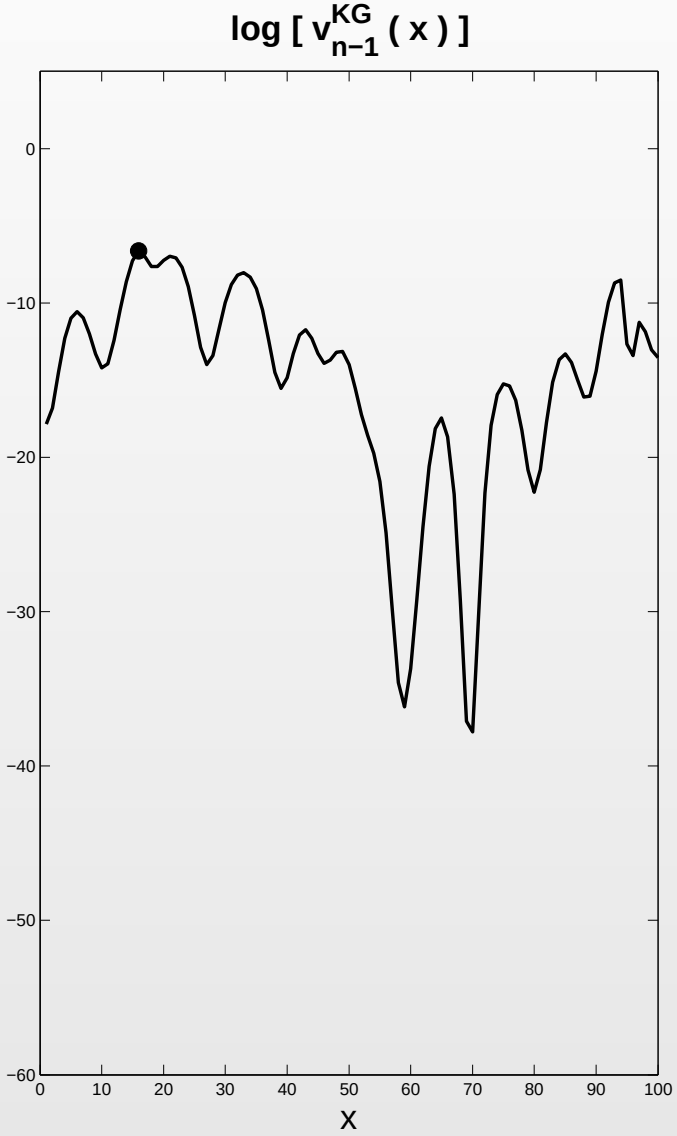
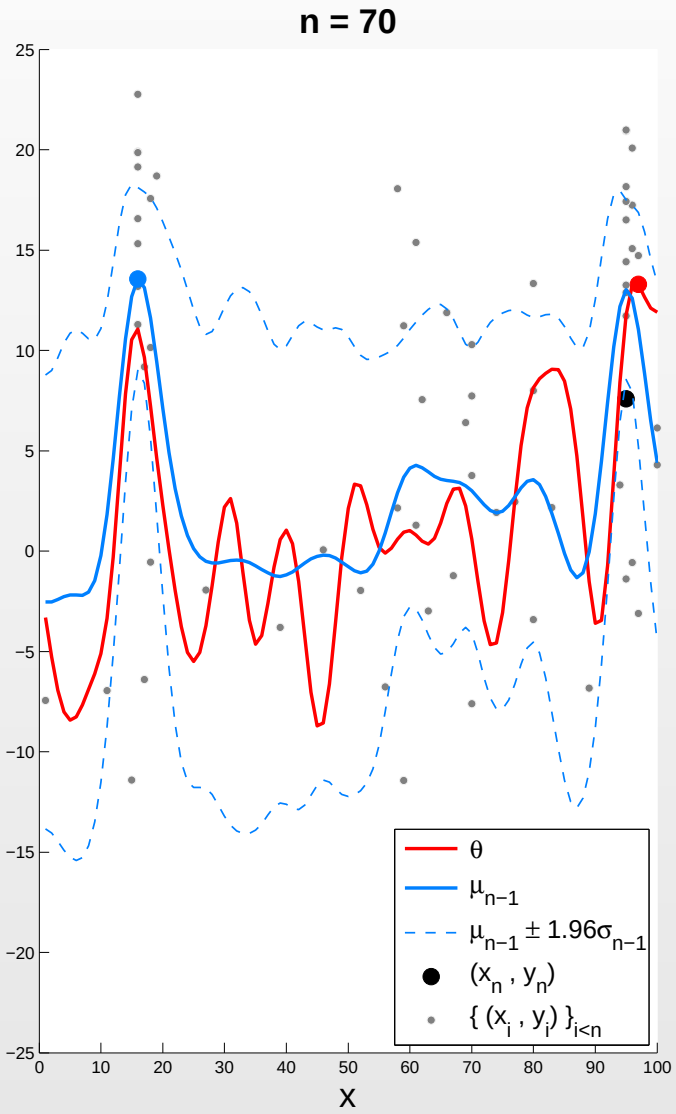


Illustration of the KG Algorithm



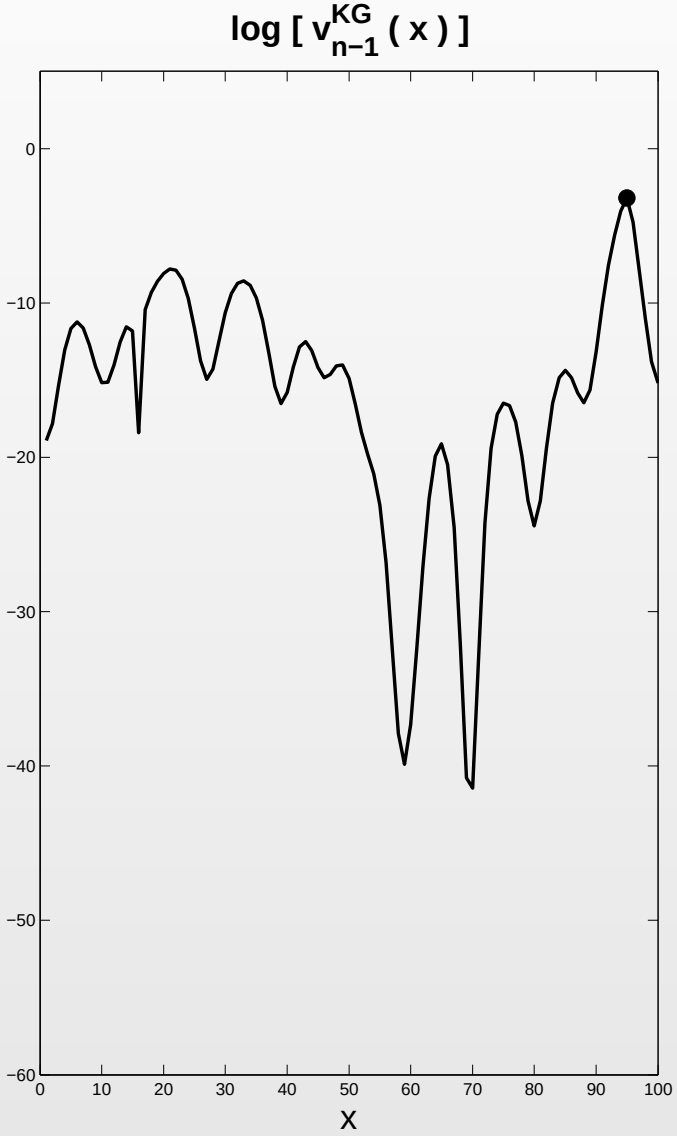
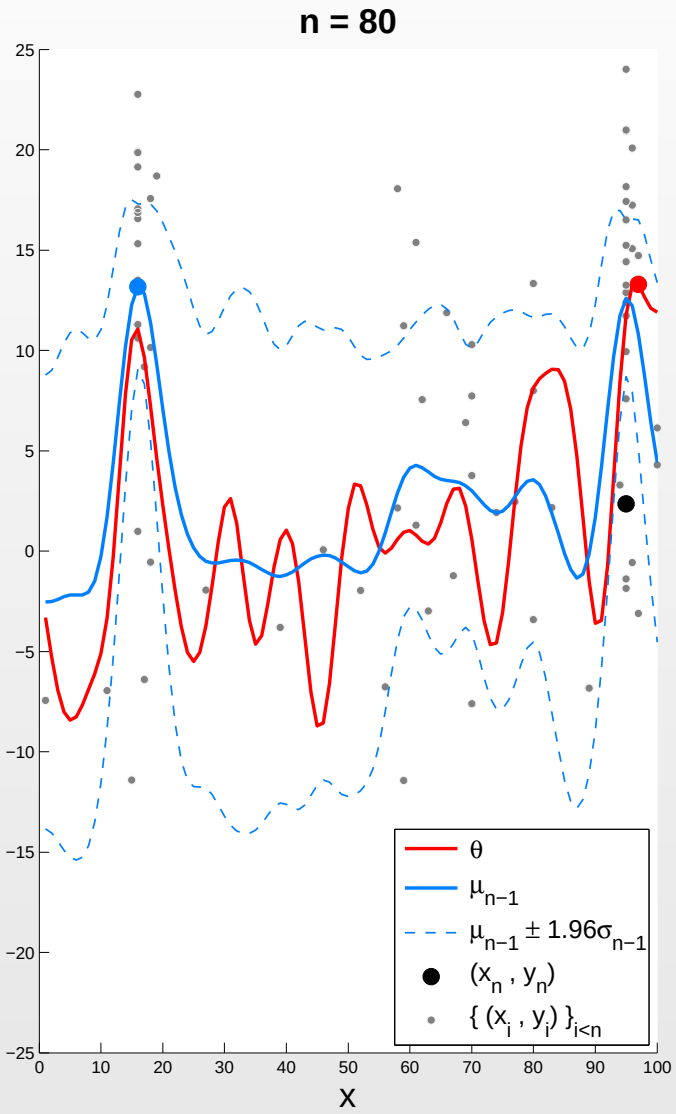


Illustration of the KG Algorithm



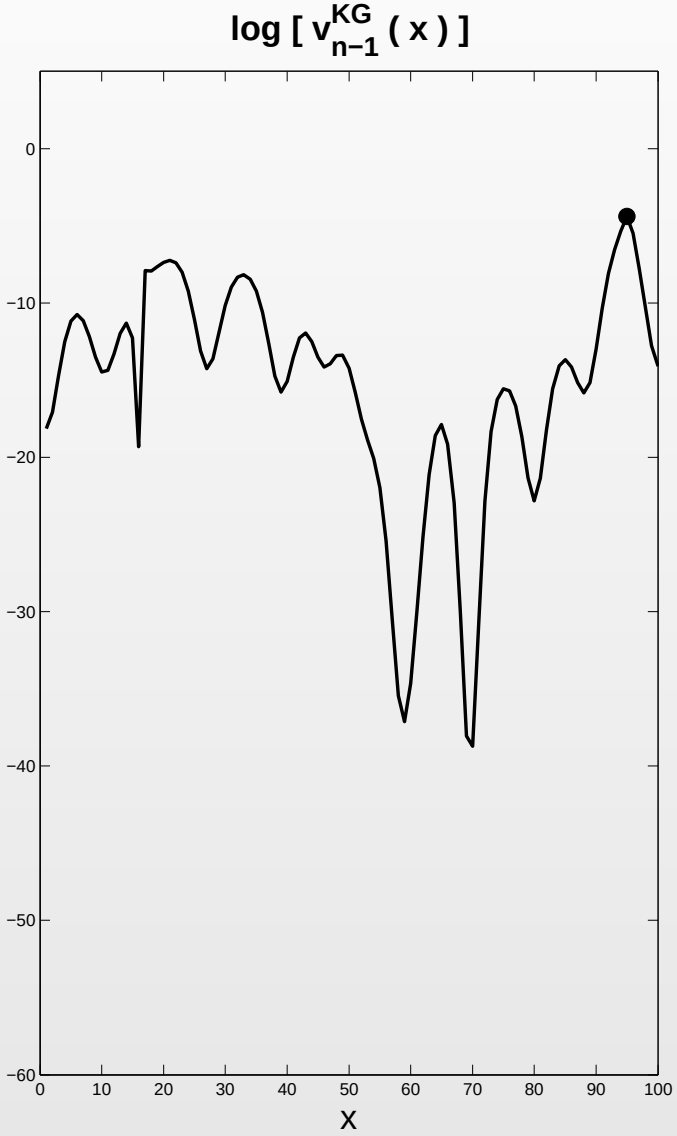
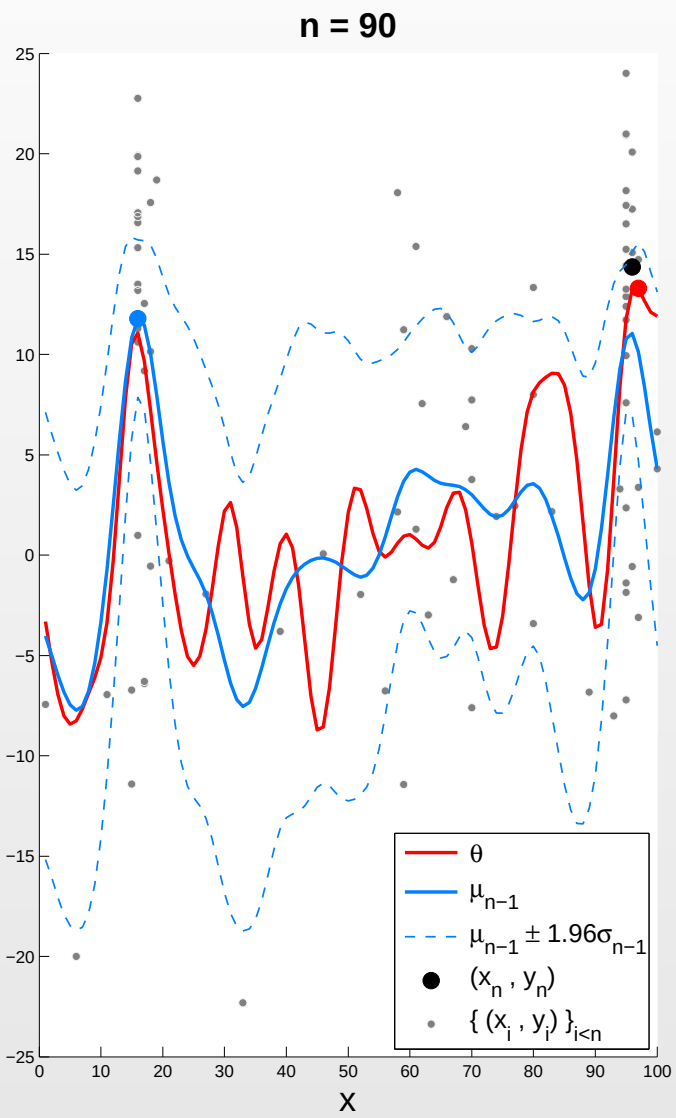


Illustration of the KG Algorithm



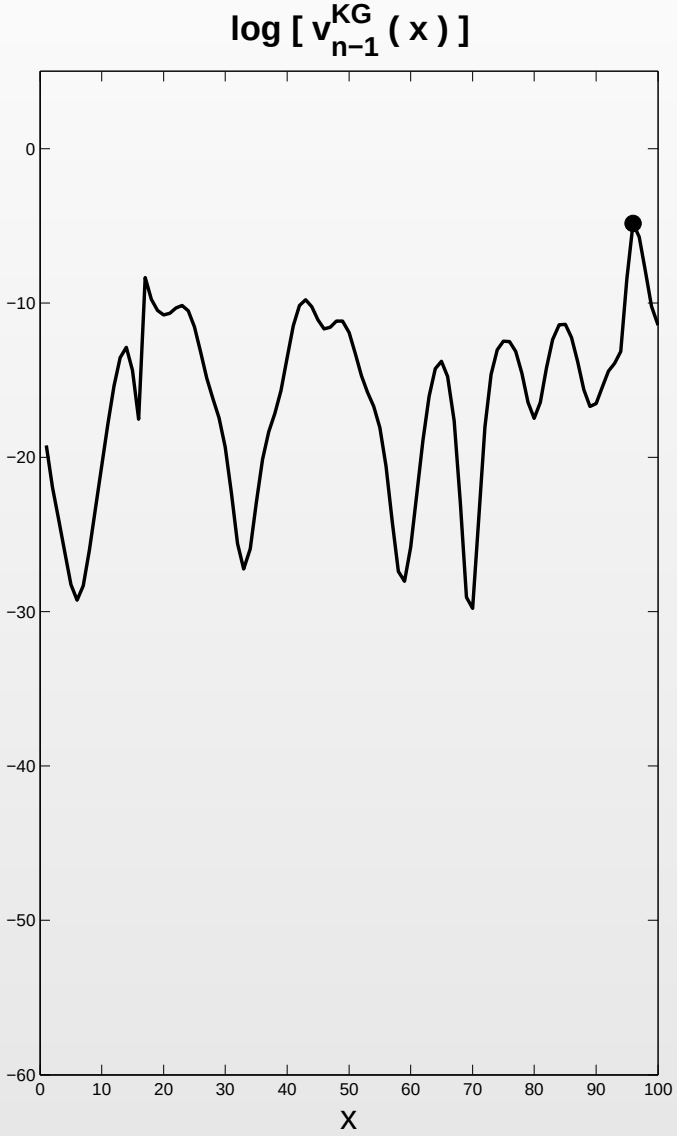
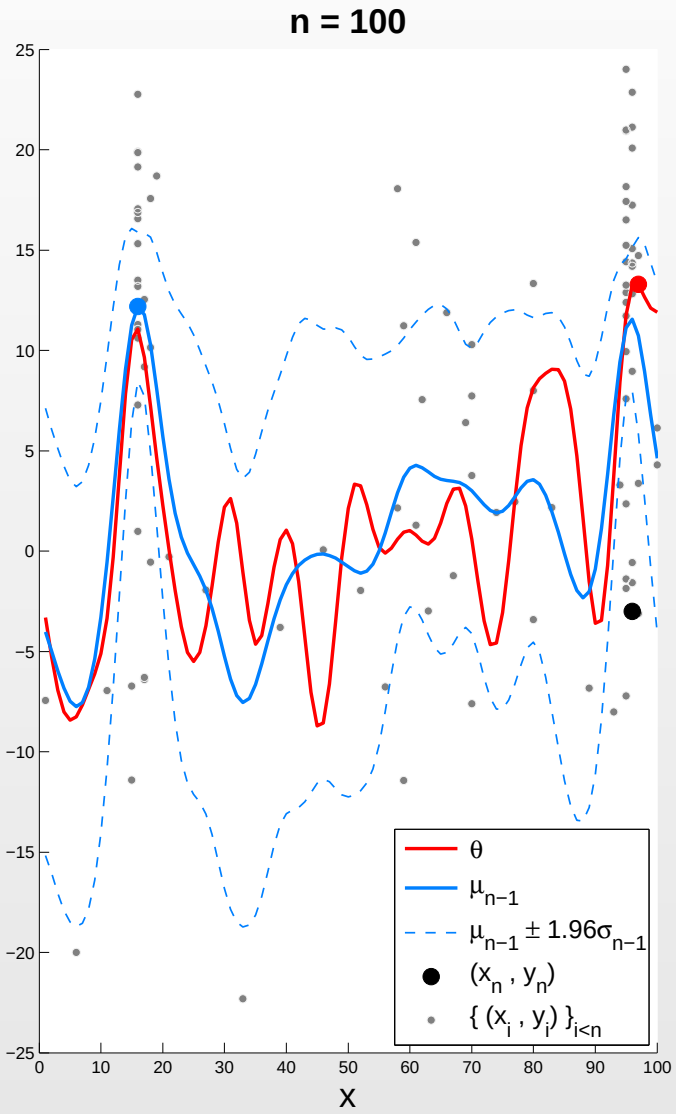


Illustration of the KG Algorithm



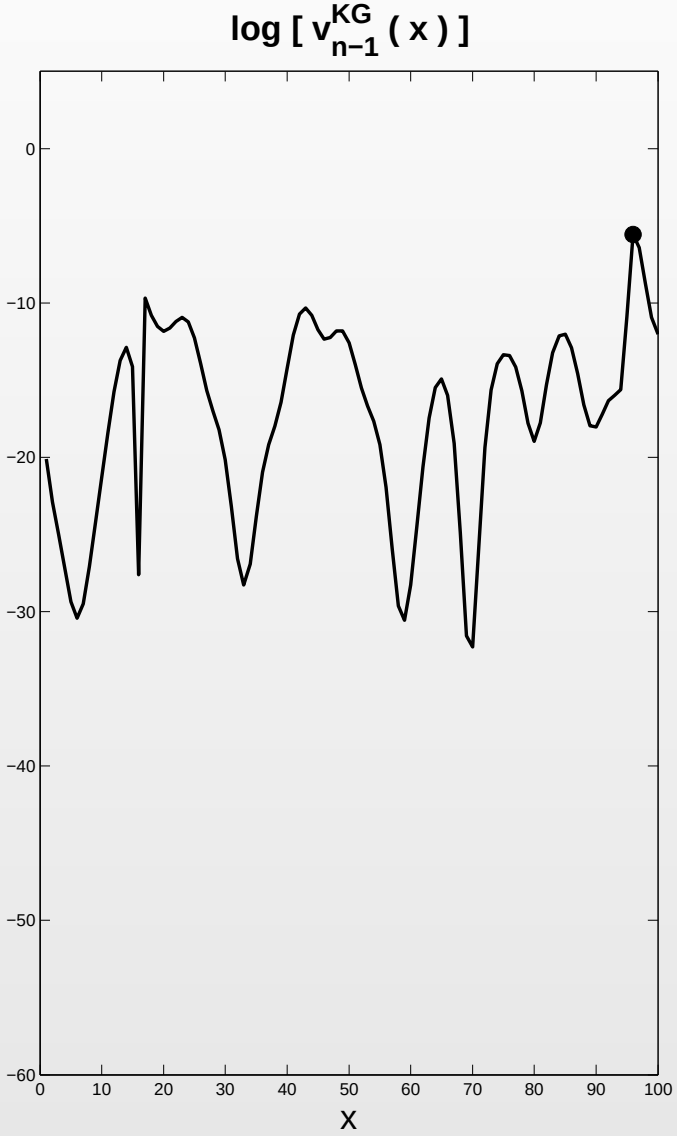
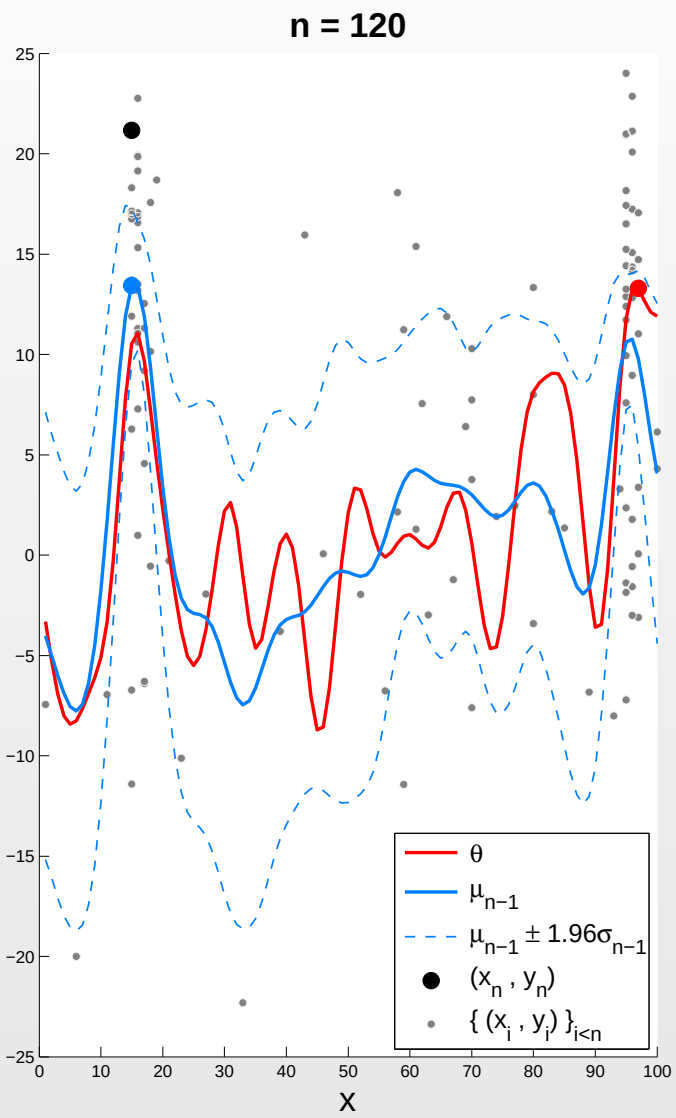


Illustration of the KG Algorithm



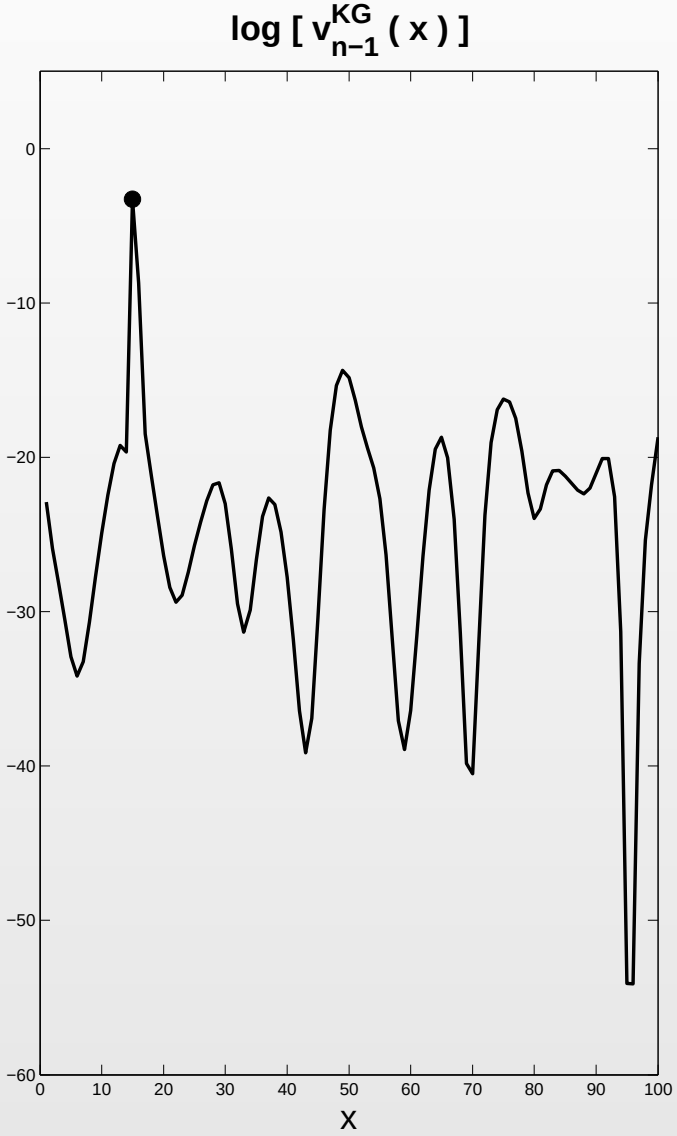
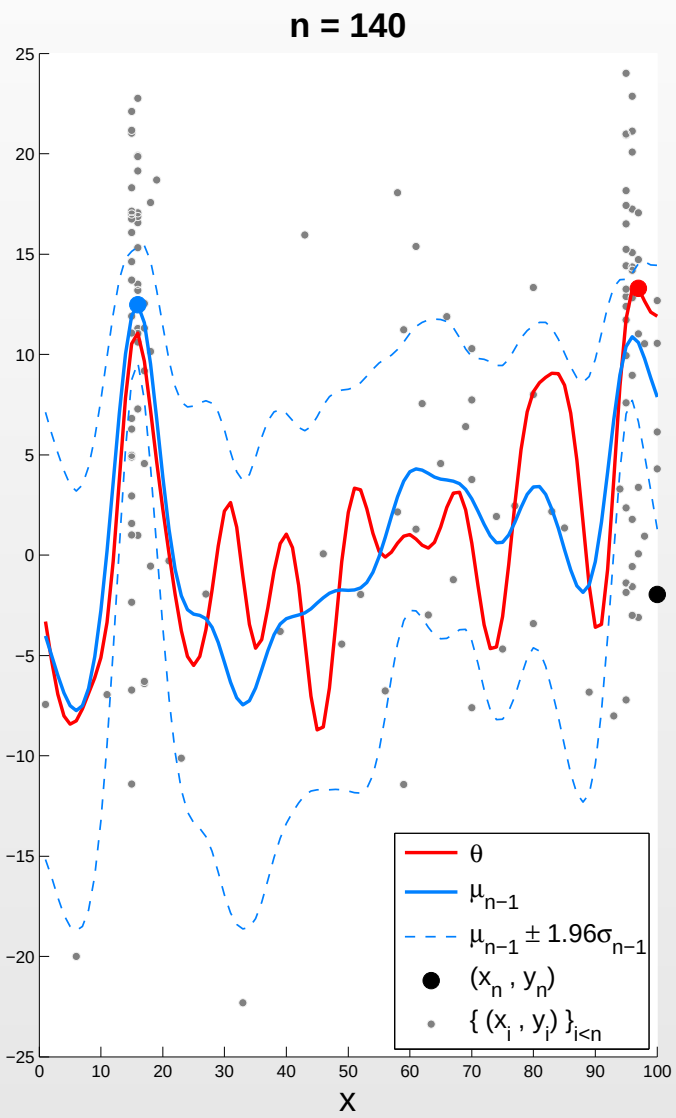


Illustration of the KG Algorithm



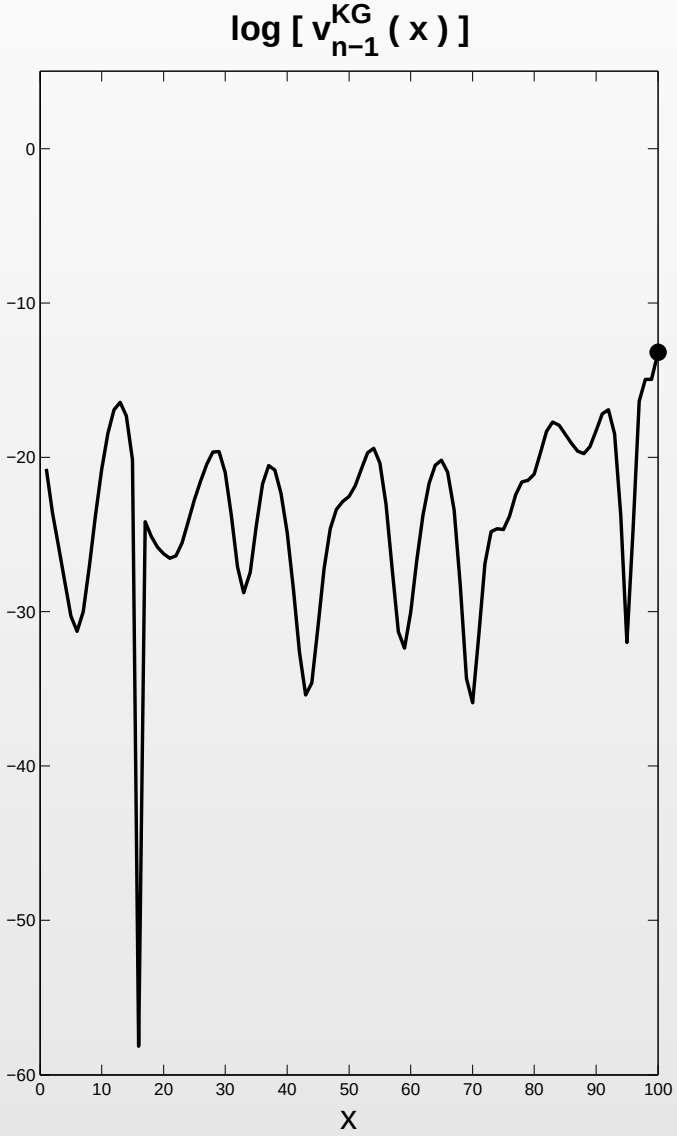
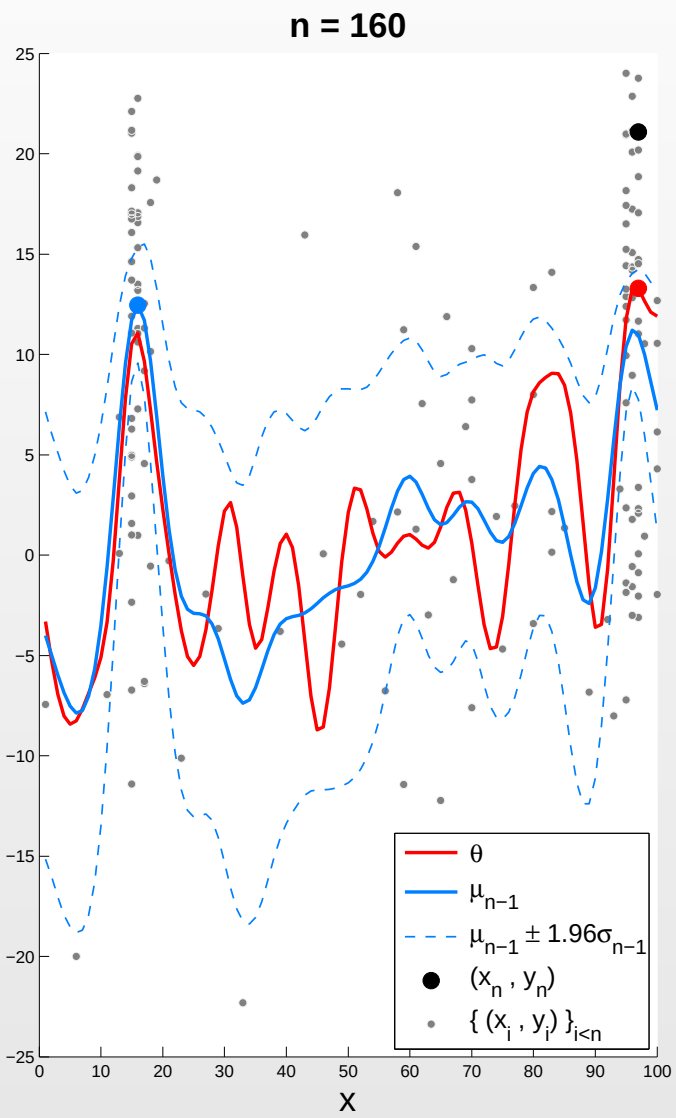


Illustration of the KG Algorithm



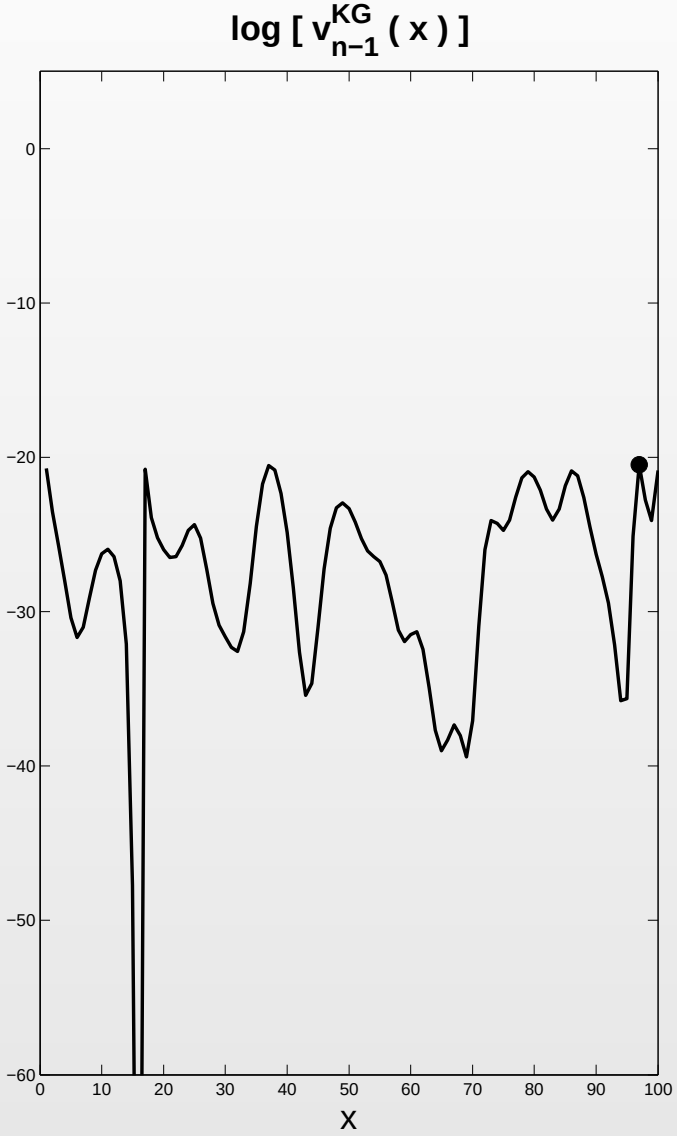
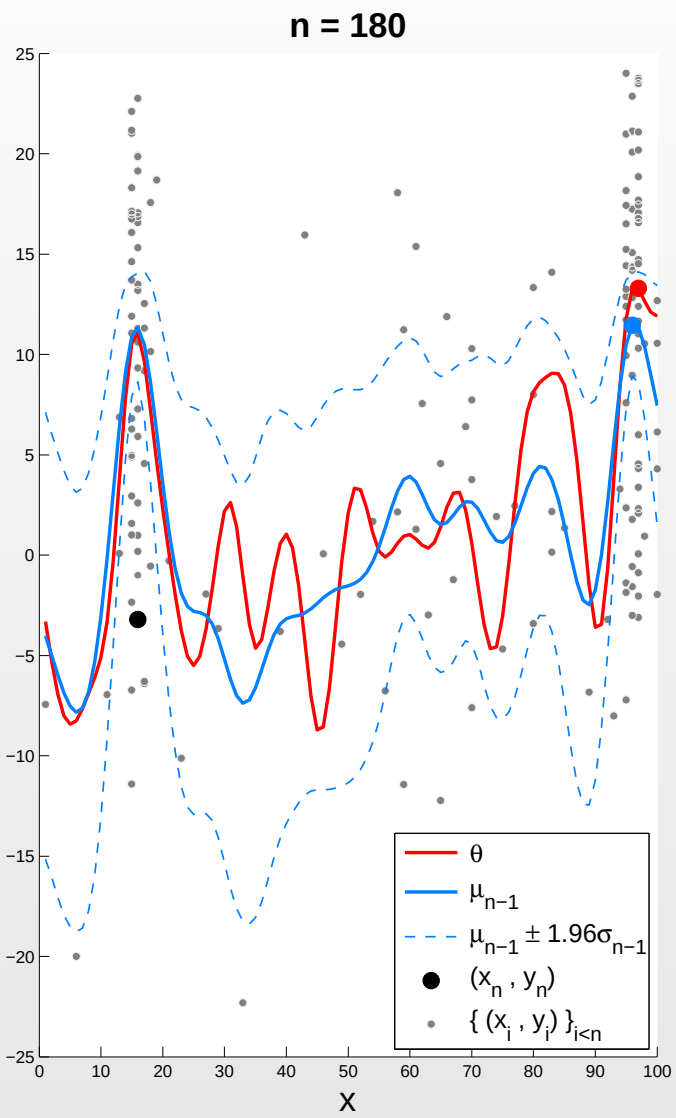


Illustration of the KG Algorithm



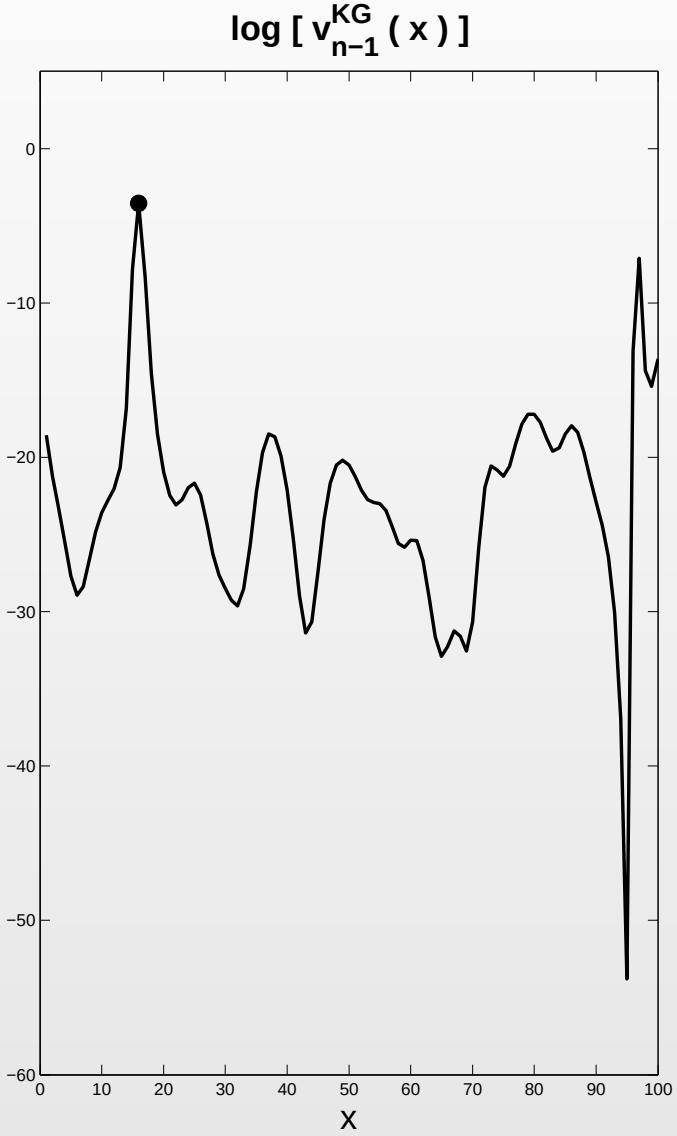
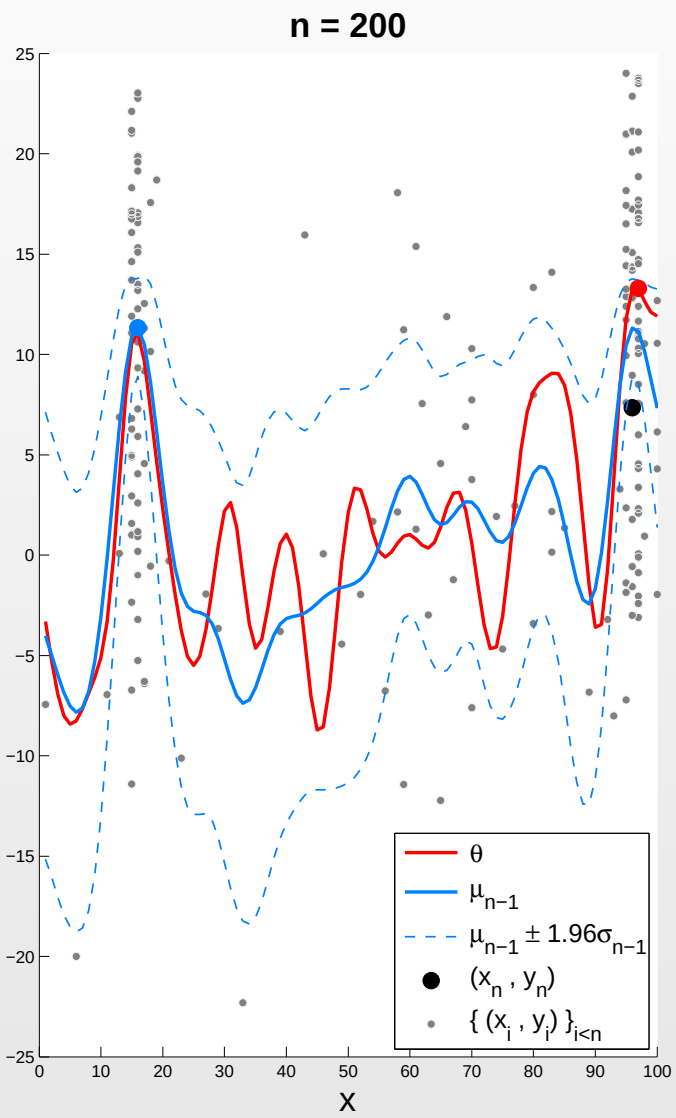


Illustration of the KG Algorithm



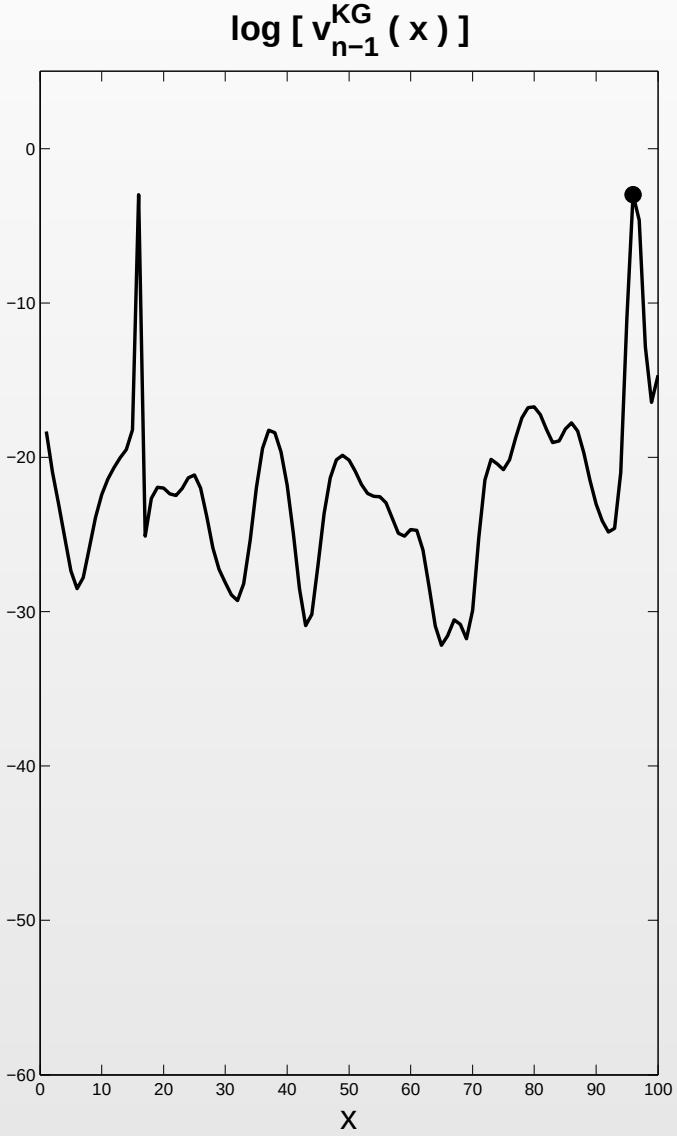
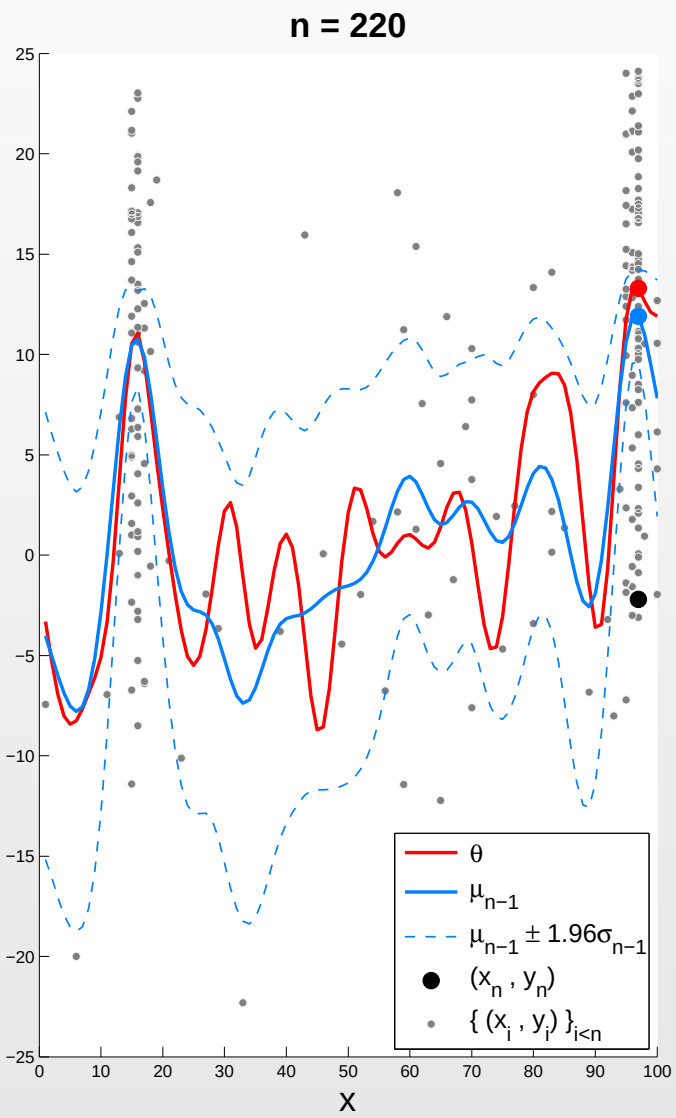


Illustration of the KG Algorithm



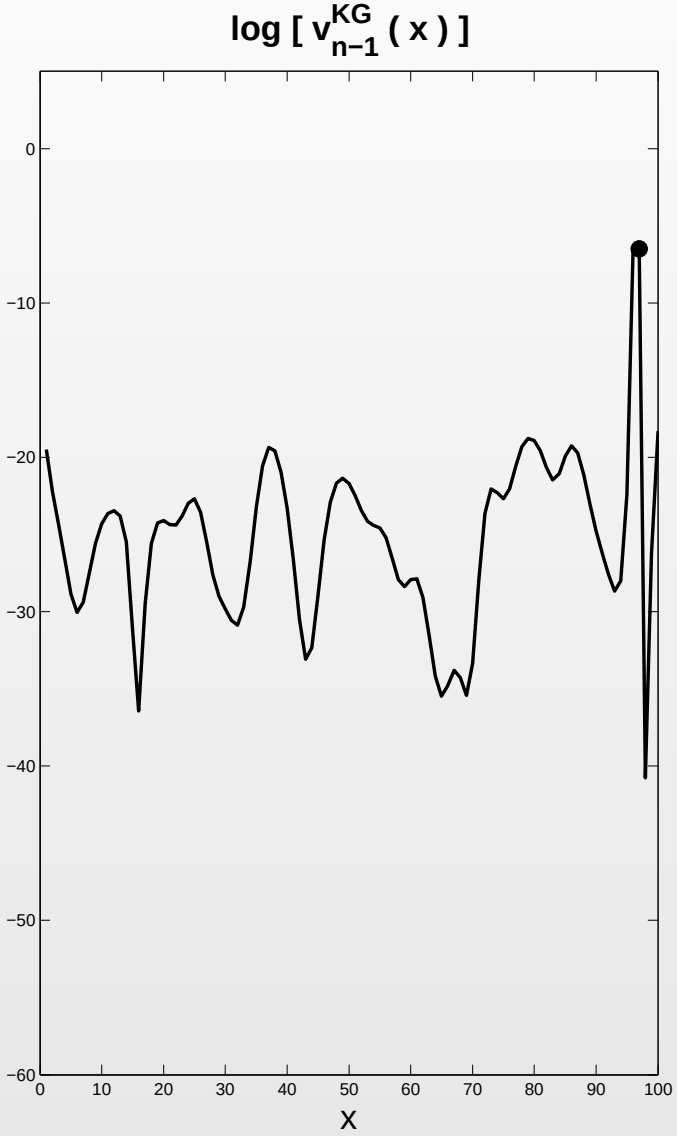
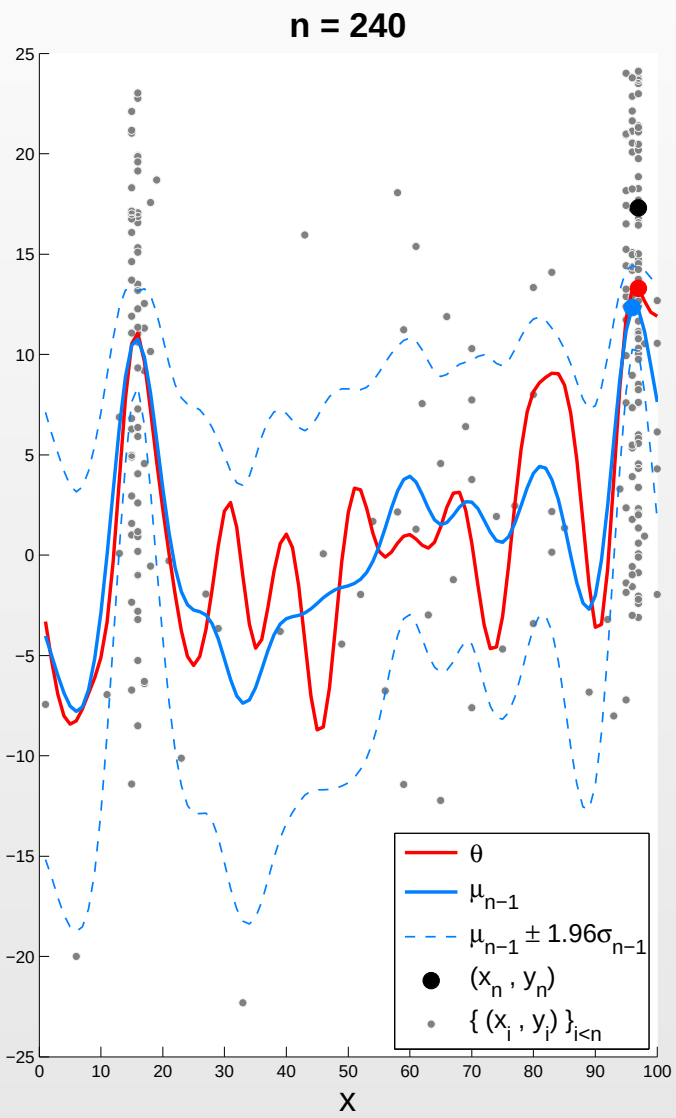


Illustration of the KG Algorithm



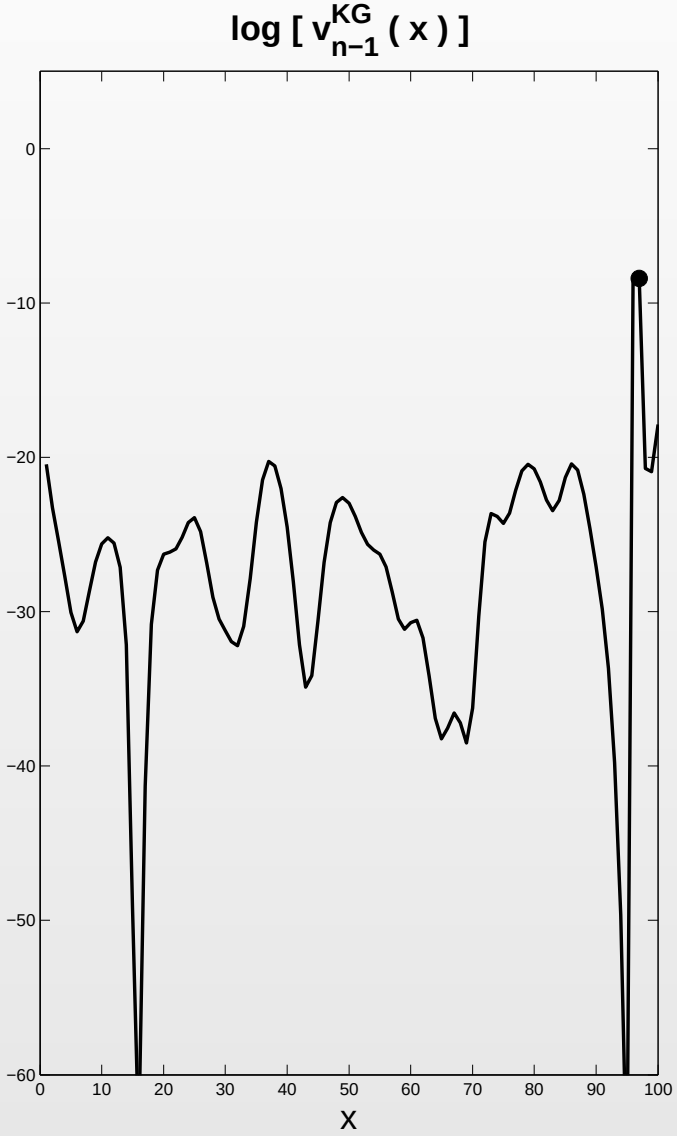
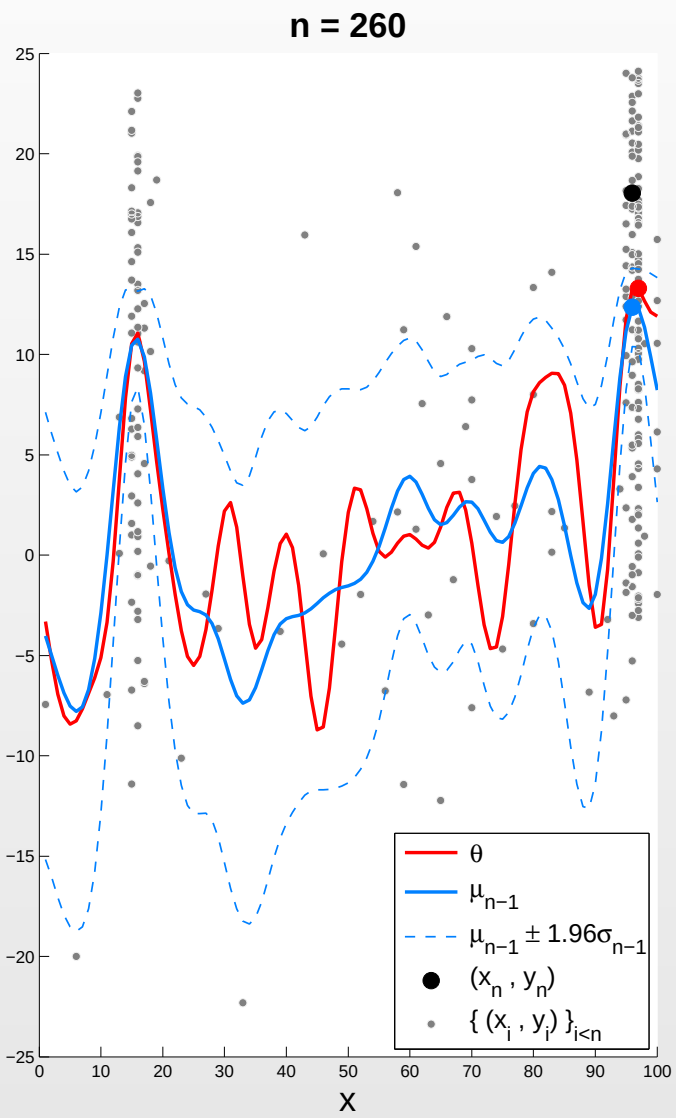


Illustration of the KG Algorithm



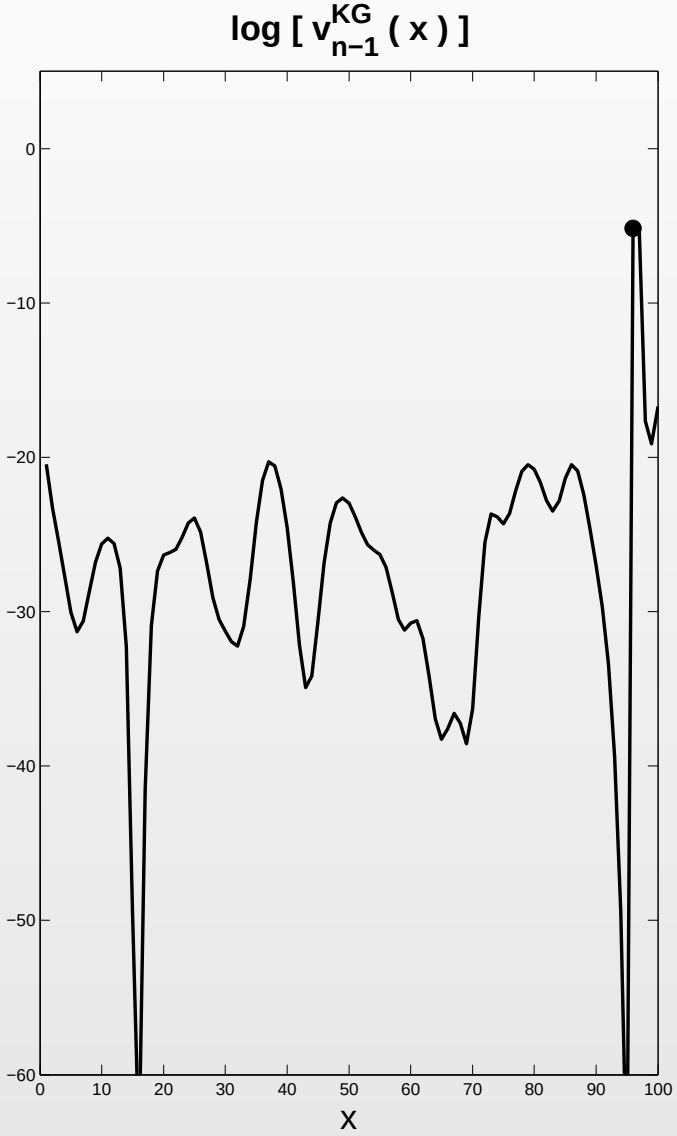
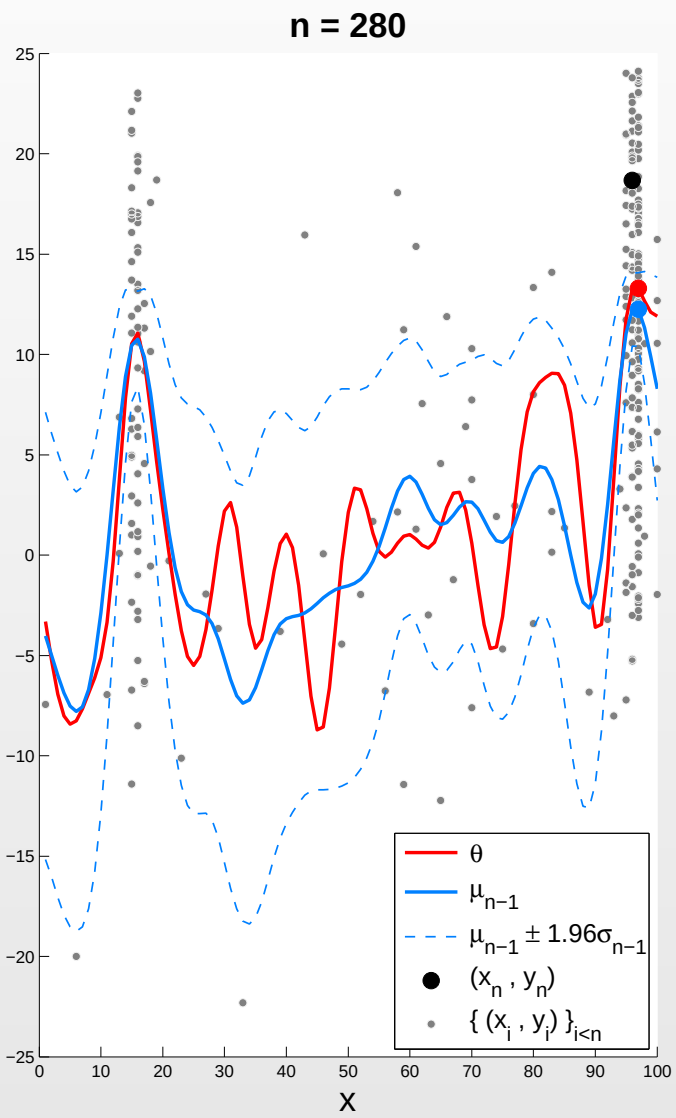


Illustration of the KG Algorithm



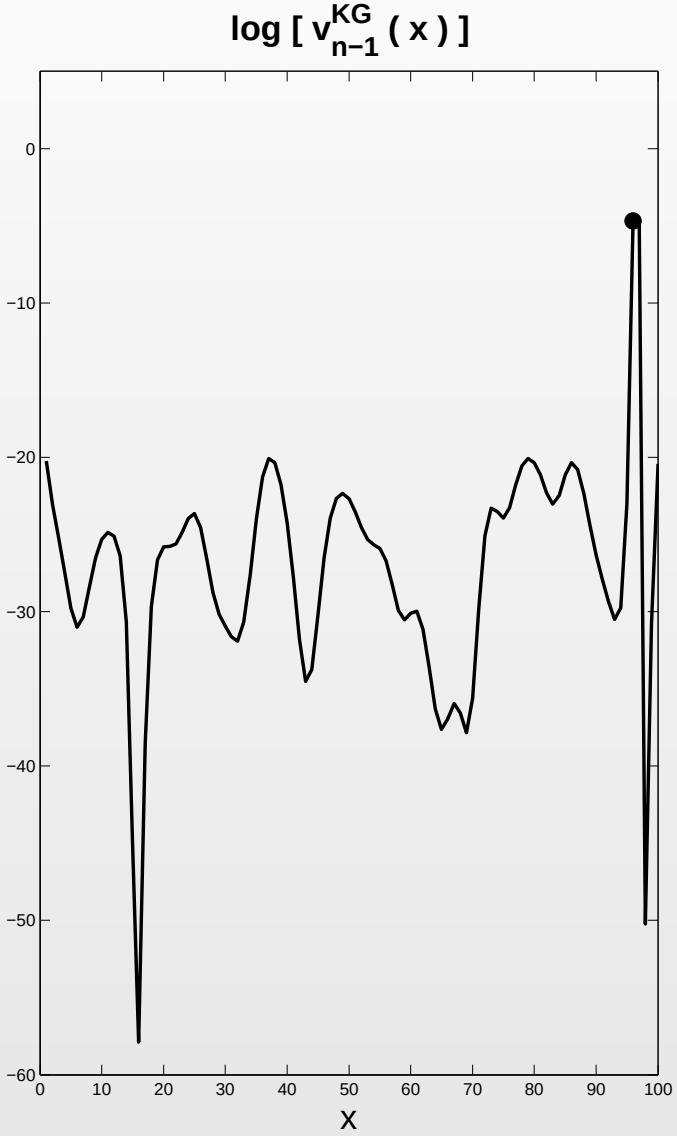
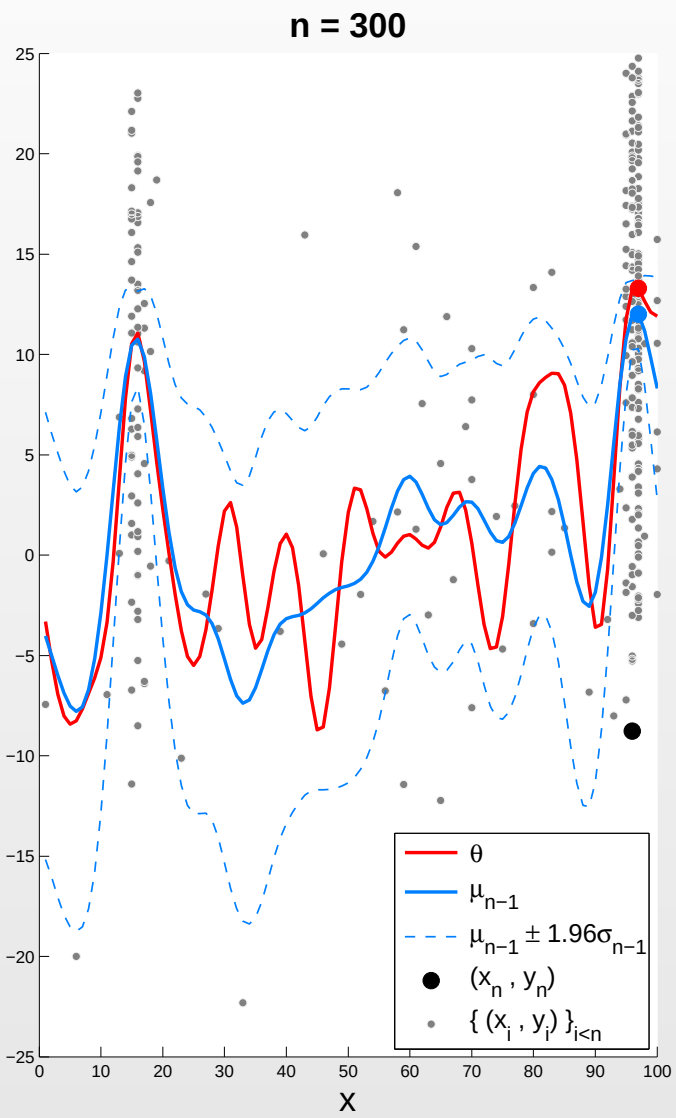


Illustration of the KG Algorithm



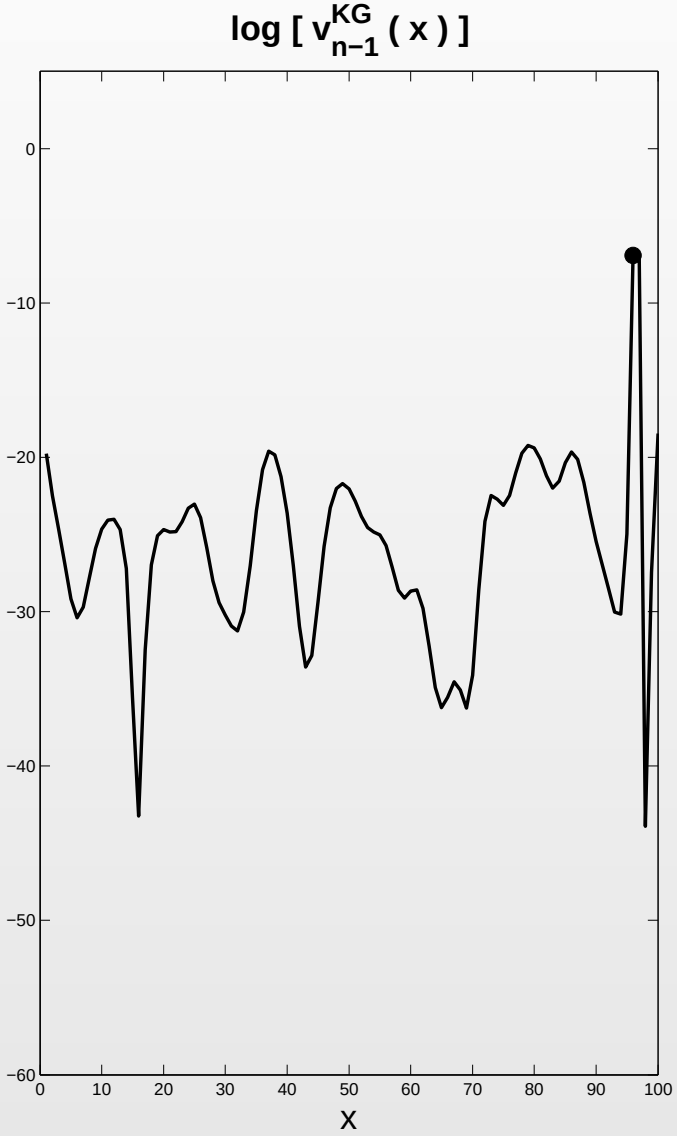
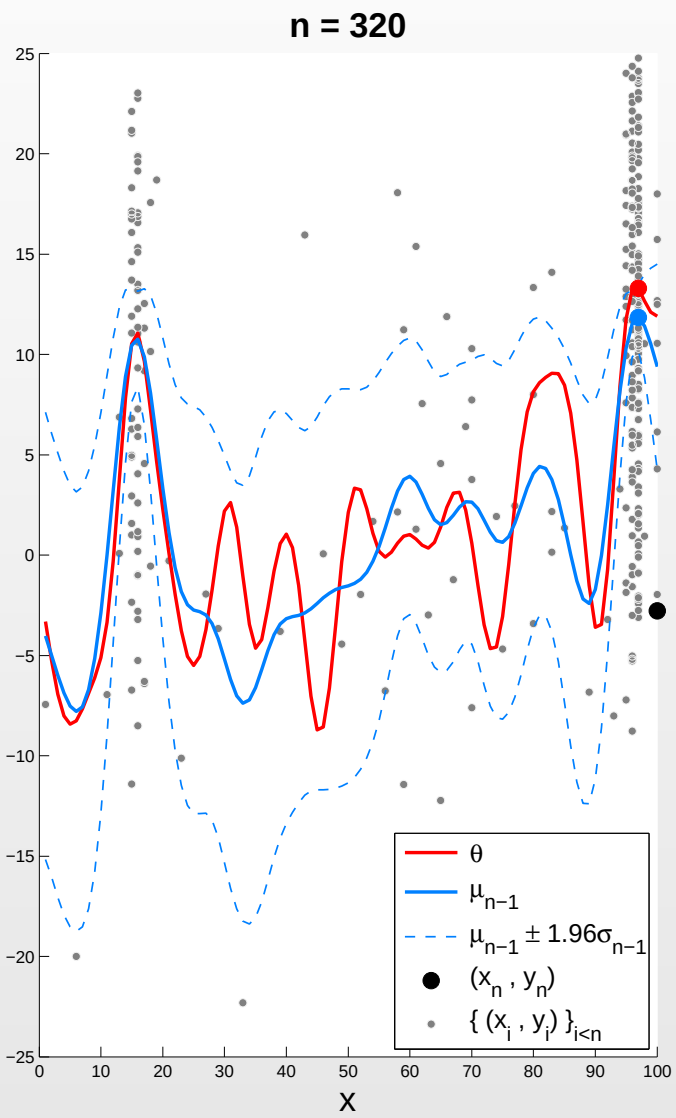


Illustration of the KG Algorithm



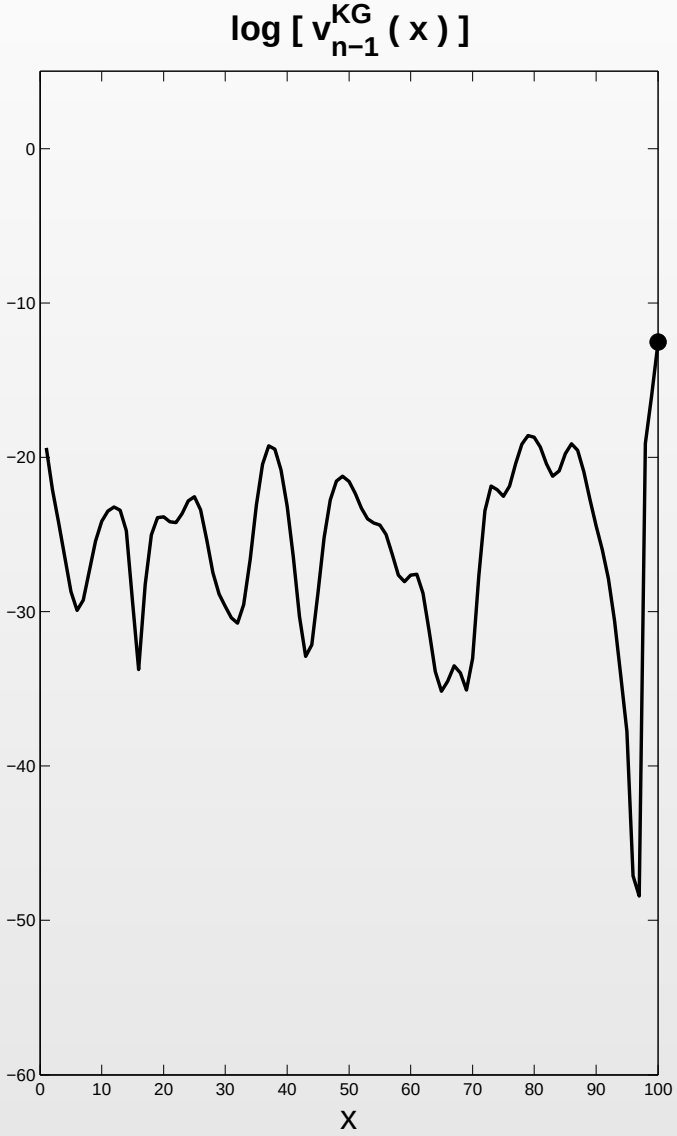
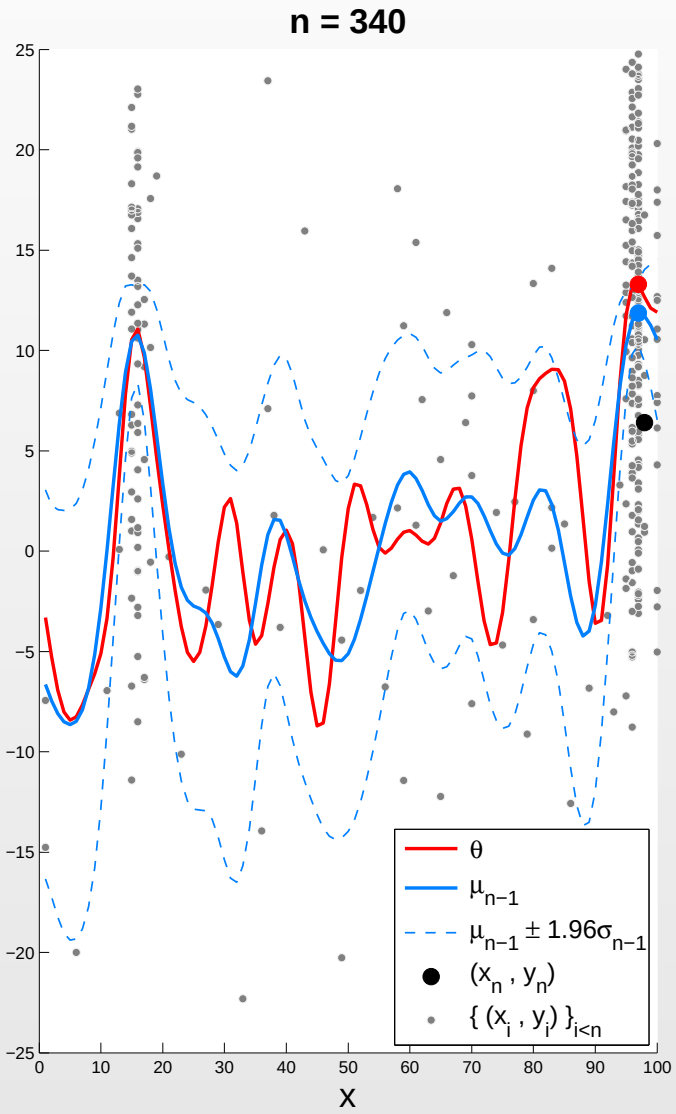


Illustration of the KG Algorithm



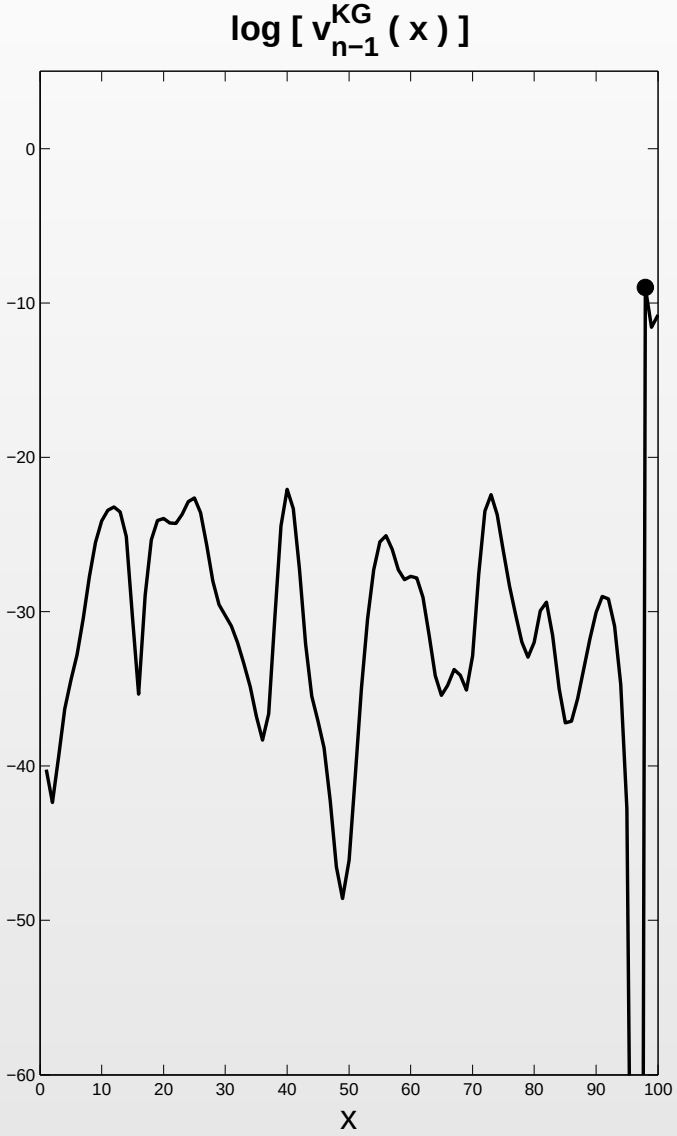
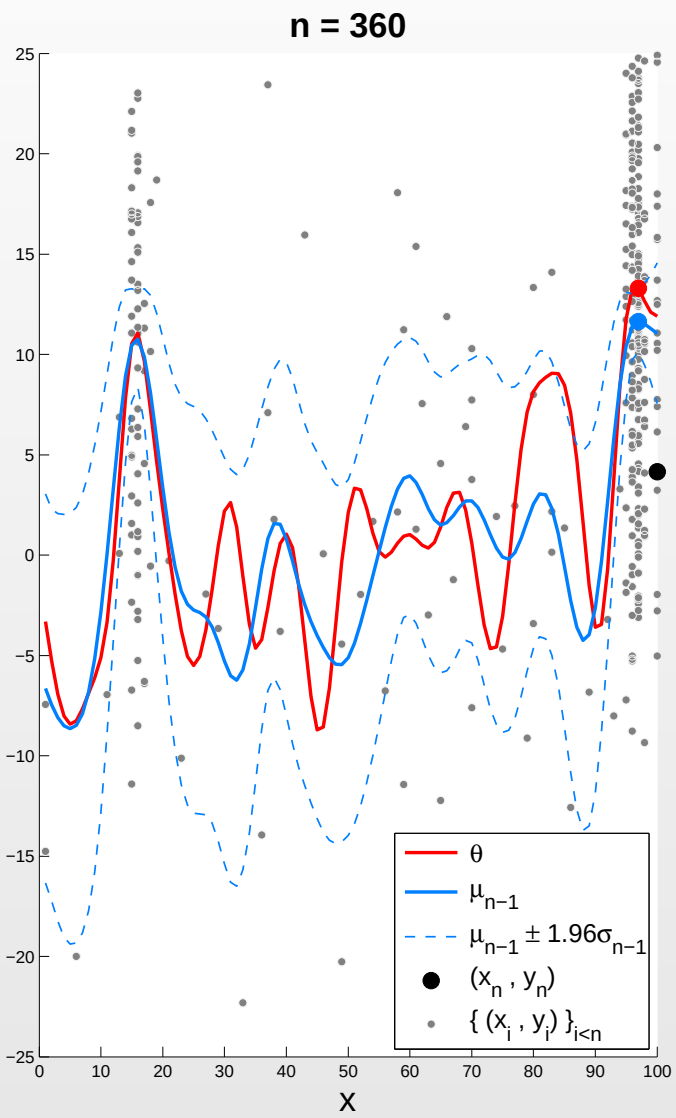


Illustration of the KG Algorithm



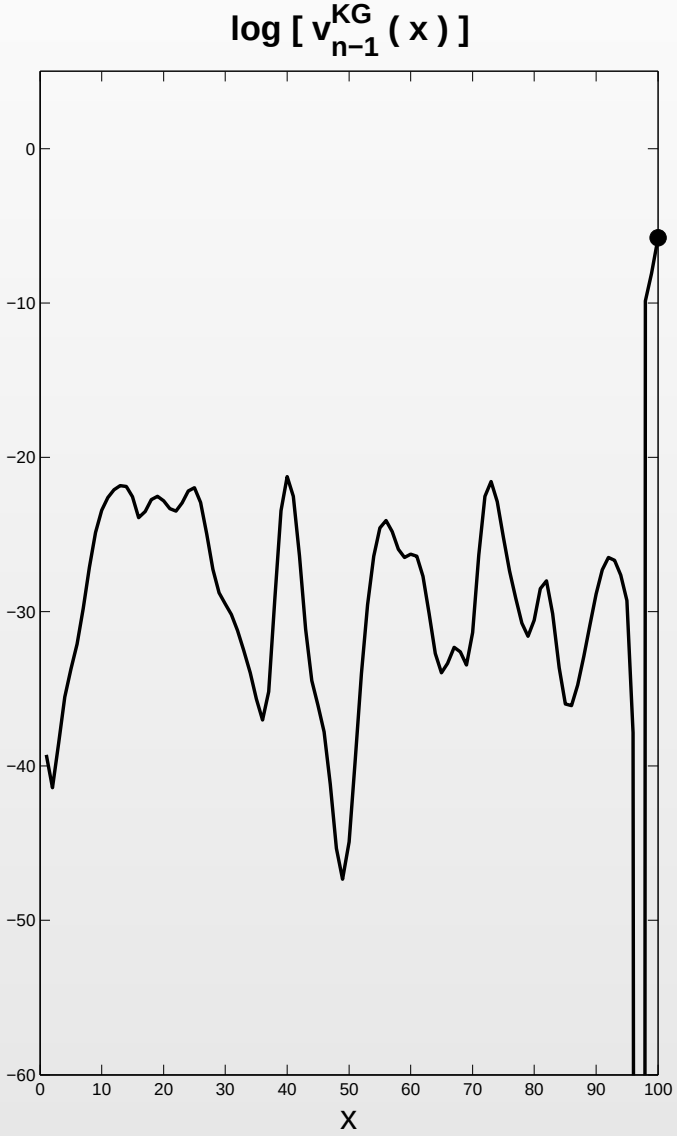
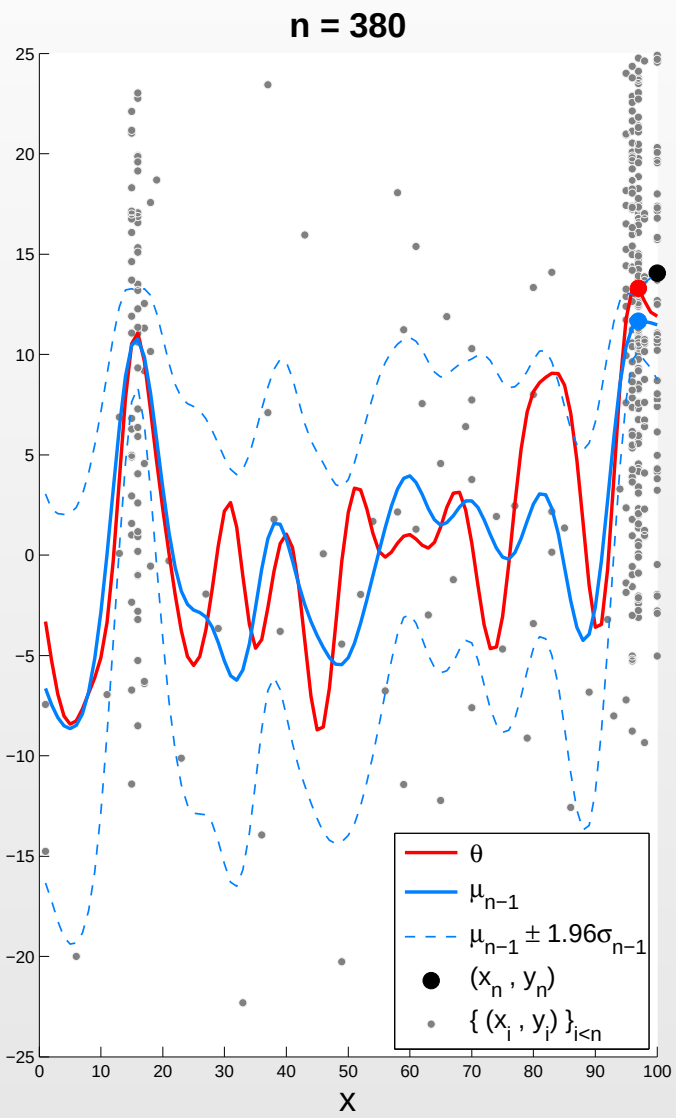


Illustration of the KG Algorithm



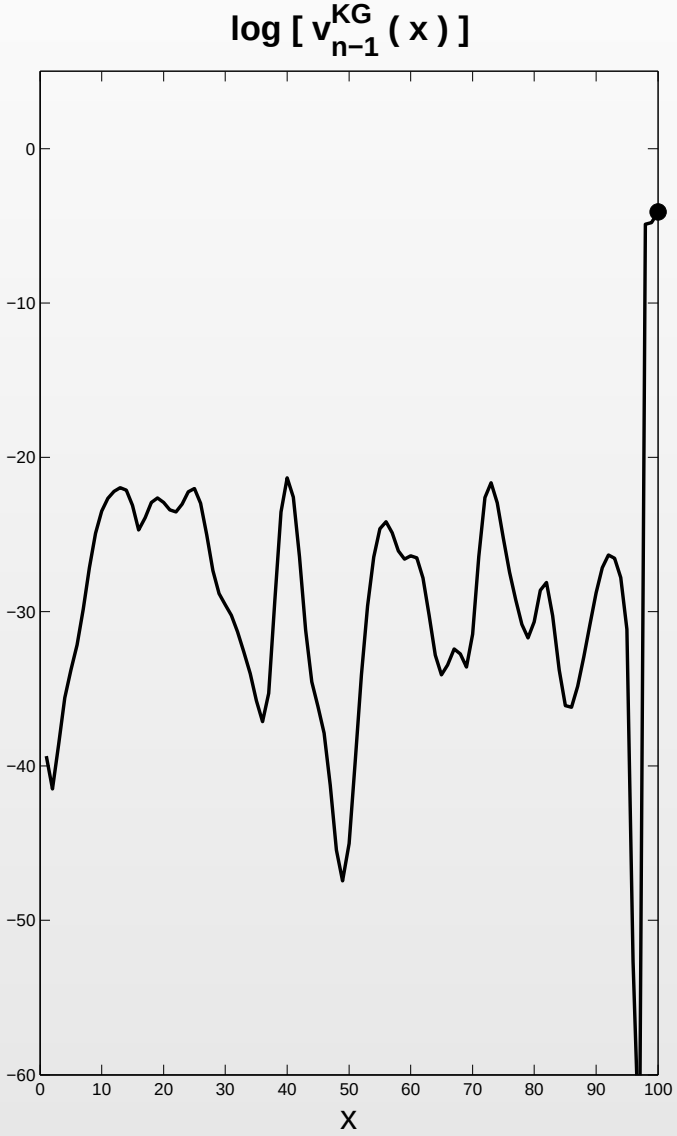
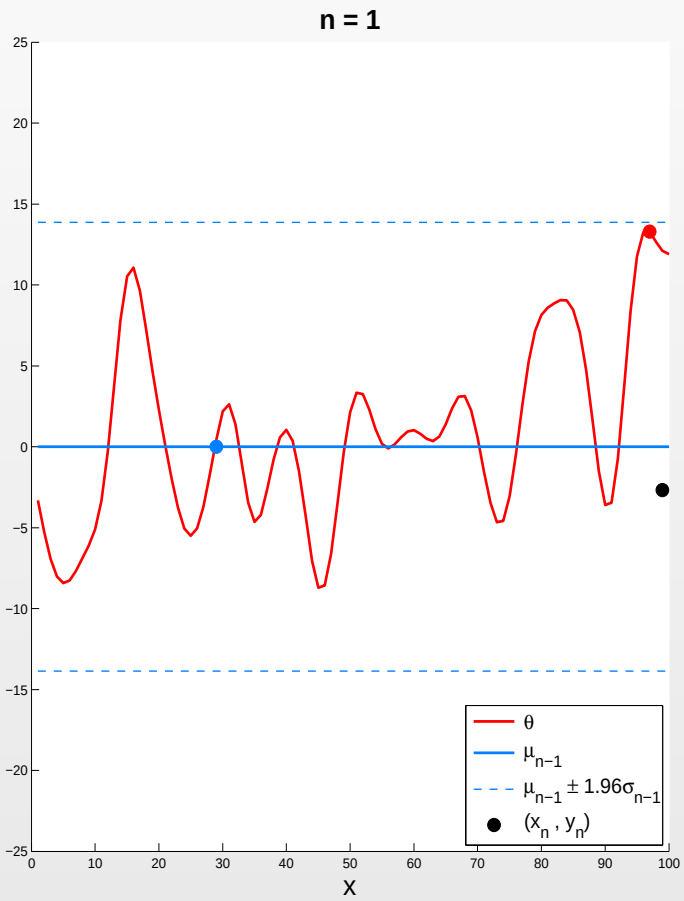


Illustration of the KG^2 Algorithm



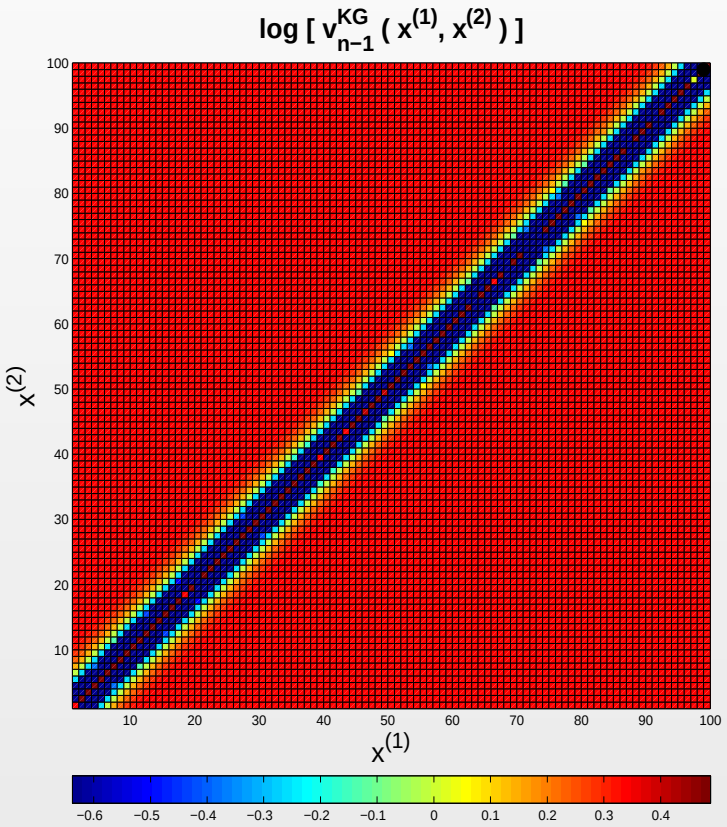
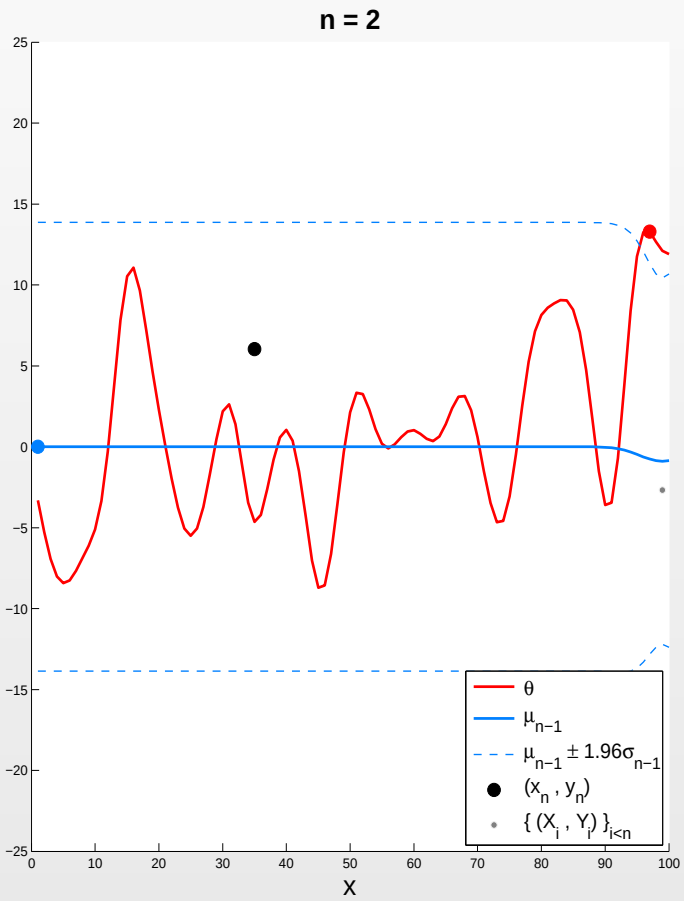


Illustration of the KG^2 Algorithm



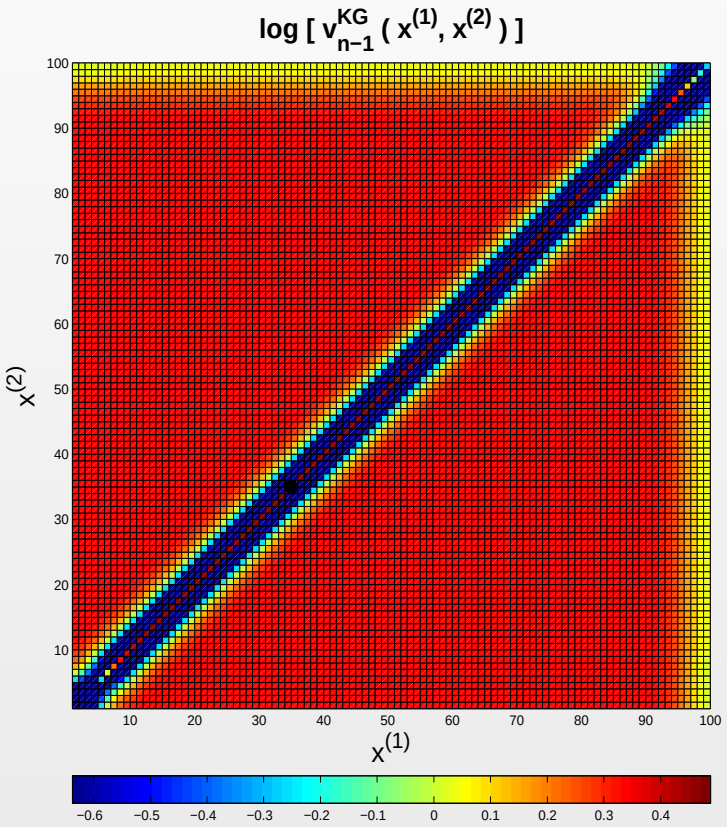
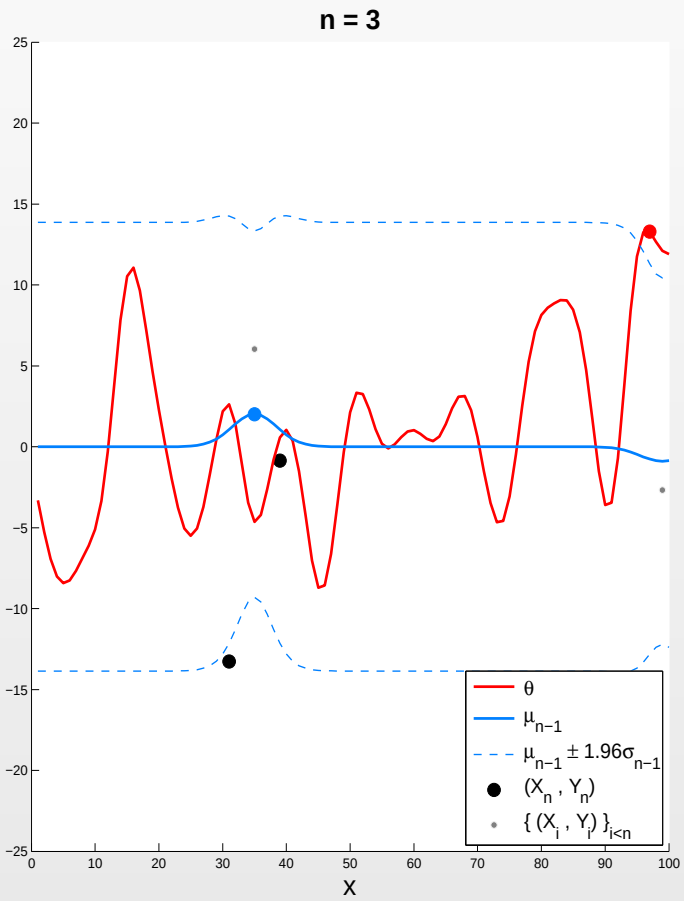


Illustration of the KG^2 Algorithm



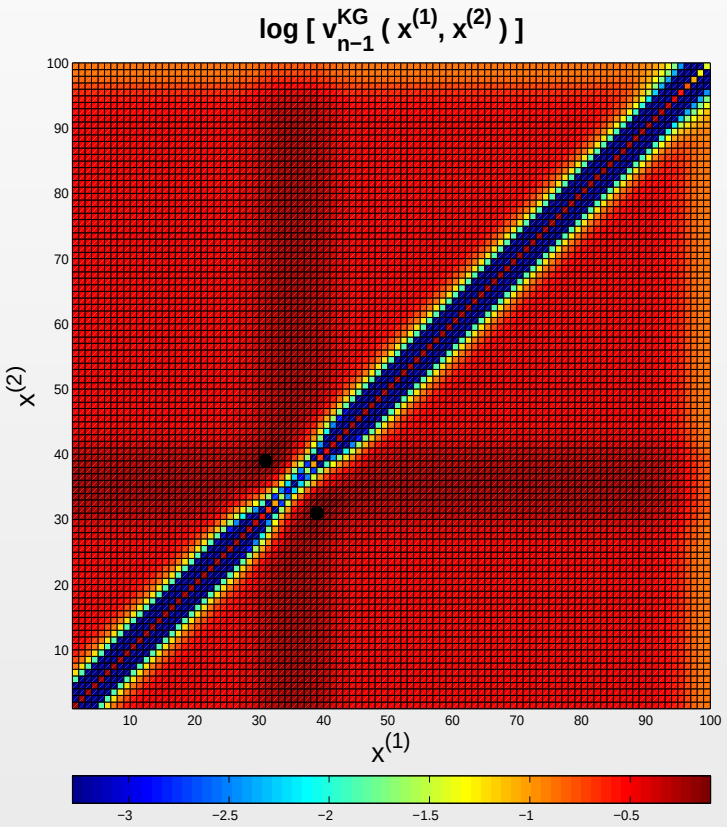
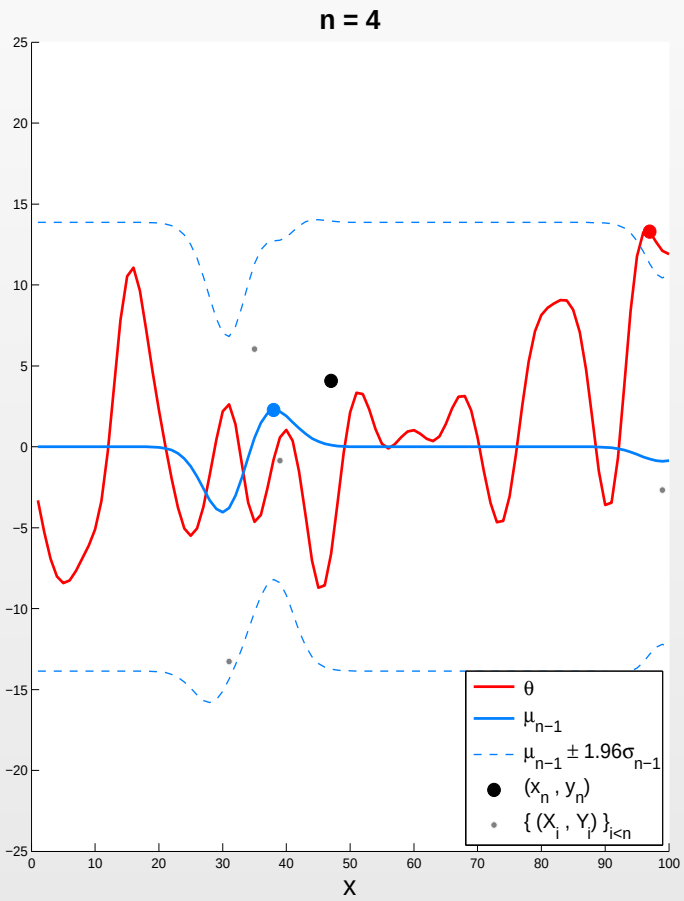


Illustration of the KG^2 Algorithm



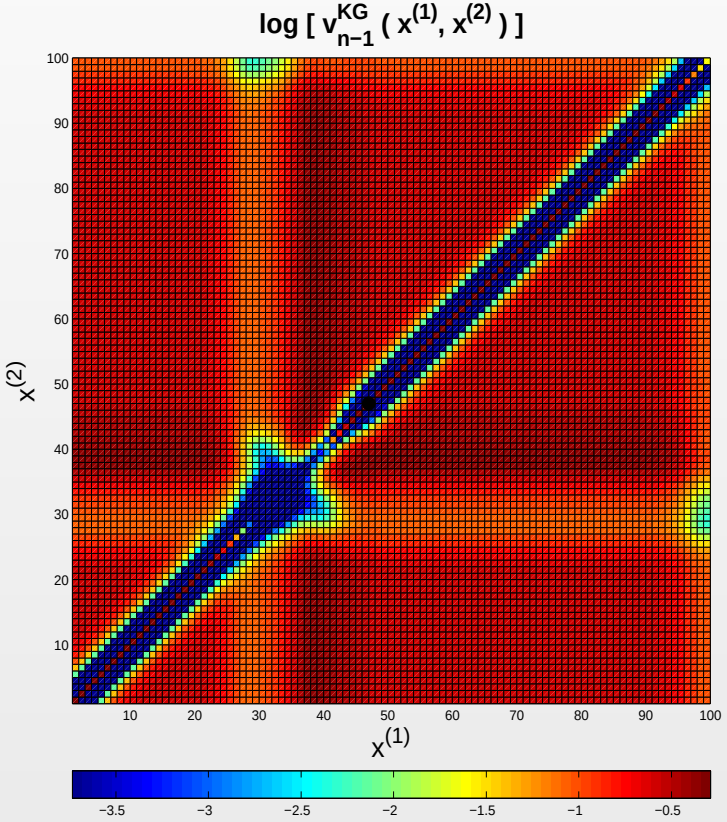
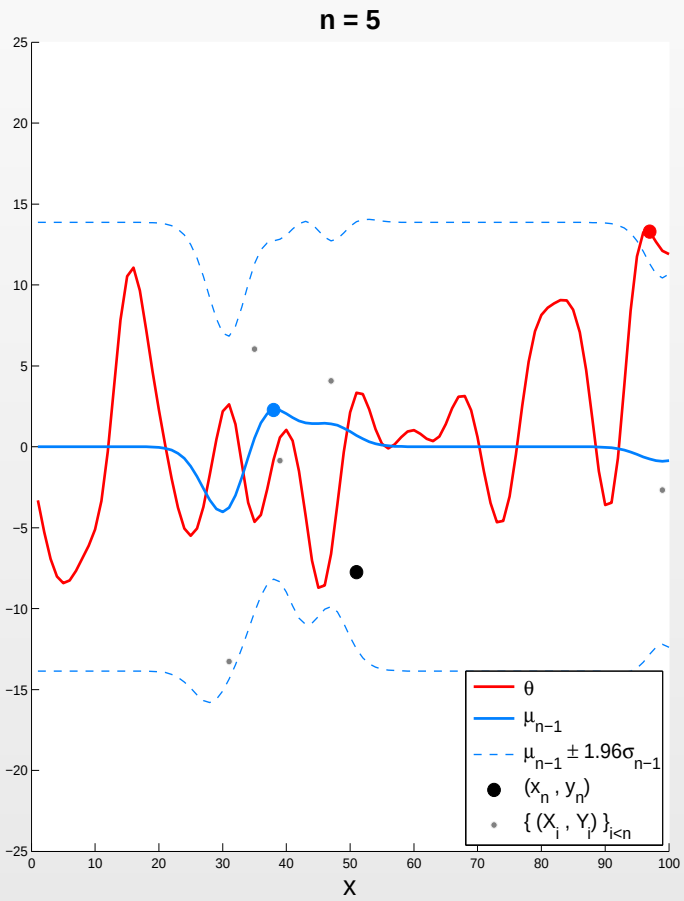


Illustration of the KG^2 Algorithm



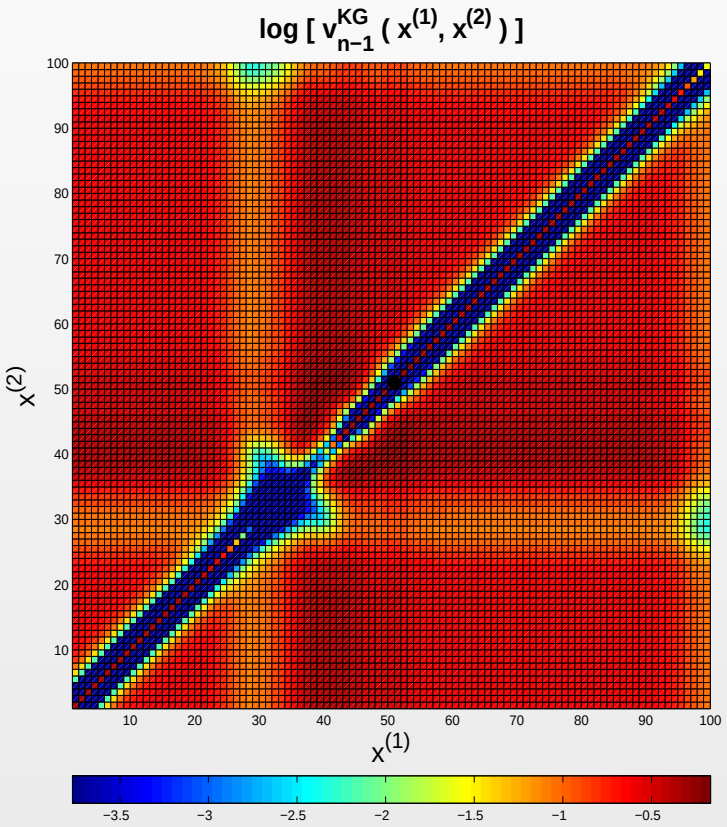
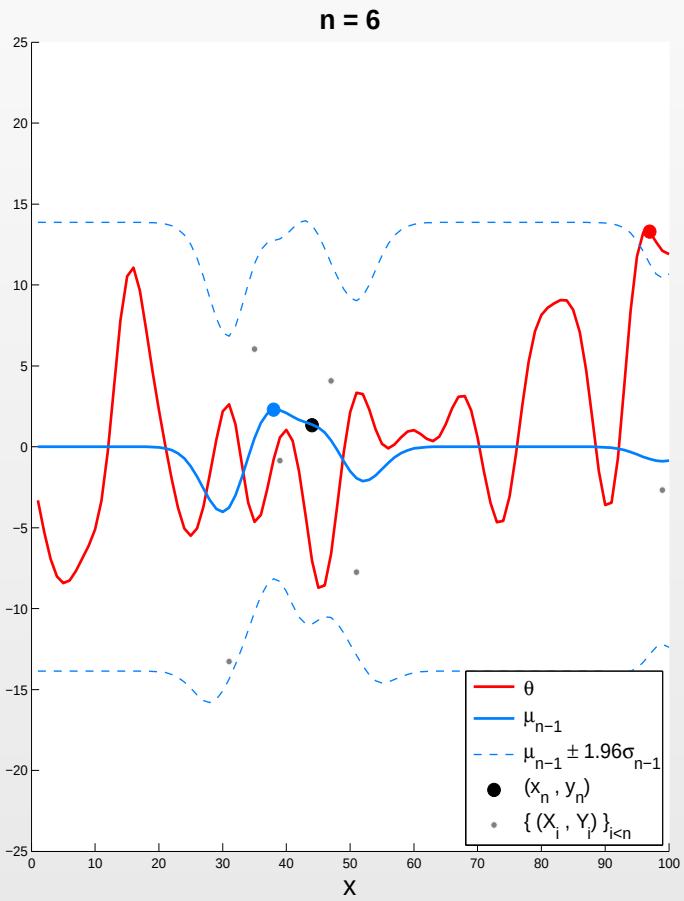


Illustration of the KG^2 Algorithm



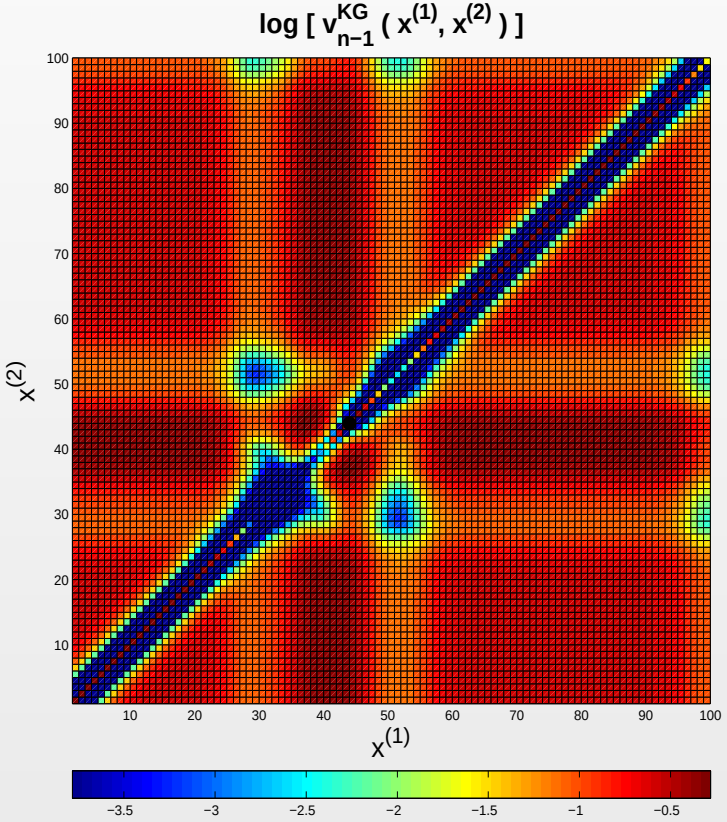
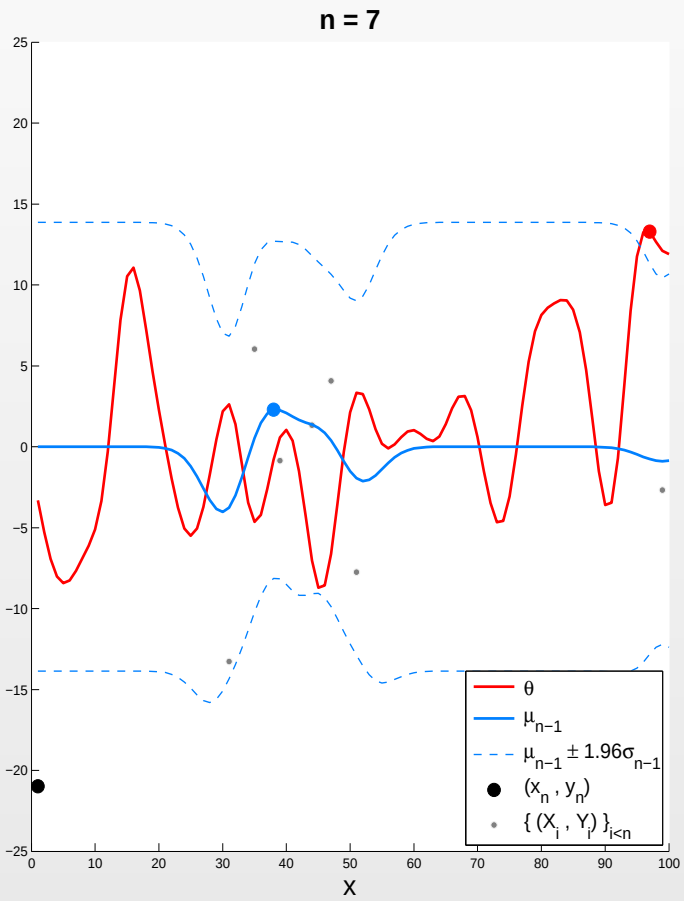


Illustration of the KG^2 Algorithm



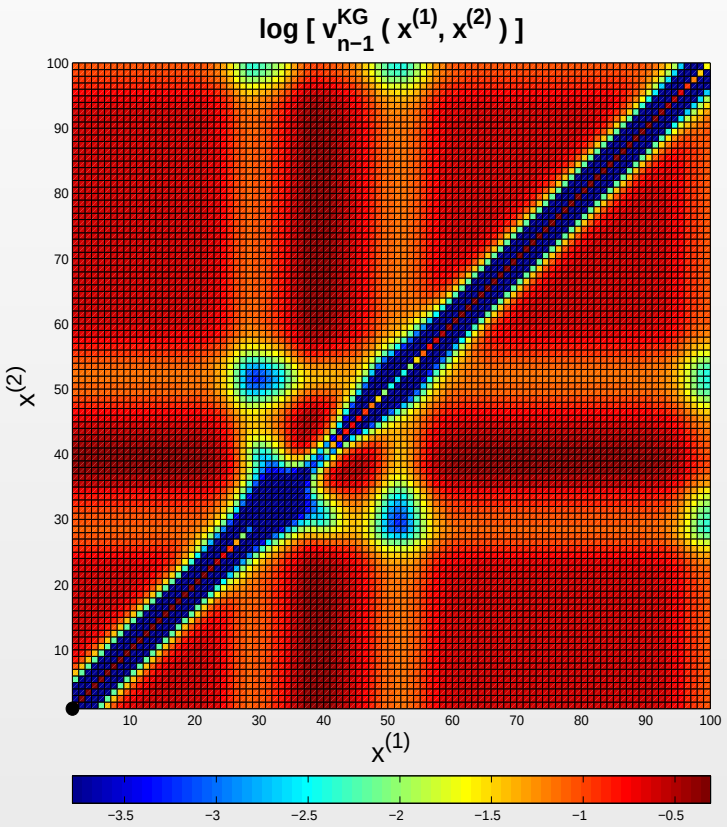
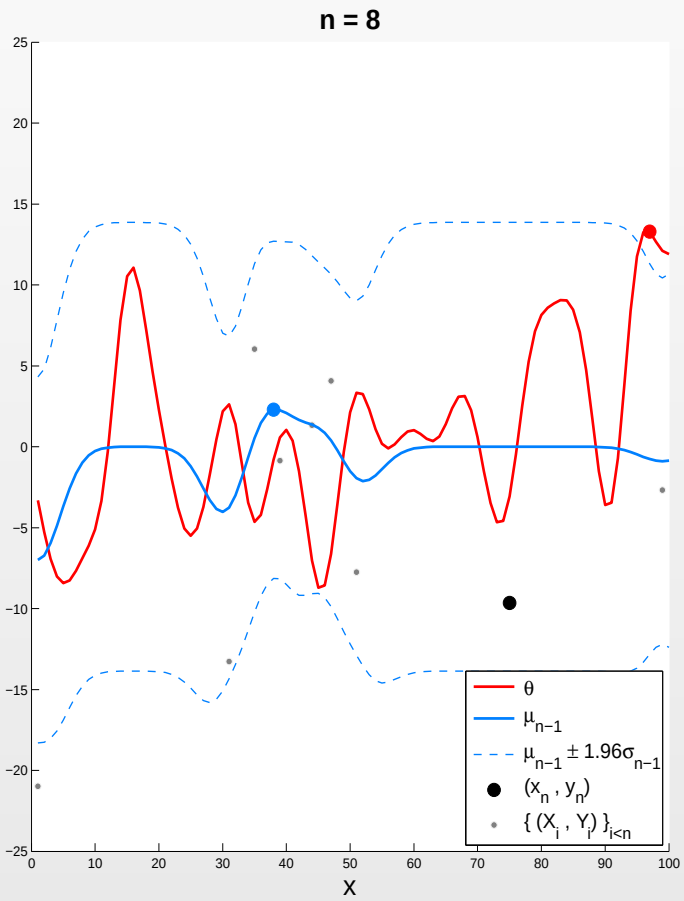


Illustration of the KG^2 Algorithm



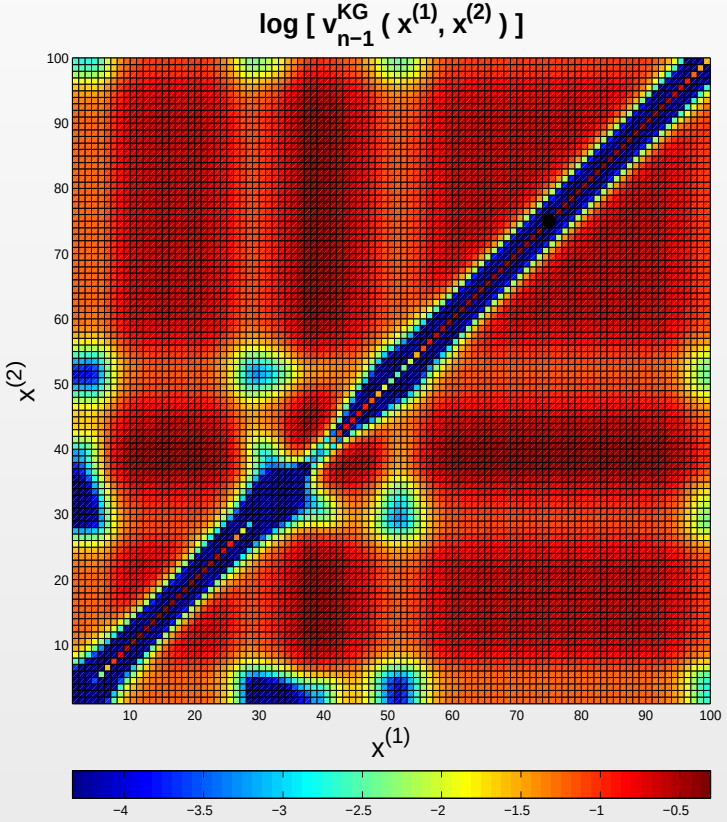
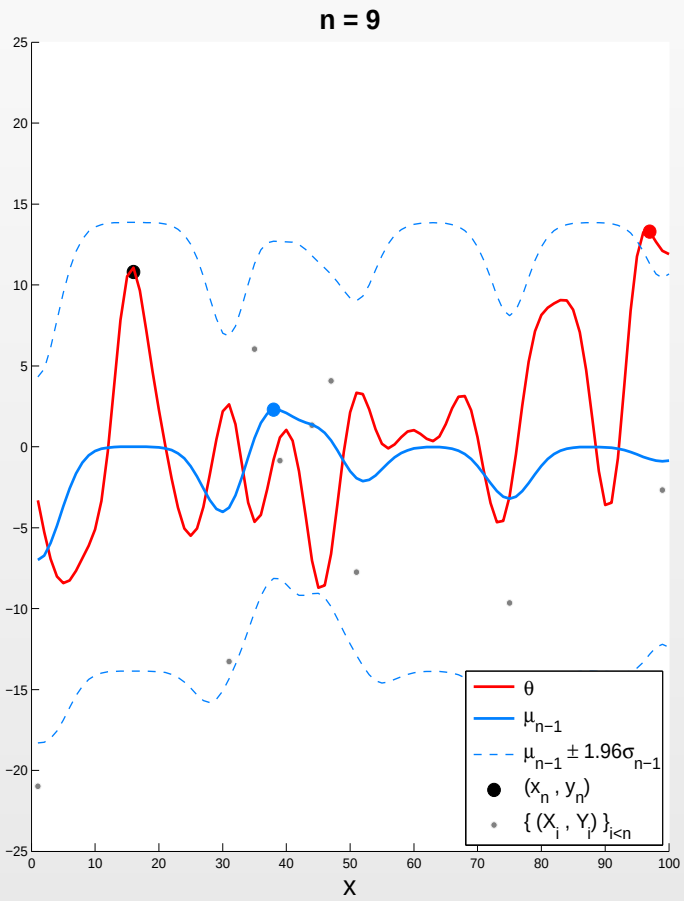


Illustration of the KG^2 Algorithm



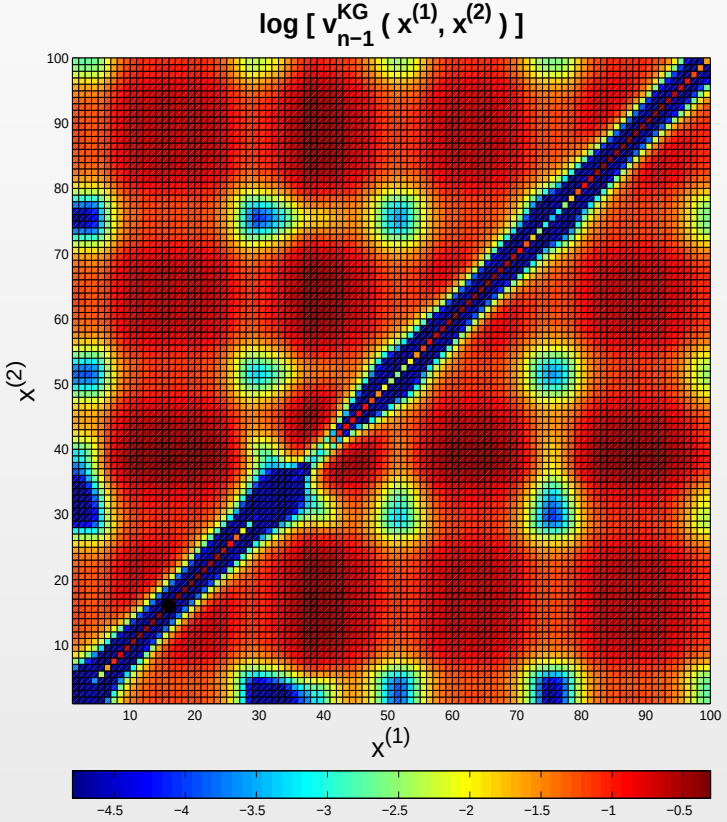
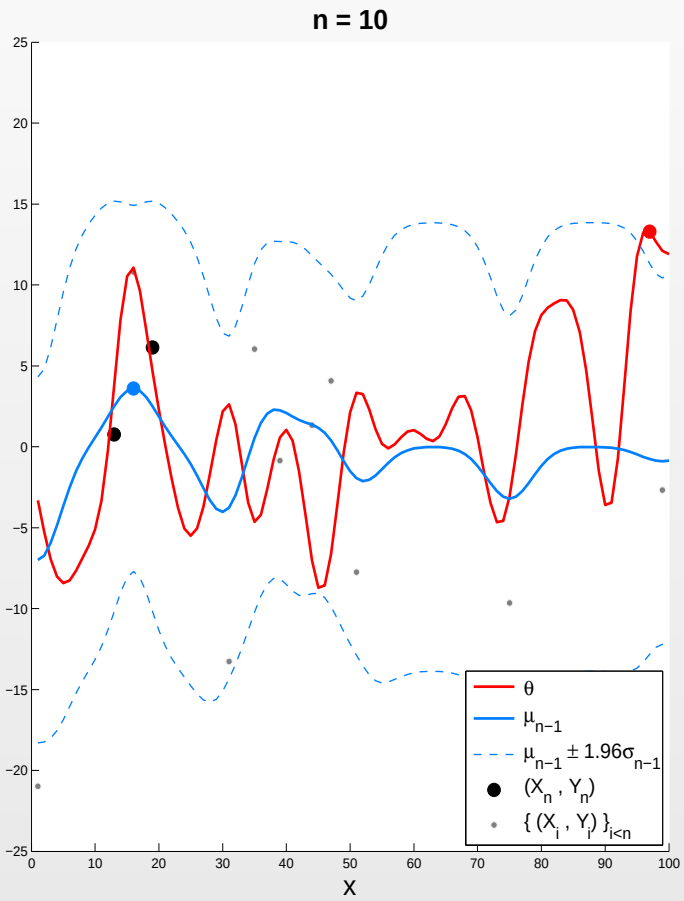


Illustration of the KG^2 Algorithm



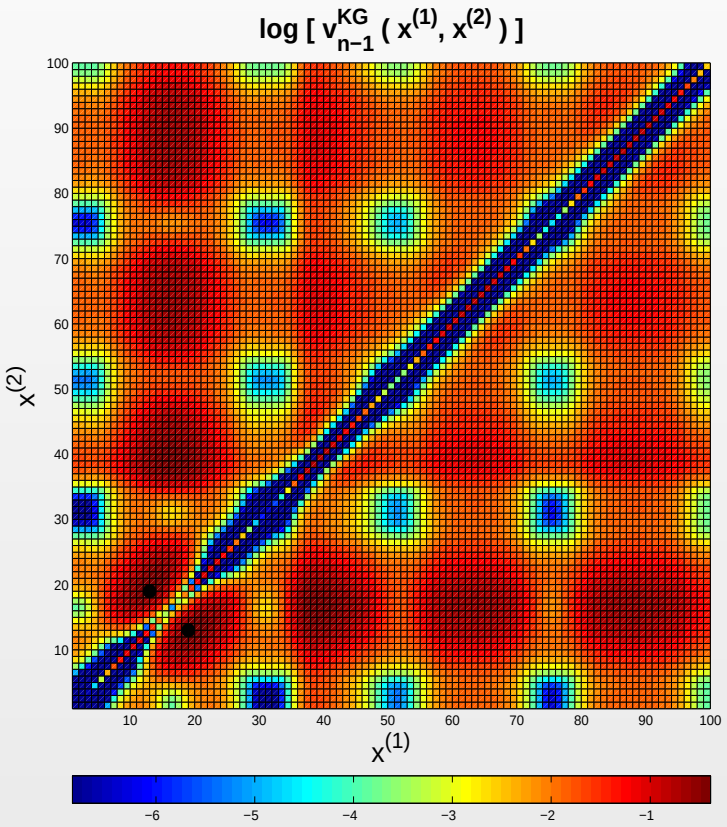
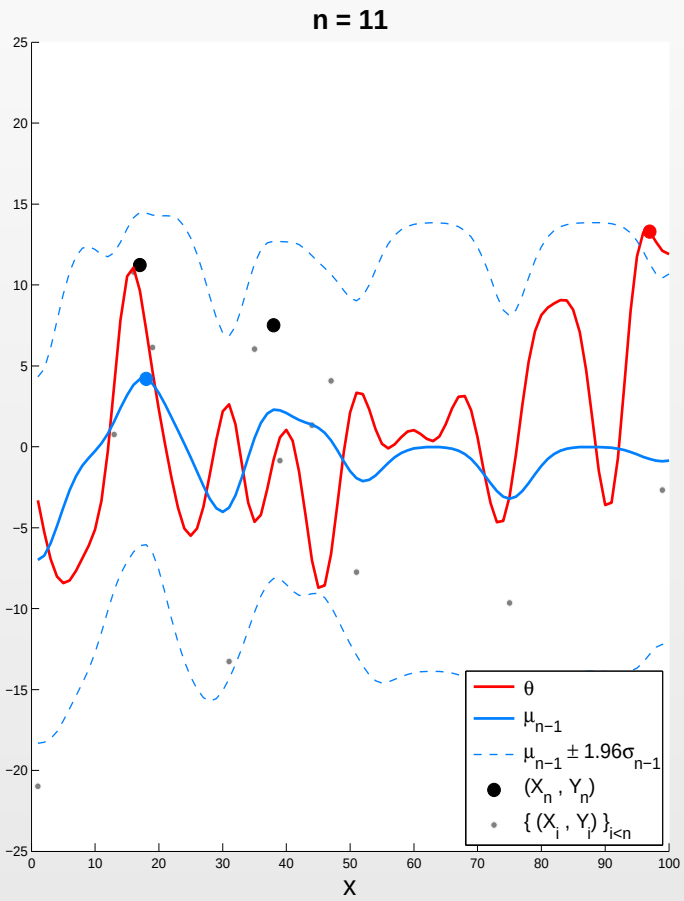


Illustration of the KG^2 Algorithm



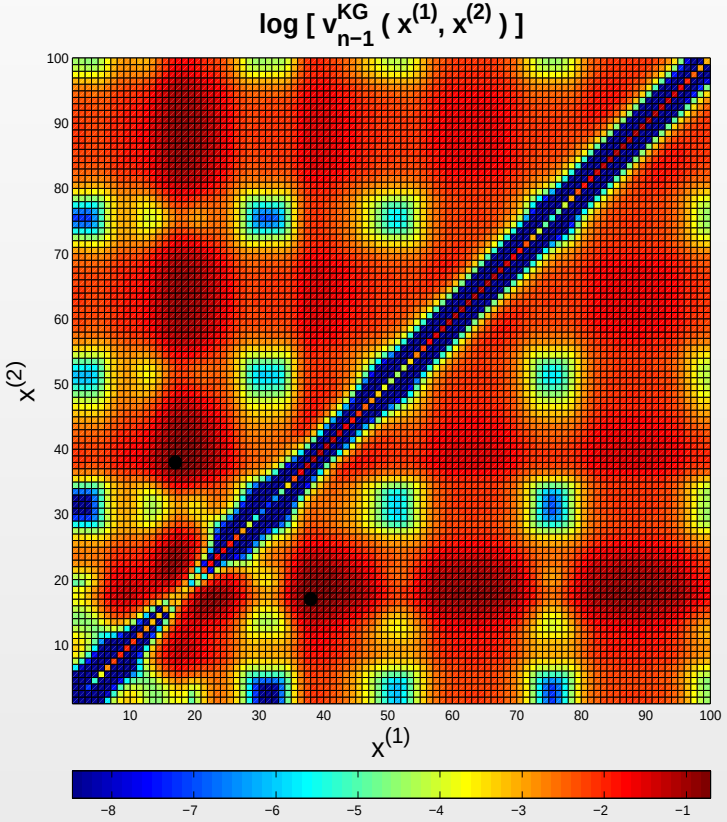
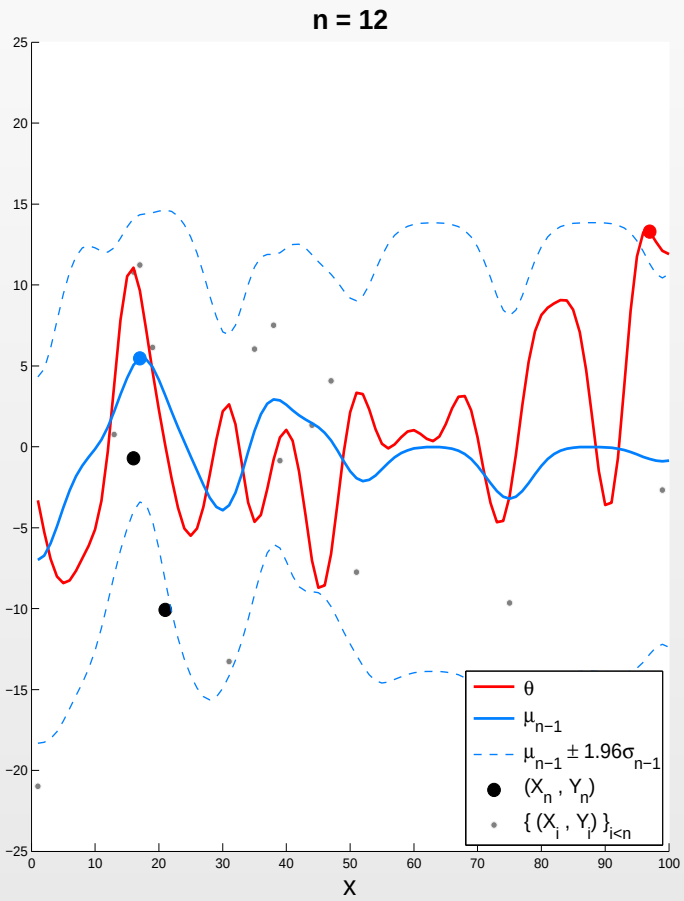


Illustration of the KG^2 Algorithm



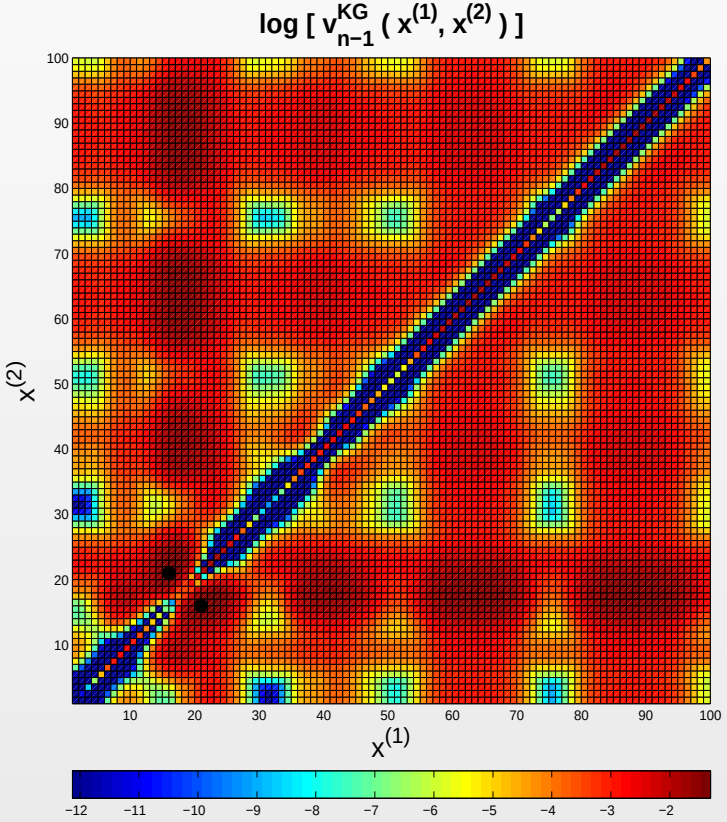
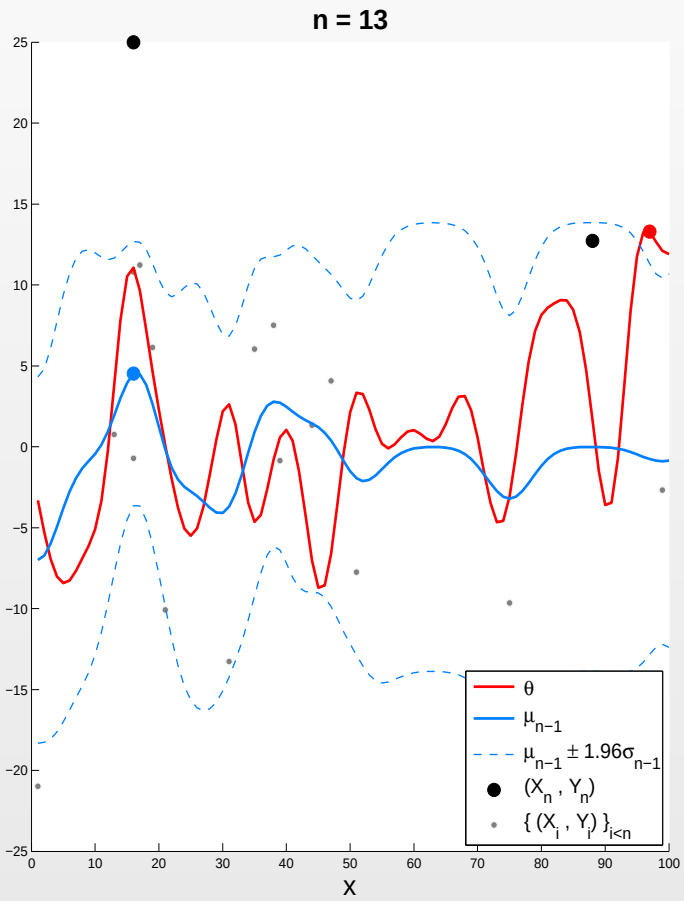


Illustration of the KG^2 Algorithm



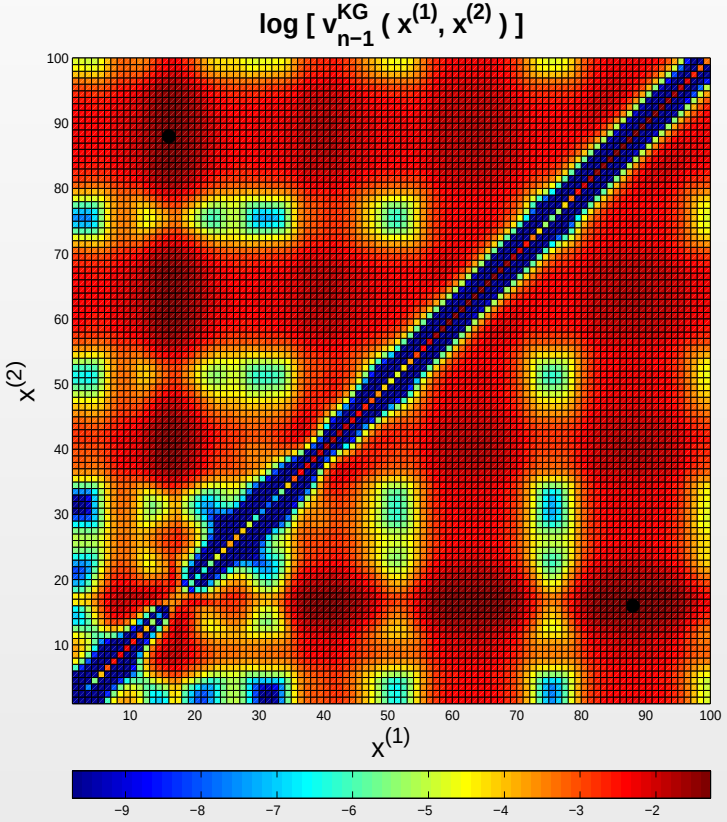
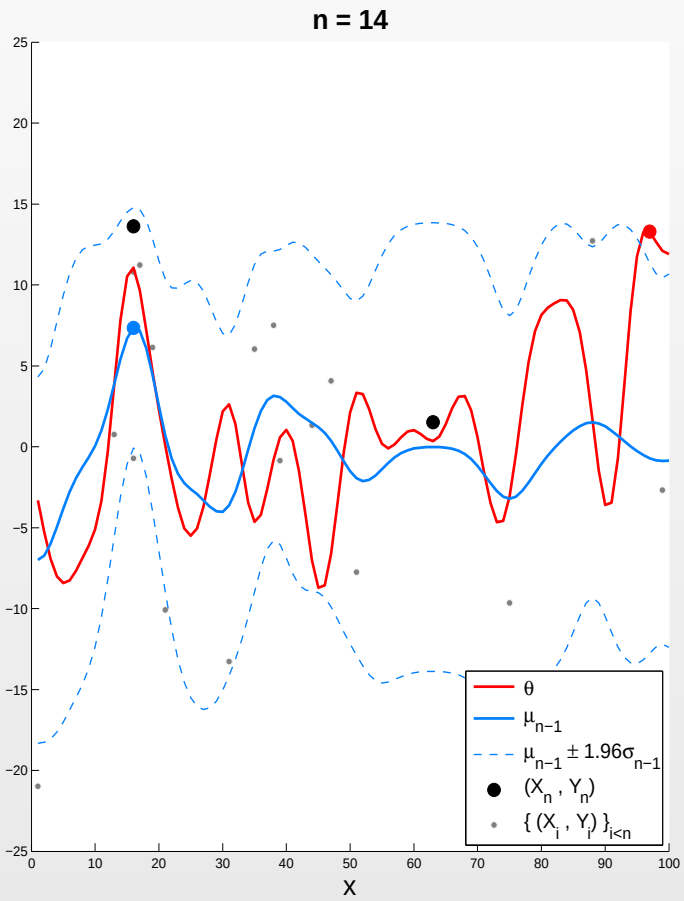


Illustration of the KG^2 Algorithm



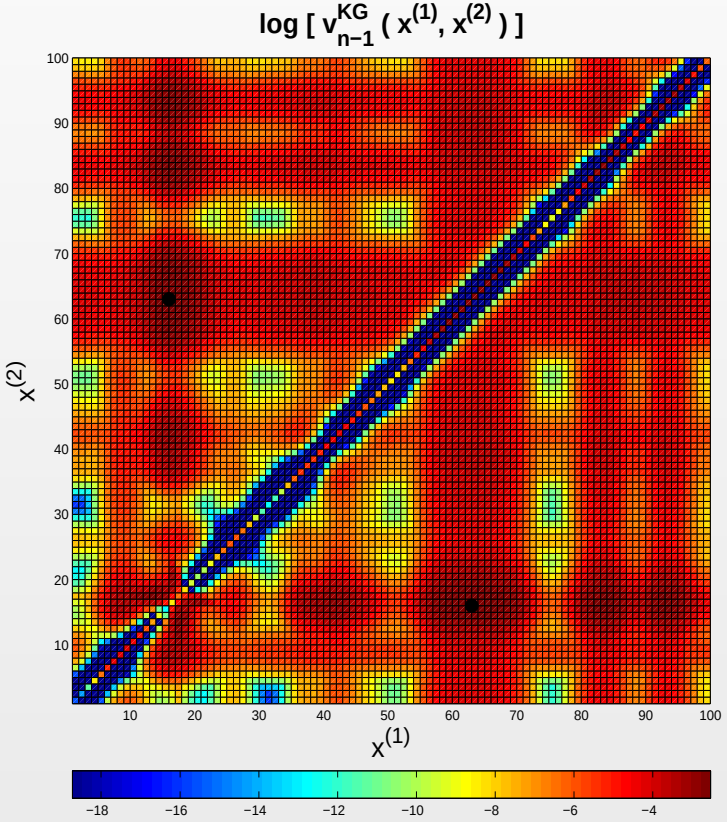
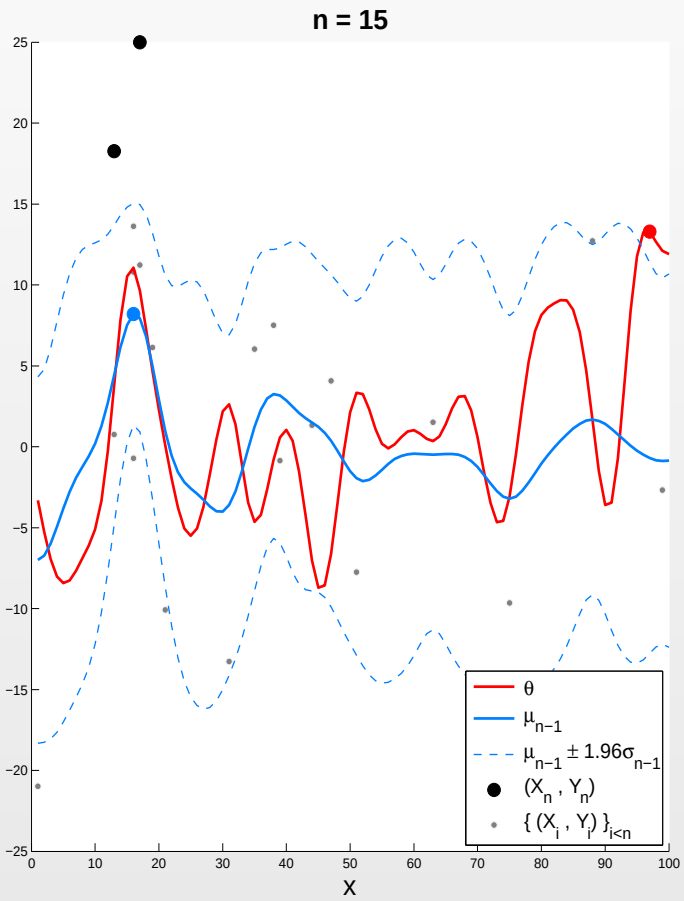


Illustration of the KG^2 Algorithm



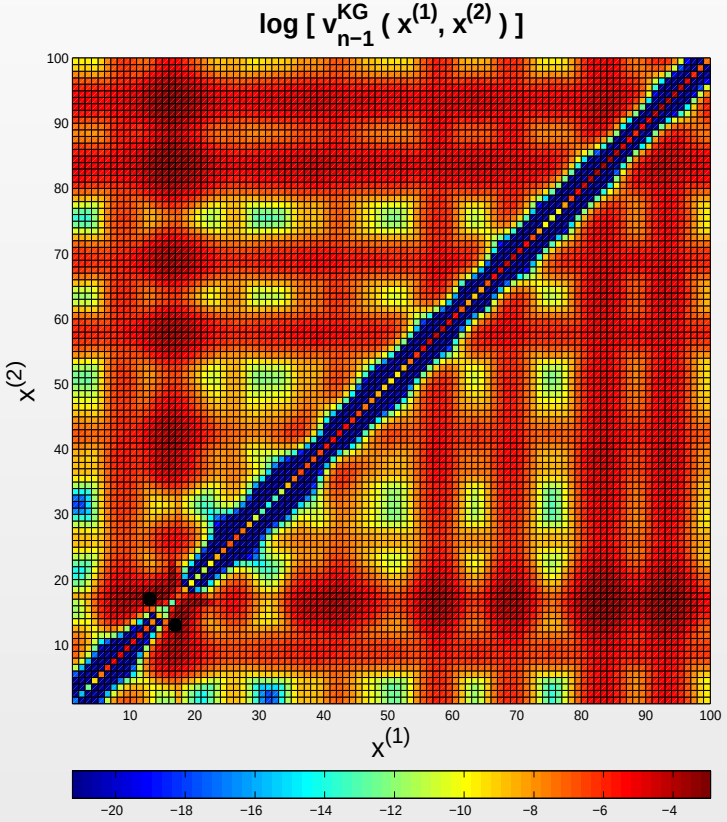
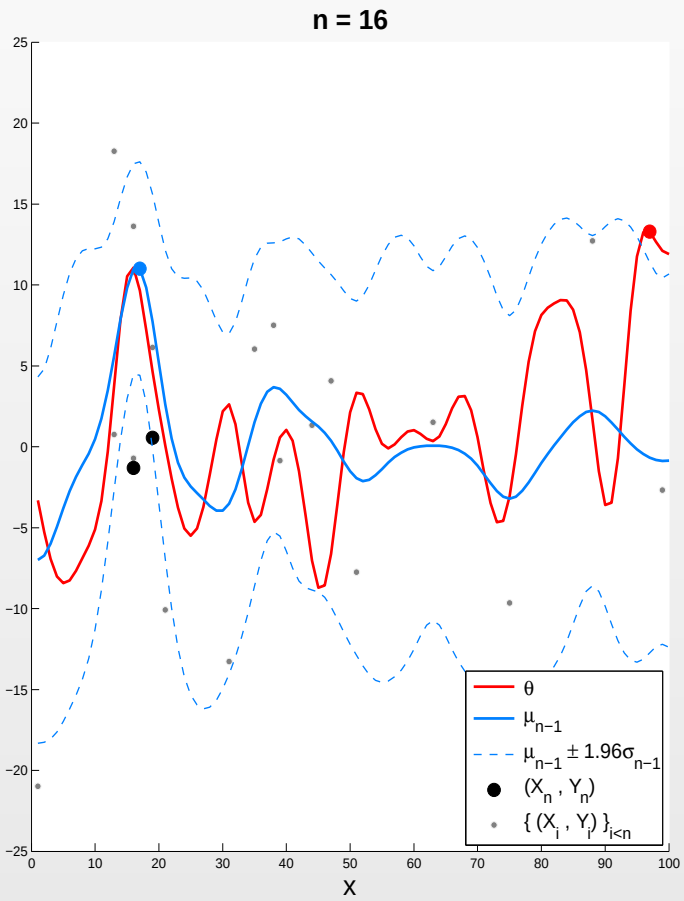


Illustration of the KG^2 Algorithm



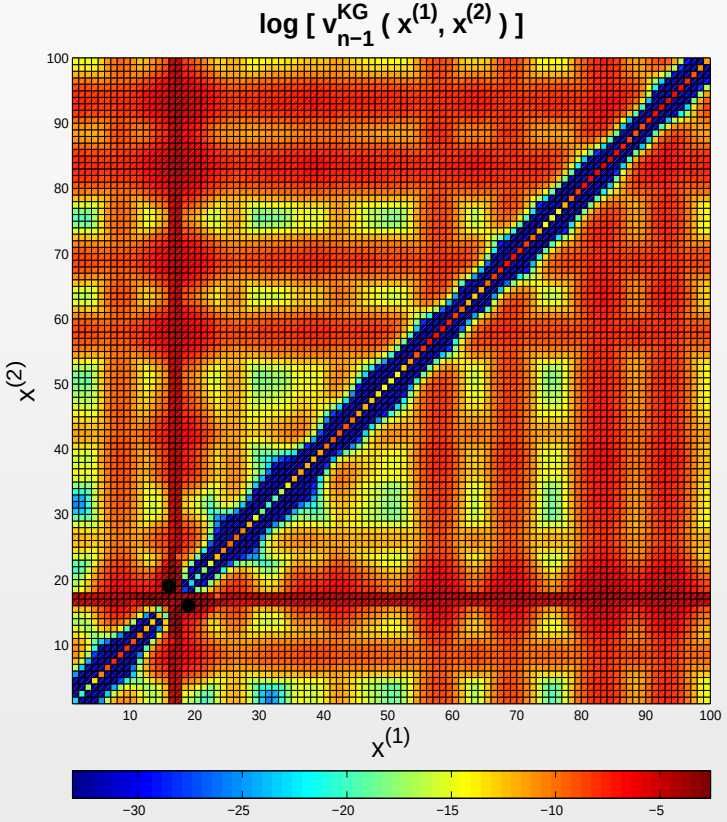
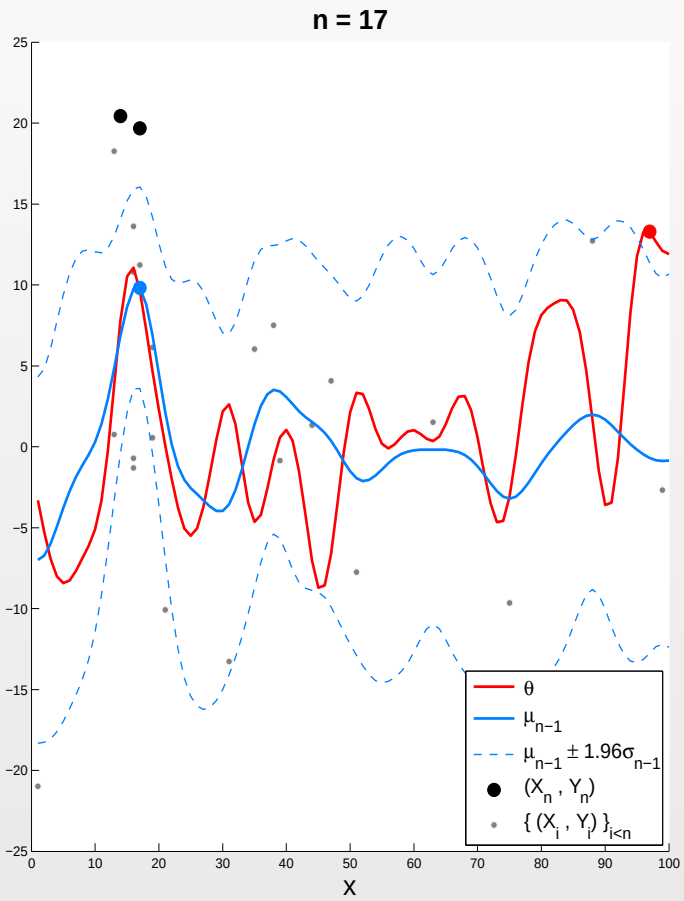


Illustration of the KG^2 Algorithm



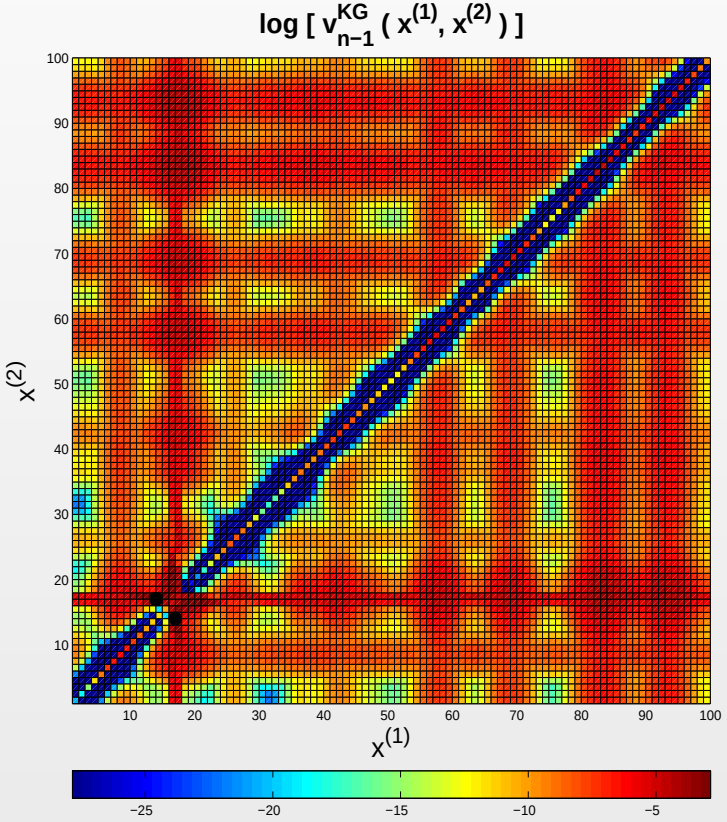
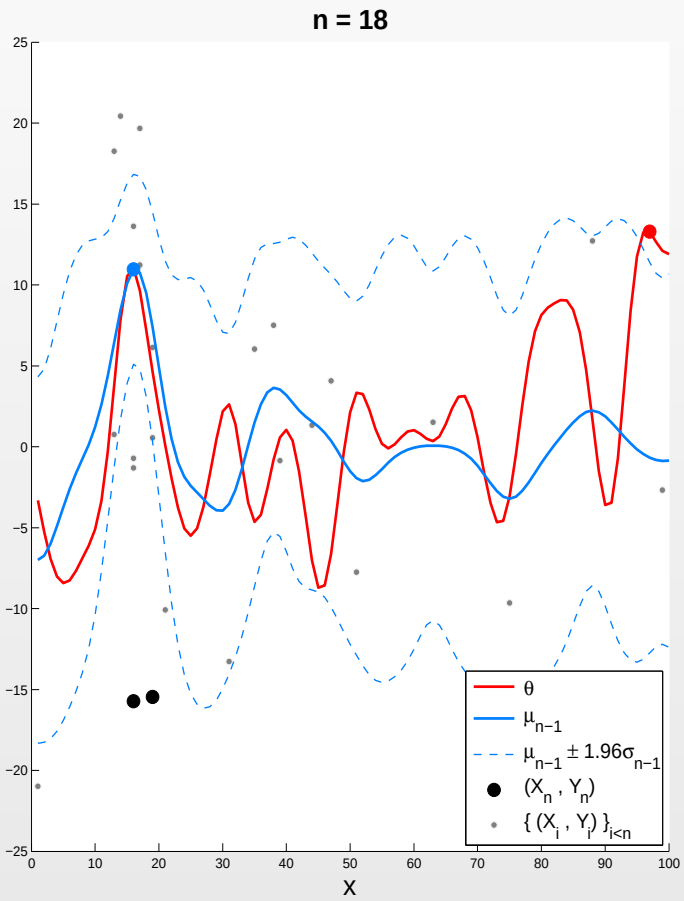


Illustration of the KG^2 Algorithm



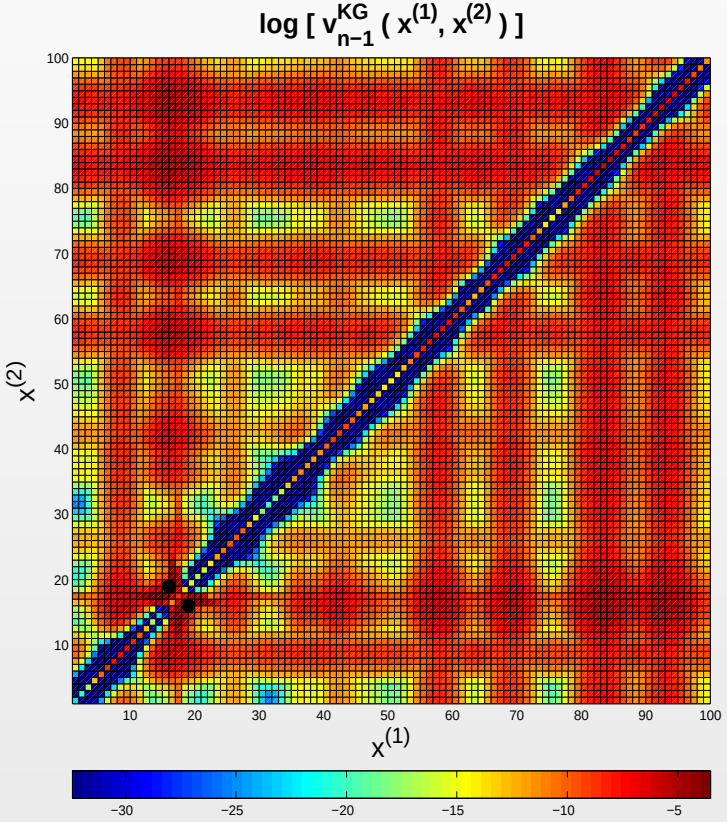
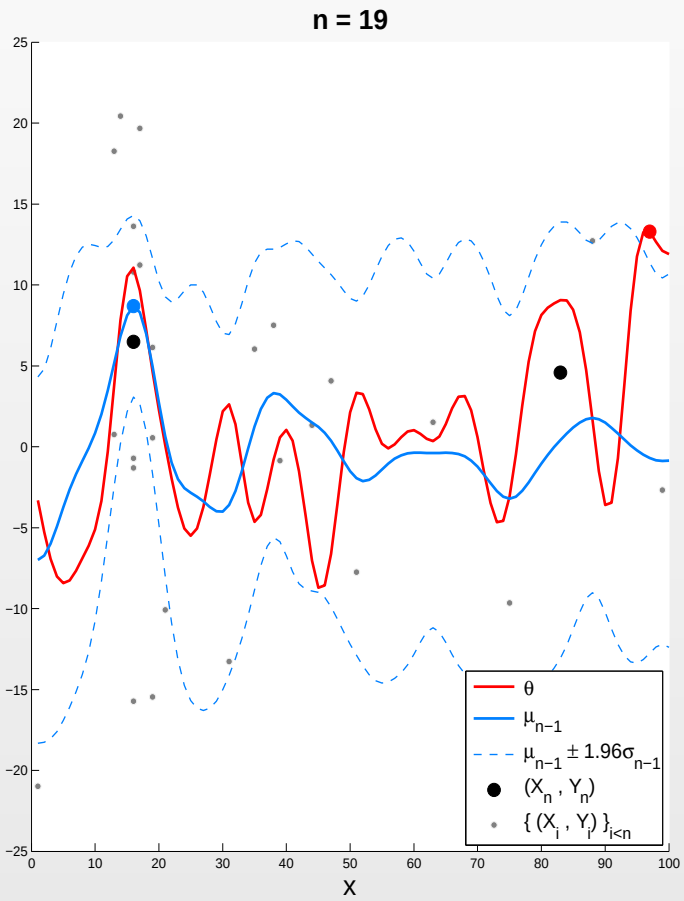


Illustration of the KG^2 Algorithm



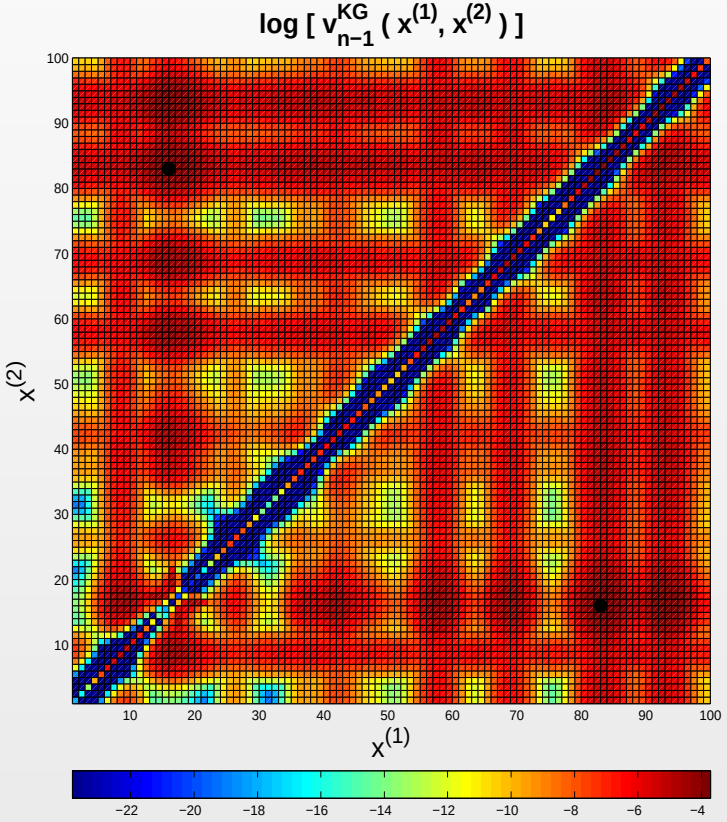
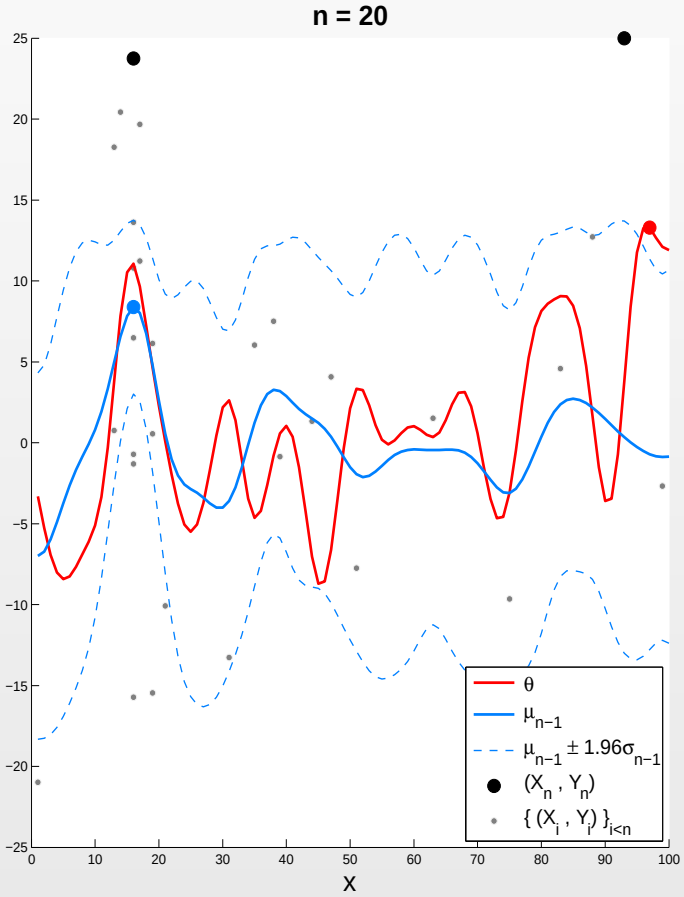


Illustration of the KG^2 Algorithm



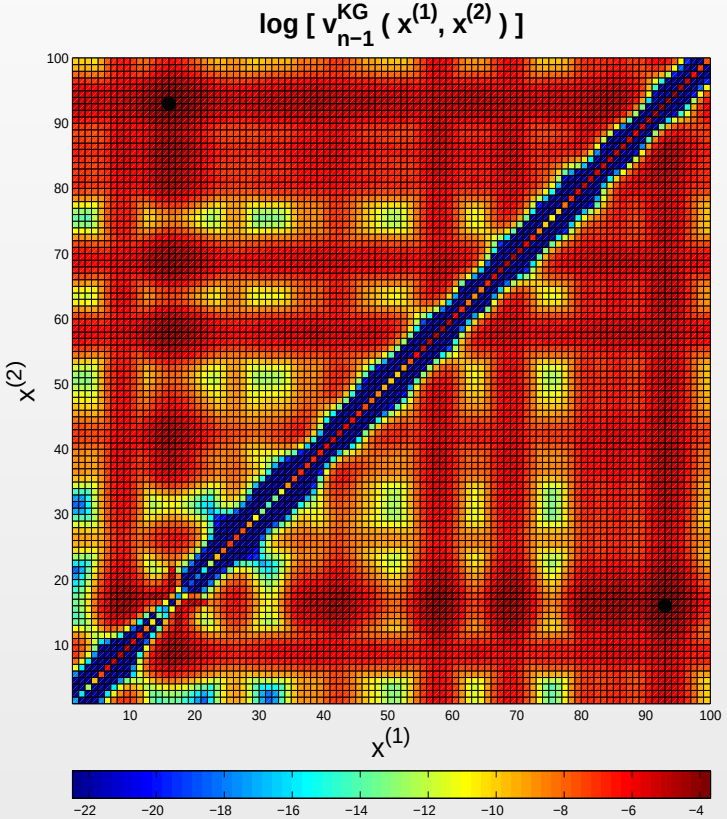
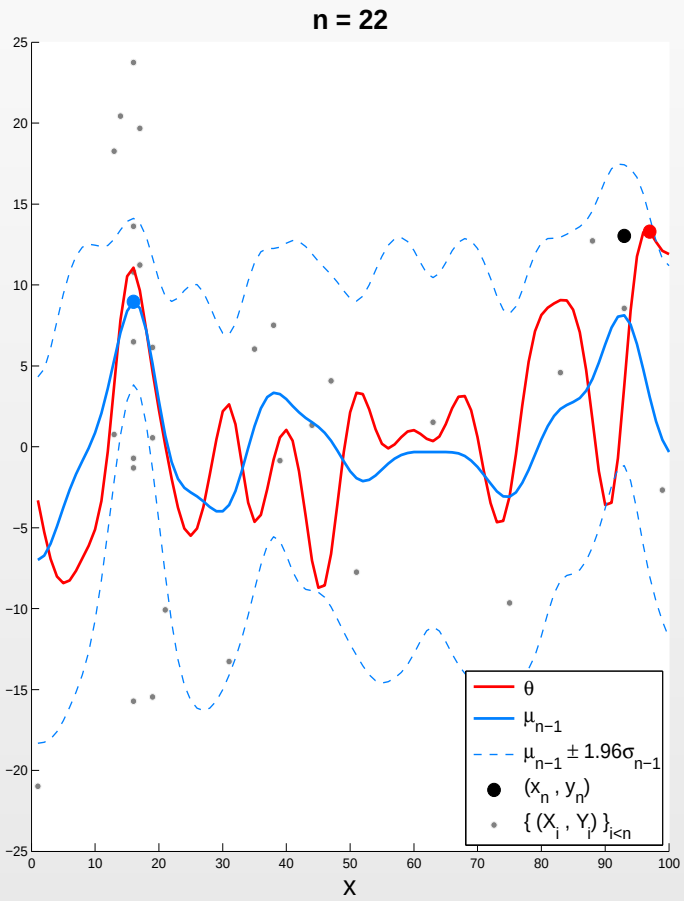


Illustration of the KG^2 Algorithm



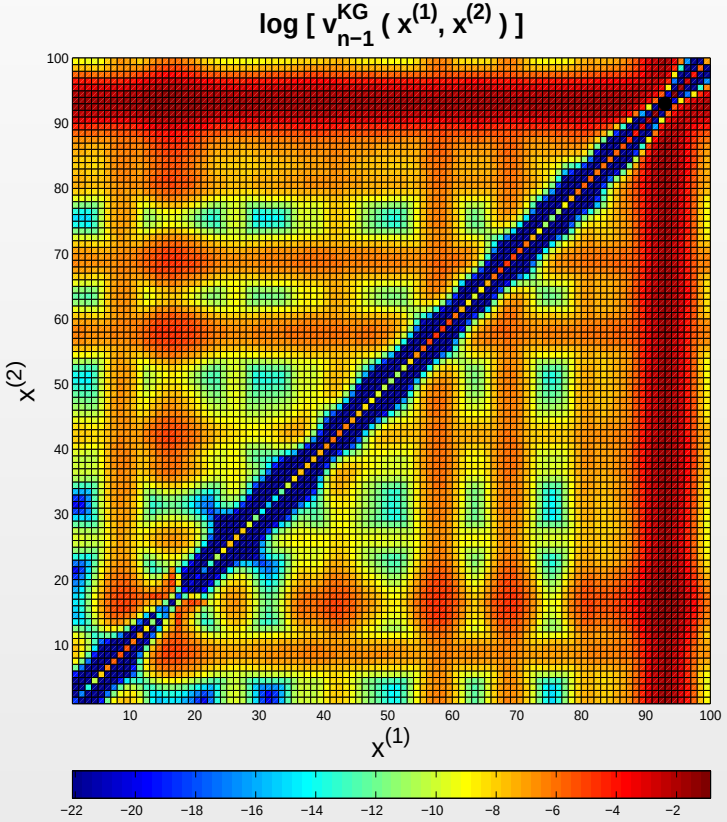
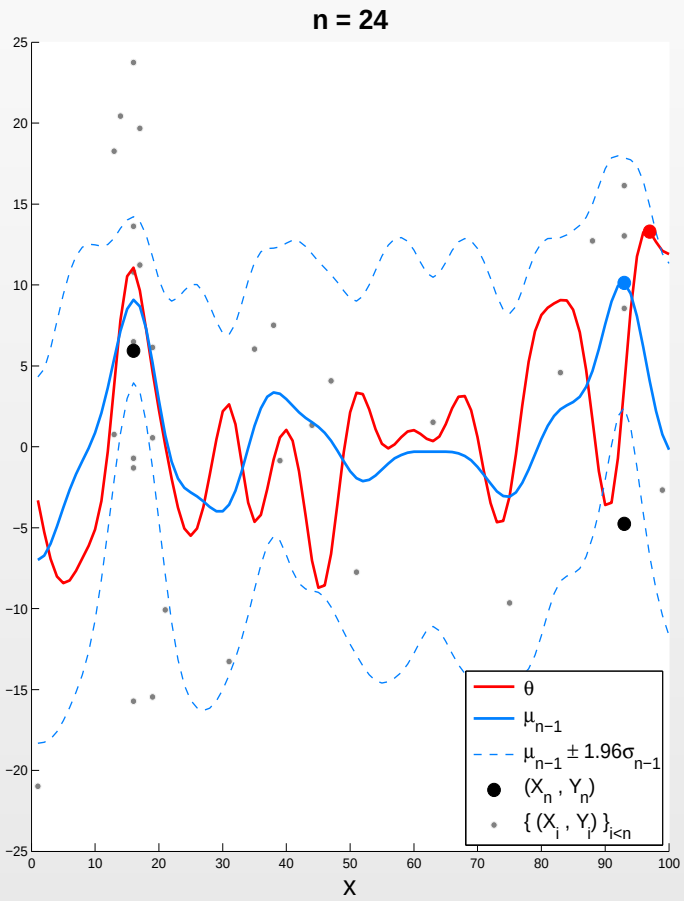


Illustration of the KG^2 Algorithm



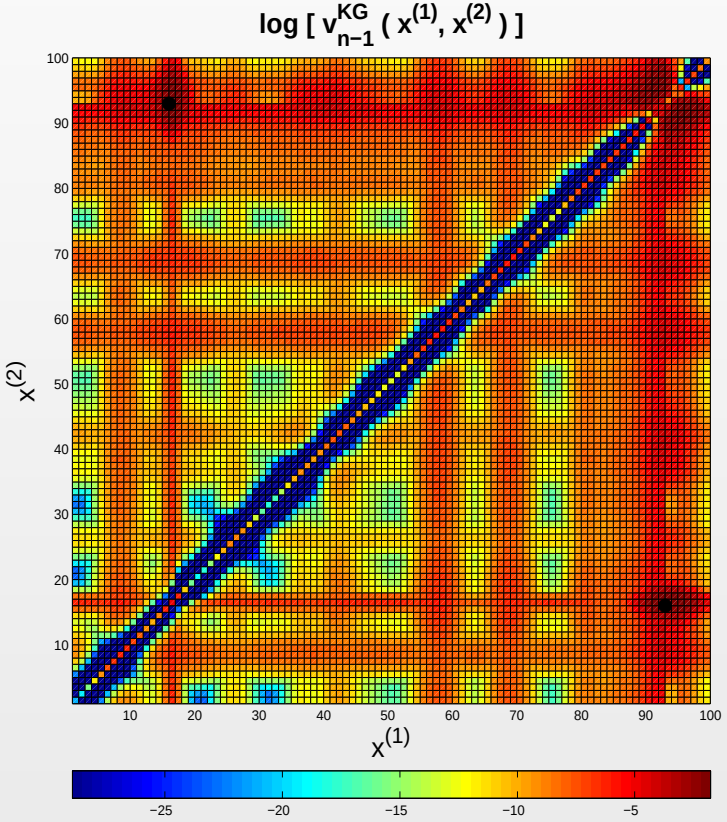
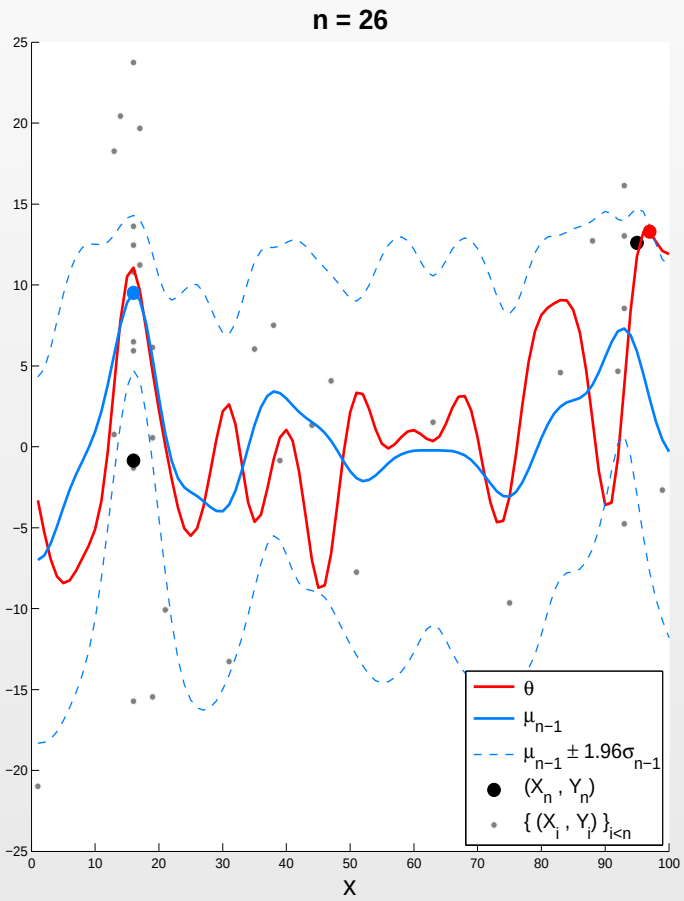


Illustration of the KG^2 Algorithm



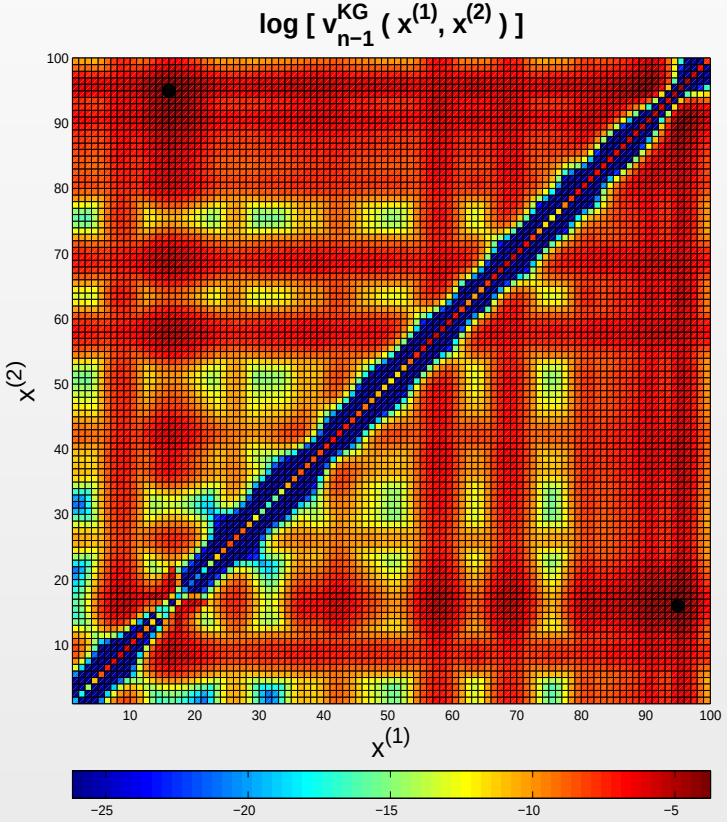
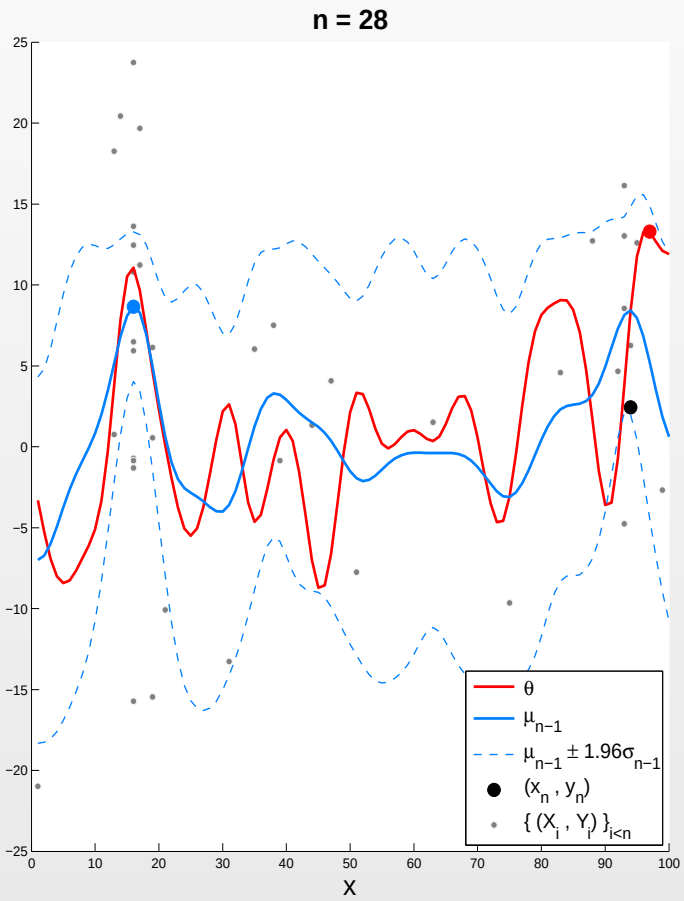


Illustration of the KG^2 Algorithm



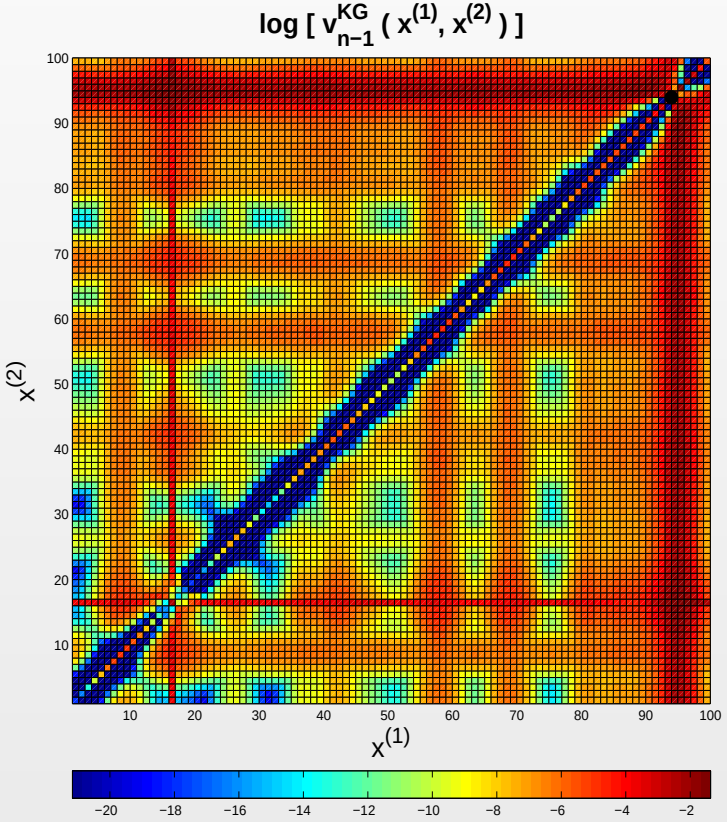
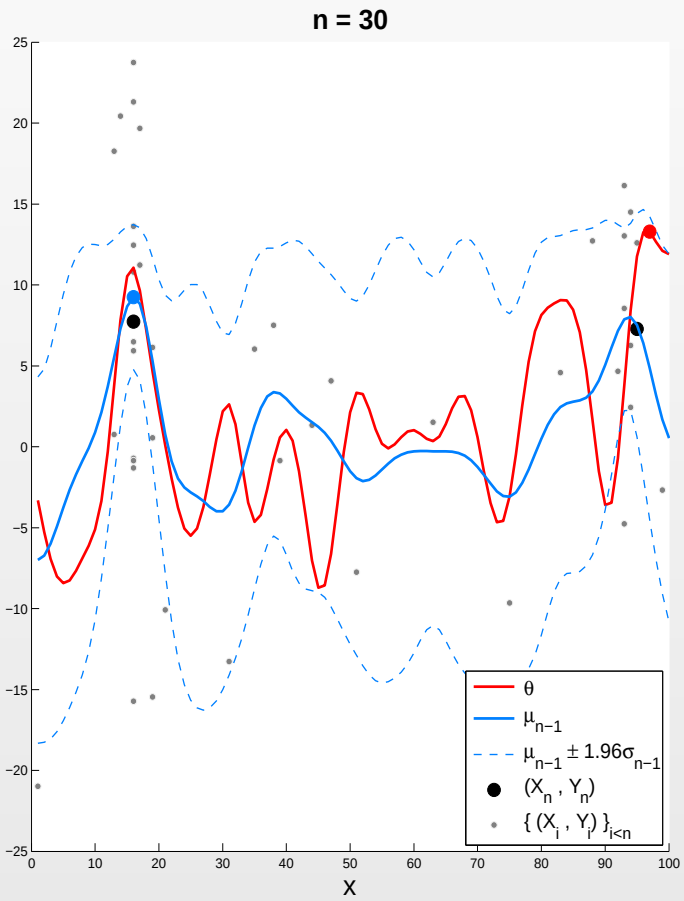


Illustration of the KG^2 Algorithm



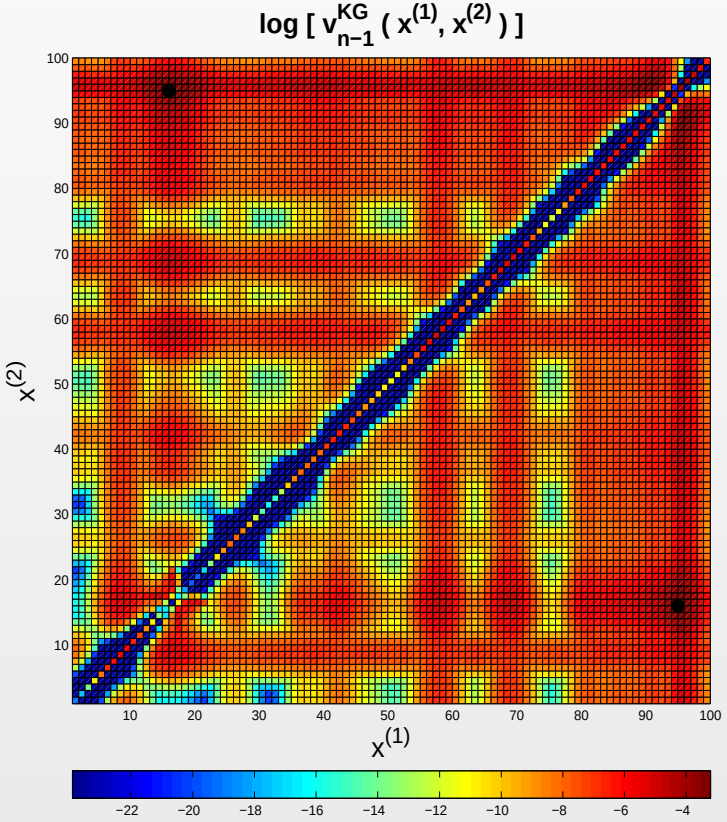
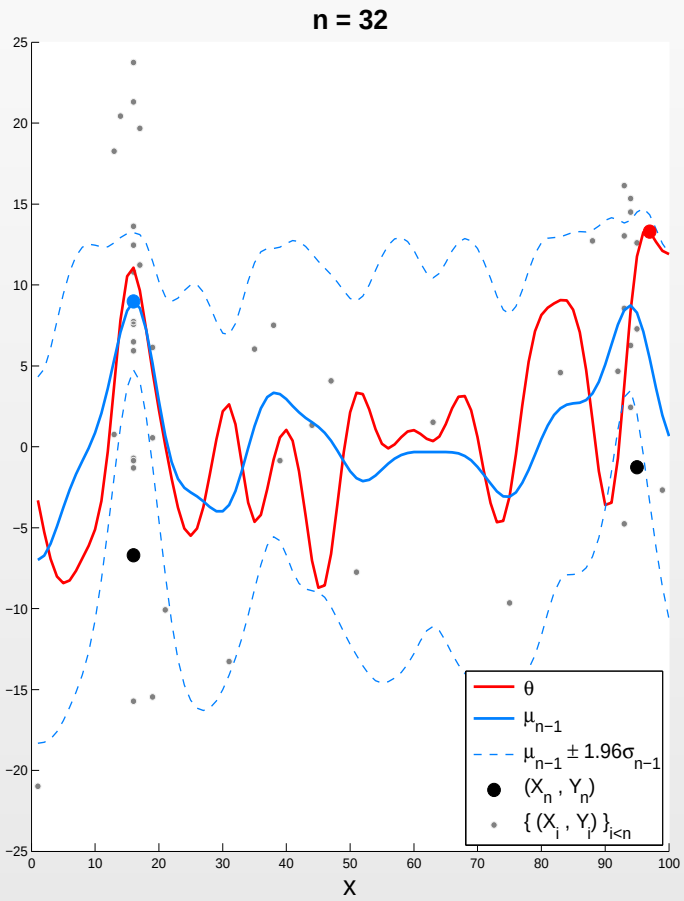


Illustration of the KG^2 Algorithm



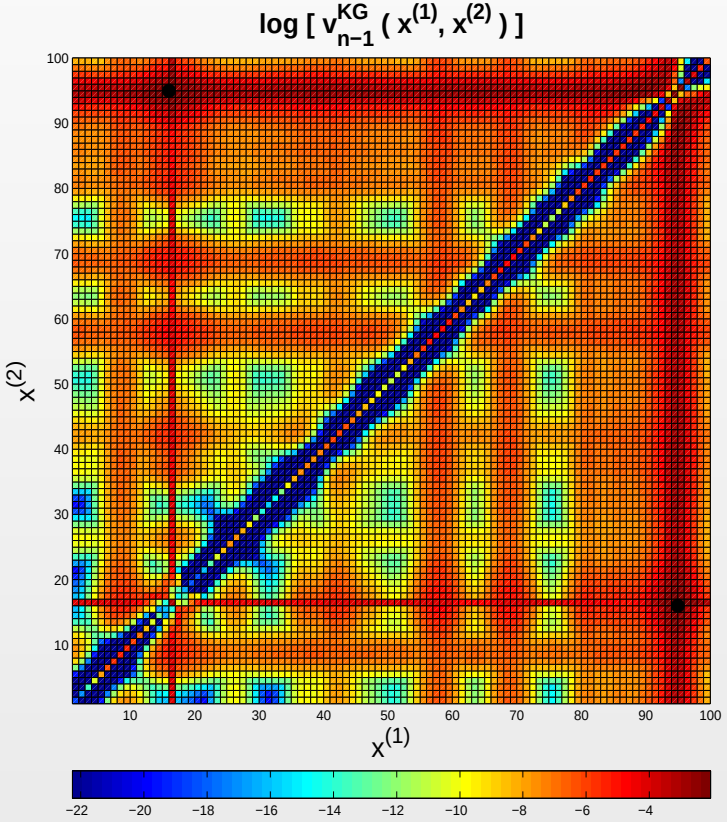
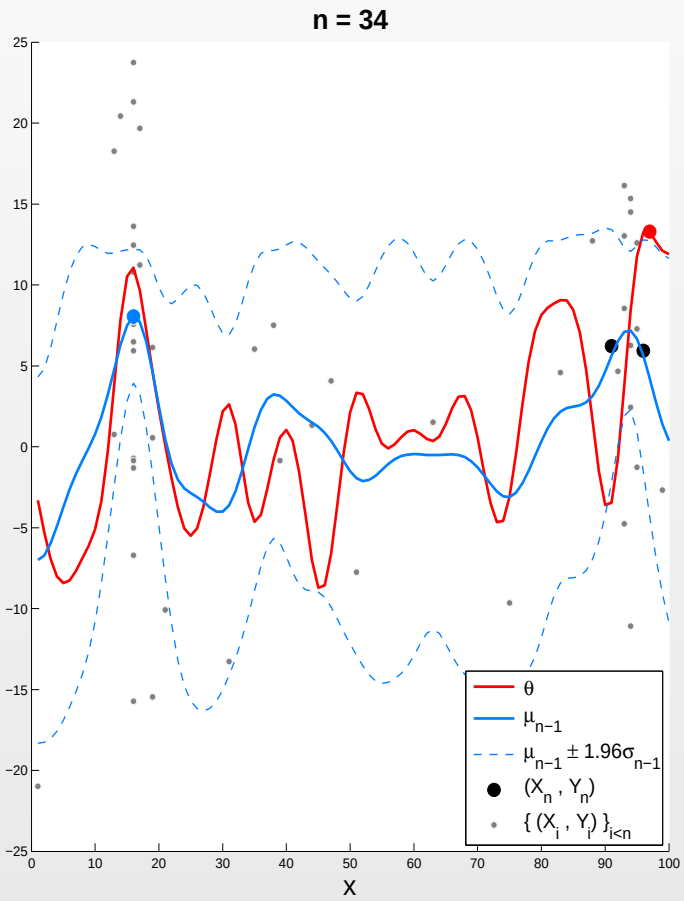


Illustration of the KG^2 Algorithm



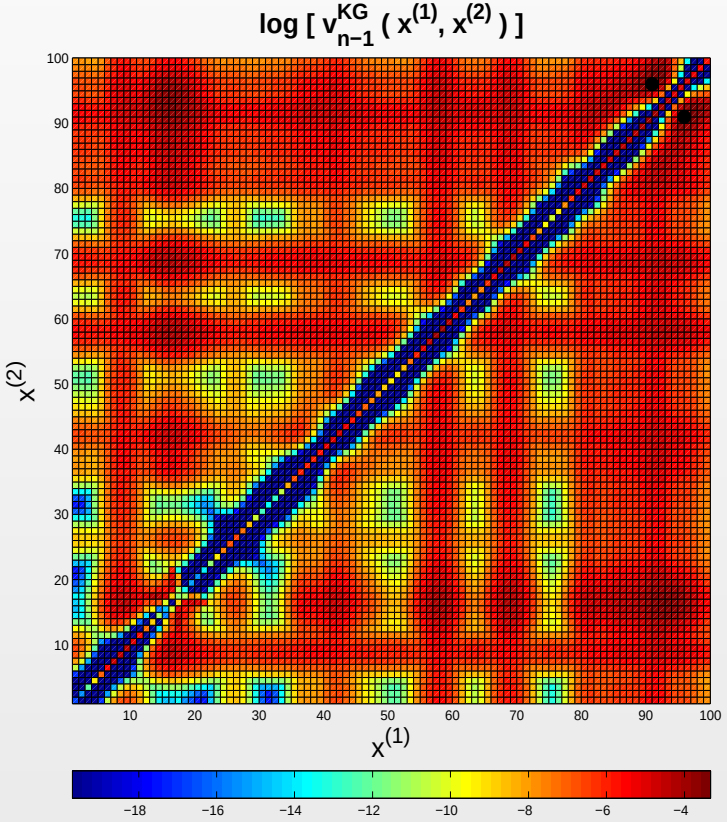
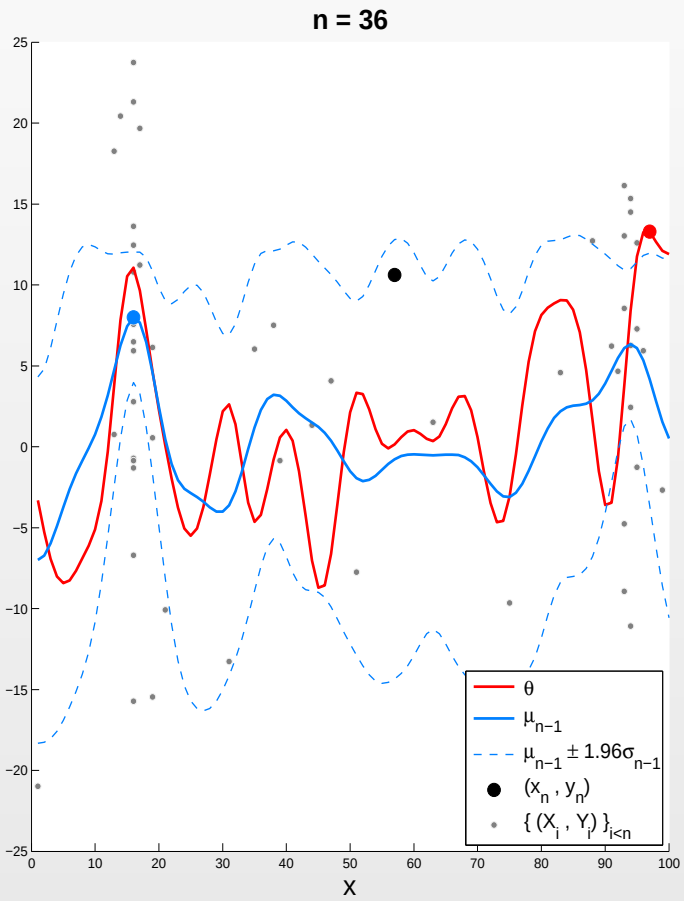


Illustration of the KG^2 Algorithm



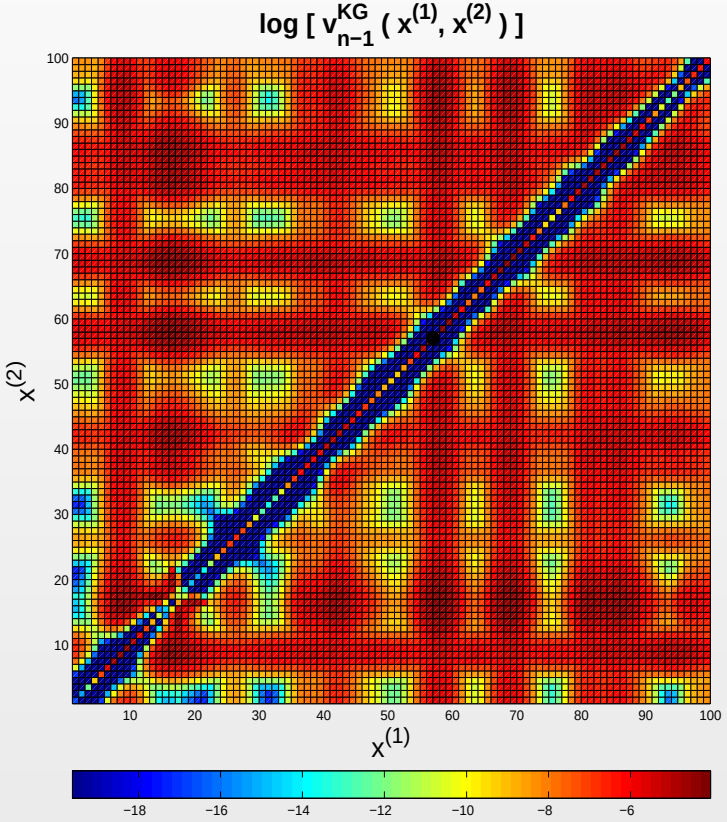
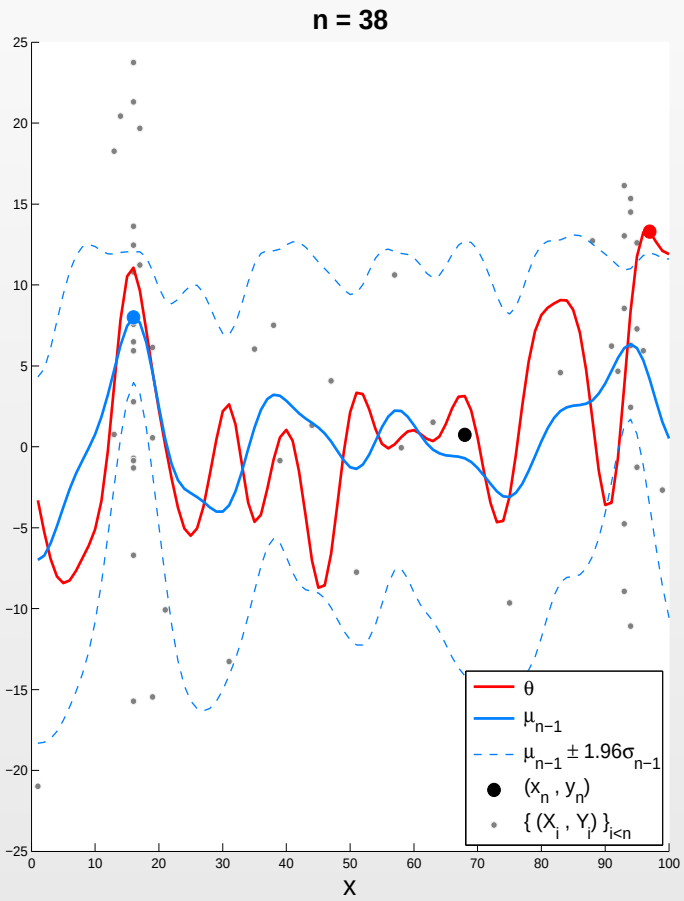


Illustration of the KG^2 Algorithm



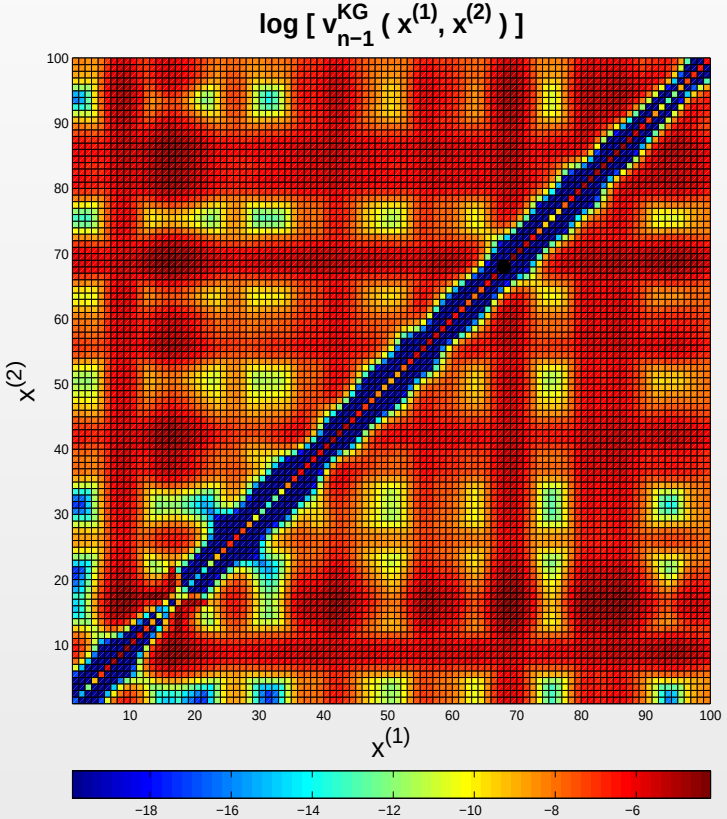
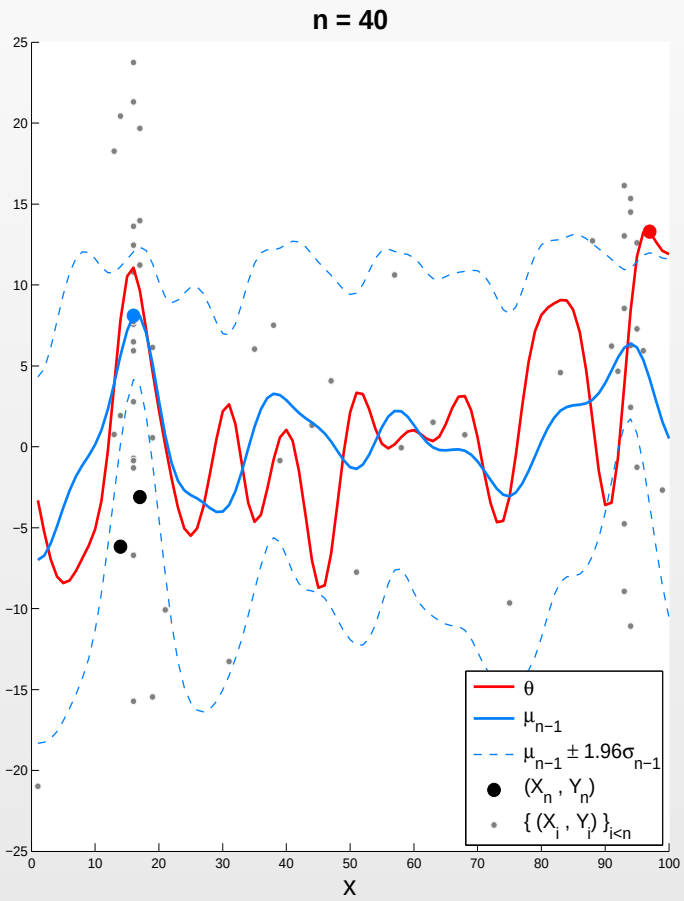


Illustration of the KG^2 Algorithm



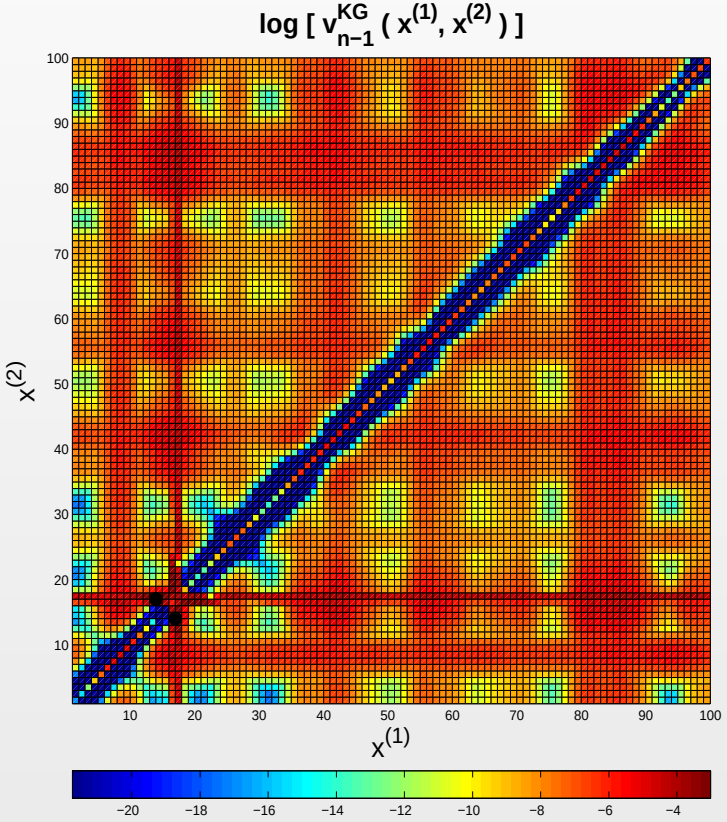
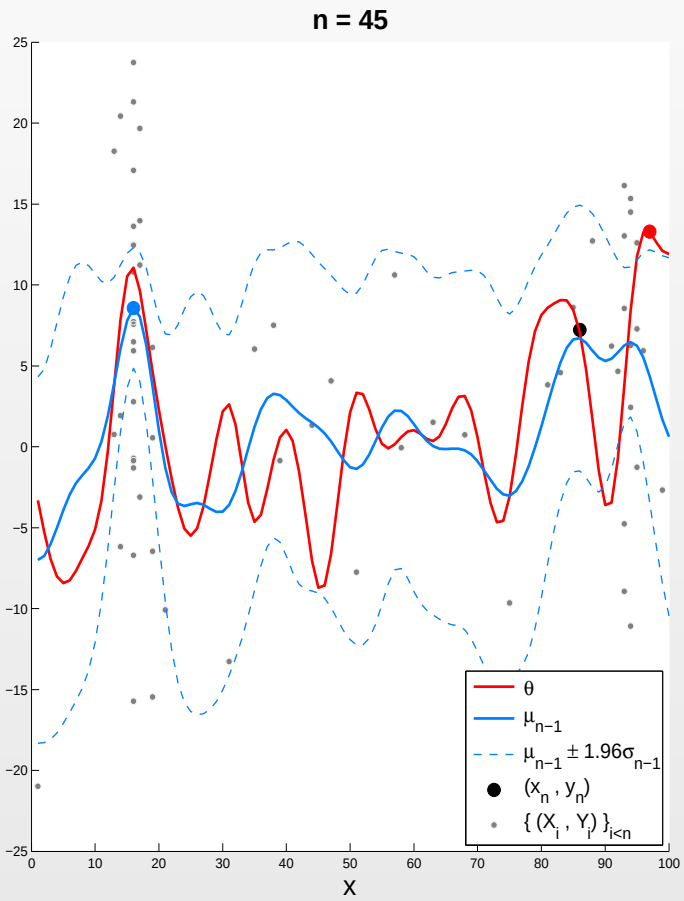


Illustration of the KG^2 Algorithm



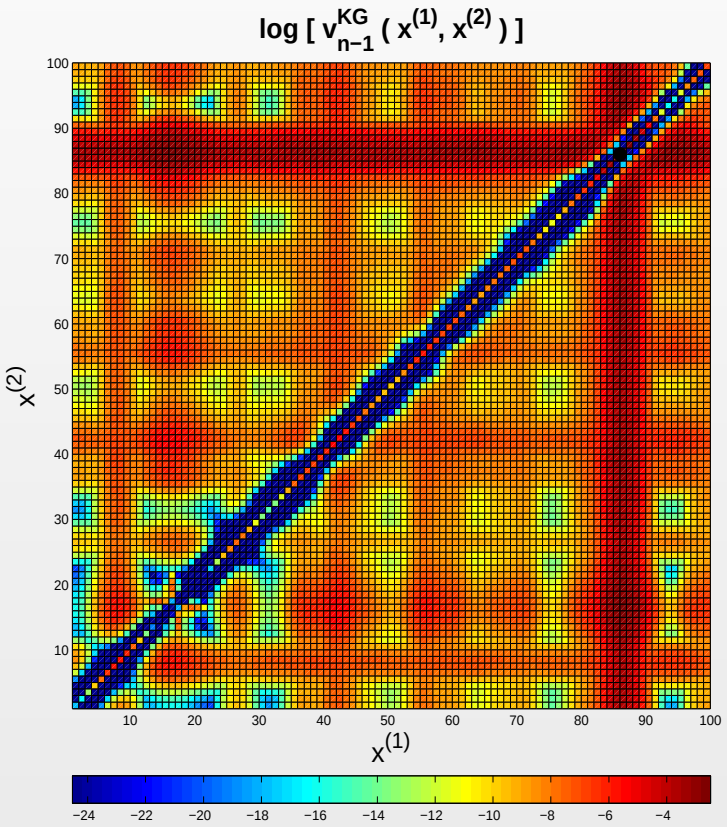
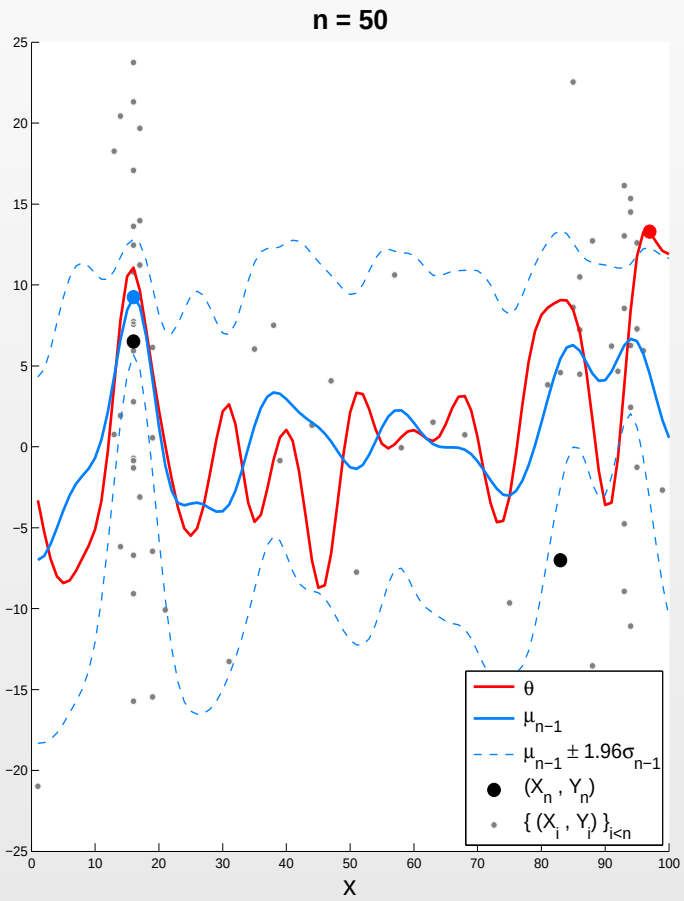


Illustration of the KG^2 Algorithm



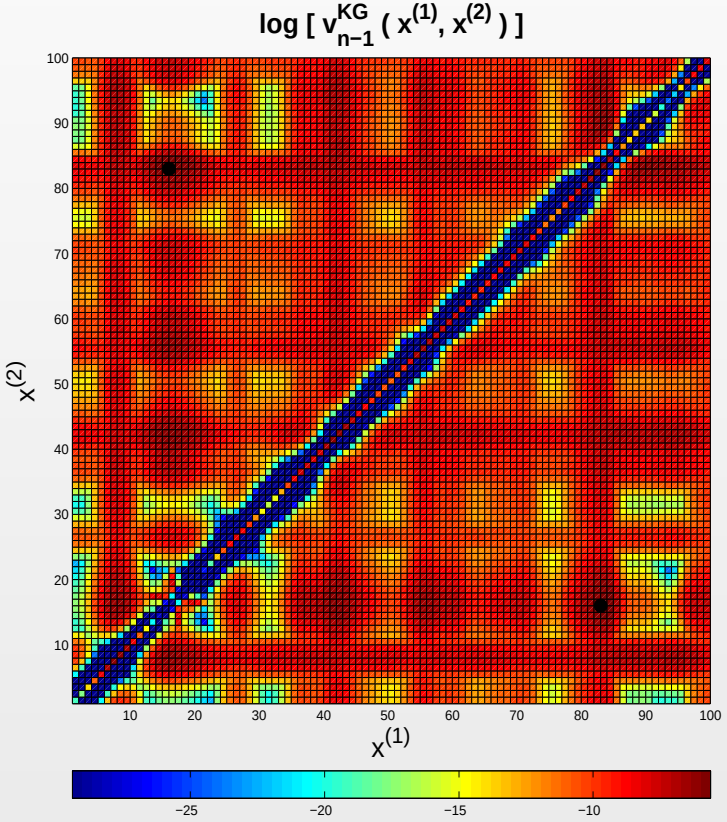
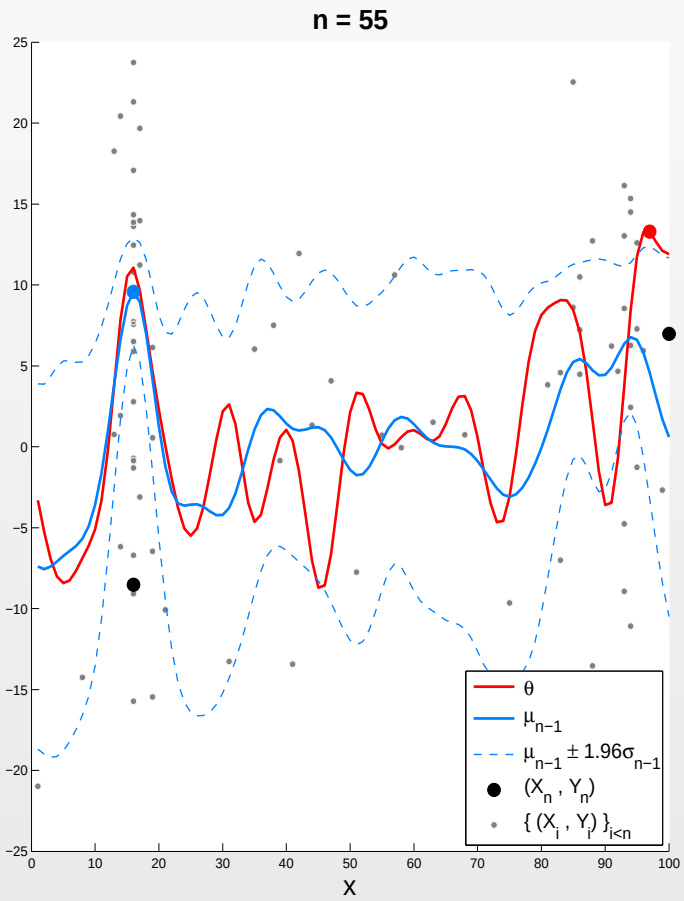


Illustration of the KG^2 Algorithm



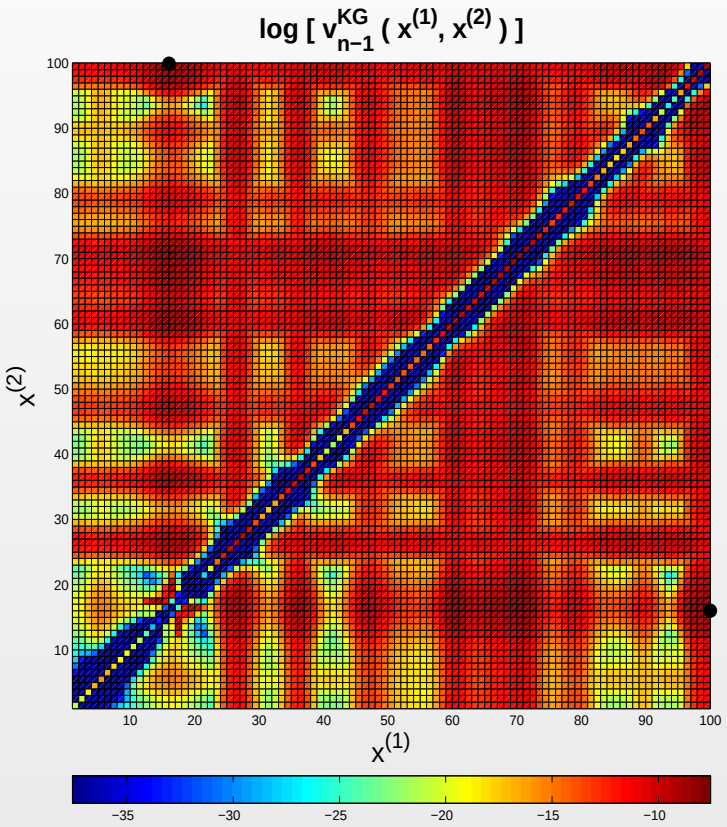
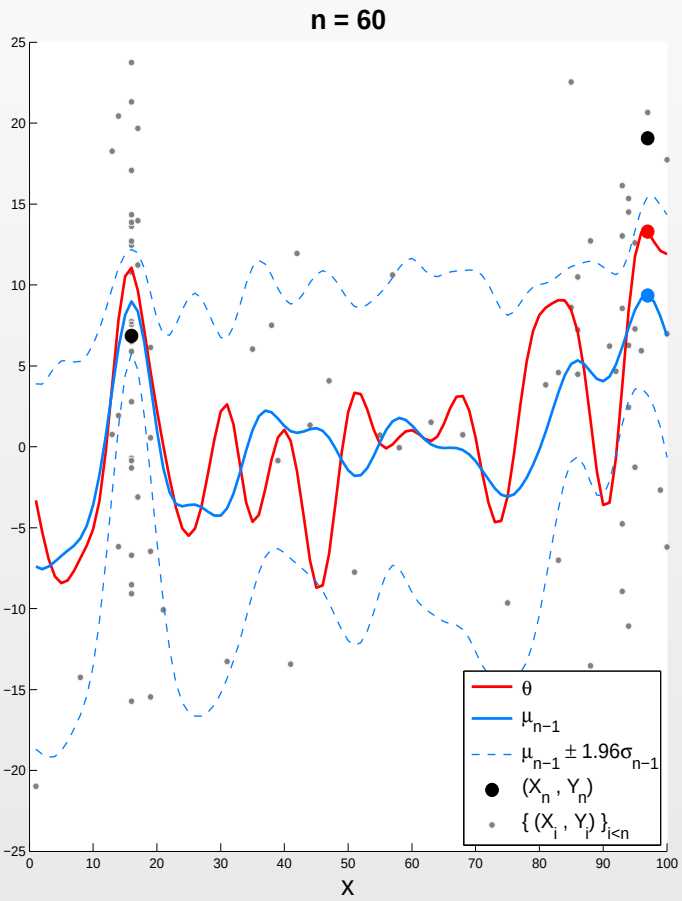


Illustration of the KG^2 Algorithm



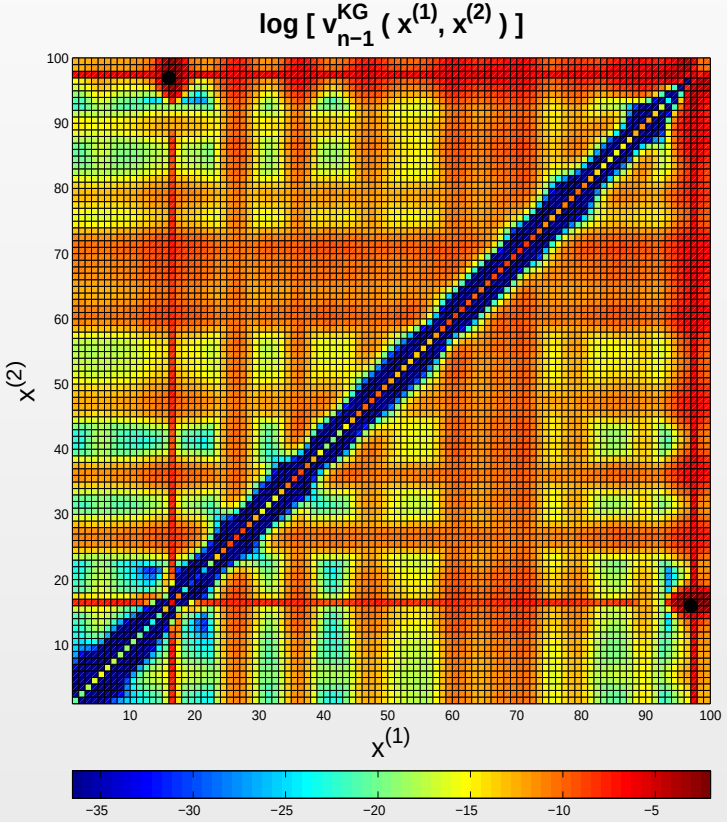
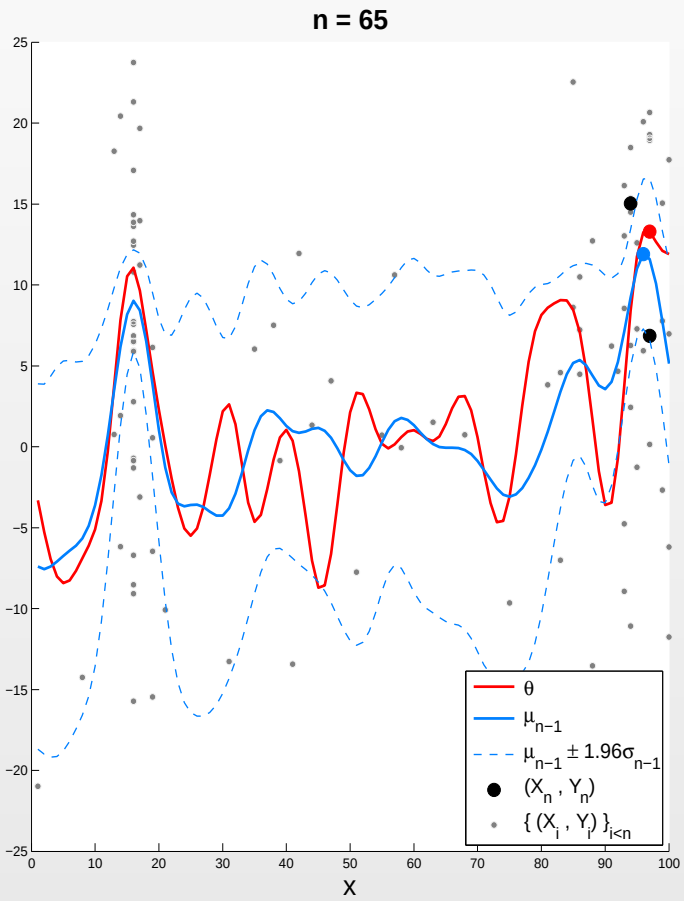


Illustration of the KG^2 Algorithm



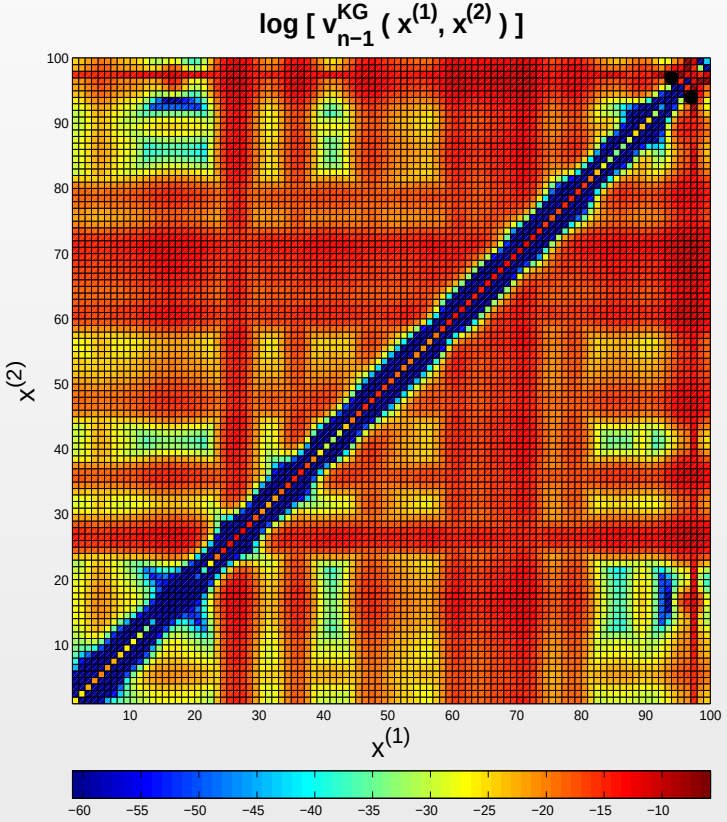
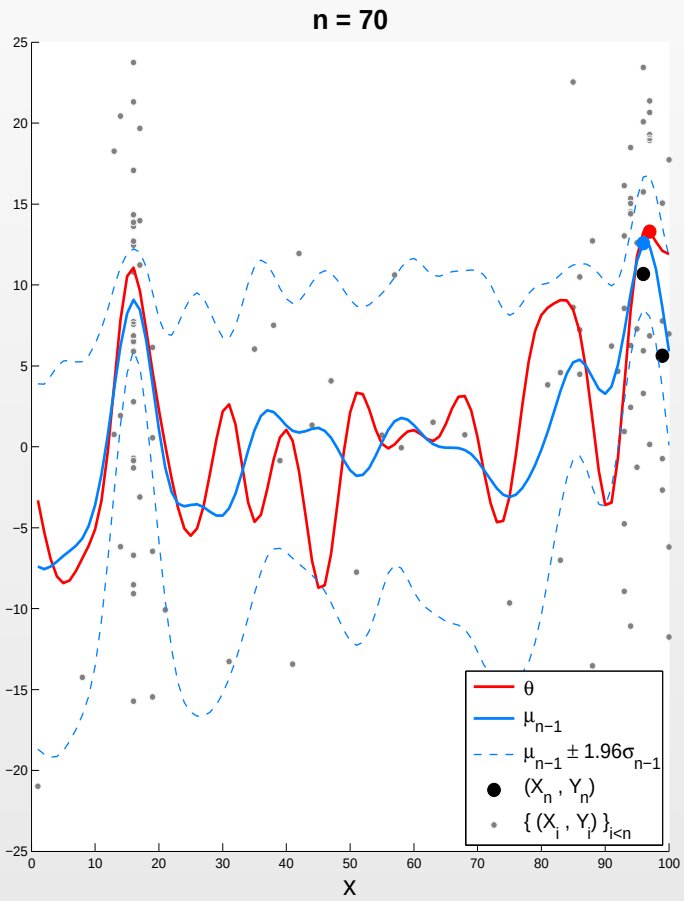


Illustration of the KG^2 Algorithm



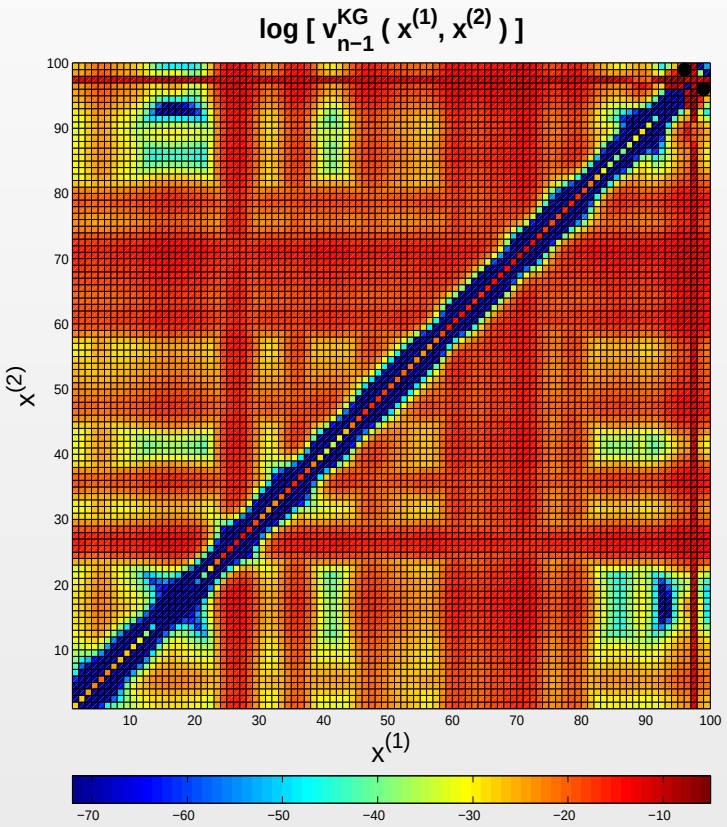
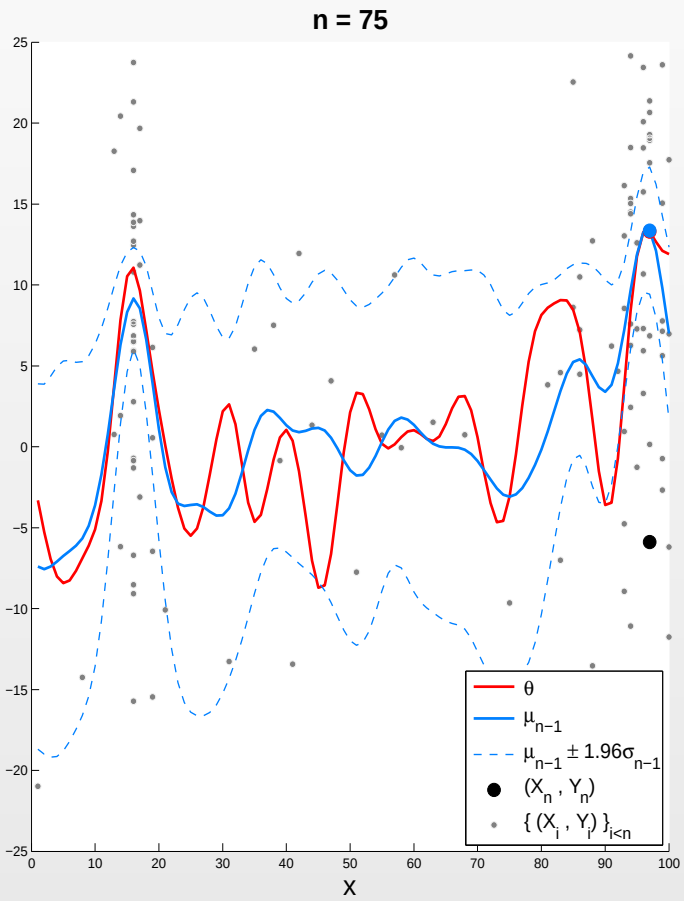


Illustration of the KG^2 Algorithm



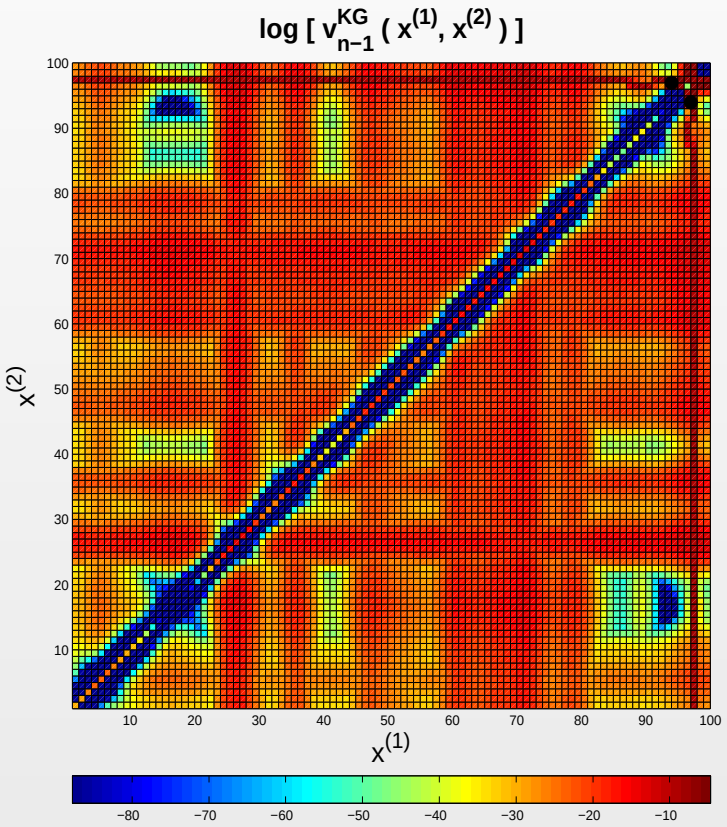
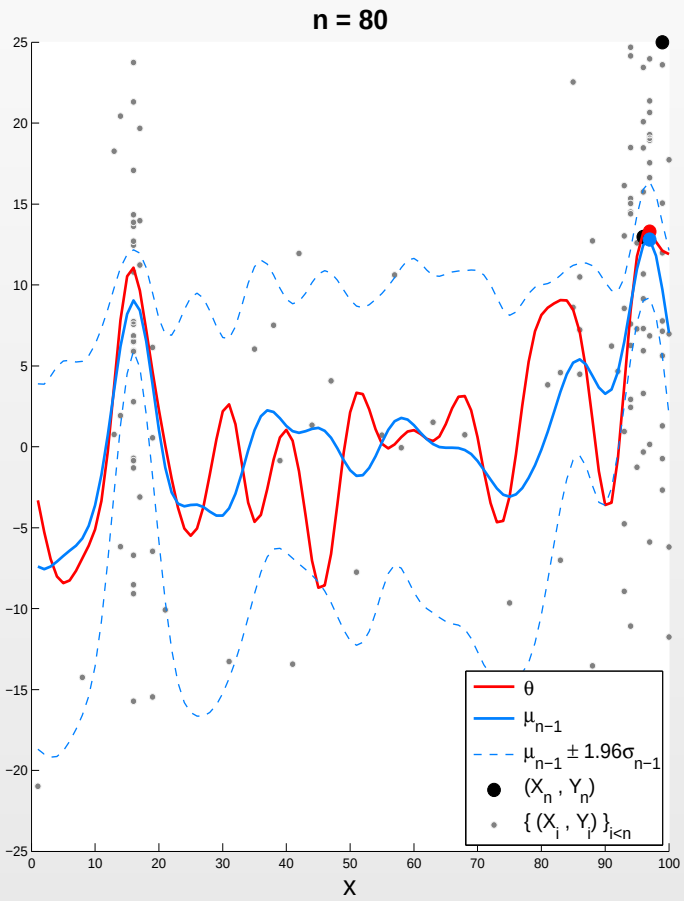


Illustration of the KG^2 Algorithm



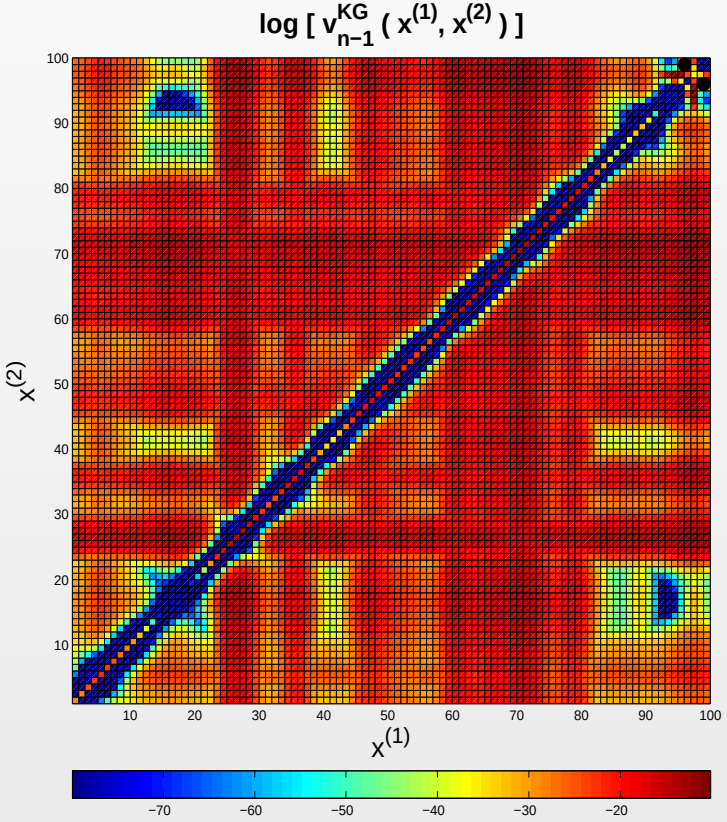
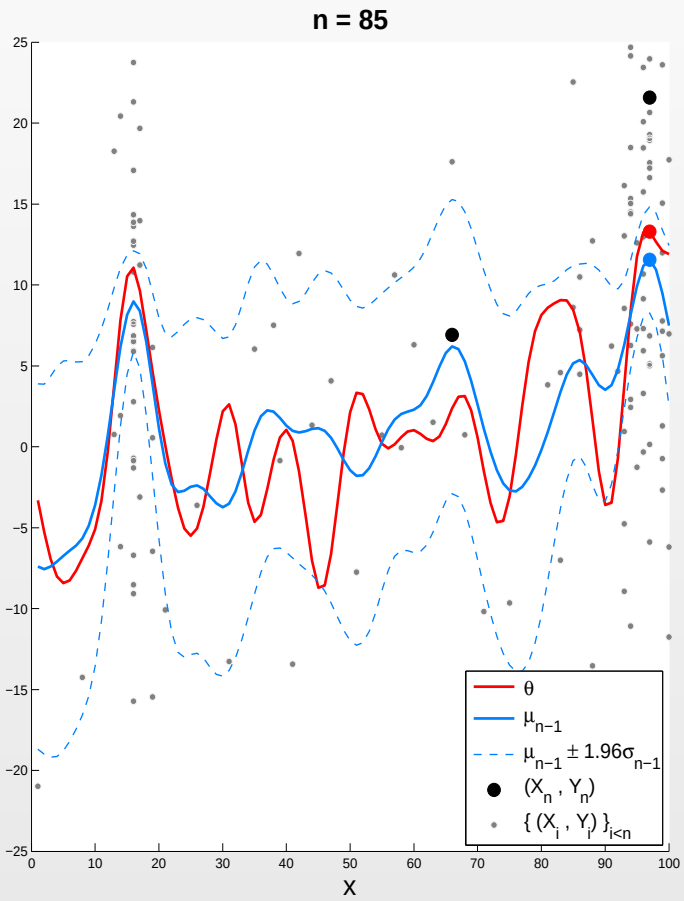


Illustration of the KG^2 Algorithm



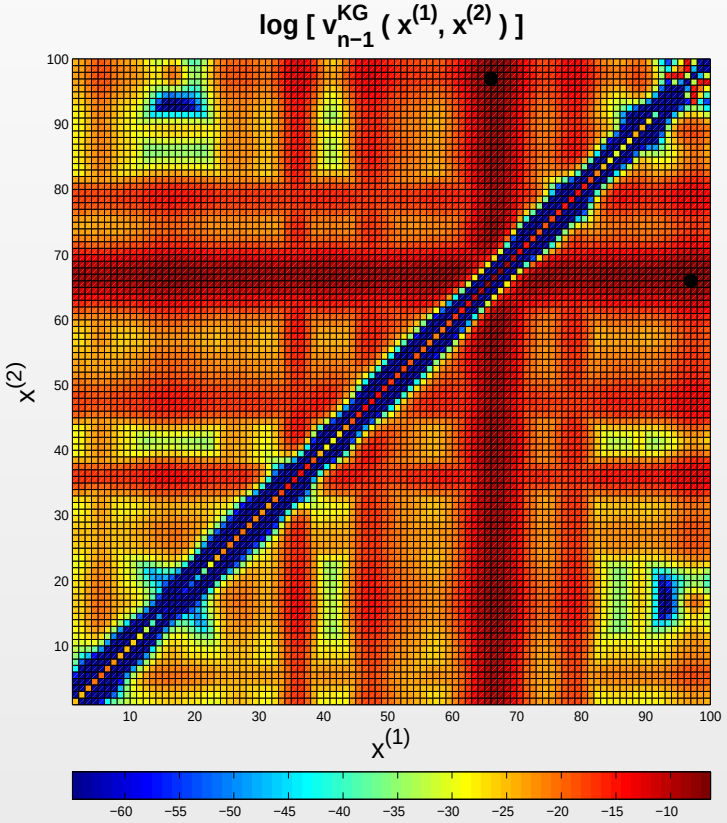
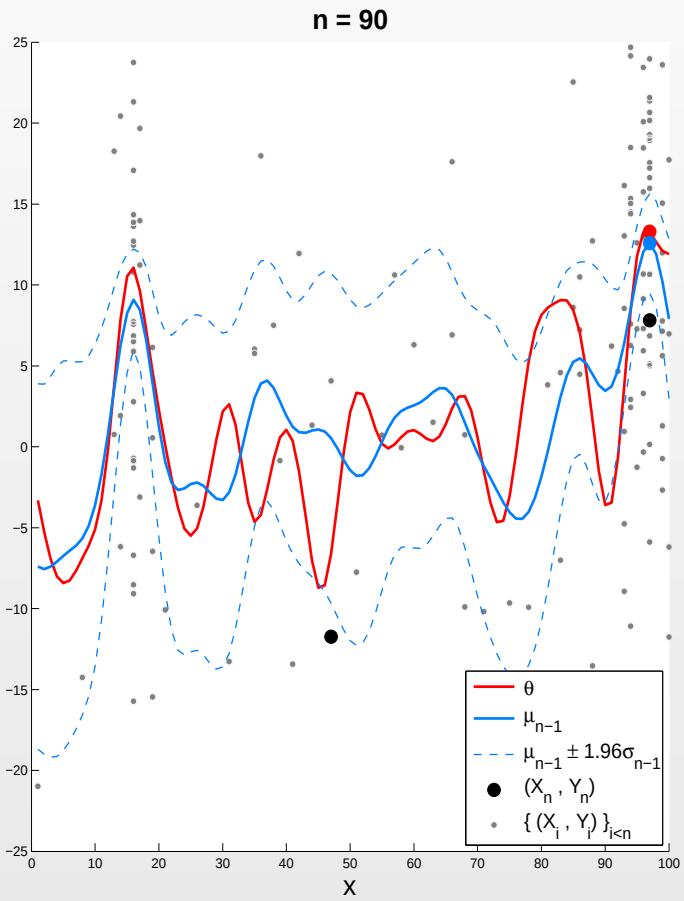


Illustration of the KG^2 Algorithm



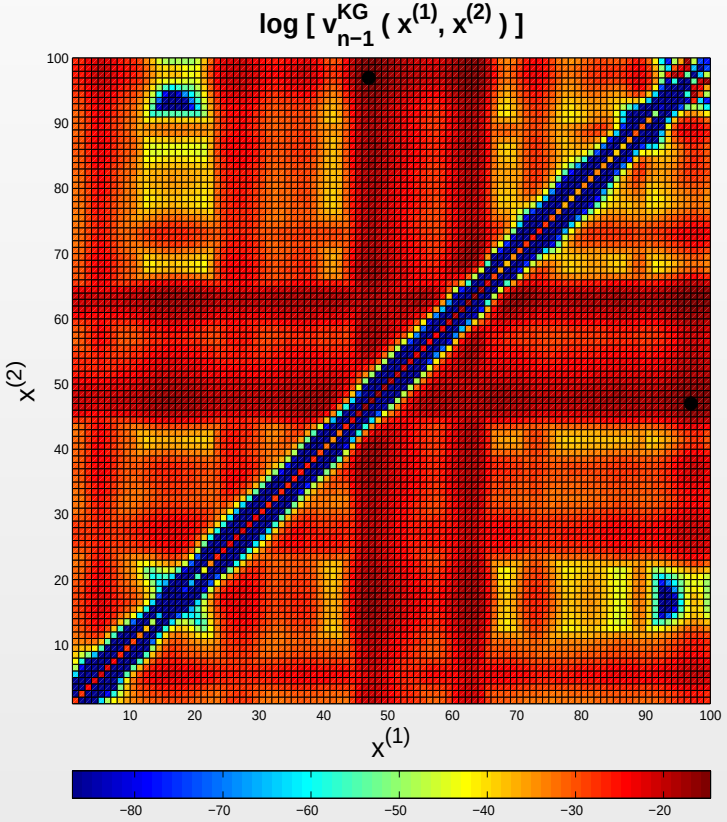
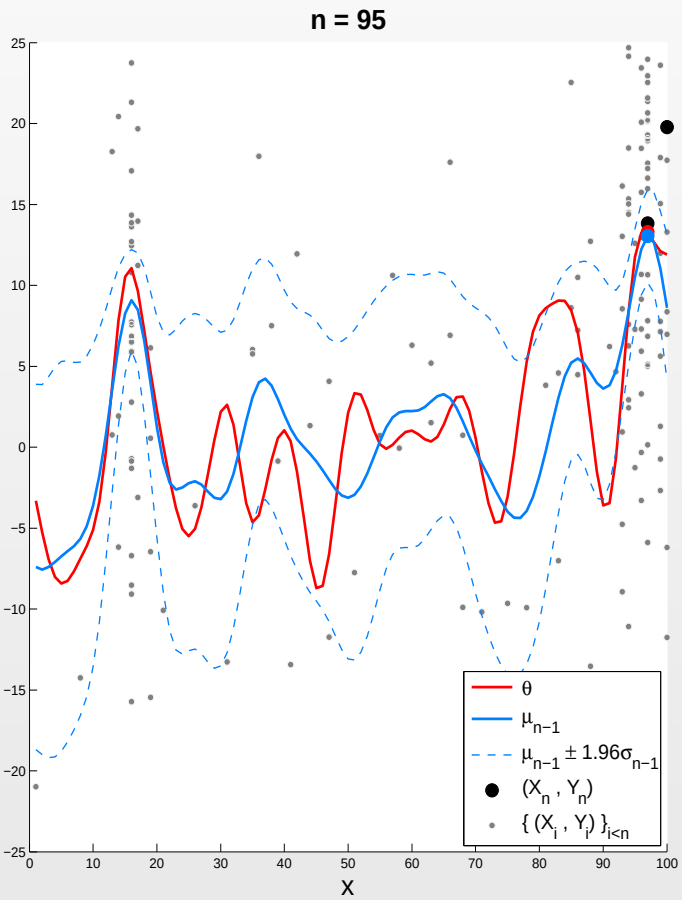


Illustration of the KG^2 Algorithm



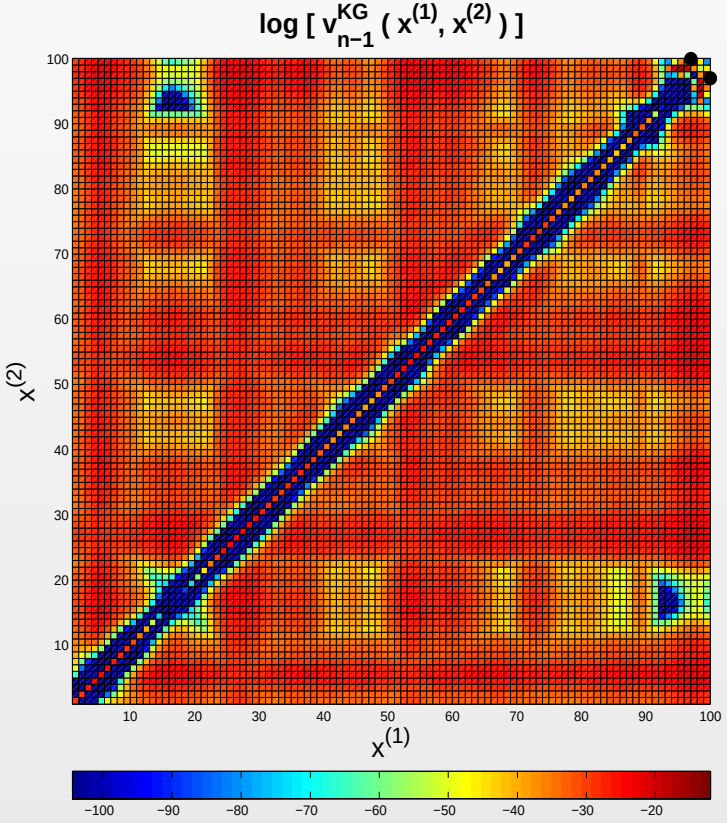
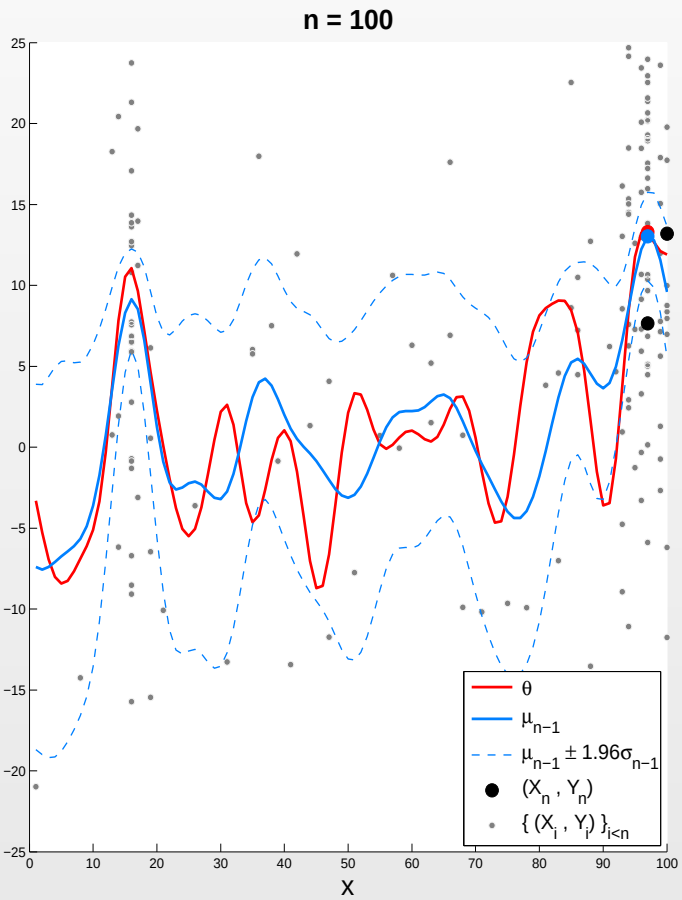
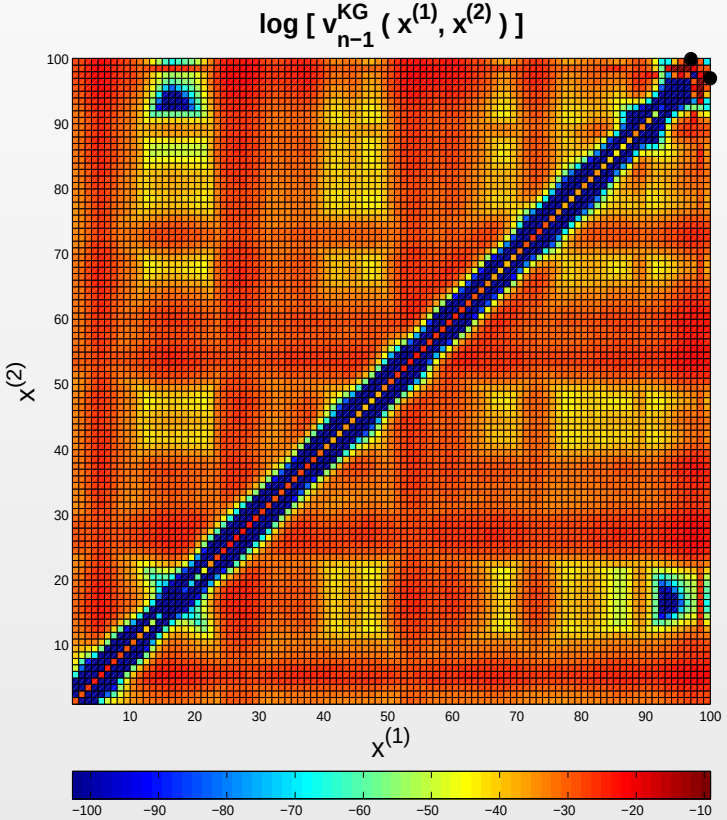


Illustration of the KG^2 Algorithm





Exploitation vs. Exploration

- KG factor is *bigger* when the posterior mean is *bigger*.
- KG factor is *bigger* when the posterior variance is *bigger*.

Exploitation vs. Exploration

- KG factor is *bigger* when the posterior mean is *bigger*.
- KG factor is *bigger* when the posterior variance is *bigger*.
- These two tendencies often push against each other, and our sampling algorithms manage to BALANCE them.

What if the Solution Space is BIG?

What if the Solution Space is BIG?

- VOI?

$$V_n(X) = \mathbb{E}_n\left[\max_x \mu_{n+1,x} \,\middle|\, X_{n+1} = X\right] - \max_x \mu_{n,x}$$
$$\approx \mathbb{E}_n\left[\max_{x \in \text{sss}} \mu_{n+1,x} \,\middle|\, X_{n+1} = X\right] - \max_{x \in \text{sss}} \mu_{n,x},$$

where sss = some small subset.

What if the Solution Space is BIG?

- VOI?

$$V_n(X) = \mathbb{E}_n\left[\max_x \mu_{n+1,x} \,\middle|\, X_{n+1} = X\right] - \max_x \mu_{n,x}$$
$$\approx \mathbb{E}_n\left[\max_{x \in \text{sss}} \mu_{n+1,x} \,\middle|\, X_{n+1} = X\right] - \max_{x \in \text{sss}} \mu_{n,x},$$

where sss = some small subset.

- Sampling Decision?

$$\operatorname*{argmax}_{X \in \cdots} \nu_n^{KG}(X) \Leftarrow \text{multi-start gradient decent}$$

What if the Solution Space is BIG?

- VOI?

$$V_n(X) = \mathbb{E}_n\left[\max_x \mu_{n+1,x} \,\middle|\, X_{n+1} = X\right] - \max_x \mu_{n,x}$$
$$\approx \mathbb{E}_n\left[\max_{x\in \text{sss}} \mu_{n+1,x} \,\middle|\, X_{n+1} = X\right] - \max_{x\in \text{sss}} \mu_{n,x},$$

where sss = some small subset.

- Sampling Decision?

$$\operatorname*{argmax}_{X\in\cdots} \nu_n^{KG}(X) \Leftarrow \text{multi-start gradient decent}$$

- Implementation Decision?

$$\operatorname*{argmax}_x \mu_{n,x} \approx \operatorname*{argmax}_{x\in \text{SAMPLED}} \mu_{n,x}$$

Why Knowledge Gradient?

We have replaced one optimization problem: $\max_x \theta_x$ with many optimization problems: $\max_X \nu_n^{KG}(X)$, for $n = 1, 2, \ldots$

WHY is this a good thing?

Why Knowledge Gradient?

We have replaced one optimization problem: $\max_x \theta_x$ with many optimization problems: $\max_X \nu_n^{KG}(X)$, for $n = 1, 2, \ldots$

WHY is this a good thing?

- Evaluating θ_x is expensive (minutes, hours, days), and derivative information is unavailable.

Why Knowledge Gradient?

We have replaced one optimization problem: $\max_x \theta_x$ with many optimization problems: $\max_X \nu_n^{KG}(X)$, for $n = 1, 2, \ldots$

WHY is this a good thing?

- Evaluating θ_x is expensive (minutes, hours, days), and derivative information is unavailable.
- Evaluating $\nu_n^{KG}(X)$ is quick (microseconds), and derivative information is available.

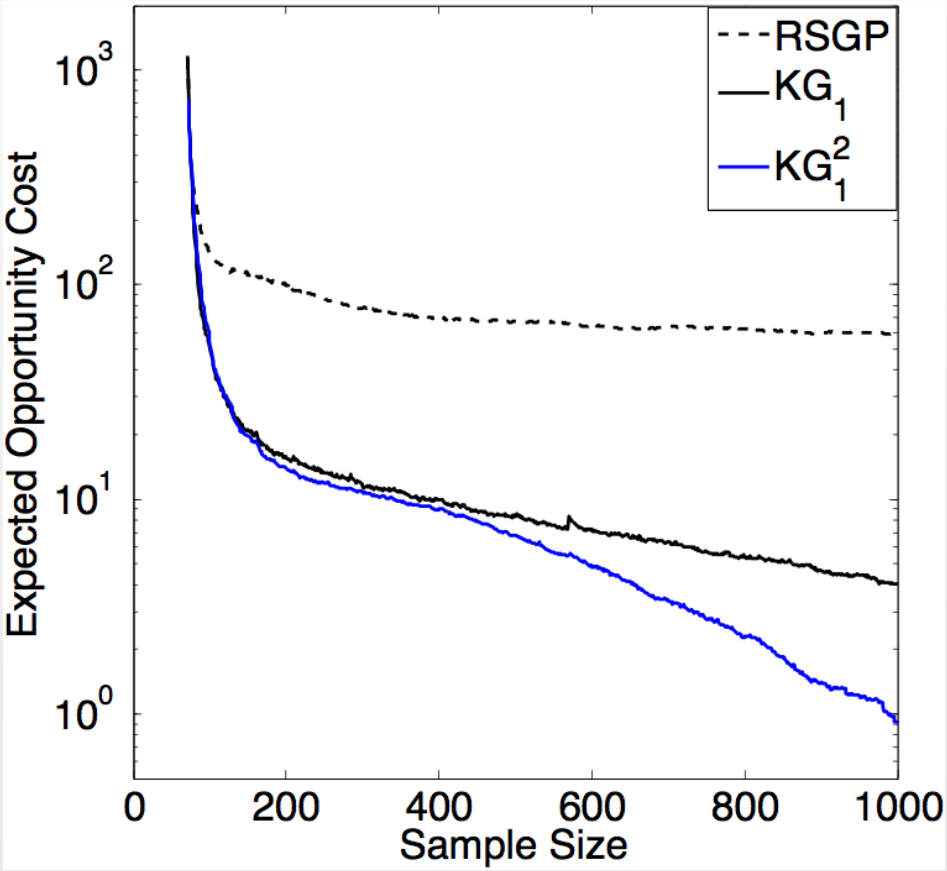
Why Knowledge Gradient?

We have replaced one optimization problem: $\max_x \theta_x$ with many optimization problems: $\max_X \nu_n^{KG}(X)$, for $n = 1, 2, \ldots$

WHY is this a good thing?

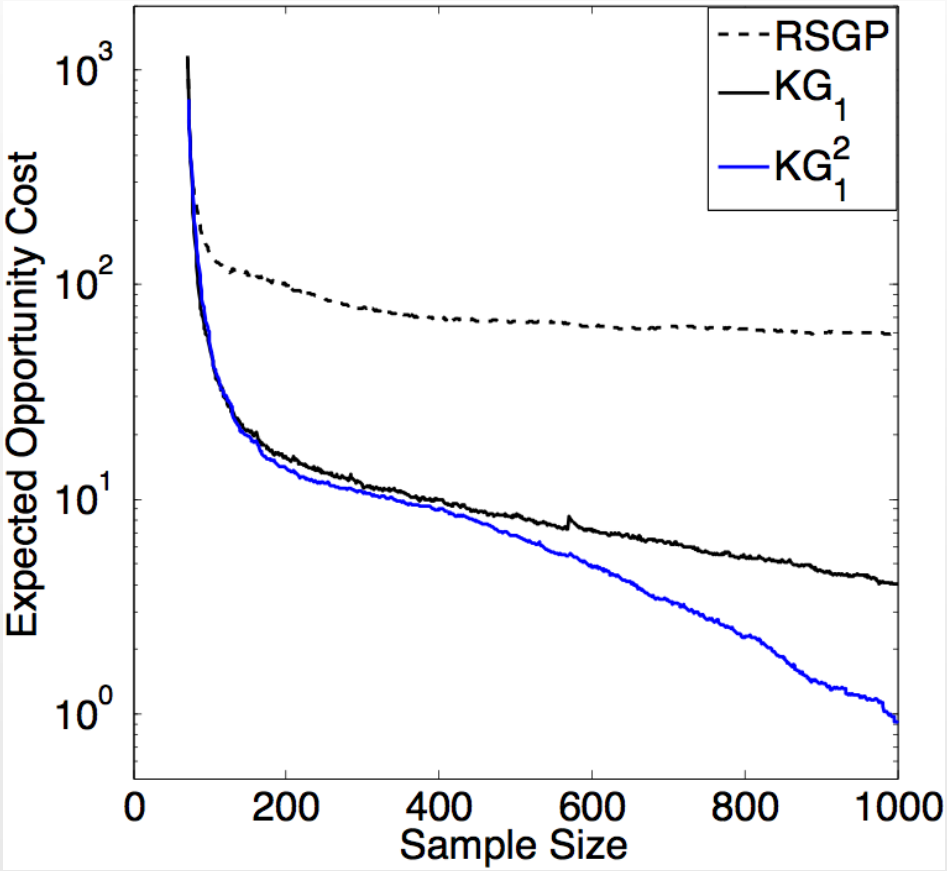
- Evaluating θ_x is expensive (minutes, hours, days), and derivative information is unavailable.
- Evaluating $\nu_n^{KG}(X)$ is quick (microseconds), and derivative information is available.
- We spend longer to decide where to take each sample, but require much fewer samples to find a good solution!

Rosenbrock with 10^6 alternatives



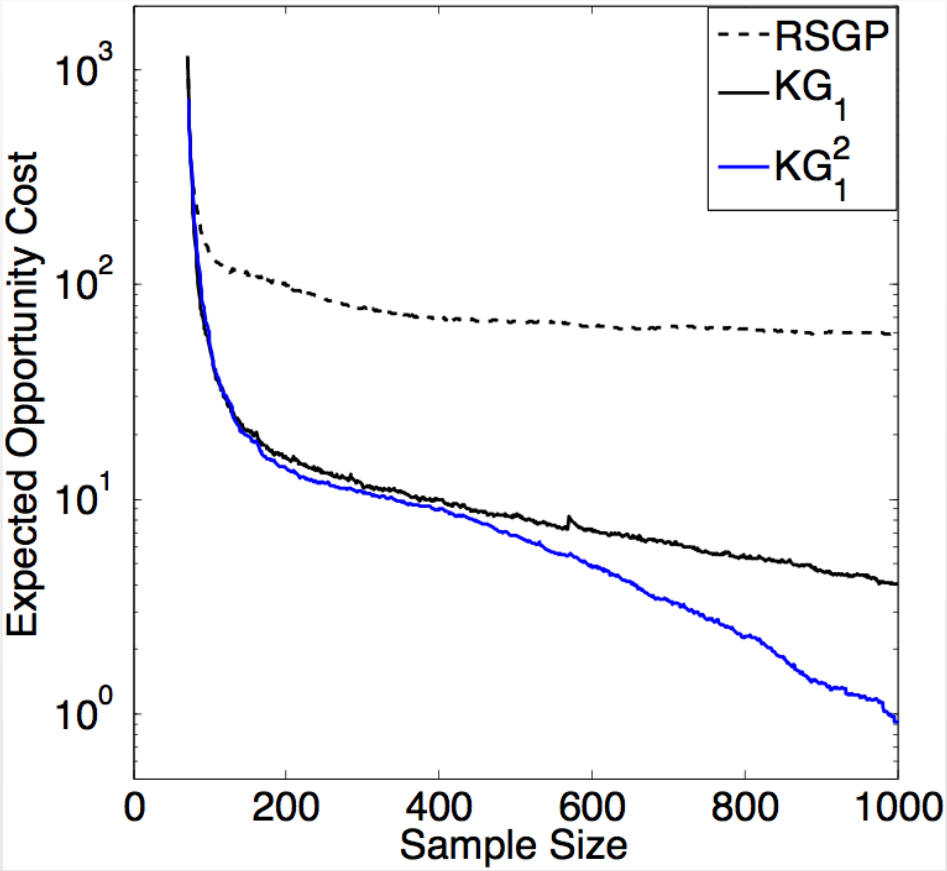
- RSGP: random search with a correlated (Gaussian Process) prior.
- *Opportunity Cost* $= \max_x \theta_x - \theta_{x_n^*}$, where $x_n^* = \text{argmax}_{x \in \text{sampled}} \mu_{n,x}$.

Rosenbrock with 10^6 alternatives



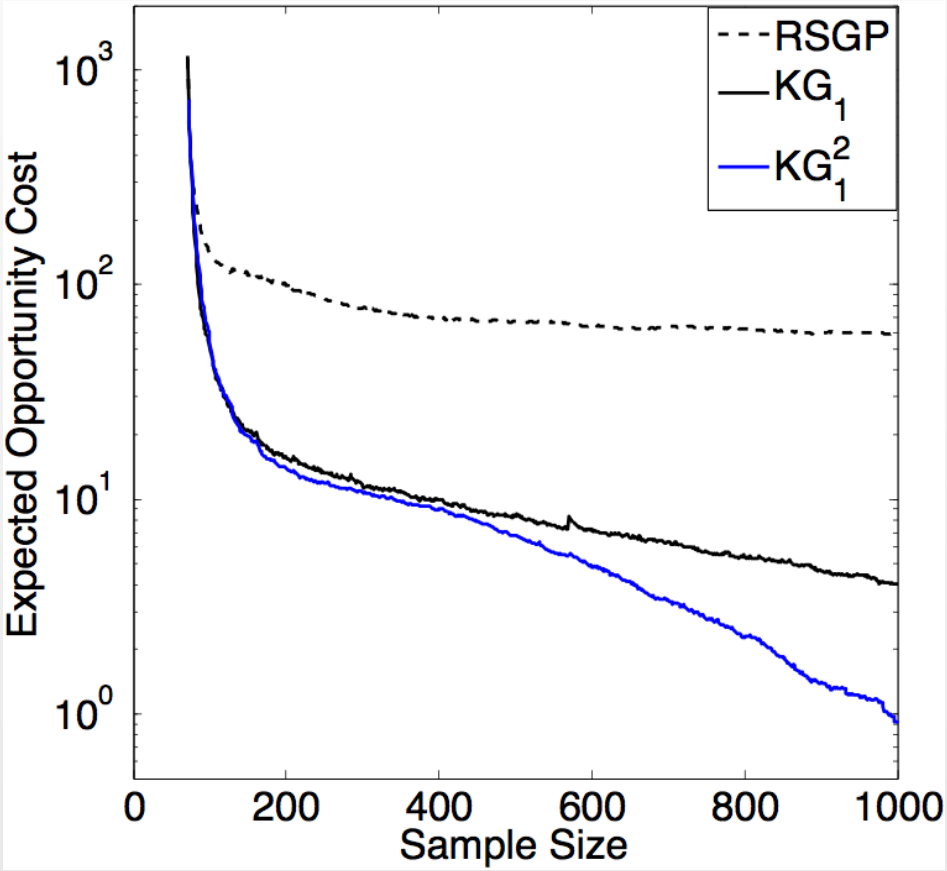
- RSGP: random search with a correlated (Gaussian Process) prior.
- *Opportunity Cost* $= \max_x \theta_x - \theta_{x_n^*}$, where $x_n^* = \text{argmax}_{x \in \text{sampled}} \mu_{n,x}$.
- The GP prior & the sampling covariances (assumed unknown) are estimated by *Maximum Likelihood*.

Rosenbrock with 10^6 alternatives



- RSGP: random search with a correlated (Gaussian Process) prior.
- *Opportunity Cost* $= \max_x \theta_x - \theta_{x_n^*}$, where $x_n^* = \text{argmax}_{x \in \text{sampled}} \mu_{n,x}$.
- The GP prior & the sampling covariances (assumed unknown) are estimated by *Maximum Likelihood*.
- $\mathrm{KG}^2 \succ \mathrm{KG} \succ \mathrm{RS}$

Rosenbrock with 10^6 alternatives

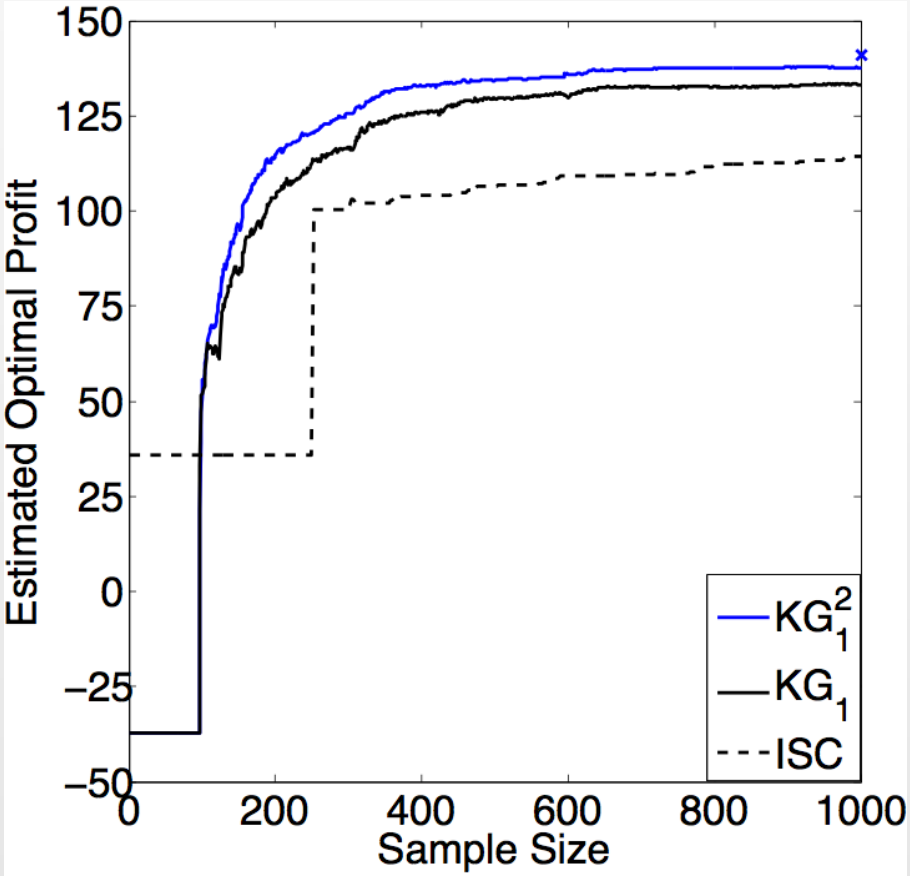


- RSGP: random search with a correlated (Gaussian Process) prior.
- *Opportunity Cost* $= \max_x \theta_x - \theta_{x_n^*}$, where $x_n^* = \text{argmax}_{x \in \text{sampled}} \mu_{n,x}$.
- The GP prior & the sampling covariances (assumed unknown) are estimated by *Maximum Likelihood*.

- $\mathrm{KG}^2 \succ \mathrm{KG} \succ \mathrm{RS}$
- correlated prior $\succ$ independent prior

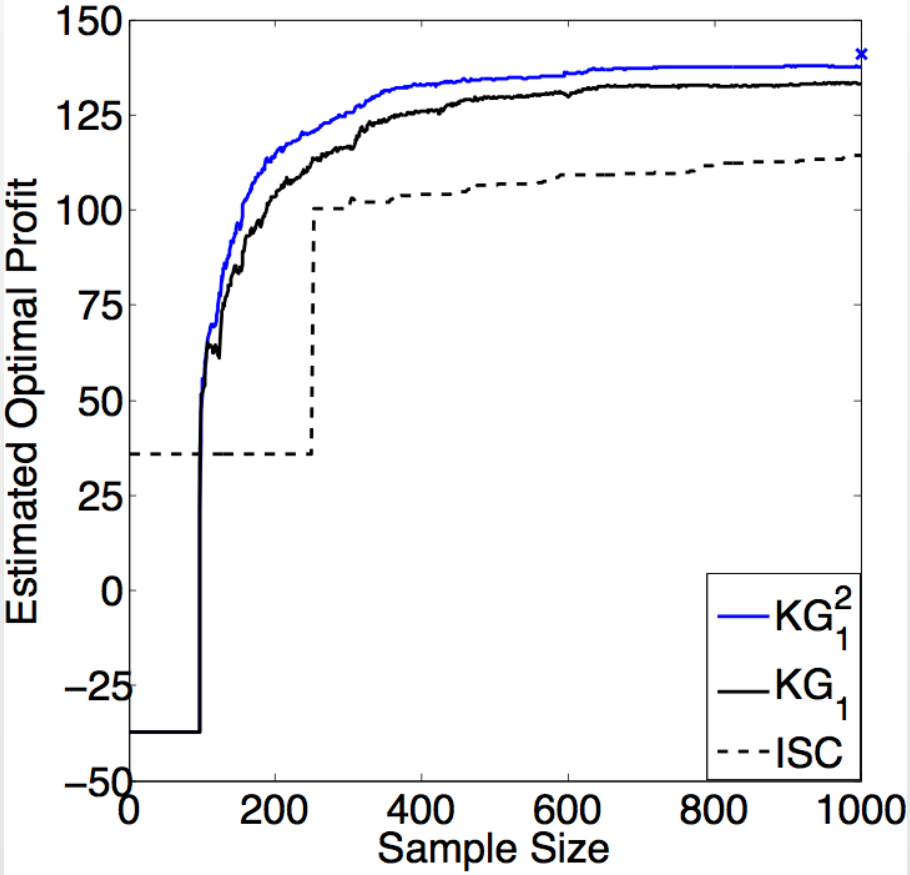
Assemble to Order problem with 21^8 alternatives

- Described in Hong et al. 2012, Hong & Nelson 2006.
- Industrial Strength COMPASS (ISC): Xu et al. 2010



Assemble to Order problem with 21^8 alternatives

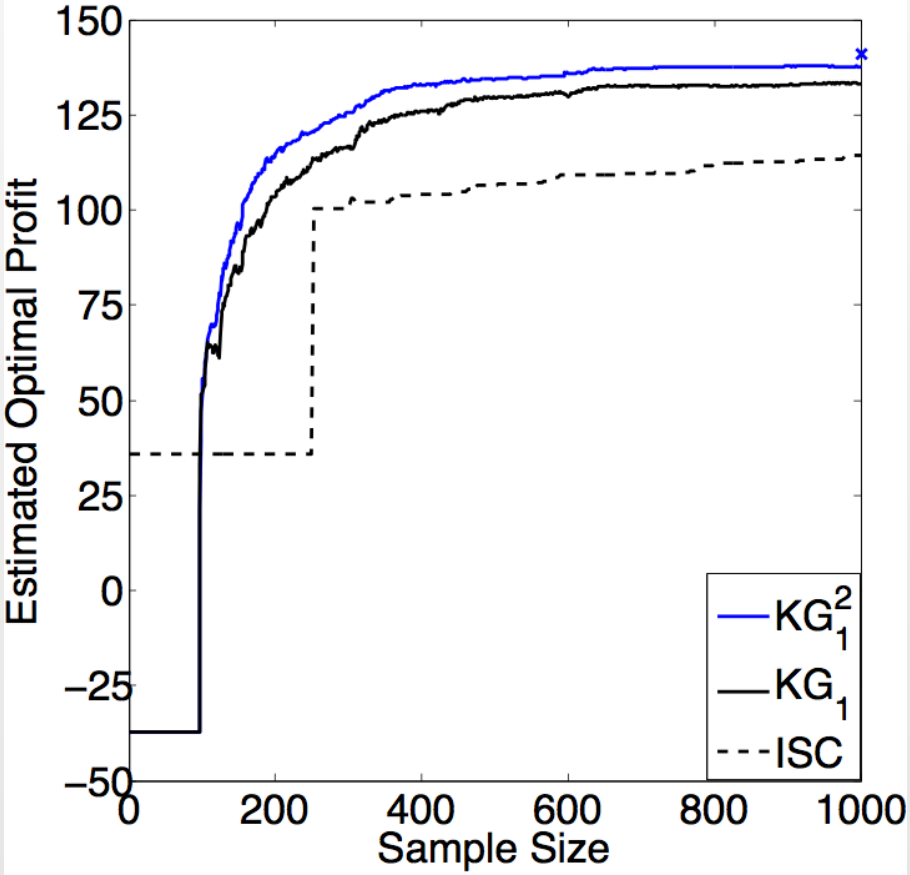
- Described in Hong et al. 2012, Hong & Nelson 2006.
- Industrial Strength COMPASS (ISC): Xu et al. 2010



KG/KG^2 spends 10 times longer than ISC to compute the 1000 sampling decisions.

Assemble to Order problem with 21^8 alternatives

- Described in Hong et al. 2012, Hong & Nelson 2006.
- Industrial Strength COMPASS (ISC): Xu et al. 2010



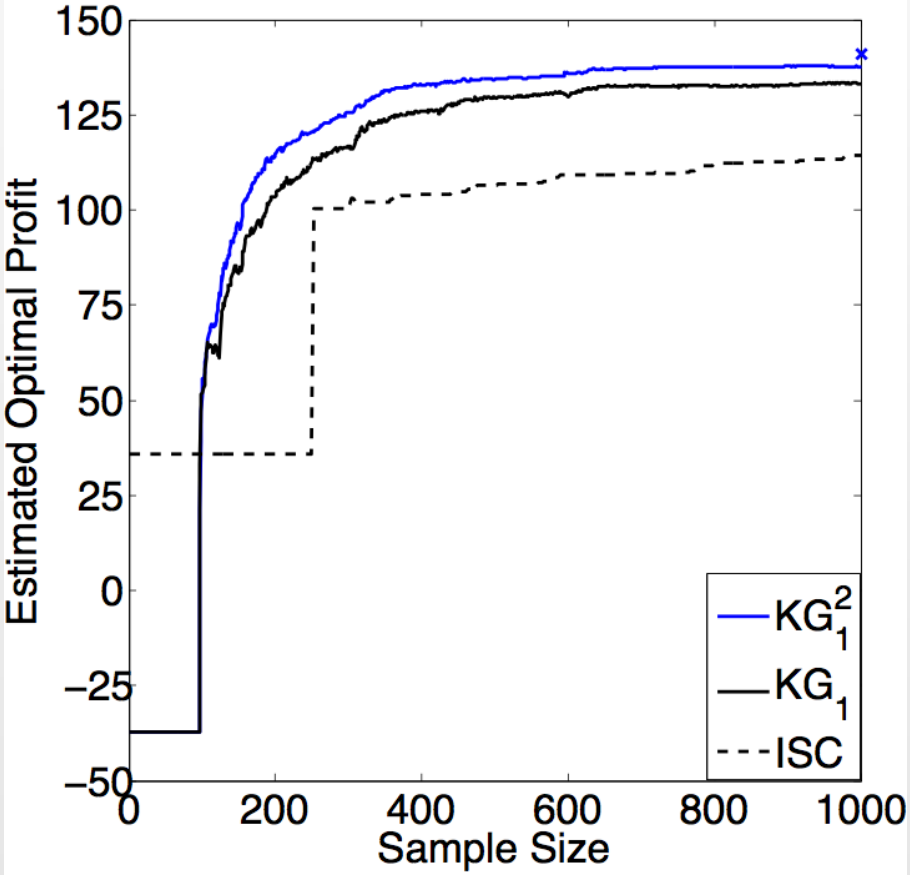
KG/KG^2 spends 10 times longer than ISC to compute the 1000 sampling decisions.

BUT

- ISC takes 1000+ samples / 20+ minutes on average to reach an average profit of 115.

Assemble to Order problem with 21^8 alternatives

- Described in Hong et al. 2012, Hong & Nelson 2006.
- Industrial Strength COMPASS (ISC): Xu et al. 2010



KG/KG^2 spends 10 times longer than ISC to compute the 1000 sampling decisions.

BUT

- ISC takes 1000+ samples / 20+ minutes on average to reach an average profit of 115.
- KG requires 300− samples / 10− minutes on average to reach 115.
- KG^2 requires 220− samples / 6− minutes on average to reach 115.

Ideal Scope: Expensive Function Evaluation

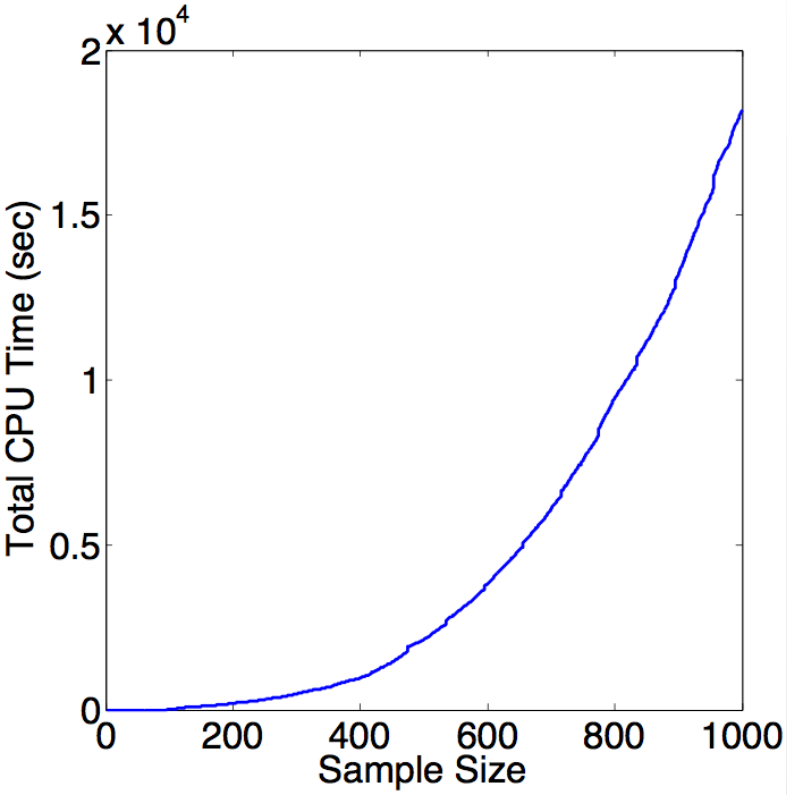


Figure: CPU time spent in a sample path of KG^2, on the ATO problem.

KG/KG^2 are less suitable for problems in which simulation can be performed very quickly.

Ideal Scope: Expensive Function Evaluation

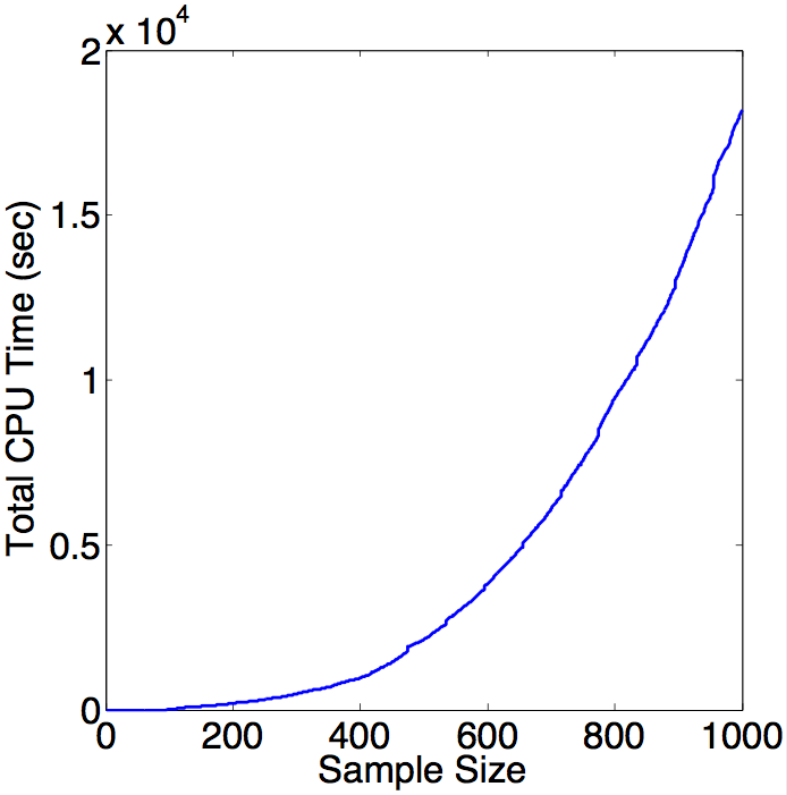


Figure: CPU time spent in a sample path of KG^2, on the ATO problem.

$\mathrm{KG}/\mathrm{KG}^2$ are less suitable for problems in which simulation can be performed very quickly.

When samples come from a complex, long-running simulator, however,

- the substantial computational consumption in deciding where to sample is relatively unimportant;
- algorithms like KG^2 that find good solutions in fewer samples also work well in terms of overall computation time.

Conclusions

- We take advantage of both
  - correlated prior beliefs & correlated sampling distributions

  in a *Bayesian value of information* framework, which brings a distinct benefit for optimization via simulation.

Conclusions

- We take advantage of both
  - correlated prior beliefs & correlated sampling distributions

  in a *Bayesian value of information* framework, which brings a distinct benefit for optimization via simulation.

- We give easy-to-verify conditions under which almost sure convergence to the optimal solution can be guaranteed.

Conclusions

- We take advantage of both
  - correlated prior beliefs & correlated sampling distributions

  in a *Bayesian value of information* framework, which brings a distinct benefit for optimization via simulation.

- We give easy-to-verify conditions under which almost sure convergence to the optimal solution can be guaranteed.

- Our algorithms demonstrate superior efficiency compared to others
  - in problems with combinatorially large solution spaces, and
  - when samples are moderately to very computationally expensive.

  Run times are a low order polynomial in the number of samples observed, rather than a low order polynomial in the size of the solution space!

THANK YOU!

References

- Brochu, E., V. Cora, and N. de Freitas. 2009, November. "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning". Technical Report TR-2009-23, Department of Computer Science, University of British Columbia.
- Chick, S., and P. Frazier. 2009a, December. "The Conjunction of the Knowledge Gradient and Economic Approach to Simulation Selection". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 528-539. Piscataway, New Jersey: Institute of Electrical and Electronics Engineerings, Inc.
- Chick, S., and P. Frazier. 2009b. "Sequential Sampling for Selection with ESP". in review.
- Chick, S., and K. Inoue. 2001. "New Procedures to Select the Best Simulated System Using Common Random Numbers". *Management Science*, 47 (8): 1133-1149.

References

- Clark, G., and W. Yang. 1986, December. "A Bonferroni selection procedure when using common random numbers with unknown variances". In *Proceedings of the 1986 Winter Simulation Conference*, edited by J. Wilson, J. Henrikson, and S. Robert, 313 - 315. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Frazier, P., and W. Powell. 2008, December. "The Knowledge-Gradient Stopping Rule for Ranking and Selection". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 305-312. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Frazier, P., and W. Powell. 2010. "Paradoxes in Learning and the Marginal Value of Information". *Decision Analysis* 7 (4): 378-403.
- Frazier, P., W. Powell, and S. Dayanik. 2008. "A Knowledge Gradient Policy for Sequential Information Collection". *SIAM Journal on Control and Optimization* 47 (5): 2410-2439.

References

- Frazier, P., W. Powell, and S. Dayanik. 2009. "The Knowledge Gradient Policy for Correlated Normal Beliefs". *INFORMS Journal on Computing* 21 (4): 599-613.
- Fu, M. C., J.-Q. Hu, C.-H. Chen, and X. Xiong. 2007. "Simulation Allocation for Determining the Best Design in the Presence of Correlated Sampling". *INFORMS J. on Computing*, 19 (1): 101-111.
- Gelman, A., J. Carlin, H. Stern, and D. Rubin. 2004. *Bayesian data analysis*. second ed. Boca Raton, FL: CRC Press.
- Gupta, S., and K. Miescke. 1996. "Bayesian look ahead one-stage sampling allocations for selection of the best population". *Journal of statistical planning and inference* 54 (2): 229-244.
- Howard, R. 1966. "Information Value Theory". *IEEE Trans. Systems Science and Cybernetics* 2 (1): 22-26.

References

- Kim, S. 2005. "Comparison with a standard via fully sequential procedures". *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 15 (2): 155-174.
- Nakayama, M. 2000. "Multiple comparisons with the best using common random numbers for steady-state simulations". *Journal of Statistical Planning and Inference*, 85 (1-2): 37-48.
- Negoescu, D., P. Frazier, and W. Powell. 2011. "The Knowledge Gradient Algorithm for Sequencing Experiments in Drug Discovery". *INFORMS Journal on Computing*, 23 (3): 346-363.
- Nelson, B., and F. Matejcik. 1995. "Using Common Random Numbers for Indifference-Zone Selection and Multiple Comparisons in Simulation". *Manage- ment Science*, 41 (12): 1935-1945.
- Rasmussen, C., and C. Williams. 2006. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.

References

- Ryzhov, I., P. Frazier, and W. Powell. 2010. "On the robustness of a one-period look-ahead policy in multi-armed bandit problems". *Procedia Computer Science* 1 (1): 1635-1644.
- Scott, W., P. Frazier, and W. Powell. 2011. "The Correlated Knowledge Gradient for Simulation Optimization of Continuous Parameters Using Gaussian Process Regression". *SIAM Journal on Optimization* 21: 996-1026.
- Yang, W., and B. Nelson. 1991. "Using Common Random Numbers and Control Variates in Multiple-Comparison Procedures". *Operations Research*, 39 (4): 583-591.