CMA-ES : Une méthode d'optimisation sans dérivées pour des problèmes difficiles

Anne Auger
RandOpt, Inria et CMAP, Ecole Polytechnique

Rencontres de la Chaire Modélisation Mathématique et Biodiversité

6 mai 2025

Derivative-free / Gradient-free optimization

Minimize $f : x = (x_1, \ldots, x_n) \in \mathbb{R}^n \mapsto f(x)$

approach $x^\star$ such that $f(x^\star) \leq f(x)$ for all x

Derivative-free / Gradient-free optimization

Minimize $f : x = (x_1, \ldots, x_n) \in \mathbb{R}^n \mapsto f(x)$

approach $x^\star$ such that $f(x^\star) \leq f(x)$ for all x

Assume: derivatives or gradient of f **not available**

Derivative-free / Gradient-free optimization

Minimize $f : x = (x_1, \ldots, x_n) \in \mathbb{R}^n \mapsto f(x)$

approach $x^\star$ such that $f(x^\star) \leq f(x)$ for all x

Assume: derivatives or gradient of f **not available**

Zero-order black-box optimization

Derivative-free / Gradient-free optimization

Minimize $f : x = (x_1, \ldots, x_n) \in \mathbb{R}^n \mapsto f(x)$

approach $x^\star$ such that $f(x^\star) \leq f(x)$ for all x

Assume: derivatives or gradient of f **not available**

Zero-order black-box optimization

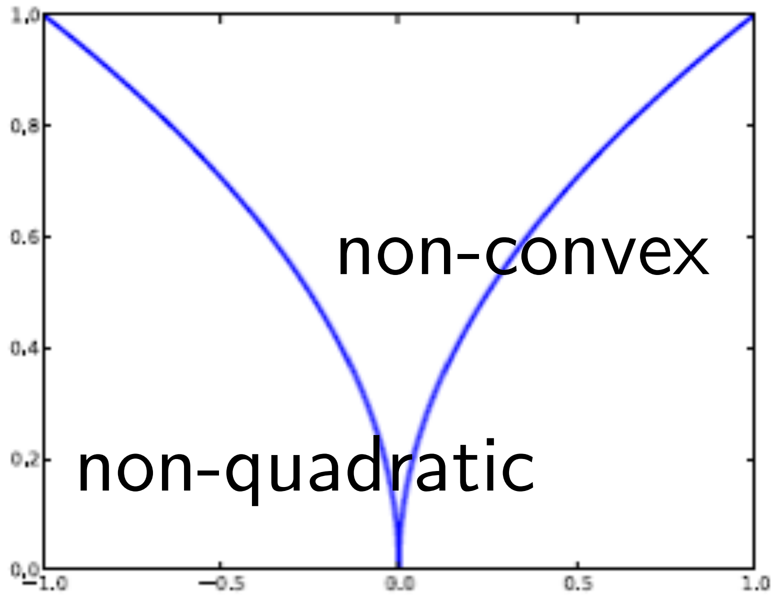
Many applications

(see also presentation Renaud)

Why stochastic numerical black-box optimization?

Difficulties inside the black-box ?

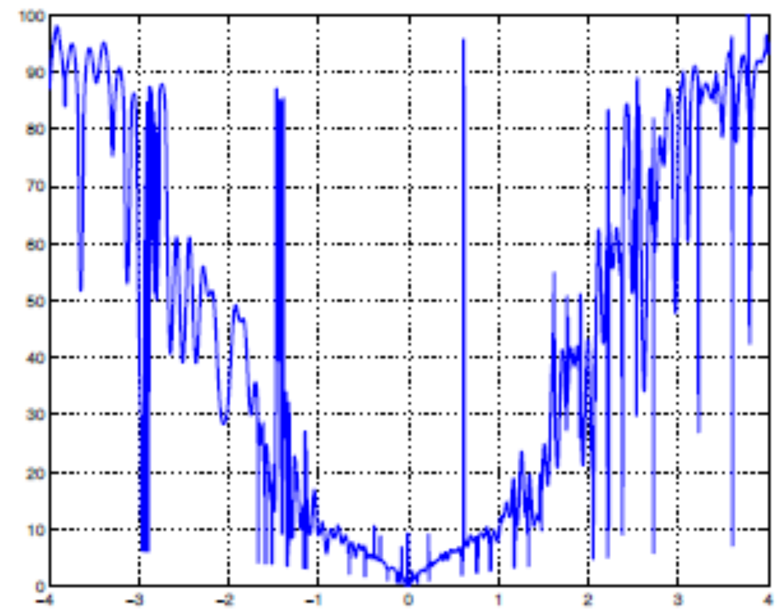
non-linear



non-convex

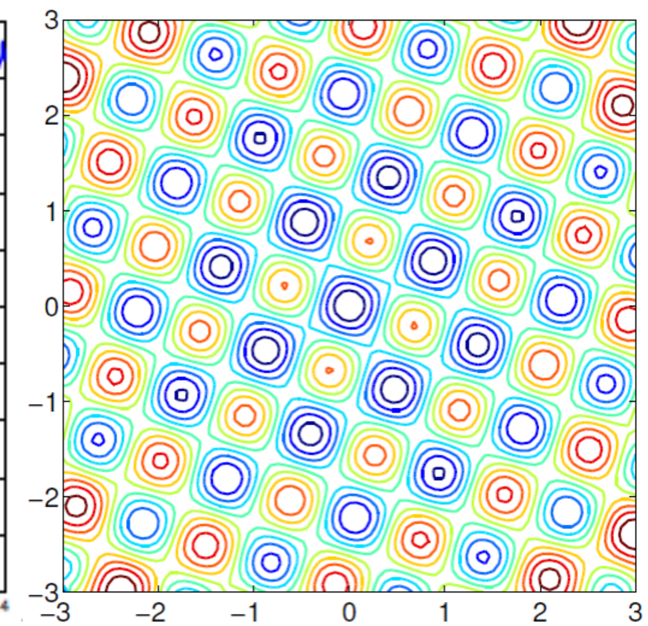
non-quadratic

non-differentiable



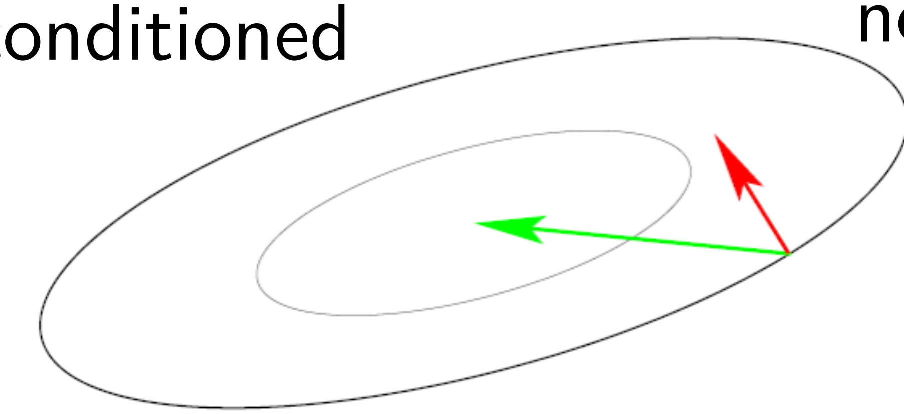
discontinuous / noisy

non-separable



multi-modal

ill-conditioned



gradient direction $-f'(\boldsymbol{x})^{\mathrm{T}}$

Newton direction $-\boldsymbol{H}^{-1}f'(\boldsymbol{x})^{\mathrm{T}}$

From Evolution Strategies (ES) to CMA-ES

Evolution Strategies (70s+)

randomized bio-inspired population-based algorithms

Rechenberg 🇩🇪 Schwefel 🇩🇪

From Evolution Strategies (ES) to CMA-ES

Evolution Strategies (70s+)

randomized bio-inspired population-based algorithms


Rechenberg 🇩🇪 Schwefel 🇩🇪

CMA-ES (1996/2001 - today)

More than **70 million downloads** of its two main Python implementations

From Evolution Strategies (ES) to CMA-ES

Evolution Strategies (70s+)

randomized bio-inspired population-based algorithms

Rechenberg 🇩🇪 Schwefel 🇩🇪

CMA-ES (1996/2001 - today)

not enough bio-inspired?

The first fathers [in Rechenberg's lab in Berlin 🇩🇪]

Hansen Ostermeier

More than **70 million downloads** of its two main Python implementations

From Evolution Strategies (ES) to CMA-ES

Evolution Strategies (70s+)
randomized bio-inspired population-based algorithms


Rechenberg 🇩🇪 Schwefel 🇩🇪

CMA-ES (1996/2001 - today)
not enough bio-inspired?

The first fathers [in Rechenberg's lab in Berlin 🇩🇪]


Hansen Ostermeier

🇫🇷 RandOpt: Auger/Hansen/Brockhoff+Chotard

🇯🇵 Akimoto

🇩🇪 Glasmachers

🇨🇦 Arnold

More than **70 million downloads** of its two main Python implementations

From a simple ES to CMA-ES

A simple Evolution Strategy: (1+1)-ES

$(1 + 1)$-ES

one parent

one offspring

Elitist selection: keep best among parent and offspring

A simple Evolution Strategy: (1+1)-ES

$t \in \mathbb{N}$ iteration index

A simple Evolution Strategy: (1+1)-ES

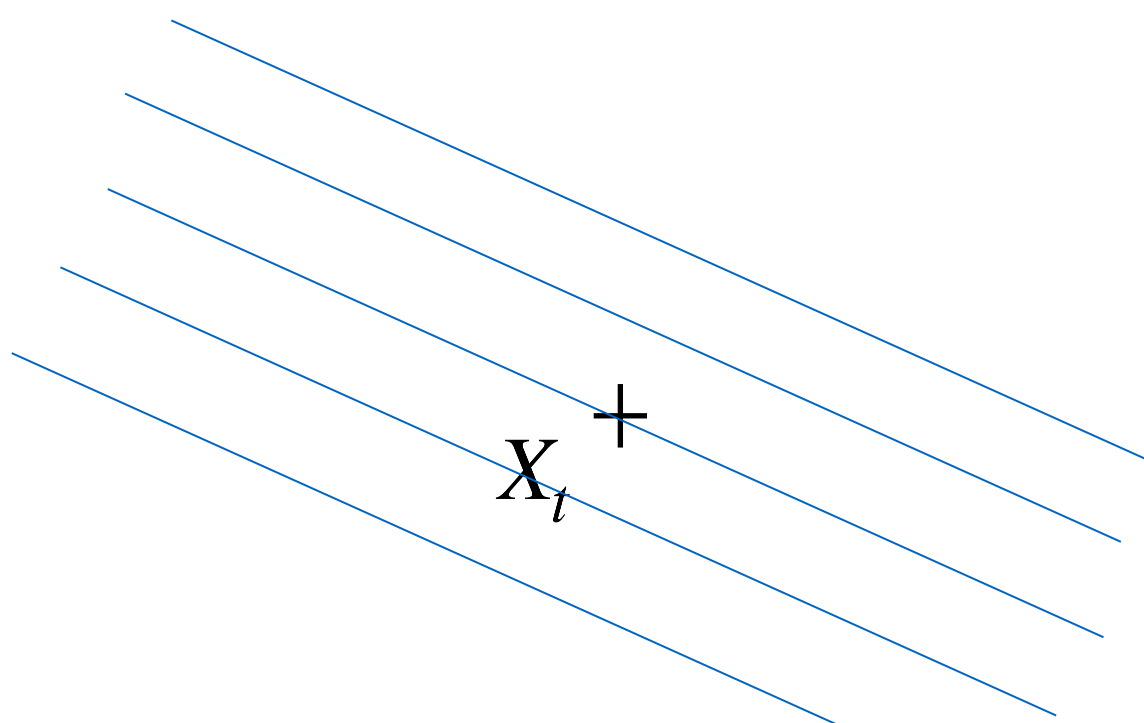
$t \in \mathbb{N}$ iteration index

$X_t \in \mathbb{R}^n$: parent at iteration t

$\sigma \in \mathbb{R}_>$: step-size

minimize $f : \mathbb{R}^n \to \mathbb{R}$

minimize $f : \mathbb{R}^2 \to \mathbb{R}$



A simple Evolution Strategy: (1+1)-ES

$t \in \mathbb{N}$ iteration index

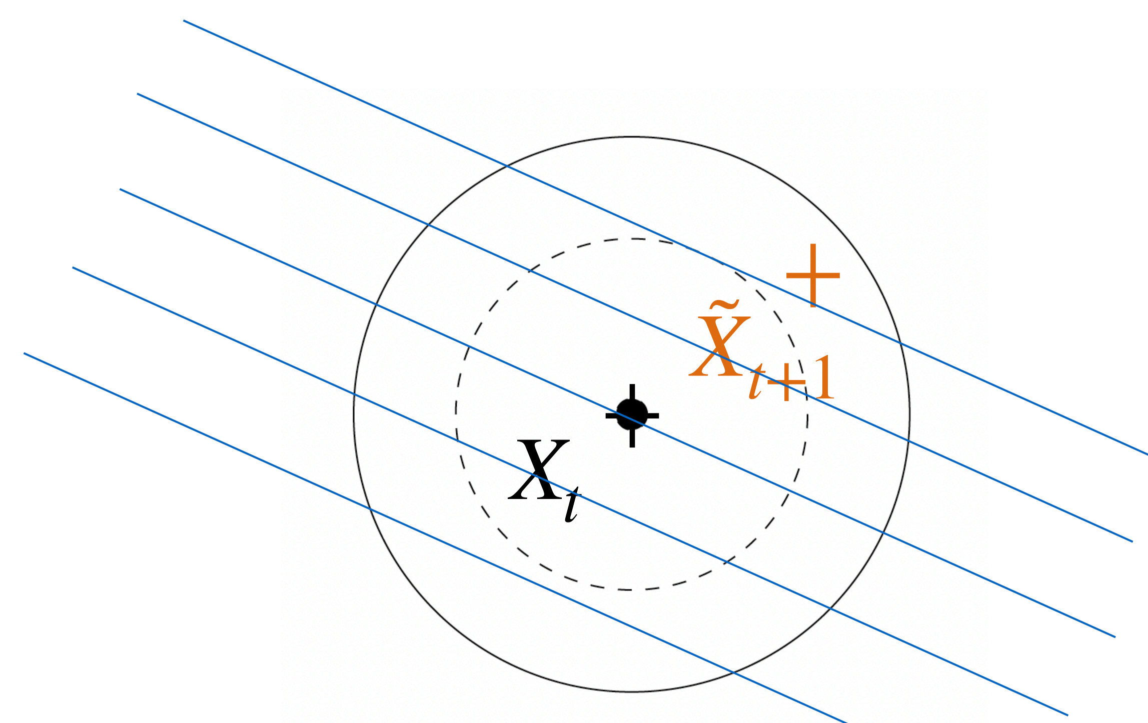
$X_t \in \mathbb{R}^n$: parent at iteration t

$\sigma \in \mathbb{R}_>$: step-size

$\tilde{X}_{t+1}$: offspring created via **mutation** of parent

$$\tilde{X}_{t+1} = X_t + \sigma \mathcal{N}(0, I_d)$$

minimize $f : \mathbb{R}^2 \to \mathbb{R}$



A simple Evolution Strategy: (1+1)-ES

$t \in \mathbb{N}$ iteration index

$X_t \in \mathbb{R}^n$: parent at iteration t

$\sigma \in \mathbb{R}_>$: step-size

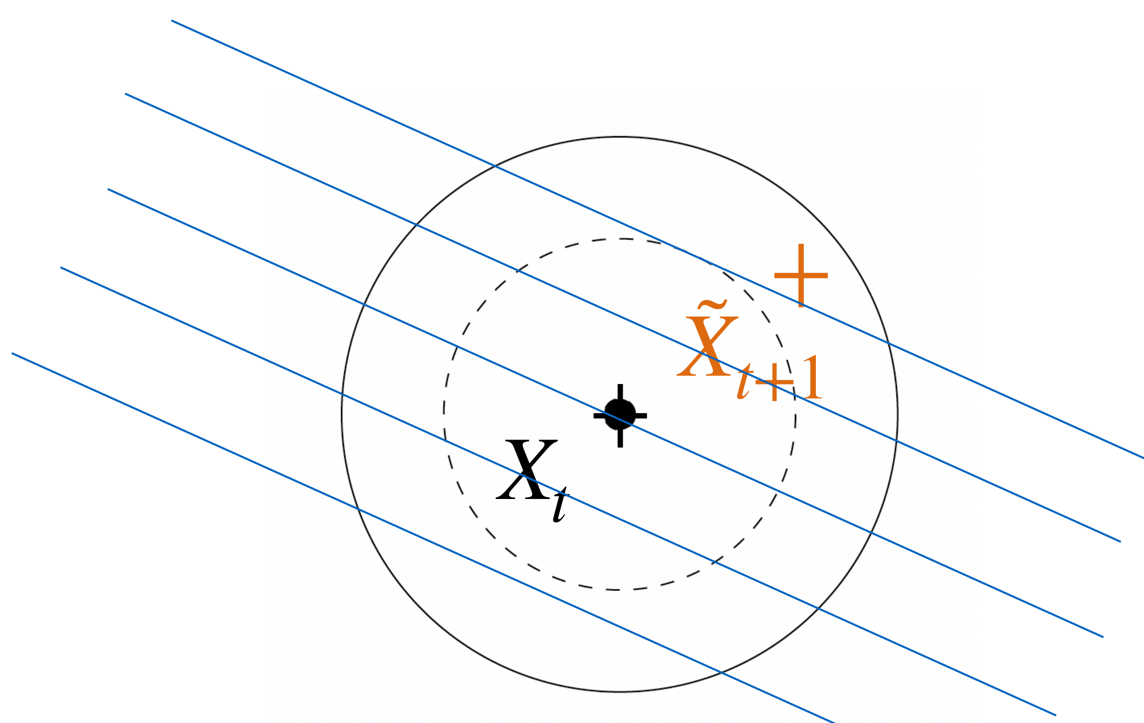
$\tilde{X}_{t+1}$: offspring created via **mutation** of parent

$$\tilde{X}_{t+1} = X_t + \sigma \mathcal{N}(0, I_d)$$

IF offspring better than parent [$f(\tilde{X}_{t+1}) \leq f(X_t)$]

$X_{t+1} = \tilde{X}_{t+1}$ #keep offspring

minimize $f : \mathbb{R}^2 \to \mathbb{R}$



A simple Evolution Strategy: (1+1)-ES

$t \in \mathbb{N}$ iteration index

$X_t \in \mathbb{R}^n$: parent at iteration t

$\sigma \in \mathbb{R}_>$: step-size

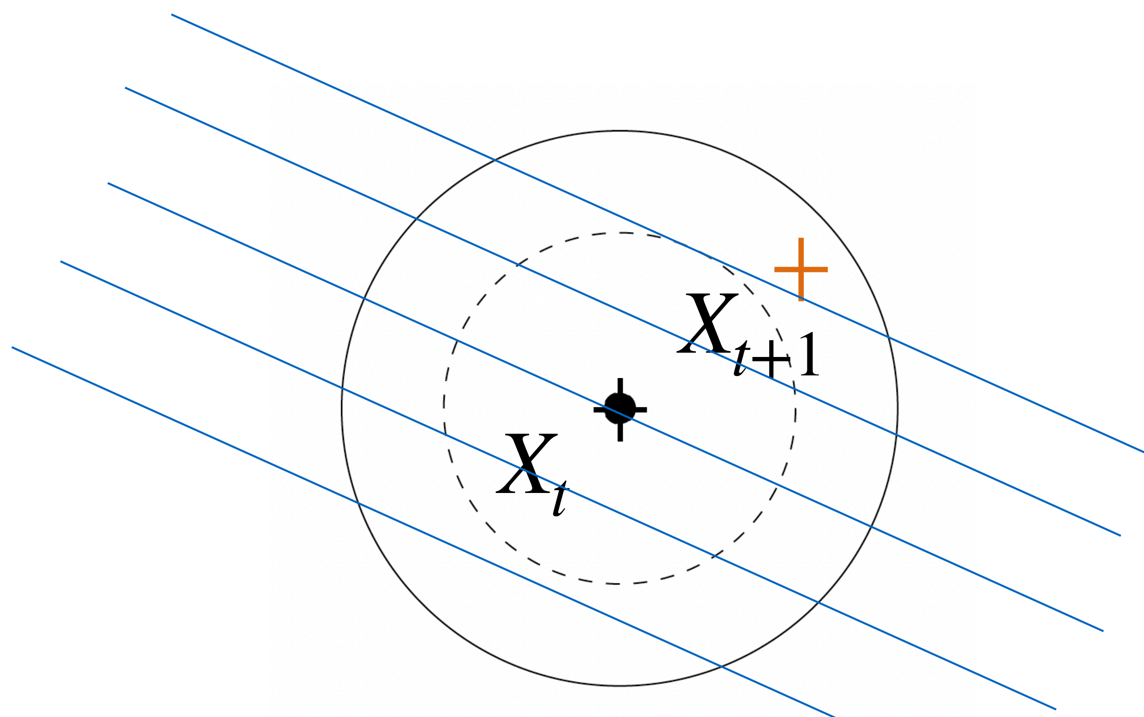
$\tilde{X}_{t+1}$: offspring created via **mutation** of parent

$$\tilde{X}_{t+1} = X_t + \sigma \mathcal{N}(0, I_d)$$

IF offspring better than parent [$f(\tilde{X}_{t+1}) \leq f(X_t)$]

minimize $f : \mathbb{R}^2 \rightarrow \mathbb{R}$

$X_{t+1} = \tilde{X}_{t+1}$ #keep offspring



A simple Evolution Strategy: (1+1)-ES

$t \in \mathbb{N}$ iteration index

$X_t \in \mathbb{R}^n$: parent at iteration t

$\sigma \in \mathbb{R}_>$: step-size

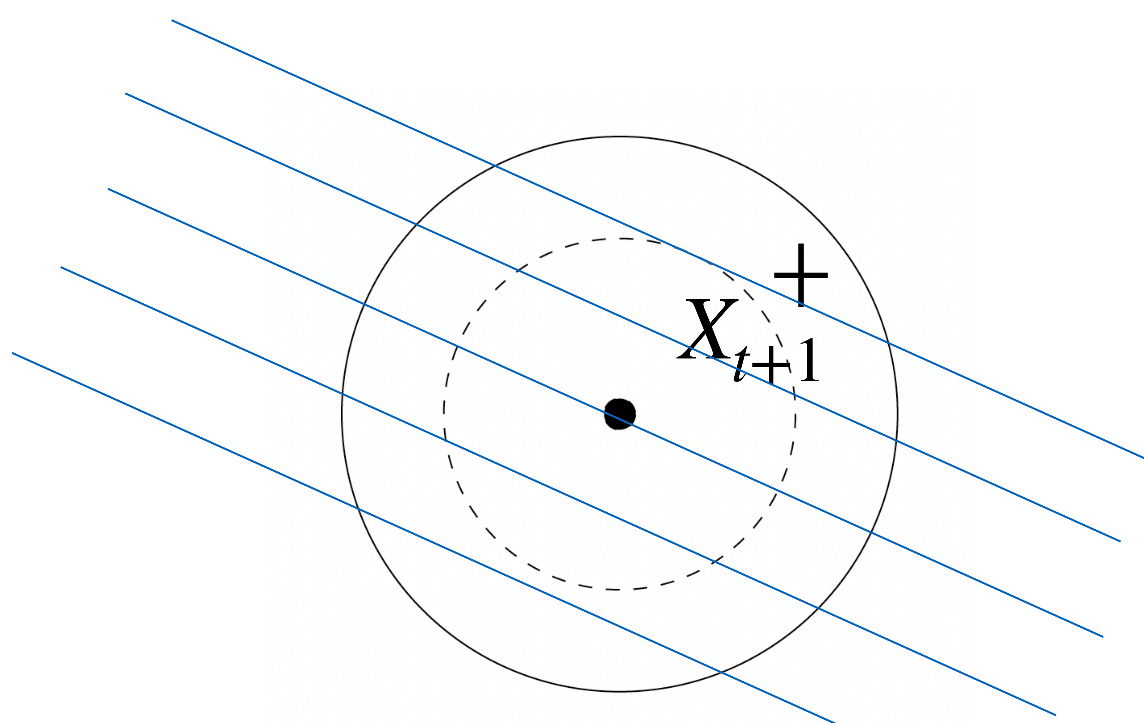
$\tilde{X}_{t+1}$: offspring created via **mutation** of parent

$$\tilde{X}_{t+1} = X_t + \sigma \mathcal{N}(0, I_d)$$

IF offspring better than parent [$f(\tilde{X}_{t+1}) \leq f(X_t)$]

$X_{t+1} = \tilde{X}_{t+1}$ #keep offspring

minimize $f : \mathbb{R}^2 \to \mathbb{R}$



A simple Evolution Strategy: (1+1)-ES

$t \in \mathbb{N}$ iteration index

$X_t \in \mathbb{R}^n$: parent at iteration t

$\sigma \in \mathbb{R}_>$: step-size

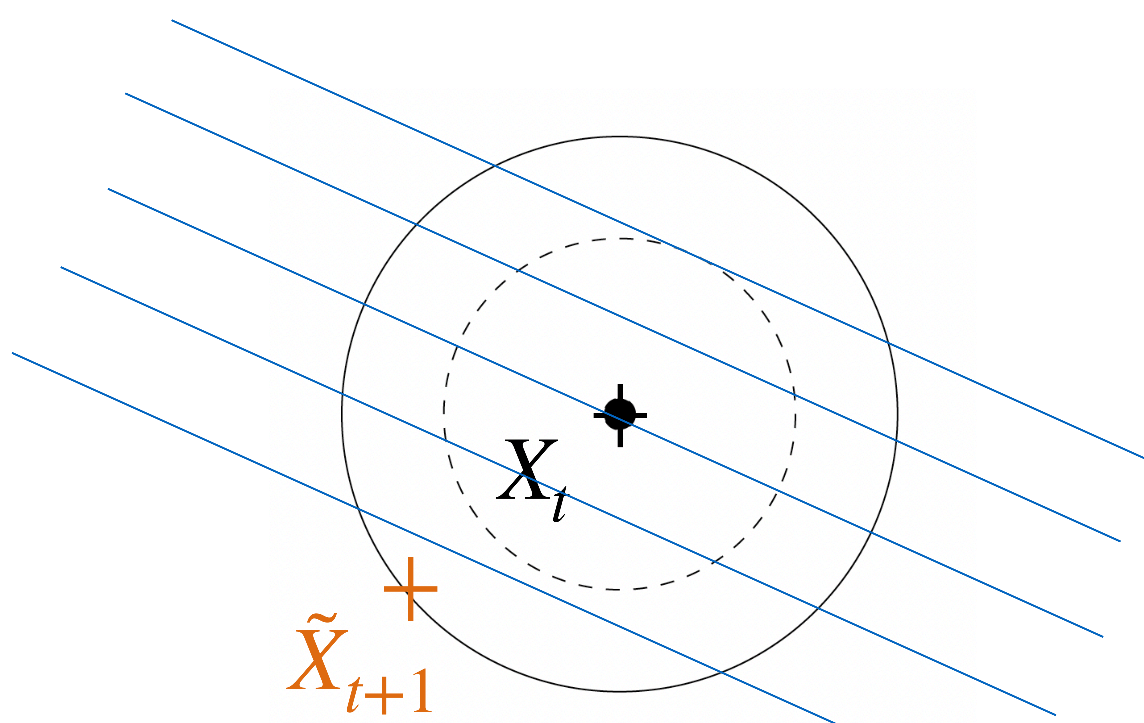
$\tilde{X}_{t+1}$: offspring created via **mutation** of parent

$$\tilde{X}_{t+1} = X_t + \sigma \mathcal{N}(0, I_d)$$

IF offspring better than parent [$f(\tilde{X}_{t+1}) \leq f(X_t)$]

minimize $f : \mathbb{R}^2 \to \mathbb{R}$

$X_{t+1} = \tilde{X}_{t+1}$ #keep offspring



A simple Evolution Strategy: (1+1)-ES

$t \in \mathbb{N}$ iteration index

$X_t \in \mathbb{R}^n$: parent at iteration t

$\sigma \in \mathbb{R}_>$: step-size

$\tilde{X}_{t+1}$: offspring created via **mutation** of parent

$$\tilde{X}_{t+1} = X_t + \sigma \mathcal{N}(0, I_d)$$

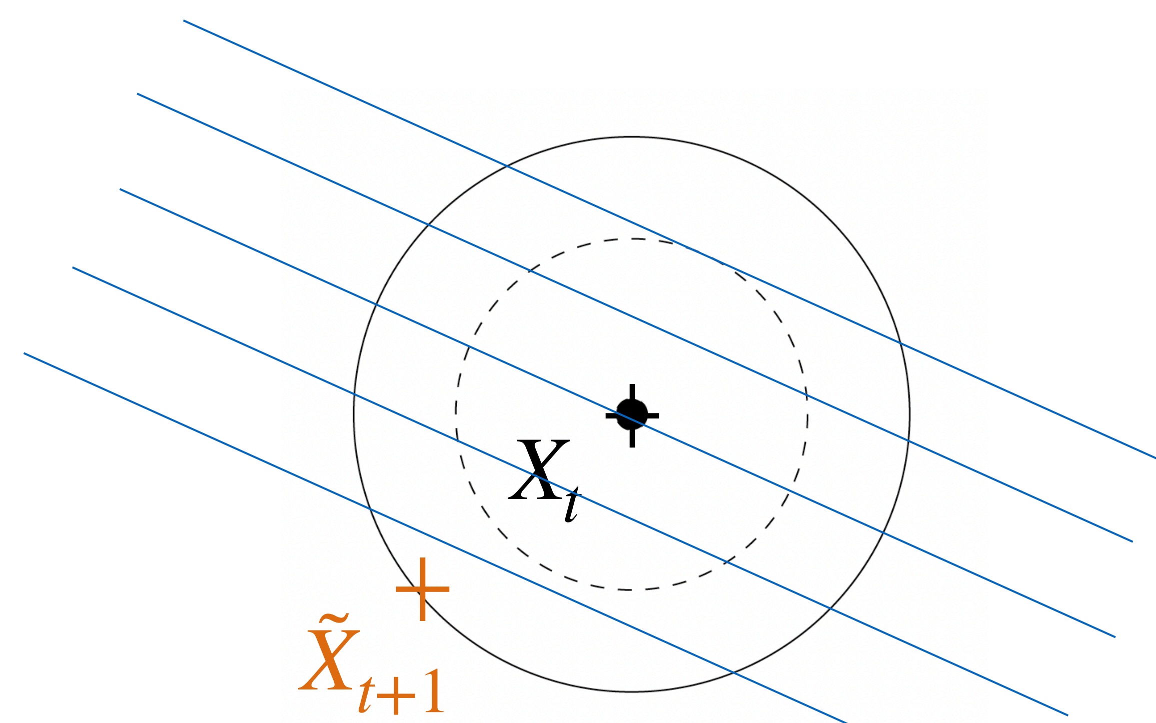
IF offspring better than parent [$f(\tilde{X}_{t+1}) \leq f(X_t)$]

$X_{t+1} = \tilde{X}_{t+1}$ #keep offspring

ELSE

$X_{t+1} = X_t$ #keep current parent

minimize $f : \mathbb{R}^2 \to \mathbb{R}$



A simple Evolution Strategy: (1+1)-ES

$t \in \mathbb{N}$ iteration index

$X_t \in \mathbb{R}^n$: parent at iteration t

$\sigma \in \mathbb{R}_>$: step-size

$\tilde{X}_{t+1}$: offspring created via **mutation** of parent

$$\tilde{X}_{t+1} = X_t + \sigma \mathcal{N}(0, I_d)$$

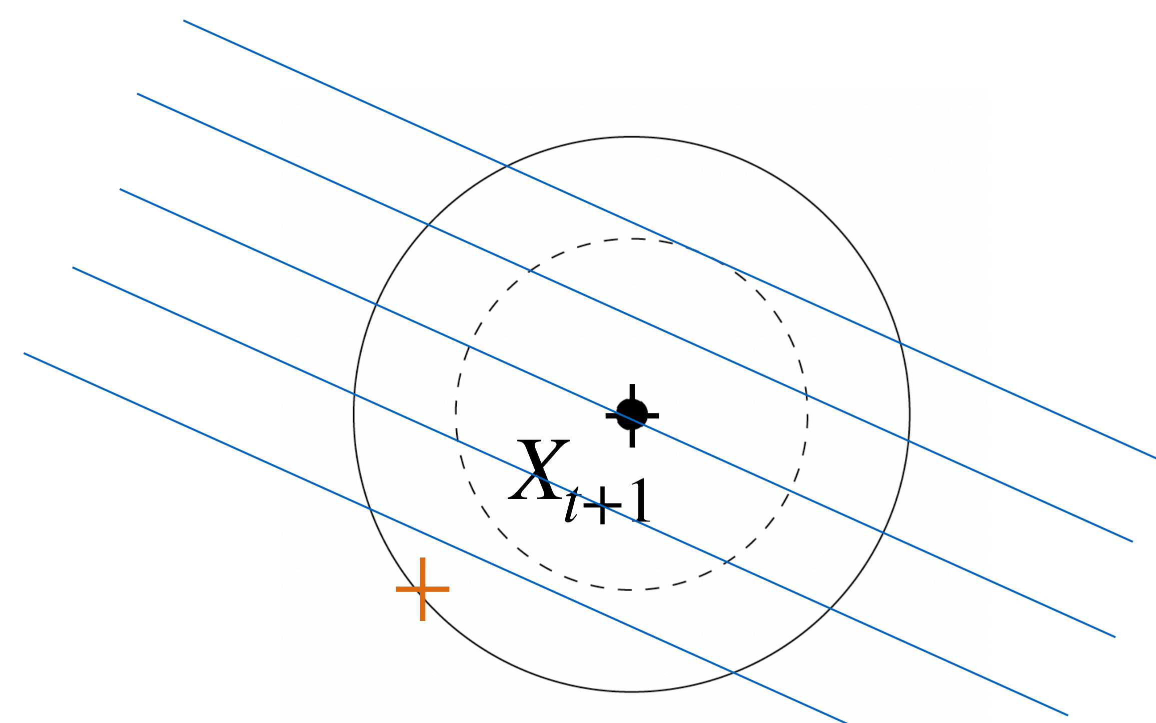
IF offspring better than parent [$f(\tilde{X}_{t+1}) \leq f(X_t)$]

$X_{t+1} = \tilde{X}_{t+1}$ #keep offspring

ELSE

$X_{t+1} = X_t$ #keep current parent

minimize $f : \mathbb{R}^2 \to \mathbb{R}$



(1+1)-ES algorithm - minimize $f : \mathbb{R}^n \to \mathbb{R}$

WHILE not happy

$$\tilde{X}_{t+1} = X_t + \sigma \mathcal{N}(0, I_d)$$

IF $f(\tilde{X}_{t+1}) \leq f(X_t)$ THEN

$X_{t+1} = \tilde{X}_{t+1}$ #keep offspring

ELSE

$X_{t+1} = X_t$ #keep current parent

ENDIF

$$t \leftarrow t + 1$$

(1+1)-ES algorithm - minimize $f : \mathbb{R}^n \to \mathbb{R}$

WHILE not happy

$$\tilde{X}_{t+1} = X_t + \sigma \mathcal{N}(0, I_d)$$

IF $f(\tilde{X}_{t+1}) \leq f(X_t)$ THEN

$X_{t+1} = \tilde{X}_{t+1}$ #keep offspring

ELSE

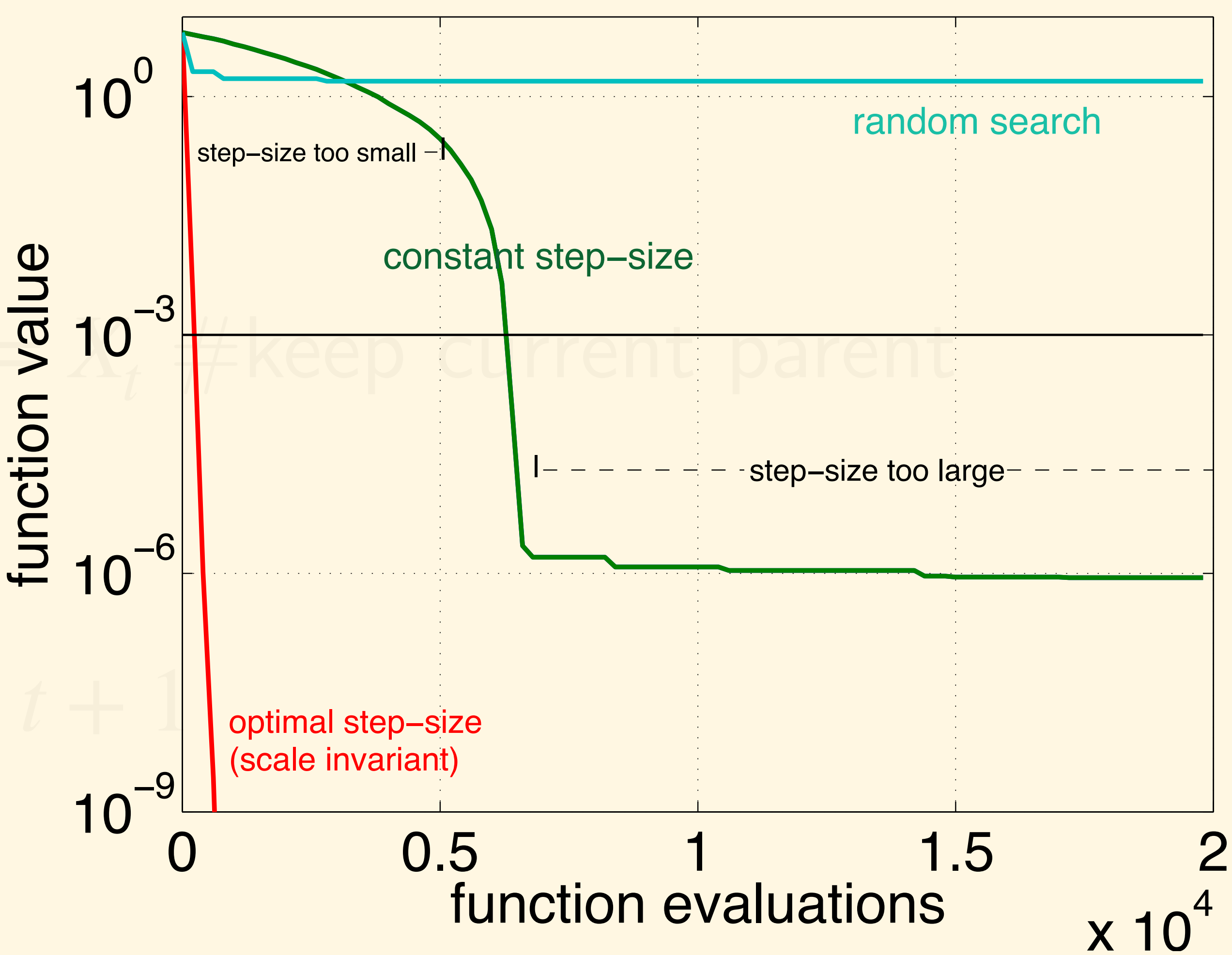
$X_{t+1} = X_t$ #keep current parent

ENDIF

$t \leftarrow t+1$

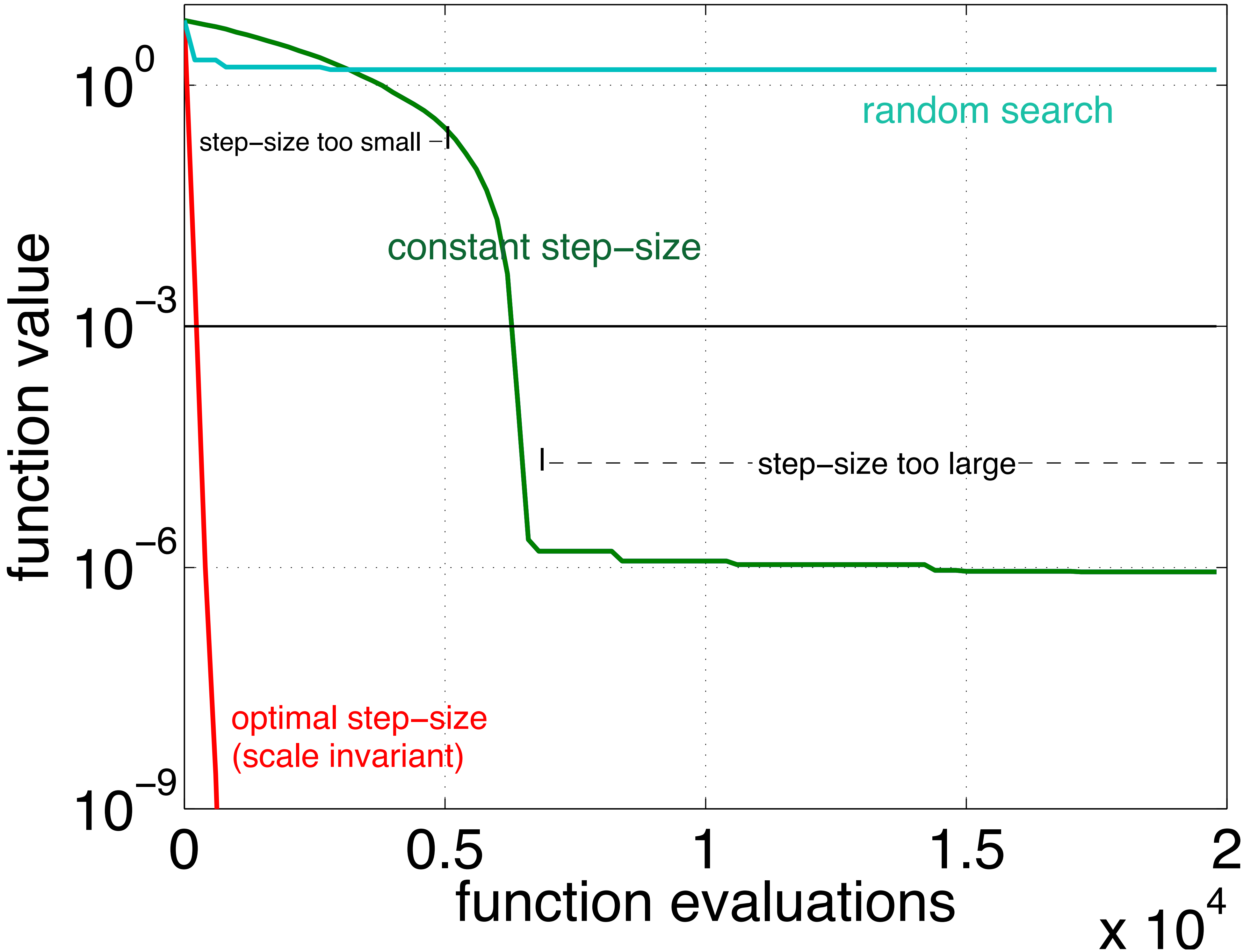
Issue #1:

- cannot converge "fast" if σ fixed



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

n=10



Need to adapt the step-size → step-size adaptive ES

(1+1)-ES algorithm with 1/5-th success rule [Rechenberg]

WHILE not happy

$\tilde{X}_{t+1} = X_t + \sigma_t \mathcal{N}(0, I_d)$

IF $f(\tilde{X}_{t+1}) \leq f(X_t)$ THEN

$X_{t+1} = \tilde{X}_{t+1}$ #keep offspring

$\sigma_{t+1} = 1.5 \times \sigma_t$ #increase step-size

ELSE

$X_{t+1} = X_t$ #keep current parent

$\sigma_{t+1} = 1.5^{-1/4} \times \sigma_t$ #decrease step-size

ENDIF

$t \leftarrow t + 1$

(1+1)-ES algorithm with 1/5-th success rule [Rechenberg]

WHILE not happy

$$\tilde{X}_{t+1} = X_t + \sigma_t \mathcal{N}(0, I_d)$$

IF $f(\tilde{X}_{t+1}) \leq f(X_t)$ THEN

$$X_{t+1} = \tilde{X}_{t+1}$$

$$\sigma_{t+1} = 1.5 \times \sigma_t$$

ELSE

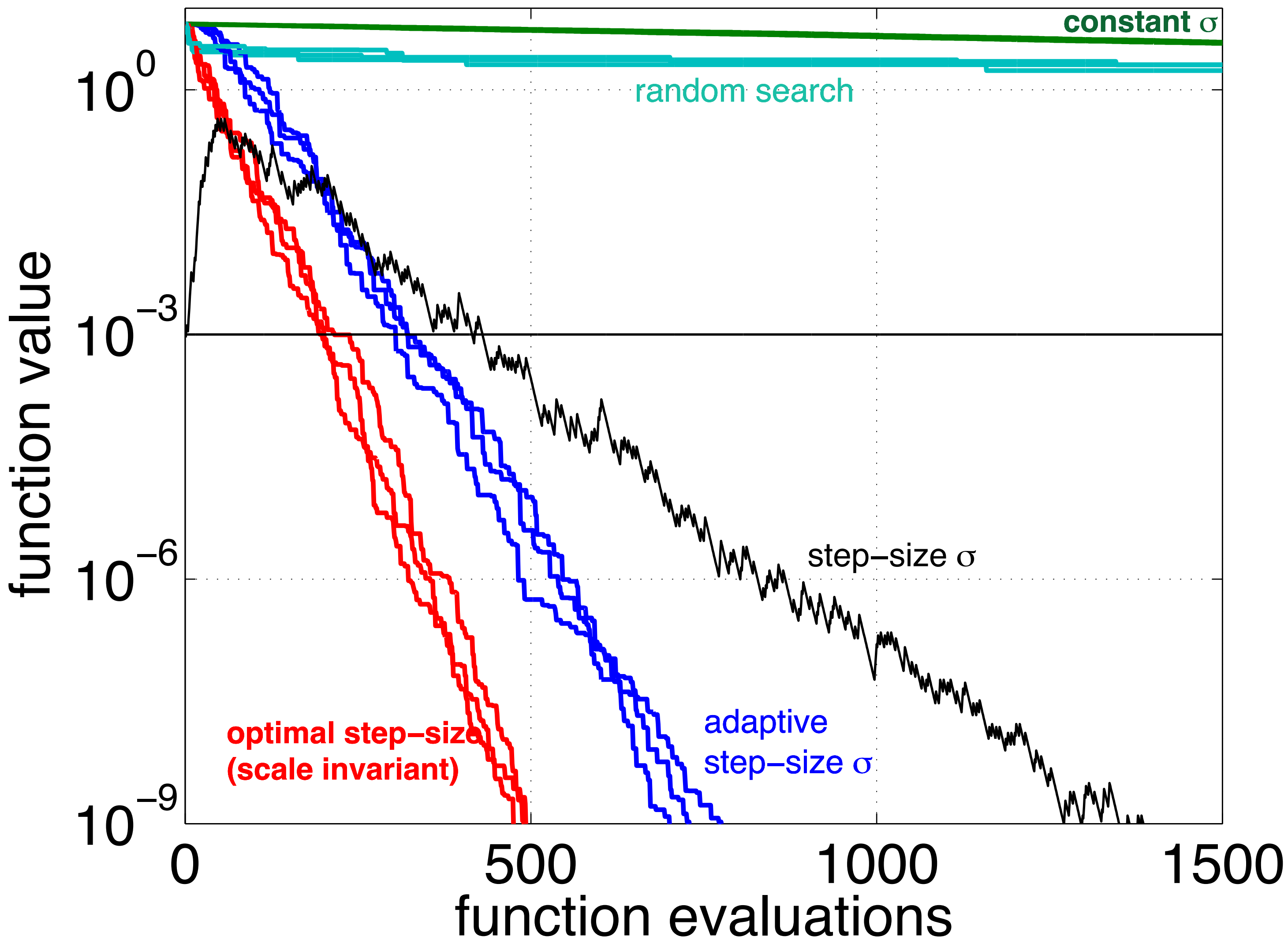
$$X_{t+1} = X_t$$

$$\sigma_{t+1} = 1.5^{-1/4} \times \sigma_t$$

ENDIF

$$t \leftarrow t + 1$$

Achieves **geometric** ("linear") convergence



$$f(x) = \sum_{i=1}^{n} x_i^2 \ , \ \text{n=10}$$

Invented (independently) in other communities

Schumer, Steiglitz, 1968 “Adaptive Step Size Random Search”

270 IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC-13, NO. 3, JUNE 1968

Adaptive Step Size Random Search

MICHAEL A. SCHUMER, MEMBER, IEEE, AND KENNETH STEIGLITZ, MEMBER, IEEE

Abstract—Fixed step size random search for minimization of functions of several parameters is described and compared with the fixed step size gradient method for a particular surface. A theoretical technique, using the optimum step size at each step, is analyzed. A practical adaptive step size random search algorithm is then proposed, and experimental experience is reported that shows the superiority of random search over other methods for sufficiently high dimension.

INTRODUCTION

Rastrigin has compared a fixed step size random search (FSSRS) method with a fixed step size gradient method and concluded that under certain circumstances FSSRS is superior. It is clear, however, that if the step size of the random search method were optimum at each step, even better performance would result. In this paper, a hypothetical random search method that uses the optimum step size at each point will be analyzed for a hyperspherical surface. An adaptive step size

Devroye 1972 “The compound random search algorithm”

THE COMPOUND RANDOM SEARCH ALGORITHM

Luc P.DEVROYE
Dept.of Electrical Engineering
Catholic University of Louvain,Belgium

ABSTRACT

The optimization of a function of many variables is investigated.A new algorithm is proposed : the compound random search algorithm (CRSA).This algorithm combines the features of random search and non-random direct search.Each part of the algorithm is discussed in detail.The CRSA is compared with several other direct search methods in many different problems.The results are promising.Some modifications of the basic search scheme make the CRSA particularly useful and efficient.

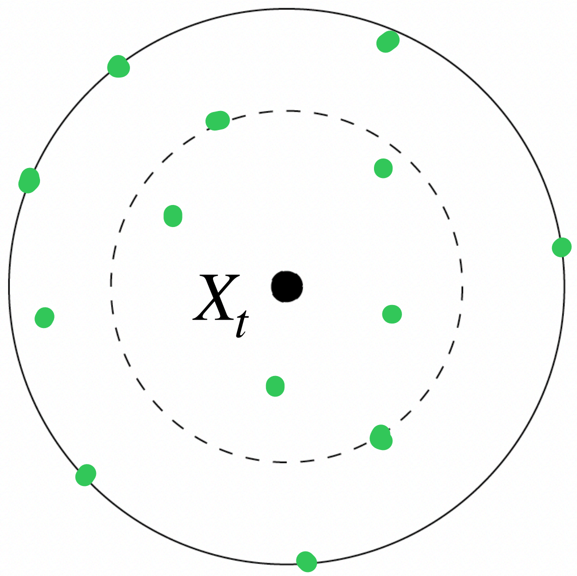
accelerate and control the search.Moreover, new information must be gathered during the search and transformed into useful data. There is a remarkable resemblance between this search and a learning process or a game against nature.

A new algorithm is developed which performs much better,even under the worst circumstances,than the powerful methods of Gucker [13] or Rosenbrock [11].Moreover,this method is one of the fastest random optimization methods when an optimum must be localized quite accurately.In our discussion, the rate of convergence is determined in

More issues with (1+1)-ES with 1/5th success rule:

candidate solutions are isotropically distributed

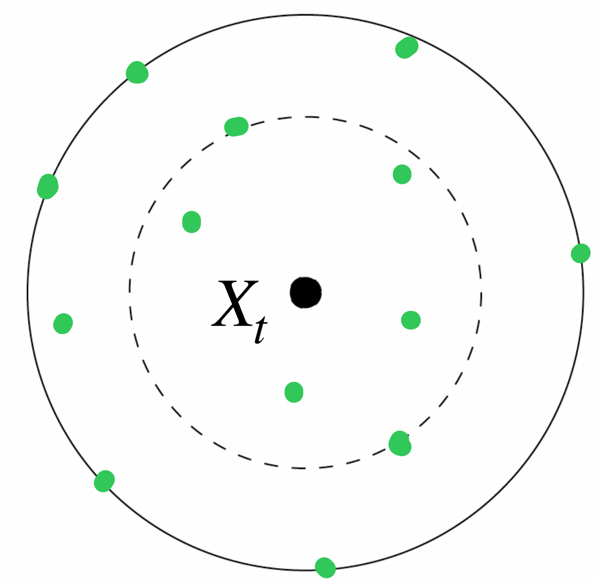
$$\tilde{X}_{t+1} = X_t + \sigma_t \mathcal{N}(0, I_d)$$



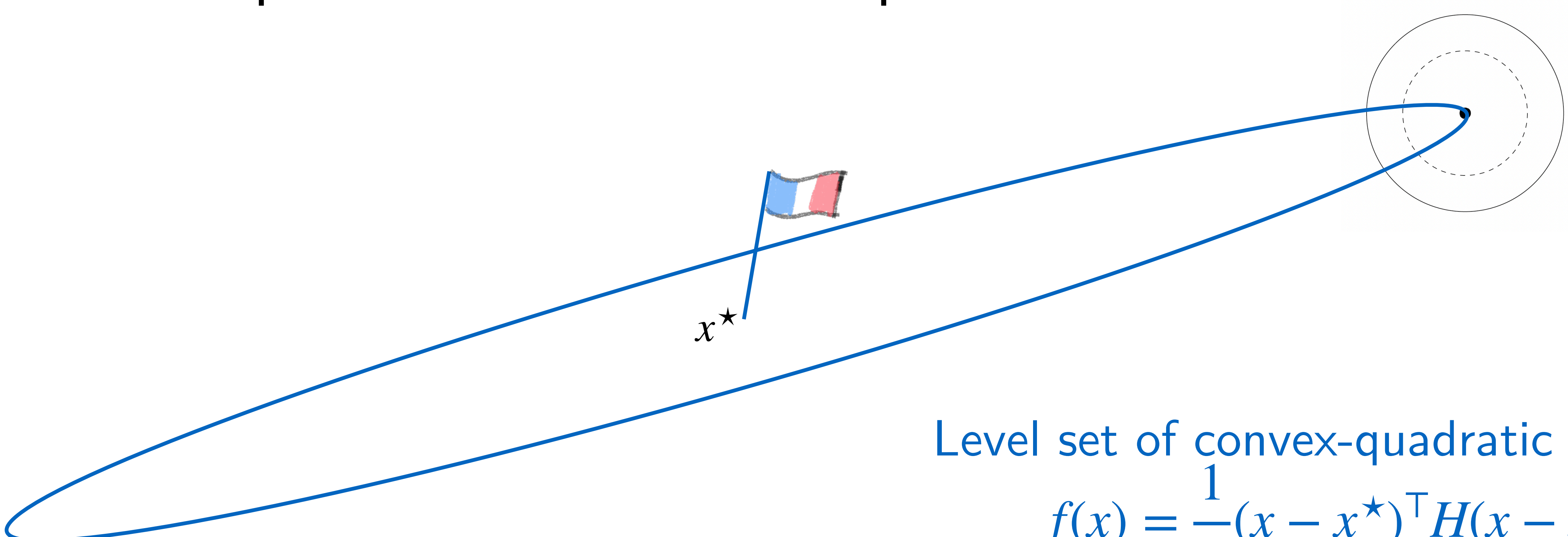
More issues with (1+1)-ES with 1/5th success rule:

candidate solutions are isotropically distributed

$$\tilde{X}_{t+1} = X_t + \sigma_t \mathcal{N}(0, I_d)$$

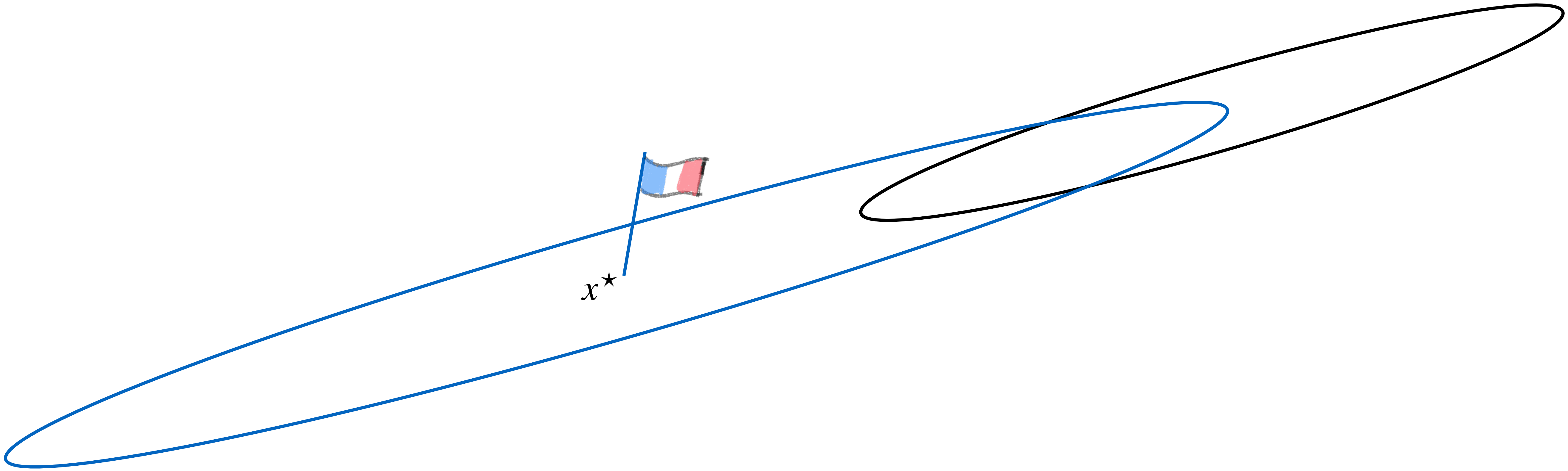


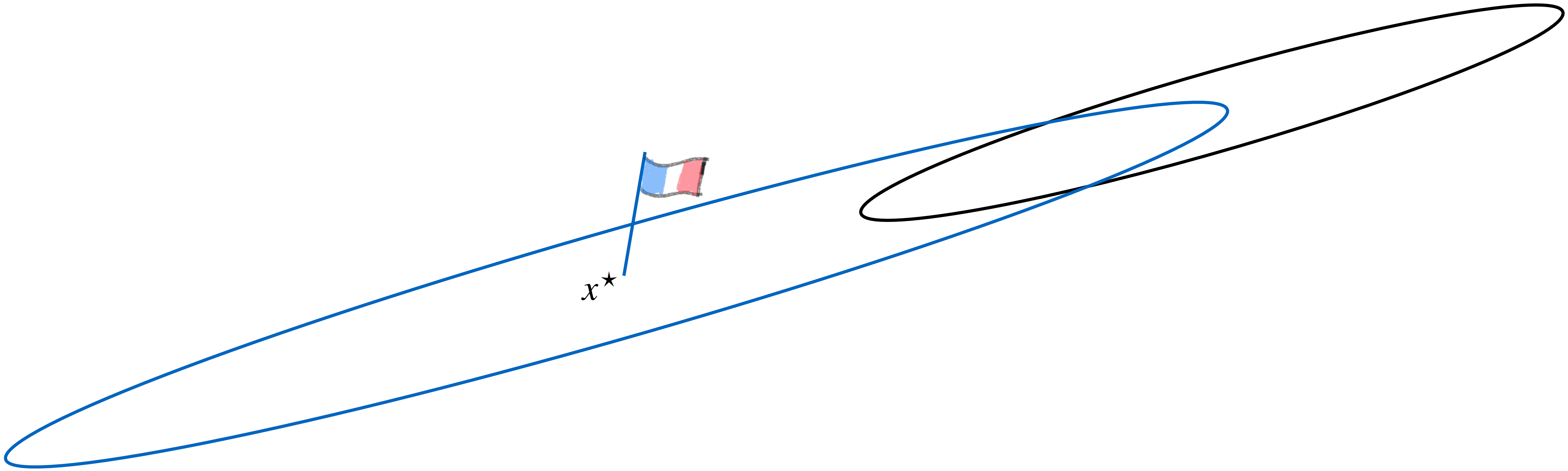
not adapted on ill-conditioned problems:



Level set of convex-quadratic function:

$$f(x) = \frac{1}{2}(x - x^\star)^\top H (x - x^\star)$$


$x^\star$



Ideally on a ill-conditionned quadratic problem: $\frac{1}{2}(x - x^\star)^\top H(x - x^\star)$
with cond(H) $>>$1 (order of 10^6) sample with:

$$X_t + \sigma_t \mathcal{N}(0, C_t) \text{ with } C_t \propto H^{-1}$$

From (1+1)-ES to $(\mu/\mu_w, \lambda)$-CMA-ES

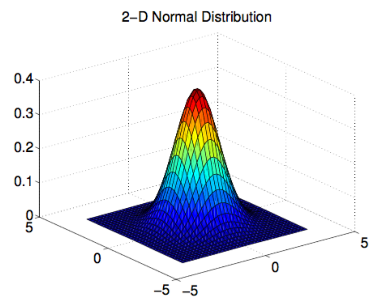
❶ adapt the covariance matrix of sampling distribution:
step-size + covariance matrix adaptive ES = CMA-ES

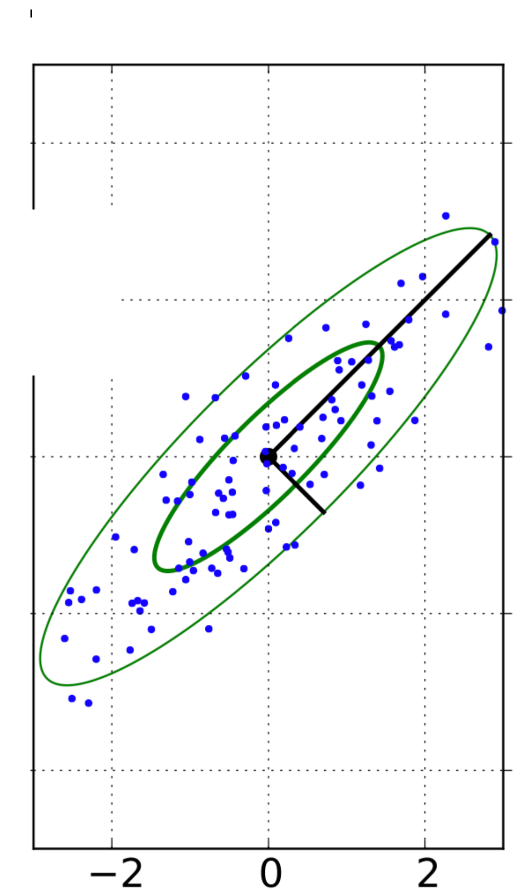
❷ use a population: $(\mu/\mu_w, \lambda)$-CMA-ES

❸ use mutation and (meta)-crossover

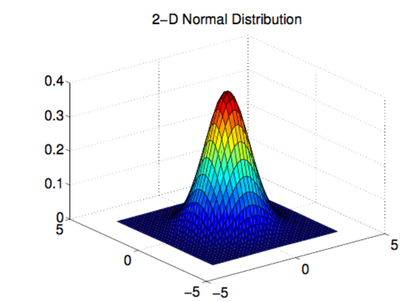
Adaptive Stochastic Optimization Algorithm

Given e.g. $\theta_t = (m_t, \sigma_t, C_t) \in \mathbb{R}^n \times \mathbb{R}_> \times \mathcal{S}^n_{++}$





Adaptive Stochastic Optimization Algorithm

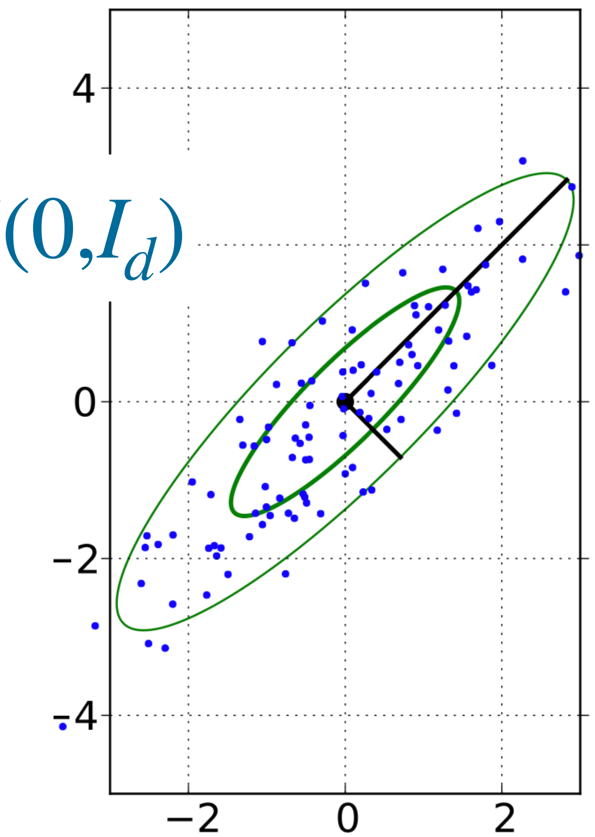


Given e.g. $\theta_t = (m_t, \sigma_t, C_t) \in \mathbb{R}^n \times \mathbb{R}_> \times \mathcal{S}^n_{++}$

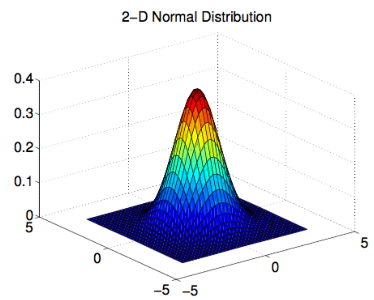
❶ Sample candidate solutions $X^i_{t+1} \sim \mathcal{N}(m_t, \sigma_t^2 C_t)$, i.e.

$$X^i_{t+1} = m_t + \sigma_t \sqrt{C_t}\, U^i_{t+1},\ i = 1, \ldots, \lambda$$

$\{U_t, t \geq 1\}$ i.i.d., $U^i_{t+1} \sim \mathcal{N}(0, I_d)$



Adaptive Stochastic Optimization Algorithm

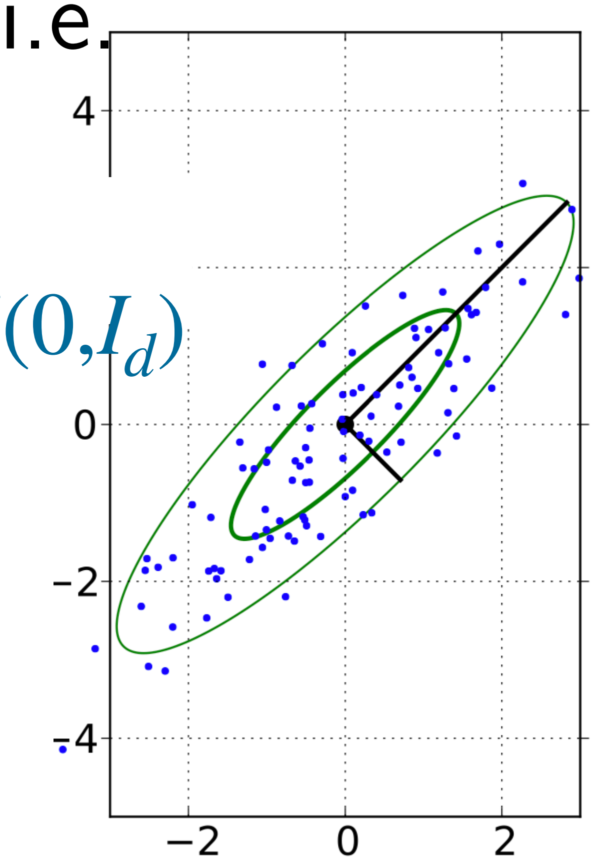


Given e.g. $\theta_t = (m_t, \sigma_t, C_t) \in \mathbb{R}^n \times \mathbb{R}_> \times \mathcal{S}^n_{++}$

❶ Sample candidate solutions $X^i_{t+1} \sim \mathcal{N}(m_t, \sigma_t^2 C_t)$, i.e.

$$X^i_{t+1} = m_t + \sigma_t \sqrt{C_t} U^i_{t+1}, \ i = 1, \ldots, \lambda$$

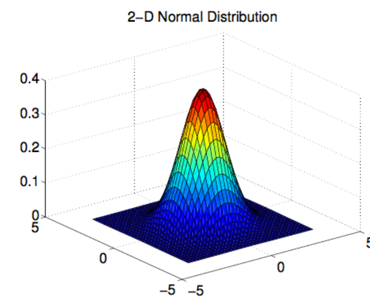
$\{U_t, t \geq 1\}$ i.i.d., $U^i_{t+1} \sim \mathcal{N}(0, I_d)$



❷ Evaluate and rank candidate solutions

$$f\left(X_{t+1}^{s_{t+1}(1)}\right) \leq \ldots \leq f\left(X_{t+1}^{s_{t+1}(\lambda)}\right)$$

Adaptive Stochastic Optimization Algorithm

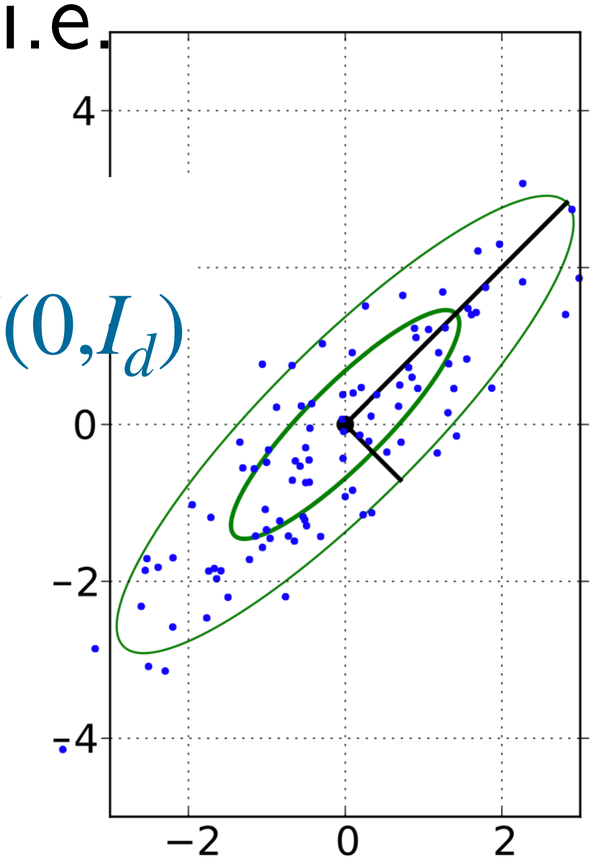


Given e.g. $\theta_t = (m_t, \sigma_t, C_t) \in \mathbb{R}^n \times \mathbb{R}_> \times \mathcal{S}^n_{++}$

❶ Sample candidate solutions $X^i_{t+1} \sim \mathcal{N}(m_t, \sigma_t^2 C_t)$, i.e.

$$X^i_{t+1} = m_t + \sigma_t \sqrt{C_t} U^i_{t+1},\ i = 1, \ldots, \lambda$$

$\{U_t, t \geq 1\}$ i.i.d., $U^i_{t+1} \sim \mathcal{N}(0, I_d)$



❷ Evaluate and rank candidate solutions

$$f\left(X^{s_{t+1}(1)}_{t+1}\right) \leq \ldots \leq f\left(X^{s_{t+1}(\lambda)}_{t+1}\right)$$

❸ Update θ_t:

$$\theta_{t+1} = G\left(\theta_t, [U^{s_{t+1}(1)}_{t+1}, \ldots, U^{s_{t+1}(\lambda)}_{t+1}]\right)$$

should drive m_t towards the optimum

Comparison-based $\Rightarrow$ Invariance strict. increasing functions

Let $g : \mathrm{Im} f \to \mathrm{Im}(g)$ be strictly increasing

$$f\left(X_{t+1}^{s(1)}\right) \leq \ldots \leq f\left(X_{t+1}^{s(\lambda)}\right)$$

$$\Leftrightarrow$$

$$g \circ f\left(X_{t+1}^{s(1)}\right) \leq \ldots \leq g \circ f\left(X_{t+1}^{s(\lambda)}\right)$$

Consequence: same sequence$\{\theta_t, t \geq 0\}$ on f or $g \circ f$

Comparison-based $\Rightarrow$ Invariance strict. increasing functions

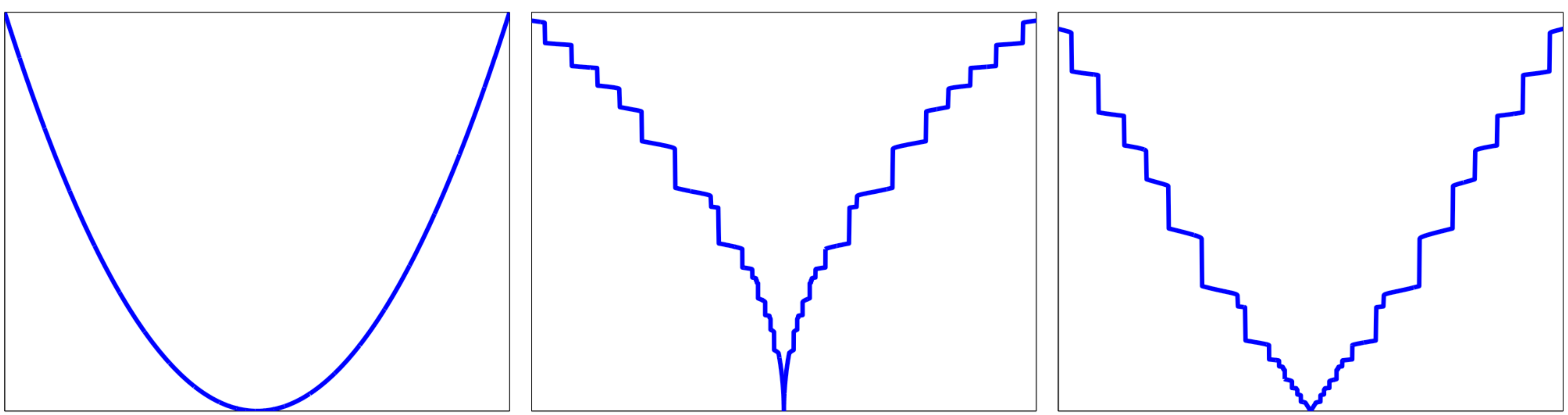
Let $g : \mathrm{Im} f \to \mathrm{Im}(g)$ be strictly increasing

$$f\left(X_{t+1}^{s(1)}\right) \leq \ldots \leq f\left(X_{t+1}^{s(\lambda)}\right)$$

$$\Leftrightarrow$$

$$g \circ f\left(X_{t+1}^{s(1)}\right) \leq \ldots \leq g \circ f\left(X_{t+1}^{s(\lambda)}\right)$$

Consequence: same sequence$\{\theta_t, t \geq 0\}$ on f or $g \circ f$



$f(x) = \|x\|^2$ $\qquad$ $g_1 \circ f$ $\qquad$ $g_2 \circ f$

CMA-ES - simplified setting $\theta_t = (m_t, \sigma_t, C_t) \in \mathbb{R}^n \times \mathbb{R}_> \times \mathcal{S}^n_{++}$

Sampling + ranking:

$X^i_{t+1} = m_t + \sigma_t \sqrt{C_t} U^i_{t+1} \quad i = 1, \dots, \lambda$

$\{U_t, t \geq 1\}$ i.i.d $U^i_{t+1} \sim \mathcal{N}(0, I_d)$

$f\left(X^{s_{t+1}(1)}_{t+1}\right) \leq \dots \leq f\left(X^{s_{t+1}(\lambda)}_{t+1}\right)$

CMA-ES - simplified setting $\theta_t = (m_t, \sigma_t, C_t) \in \mathbb{R}^n \times \mathbb{R}_> \times \mathcal{S}^n_{++}$

Sampling + ranking:

$$X^i_{t+1} = m_t + \sigma_t \sqrt{C_t} U^i_{t+1} \quad i = 1, \ldots, \lambda$$

$$f\left(X^{s_{t+1}(1)}_{t+1}\right) \leq \ldots \leq f\left(X^{s_{t+1}(\lambda)}_{t+1}\right)$$

$\{U_t, t \geq 1\}$ i.i.d $U^i_{t+1} \sim \mathcal{N}(0, I_d)$

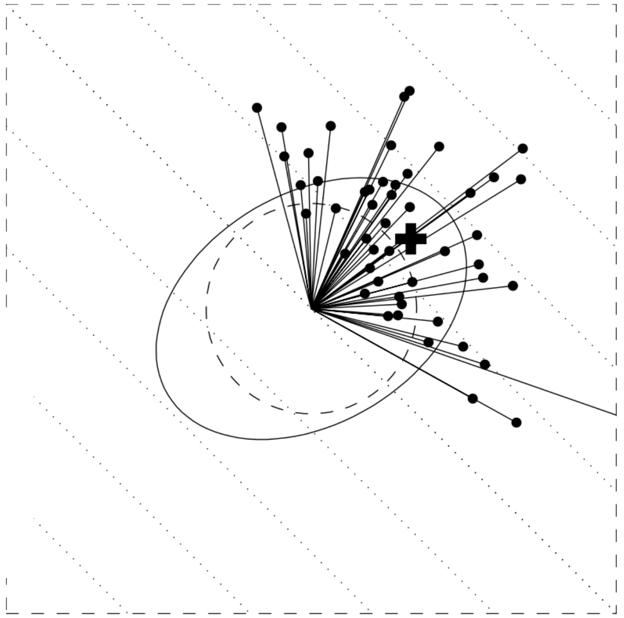
Update of θ_t:

$$m_{t+1} = \sum_{i=1}^{\mu} w_i X^{s_{t+1}(i)}_{t+1} = m_t + \sigma_t \sqrt{C_t} \sum_{i=1}^{\mu} w_i U^{s_{t+1}(i)}_{t+1}$$

$\sum_{i=1}^{\mu} w_i = 1, \mu_{\text{eff}} = 1/\sum w_i^2$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{c_\sigma}{d_\sigma}\left[\frac{\sqrt{\mu_{\text{eff}}} \| \sum_{i=1}^{\mu} w_i U^{s_{t+1}(i)}_{t+1} \|}{E[\|\mathcal{N}(0, I_d)\|]} - 1\right]\right)$$

$$C_{t+1} = (1 - c_\mu) C_t + \underbrace{c_\mu \sqrt{C_t} \left(\sum_{i=1}^{\mu} w_i U^{s_{t+1}(i)}_{t+1} [U^{s_{t+1}(i)}_{t+1}]^\top\right) \sqrt{C_t}}_{\text{rank } \mu \text{ update}}$$



Cumulative Step-size Adaptation (CSA)

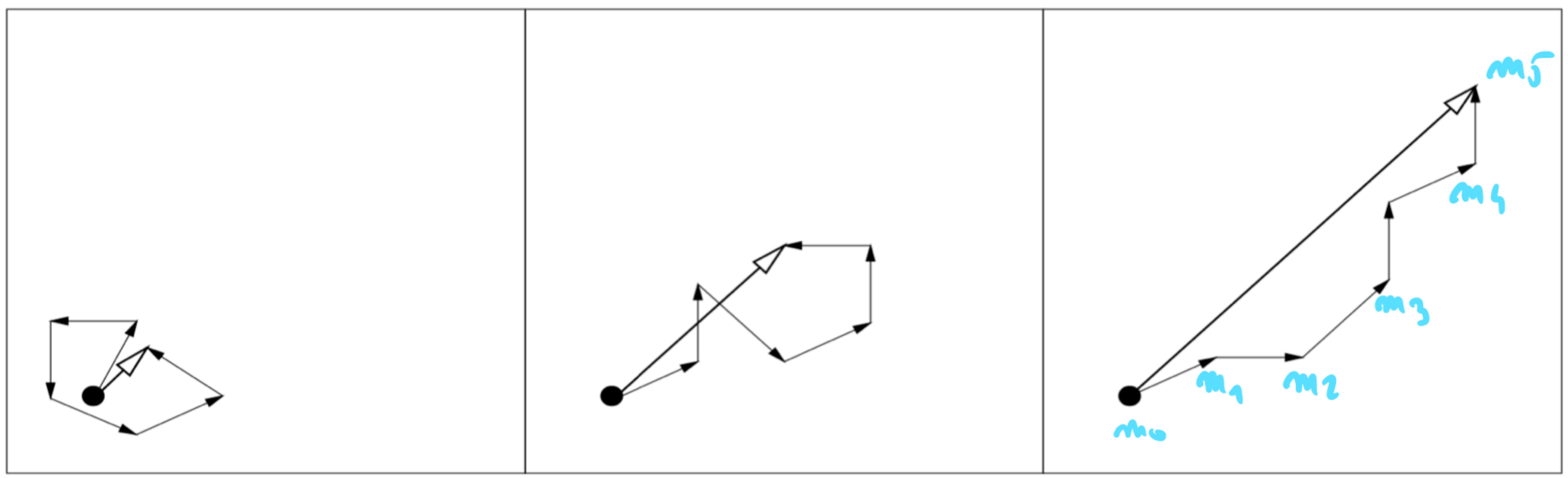
Sampling + ranking:

$$X_{t+1}^i = m_t + \sigma_t \sqrt{C_t} U_{t+1}^i \quad i = 1, \ldots, \lambda$$

$$f\left(X_{t+1}^{s_{t+1}(1)}\right) \leq \ldots \leq f\left(X_{t+1}^{s_{t+1}(\lambda)}\right)$$

$\{U_t, t \geq 1\}$ i.i.d $U_{t+1}^i \sim \mathcal{N}(0, I_d)$

Idea:



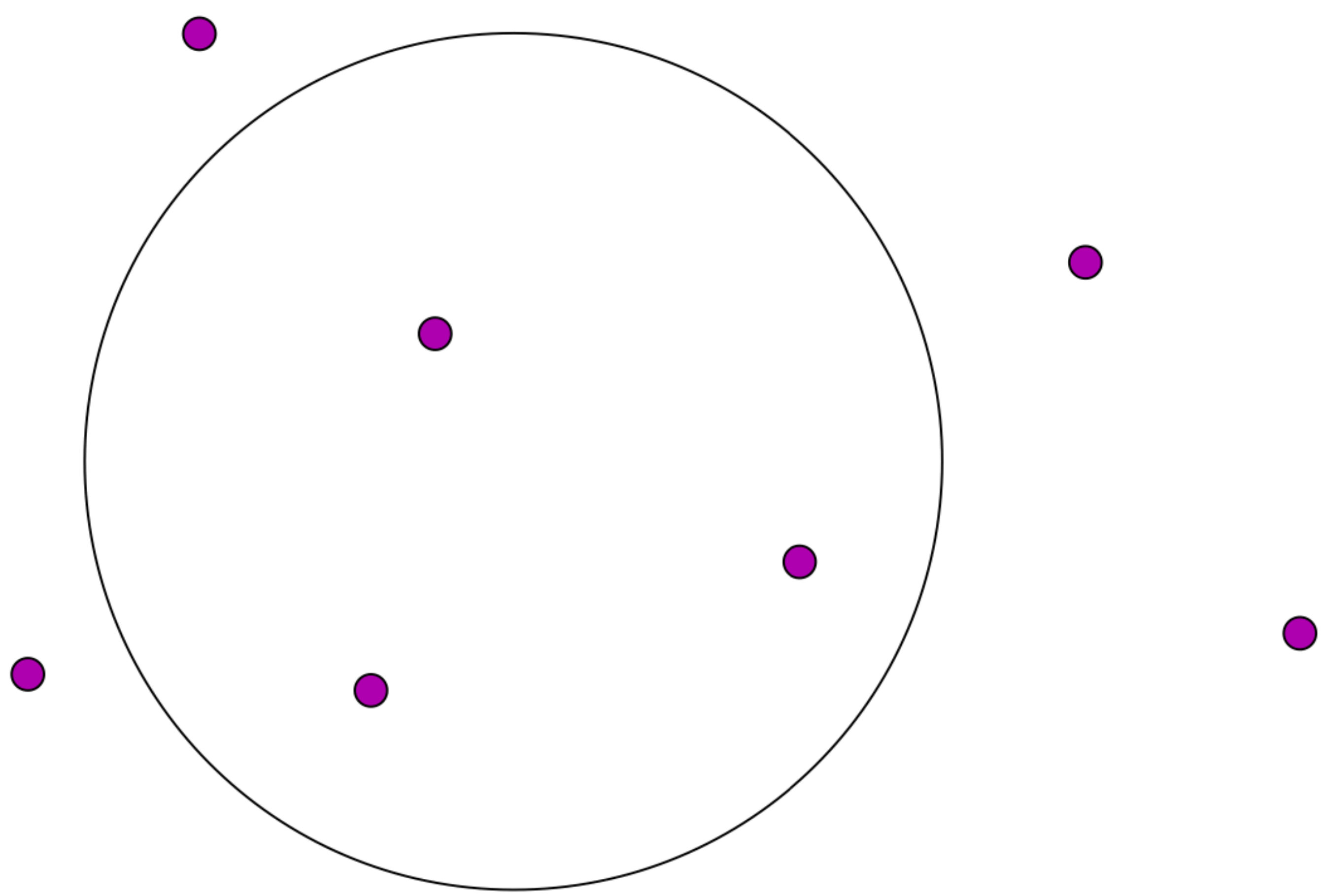
$\Downarrow$ decrease σ

$\Downarrow$ increase σ

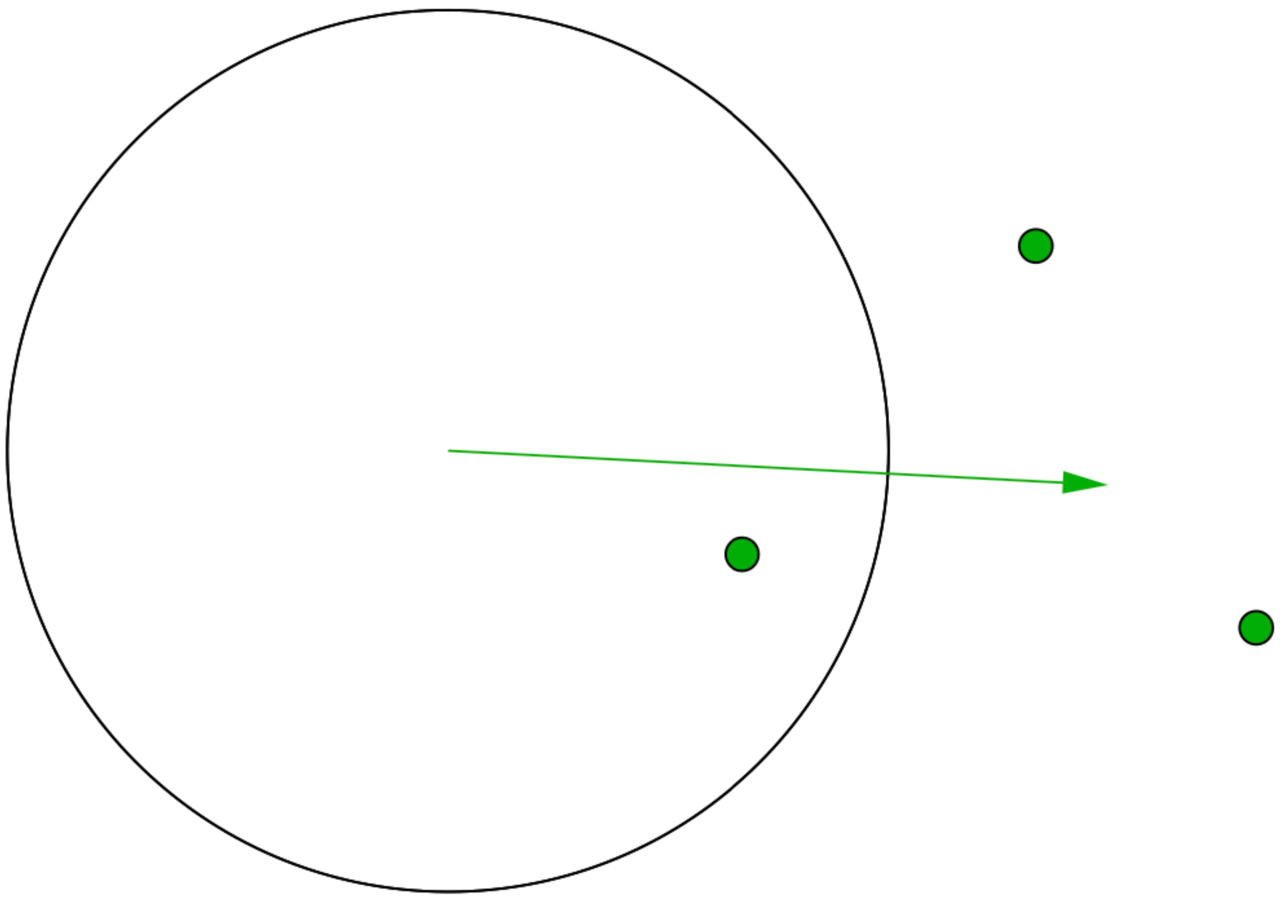
$$p_{t+1}^\sigma = (1 - c_\sigma) p_t^\sigma + \sqrt{c_\sigma (1 - c_\sigma) \mu_{\text{eff}}} \sum_{i=1}^{\mu} w_i U_{t+1}^{s_{t+1}(i)}$$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{c_\sigma}{d_\sigma} \left[\frac{\|p_{t+1}^\sigma\|}{E[\|\mathcal{N}(0, I_d)\|]} - 1 \right] \right)$$

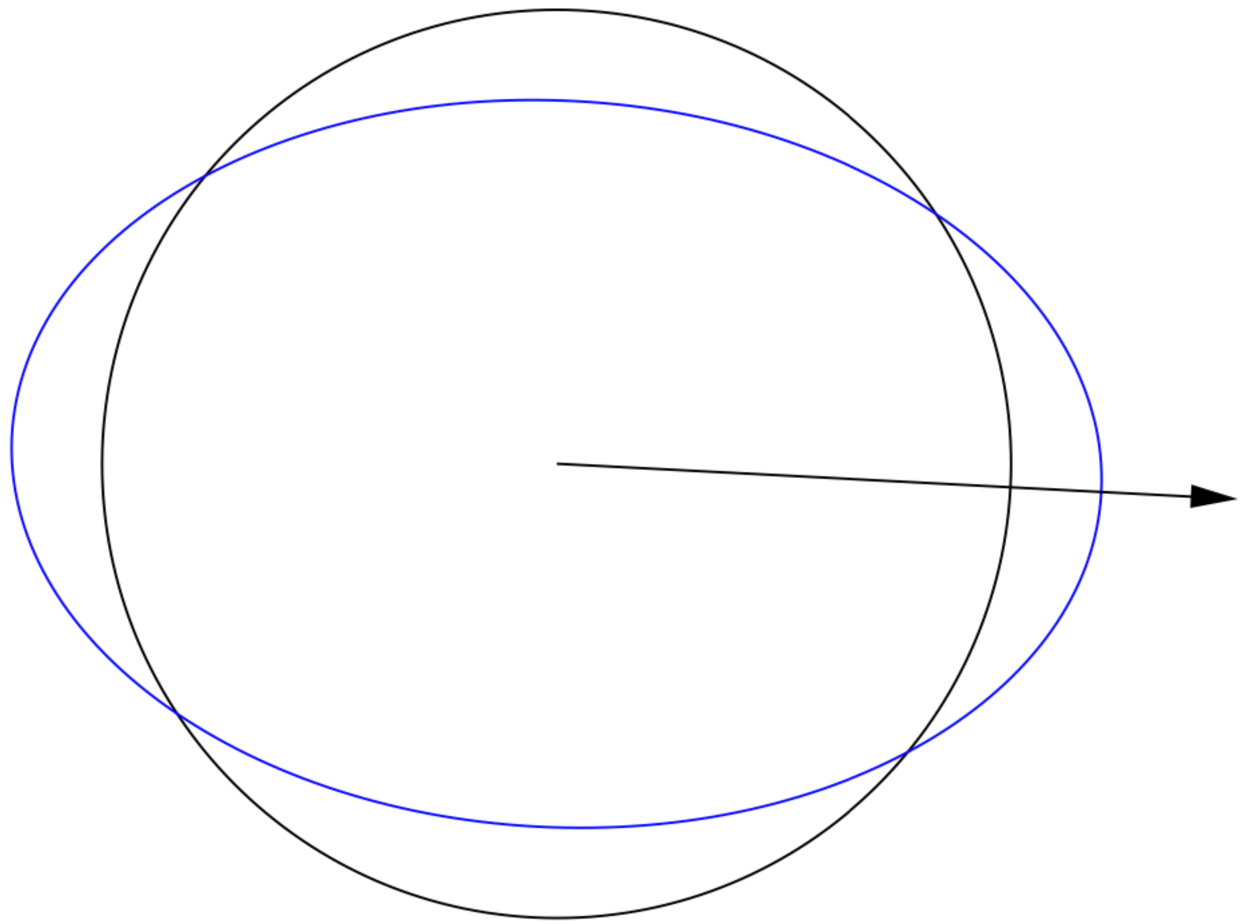
Rank-one update

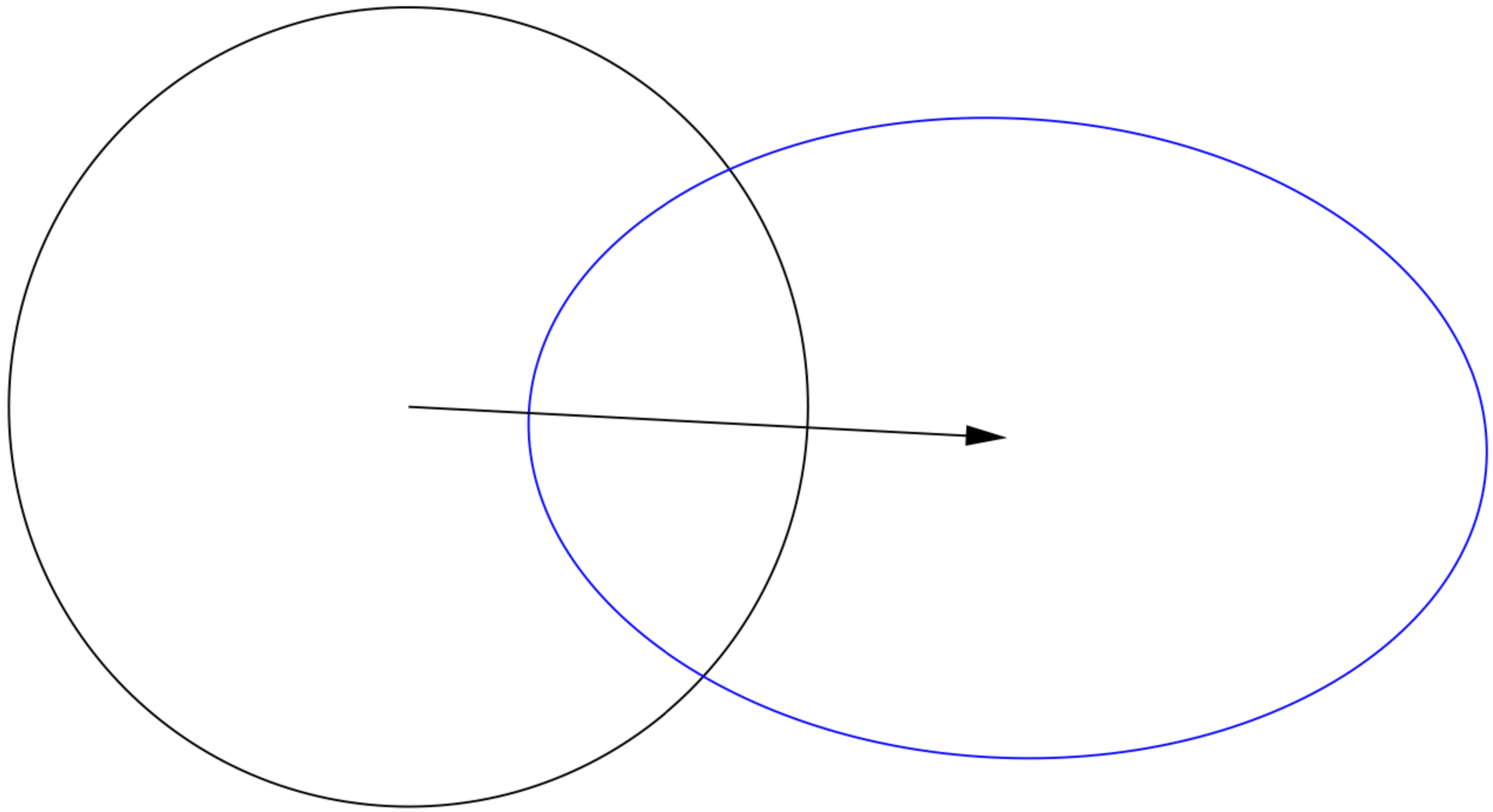


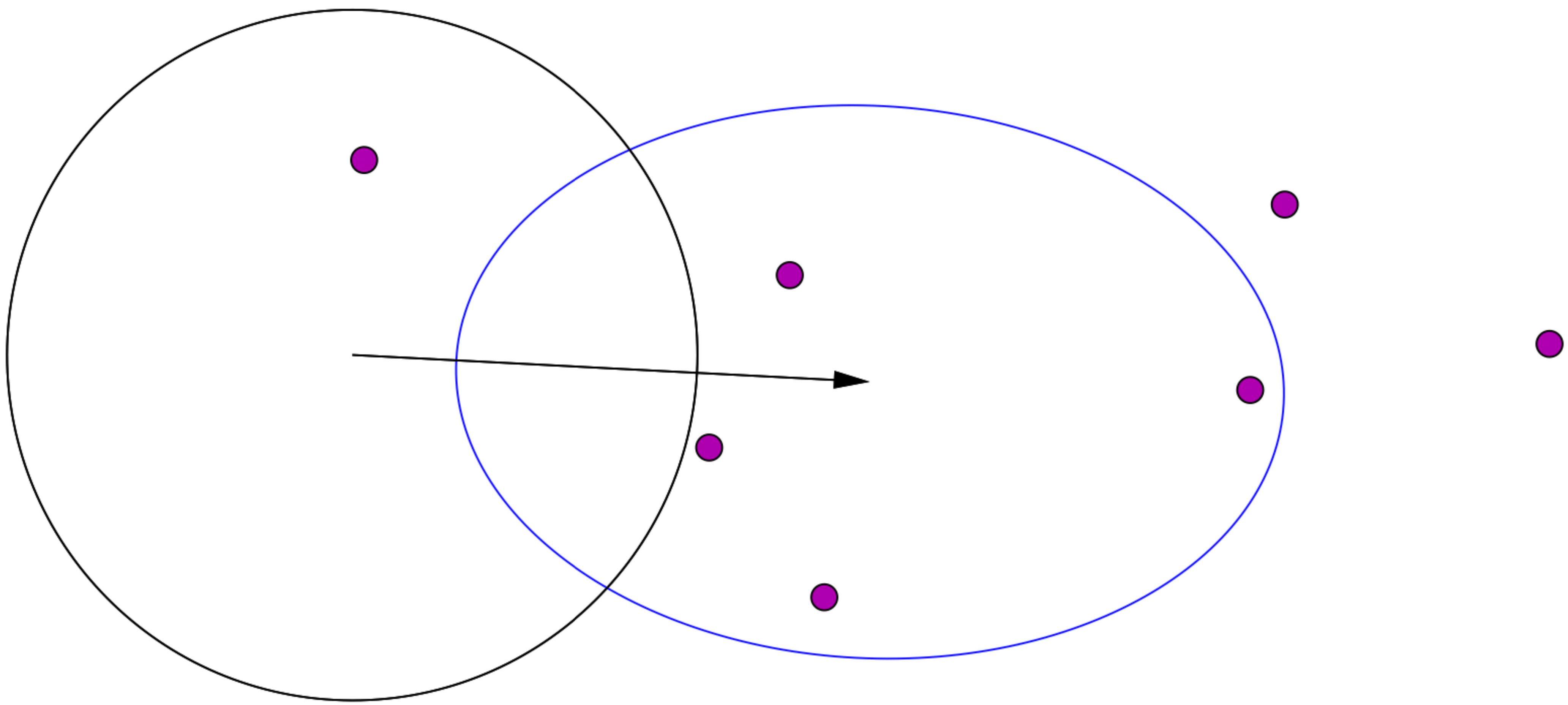
Rank-one update

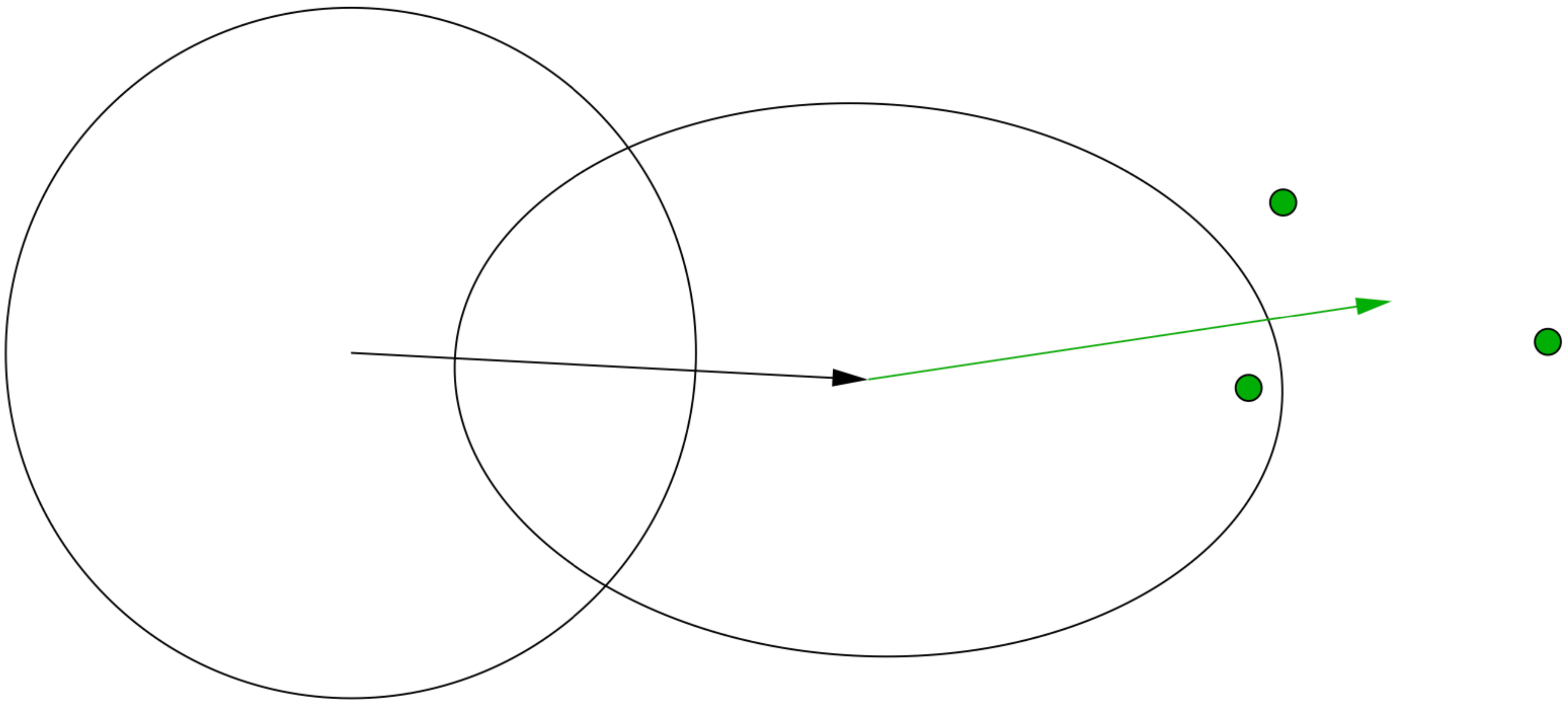


Rank-one update









$$p^c_{t+1} = (1-c_c)p^c_t + \sqrt{c_c(1-c_c)\mu_{\text{eff}}}\sqrt{C_t}\sum_{i=1}^{\mu} w_i U_{t+1}^{s_{t+1}(i)}$$

$$C_{t+1} = (1-c_\mu - c_1)C_t + c_\mu \underbrace{\sqrt{C_t}\left(\sum_{i=1}^{\mu} w_i U_{t+1}^{s_{t+1}(i)}[U_{t+1}^{s_{t+1}(i)}]^\top\right)\sqrt{C_t}}_{\textit{rank } \mu \textit{ update}} + c_1 \underbrace{p^c_{t+1}p^\top_{t+1}}_{\textit{rank 1 update}}$$

CMA-ES

Sampling + ranking:

$$X_{t+1}^{i} = m_t + \sigma_t \sqrt{C_t} U_{t+1}^{i} \quad i = 1, \ldots, \lambda$$

$\{U_t, t \geq 1\}$ i.i.d $U_{t+1}^{i} \sim \mathcal{N}(0, I_d)$

$$f\left(X_{t+1}^{s_{t+1}(1)}\right) \leq \ldots \leq f\left(X_{t+1}^{s_{t+1}(\lambda)}\right)$$

Update of θ_t:

$$m_{t+1} = \sum_{i=1}^{\mu} w_i X_{t+1}^{s_{t+1}(i)} = m_t + \sigma_t \sqrt{C_t} \sum_{i=1}^{\mu} w_i U_{t+1}^{s_{t+1}(i)}$$

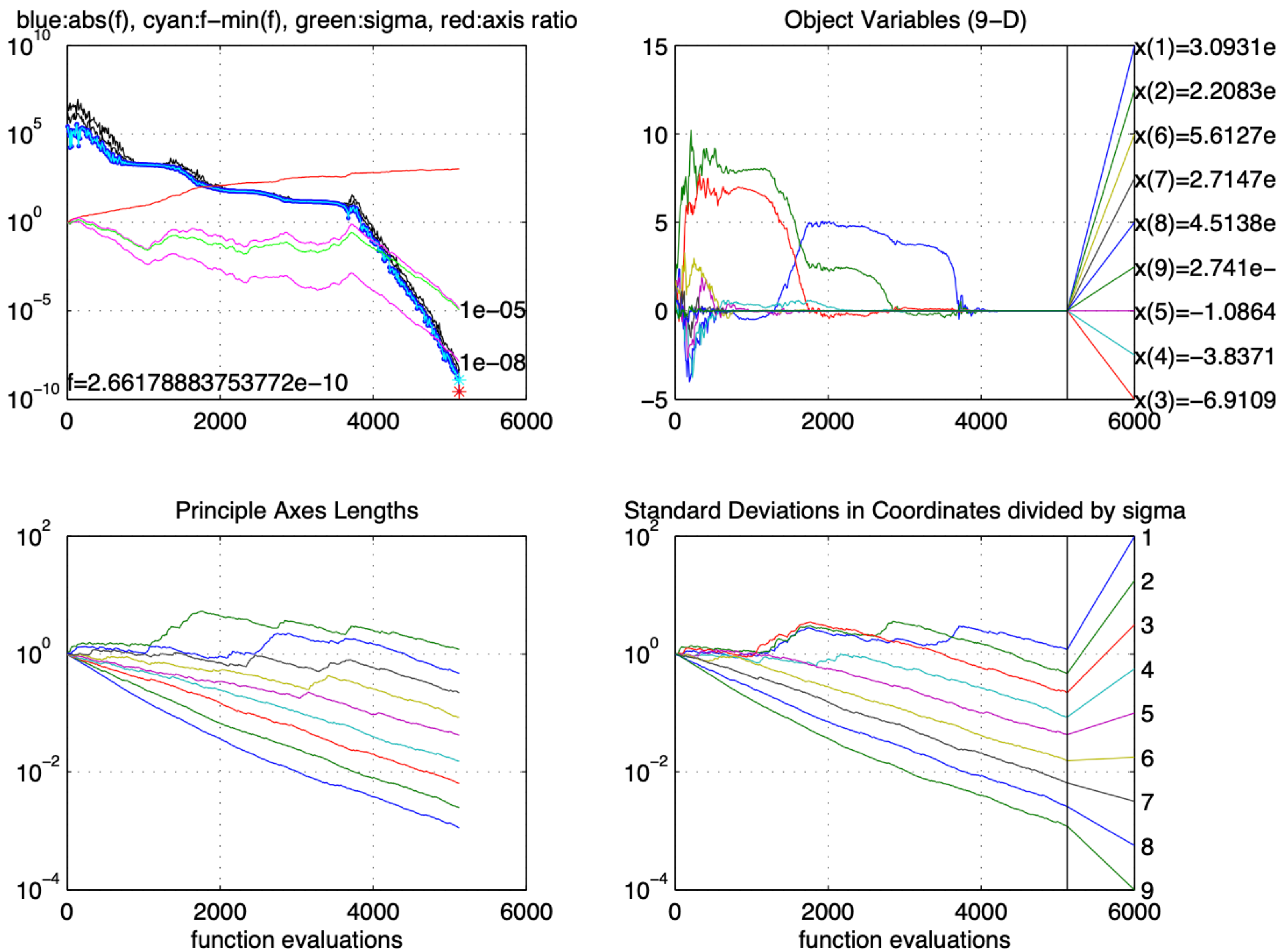
$$p_{t+1}^{\sigma} = (1 - c_\sigma) p_t^{\sigma} + \sqrt{c_\sigma (1 - c_\sigma) \mu_{\text{eff}}} \sum_{i=1}^{\mu} w_i U_{t+1}^{s_{t+1}(i)}$$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{c_\sigma}{d_\sigma} \left[\frac{\|p_{t+1}^{\sigma}\|}{E[\|\mathcal{N}(0, I_d)\|]} - 1 \right] \right)$$

$$p_{t+1}^{c} = (1 - c_c) p_t^{c} + \sqrt{c_c (1 - c_c) \mu_{\text{eff}}} \sqrt{C_t} \sum_{i=1}^{\mu} w_i U_{t+1}^{s_{t+1}(i)}$$

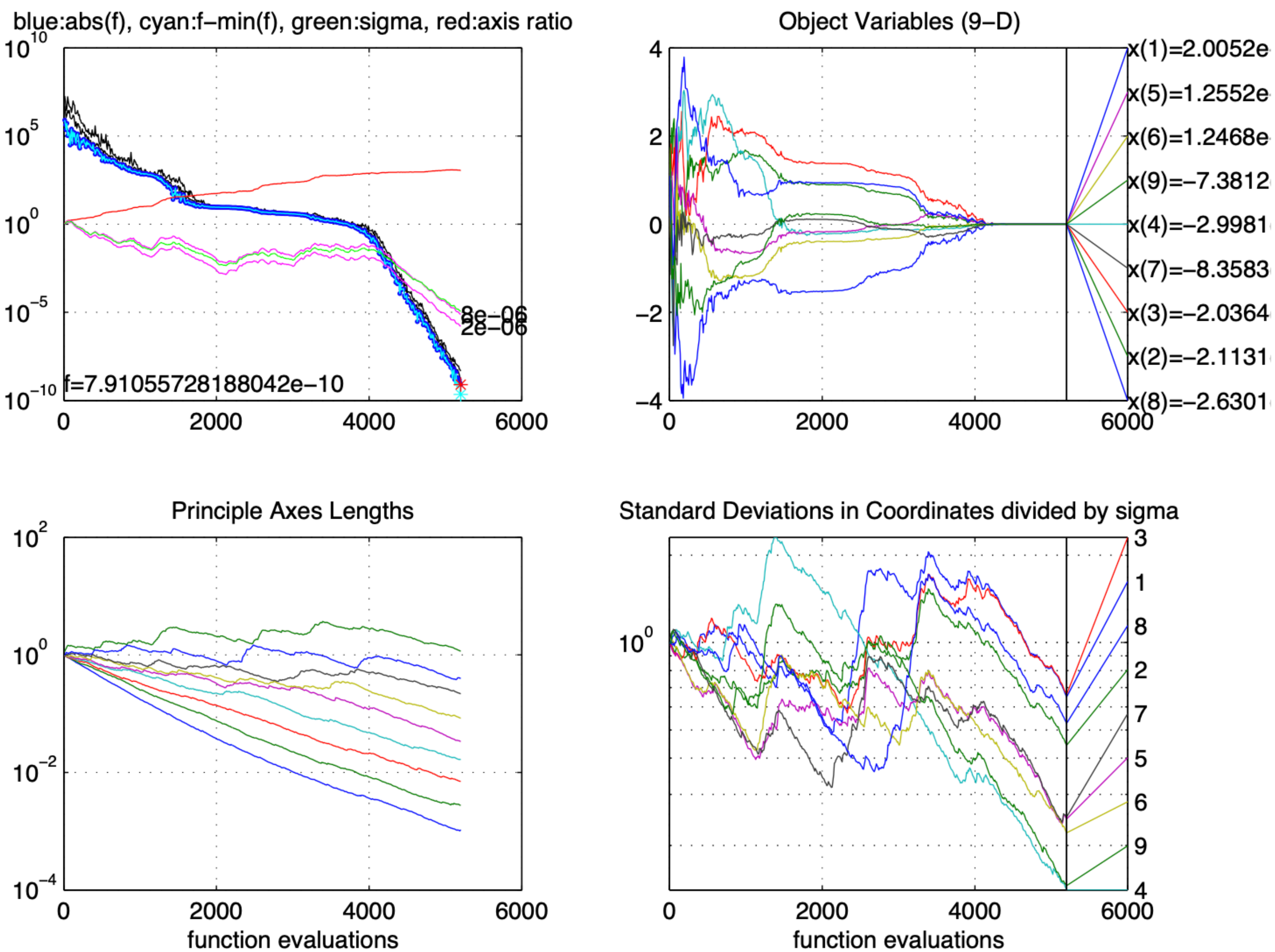
$$C_{t+1} = (1 - c_\mu - c_1) C_t + c_\mu \underbrace{\sqrt{C_t} \left(\sum_{i=1}^{\mu} w_i U_{t+1}^{s_{t+1}(i)} [U_{t+1}^{s_{t+1}(i)}]^\top \right) \sqrt{C_t}}_{\textit{rank } \mu \textit{ update}} + c_1 \underbrace{p_{t+1}^{c} p_{t+1}^{c\top}}_{\textit{rank 1 update}}$$

Visualization - experimentum crucis



$$f(\boldsymbol{x}) = \sum_{i=1}^{n} 10^{\alpha \frac{i-1}{n-1}} x_i^2, \alpha = 6$$

Visualization - experimentum crucis



$\mathbf{C} \propto \boldsymbol{H}^{-1}$ for all $g, \mathbf{H}$

$$f(\boldsymbol{x}) = g\left(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x}\right),\ g : \mathbb{R} \to \mathbb{R} \text{ stricly increasing}$$

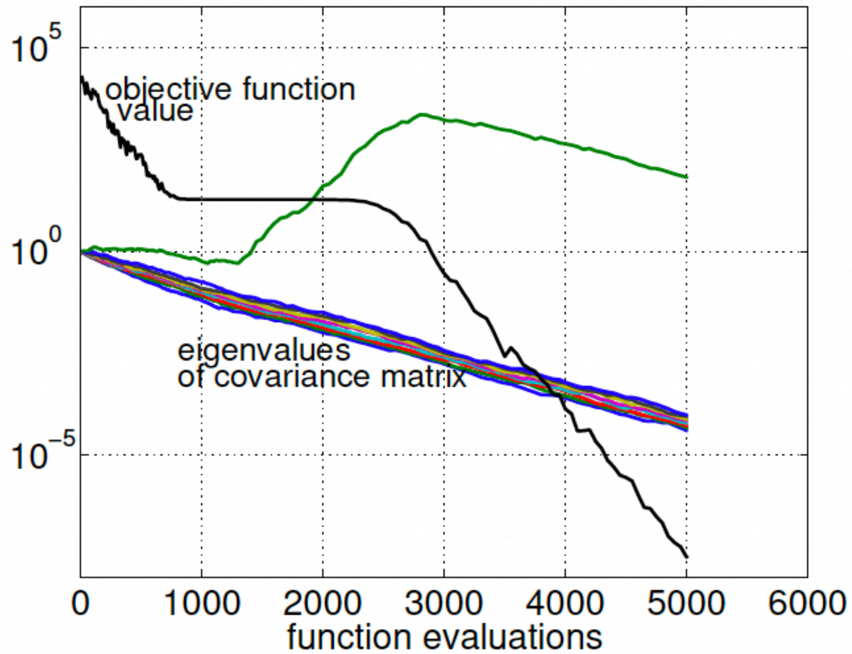
Linear Convergence and Learning Inverse Hessian

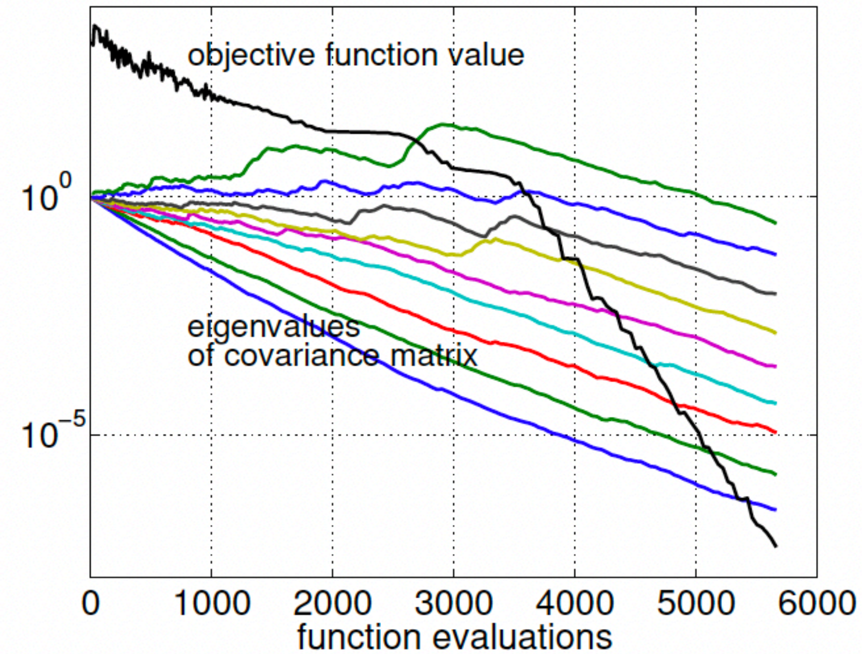
For all $g : \mathrm{Im}(f) \to \mathbb{R}$, strict increasing, for all $f(x) = \frac{1}{2}(x - x^\star)^\top H (x - x^\star)$ with $H \succ 0$ (SDP), when CMA-ES optimizes $x \mapsto g \circ f(x)$:

$$\frac{1}{t} \ln \frac{\|m_t - x^\star\|}{\|m_0 - x^\star\|} \xrightarrow[t\to\infty]{} -\mathrm{CR}$$

$$C_t \propto \alpha_t H^{-1} \quad \text{with } \alpha_t \to 0$$

Two examples:

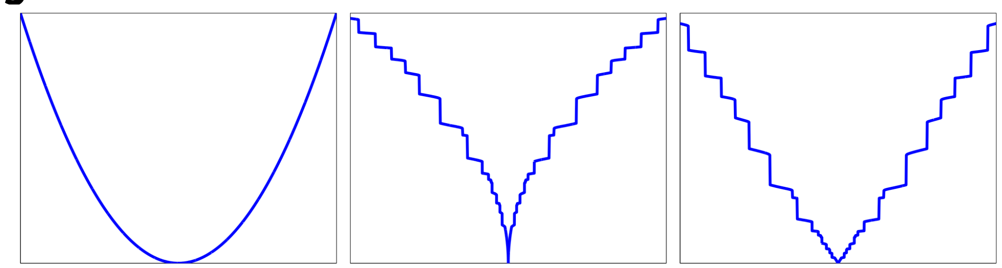




Key of success (?) - Main features

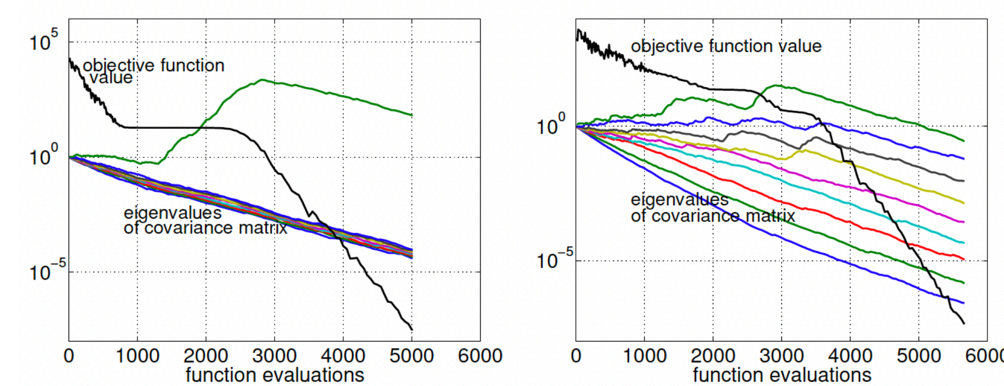
Invariances:

- to strictly increasing transformation of objective functions



- affine invariance

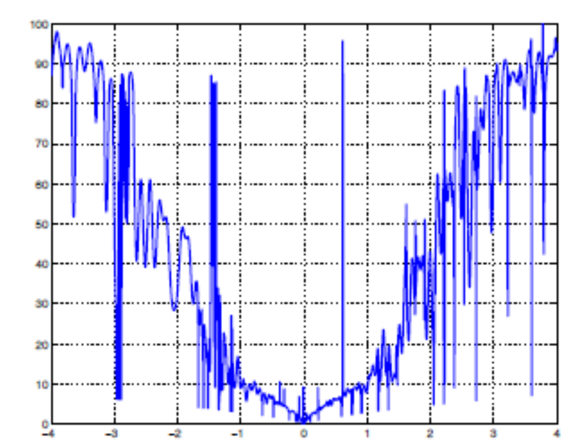
$$x \mapsto \|x\|^2 \Leftrightarrow x \mapsto x^\top H x \quad \text{for all } H \geq 0$$



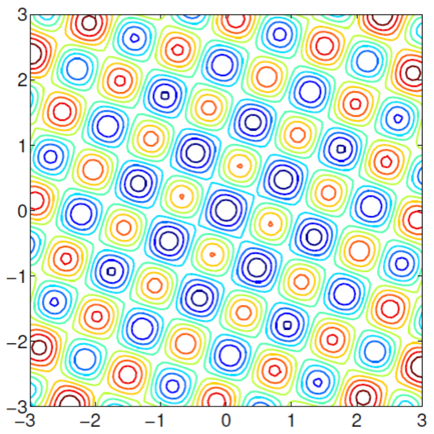
Control of diversity via step-size

- sees the function at the scale σ_t

robust to noise, outliers, irregularities



Key of success (?) - Main features



Population-based algorithm: increase λ makes algorithm more global

Careful tuning of all constants (depending on dimension, ...)

parameter-free algorithm

Original designers did not care about having a convergence proof

- successful approach: framework not restricted to updates for which proofs can be made
- was challenging to prove linear convergence of CMA-ES [PhD thesis Armand Gissler, 2024]

Thanks!