the speed–robustness trade-off for iterative optimization algorithms

Bryan Van Scoy
Miami University

Laurent Lessard
Northeastern University

Fall, 2021

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x)$$

In this talk:

- Iterative algorithms can be viewed as robust controllers.
- Algorithms can be **designed**, in much the same way that controllers can be designed.
- Controls and optimization!

Noisy oracle model

$$x^\star \in \arg\min_{x \in \mathbb{R}^d} f(x)$$

We can query a noisy oracle $g(x) = \nabla f(x) + w$,
where w is zero-mean and independent across queries.

Use cases:

- Must approximate ∇f via finite differencing.
- Requires solving auxiliary optimization problem numerically or simulating; inexact solutions.
- Empirical risk minimization in the context of learning; evaluate expected value via sample-based approximations.

Gradient descent (GD)

$$x_{t+1} = x_t - \alpha\, g(x_t)$$

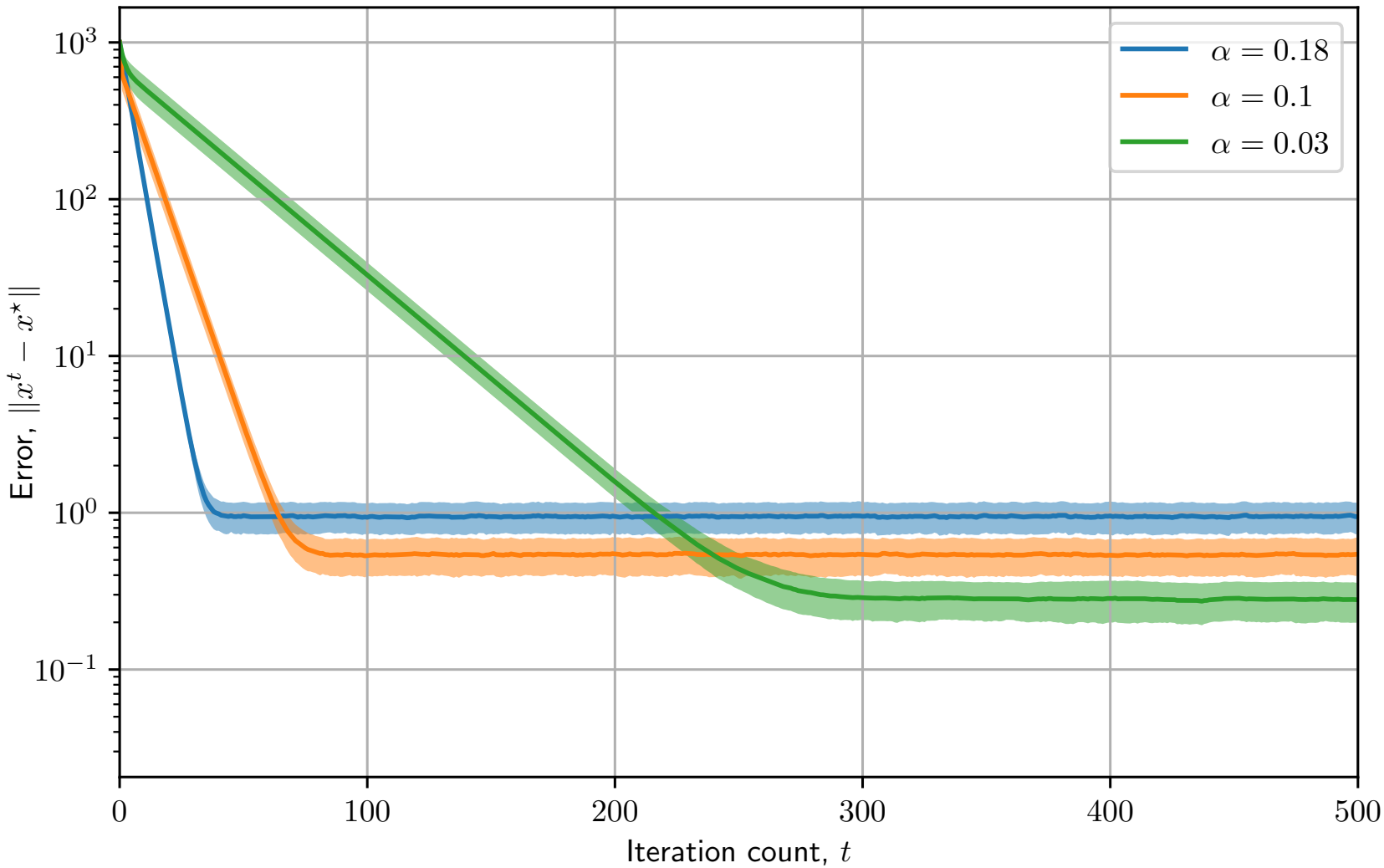
Geometric phase

- Noise is small compared to gradient
- x_t makes rapid progress toward $x^\star$

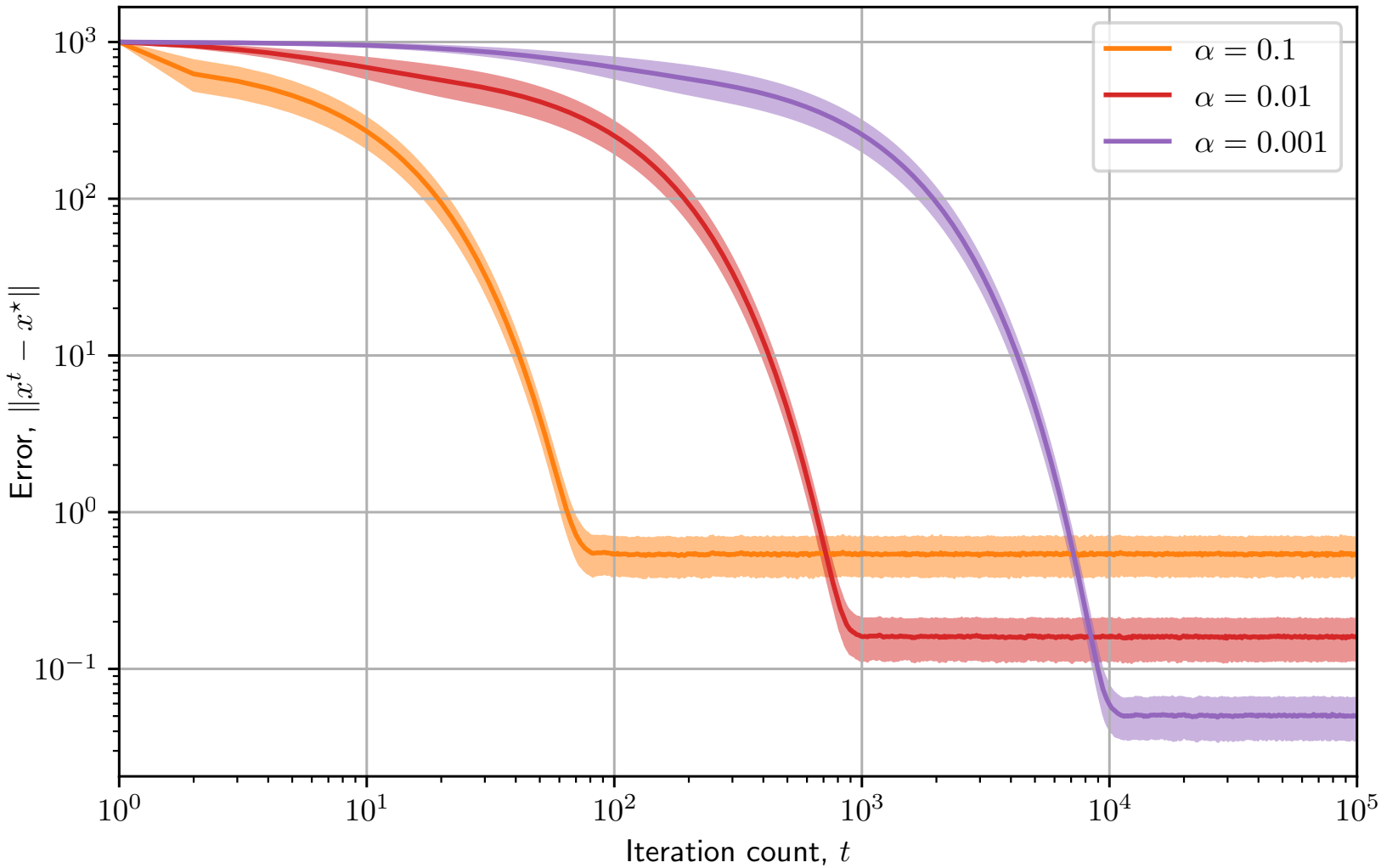
Stationary phase

- Noise is comparable to gradient
- x_t moves randomly in a ball about $x^\star$

Random quadratic function: $f(x) = x^{\mathsf{T}} Q x$, $d = 10$.
Eigenvalues satisfy $1 \leq \lambda(Q) \leq 10$.



Random quadratic function: $f(x) = x^{\mathsf{T}} Q x$, $d = 10$.
Eigenvalues satisfy $1 \leq \lambda(Q) \leq 10$.



Acceleration

Polyak acceleration (Heavy Ball)

$$x_{t+1} = x_t - \alpha\, g(x_t) + \beta(x_t - x_{t-1})$$

Nesterov acceleration (Fast Gradient)

$$\begin{aligned} y_t &= x_t + \beta(x_t - x_{t-1}) \\ x_{t+1} &= y_t - \alpha\, g(y_t) \end{aligned}$$

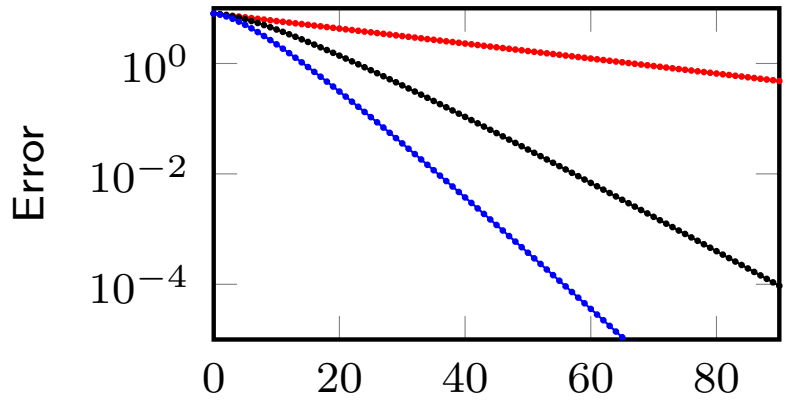
- Similar geometric & stationary phases
- More parameters to tune
- Potentially better performance!

Gradient method

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

Heavy ball method

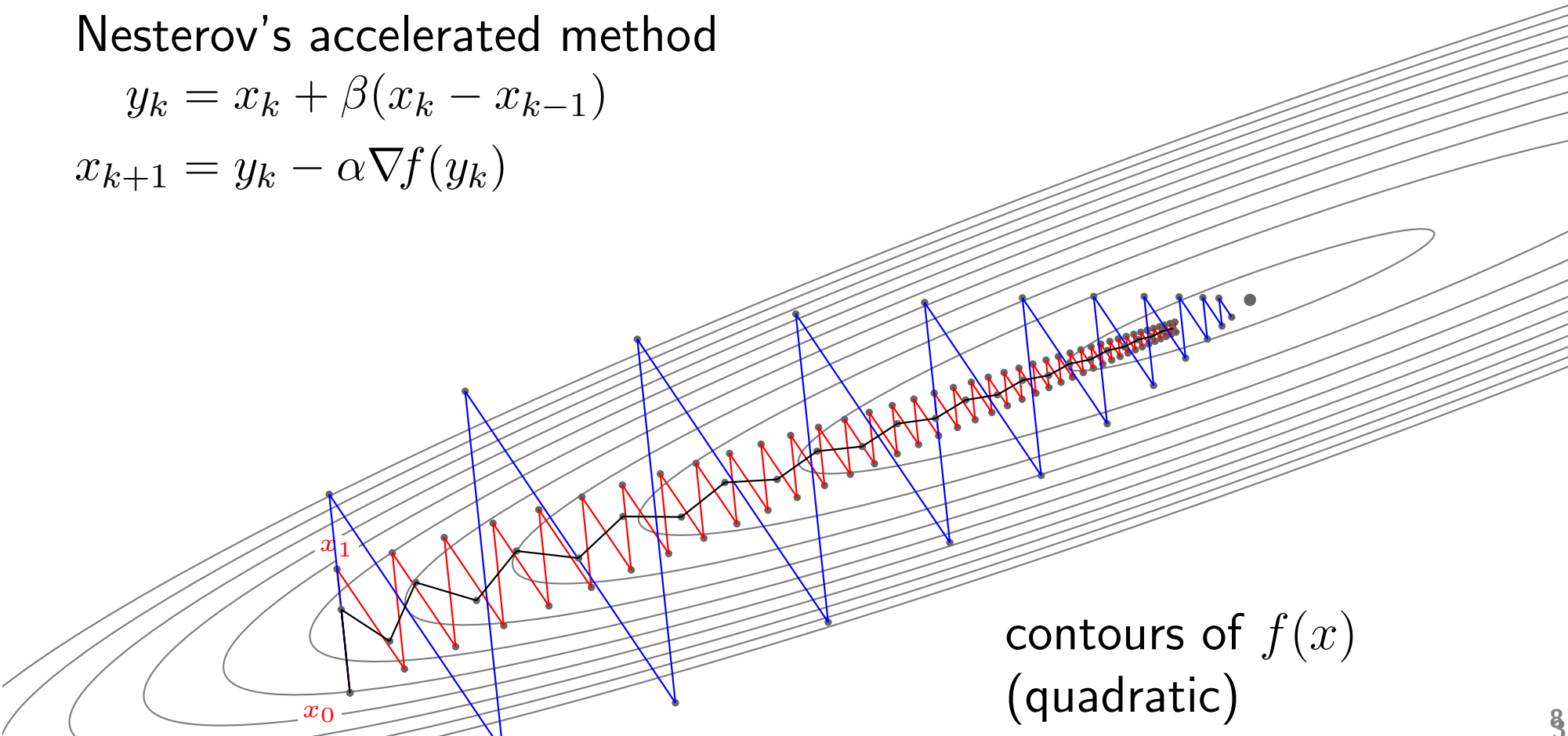
$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$$



Nesterov's accelerated method

$$y_k = x_k + \beta(x_k - x_{k-1})$$

$$x_{k+1} = y_k - \alpha \nabla f(y_k)$$



contours of $f(x)$
(quadratic)

Performance metrics

Rate of convergence (ρ)

$$\|x_k - x^\star\| \leq (\text{const}) \cdot \rho^k$$

Smaller ρ means faster convergence (no noise regime).

Sensitivity to noise (γ)

$$\limsup_{N\to\infty} \frac{1}{N} \mathbf{E} \sum_{k=0}^{N-1} \|x_k - x^\star\|^2 = \gamma^2$$

Smaller γ means more noise robustness (smaller ball).

Questions

How can we mediate the trade-off between speed and robustness for accelerated algorithms?

Can we design algorithms that are Pareto-optimal for different function classes? What will they look like?

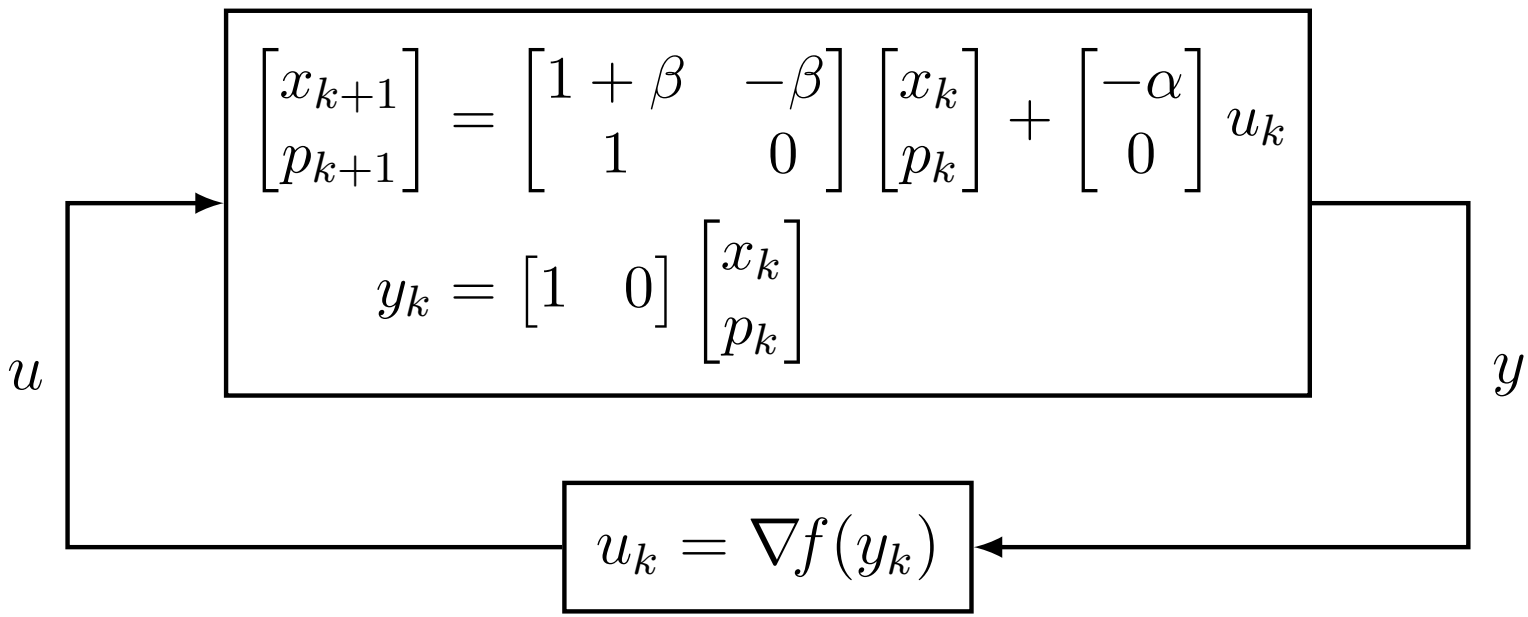
Outline

- Algorithms as dynamical systems
- Three-parameter family of algorithms
- Quadratic functions
  - Robust Heavy Ball
- Strongly convex functions
  - Robust Accelerated Method
- Numerical validation

Dynamical system interpretation

Heavy ball: $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$

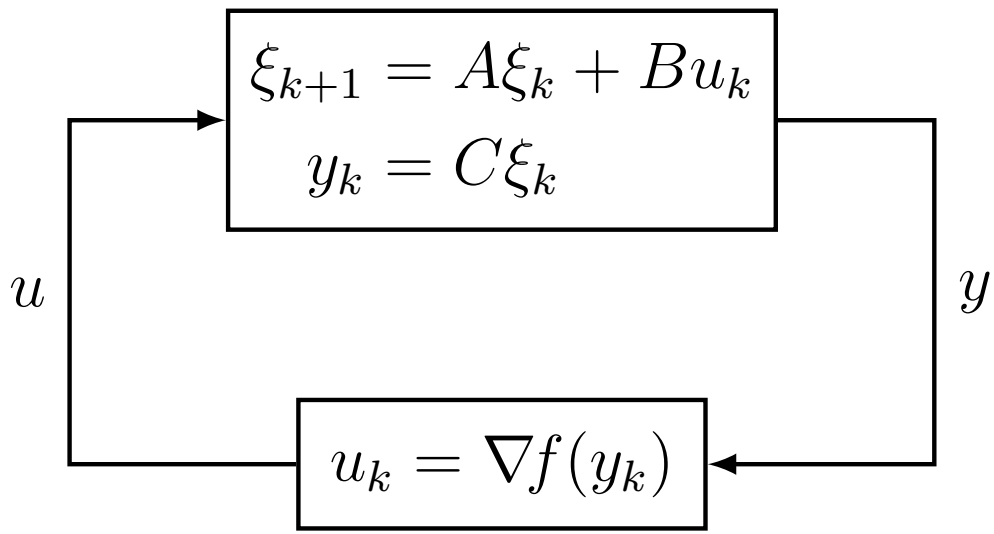
Define $u_k := \nabla f(x_k)$ and $p_k := x_{k-1}$



Dynamical system interpretation

Heavy ball: $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$

Define $u_k := \nabla f(x_k)$ and $p_k := x_{k-1}$



3-parameter family

$$
\begin{aligned}
y_t &= x_t + \beta(x_t - x_{t-1}) \\
z_t &= x_t + \eta(x_t - x_{t-1}) \\
x_{t+1} &= y_t - \alpha\, g(z_t)
\end{aligned}
$$

Generalization of Polyak and Nesterov acceleration:

- Recovers Gradient descent when $\beta = \eta = 0$.
- Recovers Polyak acceleration when $\eta = 0$.
- Recovers Nesterov acceleration when $\eta = \beta$.

3-parameter family

$$\xi_{k+1} = \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix} \xi_k + \begin{bmatrix} -\alpha \\ 0 \end{bmatrix} u_k$$
$$y_k = \begin{bmatrix} 1+\eta & -\eta \end{bmatrix} \xi_k$$

When designing algorithms, we can:

- Search over all (A, B, C) of a given size.
- Search over the specific parameterization (α, β, η)

Quadratic case $Q_{m,L}$

$Q_{m,L}$: Functions of the form $f(x) = x^{\mathsf{T}} Q x$ where $mI_d \preceq Q \preceq LI_d$

- **Heavy Ball** (HB) achieves fastest possible rate, when used with the tuning:

$$\alpha = \frac{4}{(\sqrt{L}+\sqrt{m})^2}, \qquad \beta = \left(\frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}}\right)^2, \qquad \eta = 0$$

Quadratic case $Q_{m,L}$

Let $f(y) = \frac{1}{2}(y - y^\star)^\mathsf{T} Q (y - y^\star)$.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \\ u_k &= \nabla f(y_k) + w_k \end{aligned}$$

Closed-loop map:

$$(x_{k+1} - x^\star) = (A + BQC)(x_k - x^\star) + Bw_k$$

Quadratic case $Q_{m,L}$

$$(x_{k+1} - x^\star) = (A + BQC)(x_k - x^\star) + Bw_k$$

- ρ is the rate of convergence when $w_k = 0$ (spectral radius of $A + BQC$).
- γ^2 is the squared $\mathcal{H}_2$-norm of the system (steady-state covariance).

Quadratic performance

$$\rho = \sup_{q \in [m,L]} \rho\big(A + qBC\big).$$

If $\rho < 1$, then

$$\gamma^2 = \sup_{q \in [m,L]} \sigma^2 d \cdot (B^\mathsf{T} P B),$$

where P is the solution to

$$(A + qBC)^\mathsf{T} P (A + qBC) - P + C^\mathsf{T} C = 0.$$

Both ρ and γ are nonconvex functions of (A, B, C).

Quadratic performance of 3-parameter algorithms

$$\rho = \max_{q \in \{m, L\}} \begin{cases} \sqrt{\beta - \alpha\eta q} & \text{if } \Delta < 0 \\ \frac{1}{2}\left(|\beta + 1 - \alpha q - \alpha\eta q| + \sqrt{\Delta}\right) & \text{if } \Delta \geq 0 \end{cases}$$

$$\text{where } \Delta := (\beta + 1 - \alpha q - \alpha\eta q)^2 - 4(\beta - \alpha\eta q).$$

If $\rho < 1$, then

$$\gamma^2 = \max_{q \in \{m, L\}} \frac{\sigma^2 d\, \alpha(1 + \beta + (1 + 2\eta)\alpha\eta q)}{q(1 - \beta + \alpha\eta q)(2 + 2\beta - (1 + 2\eta)\alpha q)}$$

Both are easy to evaluate and analyze!

Robust Heavy Ball (RHB)

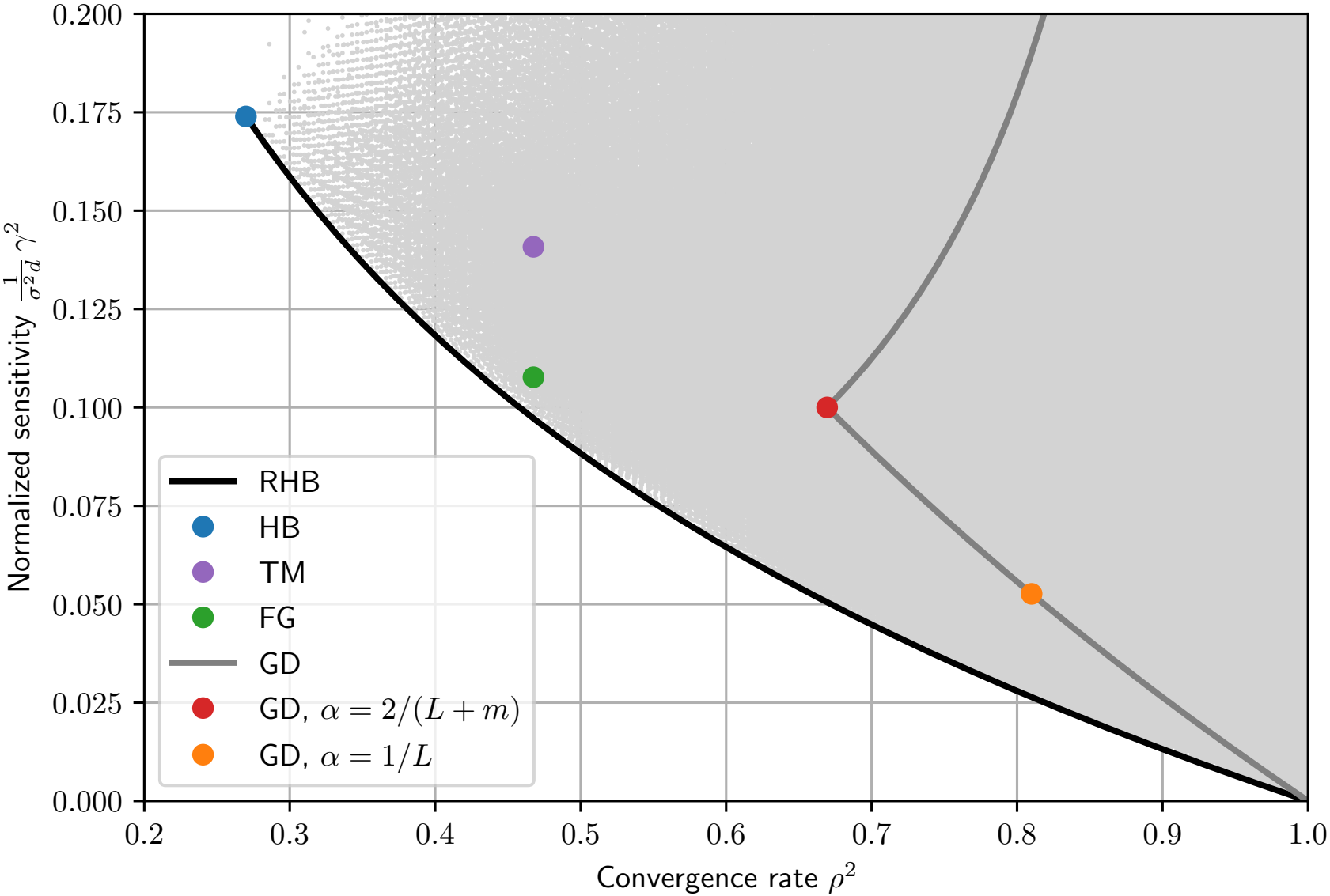
Let $\rho \in \left[\frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}}, 1\right)$. RHB is the 3-parameter algorithm

$$\alpha = \tfrac{1}{m}(1-\rho)^2, \qquad \beta = \rho^2, \qquad \eta = 0$$

On the class $Q_{m,L}$, RHB achieves

$$\rho_{\mathsf{RHB}} = \rho \quad \text{and} \quad \gamma^2_{\mathsf{RHB}} = \frac{\sigma^2 d}{m^2}\frac{1-\rho^4}{(1+\rho)^4}.$$

Setting $\rho = \frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}}$ recovers ordinary Heavy Ball.



(ρ, γ) tradeoff for $Q_{m,L}$, with $m = 1$ and $L = 10$.

Strongly convex case $F_{m,L}$

$F_{m,L}$: Differentiable functions for which:

1. $f(y) - \frac{1}{2}m\|y\|^2$ is a convex function of y
2. $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^d$

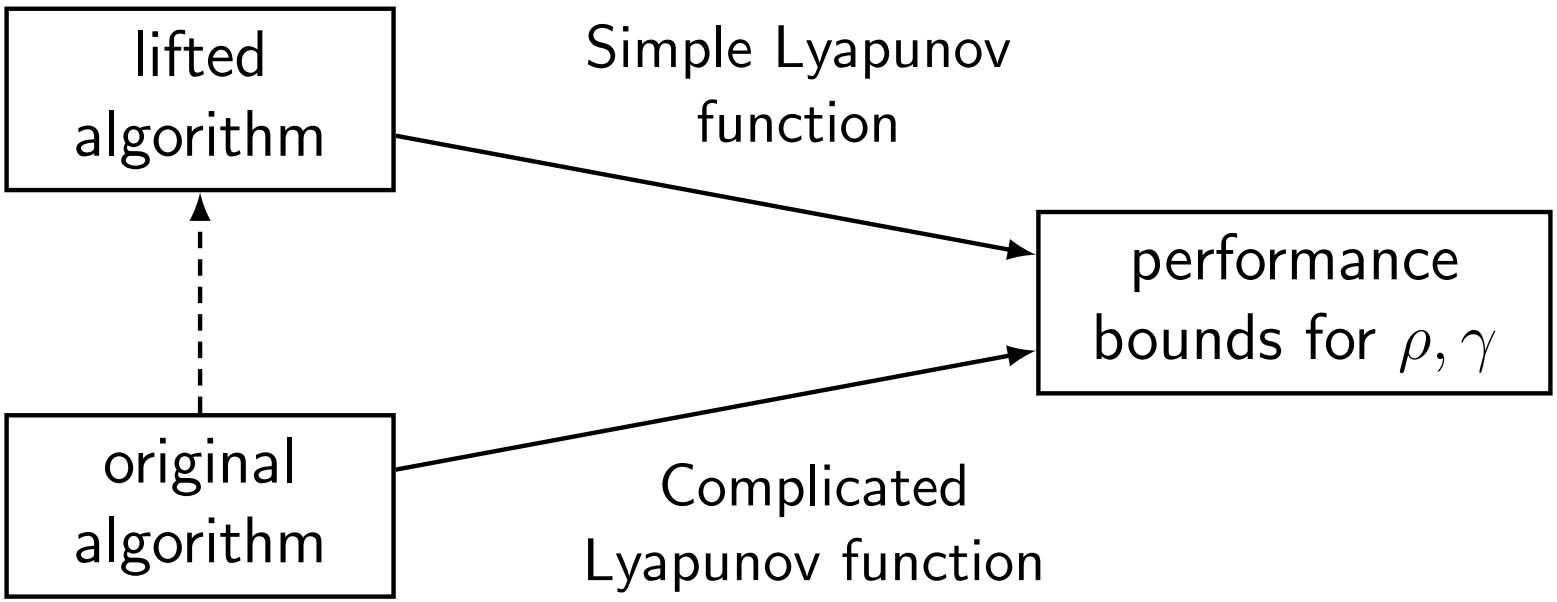
- **Triple Momentum** (TM) achieves fastest possible rate.

$$\alpha = \frac{\sqrt{L}-\sqrt{m}}{L^{3/2}}, \qquad \beta = \frac{(\sqrt{L}-\sqrt{m})^2}{L+\sqrt{mL}}, \qquad \eta = \frac{(\sqrt{L}-\sqrt{m})^2}{2L-m+\sqrt{mL}}$$

- **Fast Gradient** (FG) is a popular choice.

$$\alpha = \frac{1}{L}, \qquad \beta = \frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}}, \qquad \eta = \frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}}$$

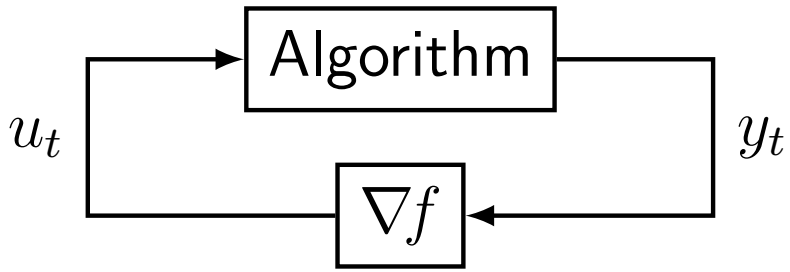
Outline of approach for $F_{m,L}$



Lifting increases the number of variables but allows use of a simpler Lyapunov function.

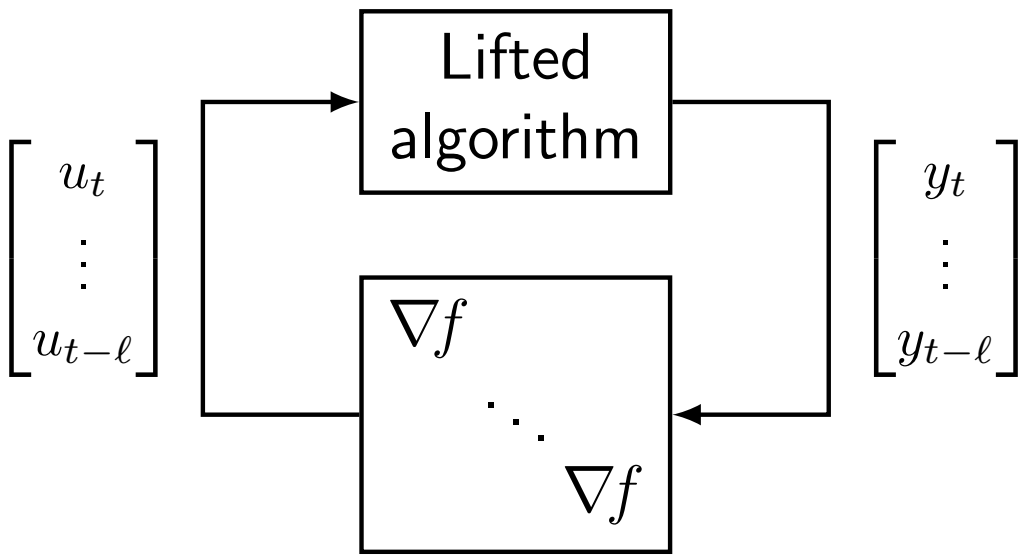
Lifted dynamics

Original system:



state: x_t

Lifted system:



state: $\bar{x}_t = \begin{bmatrix} x_t \\ y_{t-1} \\ \vdots \\ y_{t-\ell} \\ u_{t-1} \\ \vdots \\ u_{t-\ell} \end{bmatrix}$

Lyapunov approach

Certifying a convergence rate

If $x_{k+1} = f(x_k)$ and we can find a function $V(x)$ satisfying

$$V(x) \geq \|x\|^2 \qquad \text{(positivity)}$$
$$V(f(x)) \leq \rho^2 V(x) \qquad \text{(decrease condition)}$$

Then we have geometric decrease:

$$\|x_k\|^2 \leq V(x_k) \leq \rho^2 V(x_{k-1}) \leq \cdots \leq \rho^{2k} V(x_0)$$

Lyapunov approach

Certifying sensitivity

If $x_{k+1} = f(x_k, w_k)$ and we can find a function $V(x)$ satisfying

$$\mathbf{E}\, V(x) \geq 0 \qquad \text{(positivity)}$$

$$\mathbf{E}\, V(f(x,w)) - \mathbf{E}\, V(x) + \mathbf{E}\|x\|^2 \leq \gamma^2 \qquad \text{(decrease)}$$

Then we have bounded steady-state covariance:

$$\mathbf{E}\, V(x_N) - \mathbf{E}\, V(x_0) + \mathbf{E} \sum_{k=0}^{N-1} \|x_k\|^2 \leq N\gamma^2$$

$$\implies \limsup_{N\to\infty} \frac{1}{N}\, \mathbf{E} \sum_{k=0}^{N-1} \|x_k\|^2 \leq \gamma^2$$

Interpolation

Interpolation in $F_{m,L}$ (Taylor et al. 2017)

Let $y_1, \ldots, y_k \in \mathbb{R}^d$ and $u_1, \ldots, u_k \in \mathbb{R}^d$ and $f_1, \ldots, f_k \in \mathbb{R}$. The following two statements are equivalent.

1. There exists a function $f \in F_{m,L}$ such that $f(y_i) = f_i$ and $\nabla f(y_i) = u_i$ for $i = 1, \ldots, k$.
2. For all $i, j \in \{1, \ldots, k\}$,

$$\tfrac{mL}{2\,(L-m)} \left(\|y_i - y_j\|^2 + \tfrac{1}{mL} \|u_i - u_j\|^2 - \tfrac{2}{L} (u_i - u_j)^{\mathsf{T}} (y_i - y_j) \right) \leq u_i^{\mathsf{T}} (y_i - y_j) - (f_i - f_j)$$

Valid inequalities: $\Pi(\Lambda) := \sum_{i,j} \lambda_{ij} \Pi_{ij} \geq 0$.

Lyapunov with inputs

- Our system has inputs, i.e. $x_{k+1} = f(x_k, u_k)$.
- Inputs are result of feedback: $u_k = \nabla f(y_k)$.
- Interpolation conditions state that $\Pi(\Lambda) \geq 0$.
- Use S-procedure; instead, find Λ such that

$$V(f(x,u)) + \Pi(\Lambda) \leq \rho^2 V(x) \quad \text{for all } x, y, u$$

- Since $\Pi(\Lambda) \geq 0$, this implies $V(f(x,u)) \leq \rho^2 V(x)$.

Higher lifting dimension means more interpolation conditions, and potentially less conservatism.

Efficiency

- Use $V(x) = x_k^\mathsf{T} P x_k + p^\mathsf{T} f_k$; quadratic in algorithm state and linear in function values.
- Interpolation conditions $\Pi(\Lambda)$ are also quadratic in algorithm state and linear in function values.
- Search for Lyapunov function is a linear matrix inequality.
- Size does *not* depend on function domain dimension d.
- Size scales with lifting dimension ℓ.
- $\ell = 1$ appears sufficient to compute best ρ bound.
- $\ell = 4$ appears sufficient to compute best γ bound.

Given $(\alpha, \beta, \eta, m, L)$, can compute tightest possible bounds for (ρ, γ) in < 100 ms on a laptop.

Context

Connection to IQCs:

- Related to IQC approach for algorithm analysis [Lessard, Recht, Packard. 2016].
- A subset of $\Pi(\Lambda)$ corresponds to Zames–Falb IQCs.
- Results are similar when search is restricted to such Π.

Connection to PEP framework:

- Related to Performance Estimation Program [Taylor, Hendrickx, Glineur. 2017].
- Uses finite horizon performance instead
- Tight bounds, but LMI size depends on horizon length

Design challenges

- Not as straightforward as $Q_{m,L}$ case because we do not have an explicit function $(\alpha, \beta, \eta) \mapsto (\rho, \gamma)$.
- In principle, solution is a *semialgebraic set*.
- Optimality conditions yield polynomials of degree > 200 that are not solvable analytically.

Challenge is to find algorithms that:

- Have relatively simple algebraic expressions. Avoid numerical solutions if possible.
- Are as close to being optimal as possible.

General strategy

1. Use numerical solver (e.g. Nelder–Mead) to find locally optimal (α, β, η), e.g. fix ρ and minimize γ.

2. Write LMI as polynomial optimization problem: convert semidefinite constraints into determinant inequalities.

3. Substitute numerical solution to find active constraints and dual variables. At optimality, matrices in LMI will drop rank.

4. Look for analytic solution to system of active constraints. Might require trying different elimination orderings.

Robust Accelerated Method (RAM)

Let $\rho \in \left[1 - \sqrt{\frac{m}{L}}, 1\right)$. RAM is the 3-parameter algorithm

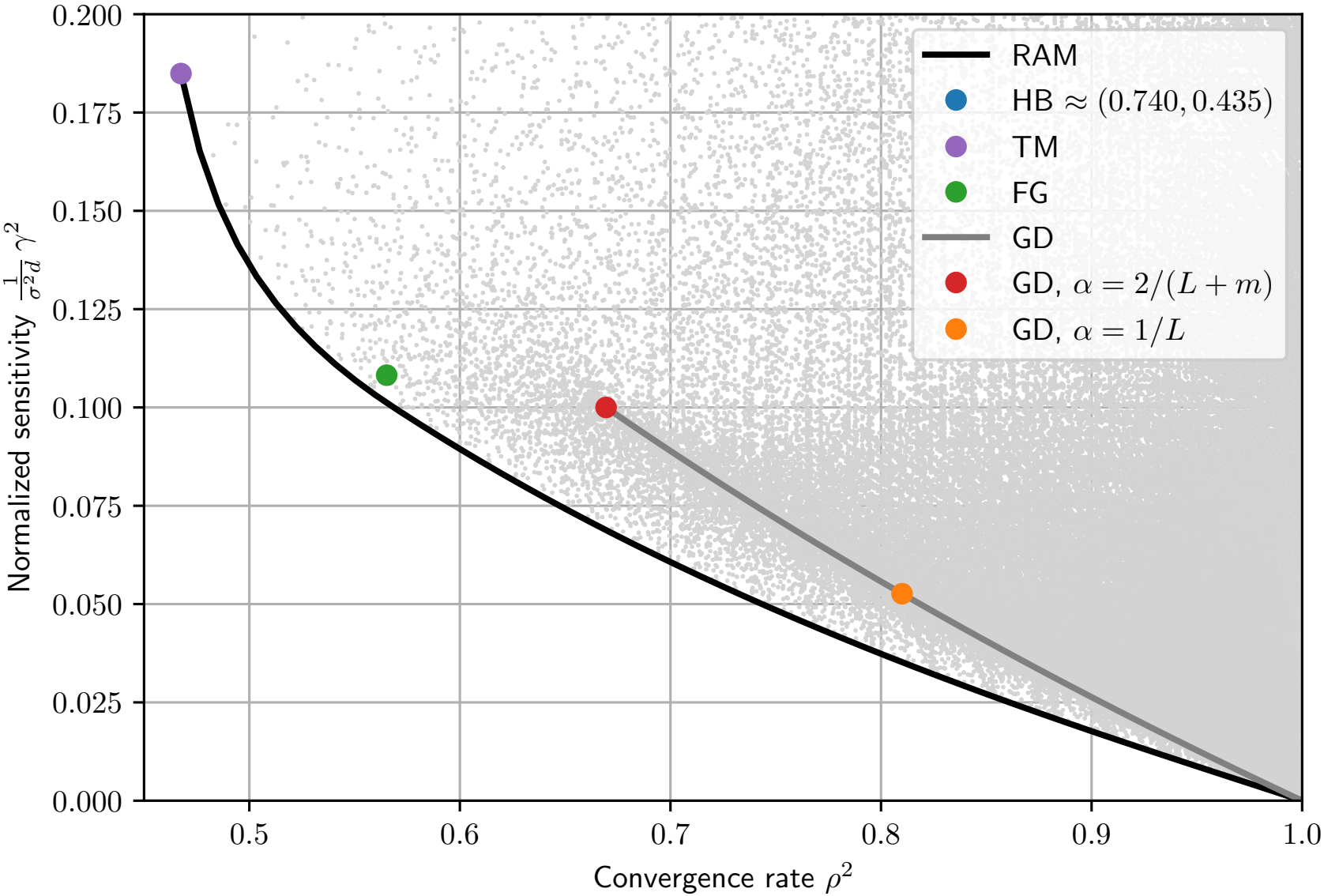
$$\alpha = \frac{(1+\rho)(1-\rho)^2}{m}, \qquad \beta = \rho\,\frac{L\,(1-\rho+2\rho^2)-m\,(1+\rho)}{(L-m)(3-\rho)},$$
$$\eta = \rho\,\frac{L\,(1-\rho^2)-m\,(1+2\rho-\rho^2)}{(L-m)(3-\rho)(1-\rho^2)}.$$

On the class $F_{m,L}$, RAM achieves $\rho_{\mathsf{RAM}} = \rho$.

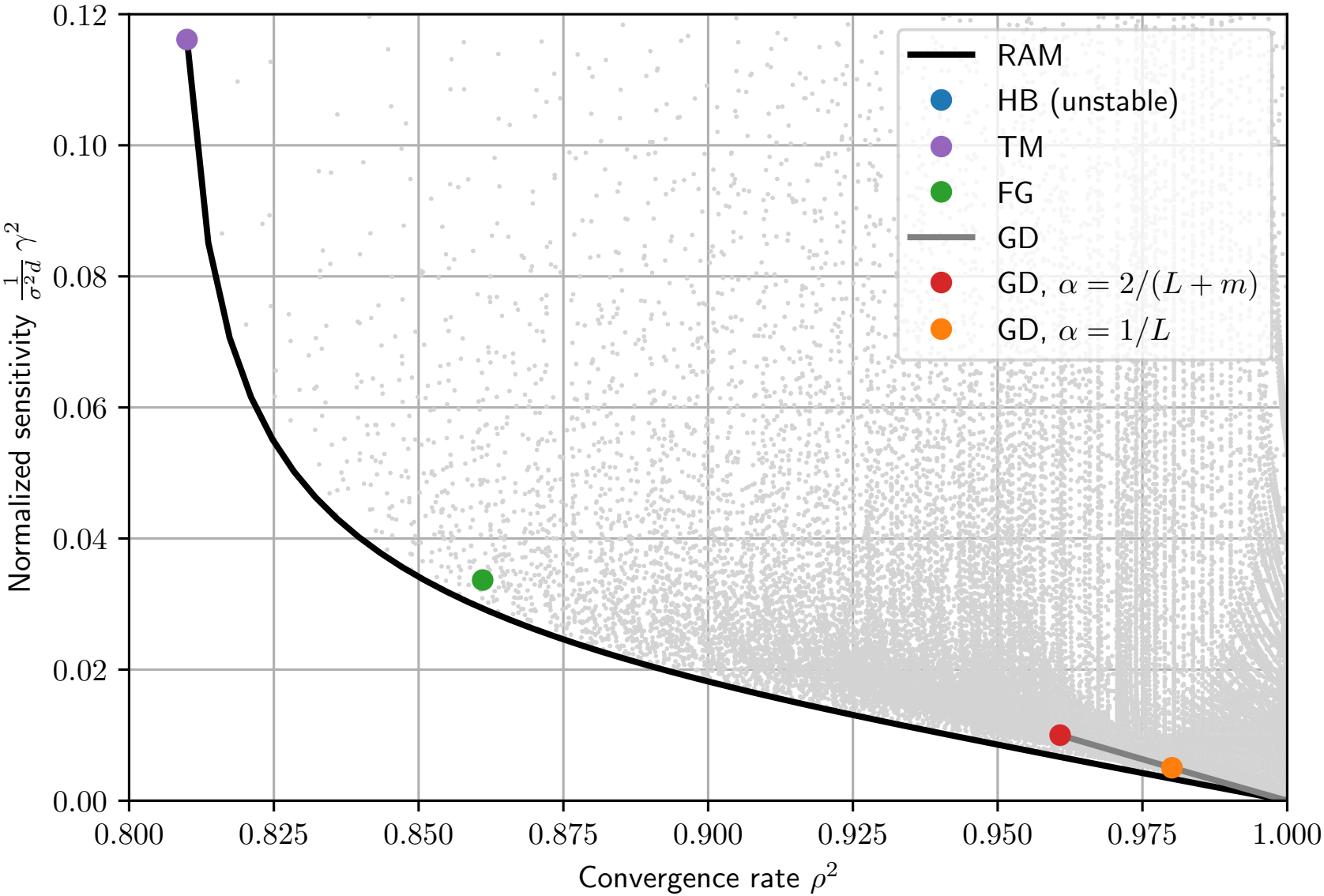
Setting $\rho = 1 - \sqrt{\frac{m}{L}}$ recovers Triple Momentum.

For larger ρ, RAM is *near*-Pareto optimal

(ρ, γ) tradeoff for $F_{1,10}$.



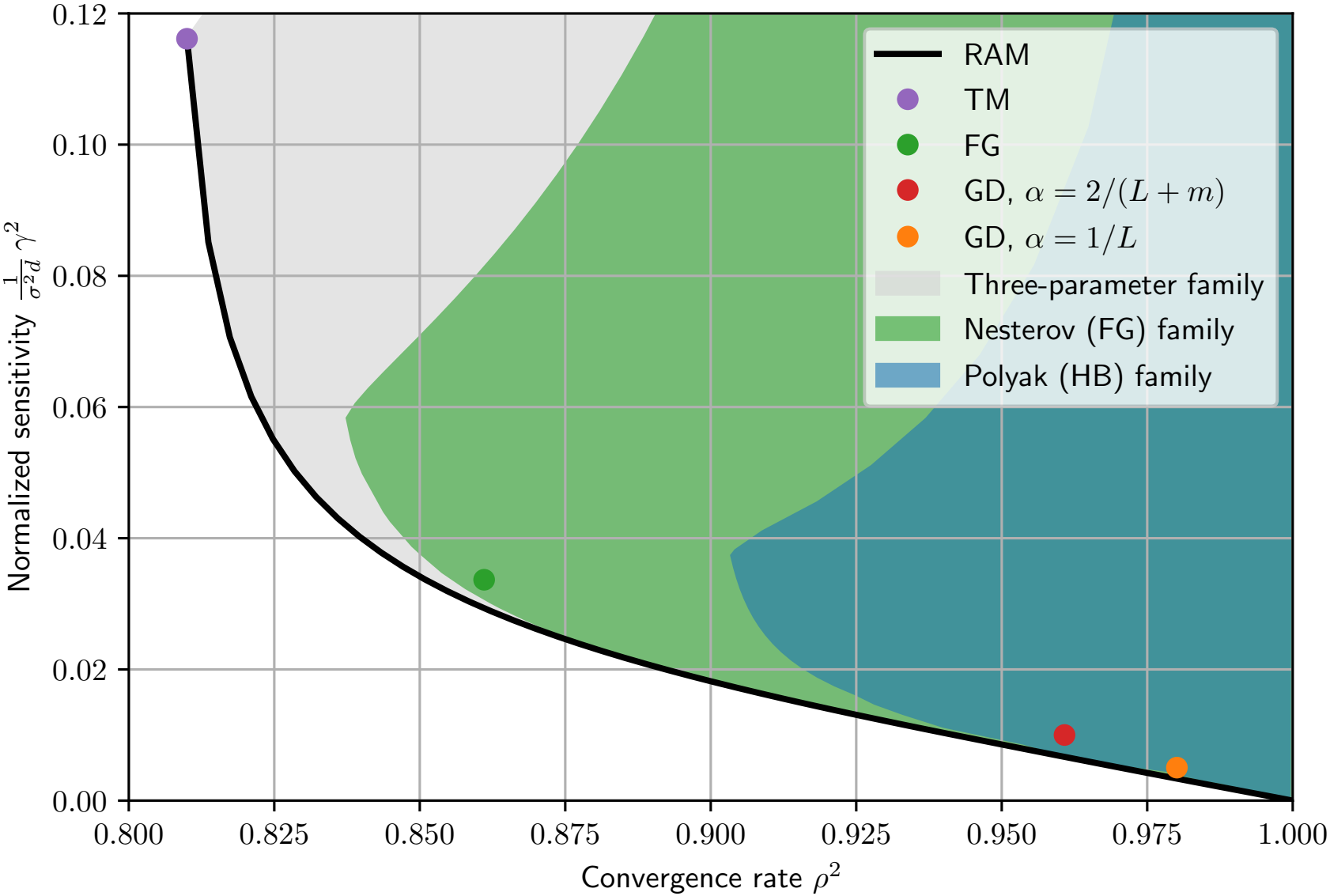
(ρ, γ) tradeoff for $F_{1,100}$.



Simulation

RAM uses (α, β, η). What if we use only Polyak or only Nesterov acceleration?

Nesterov and Polyak coverage for $F_{1,100}$.



Simulation

Nesterov's worst-case quadratic:

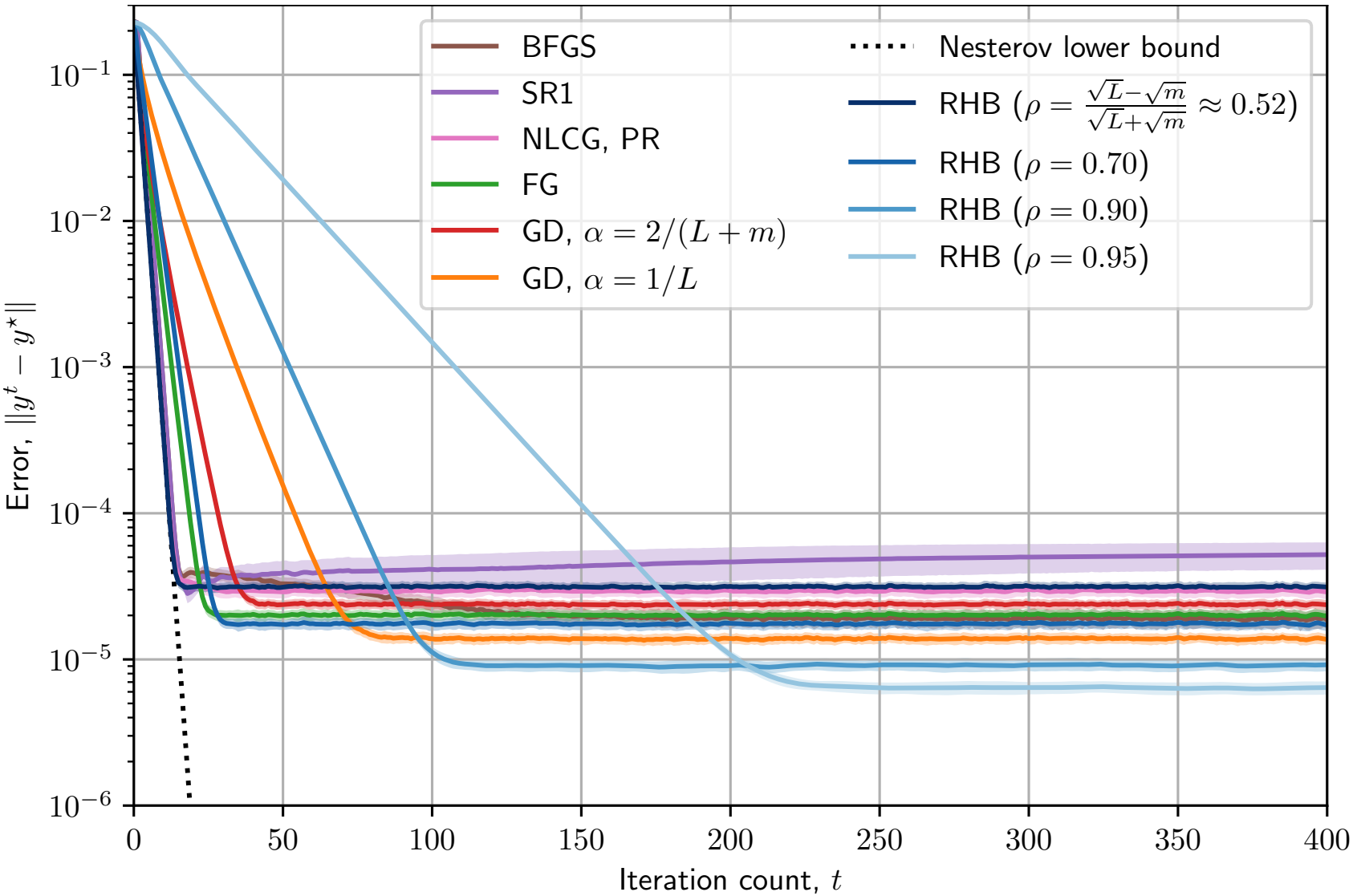
$$\nabla^2 f(x) = \begin{bmatrix} \frac{L+m}{2} & \frac{L-m}{4} & 0 & \\ \frac{L-m}{4} & \frac{L+m}{2} & \frac{L-m}{4} & \ddots \\ 0 & \frac{L-m}{4} & \frac{L+m}{2} & \ddots \\ & \ddots & \ddots & \ddots \end{bmatrix}$$

Lower bound (any algorithm):

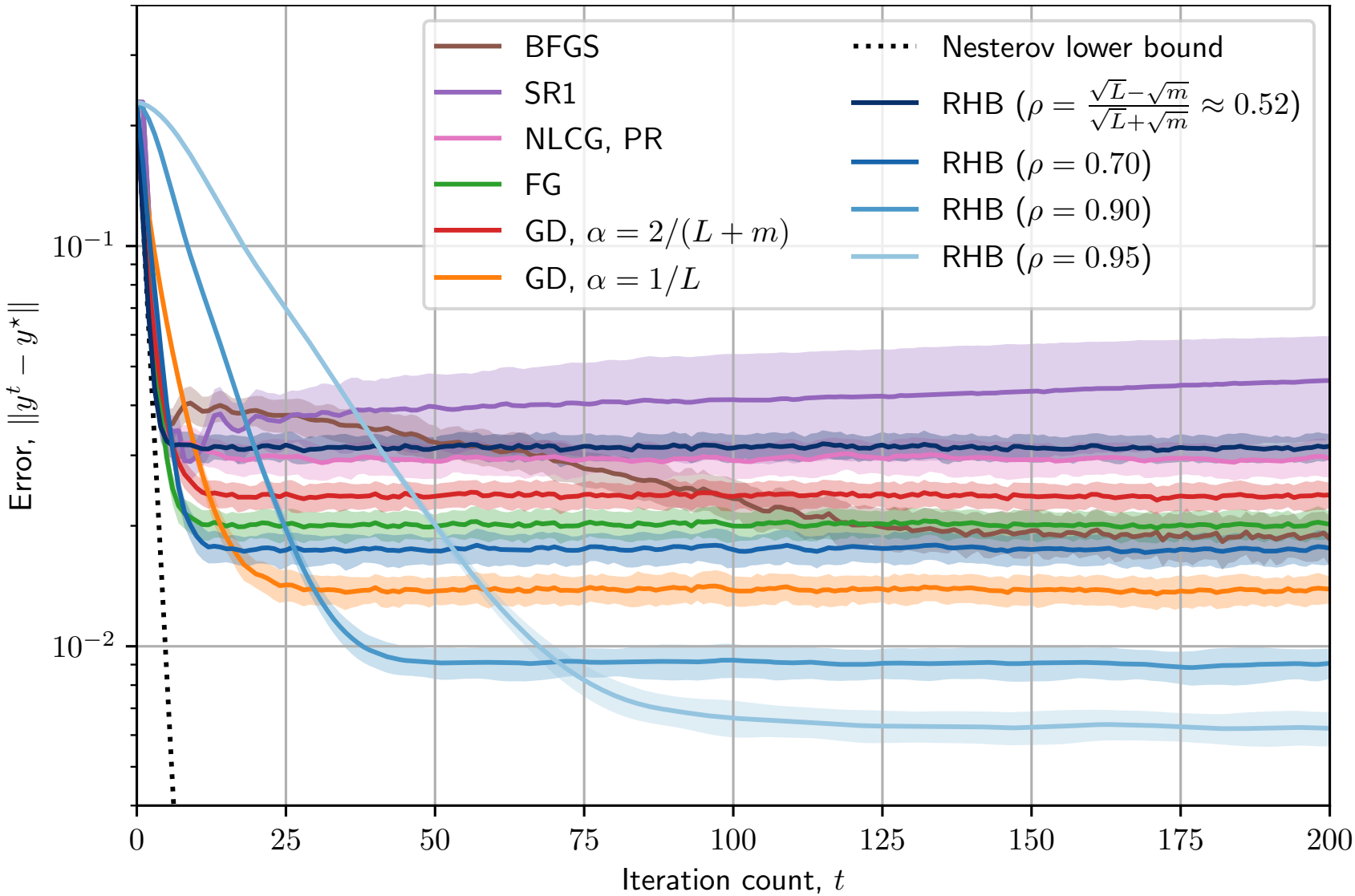
$$\|x_k - x^\star\| \geq \left(\frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}}\right)^k \|x_0 - x^\star\|$$

- Quasi-Newton methods (BFGS, SR1)
- Nonlinear conjugate gradient
- Fast Gradient, Heavy Ball, Gradient descent

Nesterov worst-case, $d = 100$, $m = 1$, $L = 10$, $\sigma = 10^{-5}$.



Nesterov worst-case, $d = 100$, $m = 1$, $L = 10$, $\sigma = 10^{-2}$.



Thank you!

- Preprint available:
  https://arxiv.org/abs/2109.05059

- Funding acknowledgement:
  NSF 1750162, 1936648

Backup Slides

Nesterov worst-case, $d = 100$, $m = 1$, $L = 10$, $\sigma = 10^{-5}$.
Using piecewise constant ρ schedule.