automating the analysis and design of large-scale optimization algorithms

Laurent Lessard

University of Wisconsin–Madison

Joint work with Ben Recht, Andy Packard,
Bin Hu, Bryan Van Scoy, Saman Cyrus

October, 2018

Unconstrained optimization:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathbb{R}^N \end{array}$$

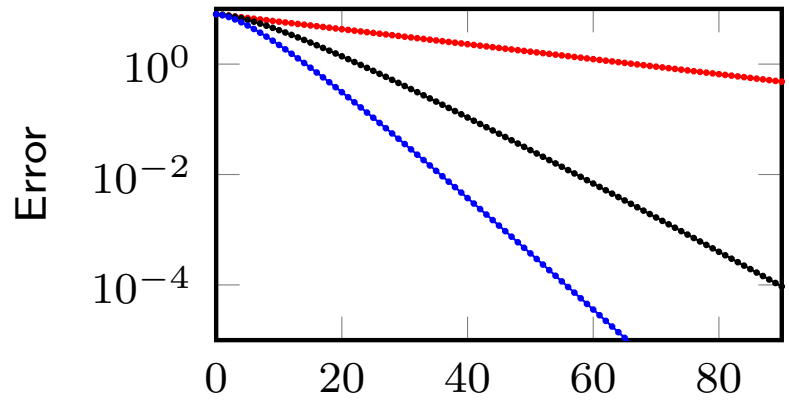
- need algorithms that are *fast* and *simple*
- currently favored family: *first-order methods*

Gradient method

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

Heavy ball method

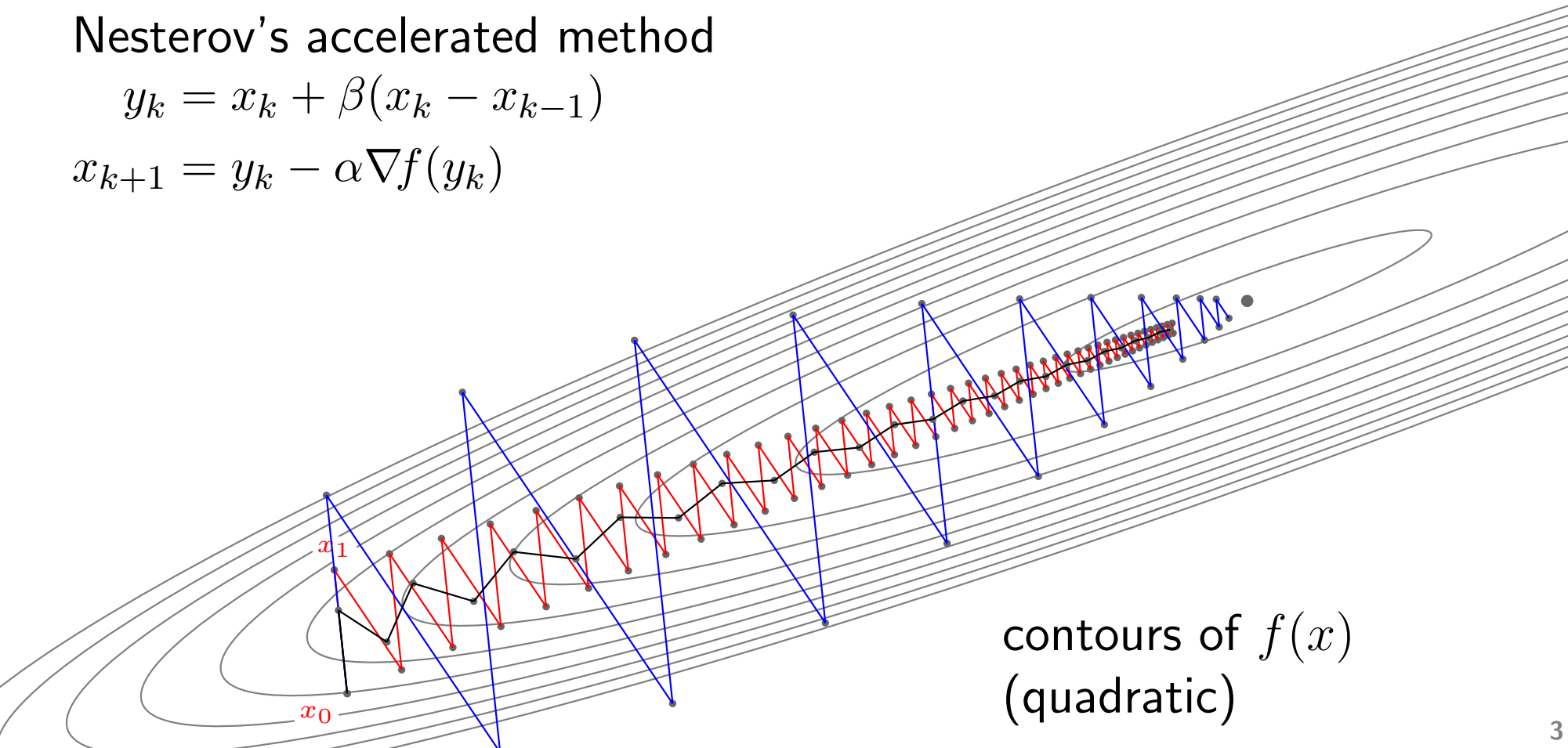
$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$$



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)

1. Many algorithms can be viewed as dynamical systems with feedback (control systems!).

$$\text{algorithm convergence} \iff \text{system stability}$$

2. By solving a small convex program, we can recover state-of-the-art convergence results for these algorithms, automatically and efficiently.

3. The ultimate goal: to move from analysis to design.

Robust algorithm selection

$G \in \mathcal{G}$: algorithm to use

$f \in \mathcal{S}$: function to minimize

$$G_{\text{opt}} = \arg\min_{G \in \mathcal{G}} \left(\max_{f \in \mathcal{S}} \text{cost}(f, G) \right)$$

Similar problem for a finite number of iterations:

- Drori, Teboulle (2012)
- Taylor, Hendrickx, Glineur (2016)

$G \in \mathcal{G}$ {

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

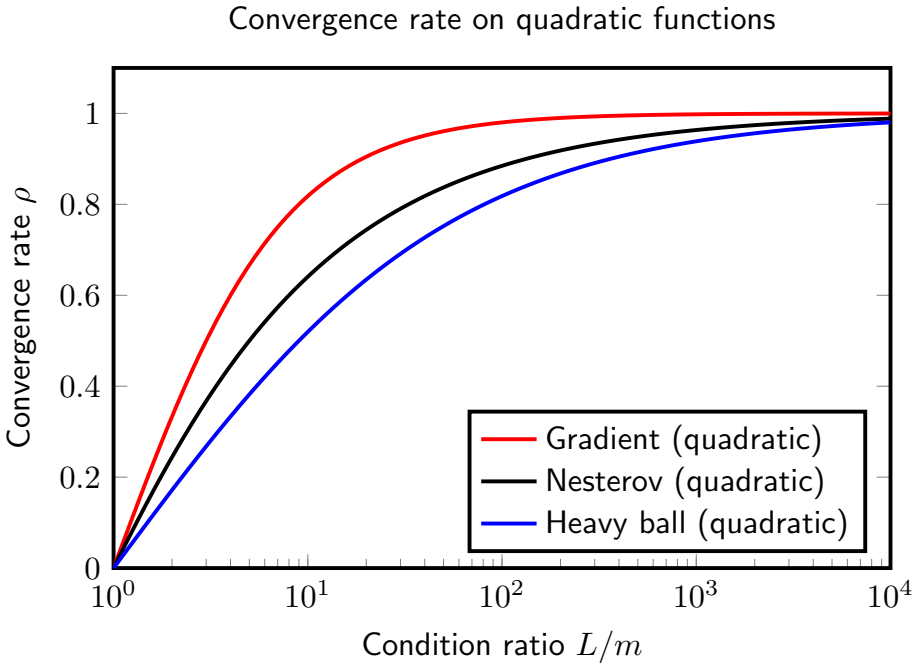
$$x_{k+1} = x_k - \alpha \nabla f\big(x_k + \beta(x_k - x_{k-1})\big) + \beta(x_k - x_{k-1})$$

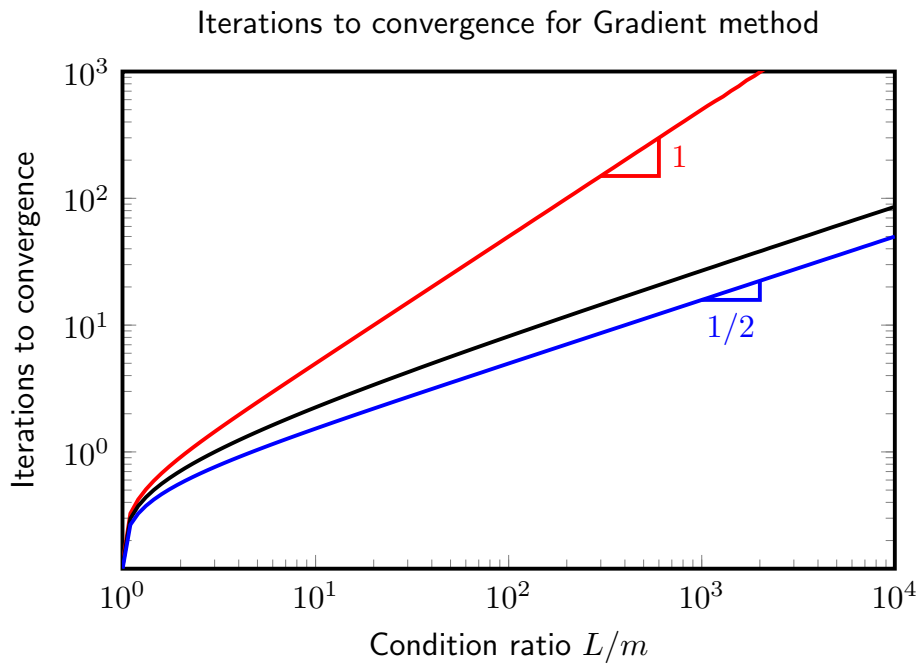
$f \in \mathcal{S}$ {

Analytically solvable:

Quadratic functions: $f(x) = \frac{1}{2} x^{\mathsf{T}} Q x - p^{\mathsf{T}} x$

with the constraint: $m \leq \lambda(Q) \leq L$





Convergence rate : $\|x_k - x_\star\| \leq C\rho^k \|x_0 - x_\star\|$

Iterations to convergence $\propto -\dfrac{1}{\log \rho}$

Robust algorithm selection

$G \in \mathcal{G}$: algorithm to use

$f \in \mathcal{S}$: function to minimize

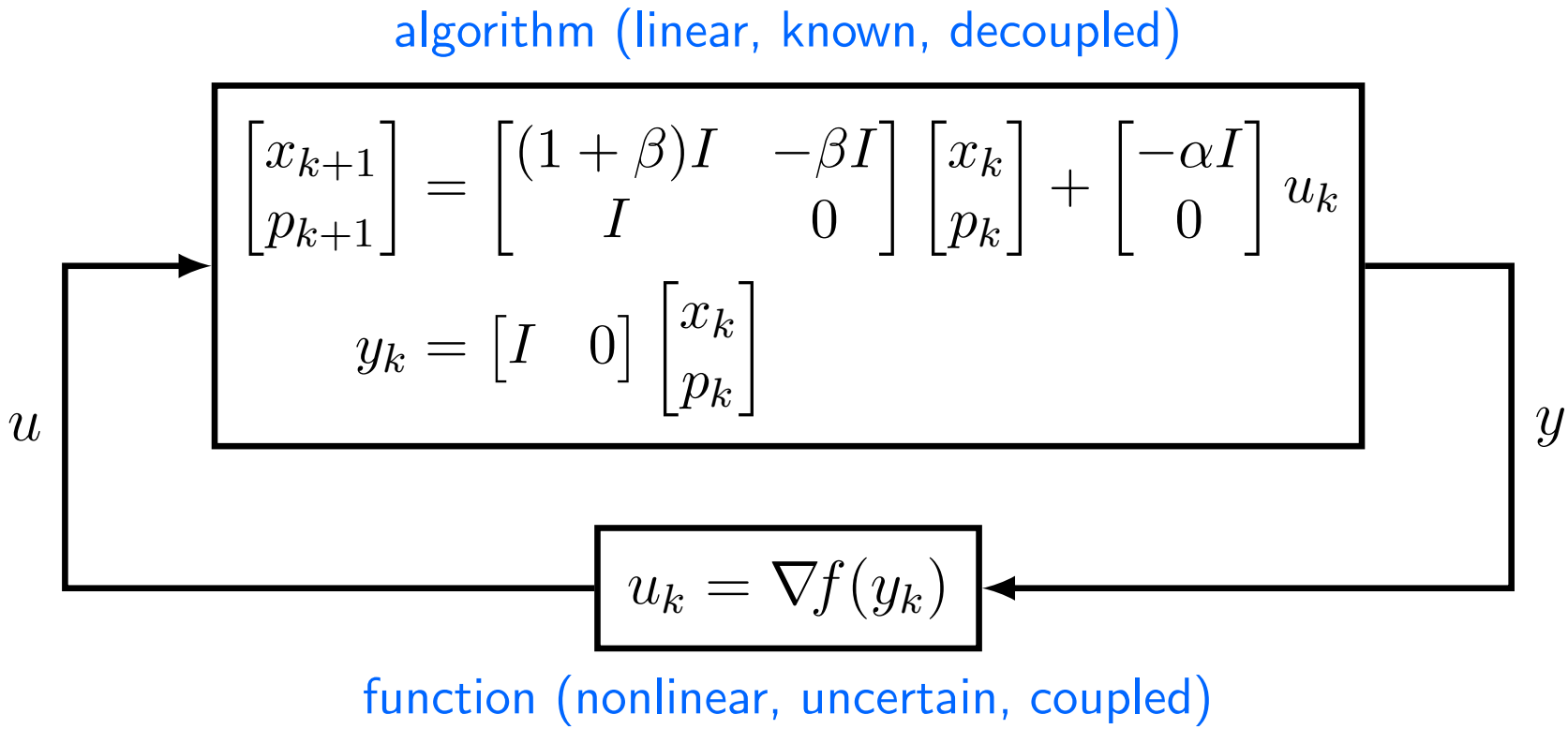
$$G_{\text{opt}} = \arg\min_{G \in \mathcal{G}} \left(\max_{f \in \mathcal{S}} \text{cost}(f, G) \right)$$

1. mathematical representation for $\mathcal{G}$
2. mathematical representation for $\mathcal{S}$
3. main robustness result

Dynamical system interpretation

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

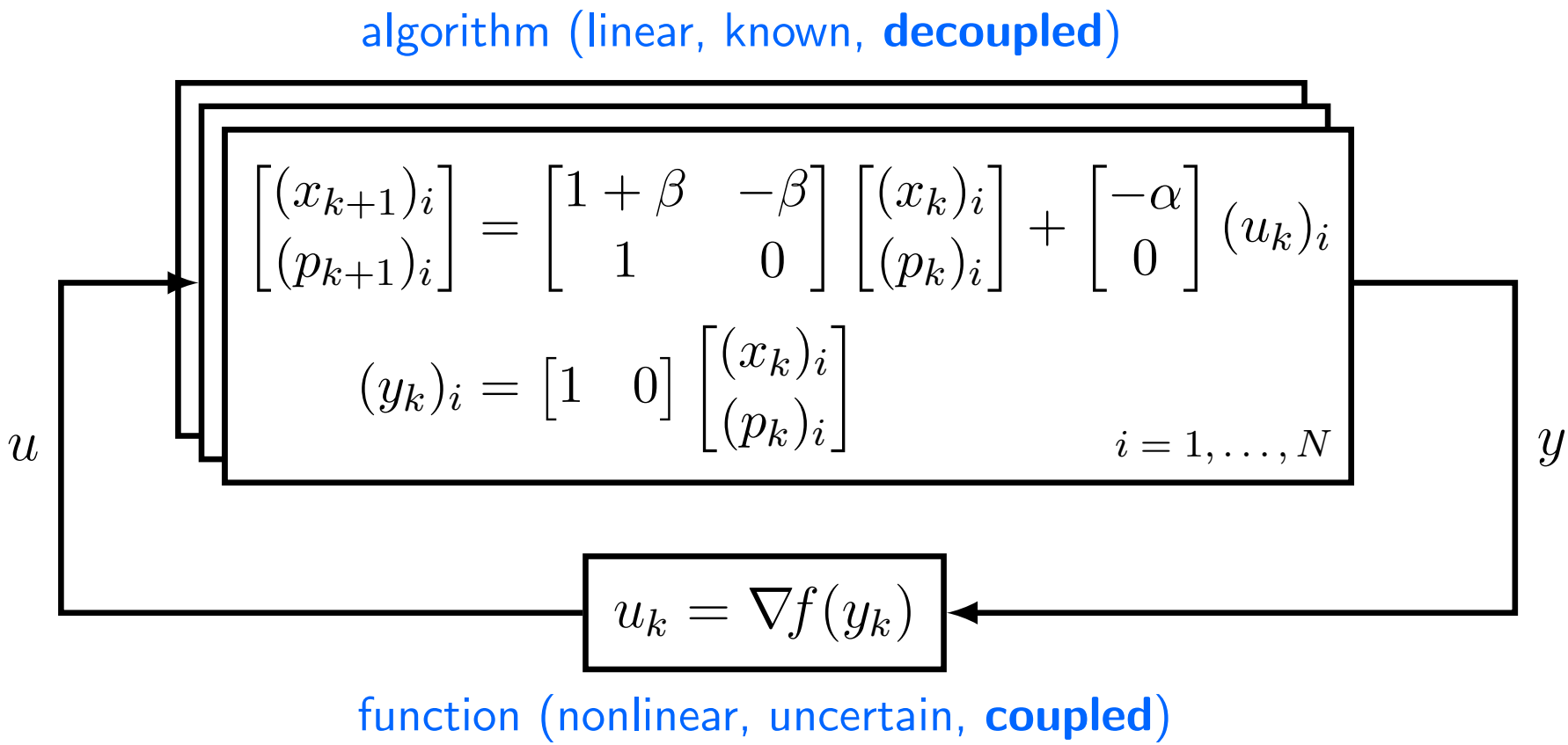
Define $u_k := \nabla f(x_k)$ and $p_k := x_{k-1}$

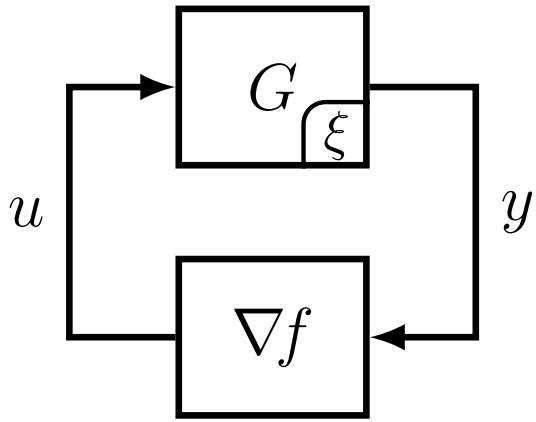


Dynamical system interpretation

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

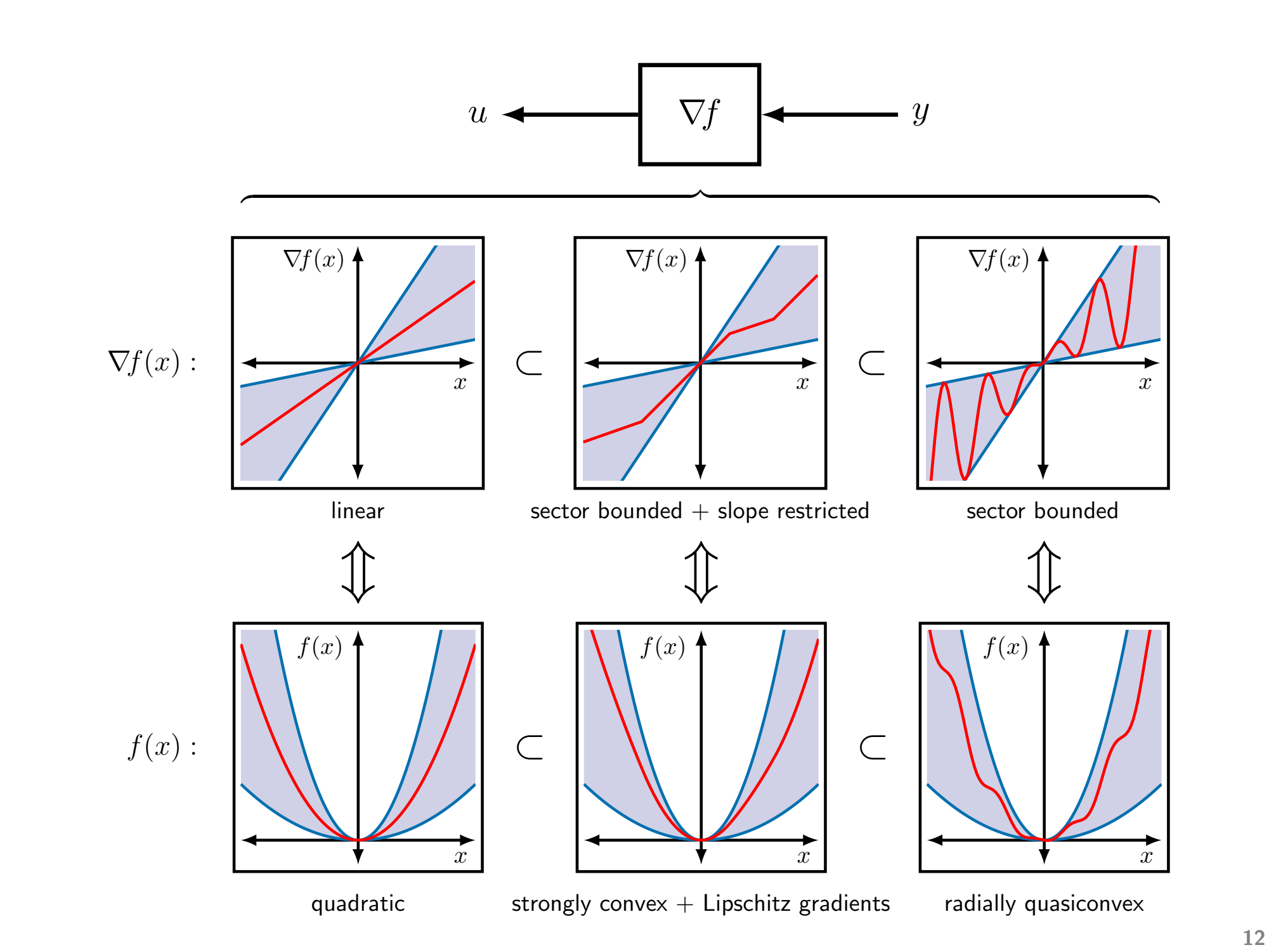
Define $u_k := \nabla f(x_k)$ and $p_k := x_{k-1}$

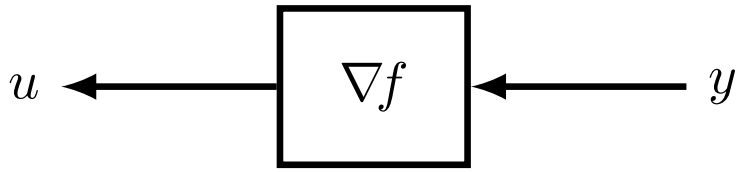




$$\xi_{k+1} = A\xi_k + Bu_k$$
$$y_k = C\xi_k$$
$$u_k = \nabla f(y_k)$$

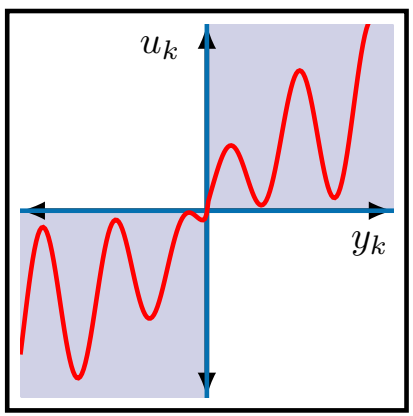
$$\left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right] = \begin{cases} \left[\begin{array}{c|c} 1 & -\alpha \\ \hline 1 & 0 \end{array}\right] & \text{Gradient} \\ \left[\begin{array}{cc|c} 1+\beta & -\beta & -\alpha \\ 1 & 0 & 0 \\ \hline 1 & 0 & 0 \end{array}\right] & \text{Heavy ball} \\ \left[\begin{array}{cc|c} 1+\beta & -\beta & -\alpha \\ 1 & 0 & 0 \\ \hline 1+\beta & -\beta & 0 \end{array}\right] & \text{Nesterov} \end{cases}$$


u
∇f
y
$\nabla f(x)$:
$\nabla f(x)$
x
$\subset$
linear
sector bounded + slope restricted
sector bounded
$f(x)$:
$f(x)$
quadratic
strongly convex + Lipschitz gradients
radially quasiconvex



Representing function classes

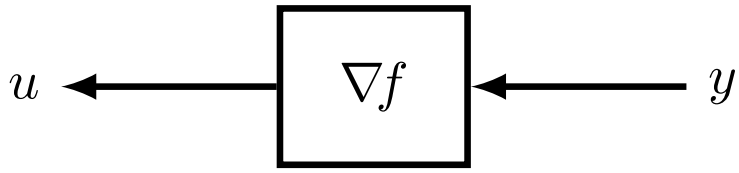
express as quadratic constraints on (y, u)



sector bounded

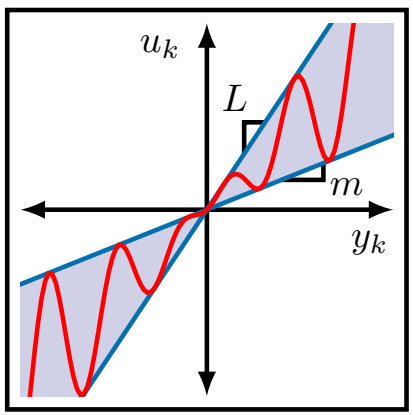
∇f is a **passive** function:

$$u_k y_k \geq 0$$



Representing function classes

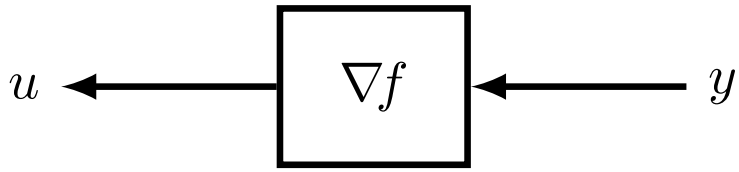
express as quadratic constraints on (y, u)



sector bounded

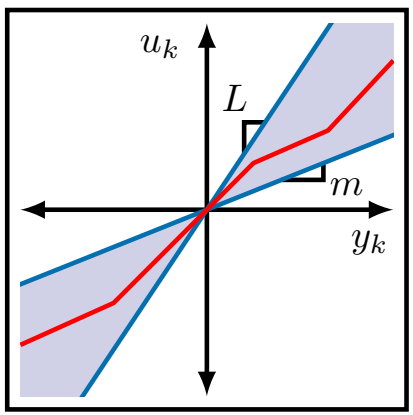
∇f is **sector-bounded**:

$$\begin{bmatrix} y_k \\ u_k \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} -2mL & m+L \\ m+L & -2 \end{bmatrix} \begin{bmatrix} y_k \\ u_k \end{bmatrix} \geq 0$$



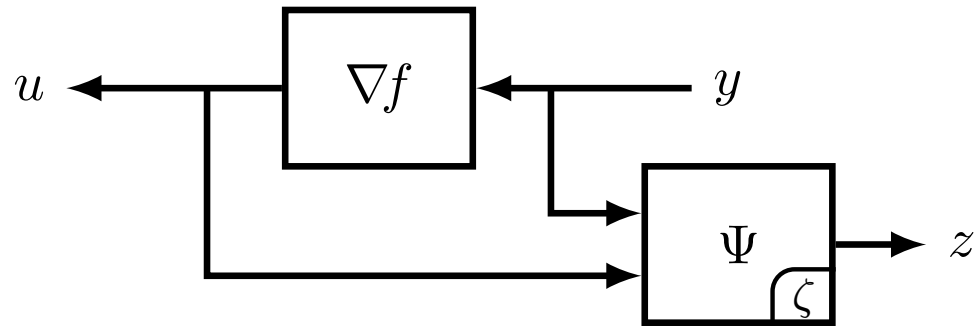
Representing function classes

express as quadratic constraints on (y, u)



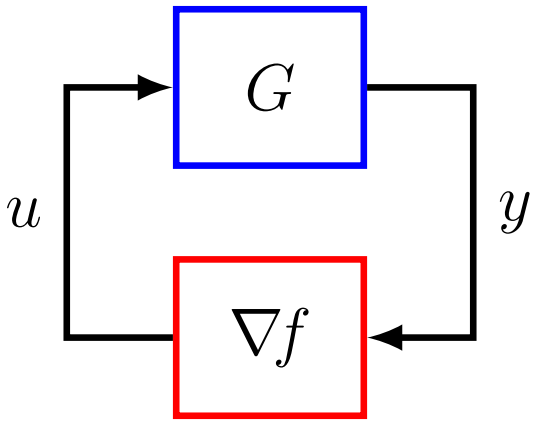
sector bounded + slope restricted

∇f is **sector-bounded + slope-restricted:** constraint on (y_k, u_k) depends on history $(y_0, \ldots, y_{k-1}, u_0, \ldots, u_{k-1})$.



Introduce extra dynamics

- Design dynamics Ψ and multiplier matrix M.
- Instead of using $q(u_k, y_k)$, use $z_k^\mathsf{T} M z_k$.
- Systematic way of doing this for strong convexity (Zames & Falb, 1968)
- General theory: Integral Quadratic Constraints (Megretski & Rantzer, 1997)

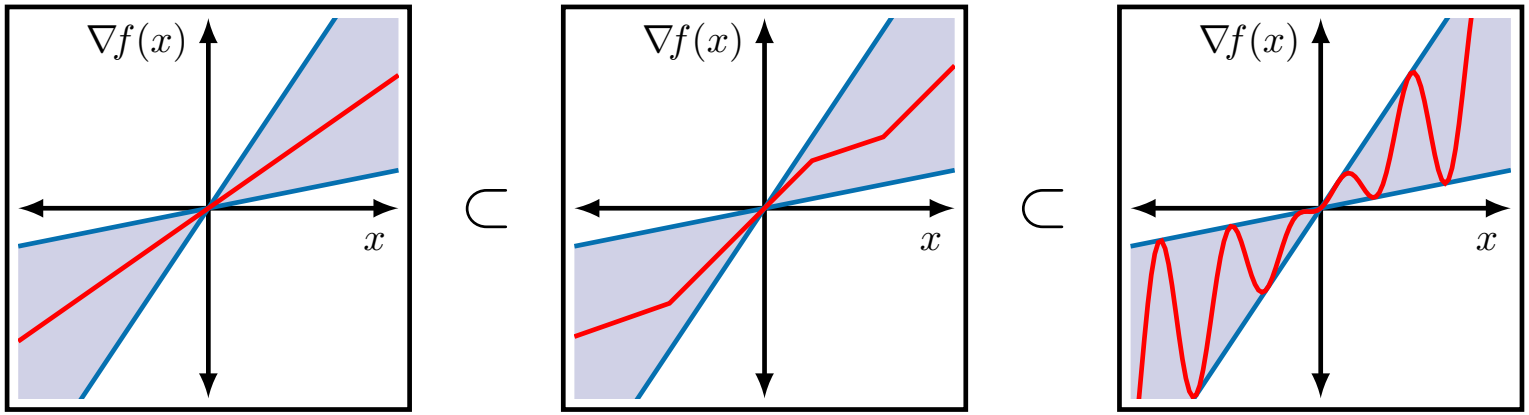


$$\left[\begin{array}{c|c} 1 & -\alpha \\ \hline 1 & 0 \end{array}\right] \quad \text{Gradient}$$

$$\left[\begin{array}{cc|c} 1+\beta & -\beta & -\alpha \\ 1 & 0 & 0 \\ \hline 1 & 0 & 0 \end{array}\right] \quad \text{Heavy ball}$$

$$\left[\begin{array}{cc|c} 1+\beta & -\beta & -\alpha \\ 1 & 0 & 0 \\ \hline 1+\beta & -\beta & 0 \end{array}\right] \quad \text{Nesterov}$$

$$\left\} \quad \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right]\right.$$



f is **quadratic** $\subset$ f is **strongly convex** $\subset$ f is **quasiconvex**

$$(\Psi, M)$$

Main result

Problem data:

- G (the algorithm)
- Ψ (what we know about f)

Size of LMI does **not** grow with problem dimension!
e.g. $P \in \mathbf{S}^{3\times 3}$, LMI $\in \mathbf{S}^{4\times 4}$

Auxiliary quantities:

- Compute $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ matrices from (G, Ψ)
- Choose a candidate rate $0 < \rho < 1$.

If there exists $P \succ 0$ such that

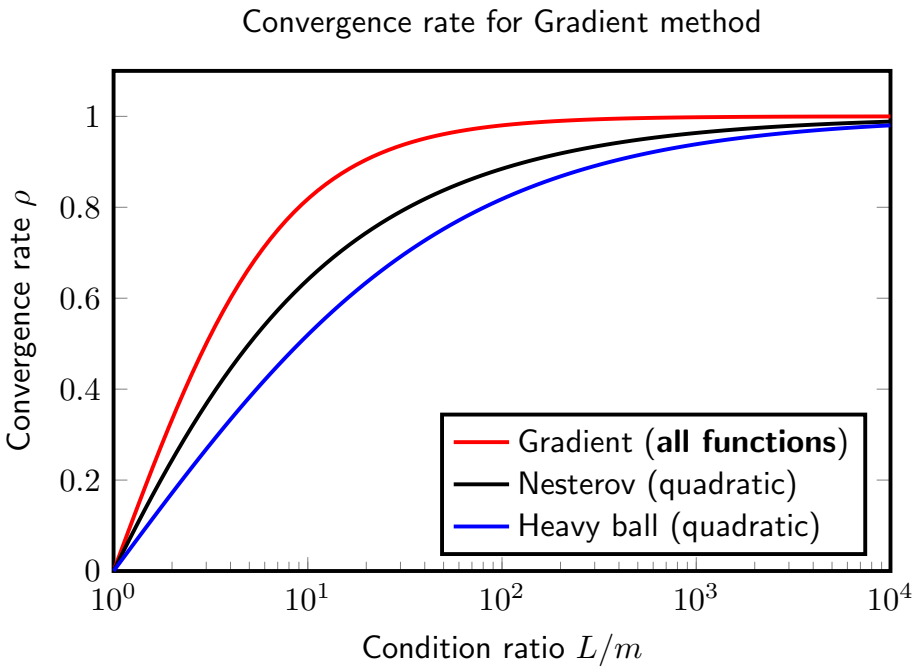
$$\begin{bmatrix} \hat{A}^{\mathsf{T}} P \hat{A} - \rho^2 P & \hat{A}^{\mathsf{T}} P \hat{B} \\ \hat{B}^{\mathsf{T}} P \hat{A} & \hat{B}^{\mathsf{T}} P \hat{B} \end{bmatrix} + \begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix}^{\mathsf{T}} M \begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix} \preceq 0$$

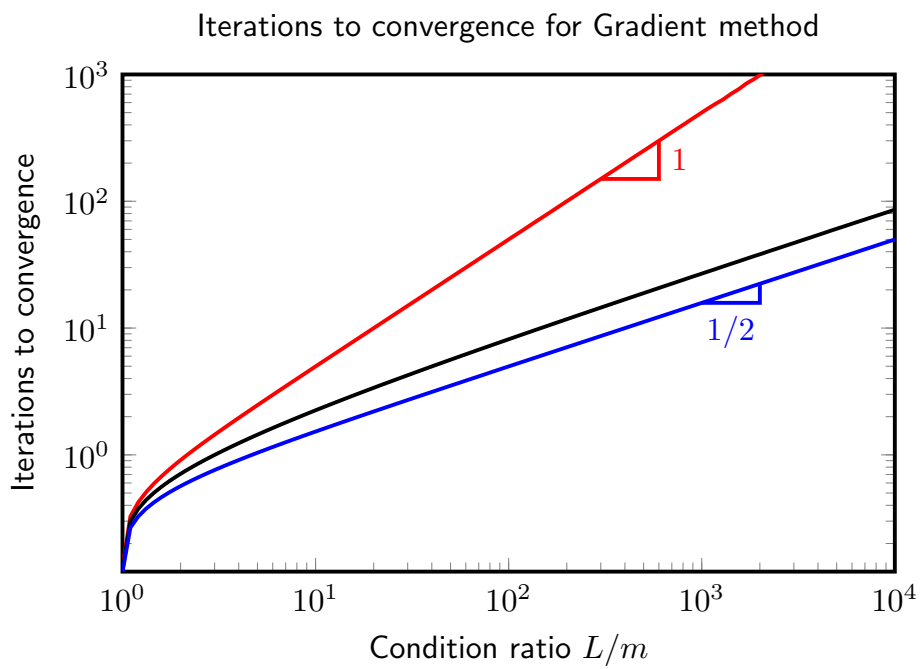
then $\|x_k - x_\star\| \leq \sqrt{\mathrm{cond}(P)}\, \rho^k \, \|x_0 - x_\star\|$ for all k.

main results:
analytic and numerical

Gradient method

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

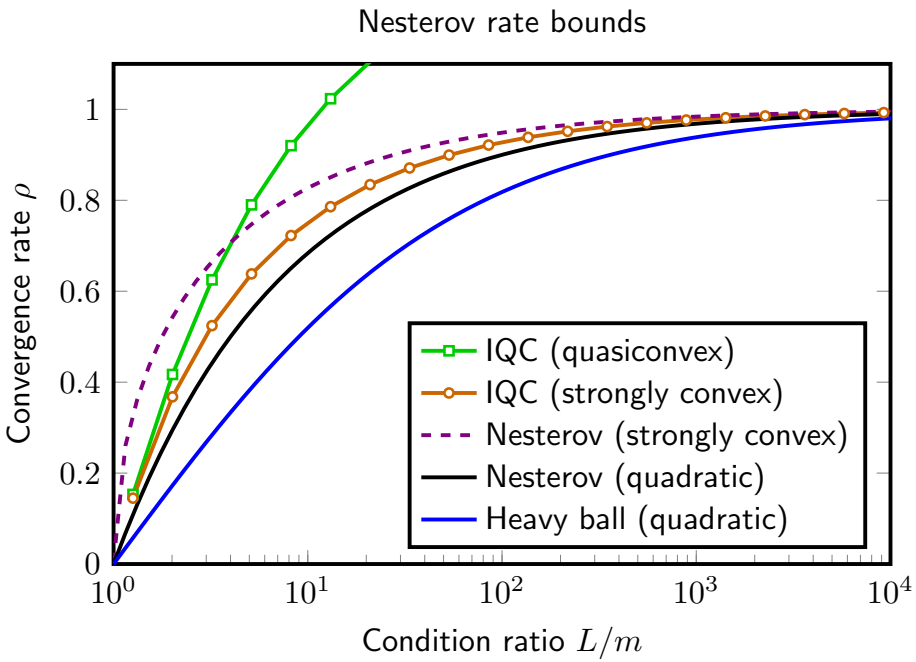


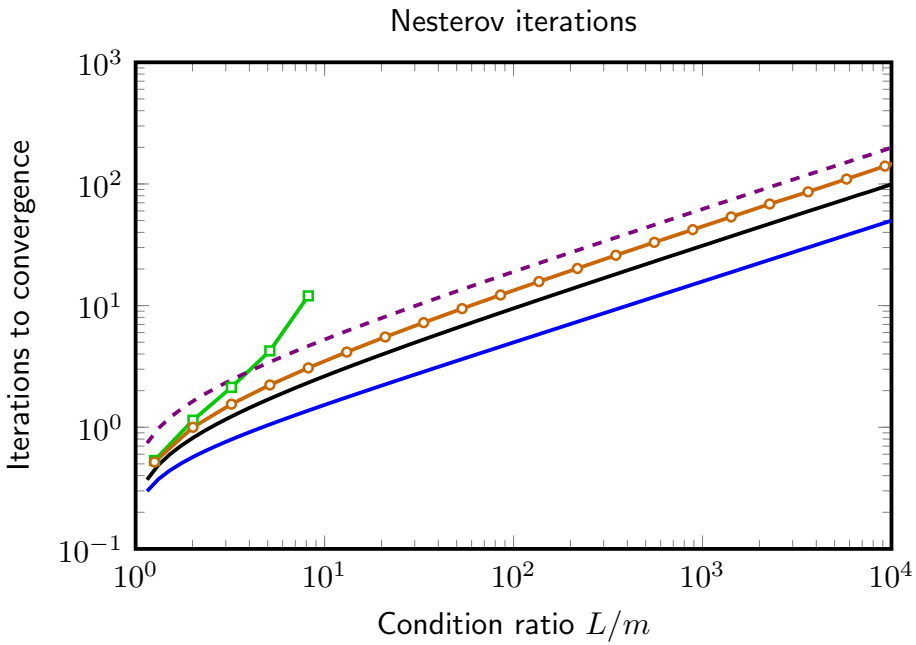


analytic solution! Same rate for: quadratics, strongly convex, or quasiconvex functions.

Nesterov's method

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

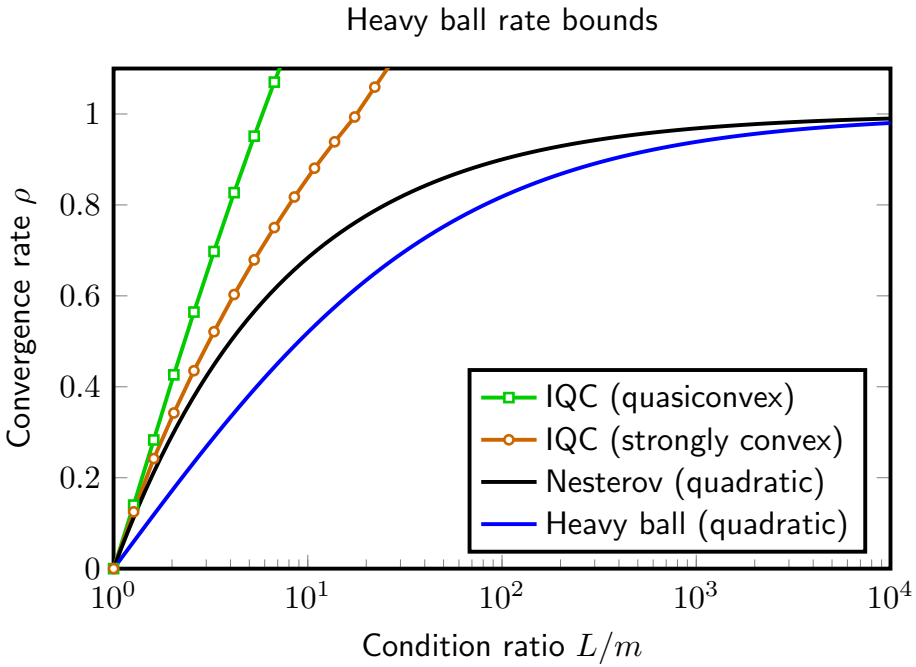


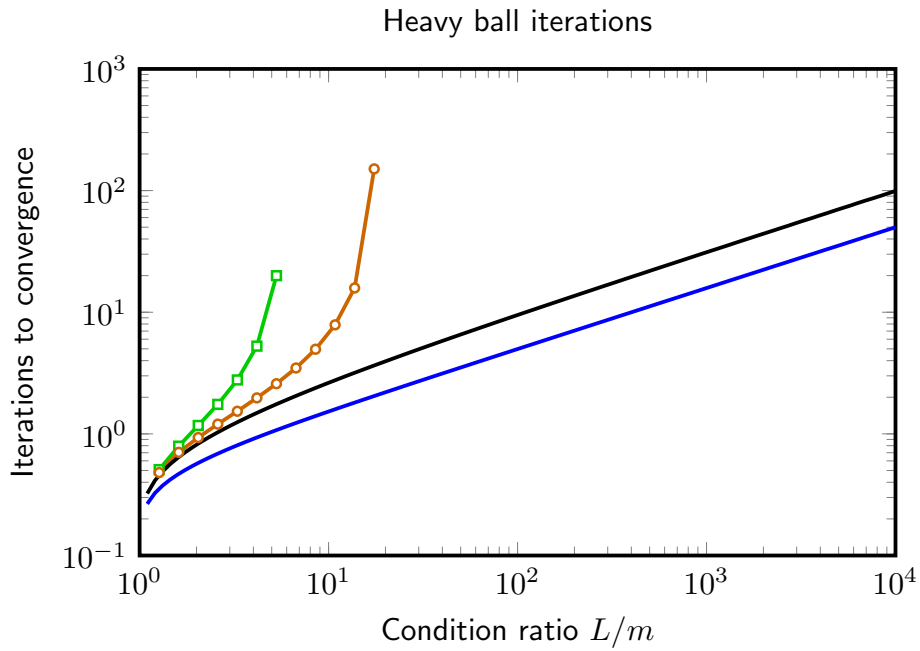


- Cannot certify stability for quasiconvex functions
- IQC bound **improves** upon best known bound!

Heavy ball method

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



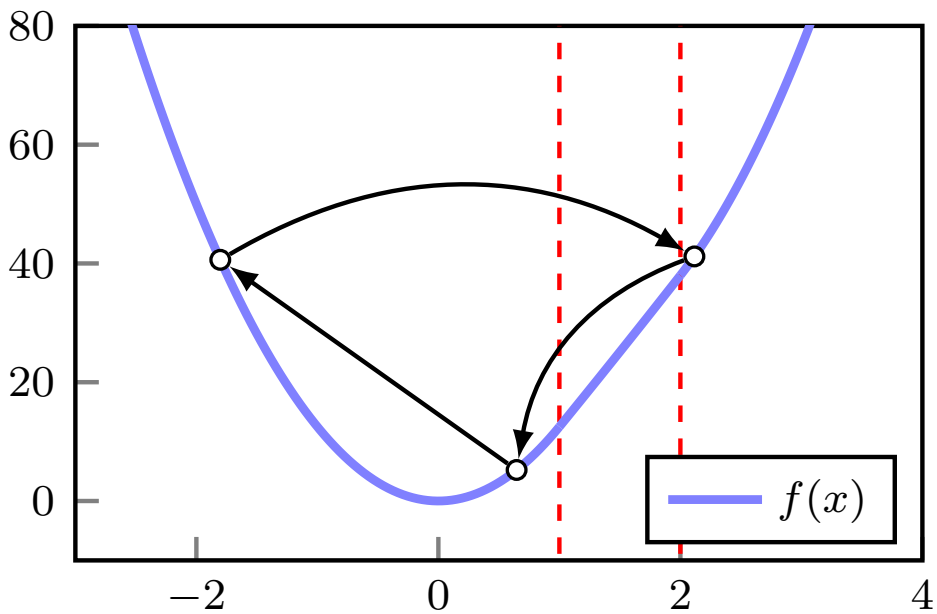


- Cannot certify stability for quasiconvex functions
- Cannot certify stability for strongly convex functions

The heavy ball method is not stable!

counterexample: $f(x) = \begin{cases} \frac{25}{2}x^2 & x < 1 \\ \frac{1}{2}x^2 + 24x - 12 & 1 \le x < 2 \\ \frac{25}{2}x^2 - 24x + 36 & x \ge 2 \end{cases}$

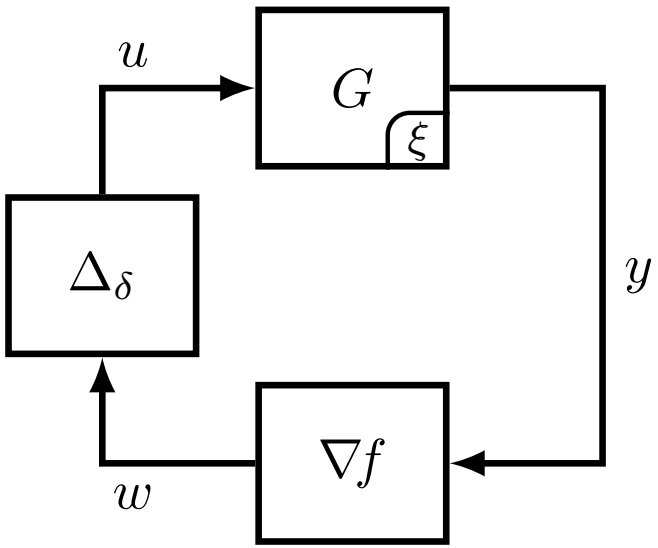
and start the heavy ball iteration at $x_0 = x_1 \in [3.07, 3.46]$.



- $L/m = 25$
- heavy ball iterations converge to a limit cycle
- simple counterexample to the Aizerman (1949) and Kalman (1957) conjectures

uncharted territory:
noise robustness and algorithm design

Noise robustness

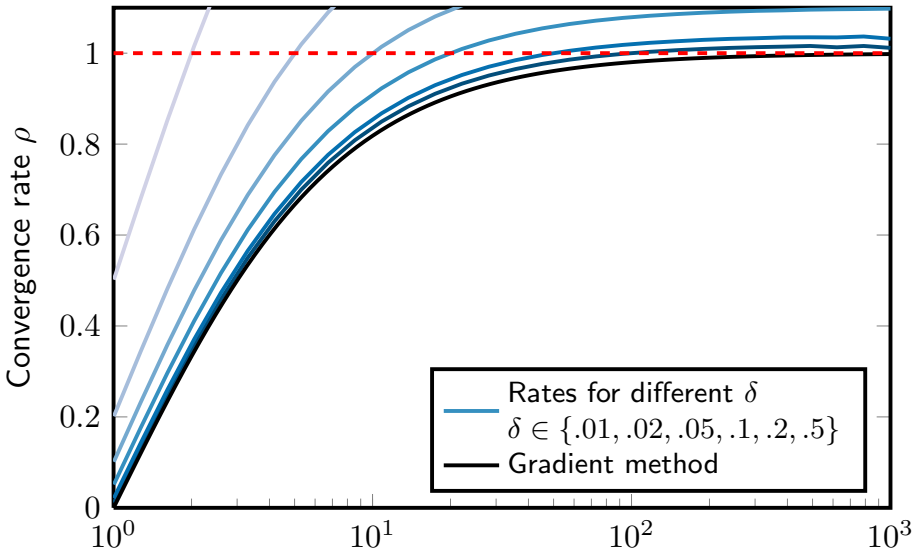


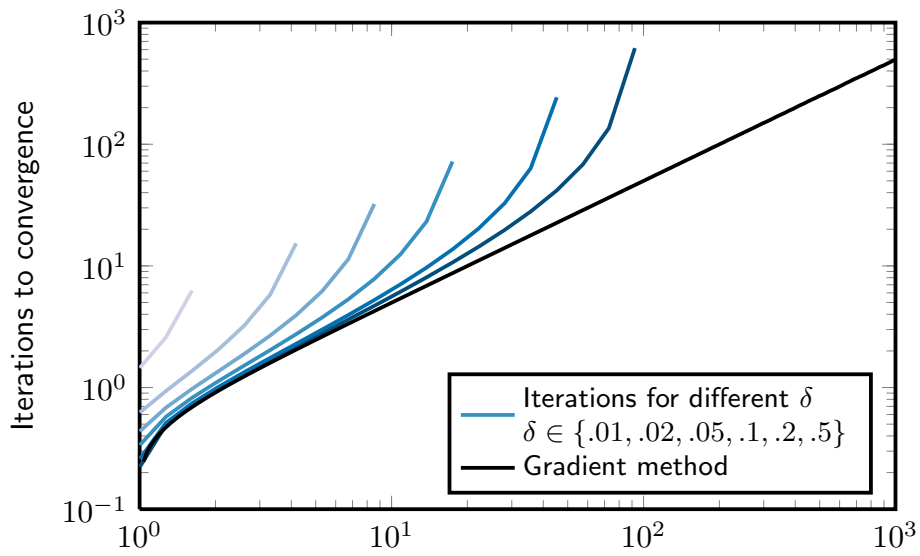
The Δ_δ block is uncertain multiplicative noise:

$$\|u_k - w_k\| \leq \delta \|w_k\|$$

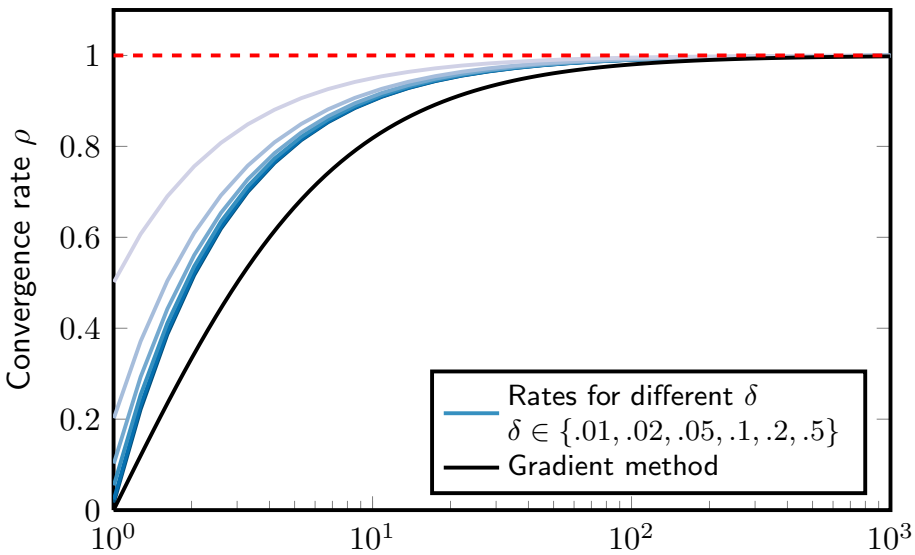
How does an algorithm perform in the presence of noise?

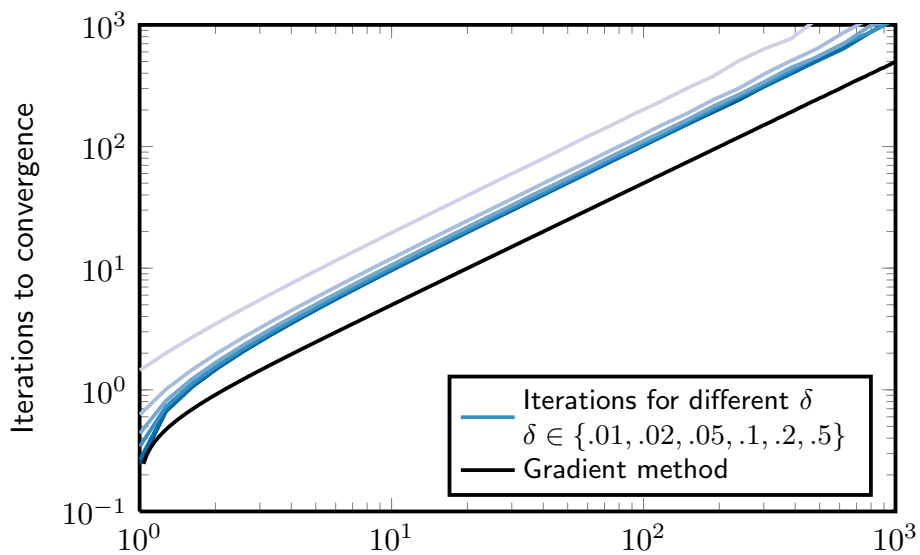
Gradient method, $\alpha = \frac{2}{L+m}$ (optimal stepsize with no noise)



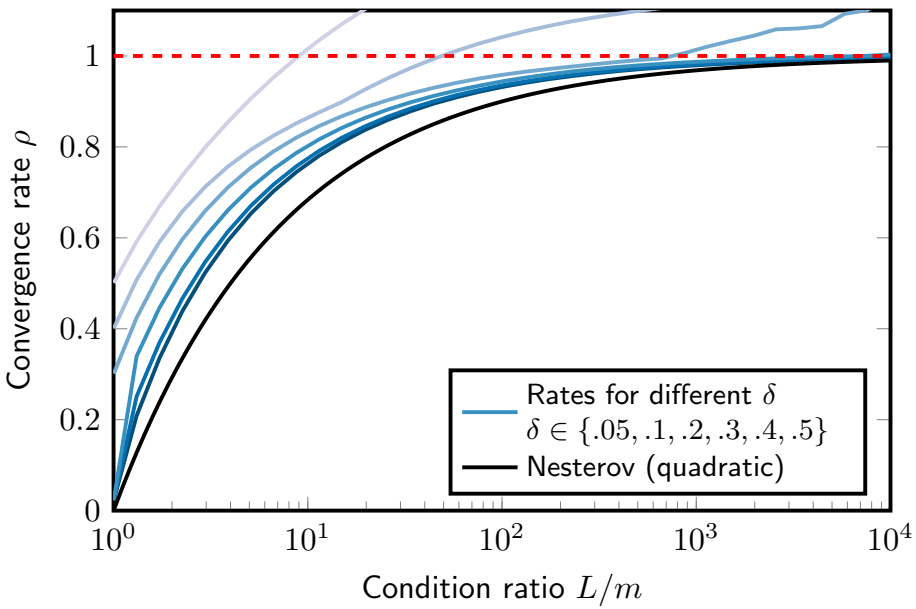


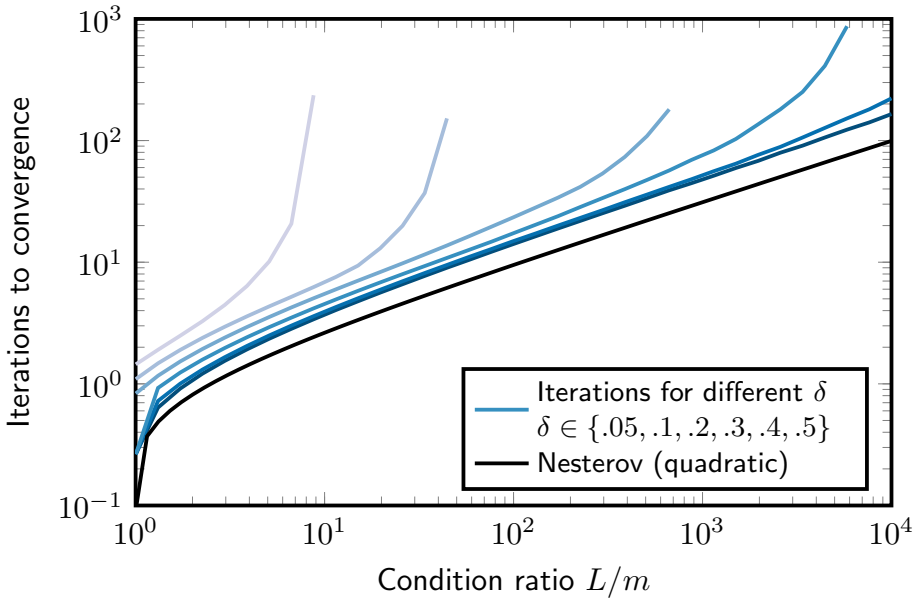
Gradient method, $\alpha = \frac{1}{L}$ (more conservative stepsize)





Nesterov's method (strongly convex f, with noise)





- Nesterov's method is not robust to noise.

can we have it all? (robustness AND performance)

Design approach

- parameterize all proper G of degree 2
- parameterization in terms of (α, β, η):

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

Special cases:

$$(\alpha, \beta, \eta) = \begin{cases} (\alpha, 0, 0) & \text{Gradient} \\ (\alpha, \beta, 0) & \text{Heavy ball} \\ (\alpha, \beta, \beta) & \text{Nesterov} \end{cases}$$

Robust Momentum Method (ACC'18)

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

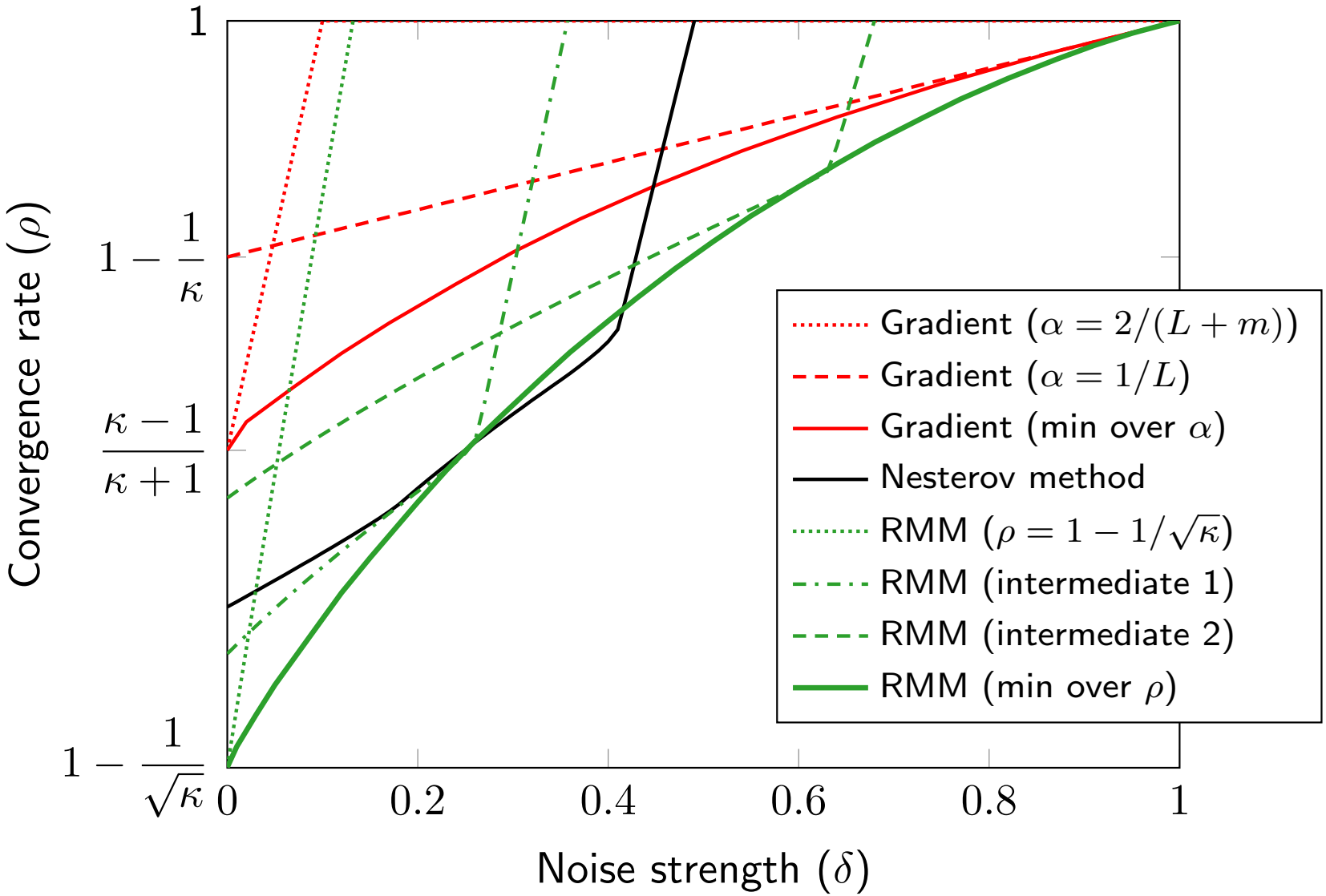
Parameters designed via root locus technique:

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

tuning parameter: $\underbrace{1 - \frac{1}{\sqrt{\kappa}}}_{\text{fast + fragile}} \leq \rho \leq \underbrace{1 - \frac{1}{\kappa}}_{\text{slow + robust}}$

- When $\rho = 1 - \frac{1}{\kappa}$, recover Gradient with $\alpha = \frac{1}{L}$
- When $\rho = 1 - \frac{1}{\sqrt{\kappa}}$, recover Triple Momentum Method with optimal tuning (Van Scoy et al, 2018)

Trade-off: performance vs robustness



Related works

In this talk

- Unified analysis framework (SIOPT'16)
- Robust momentum method (ACC'18)

Other families of algorithms

- Operator-splitting methods, e.g. ADMM (ICML'15)
- Weakly convex functions ($1/k$ and $1/k^2$ rates) (ICML'17)
- Distributed optimization algorithms (ALLER'17)
- Stochastic variance reduction, e.g. SVRG (ICML'18)

Thank you!

- Manuscripts + code available:
  https://laurentlessard.com