automating the analysis and design of large-scale optimization algorithms

Laurent Lessard

University of Wisconsin–Madison

with Ben Recht and Andy Packard, UC Berkeley

ICCOPT 2016 – Tokyo, Japan
August 8, 2016

In this talk: a general framework for obtaining performance guarantees for first-order optimization algorithms.

- uses a dynamical systems perspective and tools from robust control theory and semidefinite programming.
- universal: the same method can analyze a variety of algorithms under various assumptions.
- efficient: requires solving a very small LMI.
- it works! recovers or improves on existing results and provides new insights.

Gradient method

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

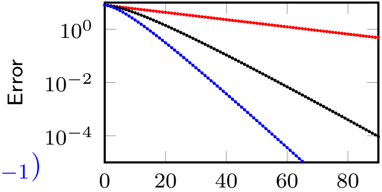
Heavy ball method

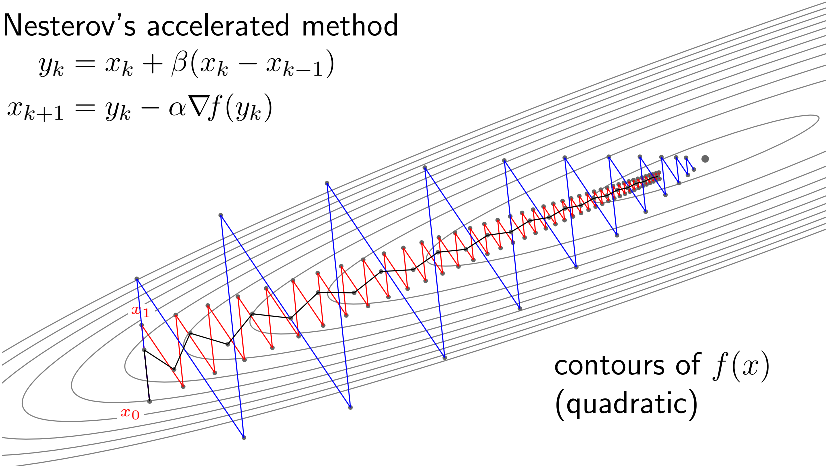
$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$$

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)

Robust algorithm selection

$f \in \mathcal{S}$: function we'd like to minimize

$G \in \mathcal{G}$: algorithm we're going to use

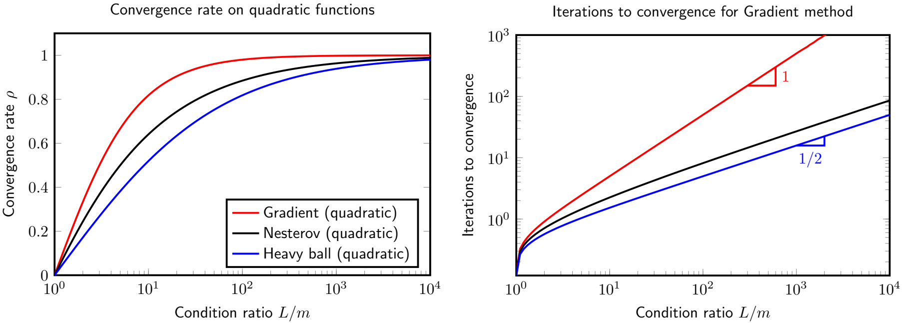
$$G_{\text{opt}} = \arg\min_{G \in \mathcal{G}} \left(\max_{f \in \mathcal{S}} \operatorname{cost}(f, G) \right)$$

Similar problem for a finite number of iterations:

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

$$G \in \mathcal{G} \begin{cases} \text{Gradient method} \\ \quad x_{k+1} = x_k - \alpha \nabla f(x_k) \\ \\ \text{Heavy ball method} \\ \quad x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}) \\ \\ \text{Nesterov's accelerated method} \\ \quad x_{k+1} = x_k - \alpha \nabla f\big(x_k + \beta(x_k - x_{k-1})\big) + \beta(x_k - x_{k-1}) \end{cases}$$

$$f \in \mathcal{S} \begin{cases} \textbf{Analytically solvable:} \\ \text{Quadratic functions: } f(x) = x^\mathsf{T} Q x - p^\mathsf{T} x \\ \text{with the constraint: } mI \preceq Q \preceq LI \end{cases}$$



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

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

Robust algorithm selection

$f \in \mathcal{S}$: function we'd like to minimize

$G \in \mathcal{G}$: algorithm we're going to use

$$G_{\mathsf{opt}} = \operatorname*{arg\,min}_{G \in \mathcal{G}} \left(\max_{f \in \mathcal{S}} \operatorname{cost}(f, G) \right)$$

(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 $p_k := x_{k-1}$

algorithm (linear, known, decoupled)

$$\begin{bmatrix} x_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} (1+\beta)I & -\beta I \\ I & 0 \end{bmatrix} \begin{bmatrix} x_k \\ p_k \end{bmatrix} + \begin{bmatrix} -\alpha I \\ 0 \end{bmatrix} u_k$$

$$y_k = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} x_k \\ p_k \end{bmatrix}$$

u

y

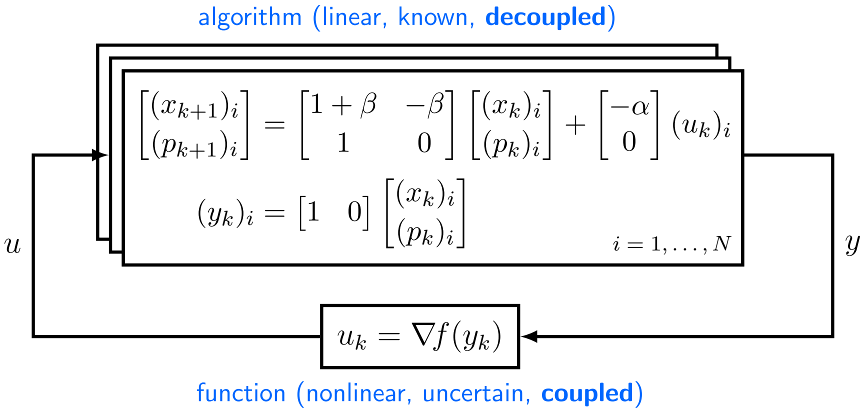
$$u_k = \nabla f(y_k)$$

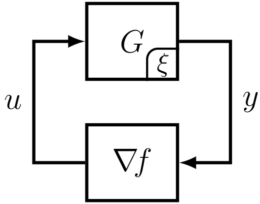
function (nonlinear, uncertain, coupled)

Dynamical system interpretation

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

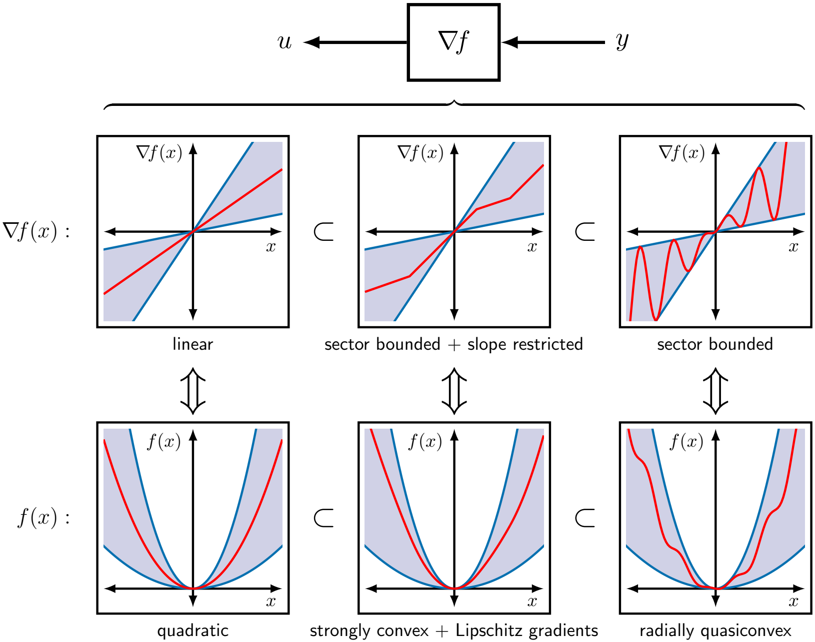
Define $u_k := \nabla f(x_k)$ and $p_k := x_{k-1}$



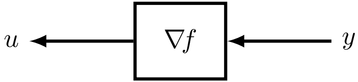


$$
\begin{aligned}
\xi_{k+1} &= A\xi_k + Bu_k \\
y_k &= C\xi_k \\
u_k &= \nabla f(y_k)
\end{aligned}
$$

$$
\left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right] =
\begin{cases}
\left[\begin{array}{c|c} 1 & -\alpha \\ \hline 1 & 0 \end{array}\right] & \text{Gradient} \\[2ex]
\left[\begin{array}{cc|c} 1+\beta & -\beta & -\alpha \\ 1 & 0 & 0 \\ \hline 1 & 0 & 0 \end{array}\right] & \text{Heavy ball} \\[2ex]
\left[\begin{array}{cc|c} 1+\beta & -\beta & -\alpha \\ 1 & 0 & 0 \\ \hline 1+\beta & -\beta & 0 \end{array}\right] & \text{Nesterov}
\end{cases}
$$

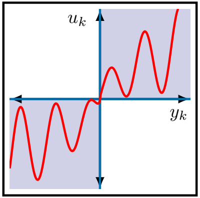


u
∇f
y
$\nabla f(x)$:
$f(x)$:
$\nabla f(x)$
x
linear
$\subset$
sector bounded + slope restricted
$\subset$
sector bounded
$\Updownarrow$
$f(x)$
quadratic
strongly convex + Lipschitz gradients
radially quasiconvex



Representing function classes

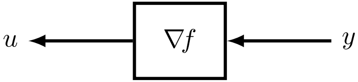
express as quadratic constraints on (y, u)



sector bounded

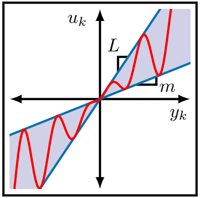
∇f is a **passive** function:

$$u_k y_k \geq 0$$



Representing function classes

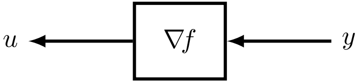
express as quadratic constraints on (y, u)



sector bounded

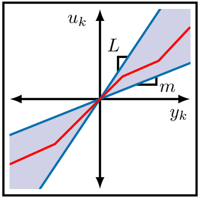
∇f is **sector-bounded**:

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



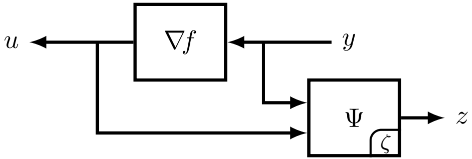
Representing function classes

express as quadratic constraints on (y, u)



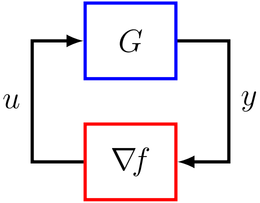
sector bounded + slope restricted

∇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 via Zames-Falb multipliers (1968).
- General theory: Integral Quadratic Constraints (Megretski & Rantzer 1997)

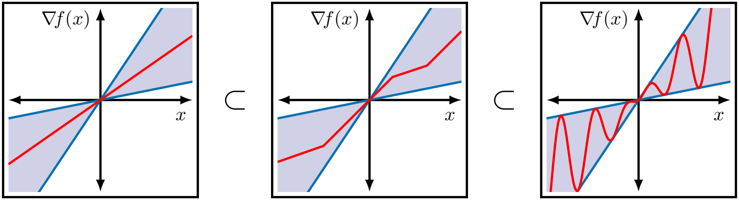


$$\left[\begin{array}{c|c} 1 & -\alpha \\ \hline 1 & 0 \end{array}\right] \quad \text{Gradient}$$

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

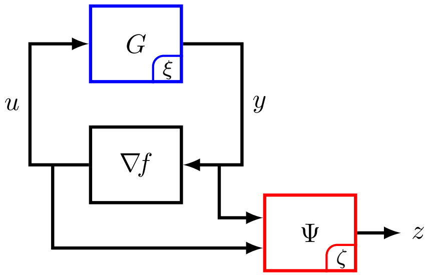
$$\left[\begin{array}{cc|c} 1+\beta & -\beta & -\alpha \\ 1 & 0 & 0 \\ \hline 1+\beta & -\beta & 0 \end{array}\right] \quad \text{Nesterov}$$

$$\left.\begin{array}{l} \text{Gradient} \\ \text{Heavy ball} \\ \text{Nesterov} \end{array}\right\} \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right]$$



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

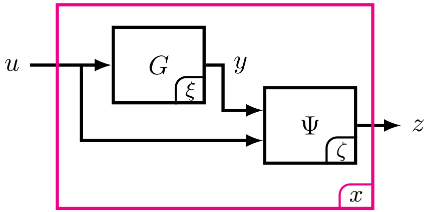
$$(\Psi, M)$$



$$\xi_{k+1} = A\xi_k + Bu_k$$
$$y_k = C\xi_k$$

$$u_k = \nabla f(y_k)$$

$$\zeta_{k+1} = A_\Psi \zeta_k + B_\Psi^y y_k + B_\Psi^u u_k$$
$$z_k = C_\Psi \zeta_k + D_\Psi^y y_k + D_\Psi^u u_k$$



$$x_{k+1} = \hat{A}x_k + \hat{B}u_k$$
$$z_k = \hat{C}x_k + \hat{D}u_k$$

where $x_k := \begin{bmatrix} \xi_k \\ \zeta_k \end{bmatrix}$ and z is quadratically constrained.

Main result

Suppose $\{x_0, x_1, \dots\}$ satisfies dynamics

$$x_{k+1} = \hat{A} x_k + \hat{B} u_k$$
$$z_k = \hat{C} x_k + \hat{D} u_k$$

where $\{z_0, z_1, \dots\}$ is constrained by

$$\sum_{k=0}^{T} \rho^{-2k} (z_k - z_\star)^\mathsf{T} M (z_k - z_\star) \geq 0 \quad \text{for all } T$$

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

If there exists $P \succ 0$ and $\lambda \geq 0$ such that

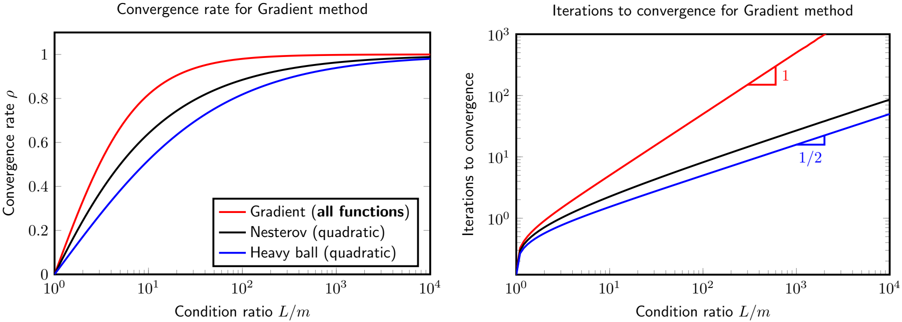
$$\begin{bmatrix} \hat{A}^\mathsf{T} P \hat{A} - \rho^2 P & \hat{A}^\mathsf{T} P \hat{B} \\ \hat{B}^\mathsf{T} P \hat{A} & \hat{B}^\mathsf{T} P \hat{B} \end{bmatrix} + \lambda \begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix}^\mathsf{T} M \begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix} \preceq 0$$

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

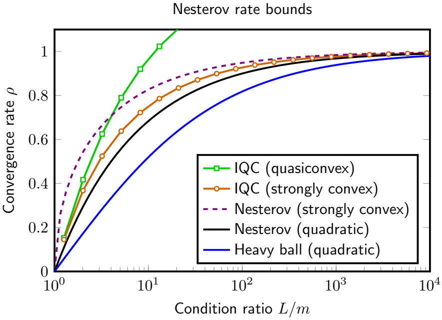
$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

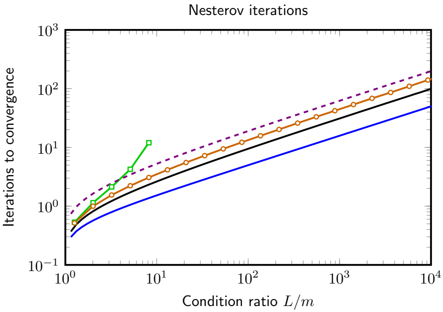


analytic solution! Same rate for: quadratics, strongly convex, or quasiconvex functions.

Nesterov's method

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

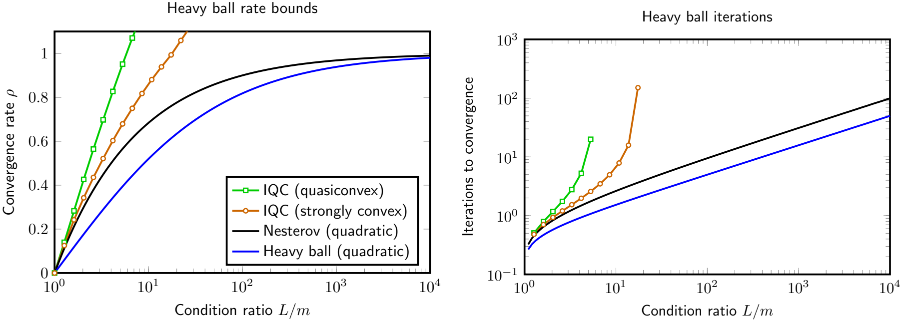




- Cannot certify stability for quasiconvex functions
- IQC bound **improves** upon best known bound!

Heavy ball method

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

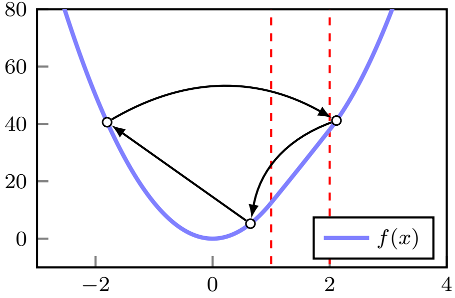


- Cannot certify stability for quasiconvex functions
- Cannot certify stability for strongly convex functions

The heavy ball method is not stable!

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

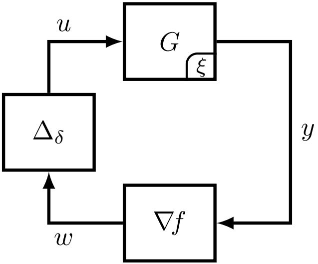
and start the heavy ball iteration at $x_0 = x_1 \in [3.07, 3.46]$.



- $L/m = 25$
- heavy ball iterations converge to a limit cycle

uncharted territory:
noise robustness and algorithm design

Noise robustness

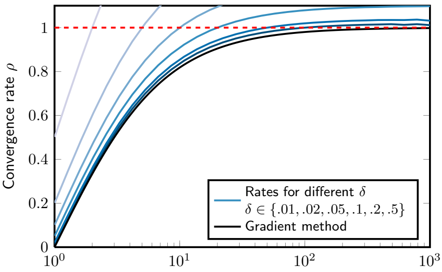


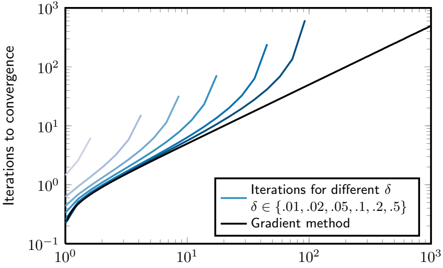
The Δ_δ block is uncertain multiplicative noise:

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

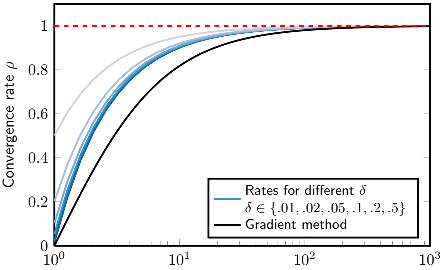
How does an algorithm perform in the presence of noise?

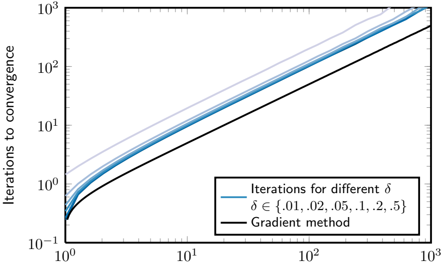
Gradient method, $\alpha = \frac{2}{L+m}$ (optimal stepsize with no noise)



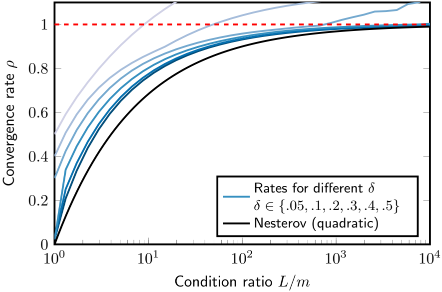


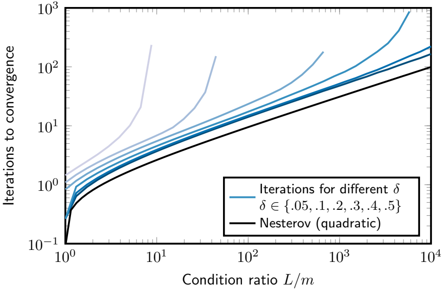
Gradient method, $\alpha = \frac{1}{L}$ (more conservative stepsize)





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





- Nesterov's method is not robust to noise.

can we have it all? (robustness AND performance)

Brute force approach

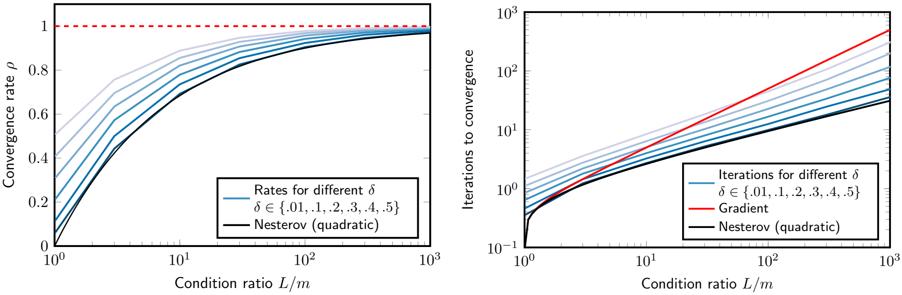
- test all strictly proper G of degree 2
- parameterization in terms of $(\alpha, \beta_1, \beta_2)$:

$$x_{k+1} = x_k - \alpha \nabla f\big(x_k + \beta_2(x_k - x_{k-1})\big) + \beta_1(x_k - x_{k-1})$$

Special cases:

$$(\alpha, \beta_1, \beta_2) = \begin{cases} (\alpha, 0, 0) & \text{Gradient} \\ (\alpha, \beta, 0) & \text{Heavy ball} \\ (\alpha, \beta, \beta) & \text{Nesterov} \end{cases}$$

Optimal designs over $(\alpha, \beta_1, \beta_2)$



- Faster than the gradient method **and** more robust to noise than Nesterov's method
- automatic algorithm design is possible!

What we have (so far!)

L, Recht, Packard (SIOPT'16)

- unified framework for algorithm analysis
- gradient, heavy ball, Nesterov
- constrained optimization, noise robustness

Nishihara, L, Recht, Packard, Jordan (ICML'15)

- application to robust ADMM tuning

Boczar, L, Recht (CDC'15)

- control theory version

Thank you!

- Manuscripts + code available:
  www.laurentlessard.com
- If you're interested, come talk to me!