

# 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

ACC 2019 workshop  
Philadelphia, July 9, 2019

## Gradient method

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

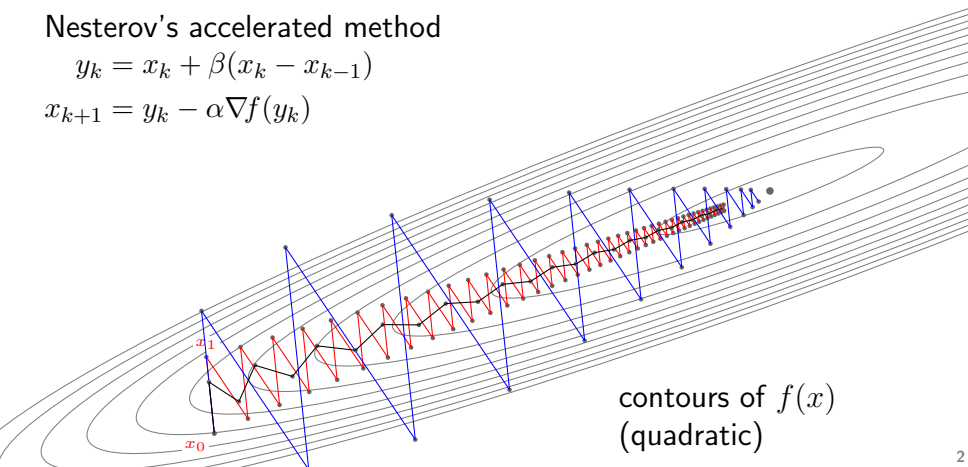
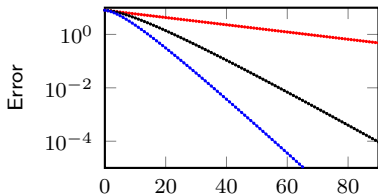
## Heavy ball method

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

## Nesterov's accelerated method

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

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



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

algorithm convergence  $\iff$  system stability

2. By solving a small convex program, we can recover (or even improve!) state-of-the-art convergence results for these algorithms, automatically and efficiently.
3. The ultimate goal: to move from analysis to design.

## Worst-case algorithm analysis

$G$  : algorithm to analyze

$f \in \mathcal{S}$  : function being minimized

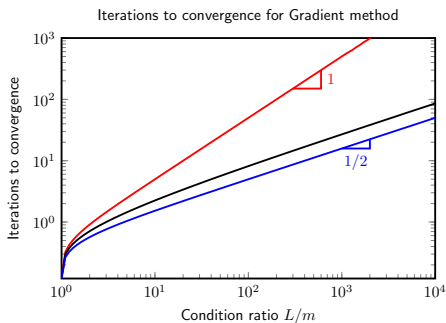
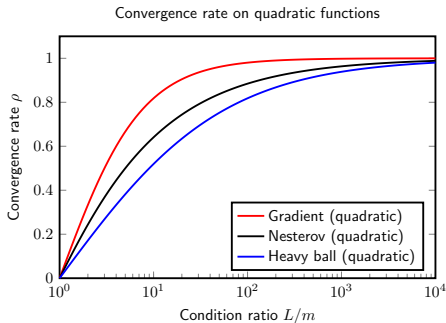
$$J_{\text{opt}} = \sup_{f \in \mathcal{S}} \text{cost}(f, G)$$

$\text{cost}(f, G)$  can be:

- $f(x_k) - f(x_*)$  (function error)
- $\|x_k - x_*\|^2$  (distance error)
- $\|\nabla f(x_k)\|^2$  (gradient)

$$G \left\{ \begin{array}{l} \text{Gradient method} \\ x_{k+1} = x_k - \alpha \nabla f(x_k) \\ \\ \text{Heavy ball method} \\ x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}) \\ \\ \text{Nesterov's accelerated method} \\ x_{k+1} = x_k - \alpha \nabla f(x_k + \beta(x_k - x_{k-1})) + \beta(x_k - x_{k-1}) \end{array} \right.$$

$$f \in \mathcal{S} \left\{ \begin{array}{l} \text{Analytically solvable:} \\ \text{Quadratic functions: } f(x) = \frac{1}{2}x^\top Qx - p^\top x \\ \text{with the constraint: } m \leq \lambda(Q) \leq L \end{array} \right.$$



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

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

## Worst-case algorithm analysis

$G$  : algorithm to analyze

$f \in \mathcal{S}$  : function being minimized

$$J_{\text{opt}} = \sup_{f \in \mathcal{S}} \text{cost}(f, G)$$

1. conventional approach
2. our approach
3. results!

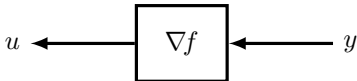
# Conventional approach

## 1. Write down everything you know

The algorithm update equations:

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

What you know about the function  $f$ ...



## Convex function

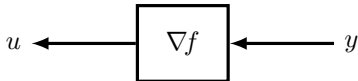
- $f(y) \geq f(x) + \nabla f(x)^\top (y - x)$
- $(\nabla f(y) - \nabla f(x))^\top (y - x) \geq 0$

## Convex with Lipschitz gradients

- $f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2} \|y - x\|^2$
- $(\nabla f(y) - \nabla f(x))^\top (y - x) \leq L \|y - x\|^2$
- $f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2L} \|\nabla f(y) - \nabla f(x)\|^2$

## Strongly convex with Lipschitz gradients

- $f(y) - f(x) - \nabla f(x)^\top (y - x) \geq \frac{1}{2(L-m)} (mL \|y - x\|^2 - 2m(\nabla f(y) - \nabla f(x))^\top (y - x) + \|\nabla f(y) - \nabla f(x)\|^2)$



## Nonconvex examples

- Weak strong convexity [Necoara et al. '15]:  
$$f(x_\star) \geq f(x) + \nabla f(x)^\top (x_\star - x) + \frac{m}{2} \|x - x_\star\|^2$$
- Restricted secant inequality [Zhang, Yin '13]:  
$$\nabla f(x)^\top (x - x_\star) \geq \frac{m}{2} \|x - x_\star\|^2$$
- Quadratic Growth [Anitescu '00]:  
$$f(x) - f(x_\star) \geq \frac{m}{2} \|x - x_\star\|^2$$
- Error bound [Luo, Tseng '93]:  
$$\frac{m}{2} \|x - x_\star\| \leq \|\nabla f(x)\|$$
- Polyak–Łojasiewicz ['63]:  
$$\frac{1}{2} \|\nabla f(x)\|^2 \geq \frac{m}{L} (f(x) - f(x_\star))$$

# Conventional approach

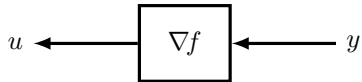
2. Combine the things you know in the right way

(be clever!)

3. Obtain result

$$\|x_k - x_\star\| \leq \rho^k \|x_0 - x_\star\|$$

We can **automate** this!



All inequalities shown were

- Quadratic in  $y$  and  $u = \nabla f(y)$
- Linear in  $f(y)$ .

# Strategy

1. Posit a Lyapunov function  $V_k$  that is quadratic in  $(x_k, x_{k-1}, \dots, x_{k-p})$  and linear in  $(f_k, f_{k-1}, \dots, f_{k-q})$ .
2. For the function class of interest, write valid inequalities. Each  $M_k^{(i)}$  is quadratic in  $(x_k, u_k)$ 's and linear in  $f_k$ 's.

$$M_k^{(i)} \geq 0 \quad \text{for } i = 1, \dots, m.$$

3. Goal is to prove that  $V_{k+1} \leq \rho^2 V_k$ .

# Strategy

For a fixed  $0 < \rho < 1$ , Look for  $V_k > 0$  and  $\lambda_i \geq 0$  such that:

$$V_{k+1} - \rho^2 V_k + \sum_{i=1}^m \lambda_i M_k^{(i)} \leq 0$$

for all values of the  $x_k$ 's and  $u_k$ 's.

Whenever  $M_k^{(i)} \geq 0$ , we have  $V_{k+1} \leq \rho^2 V_k$ , as desired.

# Strategy

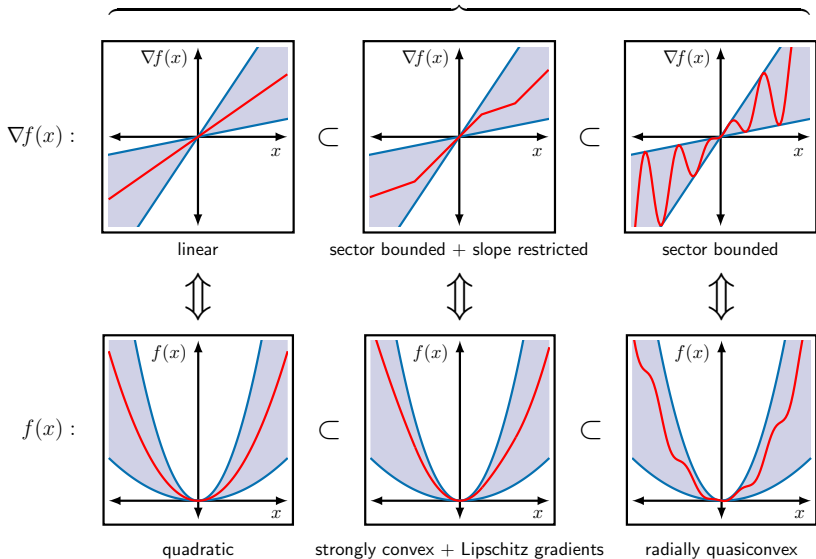
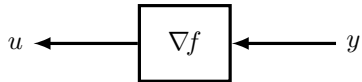
$$V_{k+1} - \rho^2 V_k + \sum_{i=1}^m \lambda_i M_k^{(i)} \leq 0$$

- Because dynamics are linear and  $V_k$ ,  $M_k^{(i)}$  are quadratic, searching for such a  $V_k$  is a **semidefinite program** (SDP)!
- The SDP is typically small (e.g.  $4 \times 4$ ). Solves in milliseconds with conventional hardware and software.
- Use bisection search to find the smallest feasible  $\rho$ .

# Connections to robust control

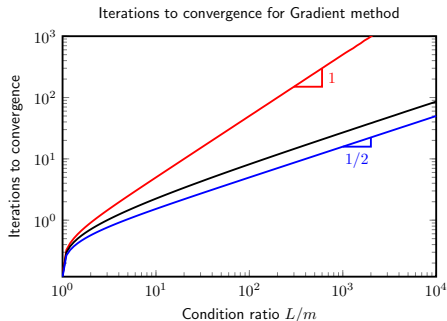
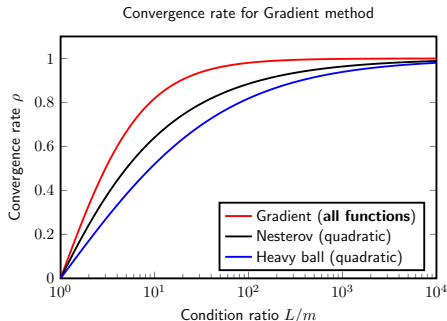
- Algorithm analysis is a [Lur'e problem](#).
- SDP is application of the [S-procedure](#).
- Lyapunov function of the form  $x_k^T P x_k + f_k$  ([Popov-type](#)) appears to be sufficient to achieve the tightest possible worst-case bounds.
- With modifications, can be framed in the context of [Integral Quadratic Constraints](#) (IQCs) from robust control [Megretski & Rantzer '97].
- For strongly convex functions with Lipschitz gradients, the best inequalities to use correspond to a particular [Zames–Falb multiplier](#) [Zames & Falb '68].

**main results:**  
analytic and numerical



# Gradient method

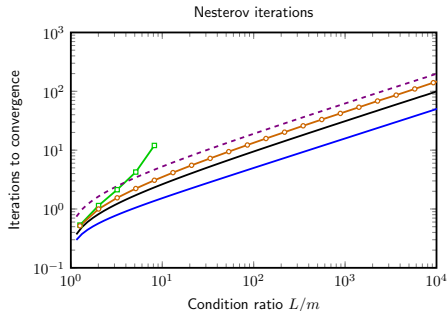
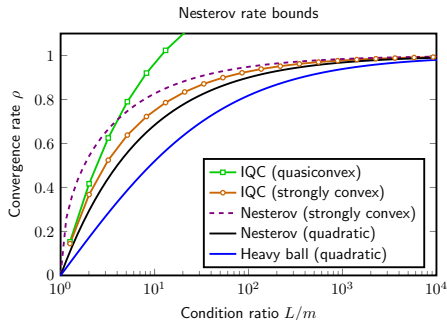
$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$



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

# Nesterov's method

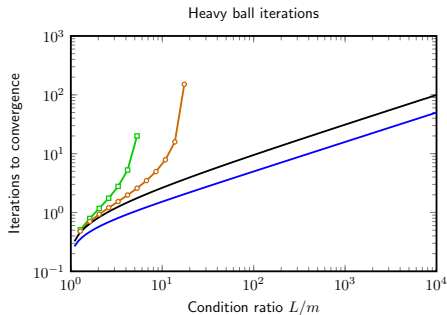
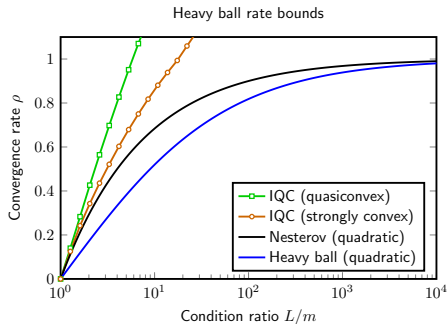
$$x_{k+1} = x_k - \alpha \nabla f(x_k + \beta(x_k - x_{k-1})) + \beta(x_k - x_{k-1})$$



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

# Heavy ball method

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

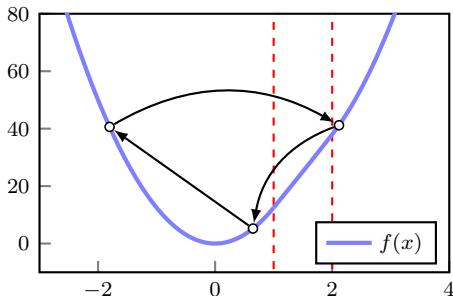


- Cannot certify stability for quasiconvex functions
- Cannot certify stability for strongly convex functions

## The heavy ball method is **not** stable!

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

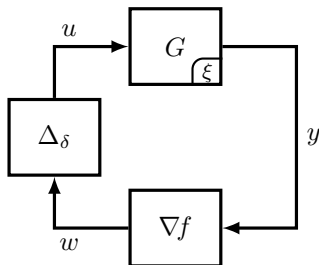
and start the heavy ball iteration at  $x_0 = x_1 \in [3.07, 3.46]$ .



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

**noise robustness and algorithm design**

# Noise robustness

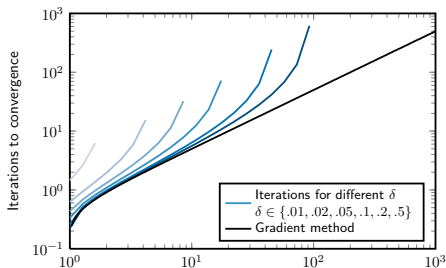
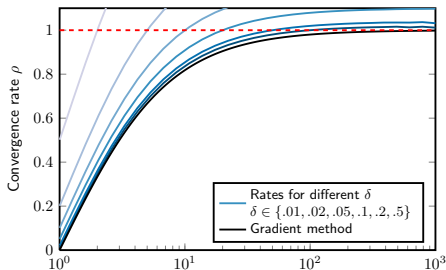


The  $\Delta_\delta$  block is uncertain multiplicative noise:

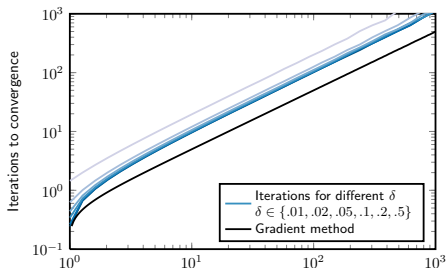
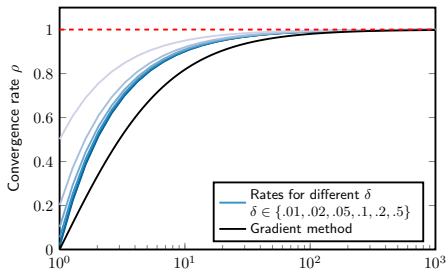
$$\|u_k - w_k\| \leq \delta \|w_k\|$$

How does an algorithm perform in the presence of adversarial noise?

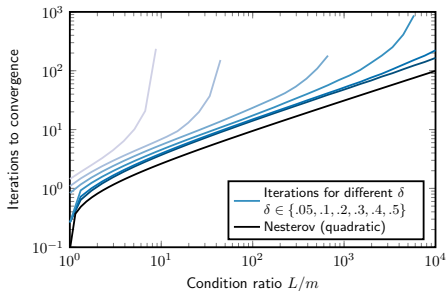
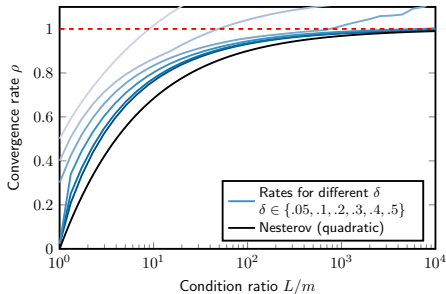
## Gradient method, $\alpha = \frac{2}{L+m}$ (optimal stepsize with no noise)



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



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



- Nesterov's method is not robust to adversarial noise.

can we have it all? (robustness AND performance)

## Design approach

- parameterize all proper  $G$  of degree 2
- parameterization in terms of  $(\alpha, \beta, \eta)$ :

$$\begin{aligned}x_{k+1} &= x_k - \alpha \nabla f(y_k) + \beta(x_k - x_{k-1}) \\y_k &= x_k + \eta(x_k - x_{k-1})\end{aligned}$$

## 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]

$$\begin{aligned}x_{k+1} &= x_k - \alpha \nabla f(y_k) + \beta(x_k - x_{k-1}) \\y_k &= x_k + \eta(x_k - x_{k-1})\end{aligned}$$

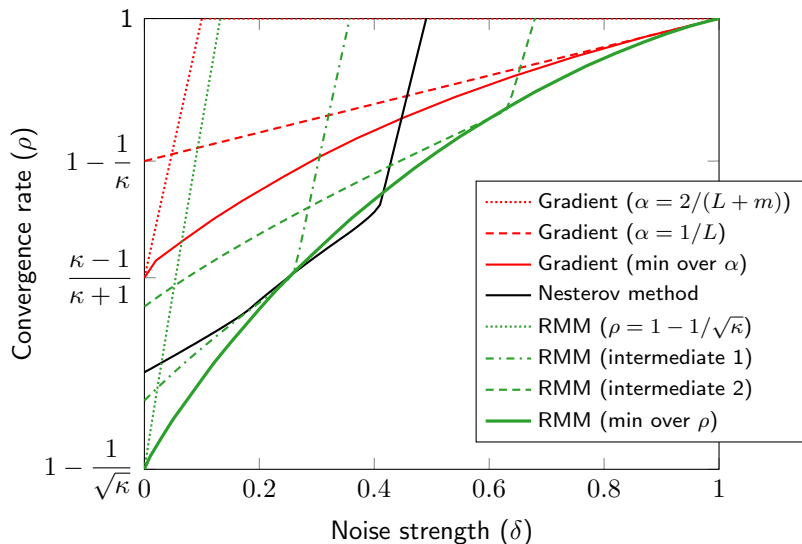
Parameters designed via root locus technique:

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

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>
- Funding acknowledgement:  
NSF 1656951, NSF 1750162.