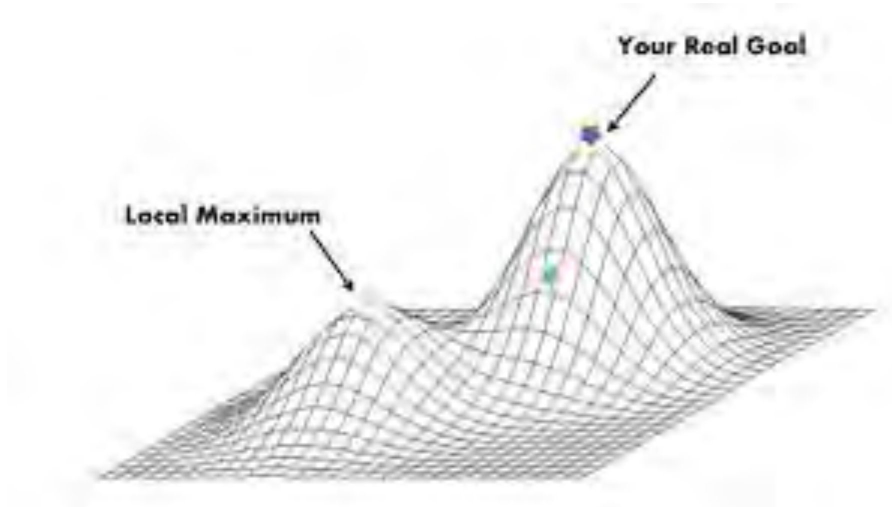
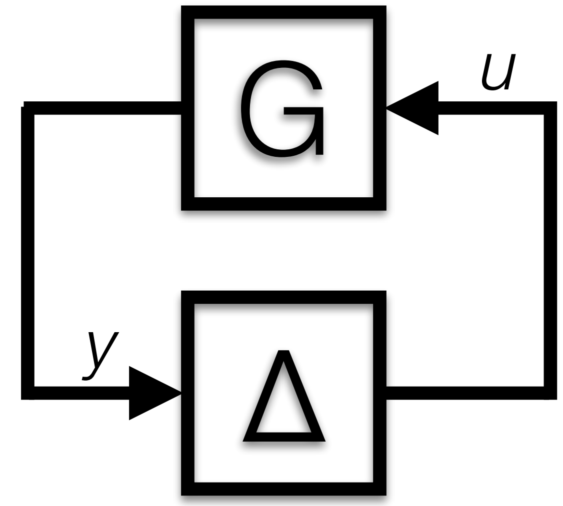
# analyzing optimization algorithms with integral quadratic constraints

Laurent Lessard, Andrew Packard, and Benjamin Recht
University of California, Berkeley

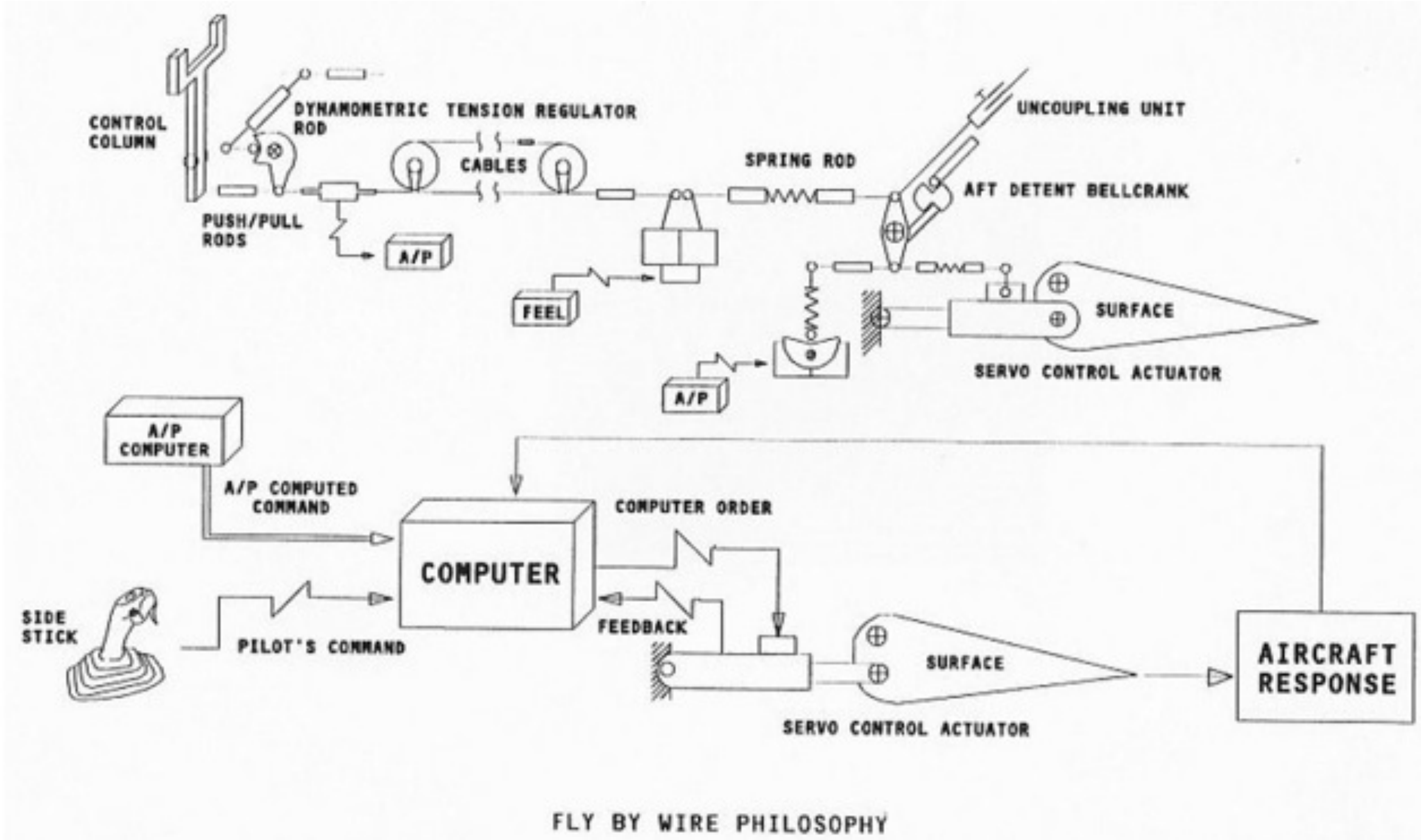


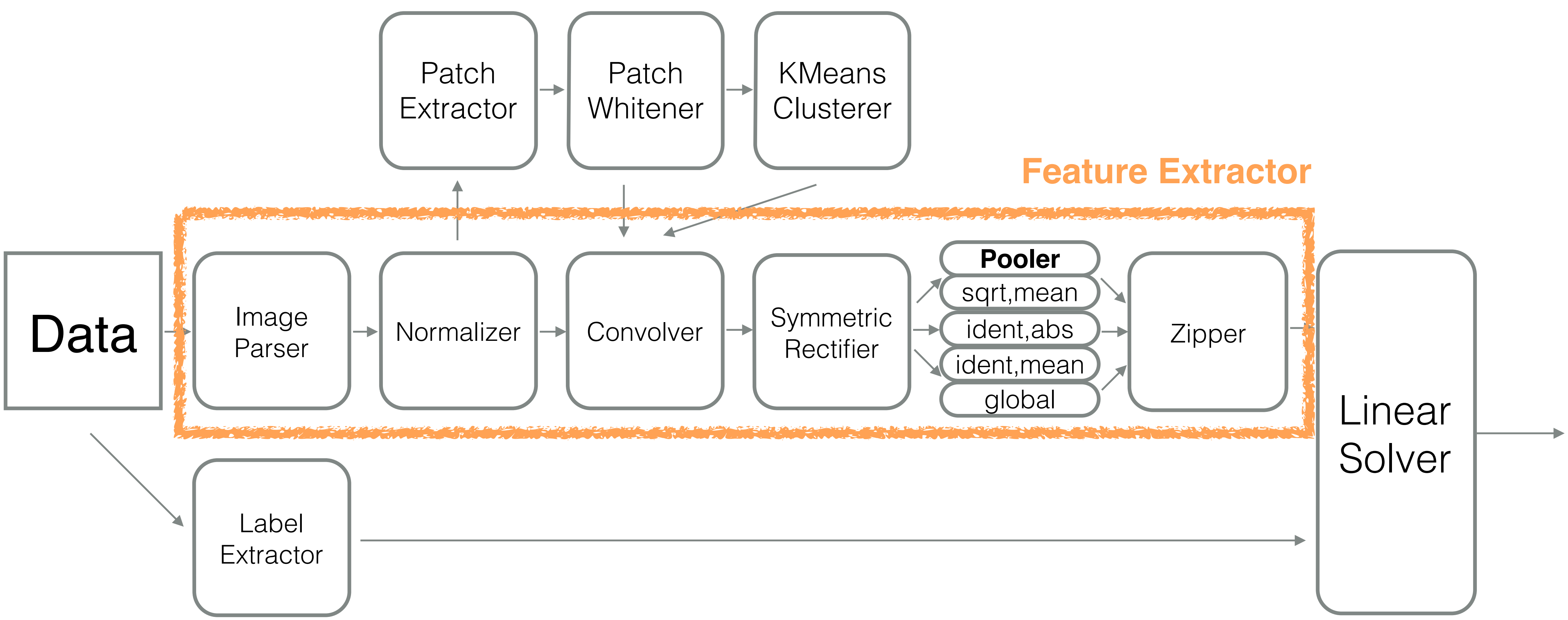




Optimization

Control

- Are joined by their arxiv category
- Controls made the SVD to SDP jump in the early 90s
- ML + Optimization perhaps now the synergistic duo
- There are many untapped analysis tools from controls


CONTROL COLUMN
DYNAMOMETRIC ROD
TENSION REGULATOR
CABLES
SPRING ROD
UNCOUPLING UNIT
AFT DETENT BELLCRANK
PUSH/PULL RODS
A/P
FEEL
SURFACE
SERVO CONTROL ACTUATOR
A/P
A/P COMPUTER
A/P COMPUTED COMMAND
COMPUTER
COMPUTER ORDER
FEEDBACK
SIDE STICK
PILOT'S COMMAND
SURFACE
SERVO CONTROL ACTUATOR
AIRCRAFT RESPONSE
FLY BY WIRE PHILOSOPHY


Patch Extractor
Patch Whitener
KMeans Clusterer
Feature Extractor
Data
Image Parser
Normalizer
Convolver
Symmetric Rectifier
Pooler
sqrt,mean
ident,abs
ident,mean
global
Zipper
Linear Solver
Label Extractor

# optimization (for big data?)

convex cost

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega \end{array}$$

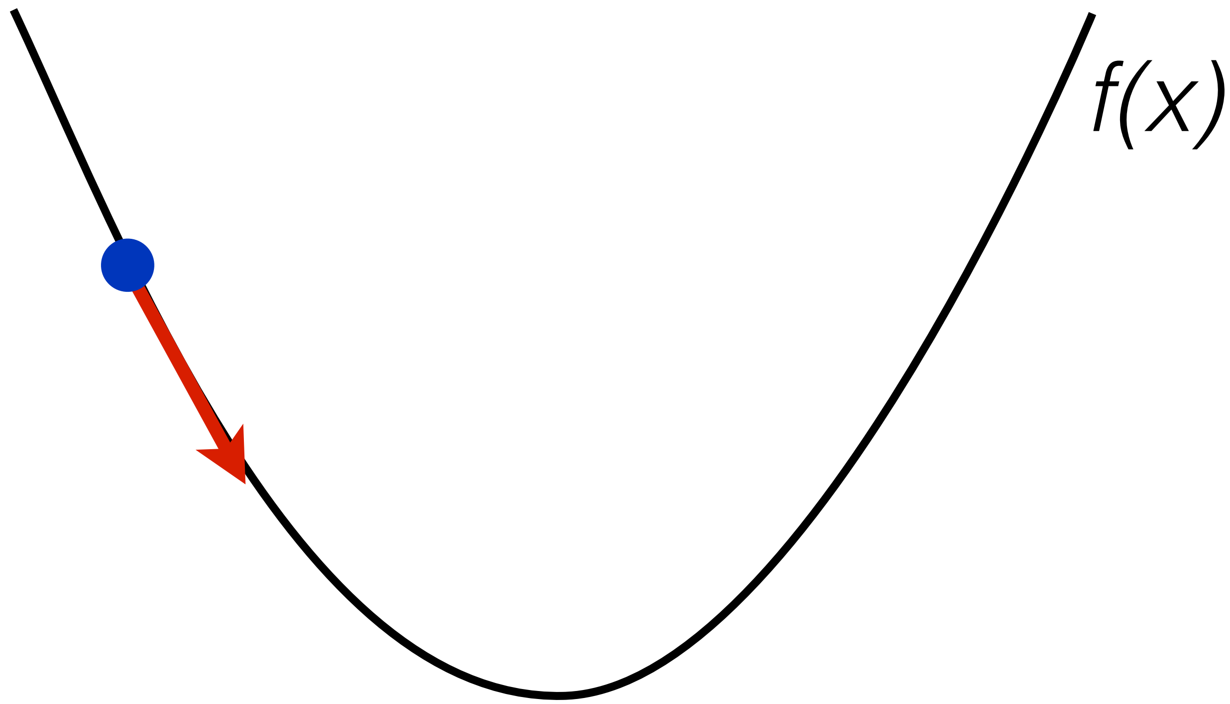
"simple," convex constraints

- closely related cousin where $P$ is a simple convex function: minimize $f(x) + P(x)$

- need algorithms that scale linearly (or sub-linearly) with dimension and data
- currently favored family are the *first-order methods*

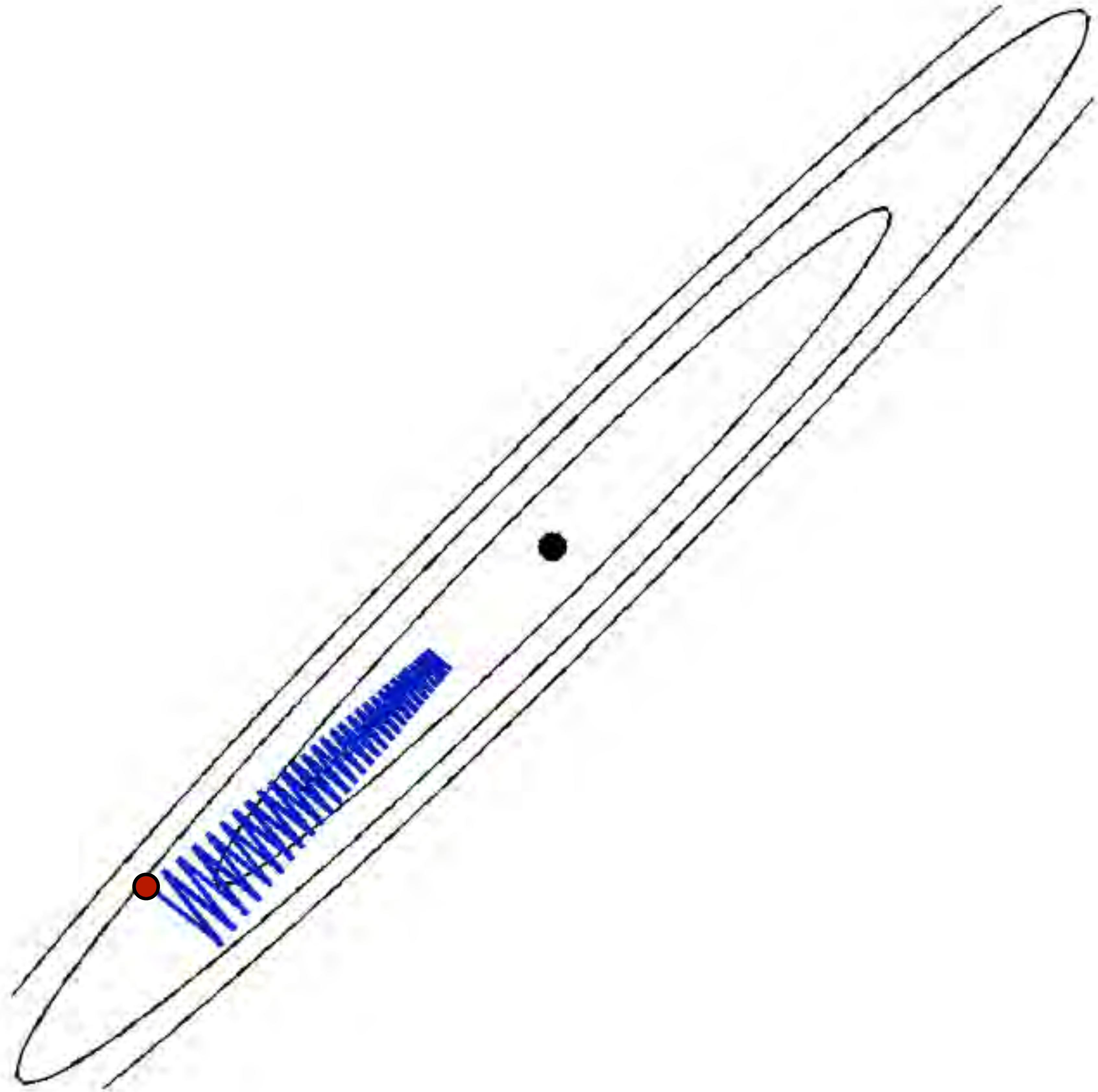
gradient descent

$$x[k+1] = x[k] - \alpha \nabla f(x[k])$$




for constrained optimization, use projected gradient descent

$$x[k+1] = \Pi_{\Omega}(x[k] - \alpha \nabla f(x[k]))$$



# acceleration/multistep

gradient method akin to an ODE

$$x[k+1] = x[k] - \alpha \nabla f(x[k])$$

$$\dot{x} = -\nabla f(x)$$

to prevent oscillation, add a second order term

$$\ddot{x} = -b\dot{x} - \nabla f(x)$$

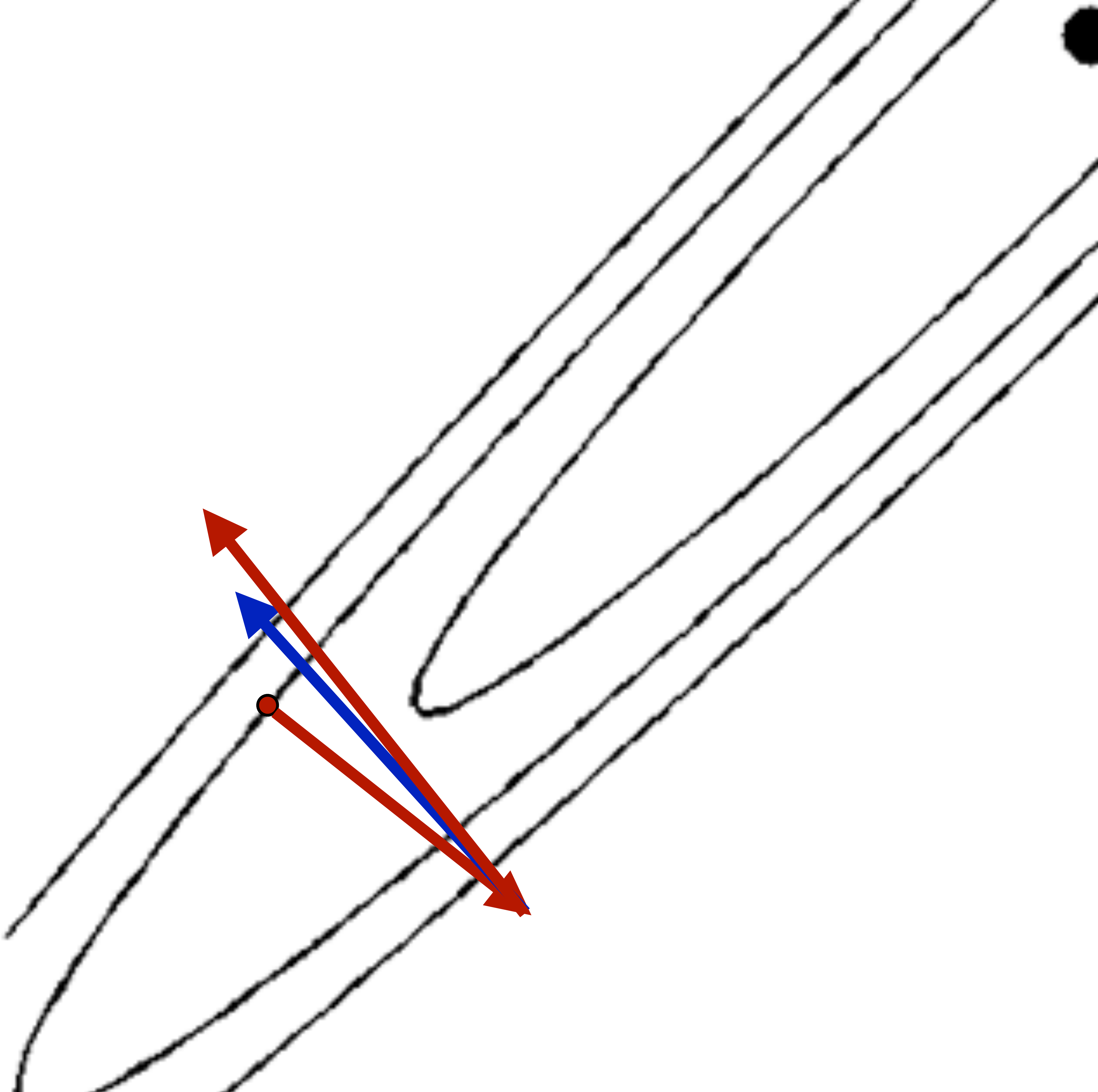
$$x[k+1] = x[k] - \alpha \nabla f(x[k]) + \beta(x[k] - x[k-1])$$

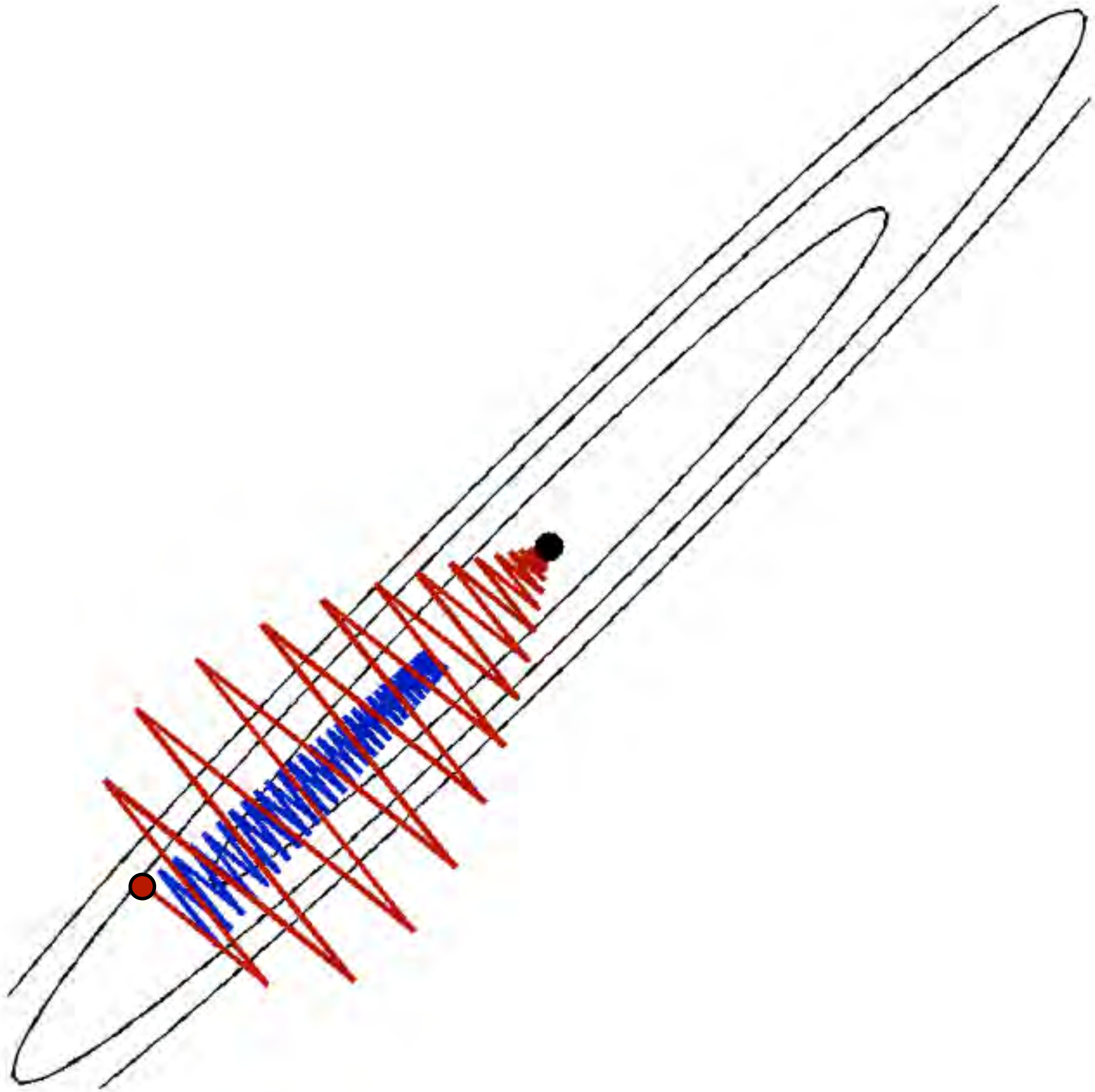
*heavy ball method* (constant α,β)

$$x[k+1] = y[k] - \alpha \nabla f(x[k])$$

$$y[k] = (1+\beta)x[k] - \beta x[k-1]$$

when *f* is quadratic, this is
*Chebyshev's iterative method*





# canonical first order methods

Gradient

$$x[k+1] = x[k] - \alpha \nabla f(x[k])$$

Heavy Ball

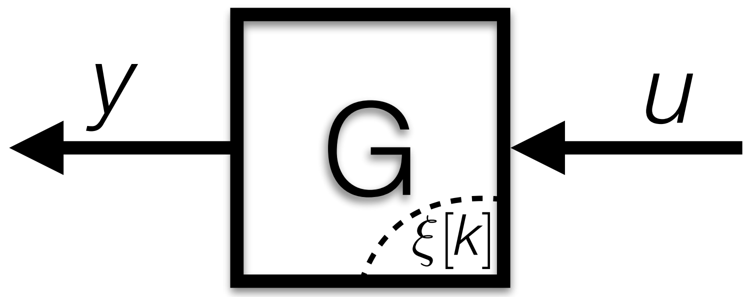
$$\begin{aligned} x[k+1] &= y[k] - \alpha \nabla f(x[k]) \\ y[k] &= (1+\beta)x[k] - \beta x[k-1] \end{aligned}$$

Nesterov

$$\begin{aligned} x[k+1] &= y[k] - \alpha \nabla f(y[k]) \\ y[k] &= (1+\beta)x[k] - \beta x[k-1] \end{aligned}$$

- each analyzed using specialized techniques
- what's the right algorithm for *my* problem?
- are there other algorithms in this space that could be more effective for specific instances?

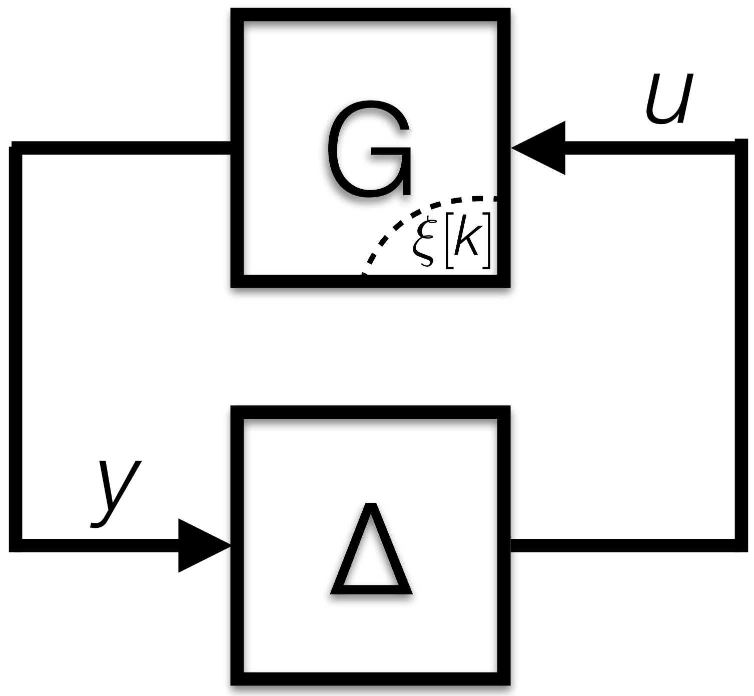
Control theory is the study of dynamical systems with inputs




$$\xi[k+1] = A\xi[k] + Bu[k]$$
$$y[k] = C\xi[k] + Du[k]$$

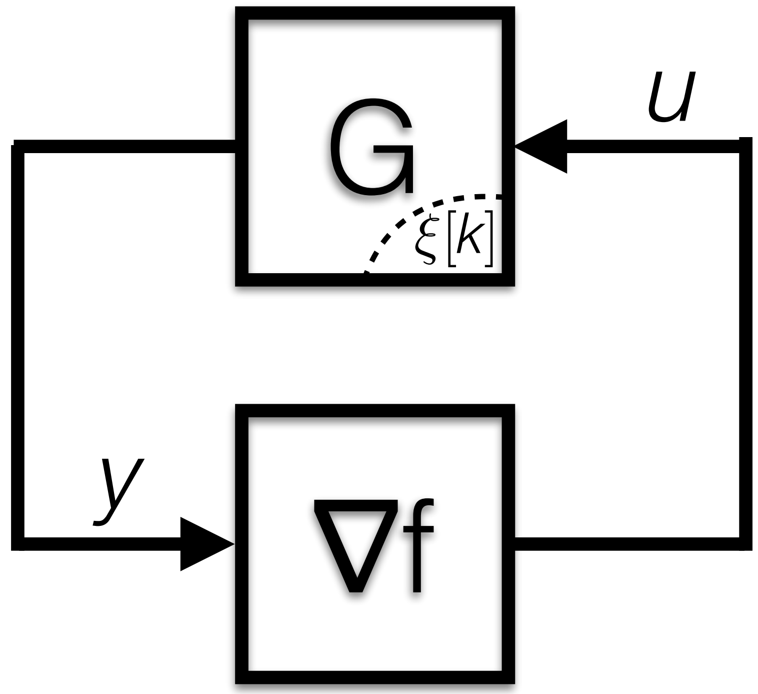
Simplest case of such systems are *linear systems*

# The Lur'e problem




$$
\begin{aligned}
\xi[k+1] &= A\xi[k] + Bu[k] \\
y[k] &= C\xi[k] + Du[k] \\
u[k] &= \Delta(y[k])
\end{aligned}
$$

- A linear dynamical system is connected in feedback with a nonlinearity.
- When do all trajectories converge to a fixed point?




$$\begin{aligned} \xi[k+1] &= A\xi[k] + Bu[k] \\ y[k] &= C\xi[k] + Du[k] \\ u[k] &= \nabla f(y[k]) \end{aligned}$$

*method*

Gradient

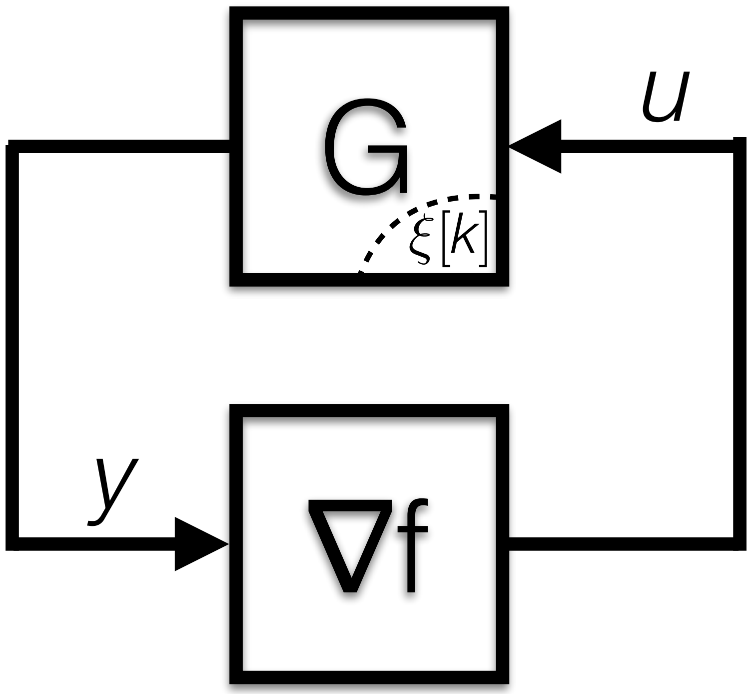
$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} 1 & -\alpha \\ \hline 1 & 0 \end{array}\right]$$

Heavy Ball

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} -\alpha \\ 0 \end{bmatrix} \\ \hline \begin{bmatrix} 1 & 0 \end{bmatrix} & 0 \end{array}\right]$$

Nesterov

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} -\alpha \\ 0 \end{bmatrix} \\ \hline \begin{bmatrix} 1+\beta & -\beta \end{bmatrix} & 0 \end{array}\right]$$




$$
\begin{aligned}
\xi[k+1] &= A\xi[k] + Bu[k] \\
y[k] &= C\xi[k] + Du[k] \\
u[k] &= \nabla f(y[k])
\end{aligned}
$$

*method*

Gradient

$$
\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} 1 & -\alpha \\ \hline 1 & 0 \end{array}\right]
$$

$$
\begin{aligned}
\xi[k+1] &= \xi[k] - \alpha u[k] \\
y[k] &= \xi[k] \\
u[k] &= \nabla f(y[k])
\end{aligned}
$$

$$
x[k+1] = x[k] - \alpha \nabla f(x[k])
$$

Nesterov

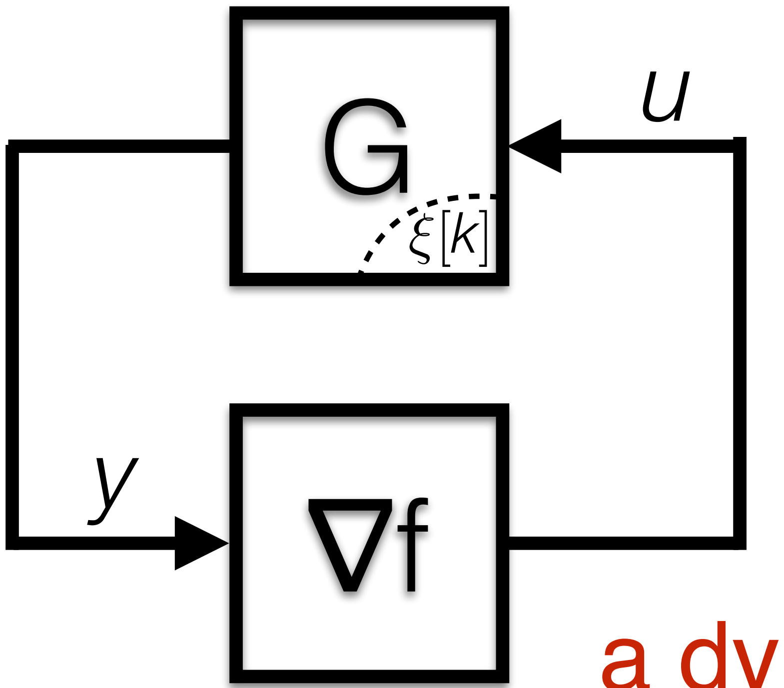
$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} -\alpha \\ 0 \end{bmatrix} \\ \hline \begin{bmatrix} 1+\beta & -\beta \end{bmatrix} & 0 \end{array}\right]$$

$$\begin{aligned} \xi_1[k+1] &= (1+\beta)\xi_1[k] - \beta\xi_2[k] - \alpha u[k] \\ \xi_2[k+1] &= \xi_1[k] \\ y[k] &= (1+\beta)\xi_1[k] - \beta\xi_2[k] \\ u[k] &= \nabla f(y[k]) \end{aligned}$$

$\xi_2[k] = \xi_1[k-1]$

$$\begin{aligned} \xi_1[k+1] &= (1+\beta)\xi_1[k] - \beta\xi_1[k-1] - \alpha u[k] \\ y[k] &= (1+\beta)\xi_1[k] - \beta\xi_1[k-1] \\ u[k] &= \nabla f(y[k]) \end{aligned}$$

$$\begin{aligned} x[k+1] &= y[k] - \alpha\nabla f(y[k]) \\ y[k] &= (1+\beta)x[k] - \beta x[k-1] \end{aligned}$$

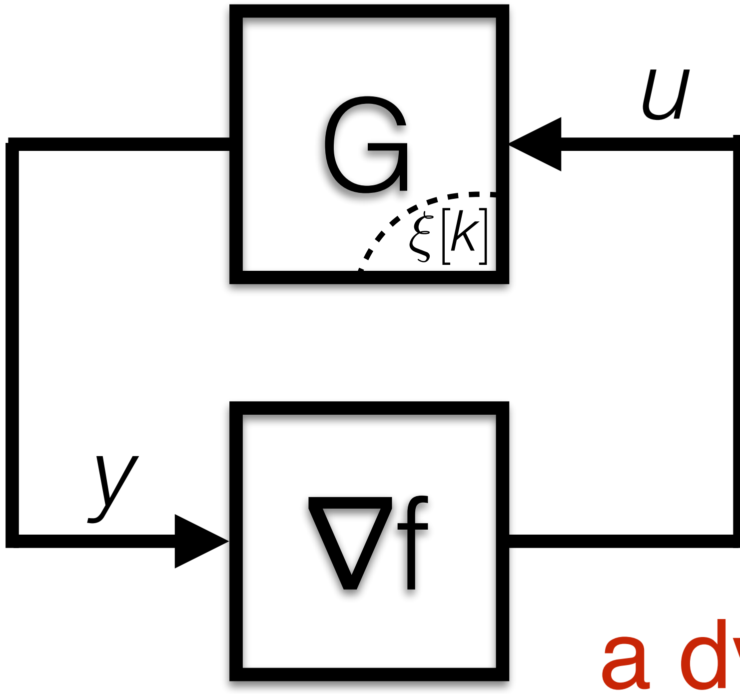



$$\begin{aligned}\xi[k+1] &= A\xi[k] + Bu[k]\\ y[k] &= C\xi[k] + Du[k]\\ u[k] &= \nabla f(y[k])\end{aligned}$$

How do you prove ~~an algorithm converges?~~ a dynamical system is stable?

**Step 1:** find a fixed point.

$$\nabla f(x_\star) = 0 \implies \begin{cases} y_\star = x_\star \\ u_\star = 0 \\ \xi_\star = A\xi_\star \\ x_\star = C\xi_\star \end{cases}$$




$$\xi[k+1] = A\xi[k] + Bu[k]$$
$$y[k] = C\xi[k] + Du[k]$$
$$u[k] = \nabla f(y[k])$$

How do you prove ~~an algorithm converges?~~ a dynamical system is stable?

**Step 2:** prove all trajectories converge to the fixed point

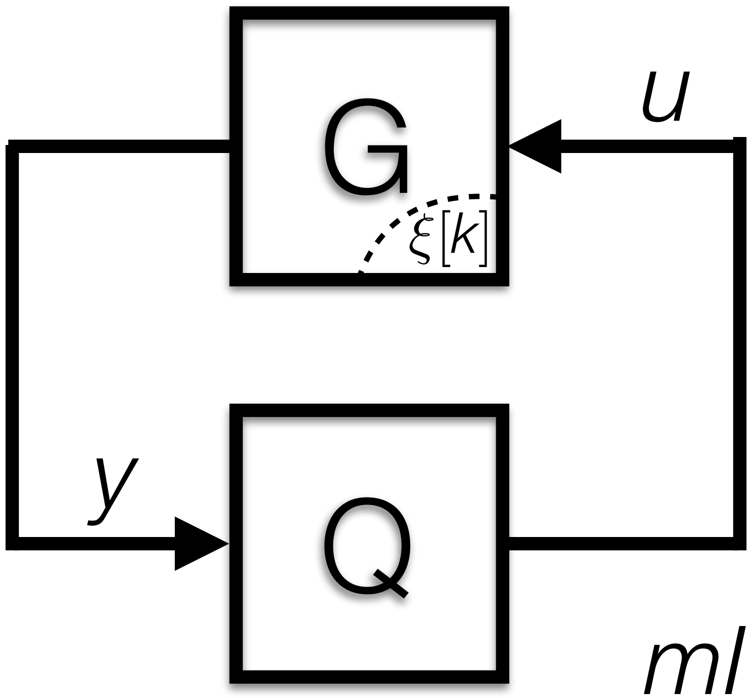
**Simple case:** $f(x) = \frac{1}{2}x^T Q x - p^T x$

$$\nabla f(x) = Qx - p \qquad x_\star = Q^{-1}p$$

$$\xi[k+1] - \xi_\star = (A + BQC)(\xi[k] - \xi_\star)$$

Necessary and sufficient condition is $\rho(A + BQC) < 1$

$$\lim_{k\to\infty} \|\xi[k] - \xi_\star\|^{1/k} \leq \rho(A + BQC)$$




$$\xi[k+1] = A\xi[k] + Bu[k]$$
$$y[k] = C\xi[k] + Du[k]$$
$$u[k] = Qy[k]$$

$$mI \preceq Q \preceq LI$$

$$\kappa = L/m$$

*method*

| | | |
|---|---|---|
| Gradient | $\alpha = \frac{2}{L+m}$ | $\rho(A+BQC) \le \frac{\kappa-1}{\kappa+1}$ |
| Heavy Ball | $\alpha = \frac{4}{(\sqrt{L}+\sqrt{m})^2}$ <br> $\beta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ | $\rho(A+BQC) \le \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{1/2}$ |
| Nesterov | $\alpha = \frac{1}{L}$ <br> $\beta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ | $\rho(A+BQC) \le 1 - \frac{1}{\sqrt{\kappa}}$ |

**Theorem:** $\rho(A) < \rho$ if and only if there exists

$$P \succeq 0 \quad \text{satisfying} \quad A^T P A - \rho^2 P \prec 0$$

**Proof:** If $\rho(A) < \rho$ , then $P = \sum_{k=0}^{\infty} \rho^{-2k} (A^T)^k A^k$

exists and satisfies the desired LMI.

Conversely, assume the LMI has a solution and let λ be an eigenvalue with corresponding eigenvector $\xi$. Then

$$\xi^T A^T P A \xi - \rho^2 \xi^T P \xi = (|\lambda|^2 - \rho^2) \xi^T P \xi < 0$$

which implies $|\lambda|^2 < \rho^2$

Theorem: $\rho(A) < \rho$ if and only if there exists

$$P \succeq 0 \quad \text{satisfying} \quad A^T P A - \rho^2 P \prec 0$$

For dynamical systems, if $\xi[k+1] = A\xi[k]$ the LMI implies

$$\xi[k+1]^T P \xi[k+1] < \rho^2 \xi[k]^T P \xi[k]$$

Iterating the recursion to k=0 gives

$$\xi[k]^T P \xi[k] < \rho^{2k} \xi[0]^T P \xi[0]$$

which in turn implies

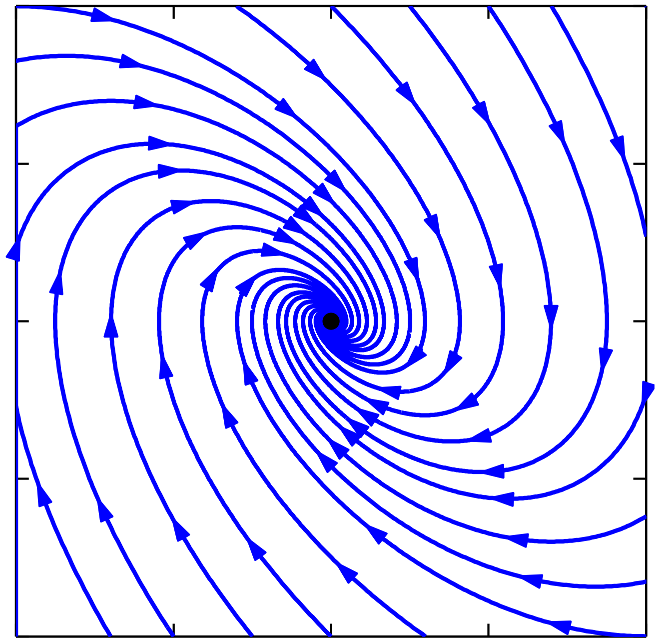
$$\|\xi[k]\| \leq \sqrt{\mathrm{cond}(P)}\rho^k \|\xi_0\|$$

# Lyapunov functions

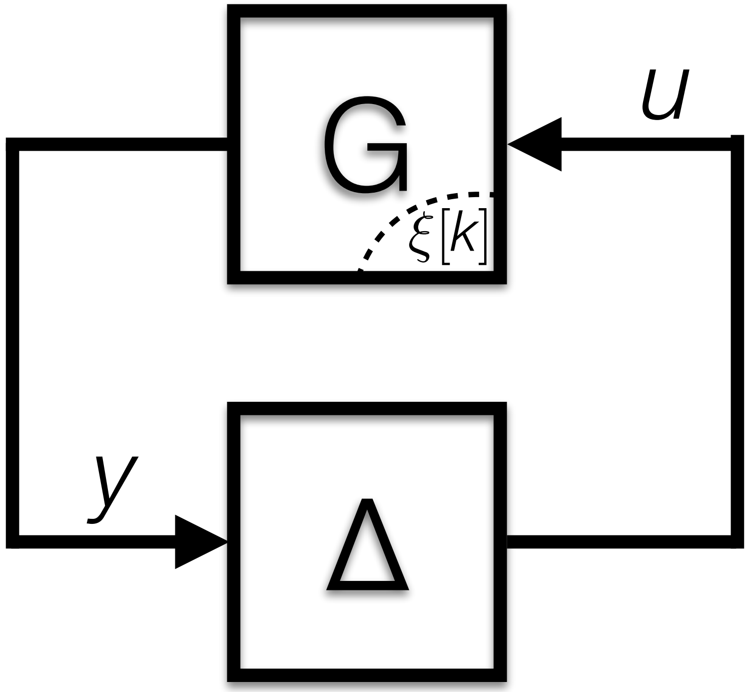
$V(x) \geq 0$

$V(x_\star) = 0$

$V(x[k]) < V(x[k-1])$



- LMI characterization of stability parametrizes quadratic Lyapunov functions for the system
- This notion generalizes to nonlinear systems




$$\xi[k+1] = A\xi[k] + Bu[k]$$
$$y[k] = C\xi[k] + Du[k]$$
$$u[k] = \Delta(y[k])$$

How do we prove the interconnection is stable?

Suppose there exists a P>0 and matrix M such that

$$\begin{bmatrix} y_1 - y_2 \\ \Delta(y_1) - \Delta(y_2) \end{bmatrix}^T M \begin{bmatrix} y_1 - y_2 \\ \Delta(y_1) - \Delta(y_2) \end{bmatrix} \geq 0 \quad \text{for all y1, y2}$$

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

Then $(\xi[k] - \xi_\star)^T P (\xi[k] - \xi_\star) \leq \rho^{2k} (\xi[0] - \xi_\star)^T P (\xi[0] - \xi_\star)$

$$\xi[k+1] = A\xi[k] + Bu[k]$$
$$y[k] = C\xi[k] + Du[k]$$
$$u[k] = \Delta(y[k])$$

and there exists a P

$$\begin{bmatrix} y_1 - y_2 \\ \Delta(y_1) - \Delta(y_2) \end{bmatrix}^T M \begin{bmatrix} y_1 - y_2 \\ \Delta(y_1) - \Delta(y_2) \end{bmatrix} \geq 0 \quad \text{for all y1, y2}$$

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

Multiply both sides by $\begin{bmatrix} \xi[k] - \xi_\star \\ u[k] - u_\star \end{bmatrix}$

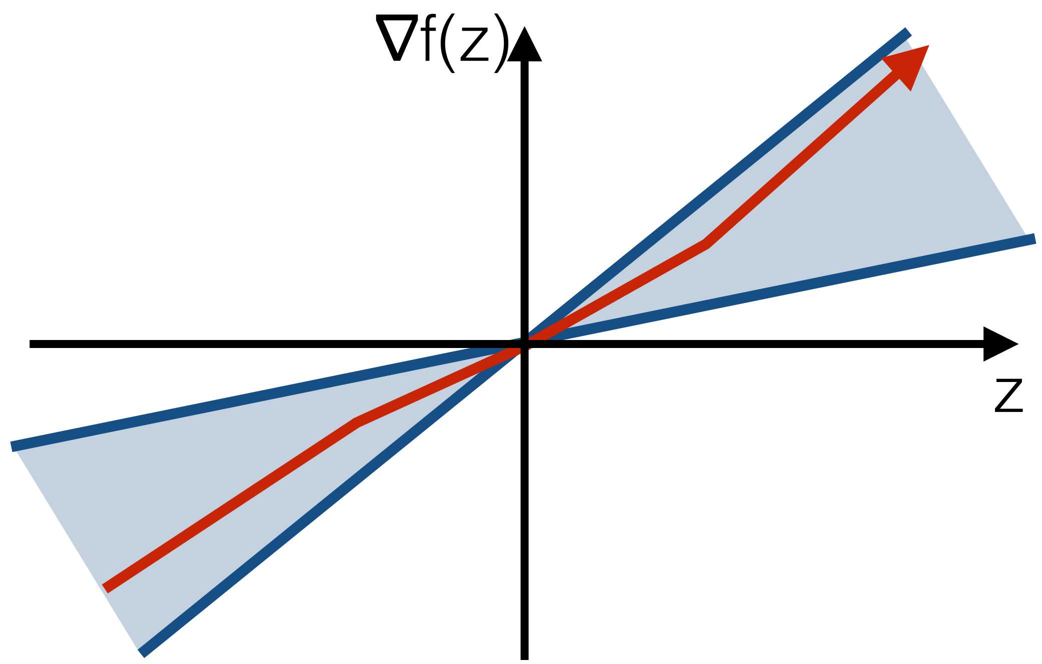
$$(\xi[k+1] - \xi_\star)^T P (\xi[k+1] - \xi_\star) - \rho^2 (\xi[k] - \xi_\star)^T P (\xi[k] - \xi_\star)$$
$$+ \begin{bmatrix} y[k] - y_\star \\ u[k] - u_\star \end{bmatrix}^T M \begin{bmatrix} y[k] - y_\star \\ u[k] - u_\star \end{bmatrix} \leq 0$$

# Gradient method

## Sector QC

$$\begin{bmatrix} z_1 - z_2 \\ \nabla f(z_1) - \nabla f(z_2) \end{bmatrix}^T \begin{bmatrix} -2mLI_d & (L+m)I_d \\ (L+m)I_d & 2I_d \end{bmatrix} \begin{bmatrix} z_1 - z_2 \\ \nabla f(z_1) - \nabla f(z_2) \end{bmatrix} \geq 0$$

*aka cocoercivity:* $\langle \nabla f(z_1) - \nabla f(z_2), z_1 - z_2 \rangle \geq \frac{1}{L} \|\nabla f(z_1) - \nabla f(z_2)\|^2$




**Proposition:** If *f* is convex, then *f* satisfies the Sector QC iff *f* has *L*-Lipschitz gradients and is strongly convex with parameter *m.*

# Gradient method

## Sector QC

$$\begin{bmatrix} z_1 - z_2 \\ \nabla f(z_1) - \nabla f(z_2) \end{bmatrix}^T \underbrace{\begin{bmatrix} -2mLI_d & (L+m)I_d \\ (L+m)I_d & 2I_d \end{bmatrix}}_{M} \begin{bmatrix} z_1 - z_2 \\ \nabla f(z_1) - \nabla f(z_2) \end{bmatrix} \geq 0$$

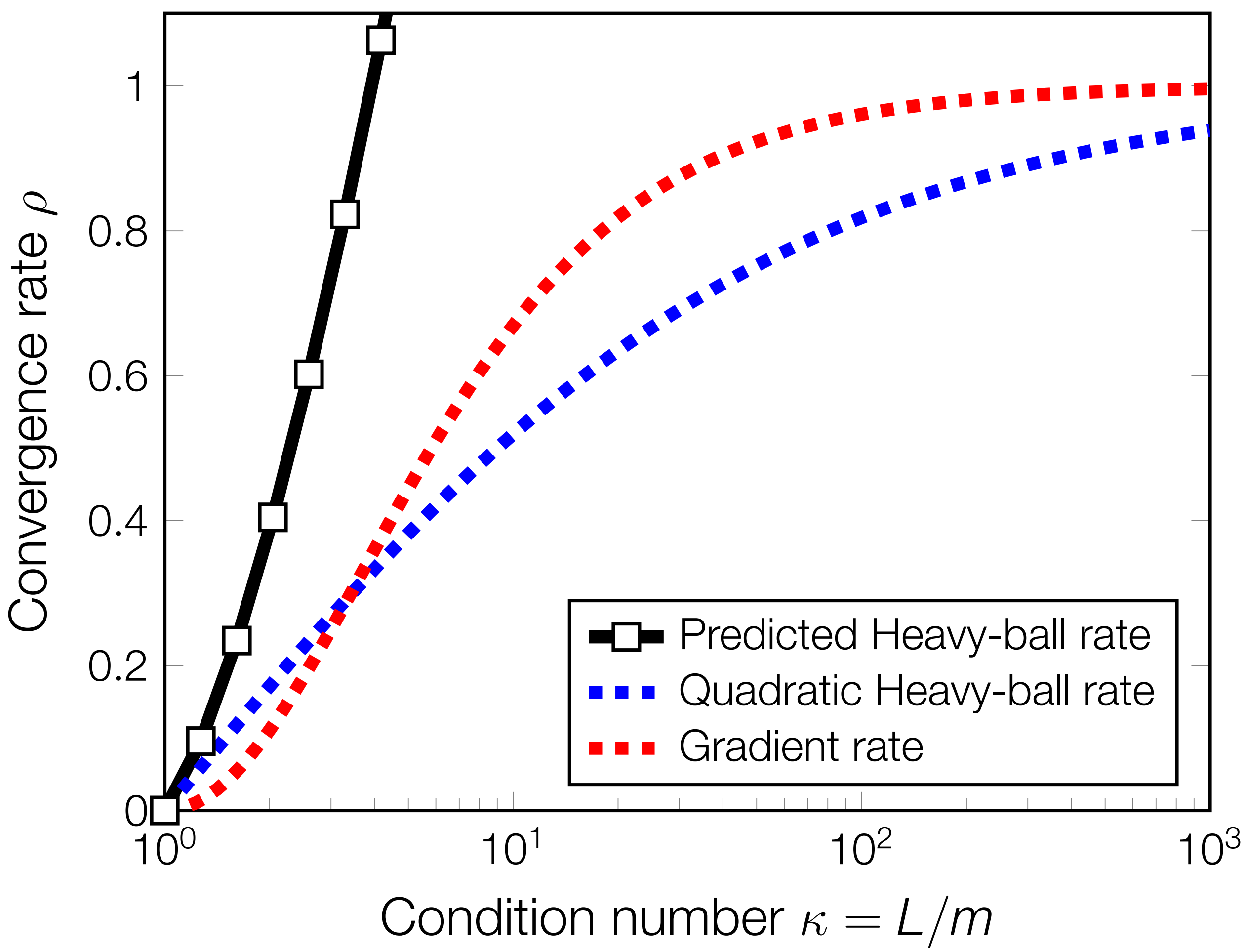
$$\begin{bmatrix} A & B \end{bmatrix}^T P \begin{bmatrix} A & B \end{bmatrix} - \begin{bmatrix} \rho^2 P & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^T M \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \preceq 0$$

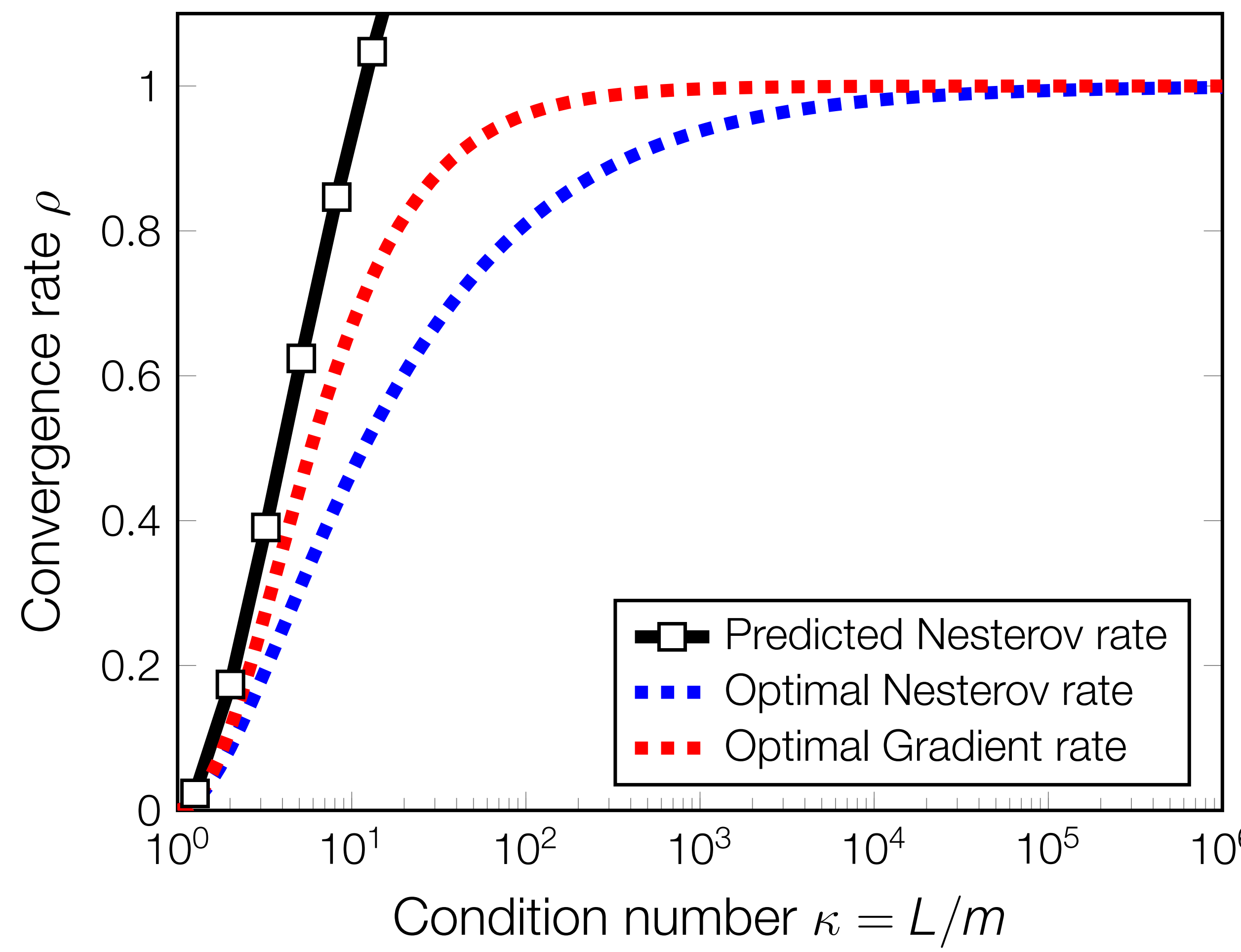
$$p \begin{bmatrix} 1-\rho^2 & -\alpha \\ -\alpha & \alpha^2 \end{bmatrix} + \mu \begin{bmatrix} -2mL & L+m \\ L+m & -2 \end{bmatrix} \preceq 0$$

Setting p=1, and setting the LMI to be exactly equal to zero, gives

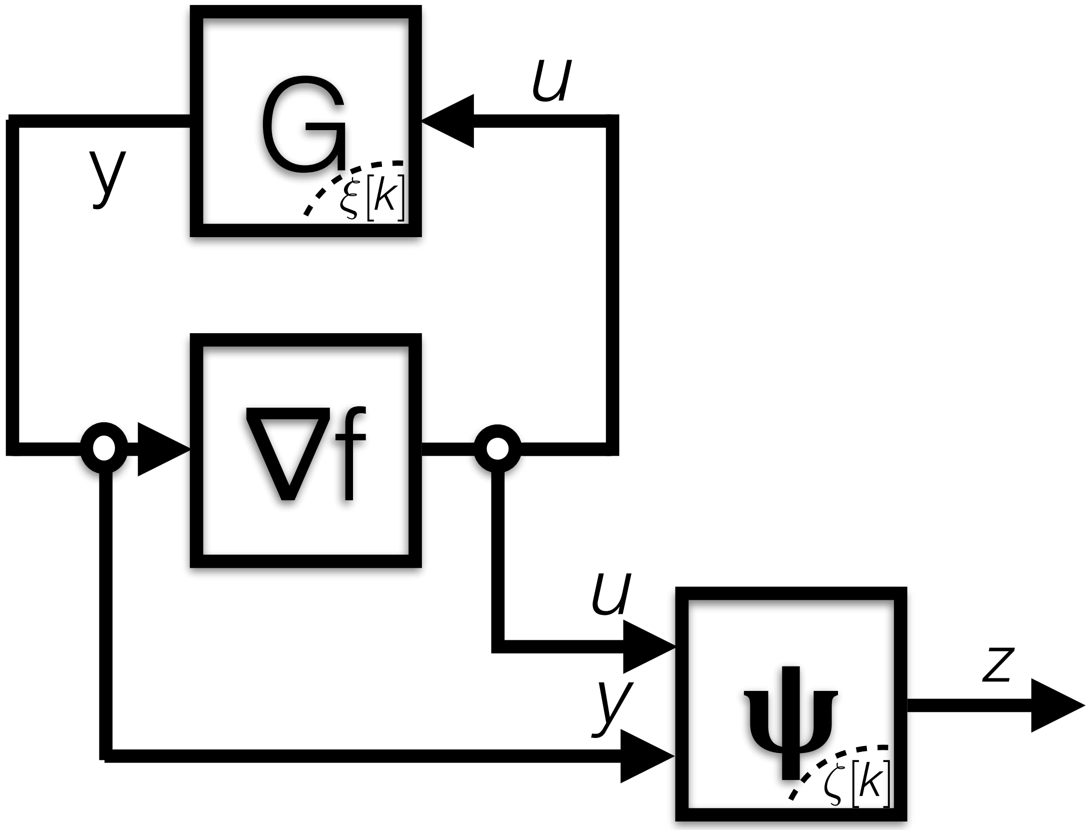
$$\rho = \frac{\kappa - 1}{\kappa + 1}$$

# Heavy Ball and Nesterov




The sector quadratic constraint is not sufficient to prove stability




$$
\begin{aligned}
\xi[k+1] &= A\xi[k] + Bu[k] \\
y[k] &= C\xi[k] + Du[k] \\
u[k] &= \nabla f(y[k])
\end{aligned}
$$

$$
\begin{aligned}
\zeta[k+1] &= A_\Psi \zeta[k] + B_\Psi^{(u)} u[k] + B_\Psi^{(y)} y[k] \\
z[k] &= C_\Psi \zeta[k] + D_\Psi^{(u)} u[k] + D_\Psi^{(y)} y[k]
\end{aligned}
$$

**Main Result (1):** Suppose that there exists a linear system $\mathbf{\Psi}$ and a matrix M such that for any sequence $y_1, \ldots, y_T$

$$\sum_{k=1}^{T} \rho^{-2k} (z[k]-z_\star)^T M (z[k]-z_\star) \geq 0$$

*integral quadratic constraint*

and there exists a P>0 such that

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

*composite system matrices*

Then $(\hat{\xi}[k]-\hat{\xi}_\star)^T P (\hat{\xi}[k]-\hat{\xi}_\star) \leq \rho^{2k} (\hat{\xi}[0]-\hat{\xi}_\star)^T P (\hat{\xi}[0]-\hat{\xi}_\star)$

# off-by-one IQC

**Main Result (2):** Let *f* be a strongly convex function with *L*-Lipschitz gradients and strong convexity parameter *m*. Then for any sequence y[0],…,y[T] with u[k] = ∇f(y[k])

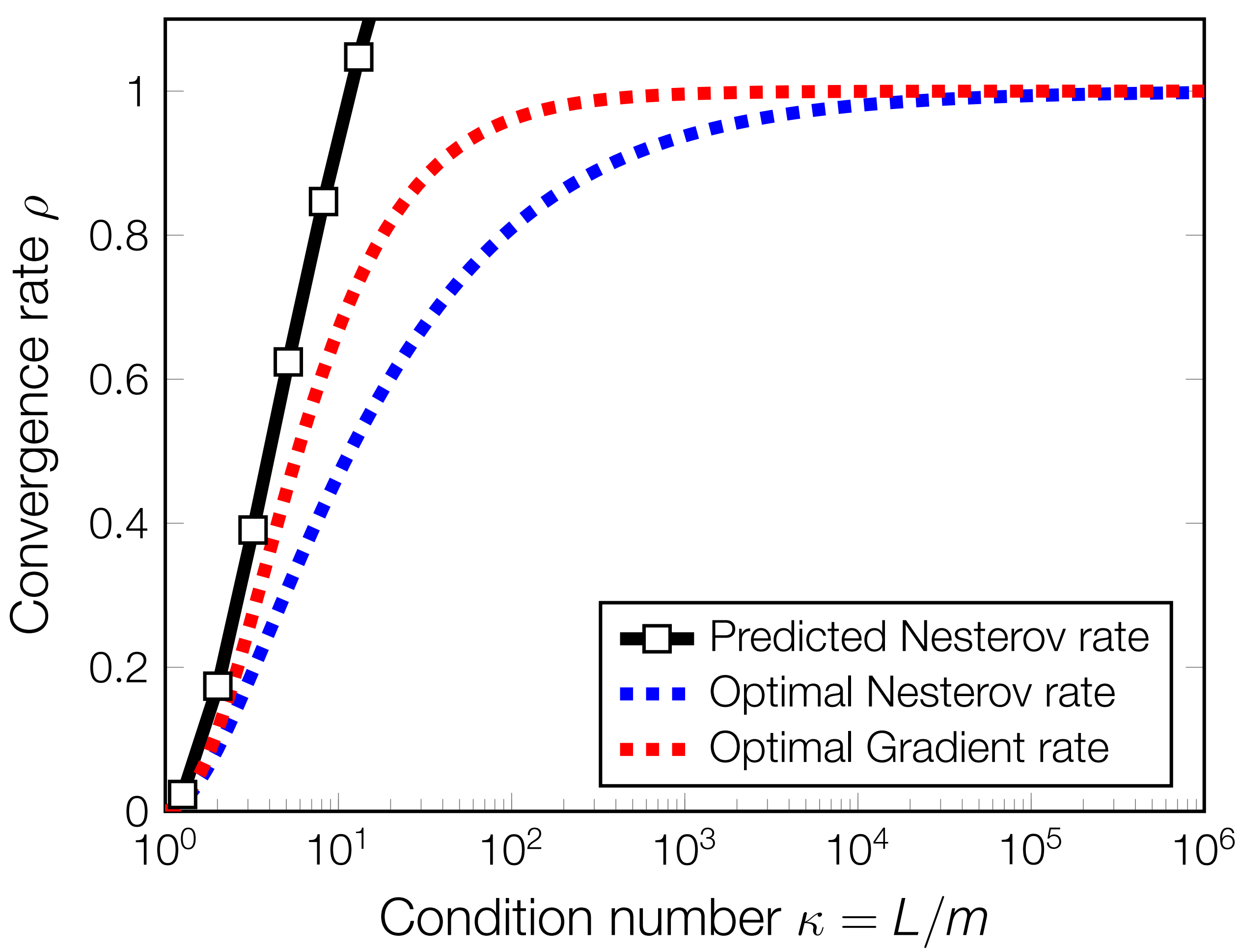
$$\sum_{k=1}^{T} \rho^{-2k}(u[k]-my[k])^T\{L(y[k]-\rho^2 y[k-1])-(u[k]-\rho^2 u[k-1])\}\geq 0$$

- Without the delay terms (ρ=0), this is just the sector QC
- Builds on *Popov* and *Zames-Falb multipliers* from control.
- Elementary proof using co-coercivity inequalities.

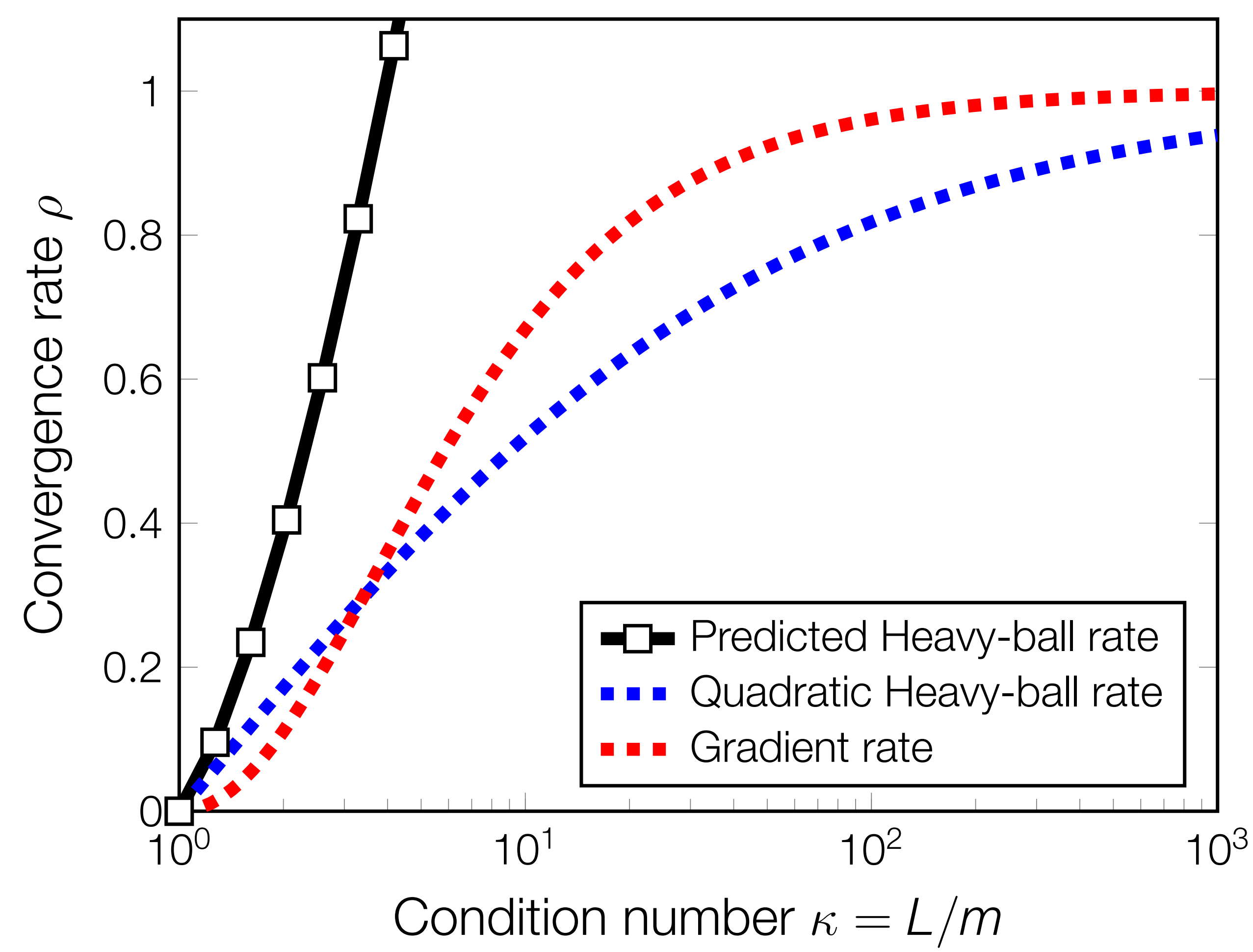
$$\sum_{k=1}^{T} \rho^{-2k}\,(z[k]-z_\star)^T M(z[k]-z_\star)\geq 0$$

$$\left[\begin{array}{c|c} A_\Psi & B_\Psi \\ \hline C_\Psi & D_\Psi \end{array}\right] = \left[\begin{array}{c|cc} 0 & \rho L I_d & \rho I_d \\ \hline -\rho I_d & L I_d & -I \\ 0 & -m I_d & I_d \end{array}\right] \qquad M = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

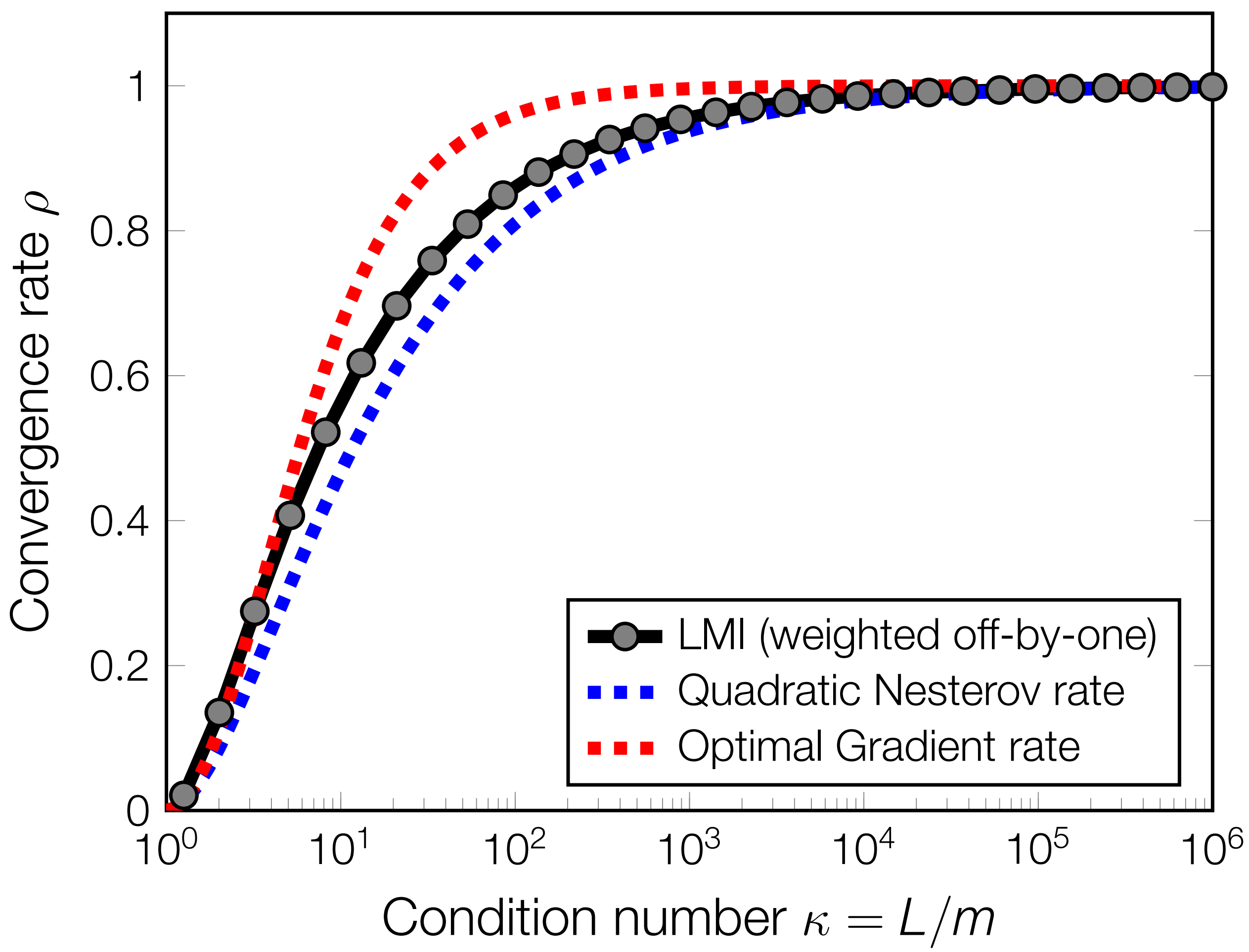
# Nesterov



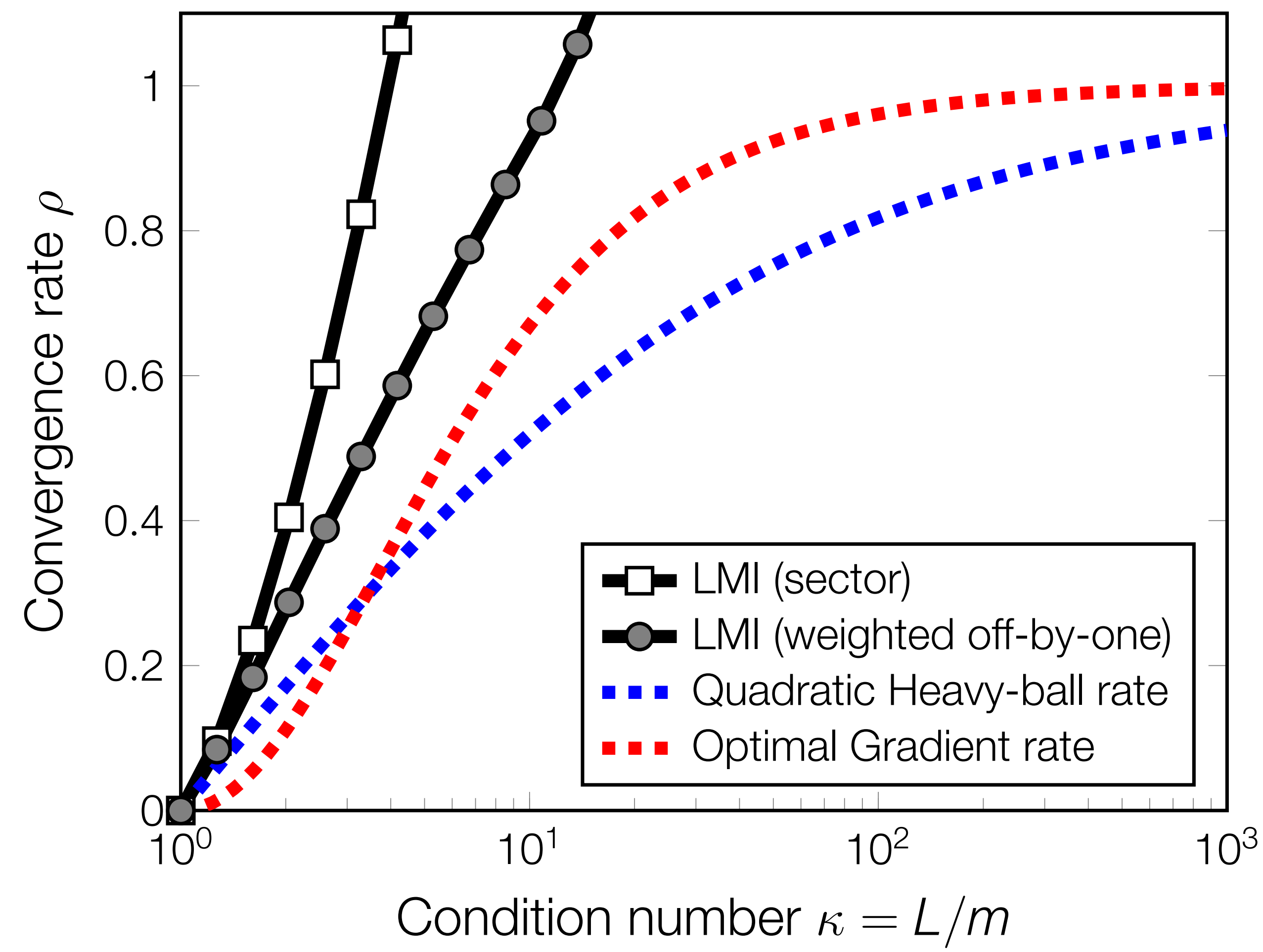

# Heavy Ball



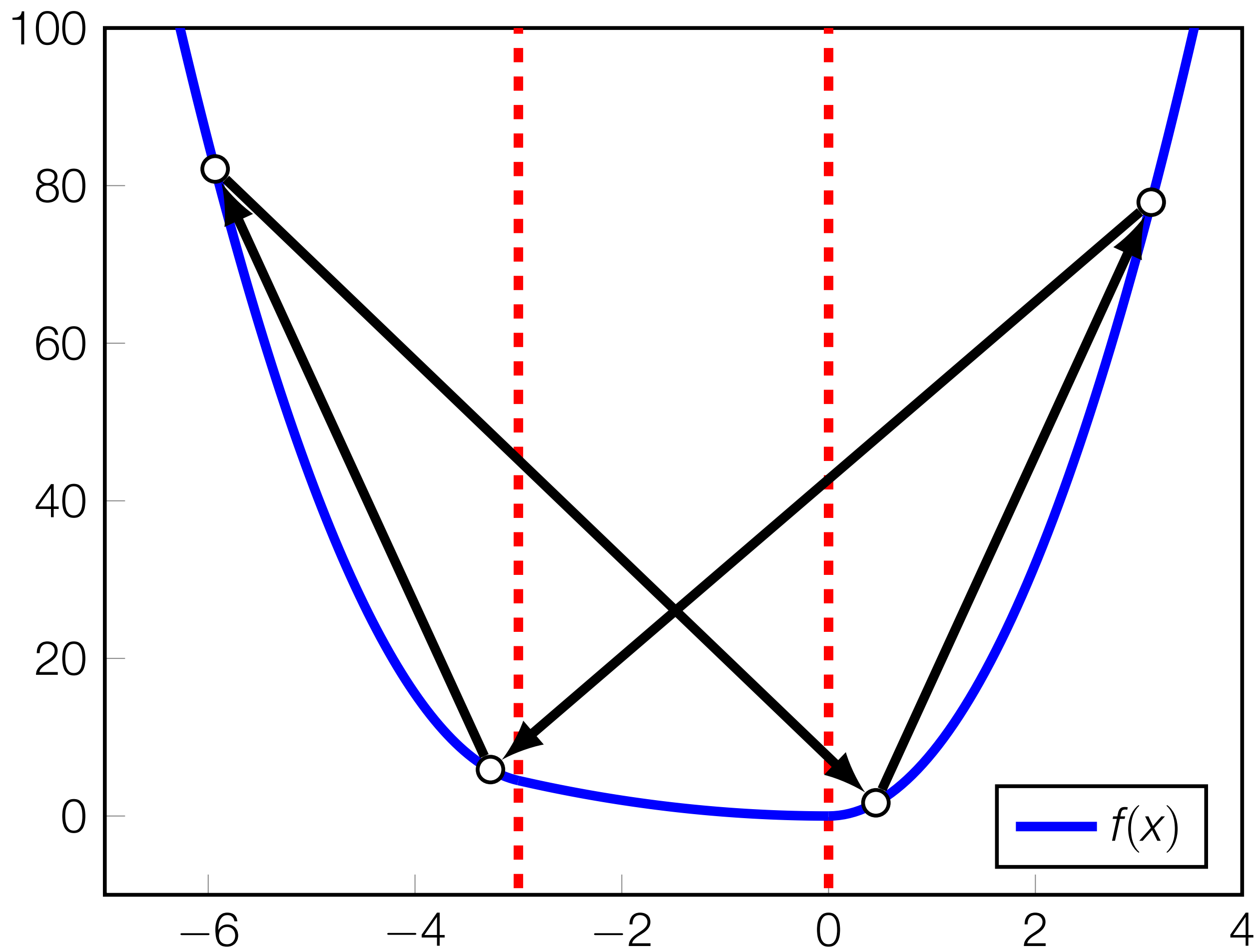
# Nesterov




# Heavy Ball



# Heavy Ball isn't stable




$$f(x) = \begin{cases} 16x^2 + 90x + 135 & x < -3 \\ x^2 & x \in [-3, 0] \\ 16x^2 & x \geq 0 \end{cases}$$

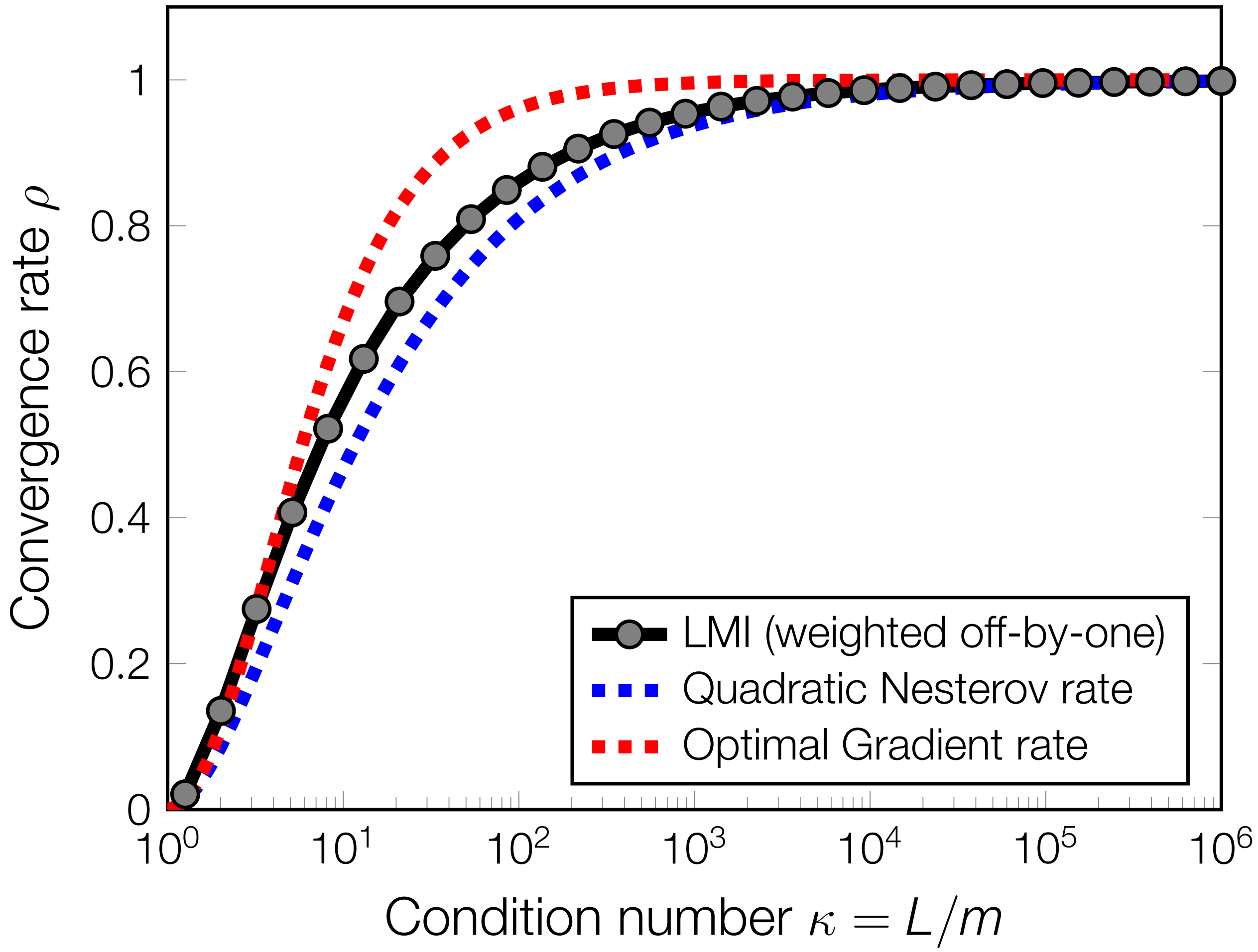
$m = 1$ $L = 16$

If you start at $x_0 \in [1.9, 2.4]$, Heavy Ball with standard parameters converges to the limit cycle.

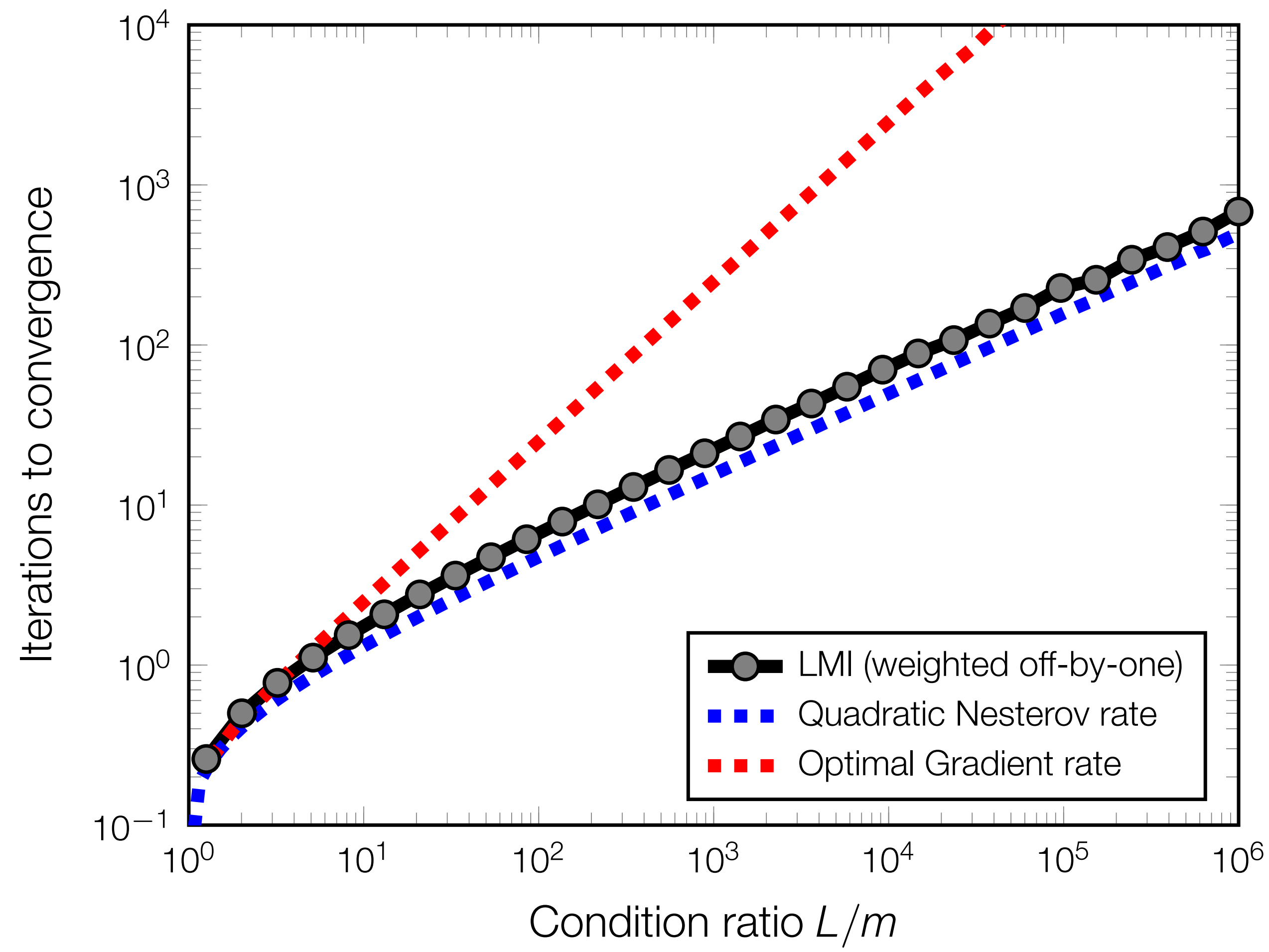
- *Aizerman's conjecture* [1949]. A linear system in feedback with a sector nonlinearity is stable if the linear system is stable for any linear gain of the sector.
- **THE AIZERMAN CONJECTURE IS FALSE** [Krasovskii 1952]
- This is a very simple counterexample.

# Nesterov

## Rate



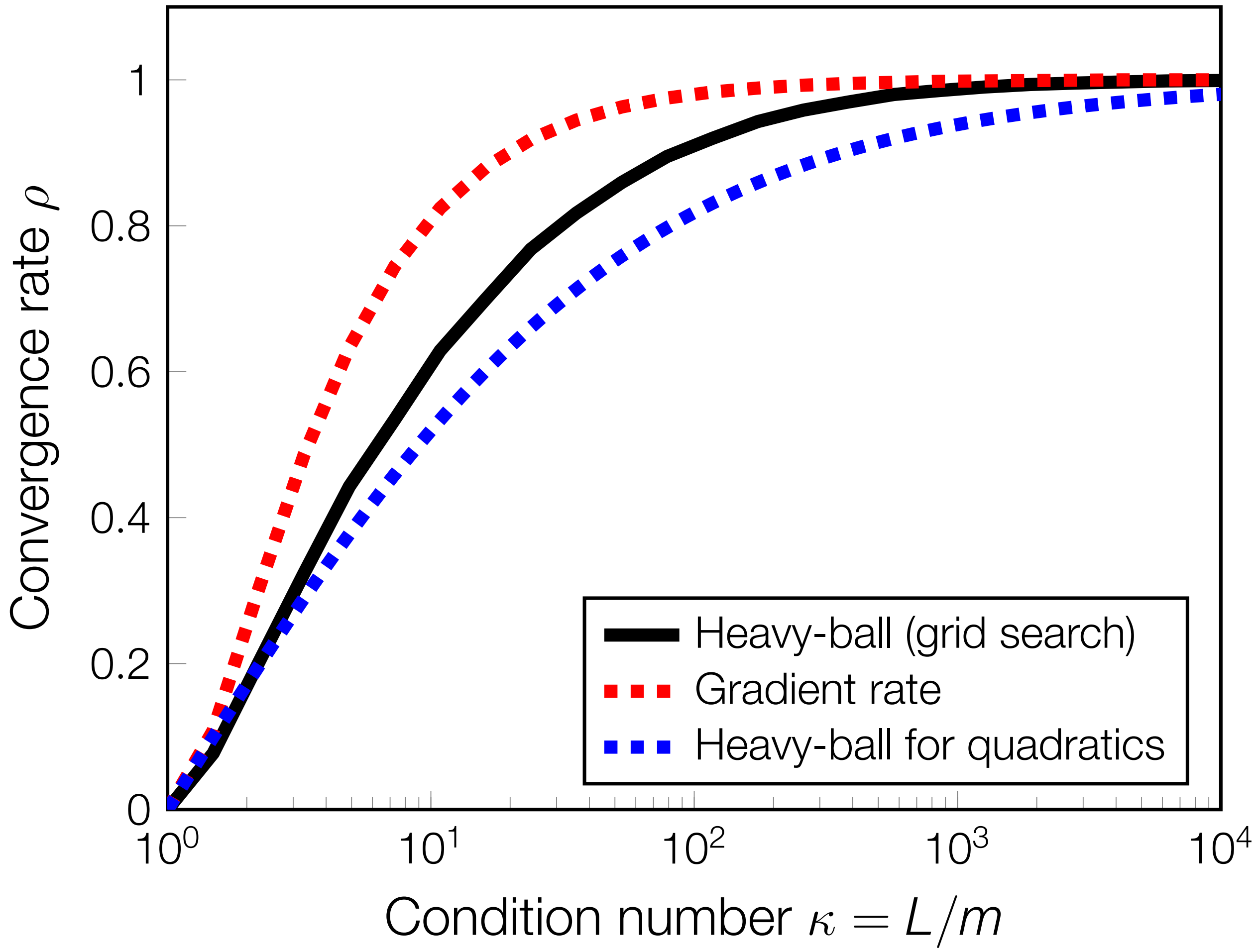

## Iterations (-log$^{-1}$ ρ)



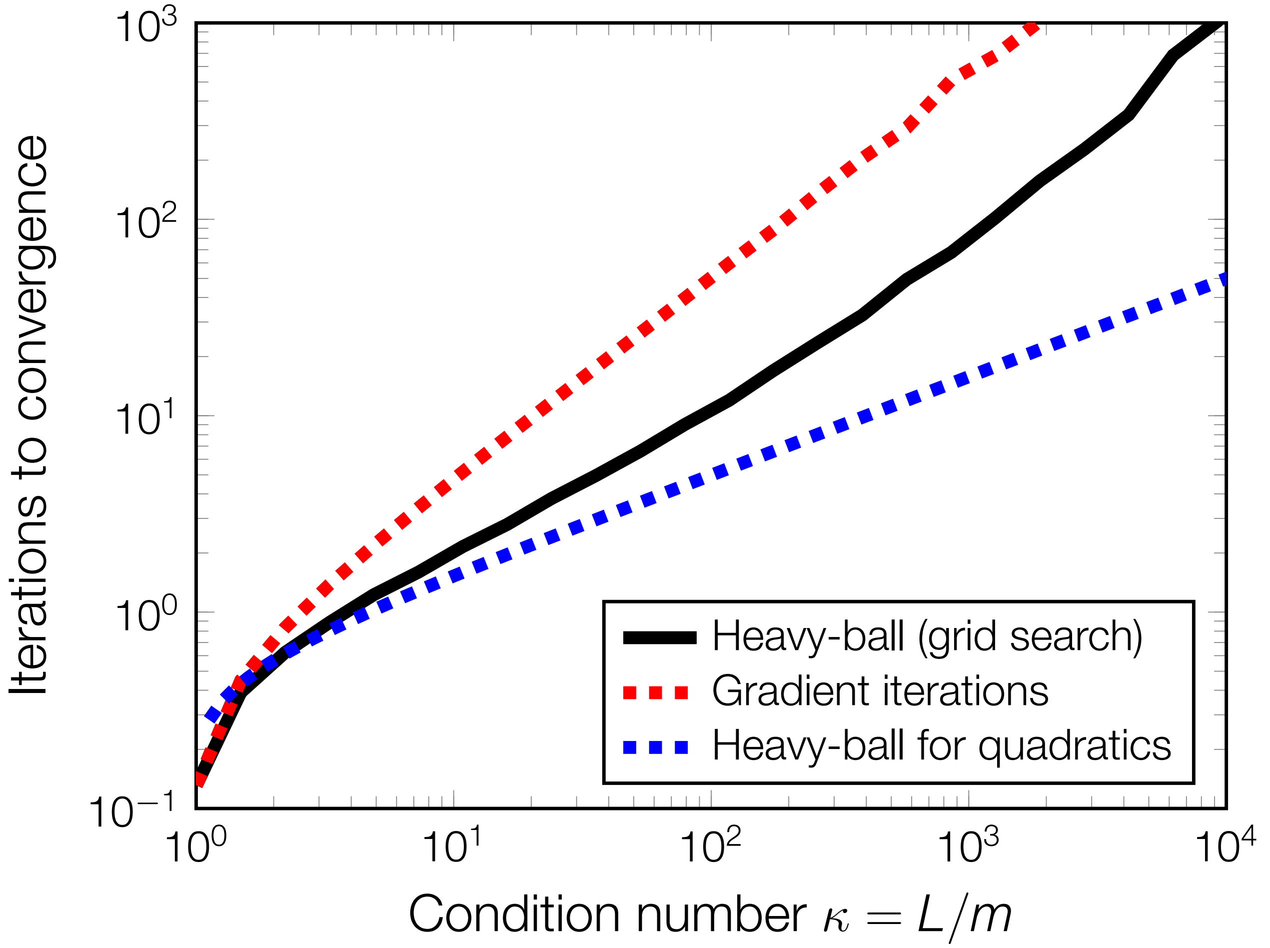

Iterations differ from the quadratic case by less than a factor of 2.

# Heavy-Ball

## Rate




## Iterations ($-\log^{-1} \rho$)




Fix $\alpha = 1/L$.

Grid search over $\beta$ to find minimal convergence rate $\rho$

# Integral Quadratic Constraints in Context

- Proposed by Megretski and Rantzer in 1996 (frequency domain)
- Generalizes the KYP Lemma/dissipativity theory
- Special case of S-Procedure/sum-of-squares hierarchy
- Drori and Teboulle 2013 used *all* quadratic constraints between time points to provide sharp analysis of gradient method for weakly convex functions.
- IQCs allow analysis which is dimension-free and certificates of size independent of the time horizon.

# Extensions

**Proximal/Projected methods**

Achieve same rate as unconstrained case via an LFT argument

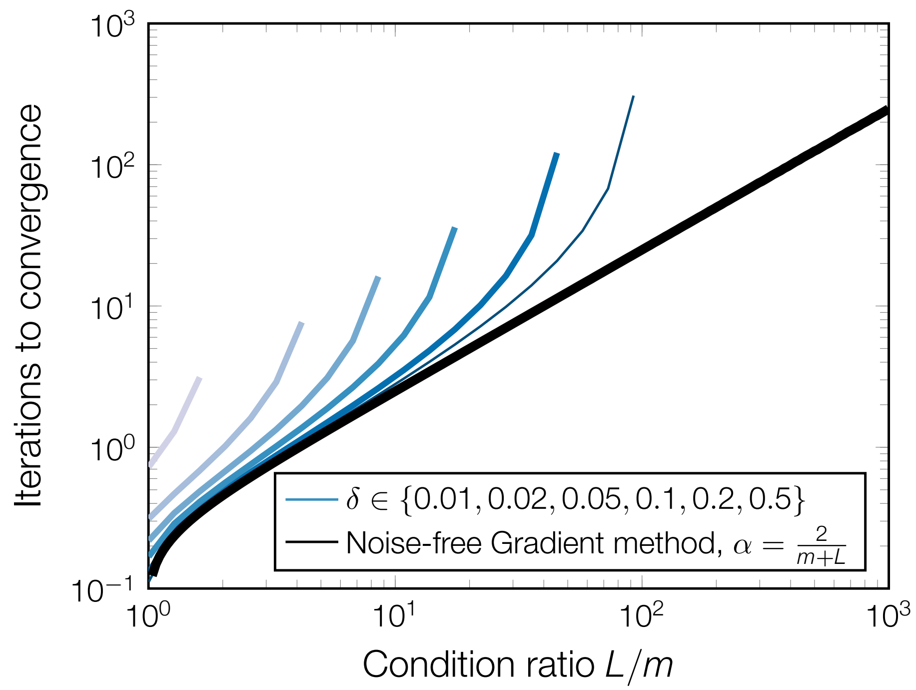
**Removing strong convexity**

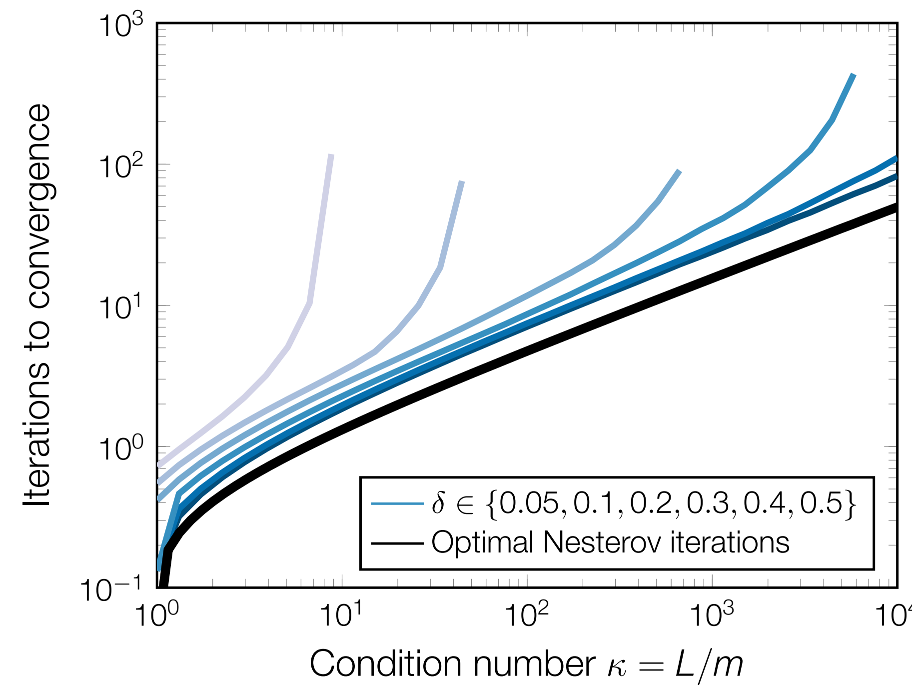
Achieve standard $\tilde{O}(\text{poly}(k^{-1}))$ rates by adding a regularization term

# Noisy Gradients

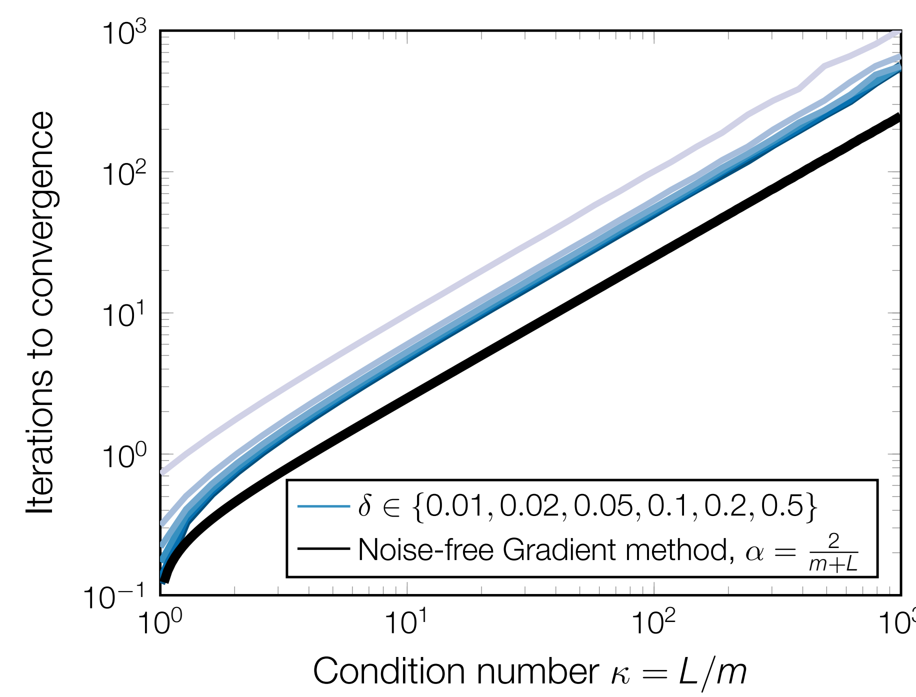
$u[k] = \nabla f(y[k]) + \omega[k]$

$\|\omega[k]\| \leq \delta \, \| \nabla f(y[k]) \|$




*Gradient method becomes robust when α=1/L*



# Synthesis (brutal forces)

- test *all* algorithms with two states
- parameterization in terms of $(\alpha,\beta_1,\beta_2)$:

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

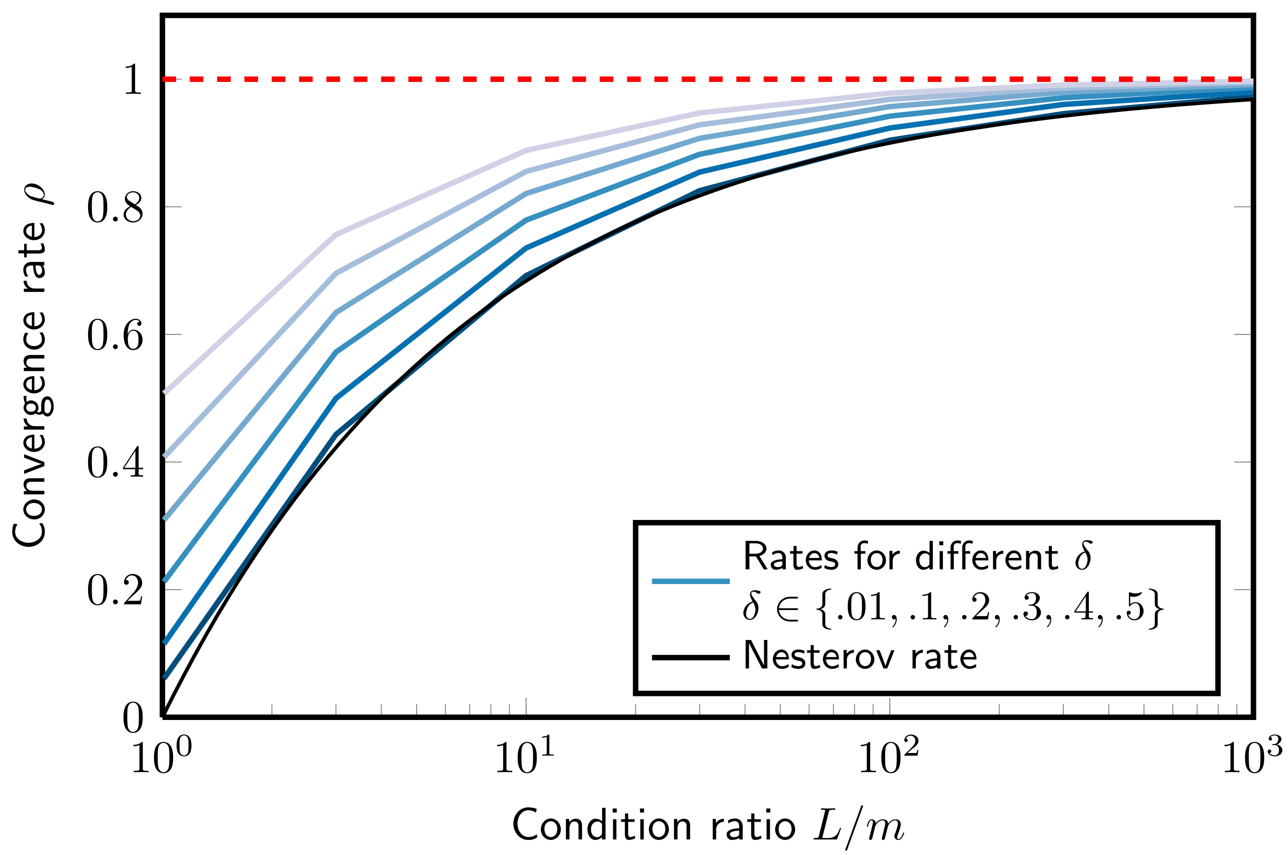
Special cases:

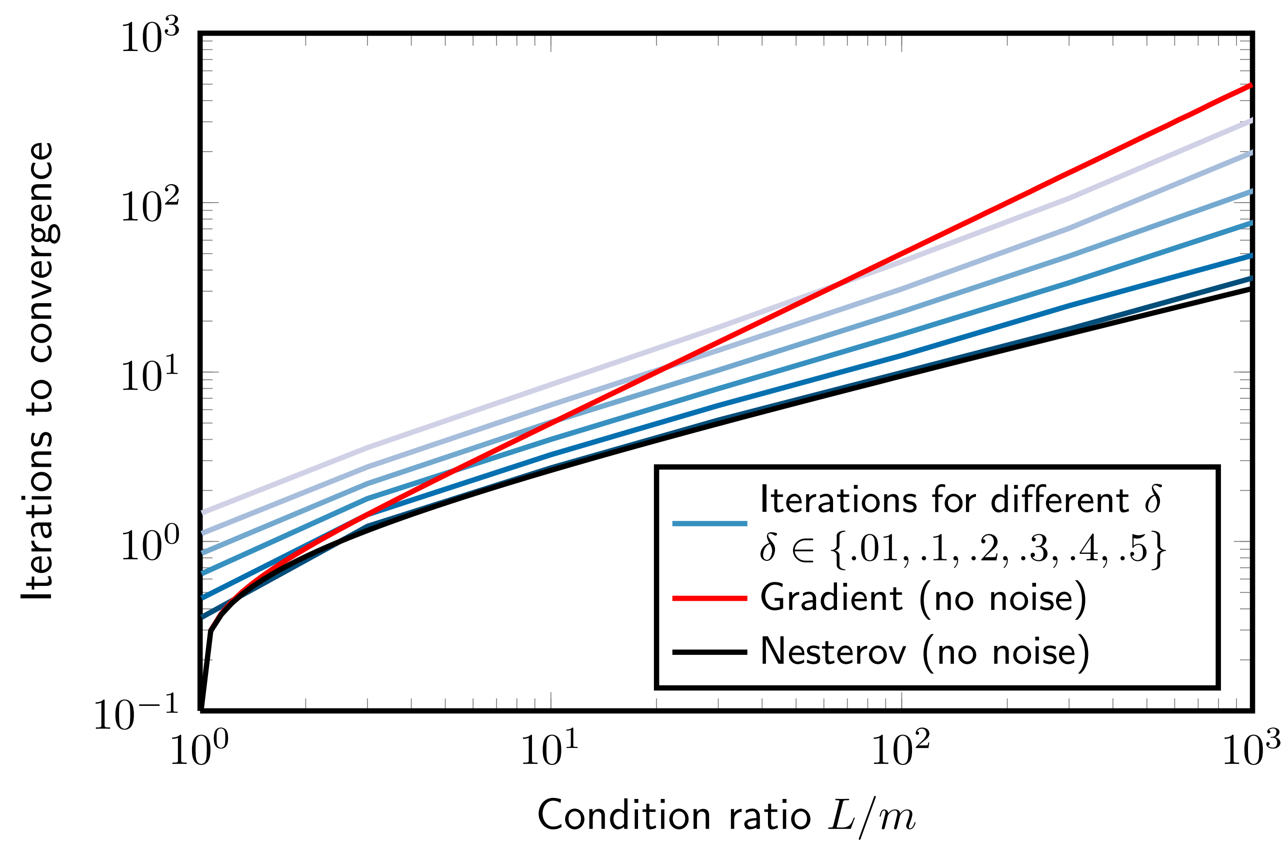
| | |
|---|---|
| gradient | $(\alpha,\beta 1,\beta 2)=(\alpha,0,0)$ |
| Heavy Ball | $(\alpha,\beta 1,\beta 2)=(\alpha,\beta,0)$ |
| Nesterov | $(\alpha,\beta 1,\beta 2)=(\alpha,\beta,\beta)$ |

# Synthesis (brutal forces)

- parameterization in terms of $(\alpha, \beta_1, \beta_2)$:

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




- Faster than the gradient method AND provably robust to noise.
- Suggests that more sophisticated algorithm design is possible.

# Conclusions

- IQCs provide a powerful proof system for algorithm analysis by replacing complicated nonlinearities with quadratic constraint sets.
- *Collects constraints about function classes, not algorithms*.
- New proofs of convergence for popular first-order methods.
- Enables numerical exploration of parameter spaces.
- Only beginning to get a sense of what IQCs can tell us about optimization schemes
- Many more control theory techniques that may provide new insight when applied to optimization and machine learning.

# Open Problems

- Improve the analysis for Nesterov’s method using refined IQCs
- An analytic proof of Nesterov’s method using IQCs
- Lower bounds using Zames-Falb IQCs and Megretski argument
- Integrating time varying plants. Is Nonlinear Conjugate Gradient actually stable?
- Is there a way to choose the stepsize using adaptive control techniques?
- New algorithm design via DK iterations and IQC-based nonlinear control synthesis.
- Stochastic coordinate descent and stochastic gradient descent via expected IQCs
- Subgaussian noise analysis via LQG and Ricatti equations
- Bringing the function value into the picture. The function itself is Lyapunov!
- Extending the library of IQCs.
- Automatically proving and deriving IQCs via sum-of-squares techniques
- Smaller function classes. With more structure, do we get better rates?
- Search for non-quadratic Lyapunov functions using IQC + SOS
- Analyzing really complicated interconnections for modular machine learning

# Acknowledgments


Laurent Lessard

Andrew Packard

- Many thanks to Elad Hazan, Ali Jadbabaie, Pablo Parrilo and Peter Seiler for many helpful pointers and discussions


- Preprint at arxiv.org/abs/1408.3595

Thanks!