Perspectives on the analysis and design of optimization algorithms

Adrien Taylor

Public set of slides – 2025


François Glineur


Julien Hendrickx


Etienne de Klerk


Ernest Ryu


Aymeric Dieuleveut


Pontus Giselsson


Francis Bach


Jérôme Bolte


Yoel Drori


Alexandre d'Aspremont


Pierre Gaillard

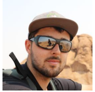

Bryan Van Scoy


Laurent Lessard


Sebastian Banert


Céline Moucer


Wouter Koolen


Baptiste Goujaud


Julien Weibel


Mathieu Barré


Radu Dragomir


Shuvomoy Das Gupta


Gauthier Gidel

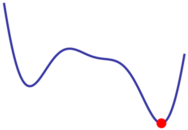

Eduard Gorbunov


Manu Upadhyaya

Context: numerical (continuous) optimization

Minimize $f : \mathbb{R}^d \to \mathbb{R}$ (e.g., with f continuous)

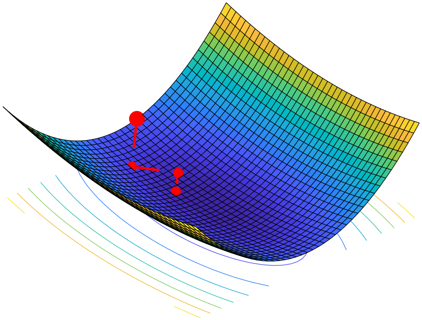
$$f(x_\star) \triangleq \min_{x \in \mathbb{R}^d} f(x).$$



Ubiquitous in applied mathematics and computer science.

Numerous applications for modeling (physics, economics), estimation (statistics, machine learning), decisions (control, operations research).

Usually solved via **iterative algorithm** generating sequence $x_0, x_1, \ldots, x_N$.



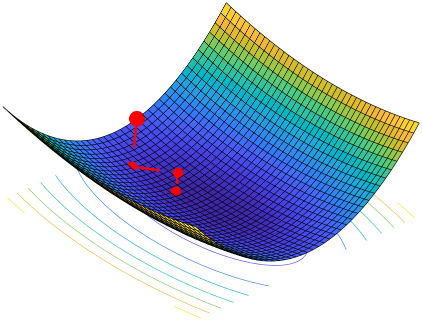
Gradient descent (stepsize α)

for $k = 0, 1, \ldots$ **do**

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

end for

Usually solved via **iterative algorithm** generating sequence $x_0, x_1, \ldots, x_N$.



Gradient descent (stepsize α)

for $k = 0, 1, \ldots$ **do**

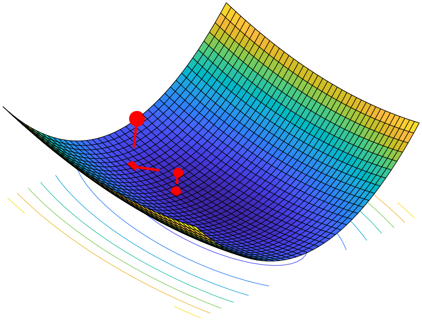
$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

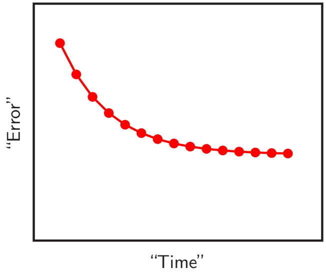
end for

What to expect from the output of the algorithm?

For instance: **bounds** on certain notions of "error": $f(x_k) - f(x_\star)$, $\|x_k - x_\star\|$, $\|\nabla f(x_k)\|$, etc.

Usually solved via **iterative algorithm** generating sequence $x_0, x_1, \ldots, x_N$.

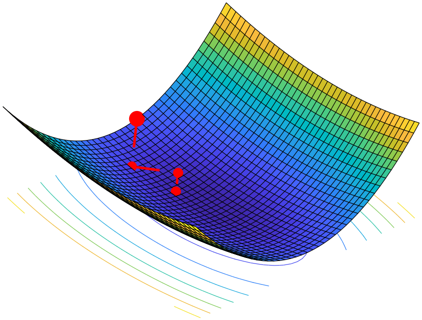


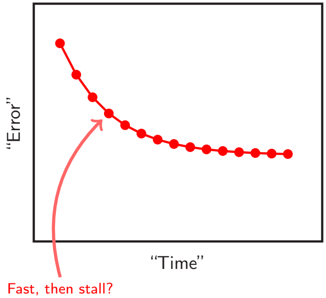


What to expect from the output of the algorithm?

For instance: **bounds** on certain notions of "error": $f(x_k) - f(x_\star)$, $\|x_k - x_\star\|$, $\|\nabla f(x_k)\|$, etc.

Usually solved via **iterative algorithm** generating sequence $x_0, x_1, \ldots, x_N$.

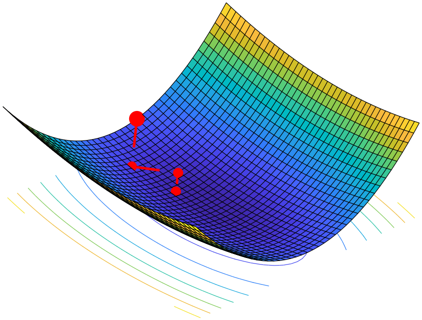


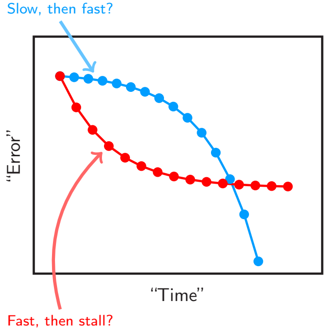


What to expect from the output of the algorithm?

For instance: **bounds** on certain notions of "error": $f(x_k) - f(x_\star)$, $\|x_k - x_\star\|$, $\|\nabla f(x_k)\|$, etc.

Usually solved via **iterative algorithm** generating sequence $x_0, x_1, \ldots, x_N$.

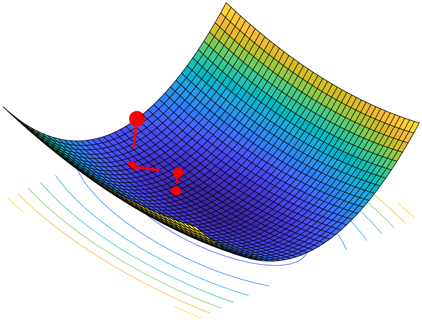


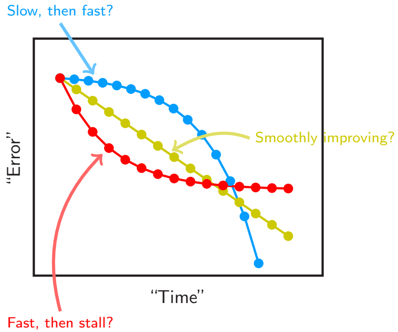


What to expect from the output of the algorithm?

For instance: **bounds** on certain notions of "error": $f(x_k) - f(x_\star)$, $\|x_k - x_\star\|$, $\|\nabla f(x_k)\|$, etc.

Usually solved via **iterative algorithm** generating sequence $x_0, x_1, \dots, x_N$.

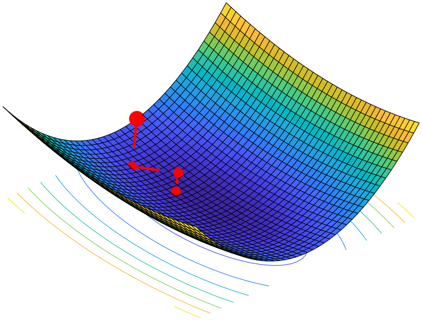


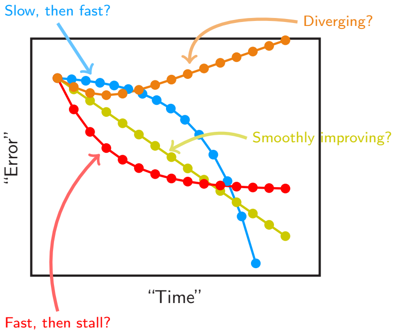


What to expect from the output of the algorithm?

For instance: **bounds** on certain notions of "error": $f(x_k) - f(x_\star)$, $\|x_k - x_\star\|$, $\|\nabla f(x_k)\|$, etc.

Usually solved via **iterative algorithm** generating sequence $x_0, x_1, \ldots, x_N$.





What to expect from the output of the algorithm?

For instance: **bounds** on certain notions of "error": $f(x_k) - f(x_\star)$, $\|x_k - x_\star\|$, $\|\nabla f(x_k)\|$, etc.

How to show that an algorithm works?

How to show that an algorithm works?

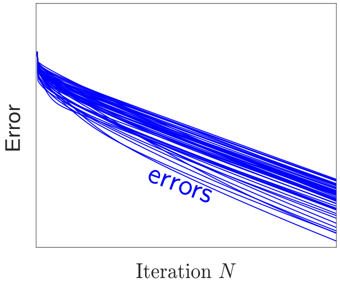
- $\diamond$ Assumptions (no free lunch).

How to show that an algorithm works?

- ◇ Assumptions (no free lunch).
- ◇ Here: worst-case perspective.

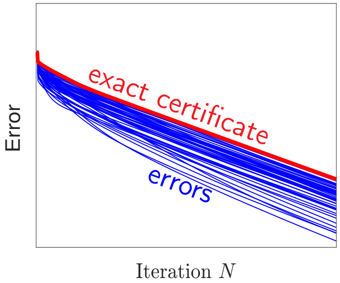
How to show that an algorithm works?

- ◇ Assumptions (no free lunch).
- ◇ Here: worst-case perspective.



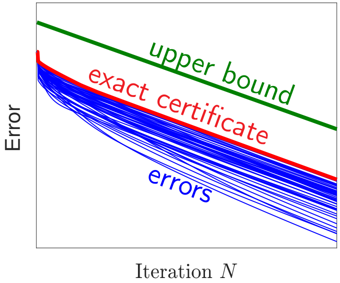
How to show that an algorithm works?

- ◇ Assumptions (no free lunch).
- ◇ Here: worst-case perspective.



How to show that an algorithm works?

- ◇ Assumptions (no free lunch).
- ◇ Here: worst-case perspective.



Constructive approach to performance analysis

Towards structured analyses

Towards optimal algorithms

Concluding remarks

Constructive approach to performance analysis

Towards structured analyses

Towards optimal algorithms

Concluding remarks

Example: analysis of a gradient method

Let $f : \mathbb{R}^d \to \mathbb{R}$ (continuously differentiable). Find $x_\star \in \mathbb{R}^d$ such that

$$f(x_\star) = \min_{x \in \mathbb{R}^d} f(x),$$

with f is L-smooth and μ-strongly convex ($f \in \mathcal{F}_{\mu,L}$).

Example: analysis of a gradient method

Let $f : \mathbb{R}^d \to \mathbb{R}$ (continuously differentiable). Find $x_\star \in \mathbb{R}^d$ such that

$$f(x_\star) = \min_{x \in \mathbb{R}^d} f(x),$$

with f is L-smooth and μ-strongly convex ($f \in \mathcal{F}_{\mu,L}$).

(Gradient method) We decide to use: $x_{k+1} = x_k - \alpha \nabla f(x_k)$

Example: analysis of a gradient method

Let $f : \mathbb{R}^d \to \mathbb{R}$ (continuously differentiable). Find $x_\star \in \mathbb{R}^d$ such that

$$f(x_\star) = \min_{x \in \mathbb{R}^d} f(x),$$

with f is L-smooth and μ-strongly convex ($f \in \mathcal{F}_{\mu,L}$).

(Gradient method) We decide to use: $x_{k+1} = x_k - \alpha \nabla f(x_k)$

Question: what *a priori* guarantees after N iterations?

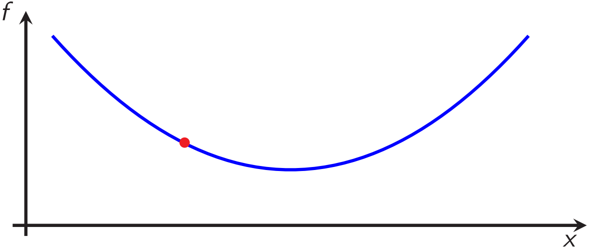
Examples: what about $f(x_N) - f(x_\star)$, $\|\nabla f(x_N)\|$, $\|x_N - x_\star\|$?

About the assumptions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$:

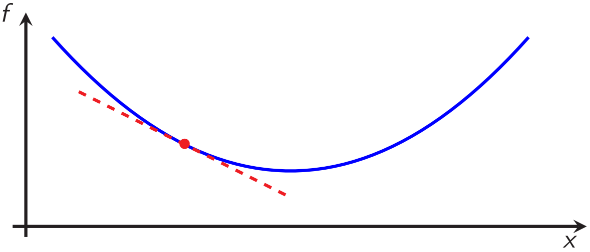
About the assumptions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$:



About the assumptions

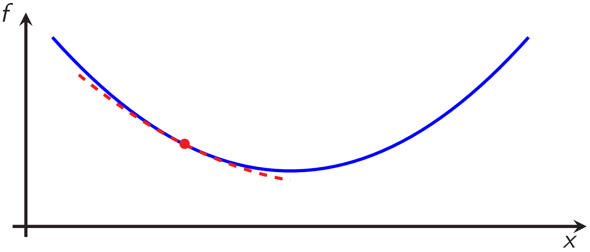
A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$:



(1) (Convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle$,

About the assumptions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$:

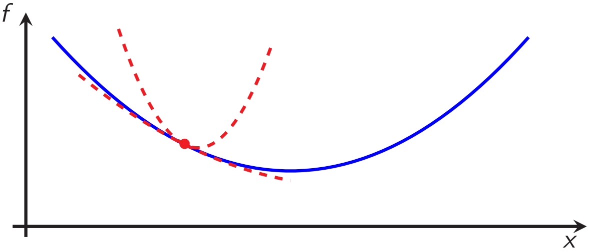


(1) (Convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle$,

(1b) (μ-strong convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2} \|x - y\|^2$,

About the assumptions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$:



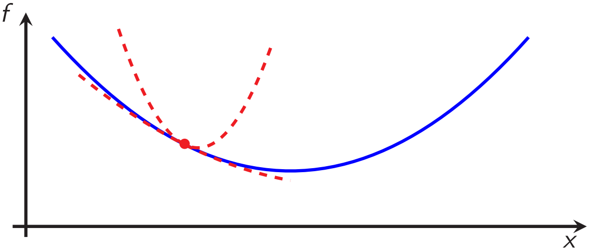
(1) (Convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle,$

(1b) (μ-strong convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2}\|x - y\|^2,$

(2) (L-smoothness) $f(x) \leqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|^2,$

About the assumptions

A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$:



$$(1) \quad \text{(Convexity)}\ f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle,$$

$$(1b) \quad (\mu\text{-strong convexity})\ f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2} \|x - y\|^2,$$

$$(2) \quad \text{(L-smoothness)}\ f(x) \leqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|^2,$$

$$(1\&2) \quad \langle \nabla f(x) - \nabla f(y); x - y \rangle \geqslant \frac{1}{L+\mu} \|\nabla f(x) - \nabla f(y)\|^2 + \frac{\mu L}{L+\mu} \|x - y\|^2.$$

Convergence rate of a gradient step

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau\|x_0 - x_\star\|^2,$$

for all

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

$$\|x_1 - x_\star\|^2$$

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\, f(x)$?

$$\|x_1 - x_\star\|^2 = \|x_0 - x_\star\|^2 - 2\alpha\langle \nabla f(x_0); x_0 - x_\star\rangle + \alpha^2 \|\nabla f(x_0)\|^2$$

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

$$\|x_1 - x_\star\|^2 = \|x_0 - x_\star\|^2 - 2\alpha \langle \nabla f(x_0); x_0 - x_\star \rangle + \alpha^2 \|\nabla f(x_0)\|^2$$

$$\Big\downarrow \ \text{Inequality (1\&2)}$$

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

$$\|x_1 - x_\star\|^2 = \|x_0 - x_\star\|^2 - 2\alpha \langle \nabla f(x_0); x_0 - x_\star \rangle + \alpha^2 \|\nabla f(x_0)\|^2$$

$$\Big\downarrow \text{ Inequality (1\&2)}$$

$$\leqslant \left(1 - \tfrac{2\alpha L \mu}{L+\mu}\right) \|x_0 - x_\star\|^2 + \alpha \left(\alpha - \tfrac{2}{L+\mu}\right) \|\nabla f(x_0)\|^2$$

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

$$\|x_1 - x_\star\|^2 = \|x_0 - x_\star\|^2 - 2\alpha \langle \nabla f(x_0); x_0 - x_\star \rangle + \alpha^2 \|\nabla f(x_0)\|^2$$

$$\Big\downarrow \text{ Inequality (1\&2)}$$

$$\leqslant \left(1 - \frac{2\alpha L \mu}{L+\mu}\right) \|x_0 - x_\star\|^2 + \alpha \left(\alpha - \frac{2}{L+\mu}\right) \|\nabla f(x_0)\|^2$$

$$\Big\downarrow \text{ if } 0 \leqslant \alpha \leqslant \frac{2}{L+\mu}$$

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

$$\|x_1 - x_\star\|^2 = \|x_0 - x_\star\|^2 - 2\alpha\langle \nabla f(x_0); x_0 - x_\star\rangle + \alpha^2 \|\nabla f(x_0)\|^2$$

$$\Big\downarrow \text{ Inequality (1\&2)}$$

$$\leqslant \left(1 - \tfrac{2\alpha L\mu}{L+\mu}\right)\|x_0 - x_\star\|^2 + \alpha\left(\alpha - \tfrac{2}{L+\mu}\right)\|\nabla f(x_0)\|^2$$

$$\Big\downarrow \text{ if } 0 \leqslant \alpha \leqslant \tfrac{2}{L+\mu}$$

$$\leqslant (1 - \alpha\mu)^2 \|x_0 - x_\star\|^2.$$

Legitimate questions about performance analyses?

Legitimate questions (gradient descent, one iteration):

- anything improvable? Realistic analyses?
- How to choose the right inequalities to combine?
- Why studying this specific quantity? Possible to adapt to other quantities?
- Unique way to arrive to the desired result?
- How likely are we to find such proofs in more complicated cases?

Legitimate questions about performance analyses?

Lemma 3. *Assume that the function is L-smooth and μ strongly-convex and satisfies the strong-growth condition in Equation 10. Then, using the updates in Equation 3-5 and setting the parameters according to Equations 11-15, if $\eta \leq \frac{1}{\rho L}$, then the following relation holds:*

$$b_{k+1}^2 \gamma_k^2 \left[\mathbb{E} f(w_{k+1}) - f^*\right] \leq \frac{a_0^2}{\rho\eta}\left[f(x_0) - f^*\right] + \frac{b_0^2}{2\rho\eta}\|x_0 - w^*\|^2 + \frac{\sigma^2\eta}{\rho}\sum_{i=0}^{k}[\gamma_i^2 b_{i+1}^2]$$

Proof.

Let $r_{k+1} = \|v_{k+1} - w^*\|$, then using equation 5

$$r_{k+1}^2 = \|\beta_k v_k + (1-\beta_k)\zeta_k - w^* - \gamma_k \eta \nabla f(\zeta_k, z_k)\|^2$$
$$r_{k+1}^2 = \|\beta_k v_k + (1-\beta_k)\zeta_k - w^*\|^2 + \gamma_k^2\eta^2\|\nabla f(\zeta_k, z_k)\|^2 + 2\gamma_k\eta\langle w^* - \beta_k v_k - (1-\beta_k)\zeta_k, \nabla f(\zeta_k, z_k)\rangle$$

Taking expecation wrt to z_k,

$$\mathbb{E}[r_{k+1}^2] = \mathbb{E}[\|\beta_k v_k + (1-\beta_k)\zeta_k - w^*\|^2] + \gamma_k^2\eta^2\mathbb{E}\|\nabla f(\zeta_k, z_k)\|^2 + 2\gamma_k\eta\left[\mathbb{E}\langle w^* - \beta_k v_k - (1-\beta_k)\zeta_k, \nabla f(\zeta_k, z_k)\rangle\right]$$
$$\leq \|\beta_k v_k + (1-\beta_k)\zeta_k - w^*\|^2 + \gamma_k^2\eta^2\rho\|\nabla f(\zeta_k)\|^2 + 2\gamma_k\eta\left[\langle w^* - \beta_k v_k - (1-\beta_k)\zeta_k, \nabla f(\zeta_k)\rangle\right] + \gamma_k^2\eta^2\sigma^2$$
$$= \|\beta_k(v_k - w^*) + (1-\beta_k)(\zeta_k - w^*)\|^2 + \gamma_k^2\eta^2\rho\|\nabla f(\zeta_k)\|^2 + 2\gamma_k\eta\left[\langle w^* - \beta_k v_k - (1-\beta_k)\zeta_k, \nabla f(\zeta_k)\rangle\right] + \gamma_k^2\eta^2\sigma^2$$
$$\leq \beta_k\|v_k - w^*\|^2 + (1-\beta_k)\|\zeta_k - w^*\|^2 + \gamma_k^2\eta^2\rho\|\nabla f(\zeta_k)\|^2 + 2\gamma_k\eta\left[\langle w^* - \beta_k v_k - (1-\beta_k)\zeta_k, \nabla f(\zeta_k)\rangle\right] + \gamma_k^2\eta^2\sigma^2$$

(By convexity of $\|\cdot\|^2$)

$$= \beta_k r_k^2 + (1-\beta_k)\|\zeta_k - w^*\|^2 + \gamma_k^2\eta^2\rho\|\nabla f(\zeta_k)\|^2 + 2\gamma_k\eta\left[\langle w^* - \beta_k v_k - (1-\beta_k)\zeta_k, \nabla f(\zeta_k)\rangle\right] + \gamma_k^2\eta^2\sigma^2$$
$$= \beta_k r_k^2 + (1-\beta_k)\|\zeta_k - w^*\|^2 + \gamma_k^2\eta^2\rho\|\nabla f(\zeta_k)\|^2 + 2\gamma_k\eta\left[\langle \beta_k(\zeta_k - v_k) + w^* - \zeta_k, \nabla f(\zeta_k)\rangle\right] + \gamma_k^2\eta^2\sigma^2$$
$$= \beta_k r_k^2 + (1-\beta_k)\|\zeta_k - w^*\|^2 + \gamma_k^2\eta^2\rho\|\nabla f(\zeta_k)\|^2 + 2\gamma_k\eta\left[\langle \frac{\beta_k(1-\alpha_k)}{\alpha_k}(w_k - \zeta_k) + w^* - \zeta_k, \nabla f(\zeta_k)\rangle\right] + \gamma_k^2\eta^2\sigma^2$$

(From equation 4)

$$= \beta_k r_k^2 + (1-\beta_k)\|\zeta_k - w^*\|^2 + \gamma_k^2\eta^2\rho\|\nabla f(\zeta_k)\|^2 + 2\gamma_k\eta\left[\frac{\beta_k(1-\alpha_k)}{\alpha_k}\langle \nabla f(\zeta_k), (w_k - \zeta_k)\rangle + \langle \nabla f(\zeta_k), w^* - \zeta_k\rangle\right] + \gamma_k^2\eta^2\sigma^2$$
$$\leq \beta_k r_k^2 + (1-\beta_k)\|\zeta_k - w^*\|^2 + \gamma_k^2\eta^2\rho\|\nabla f(\zeta_k)\|^2 + 2\gamma_k\eta\left[\frac{\beta_k(1-\alpha_k)}{\alpha_k}(f(w_k) - f(\zeta_k)) + \langle \nabla f(\zeta_k), w^* - \zeta_k\rangle\right] + \gamma_k^2\eta^2\sigma^2$$

(By convexity)

By strong convexity,

$$\mathbb{E}[r_{k+1}^2] \leq \beta_k r_k^2 + (1-\beta_k)\|\zeta_k - w^*\|^2 + \gamma_k^2\eta^2\rho\|\nabla f(\zeta_k)\|^2$$
$$+ 2\gamma_k\eta\left[\frac{\beta_k(1-\alpha_k)}{\alpha_k}(f(w_k) - f(\zeta_k)) + f^* - f(\zeta_k) - \frac{\mu}{2}\|\zeta_k - w^*\|^2\right] + \gamma_k^2\eta^2\sigma^2 \quad (16)$$

By Lipschitz continuity of the gradient,

$$f(w_{k+1}) - f(\zeta_k) \leq \langle \nabla f(\zeta_k), w_{k+1} - \zeta_k\rangle + \frac{L}{2}\|w_{k+1} - \zeta_k\|^2$$
$$\leq -\eta\langle \nabla f(\zeta_k), \nabla f(\zeta_k, z_k)\rangle + \frac{L\eta^2}{2}\|\nabla f(\zeta_k, z_k)\|^2$$

Taking expectation wrt z_k and using equations 9, 10

$$\mathbb{E}[f(w_{k+1}) - f(\zeta_k)] \leq -\eta\|\nabla f(\zeta_k)\|^2 + \frac{L\rho\eta^2}{2}\|\nabla f(\zeta_k)\|^2 + \frac{L\eta^2\sigma^2}{2}$$
$$\mathbb{E}[f(w_{k+1}) - f(\zeta_k)] \leq \left[-\eta + \frac{L\rho\eta^2}{2}\right]\|\nabla f(\zeta_k)\|^2 + \frac{L\eta^2\sigma^2}{2}$$

If $\eta \leq \frac{1}{\rho L}$,

$$\mathbb{E}[f(w_{k+1}) - f(\zeta_k)] \leq \left(\frac{-\eta}{2}\right)\|\nabla f(\zeta_k)\|^2 + \frac{L\eta^2\sigma^2}{2}$$
$$\implies \|\nabla f(\zeta_k)\|^2 \leq \left(\frac{2}{\eta}\right)\mathbb{E}[f(\zeta_k) - f(w_{k+1})] + L\eta\sigma^2 \quad (17)$$

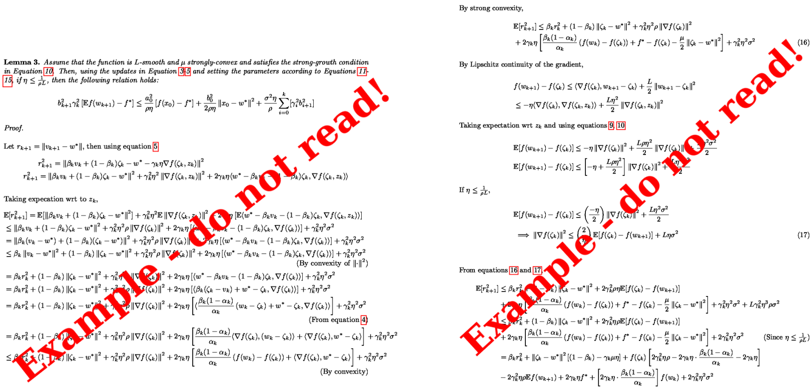
From equations 16 and 17

$$\mathbb{E}[r_{k+1}^2] \leq \beta_k r_k^2 + (1-\beta_k)\|\zeta_k - w^*\|^2 + 2\gamma_k^2\rho\eta\mathbb{E}[f(\zeta_k) - f(w_{k+1})]$$
$$+ 2\gamma_k\eta\left[\frac{\beta_k(1-\alpha_k)}{\alpha_k}(f(w_k) - f(\zeta_k)) + f^* - f(\zeta_k) - \frac{\mu}{2}\|\zeta_k - w^*\|^2\right] + \gamma_k^2\eta^2\sigma^2 + L\gamma_k^2\eta^3\rho\sigma^2$$
$$\leq \beta_k r_k^2 + (1-\beta_k)\|\zeta_k - w^*\|^2 + 2\gamma_k^2\eta\rho\mathbb{E}[f(\zeta_k) - f(w_{k+1})]$$
$$+ 2\gamma_k\eta\left[\frac{\beta_k(1-\alpha_k)}{\alpha_k}(f(w_k) - f(\zeta_k)) + f^* - f(\zeta_k) - \frac{\mu}{2}\|\zeta_k - w^*\|^2\right] + 2\gamma_k^2\eta^2\sigma^2 \quad (\text{Since } \eta \leq \tfrac{1}{\rho L})$$
$$= \beta_k r_k^2 + \|\zeta_k - w^*\|^2[(1-\beta_k) - \gamma_k\mu\eta] + f(\zeta_k)\left[2\gamma_k^2\eta\rho - 2\gamma_k\eta\cdot\frac{\beta_k(1-\alpha_k)}{\alpha_k} - 2\gamma_k\eta\right]$$
$$- 2\gamma_k^2\eta\rho\mathbb{E}f(w_{k+1}) + 2\gamma_k\eta f^* + \left[2\gamma_k\eta\cdot\frac{\beta_k(1-\alpha_k)}{\alpha_k}\right]f(w_k) + 2\gamma_k^2\eta^2\sigma^2$$

Example - do not read!

Example - do not read!

Legitimate questions about performance analyses?



- ⚠ Error-prone
- ⚠ Technical, lack global insights.
- ⚠ Few proof patterns.

- ? Easily fixable?
- ? Simple to adapt to variations of target inequality?
- ? Simple to adapt to algorithmic variations?

Convergence rate of a gradient step

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

Computing τ?[1]

[1] Original idea from [Drori and Teboulle, 2014]. Developments here from [T, Hendrickx, Glineur, 2017].

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \operatorname*{argmin}_x f(x)$?

Computing τ?[1]

$$\tau = \max_{f, x_0, x_1, x_\star} \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

$$\text{s.t. } f \in \mathcal{F}_{\mu,L} \qquad \text{Functional class}$$

[1] Original idea from [Drori and Teboulle, 2014]. Developments here from [T, Hendrickx, Glineur, 2017].

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

Computing τ?[1]

$$\begin{aligned} \tau = \max_{f, x_0, x_1, x_\star} \ & \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2} & \\ \text{s.t. } & f \in \mathcal{F}_{\mu,L} & \text{Functional class} \\ & x_1 = x_0 - \alpha \nabla f(x_0) & \text{Algorithm} \\ & \nabla f(x_\star) = 0 & \text{Optimality of } x_\star \end{aligned}$$

[1] Original idea from [Drori and Teboulle, 2014]. Developments here from [T, Hendrickx, Glineur, 2017].

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- $d \in \mathbb{N}$, L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \alpha \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

Computing τ?[1]

$$\begin{aligned} \tau = \max_{f, x_0, x_1, x_\star} \ & \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2} & \\ \text{s.t. } \ & f \in \mathcal{F}_{\mu,L} & \text{Functional class} \\ & x_1 = x_0 - \alpha \nabla f(x_0) & \text{Algorithm} \\ & \nabla f(x_\star) = 0 & \text{Optimality of } x_\star \end{aligned}$$

Variables: f, x_0, x_1, $x_\star$.

Parameters: μ, L, α.

[1] Original idea from [Drori and Teboulle, 2014]. Developments here from [T, Hendrickx, Glineur, 2017].

Sampled version

Sampled version

- Performance estimation problem :

$$\max_{f, x_0, x_1, x_\star} \quad \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$
$$\text{subject to} \quad f \text{ is } L\text{-smooth and } \mu\text{-strongly convex,}$$
$$x_1 = x_0 - \alpha \nabla f(x_0)$$
$$\nabla f(x_\star) = 0.$$

Sampled version

- Performance estimation problem **(Variables: f, x_0, x_1, $x_\star$)**:

$$
\begin{aligned}
\max_{f, x_0, x_1, x_\star} \quad & \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2} \\
\text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \\
& x_1 = x_0 - \alpha \nabla f(x_0) \\
& \nabla f(x_\star) = 0.
\end{aligned}
$$

Sampled version

- Performance estimation problem **(Variables: f, x_0, x_1, $x_\star$)**:

$$\max_{f, x_0, x_1, x_\star} \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

$$\text{subject to} \quad f \text{ is } L\text{-smooth and } \mu\text{-strongly convex,}$$

$$x_1 = x_0 - \alpha \nabla f(x_0)$$

$$\nabla f(x_\star) = 0.$$

- Sampled version:

Sampled version

◇ Performance estimation problem **(Variables: f, x_0, x_1, $x_\star$)**:

$$\begin{aligned} \max_{f, x_0, x_1, x_\star} \quad & \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \\ & x_1 = x_0 - \alpha \nabla f(x_0) \\ & \nabla f(x_\star) = 0. \end{aligned}$$

◇ Sampled version:

$$\begin{aligned} \max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad & \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases} \\ & x_1 = x_0 - \alpha g_0 \\ & g_\star = 0. \end{aligned}$$

Sampled version

- Performance estimation problem **(Variables: f, x_0, x_1, $x_\star$)**:

$$\max_{f, x_0, x_1, x_\star} \quad \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

$$\begin{aligned} \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \\ & x_1 = x_0 - \alpha \nabla f(x_0) \\ & \nabla f(x_\star) = 0. \end{aligned}$$

- Sampled version: **(Variables: x_0, x_1, $x_\star$, g_0, $g_\star$, f_0, $f_\star$)**

$$\max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

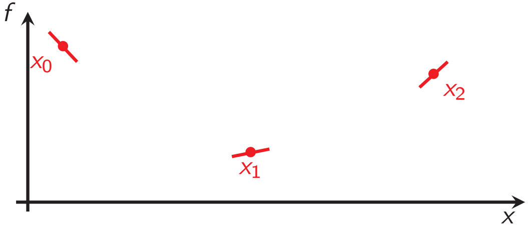
$$\begin{aligned} \text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases} \\ & x_1 = x_0 - \alpha g_0 \\ & g_\star = 0. \end{aligned}$$

Smooth strongly convex interpolation (or extension)

Let I index set, and associated $\{(x_i, g_i, f_i)\}_{i\in I}$: points x_i, (sub)gradients g_i and function values f_i.

Smooth strongly convex interpolation (or extension)

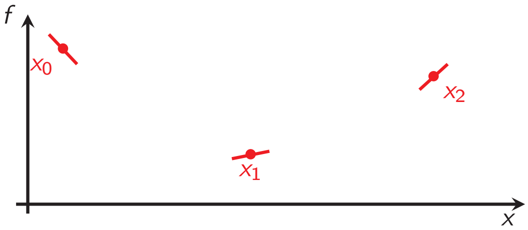
Let I index set, and associated $\{(x_i, g_i, f_i)\}_{i\in I}$: points x_i, (sub)gradients g_i and function values f_i.



? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that $f(x_i) = f_i$, and $g_i = \nabla f(x_i)$ $\forall i \in I$?

Smooth strongly convex interpolation (or extension)

Let I index set, and associated $\{(x_i, g_i, f_i)\}_{i\in I}$: points x_i, (sub)gradients g_i and function values f_i.



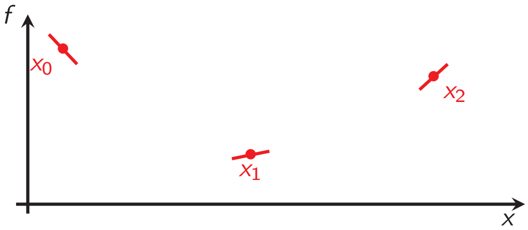
? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that $f(x_i) = f_i$, and $g_i = \nabla f(x_i)\ \forall i \in I$?

- Necessary and sufficient condition: $\forall i, j \in I$

$$f_i \geqslant f_j + \langle g_j, x_i - x_j \rangle + \frac{1}{2L}\|g_i - g_j\|^2 + \frac{\mu}{2(1-\mu/L)}\left\|x_i - x_j - \frac{1}{L}(g_i - g_j)\right\|^2.$$

Smooth strongly convex interpolation (or extension)

Let I index set, and associated $\{(x_i, g_i, f_i)\}_{i\in I}$: points x_i, (sub)gradients g_i and function values f_i.



? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that $f(x_i) = f_i$, and $g_i = \nabla f(x_i)\ \forall i \in I$?

- Necessary and sufficient condition: $\forall i, j \in I$

$$f_i \geqslant f_j + \langle g_j, x_i - x_j\rangle + \tfrac{1}{2L}\|g_i - g_j\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_i - x_j - \tfrac{1}{L}(g_i - g_j)\right\|^2.$$

- Simpler example: pick $\mu = 0$ and $L = \infty$ (just convexity):

$$f_i \geqslant f_j + \langle g_j, x_i - x_j\rangle.$$

Replace constraints

Replace constraints

- Interpolation conditions allow removing red constraints

$$\max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

$$\text{subject to} \quad \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases}$$

$$x_1 = x_0 - \alpha g_0$$

$$g_\star = 0,$$

Replace constraints

- Interpolation conditions allow removing red constraints

$$\max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

$$\text{subject to} \quad \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases}$$

$$x_1 = x_0 - \alpha g_0$$

$$g_\star = 0,$$

- replacing them by

$$f_\star \geqslant f_0 + \langle g_0, x_\star - x_0 \rangle + \tfrac{1}{2L}\|g_\star - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_\star - x_0 - \tfrac{1}{L}(g_\star - g_0)\right\|^2$$

$$f_0 \geqslant f_\star + \langle g_\star, x_0 - x_\star \rangle + \tfrac{1}{2L}\|g_0 - g_\star\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}(g_0 - g_\star)\right\|^2.$$

Replace constraints

- Interpolation conditions allow removing red constraints

$$\max_{\substack{x_0,x_1,x_\star\\ g_0,g_\star\\ f_0,f_\star}} \quad \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

$$\text{subject to} \quad \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0,\star \\ g_i = \nabla f(x_i) & i = 0,\star \end{cases}$$

$$x_1 = x_0 - \alpha g_0$$

$$g_\star = 0,$$

- replacing them by

$$f_\star \geqslant f_0 + \langle g_0, x_\star - x_0\rangle + \tfrac{1}{2L}\|g_\star - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_\star - x_0 - \tfrac{1}{L}(g_\star - g_0)\right\|^2$$

$$f_0 \geqslant f_\star + \langle g_\star, x_0 - x_\star\rangle + \tfrac{1}{2L}\|g_0 - g_\star\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}(g_0 - g_\star)\right\|^2.$$

- Same optimal value (no relaxation): non-convex quadratic problem.

Semidefinite lifting

- Define $P \triangleq [x_0 - x_\star,\ g_0] \in \mathbb{R}^{d\times 2}$ and $F \triangleq f_0 - f_\star$.

Semidefinite lifting

- Define $P \triangleq [x_0 - x_\star,\ g_0] \in \mathbb{R}^{d\times 2}$ and $F \triangleq f_0 - f_\star$.
- Using new variables $G \succcurlyeq 0$ and F

$$G \triangleq P^T P = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix} \succcurlyeq 0,$$

Semidefinite lifting

- Define $P \triangleq [x_0 - x_\star,\ g_0] \in \mathbb{R}^{d\times 2}$ and $F \triangleq f_0 - f_\star$.
- Using new variables $G \succcurlyeq 0$ and F

$$G \triangleq P^T P = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix} \succcurlyeq 0,$$

- previous problem can be relaxed to 2×2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{1,1} + \alpha^2 G_{2,2} - 2\alpha G_{1,2} \\ \text{subject to} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0 \end{aligned}$$

Semidefinite lifting

- Define $P \triangleq [x_0 - x_\star,\ g_0] \in \mathbb{R}^{d\times 2}$ and $F \triangleq f_0 - f_\star$.
- Using new variables $G \succcurlyeq 0$ and F

$$G \triangleq P^T P = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix} \succcurlyeq 0,$$

- previous problem can be relaxed to 2×2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{1,1} + \alpha^2 G_{2,2} - 2\alpha G_{1,2} \\ \text{subject to} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0 \end{aligned}$$

(using homogeneity argument and substituting x_1 and $g_\star$).

Semidefinite lifting

- Define $P \triangleq [x_0 - x_\star,\ g_0] \in \mathbb{R}^{d\times 2}$ and $F \triangleq f_0 - f_\star$.
- Using new variables $G \succcurlyeq 0$ and F

$$G \triangleq P^T P = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix} \succcurlyeq 0,$$

- previous problem can be relaxed to 2×2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{1,1} + \alpha^2 G_{2,2} - 2\alpha G_{1,2} \\ \text{subject to} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0 \end{aligned}$$

(using homogeneity argument and substituting x_1 and $g_\star$).
- Assuming $x_0, x_\star, g_0 \in \mathbb{R}^d$ with $d \geqslant 2$, same optimal value as original problem!

Semidefinite lifting

- Define $P \triangleq [x_0 - x_\star,\ g_0] \in \mathbb{R}^{d\times 2}$ and $F \triangleq f_0 - f_\star$.
- Using new variables $G \succcurlyeq 0$ and F

$$G \triangleq P^T P = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix} \succcurlyeq 0,$$

- previous problem can be relaxed to 2×2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{1,1} + \alpha^2 G_{2,2} - 2\alpha G_{1,2} \\ \text{subject to} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0 \end{aligned}$$

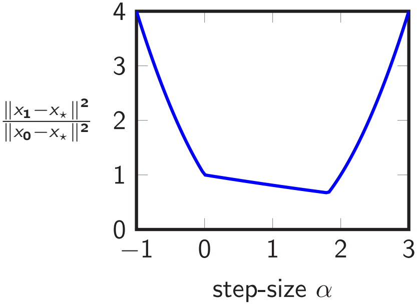
  (using homogeneity argument and substituting x_1 and $g_\star$).
- Assuming $x_0, x_\star, g_0 \in \mathbb{R}^d$ with $d \geqslant 2$, same optimal value as original problem!
- For $d = 1$ same as original problem by adding $\mathrm{rank}(G) \leqslant 1$.

Numerical solution of the SDP

Fix $L = 1$, $\mu = .1$ and solve the SDP for a few values of α.

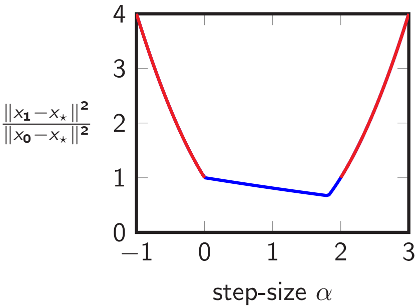
Numerical solution of the SDP

Fix $L = 1$, $\mu = .1$ and solve the SDP for a few values of α.



Numerical solution of the SDP

Fix $L = 1$, $\mu = .1$ and solve the SDP for a few values of α.



Observations:

- ◇ numerics match the known $\max\{(1-\alpha L)^2, (1-\alpha\mu)^2\}$
- ◇ recovers that gradient descent converges for $\alpha \in (0, 2/L)$ (divergence otherwise).

Dual problem

- Dual problem is

$$
\begin{aligned}
\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \quad & \tau \\
\text{subject to } S = & \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \alpha - \frac{\lambda_1 (L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda_1 (L+\mu)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0 \\
0 = & \lambda_1 - \lambda_2.
\end{aligned}
$$

Dual problem

- Dual problem is

$$\begin{aligned} \min_{\tau,\lambda_1,\lambda_2\geqslant 0} \ & \tau \\ \text{subject to } & S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0 \\ & 0 = \lambda_1 - \lambda_2. \end{aligned}$$

- Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate

Dual problem

- Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \tau$$
$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0$$
$$0 = \lambda_1 - \lambda_2.$$

- Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate
- Direct consequence: for any $\tau \geqslant 0$ we have

$$\|x_1 - x_\star\|^2 \leqslant \tau\|x_0 - x_\star\|^2 \text{ for all } f \in \mathcal{F}_{\mu,L}, \text{ all } x_0 \in \mathbb{R}^d, \text{ all } d \in \mathbb{N}, \text{ with } x_1 = x_0 - \alpha\nabla f(x_0).$$
$$\Uparrow$$
$$\exists\lambda \geqslant 0 : \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \alpha - \frac{\lambda(L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda(L+\mu)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0$$

Dual problem

- Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \tau$$
$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0$$
$$0 = \lambda_1 - \lambda_2.$$

- Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate ($\Uparrow$).
- Direct consequence: for any $\tau \geqslant 0$ we have

$$\|x_1 - x_\star\|^2 \leqslant \tau\|x_0 - x_\star\|^2 \text{ for all } f \in \mathcal{F}_{\mu,L}, \text{ all } x_0 \in \mathbb{R}^d, \text{ all } d \in \mathbb{N}, \text{ with } x_1 = x_0 - \alpha\nabla f(x_0).$$
$$\Uparrow$$
$$\exists\lambda \geqslant 0 : \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \alpha - \frac{\lambda(L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda(L+\mu)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0$$

Dual problem

- Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \tau$$
$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0$$
$$0 = \lambda_1 - \lambda_2.$$

- Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate ($\Uparrow$).
- Direct consequence: for any $\tau \geqslant 0$ we have

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2 \text{ for all } f \in \mathcal{F}_{\mu,L}, \text{ all } x_0 \in \mathbb{R}^d, \text{ all } d \in \mathbb{N}, \text{ with } x_1 = x_0 - \alpha \nabla f(x_0).$$
$$\Uparrow$$
$$\exists \lambda \geqslant 0 : \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \alpha - \frac{\lambda(L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda(L+\mu)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0$$

- Strong duality holds ($\exists$ Slater point): any valid worst-case convergence rate $\equiv$ valid dual feasible point ($\Downarrow$).

Dual problem

- Dual problem is

$$\begin{aligned} \min_{\tau,\lambda_1,\lambda_2\geqslant 0} \quad & \tau \\ \text{subject to } S = & \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda_1(L+\mu)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0 \\ & 0 = \lambda_1 - \lambda_2. \end{aligned}$$

- Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate ($\Uparrow$).
- Direct consequence: for any $\tau \geqslant 0$ we have

$$\|x_1 - x_\star\|^2 \leqslant \tau\|x_0 - x_\star\|^2 \text{ for all } f \in \mathcal{F}_{\mu,L}, \text{ all } x_0 \in \mathbb{R}^d, \text{ all } d \in \mathbb{N}, \text{ with } x_1 = x_0 - \alpha\nabla f(x_0).$$

$$\Uparrow \quad \Downarrow$$

$$\exists\lambda \geqslant 0 : \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \alpha - \frac{\lambda(L+\mu)}{2(L-\mu)} \\ \alpha - \frac{\lambda(L+\mu)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \alpha^2 \end{bmatrix} \succcurlyeq 0$$

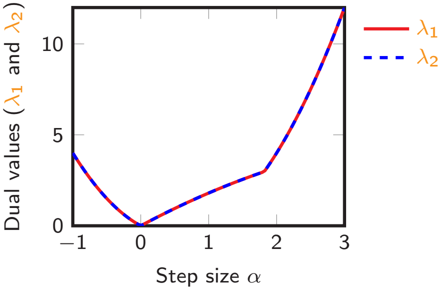
- Strong duality holds ($\exists$ Slater point): any valid worst-case convergence rate $\equiv$ valid dual feasible point ($\Downarrow$).

Dual solutions

Fix $L = 1$, $\mu = .1$ and solve the dual SDP for a few values of α.

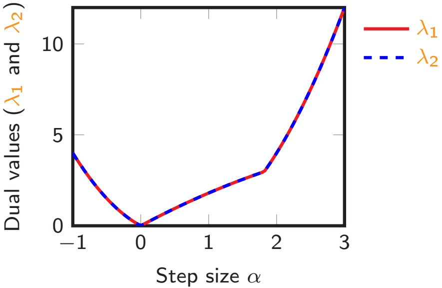
Dual solutions

Fix $L = 1$, $\mu = .1$ and solve the dual SDP for a few values of α.



Dual solutions

Fix $L = 1$, $\mu = .1$ and solve the dual SDP for a few values of α.



Numerics match $\lambda_1 = \lambda_2 = 2|\alpha|\rho(\alpha)$ with $\rho(\alpha) = \max\{\alpha L - 1, 1 - \alpha\mu\}$.

Recovering a “standard” proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

Recovering a "standard" proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

$$f_0 \geqslant f_\star + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_1$$

$$f_\star \geqslant f_0 + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_2$$

Recovering a "standard" proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

$$f_0 \geqslant f_\star + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_1$$

$$f_\star \geqslant f_0 + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_2$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

Recovering a "standard" proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

$$f_0 \geqslant f_\star + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_1 = 2\alpha(1-\alpha\mu)$$

$$f_\star \geqslant f_0 + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_2 = 2\alpha(1-\alpha\mu)$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

Recovering a "standard" proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

$$f_0 \geqslant f_\star + \frac{1}{2L}\|\nabla f(x_0)\|^2 + \frac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \frac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_1 = 2\alpha(1-\alpha\mu)$$

$$f_\star \geqslant f_0 + \langle \nabla f(x_0), x_\star - x_0\rangle + \frac{1}{2L}\|\nabla f(x_0)\|^2 + \frac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \frac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_2 = 2\alpha(1-\alpha\mu)$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$\|x_1 - x_\star\|^2 \leqslant (1-\alpha\mu)^2 \|x_0 - x_\star\|^2 \underbrace{- \alpha\frac{2-\alpha(L+\mu)}{L-\mu}\|\mu(x_0 - x_\star) - \nabla f(x_0)\|^2}_{},$$

Recovering a "standard" proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

$$f_0 \geqslant f_\star + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_1 = 2\alpha(1-\alpha\mu)$$

$$f_\star \geqslant f_0 + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_2 = 2\alpha(1-\alpha\mu)$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$\|x_1 - x_\star\|^2 \leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2 - \underbrace{\alpha\tfrac{2-\alpha(L+\mu)}{L-\mu}\|\mu(x_0 - x_\star) - \nabla f(x_0)\|^2}_{\geqslant 0},$$

Recovering a "standard" proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

$$f_0 \geqslant f_\star + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_1 = 2\alpha(1-\alpha\mu)$$

$$f_\star \geqslant f_0 + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_2 = 2\alpha(1-\alpha\mu)$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$\begin{aligned}\|x_1 - x_\star\|^2 &\leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2 - \underbrace{\alpha\tfrac{2-\alpha(L+\mu)}{L-\mu}\|\mu(x_0 - x_\star) - \nabla f(x_0)\|^2}_{\geqslant 0},\\ &\leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2,\end{aligned}$$

Recovering a "standard" proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

$$f_0 \geqslant f_\star + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \quad : \lambda_1 = 2\alpha(1-\alpha\mu)$$

$$f_\star \geqslant f_0 + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \quad : \lambda_2 = 2\alpha(1-\alpha\mu)$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$\begin{aligned}\|x_1 - x_\star\|^2 &\leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2 - \underbrace{\alpha\tfrac{2-\alpha(L+\mu)}{L-\mu}\|\mu(x_0 - x_\star) - \nabla f(x_0)\|^2}_{\geqslant 0}, \\ &\leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2,\end{aligned}$$

leading to $\|x_1 - x_\star\|^2 \leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2$

Recovering a “standard” proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

$$f_0 \geqslant f_\star + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_1 = 2\alpha(1-\alpha\mu)$$

$$f_\star \geqslant f_0 + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_2 = 2\alpha(1-\alpha\mu)$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$\|x_1 - x_\star\|^2 \leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2 - \underbrace{\alpha\tfrac{2-\alpha(L+\mu)}{L-\mu}\|\mu(x_0 - x_\star) - \nabla f(x_0)\|^2}_{\geqslant 0,\ \text{or } = 0 \text{ when worst-case is achieved}},$$

$$\leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2,$$

leading to $\|x_1 - x_\star\|^2 \leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2$

Recovering a "standard" proof

Gradient with $\alpha = \frac{1}{L}$. Perform weighted sum of two inequalities

$$f_0 \geqslant f_\star + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_1 = 2\alpha(1-\alpha\mu)$$

$$f_\star \geqslant f_0 + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_2 = 2\alpha(1-\alpha\mu)$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$\begin{aligned}\|x_1 - x_\star\|^2 &\leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2 - \underbrace{\alpha\tfrac{2-\alpha(L+\mu)}{L-\mu}\|\mu(x_0 - x_\star) - \nabla f(x_0)\|^2}_{\geqslant 0,\ \text{or } = 0 \text{ when worst-case is achieved}},\\ &\leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2,\end{aligned}$$

leading to $\|x_1 - x_\star\|^2 \leqslant (1-\alpha\mu)^2\,\|x_0 - x_\star\|^2$ (tight).

What did we do, so far?

Summary:

What did we do, so far?

Summary:

- we computed the smallest $\tau(\mu, L, \alpha)$ such that

$$\|x_1 - x_\star\|^2 \leqslant \tau(\mu, L, \alpha)\,\|x_0 - x_\star\|^2$$

  is satisfied for all $x_0 \in \mathbb{R}^d$, $d \in \mathbb{N}$, $f \in \mathcal{F}_{\mu,\mathrm{L}}$, and $x_1 = x_0 - \alpha\nabla f(x_0)$.

What did we do, so far?

Summary:

- we computed the smallest $\tau(\mu, L, \alpha)$ such that

$$\|x_1 - x_\star\|^2 \leqslant \tau(\mu, L, \alpha)\,\|x_0 - x_\star\|^2$$

  is satisfied for all $x_0 \in \mathbb{R}^d$, $d \in \mathbb{N}$, $f \in \mathcal{F}_{\mu,\mathrm{L}}$, and $x_1 = x_0 - \alpha \nabla f(x_0)$.
- Feasible points to primal SDP correspond to lower bounds on $\tau(\mu, L, \alpha)$.

What did we do, so far?

Summary:

- we computed the smallest $\tau(\mu, L, \alpha)$ such that

$$\|x_1 - x_\star\|^2 \leqslant \tau(\mu, L, \alpha)\, \|x_0 - x_\star\|^2$$

  is satisfied for all $x_0 \in \mathbb{R}^d$, $d \in \mathbb{N}$, $f \in \mathcal{F}_{\mu,L}$, and $x_1 = x_0 - \alpha \nabla f(x_0)$.
- Feasible points to primal SDP correspond to lower bounds on $\tau(\mu, L, \alpha)$.
- Feasible points to dual SDP correspond to upper bounds on $\tau(\mu, L, \alpha)$.

What did we do, so far?

Summary:

- we computed the smallest $\tau(\mu, L, \alpha)$ such that

$$\|x_1 - x_\star\|^2 \leqslant \tau(\mu, L, \alpha)\,\|x_0 - x_\star\|^2$$

  is satisfied for all $x_0 \in \mathbb{R}^d$, $d \in \mathbb{N}$, $f \in \mathcal{F}_{\mu,L}$, and $x_1 = x_0 - \alpha\nabla f(x_0)$.
- Feasible points to primal SDP correspond to lower bounds on $\tau(\mu, L, \alpha)$.
- Feasible points to dual SDP correspond to upper bounds on $\tau(\mu, L, \alpha)$.
  - proof via linear combinations of interpolation inequalities (evaluated at $\{x_k\}_k$ and $x_\star$),
  - proofs can be rewritten as a "sum-of-squares" certificates (sum of squared norms).

When does it work?

The methodology applies, as is, as soon as:

When does it work?

The methodology applies, as is, as soon as:

- ⋄ performance measure and initial condition are linear in G and F,

When does it work?

The methodology applies, as is, as soon as:

- ◇ performance measure and initial condition are linear in G and F,
- ◇ interpolation inequalities are linear in G and F,

When does it work?

The methodology applies, as is, as soon as:

- ◇ performance measure and initial condition are linear in G and F,
- ◇ interpolation inequalities are linear in G and F,
- ◇ algorithm can be described linearly in G and F,

When does it work?

The methodology applies, as is, as soon as:

- ◇ performance measure and initial condition are linear in G and F,
- ◇ interpolation inequalities are linear in G and F,
- ◇ algorithm can be described linearly in G and F,

(but other cases are not doomed).

Software: PEPit/PESTO

PEPit

Search docs

CONTENTS:

- Welcome to PEPit's documentation!
- Quick start guide
- API and modules
- Examples
- What's new in PEPit
- Contributing

Welcome to PEPit's documentation! View page source

Welcome to PEPit's documentation!

Contents:

PEPit: Performance Estimation in Python

Tests passing | codecov 89% | docs passing | pypi package 0.3.2 | downloads 29k | license MIT

This open source Python library provides a generic way to use PEP framework in Python. Performance estimation problems were introduced in 2014 by **Yoel Drori** and **Marc Teboulle**, see [1]. PEPit is mainly based on the formalism and developments from [2, 3] by a subset of the authors of this toolbox. A friendly informal introduction to this formalism is available in this blog post and a corresponding Matlab library is presented in [4] (PESTO).

Website and documentation of PEPit: https://pepit.readthedocs.io/

Source Code (MIT): https://github.com/PerformanceEstimation/PEPit

Using and citing the toolbox

This code comes jointly with the following `reference`:

```
B. Goujaud, C. Moucer, F. Glineur, J. Hendrickx, A. Taylor, A. Dieuleveut (2022).
```

Example 1: gradient methods and momentum

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

with f an L-smooth convex function.

Example 1: gradient methods and momentum

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

with f an L-smooth convex function. Three algorithms:

Example 1: gradient methods and momentum

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

with f an L-smooth convex function. Three algorithms:

- Gradient descent: $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$.

Example 1: gradient methods and momentum

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

with f an L-smooth convex function. Three algorithms:

- Gradient descent: $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$.
- Heavy-ball method [Polyak, 1964] $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$

Example 1: gradient methods and momentum

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

with f an L-smooth convex function. Three algorithms:

- Gradient descent: $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$.
- Heavy-ball method [Polyak, 1964] $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$
  (choice: $\alpha = \frac{1}{2L}$, $\beta = \sqrt{1 - L\alpha}$, see [Ghadimi, Feyzmahdavian, Johansson, 2015])

Example 1: gradient methods and momentum

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

with f an L-smooth convex function. Three algorithms:

- Gradient descent: $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$.
- Heavy-ball method [Polyak, 1964] $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$
  (choice: $\alpha = \frac{1}{2L}$, $\beta = \sqrt{1 - L\alpha}$, see [Ghadimi, Feyzmahdavian, Johansson, 2015])
- Accelerated gradient method [Nesterov, 1983]:

$$\begin{aligned} x_{k+1} &= y_k - \tfrac{1}{L}\nabla f(x_k) \\ y_{k+1} &= x_{k+1} + \tfrac{k-1}{k+1}(x_{k+1} - x_k). \end{aligned}$$

Example 1: gradient methods and momentum

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

with f an L-smooth convex function. Three algorithms:

- ◇ Gradient descent: $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$.
- ◇ Heavy-ball method [Polyak, 1964] $x_{k+1} = x_k - \alpha\nabla f(x_k) + \beta(x_k - x_{k-1})$
  (choice: $\alpha = \frac{1}{2L}$, $\beta = \sqrt{1 - L\alpha}$, see [Ghadimi, Feyzmahdavian, Johansson, 2015])
- ◇ Accelerated gradient method [Nesterov, 1983]:

$$x_{k+1} = y_k - \tfrac{1}{L}\nabla f(x_k)$$
$$y_{k+1} = x_{k+1} + \tfrac{k-1}{k+1}(x_{k+1} - x_k).$$

What can we guarantee on...

$$\frac{f(x_N) - f(x_\star)}{\|x_0 - x_\star\|^2} \leqslant ? \qquad \frac{\|\nabla f(x_N)\|^2}{\|x_0 - x_\star\|^2} \leqslant ? \qquad \frac{\min_{0\leqslant k\leqslant N}\|\nabla f(x_k)\|^2}{\|x_0 - x_\star\|^2} \leqslant ?$$

Example 2: a primal-dual proximal point

Minimize sum of two convex (ccp) functions

$$\min_{x \in \mathbb{R}^d} f(x) + h(x)$$

assume $\exists x_\star, y_\star$ (KKT point): $-y_\star \in \partial f(x_\star)$, $x_\star \in \partial h^*(y_\star)$.

Example 2: a primal-dual proximal point

Minimize sum of two convex (ccp) functions

$$\min_{x\in\mathbb{R}^d} f(x) + h(x)$$

assume $\exists x_\star, y_\star$ (KKT point): $-y_\star \in \partial f(x_\star)$, $x_\star \in \partial h^*(y_\star)$.

Primal-dual proximal point algorithm (see, e.g., [Rockafellar, 1976])

Input: f, h convex (ccp) functions, $(y_0, x_0) \in \mathbb{R}^d \times \mathbb{R}^d$.

For $k = 0, 1, \ldots$

$$\begin{aligned}(y_{k+1}, x_{k+1}) = &\operatorname*{argmax}_{y\in\mathbb{R}^d} \operatorname*{argmin}_{x\in\mathbb{R}^d} \Big\{ f(x) - h^*(y) + \langle y, x\rangle \\ &\qquad + \tfrac{1}{2\alpha}\|x - x_k\|^2 - \tfrac{1}{2\alpha}\|y - y_k\|^2 \Big\}\end{aligned}$$

Example 2: a primal-dual proximal point

Minimize sum of two convex (ccp) functions

$$\min_{x \in \mathbb{R}^d} f(x) + h(x)$$

assume $\exists x_\star, y_\star$ (KKT point): $-y_\star \in \partial f(x_\star)$, $x_\star \in \partial h^*(y_\star)$.

Primal-dual proximal point algorithm (see, e.g., [Rockafellar, 1976])

Input: f, h convex (ccp) functions, $(y_0, x_0) \in \mathbb{R}^d \times \mathbb{R}^d$.

For $k = 0, 1, \ldots$

$$(y_{k+1}, x_{k+1}) = \operatorname*{argmax}_{y \in \mathbb{R}^d} \operatorname*{argmin}_{x \in \mathbb{R}^d} \Big\{ f(x) - h^*(y) + \langle y, x \rangle + \tfrac{1}{2\alpha} \|x - x_k\|^2 - \tfrac{1}{2\alpha} \|y - y_k\|^2 \Big\}$$

What guarantees of type (for some elements of ∂f and ∂h^*)

$$\frac{\|\partial f(x_N) + y_N\|^2 + \|x_N - \partial h^*(y_N)\|^2}{\|x_0 - x_\star\|^2 + \|y_0 - y_\star\|^2} \leqslant \tau(N, \alpha)?$$

Recap'

Recap'

- Worst-case guarantees *cannot be improved*, systematic approach,
- allows reaching analyses that could barely be obtained by hand,
- fair amount of scenarios/algorithms (e.g., stochastic, distributed, error feedback, etc.),
- SDPs typically become prohibitively large in a variety of scenarios,
- transient behavior VS. asymptotic behavior: might be hard to distinguish with small N,
- proofs (may be) quite involved and hard to intuit,
- proofs (may be) hard to generalize.

A few instructive examples

Worst-case analysis for fixed-point iterations:

- Lieder (2021). "On the convergence of the Halpern-iteration". Optimization letters 15(2).

Analysis of the proximal-point algorithm for monotone inclusions:

- Gu, Yang (2019). "Optimal nonergodic sublinear convergence rate of the proximal point algorithm for maximal monotone inclusion problems". SIAM Journal on Optimization 30(3).

Application to nonconvex optimization:

- Abbaszadehpeivasti, de Klerk, Zamani (2022). "The exact worst-case convergence rate of the gradient method with fixed step lengths for L-smooth functions". Optimization Letters 16(6).

Applications to distributed optimization:

- Sundararajan, Van Scoy, Lessard (2020). "Analysis and design of first-order distributed optimization algorithms over time-varying graphs." IEEE Transactions on Control of Network Systems 7(4).
- Colla, Hendrickx (2023). "Automatic performance estimation for decentralized optimization." IEEE Transactions on Automatic Control 68(12).

Gradient descent for smooth convex minimization (definitive answers):

- Teboulle, Vaisbourd (2023). "An elementary approach to tight worst case complexity analysis of gradient based methods." Mathematical Programming 201(1).

A few references

Historical reference:

- Drori, Teboulle (2014). "Performance of first-order methods for smooth convex minimization: a novel approach." Mathematical Programming 145 (1).

Main messages of this part:

- T, Hendrickx, Glineur (2017). "Smooth strongly convex interpolation and exact worst-case performance of first-order methods." Mathematical Programming 161.

- Goujaud, Dieuleveut, T (2023). "On fundamental proof structures in first-order optimization." Conference on Decision and Control (CDC).

- Goujaud, Moucer, Glineur, Hendrickx, T, Dieuleveut (2024). "PEPit: computer-assisted worst-case analyses of first-order optimization methods in Python." Mathematical Programming Computation 16(3).

A few references

Historical reference:

- Drori, Teboulle (2014). "Performance of first-order methods for smooth convex minimization: a novel approach." Mathematical Programming 145 (1).

Main messages of this part:

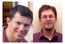
- T, Hendrickx, Glineur (2017). "Smooth strongly convex interpolation and exact worst-case performance of first-order methods." Mathematical Programming 161.

- Goujaud, Dieuleveut, T (2023). "On fundamental proof structures in first-order optimization." Conference on Decision and Control (CDC).

- Goujaud, Moucer, Glineur, Hendrickx, T, Dieuleveut (2024). "PEPit: computer-assisted worst-case analyses of first-order optimization methods in Python." Mathematical Programming Computation 16(3).

To go further:

- T, Hendrickx, Glineur (2017). "Exact worst-case performance of first-order methods for composite convex optimization." SIAM Journal on Optimization 27(3).



- Dragomir, T, d'Aspremont, Bolte (2022). "Optimal complexity and certification of Bregman first-order methods." Mathematical Programming 194.

- Barré, T, Bach (2023). "Principled analyses and design of first-order methods with inexact proximal operators." Mathematical Programming 201(1).

Constructive approach to performance analysis

Towards structured analyses

Towards optimal algorithms

Concluding remarks

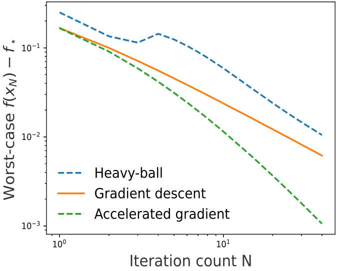
Structured performance analyses

Structured performance analyses

So far: we searched for iteration-dependent analyses.

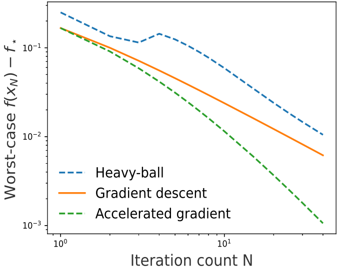
Structured performance analyses

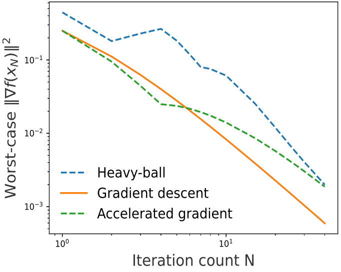
So far: we searched for iteration-dependent analyses.



Structured performance analyses

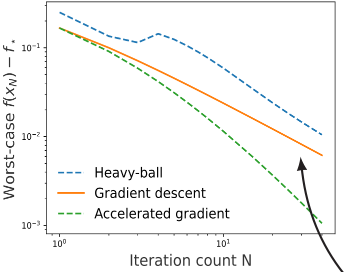
So far: we searched for iteration-dependent analyses.

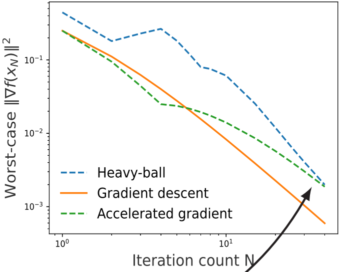




Structured performance analyses

So far: we searched for iteration-dependent analyses.





What to expect for larger N?

What is a Lyapunov function?[2,3,4]

Dynamical system: $\xi_{k+1} = F(\xi_k)$ with fixed point $\xi_\star = F(\xi_\star)$.

[2] See, e.g., [Lyapunov and Fuller, 1992]; original text in Russian [Lyapunov, 1892]. Many possible variations around our definition.

[3] See, e.g., [Bansal and Gupta, 2019] or [Wilson, Recht, Jordan, 2021] in the context of first-order optimization.

[4] Many traditional analyses follow such patterns, see, e.g. [Polyak, 1964], [Nesterov, 1983].

What is a Lyapunov function?[2,3,4]

Dynamical system: $\xi_{k+1} = F(\xi_k)$ with fixed point $\xi_\star = F(\xi_\star)$. Convergence of $\{\xi_k\}_k$ towards $\xi_\star$?

---

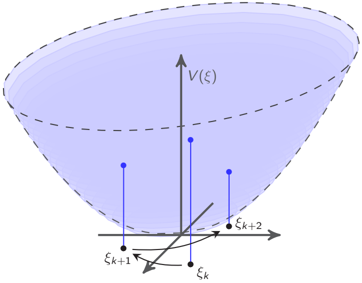
[2] See, e.g., [Lyapunov and Fuller, 1992]; original text in Russian [Lyapunov, 1892]. Many possible variations around our definition.

[3] See, e.g., [Bansal and Gupta, 2019] or [Wilson, Recht, Jordan, 2021] in the context of first-order optimization.

[4] Many traditional analyses follow such patterns, see, e.g. [Polyak, 1964], [Nesterov, 1983].

What is a Lyapunov function?[2,3,4]

Dynamical system: $\xi_{k+1} = F(\xi_k)$ with fixed point $\xi_\star = F(\xi_\star)$. Convergence of $\{\xi_k\}_k$ towards $\xi_\star$?



[2] See, e.g., [Lyapunov and Fuller, 1992]; original text in Russian [Lyapunov, 1892]. Many possible variations around our definition.

[3] See, e.g., [Bansal and Gupta, 2019] or [Wilson, Recht, Jordan, 2021] in the context of first-order optimization.

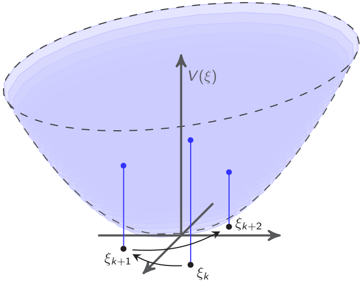
[4] Many traditional analyses follow such patterns, see, e.g. [Polyak, 1964], [Nesterov, 1983].

What is a Lyapunov function?[2,3,4]

Dynamical system: $\xi_{k+1} = F(\xi_k)$ with fixed point $\xi_\star = F(\xi_\star)$. Convergence of $\{\xi_k\}_k$ towards $\xi_\star$?

Lyapunov function:

- $V(\xi) \geqslant 0 \ \forall \xi$,
- $V(\xi) = 0 \Leftrightarrow \xi = \xi_\star$,
- $\nu(\|\xi - \xi_\star\|) \leqslant V(\xi)$ (for some increasing $\nu(\cdot)$),
- $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ (for some $\rho < 1$)



[2] See, e.g., [Lyapunov and Fuller, 1992]; original text in Russian [Lyapunov, 1892]. Many possible variations around our definition.

[3] See, e.g., [Bansal and Gupta, 2019] or [Wilson, Recht, Jordan, 2021] in the context of first-order optimization.

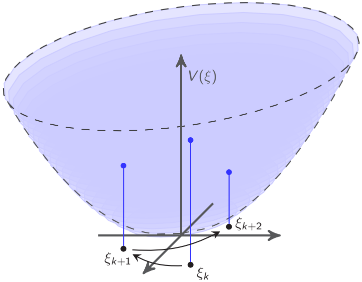
[4] Many traditional analyses follow such patterns, see, e.g. [Polyak, 1964], [Nesterov, 1983].

What is a Lyapunov function?[2,3,4]

Dynamical system: $\xi_{k+1} = F(\xi_k)$ with fixed point $\xi_\star = F(\xi_\star)$. Convergence of $\{\xi_k\}_k$ towards $\xi_\star$?

Lyapunov function:

- $V(\xi) \geqslant 0 \ \forall \xi$,
- $V(\xi) = 0 \ \Leftrightarrow \xi = \xi_\star$,
- $\nu(\|\xi - \xi_\star\|) \leqslant V(\xi)$ (for some increasing $\nu(\cdot)$),
- $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ (for some $\rho < 1$)



Why nice? Pick for instance $\nu(\|\xi - \xi_\star\|) = \|\xi - \xi_\star\|^2$

[2] See, e.g., [Lyapunov and Fuller, 1992]; original text in Russian [Lyapunov, 1892]. Many possible variations around our definition.

[3] See, e.g., [Bansal and Gupta, 2019] or [Wilson, Recht, Jordan, 2021] in the context of first-order optimization.

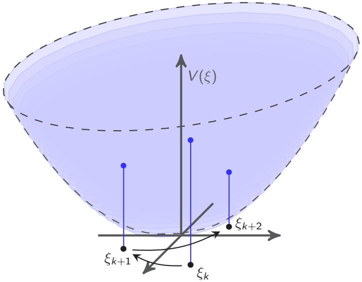
[4] Many traditional analyses follow such patterns, see, e.g. [Polyak, 1964], [Nesterov, 1983].

What is a Lyapunov function?[2,3,4]

Dynamical system: $\xi_{k+1} = F(\xi_k)$ with fixed point $\xi_\star = F(\xi_\star)$. Convergence of $\{\xi_k\}_k$ towards $\xi_\star$?

Lyapunov function:

- $V(\xi) \geqslant 0 \; \forall \xi$,
- $V(\xi) = 0 \; \Leftrightarrow \xi = \xi_\star$,
- $\nu(\|\xi - \xi_\star\|) \leqslant V(\xi)$ (for some increasing $\nu(\cdot)$),
- $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ (for some $\rho < 1$)



Why nice? Pick for instance $\nu(\|\xi - \xi_\star\|) = \|\xi - \xi_\star\|^2$

$$V(\xi_N) \leqslant \rho V(\xi_{N-1})$$

[2] See, e.g., [Lyapunov and Fuller, 1992]; original text in Russian [Lyapunov, 1892]. Many possible variations around our definition.

[3] See, e.g., [Bansal and Gupta, 2019] or [Wilson, Recht, Jordan, 2021] in the context of first-order optimization.

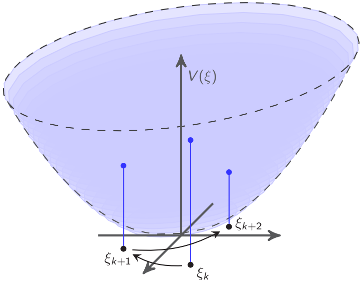
[4] Many traditional analyses follow such patterns, see, e.g. [Polyak, 1964], [Nesterov, 1983].

What is a Lyapunov function?[2,3,4]

Dynamical system: $\xi_{k+1} = F(\xi_k)$ with fixed point $\xi_\star = F(\xi_\star)$. Convergence of $\{\xi_k\}_k$ towards $\xi_\star$?

Lyapunov function:

- $V(\xi) \geqslant 0\ \forall \xi$,
- $V(\xi) = 0 \Leftrightarrow \xi = \xi_\star$,
- $\nu(\|\xi - \xi_\star\|) \leqslant V(\xi)$ (for some increasing $\nu(\cdot)$),
- $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ (for some $\rho < 1$)



Why nice? Pick for instance $\nu(\|\xi - \xi_\star\|) = \|\xi - \xi_\star\|^2$

$$V(\xi_N) \leqslant \rho V(\xi_{N-1}) \leqslant \ldots \leqslant \rho^N V(\xi_0).$$

[2] See, e.g., [Lyapunov and Fuller, 1992]; original text in Russian [Lyapunov, 1892]. Many possible variations around our definition.

[3] See, e.g., [Bansal and Gupta, 2019] or [Wilson, Recht, Jordan, 2021] in the context of first-order optimization.

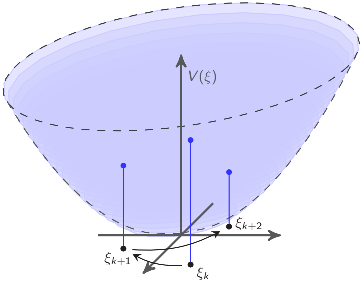
[4] Many traditional analyses follow such patterns, see, e.g. [Polyak, 1964], [Nesterov, 1983].

What is a Lyapunov function?[2,3,4]

Dynamical system: $\xi_{k+1} = F(\xi_k)$ with fixed point $\xi_\star = F(\xi_\star)$. Convergence of $\{\xi_k\}_k$ towards $\xi_\star$?

Lyapunov function:

- $V(\xi) \geqslant 0 \ \forall \xi$,
- $V(\xi) = 0 \ \Leftrightarrow \xi = \xi_\star$,
- $\nu(\|\xi - \xi_\star\|) \leqslant V(\xi)$ (for some increasing $\nu(\cdot)$),
- $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ (for some $\rho < 1$)



Why nice? Pick for instance $\nu(\|\xi - \xi_\star\|) = \|\xi - \xi_\star\|^2$

$$\|\xi_N - \xi_\star\|^2 \leqslant V(\xi_N) \leqslant \rho V(\xi_{N-1}) \leqslant \ldots \leqslant \rho^N V(\xi_0).$$

[2] See, e.g., [Lyapunov and Fuller, 1992]; original text in Russian [Lyapunov, 1892]. Many possible variations around our definition.

[3] See, e.g., [Bansal and Gupta, 2019] or [Wilson, Recht, Jordan, 2021] in the context of first-order optimization.

[4] Many traditional analyses follow such patterns, see, e.g. [Polyak, 1964], [Nesterov, 1983].

Lyapunov functions for gradient descent

Gradient descent: $x_{k+1} = x_k - \alpha \nabla f(x_k)$. Reasonable to choose

$$\xi_k = [x_k - x_\star, \nabla f(x_k), f(x_k) - f(x_\star)] \text{ and } \xi_\star = [0, 0, 0]$$

Lyapunov functions for gradient descent

Gradient descent: $x_{k+1} = x_k - \alpha \nabla f(x_k)$. Reasonable to choose

$$\xi_k = [x_k - x_\star, \nabla f(x_k), f(x_k) - f(x_\star)] \text{ and } \xi_\star = [0, 0, 0]$$

with

$$V(\xi_k) = \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix}^\top (P \otimes I_d) \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix} + p \left(f(x_k) - f(x_\star) \right).$$

Lyapunov functions for gradient descent

Gradient descent: $x_{k+1} = x_k - \alpha \nabla f(x_k)$. Reasonable to choose

$$\xi_k = [x_k - x_\star, \nabla f(x_k), f(x_k) - f(x_\star)] \text{ and } \xi_\star = [0, 0, 0]$$

with

$$V(\xi_k) = \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix}^\top (P \otimes I_d) \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix} + p \left(f(x_k) - f(x_\star) \right).$$

In other words:

$$V(\xi_k) = P_{1,1}\|x_k - x_\star\|^2 + 2P_{1,2}\langle \nabla f(x_k);\, x_k - x_\star \rangle + P_{2,2}\|\nabla f(x_k)\|^2 + p(f(x_k) - f(x_\star)).$$

Lyapunov functions for gradient descent

Gradient descent: $x_{k+1} = x_k - \alpha \nabla f(x_k)$. Reasonable to choose

$$\xi_k = [x_k - x_\star, \nabla f(x_k), f(x_k) - f(x_\star)] \text{ and } \xi_\star = [0, 0, 0]$$

with

$$V(\xi_k) = \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix}^\top (P \otimes I_d) \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix} + p\left(f(x_k) - f(x_\star)\right).$$

In other words:

$$V(\xi_k) = P_{1,1}\|x_k - x_\star\|^2 + 2P_{1,2}\langle \nabla f(x_k); x_k - x_\star\rangle + P_{2,2}\|\nabla f(x_k)\|^2 + p(f(x_k) - f(x_\star)).$$

Goal: characterize the set of $(P, p) \in \mathbb{S}^2 \times \mathbb{R}$:

Lyapunov functions for gradient descent

Gradient descent: $x_{k+1} = x_k - \alpha \nabla f(x_k)$. Reasonable to choose

$$\xi_k = [x_k - x_\star, \nabla f(x_k), f(x_k) - f(x_\star)] \text{ and } \xi_\star = [0, 0, 0]$$

with

$$V(\xi_k) = \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix}^\top (P \otimes I_d) \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix} + p\left(f(x_k) - f(x_\star)\right).$$

In other words:

$$V(\xi_k) = P_{1,1}\|x_k - x_\star\|^2 + 2P_{1,2}\langle \nabla f(x_k); x_k - x_\star\rangle + P_{2,2}\|\nabla f(x_k)\|^2 + p(f(x_k) - f(x_\star)).$$

Goal: characterize the set of $(P, p) \in \mathbb{S}^2 \times \mathbb{R}$:

- for which $V(\xi) \geqslant \|x - x_\star\|^2$ for all $d \in \mathbb{N}$, $\xi \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, and

Lyapunov functions for gradient descent

Gradient descent: $x_{k+1} = x_k - \alpha\nabla f(x_k)$. Reasonable to choose

$$\xi_k = [x_k - x_\star, \nabla f(x_k), f(x_k) - f(x_\star)] \text{ and } \xi_\star = [0,0,0]$$

with

$$V(\xi_k) = \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix}^\top (P \otimes I_d) \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix} + p\left(f(x_k) - f(x_\star)\right).$$

In other words:

$$V(\xi_k) = P_{1,1}\|x_k - x_\star\|^2 + 2P_{1,2}\langle \nabla f(x_k); x_k - x_\star\rangle + P_{2,2}\|\nabla f(x_k)\|^2 + p(f(x_k) - f(x_\star)).$$

Goal: characterize the set of $(P, p) \in \mathbb{S}^2 \times \mathbb{R}$:

- for which $V(\xi) \geqslant \|x - x_\star\|^2$ for all $d \in \mathbb{N}$, $\xi \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, and
- for which $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ for all $d \in \mathbb{N}$, $\xi_k \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, compatible ξ_{k+1}.

Verifying a Lyapunov function

Verifying a Lyapunov function

Goal: characterize the set of $(P, p) \in \mathbb{S}^2 \times \mathbb{R}$:

- for which $V(\xi) \geqslant \|x - x_\star\|^2$ for all $d \in \mathbb{N}$, $\xi \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, and
- for which $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ for all $d \in \mathbb{N}$, $\xi_k \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, compatible ξ_{k+1}.

Verifying a Lyapunov function

Goal: characterize the set of $(P, p) \in \mathbb{S}^2 \times \mathbb{R}$:

- for which $V(\xi) \geqslant \|x - x_\star\|^2$ for all $d \in \mathbb{N}$, $\xi \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, and
- for which $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ for all $d \in \mathbb{N}$, $\xi_k \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, compatible ξ_{k+1}.

Verify first property

Verifying a Lyapunov function

Goal: characterize the set of $(P, p) \in \mathbb{S}^2 \times \mathbb{R}$:

- for which $V(\xi) \geqslant \|x - x_\star\|^2$ for all $d \in \mathbb{N}$, $\xi \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, and
- for which $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ for all $d \in \mathbb{N}$, $\xi_k \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, compatible ξ_{k+1}.

Verify first property $\Leftrightarrow \forall \epsilon \geqslant 0$:

$$\begin{aligned} \epsilon \leqslant \inf_{\substack{d \in \mathbb{N}, f \in \mathcal{F}_{\mu,L} \\ x, x_\star \in \mathbb{R}^d}} \; & V\left([x - x_\star, \nabla f(x), f(x) - f(x_\star)]\right) \\ \text{s.t. } & \|x - x_\star\|^2 \geqslant \epsilon, \\ & \nabla f(x_\star) = 0, \end{aligned}$$

Verifying a Lyapunov function

Goal: characterize the set of $(P, p) \in \mathbb{S}^2 \times \mathbb{R}$:

- for which $V(\xi) \geqslant \|x - x_\star\|^2$ for all $d \in \mathbb{N}$, $\xi \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, and
- for which $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ for all $d \in \mathbb{N}$, $\xi_k \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, compatible ξ_{k+1}.

Verify first property $\Leftrightarrow \forall \epsilon \geqslant 0$:

$$\begin{aligned}\epsilon \leqslant \inf_{\substack{d \in \mathbb{N}, f \in \mathcal{F}_{\mu,L} \\ x, x_\star \in \mathbb{R}^d}} \; & V\left([x - x_\star, \nabla f(x), f(x) - f(x_\star)]\right) \\ \text{s.t. } & \|x - x_\star\|^2 \geqslant \epsilon, \\ & \nabla f(x_\star) = 0,\end{aligned}$$

...optimization on space $f \in \mathcal{F}_{\mu,L}$. Optimization variables: $d, f, x, x_\star$.

Verifying a Lyapunov function

Goal: characterize the set of $(P, p) \in \mathbb{S}^2 \times \mathbb{R}$:

- for which $V(\xi) \geqslant \|x - x_\star\|^2$ for all $d \in \mathbb{N}$, $\xi \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, and
- for which $V(\xi_{k+1}) \leqslant \rho V(\xi_k)$ for all $d \in \mathbb{N}$, $\xi_k \in (\mathbb{R}^d)^2 \times (\mathbb{R})^2$, $f \in \mathcal{F}_{\mu,L}$, compatible ξ_{k+1}.

Verify first property $\Leftrightarrow \forall \epsilon \geqslant 0$:

$$\begin{aligned} \epsilon \leqslant \inf_{\substack{d \in \mathbb{N}, f \in \mathcal{F}_{\mu,L} \\ x, x_\star \in \mathbb{R}^d}} \quad & V\left([x - x_\star, \nabla f(x), f(x) - f(x_\star)]\right) \\ \text{s.t. } & \|x - x_\star\|^2 \geqslant \epsilon, \\ & \nabla f(x_\star) = 0, \end{aligned}$$

...optimization on space $f \in \mathcal{F}_{\mu,L}$. Optimization variables: $d, f, x, x_\star$.

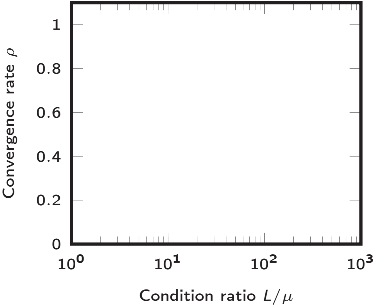
$\Rightarrow$ both conditions can be reframed as LMIs (bonus: linear in $(P, p) \in \mathbb{S}^2 \times \mathbb{R}$).

Examples: vanilla first-order methods

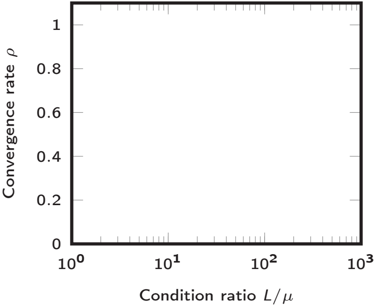
Method	α	β	γ

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

“Lyapunov rates“



“Lyapunov rates” with $P \succcurlyeq 0$, $p \geqslant 0$

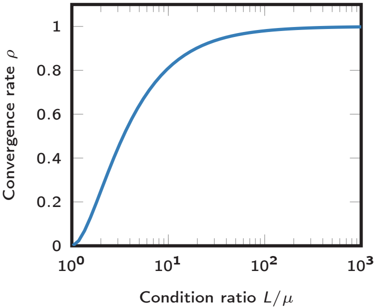


Examples: vanilla first-order methods

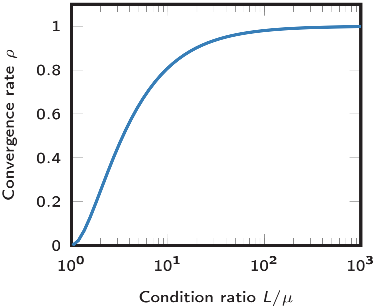
Method	α	β	γ
GD	$\frac{1}{L}$	0	0

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

“Lyapunov rates“



“Lyapunov rates” with $P \succcurlyeq 0$, $p \geqslant 0$

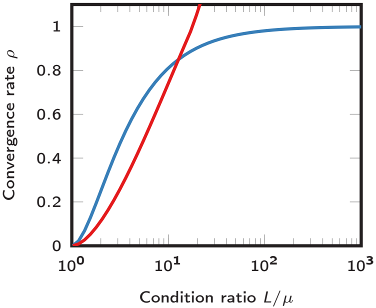


Examples: vanilla first-order methods

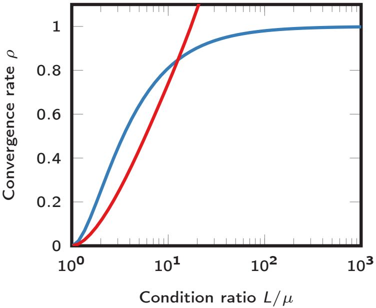
Method	α	β	γ
GD	$\frac{1}{L}$	0	0
Polyak's	$\frac{4}{(\sqrt{L}+\sqrt{\mu})^2}$	$\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^2$	0

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

“Lyapunov rates“



“Lyapunov rates” with $P \succcurlyeq 0$, $p \geqslant 0$

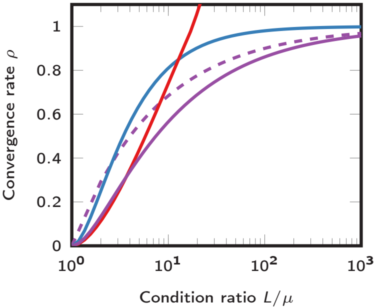


Examples: vanilla first-order methods

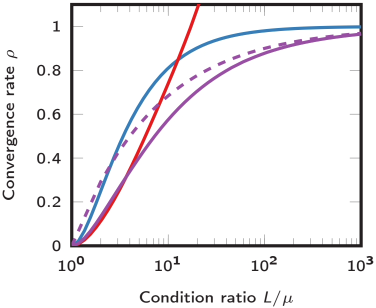
$$x_k = y_k + \gamma(y_k - y_{k-1})$$
$$y_{k+1} = y_k + \beta(y_k - y_{k-1}) - \alpha \nabla f(x_k)$$

Method	α	β	γ
GD	$\frac{1}{L}$	0	0
Polyak's	$\frac{4}{(\sqrt{L}+\sqrt{\mu})^2}$	$\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^2$	0
Nesterov's	$\frac{1}{L}$	$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$	$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$

"Lyapunov rates"



"Lyapunov rates" with $P \succcurlyeq 0$, $p \geqslant 0$

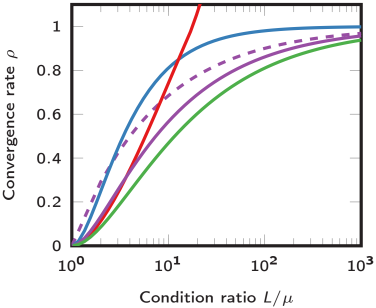


Examples: vanilla first-order methods

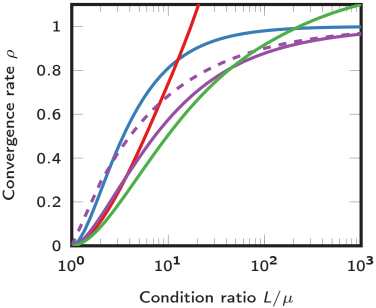
$$x_k = y_k + \gamma(y_k - y_{k-1})$$
$$y_{k+1} = y_k + \beta(y_k - y_{k-1}) - \alpha \nabla f(x_k)$$

Method	α	β	γ
GD	$\frac{1}{L}$	0	0
Polyak's	$\frac{4}{(\sqrt{L}+\sqrt{\mu})^2}$	$\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^2$	0
Nesterov's	$\frac{1}{L}$	$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$	$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$
Van Scoy's	$\frac{2\sqrt{L}-\sqrt{\mu}}{L\sqrt{L}}$	$\frac{(\sqrt{\kappa}-1)^2}{\kappa+\sqrt{\kappa}}$	$\frac{(\sqrt{\kappa}-1)^2}{2\kappa+\sqrt{\kappa}-1}$

"Lyapunov rates"



"Lyapunov rates" with $P \succcurlyeq 0$, $p \geqslant 0$



Primal-Dual Hybrid Gradient (PDHG)[4]

$$\min_{x \in \mathbb{R}^d} f(x) + h(x),$$

with f, h convex (closed, proper) functions and prox_f, prox_h simple to evaluate.

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau y_k\right),$$
$$y_{k+1} = \mathrm{prox}_{\sigma h^*}\left(y_k + \sigma\left(x_{k+1} + \theta\left(x_{k+1} - x_k\right)\right)\right).$$

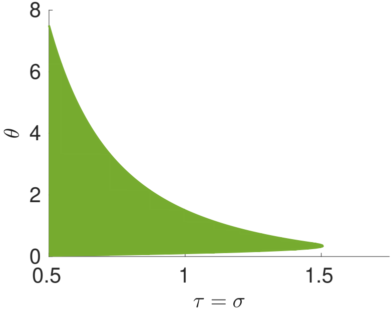
[4] See [Chambolle and Pock, 2011].

Primal-Dual Hybrid Gradient (PDHG)[4]

$$\min_{x \in \mathbb{R}^d} f(x) + h(x),$$

with f, h convex (closed, proper) functions and prox_f, prox_h simple to evaluate.

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau y_k\right),$$
$$y_{k+1} = \mathrm{prox}_{\sigma h^*}\left(y_k + \sigma\left(x_{k+1} + \theta\left(x_{k+1} - x_k\right)\right)\right).$$



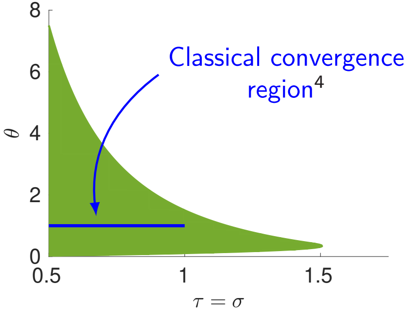
[4] See [Chambolle and Pock, 2011].

Primal-Dual Hybrid Gradient (PDHG)[4]

$$\min_{x\in\mathbb{R}^d} f(x) + h(x),$$

with f, h convex (closed, proper) functions and prox_f, prox_h simple to evaluate.

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau y_k\right),$$
$$y_{k+1} = \mathrm{prox}_{\sigma h^*}\left(y_k + \sigma\left(x_{k+1} + \theta\left(x_{k+1} - x_k\right)\right)\right).$$



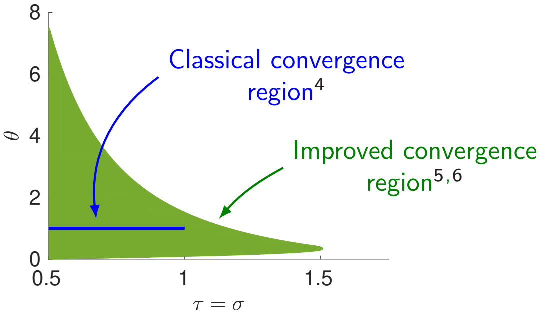
[4] See [Chambolle and Pock, 2011].

Primal-Dual Hybrid Gradient (PDHG)[4]

$$\min_{x\in\mathbb{R}^d} f(x) + h(x),$$

with f, h convex (closed, proper) functions and prox_f, prox_h simple to evaluate.

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau y_k\right),$$
$$y_{k+1} = \mathrm{prox}_{\sigma h^*}\left(y_k + \sigma\left(x_{k+1} + \theta\left(x_{k+1} - x_k\right)\right)\right).$$



[4] See [Chambolle and Pock, 2011].

[5] Code available here: https://github.com/ManuUpadhyaya/TightLyapunovAnalysis.

[6] Improved region partially described (closed-forms) in [Banert, Upadhyaya, Giselsson, 2023].

Cycles

Heavy-ball

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

Pick specific (α, β) and fix cycle length K.

Cycles

Heavy-ball

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

Pick specific (α, β) and fix cycle length K.

Look for non-trivial cycles of length $K \in \mathbb{N}$ by solving:

$$\begin{aligned}
\min_{\substack{f \\ x_0,\ldots,x_{K+1}}} \quad & \|x_K - x_0\|^2 + \|x_{K+1} - x_1\|^2 & \\
\text{s.t.} \quad & f \in \mathcal{F}_{\mu,L} & \text{Functional class} \\
& x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}) & \text{Algorithm} \\
& \|x_0 - x_1\|^2 \geqslant 1 & \text{Non-trivial cycle}
\end{aligned}$$

Cycles

Heavy-ball

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

Pick specific (α, β) and fix cycle length K.

Look for non-trivial cycles of length $K \in \mathbb{N}$ by solving:

$$\begin{aligned}
\min_{\substack{f \\ x_0,\ldots,x_{K+1}}} \quad & \|x_K - x_0\|^2 + \|x_{K+1} - x_1\|^2 & \\
\text{s.t.} \quad & f \in \mathcal{F}_{\mu,L} & \text{Functional class} \\
& x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}) & \text{Algorithm} \\
& \|x_0 - x_1\|^2 \geqslant 1 & \text{Non-trivial cycle}
\end{aligned}$$

From same steps as before $\to$ SDP formulation $\to$ LP (via convexity and symmetries).

Heavy-ball method: Lyapunov vs. cycles

Heavy-ball: $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$. Choices of (α, β) for convergence?[7,8,9]

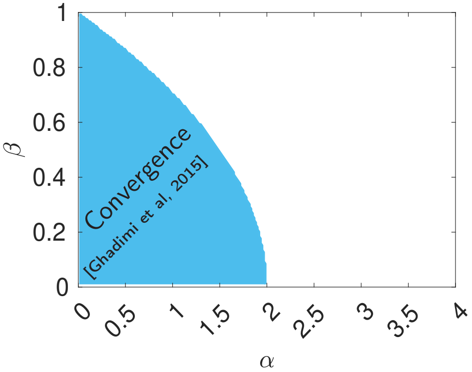
[7] Classical region from [Ghadimi, Feyzmahdavian, Johansson, 2015]

[8] Known 3-cycle for optimal quadratic tuning of HB when used beyond quadratics [Lessard, Recht, Packard, 2016].

[9] Goujaud, T, Dieuleveut (2023). "Provable non-accelerations of the heavy-ball method." Preprint.

Heavy-ball method: Lyapunov vs. cycles

Heavy-ball: $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$. Choices of (α, β) for convergence?[7,8,9]



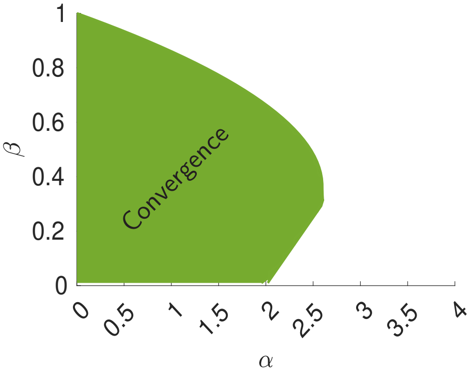
[7] Classical region from [Ghadimi, Feyzmahdavian, Johansson, 2015]

[8] Known 3-cycle for optimal quadratic tuning of HB when used beyond quadratics [Lessard, Recht, Packard, 2016].

[9] Goujaud, T, Dieuleveut (2023). "Provable non-accelerations of the heavy-ball method." Preprint.

Heavy-ball method: Lyapunov vs. cycles

Heavy-ball: $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$. Choices of (α, β) for convergence?[7,8,9]



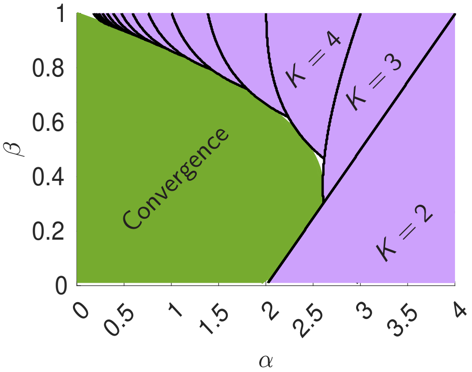
[7] Classical region from [Ghadimi, Feyzmahdavian, Johansson, 2015]

[8] Known 3-cycle for optimal quadratic tuning of HB when used beyond quadratics [Lessard, Recht, Packard, 2016].

[9] Goujaud, T, Dieuleveut (2023). "Provable non-accelerations of the heavy-ball method." Preprint.

Heavy-ball method: Lyapunov vs. cycles

Heavy-ball: $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$. Choices of (α, β) for convergence?[7,8,9]



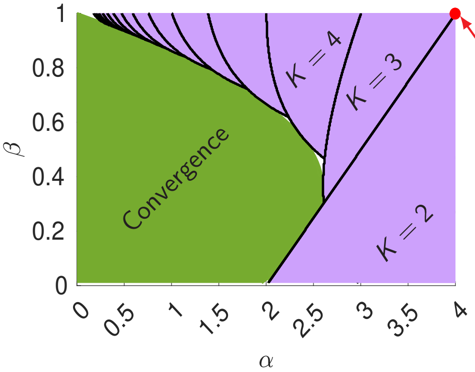
[7] Classical region from [Ghadimi, Feyzmahdavian, Johansson, 2015]

[8] Known 3-cycle for optimal quadratic tuning of HB when used beyond quadratics [Lessard, Recht, Packard, 2016].

[9] Goujaud, T, Dieuleveut (2023). "Provable non-accelerations of the heavy-ball method." Preprint.

Heavy-ball method: Lyapunov vs. cycles

Heavy-ball: $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$. Choices of (α, β) for convergence?[7,8,9]



"Optimal tuning" for quadratic optimization[8]
(numerics for $L/\mu = 10^7$)

---

[7] Classical region from [Ghadimi, Feyzmahdavian, Johansson, 2015]

[8] Known 3-cycle for optimal quadratic tuning of HB when used beyond quadratics [Lessard, Recht, Packard, 2016].

[9] Goujaud, T, Dieuleveut (2023). "Provable non-accelerations of the heavy-ball method." Preprint.

Recap'

Recap'

- ☺ Smaller-dimensional certification problems.
- ☺ Broader sets of scenarios within reach.
- ☺ Simpler analysis structures, more likely to be human-readable.
- ☹ Tightness is lost ("best certification with quadratic Lyapunov functions", instead).
- ☹ Proofs (may be) involved and hard to intuit.
- ☹ Proofs (may be) hard to generalize.

A few references

- T, Van Scoy, Lessard (2018). "Lyapunov functions for first-order methods: Tight automated convergence guarantees." International Conference on Machine Learning (ICML).

- T, Bach (2019). "Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions." Conference on Learning Theory (COLT).

- d'Aspremont, Scieur, Taylor (2021). "Acceleration methods." Foundations and Trends in Optimization 5(1-2).

- Moucer, T, Bach (2023). "A systematic approach to Lyapunov analyses of continuous-time models in convex optimization." SIAM Journal on Optimization 33(3).

- Goujaud, T, Dieuleveut (2023). "Provable non-accelerations of the heavy-ball method." Preprint.

- Upadhyaya, Banert, T, Giselsson (2025). "Automated tight Lyapunov analysis for first-order methods." Mathematical Programming.

Constructive approach to performance analysis

Towards structured analyses

Towards optimal algorithms

Concluding remarks

Creating new algorithms

A "generic" first-order method

$$
\begin{aligned}
w_1 &= w_0 - \alpha_{1,0}\nabla f(w_0) \\
w_2 &= w_1 - \alpha_{2,0}\nabla f(w_0) - \alpha_{2,1}\nabla f(w_1) \\
w_3 &= w_2 - \alpha_{3,0}\nabla f(w_0) - \alpha_{3,1}\nabla f(w_1) - \alpha_{3,2}\nabla f(w_2) \\
&\vdots \\
w_N &= w_{N-1} - \sum_{i=0}^{N-1} \alpha_{N,i}\nabla f(w_i),
\end{aligned}
\tag{FOM}
$$

for some coefficients $\{\alpha_{i,j}\}$. Generic **non-adaptive** first-order method.

Creating new algorithms

A "generic" first-order method

$$
\begin{aligned}
w_1 &= w_0 - \alpha_{1,0} \nabla f(w_0) \\
w_2 &= w_1 - \alpha_{2,0} \nabla f(w_0) - \alpha_{2,1} \nabla f(w_1) \\
w_3 &= w_2 - \alpha_{3,0} \nabla f(w_0) - \alpha_{3,1} \nabla f(w_1) - \alpha_{3,2} \nabla f(w_2) \\
&\vdots \\
w_N &= w_{N-1} - \sum_{i=0}^{N-1} \alpha_{N,i} \nabla f(w_i),
\end{aligned}
\tag{FOM}
$$

for some coefficients $\{\alpha_{i,j}\}$. Generic **non-adaptive** first-order method.

How to choose $\{\alpha_{i,j}\}$?

Creating new algorithms

A "generic" first-order method

$$
\begin{aligned}
w_1 &= w_0 - \alpha_{1,0}\nabla f(w_0) \\
w_2 &= w_1 - \alpha_{2,0}\nabla f(w_0) - \alpha_{2,1}\nabla f(w_1) \\
w_3 &= w_2 - \alpha_{3,0}\nabla f(w_0) - \alpha_{3,1}\nabla f(w_1) - \alpha_{3,2}\nabla f(w_2) \\
&\ \vdots \\
w_N &= w_{N-1} - \sum_{i=0}^{N-1} \alpha_{N,i}\nabla f(w_i),
\end{aligned}
\tag{FOM}
$$

for some coefficients $\{\alpha_{i,j}\}$. Generic **non-adaptive** first-order method.

How to choose $\{\alpha_{i,j}\}$?

- ◇ pick a performance criterion, for instance $\frac{\|w_N - w_\star\|^2}{\|w_0 - w_\star\|^2}$,

Creating new algorithms

A "generic" first-order method

$$\begin{aligned} w_1 &= w_0 - \alpha_{1,0}\nabla f(w_0) \\ w_2 &= w_1 - \alpha_{2,0}\nabla f(w_0) - \alpha_{2,1}\nabla f(w_1) \\ w_3 &= w_2 - \alpha_{3,0}\nabla f(w_0) - \alpha_{3,1}\nabla f(w_1) - \alpha_{3,2}\nabla f(w_2) \\ &\vdots \\ w_N &= w_{N-1} - \sum_{i=0}^{N-1} \alpha_{N,i}\nabla f(w_i), \end{aligned} \tag{FOM}$$

for some coefficients $\{\alpha_{i,j}\}$. Generic **non-adaptive** first-order method.

How to choose $\{\alpha_{i,j}\}$?

- ◇ pick a performance criterion, for instance $\frac{\|w_N - w_\star\|^2}{\|w_0 - w_\star\|^2}$,
- ◇ solve the minimax (minimize worst-case): $\min_{\{\alpha_{i,j}\}_{i,j}} \max_{f\in\mathcal{F},\{w_i\}} \frac{\|w_N - w_\star\|^2}{\|w_0 - w_\star\|^2}$.

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|w_N - w_\star\|^2}{\|w_0 - w_\star\|^2}?$$

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{\|w_N - w_\star\|^2}{\|w_0 - w_\star\|^2}? \qquad \min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{f(w_N) - f(w_\star)}{f(w_0) - f(w_\star)}?$$

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{\|w_N - w_\star\|^2}{\|w_0 - w_\star\|^2}? \quad \min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{f(w_N) - f(w_\star)}{f(w_0) - f(w_\star)}? \quad \min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{\|\nabla f(w_N)\|^2}{\|\nabla f(w_0)\|^2}?$$

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{\|w_N - w_\star\|^2}{\|w_0 - w_\star\|^2}? \quad \min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{f(w_N) - f(w_\star)}{f(w_0) - f(w_\star)}? \quad \min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{\|\nabla f(w_N)\|^2}{\|\nabla f(w_0)\|^2}?$$

◇ Convex relaxations,

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|w_N - w_\star\|^2}{\|w_0 - w_\star\|^2}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{f(w_N) - f(w_\star)}{f(w_0) - f(w_\star)}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|\nabla f(w_N)\|^2}{\|\nabla f(w_0)\|^2}?$$

- ◇ Convex relaxations,
- ◇ analogies (e.g., with conjugate gradient methods)

$$w_{k+1} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{f(w) : \; w \in w_0 + \mathrm{span}\{\nabla f(w_0), \ldots, \nabla f(w_k)\}\},$$

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{\|w_N - w_\star\|^2}{\|w_0 - w_\star\|^2}? \quad \min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{f(w_N) - f(w_\star)}{f(w_0) - f(w_\star)}? \quad \min_{\{\alpha_{i,j}\}} \max_{f\in\mathcal{F}} \frac{\|\nabla f(w_N)\|^2}{\|\nabla f(w_0)\|^2}?$$

- ◇ Convex relaxations,
- ◇ analogies (e.g., with conjugate gradient methods)

$$w_{k+1} \in \operatorname*{argmin}_{x\in\mathbb{R}^d} \{f(w) : \; w \in w_0 + \operatorname{span}\{\nabla f(w_0), \ldots, \nabla f(w_k)\}\},$$

- ◇ brutal approaches.

Numerical example I

Worst-case performance $\frac{f(w_N)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

Numerical example I

Worst-case performance $\frac{f(w_N)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,

Numerical example I

Worst-case performance $\frac{f(w_N)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),

Numerical example I

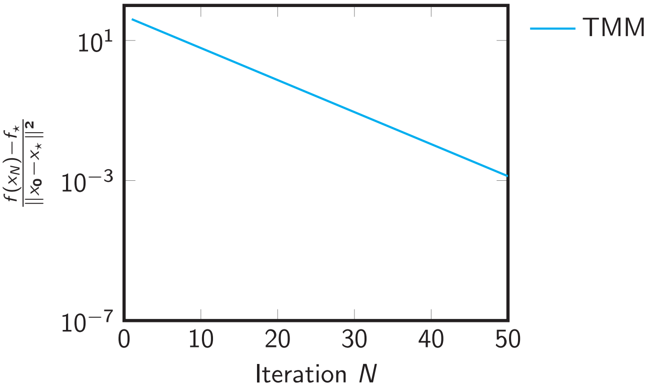
Worst-case performance $\frac{f(w_N)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- Lower complexity bound (numerically generated).

Numerical example I

Worst-case performance $\frac{f(w_N)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

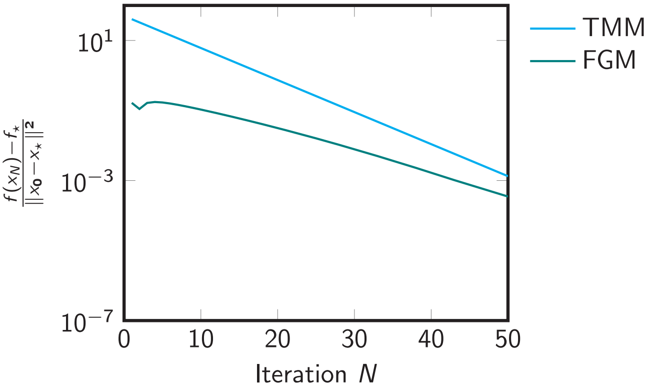
- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- Lower complexity bound (numerically generated).



Numerical example I

Worst-case performance $\frac{f(w_N)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

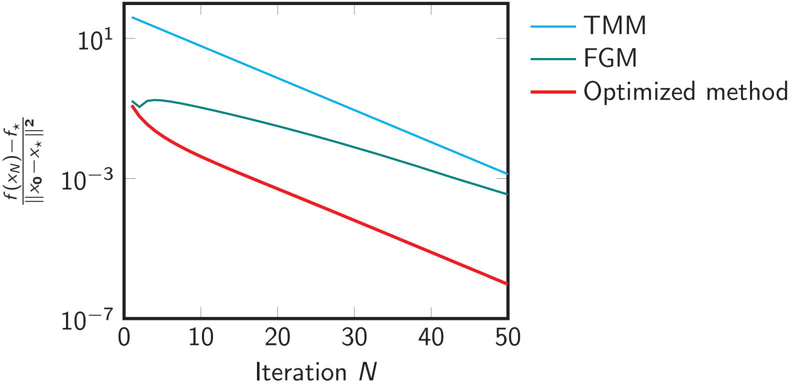
- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- Lower complexity bound (numerically generated).



Numerical example I

Worst-case performance $\frac{f(w_N)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

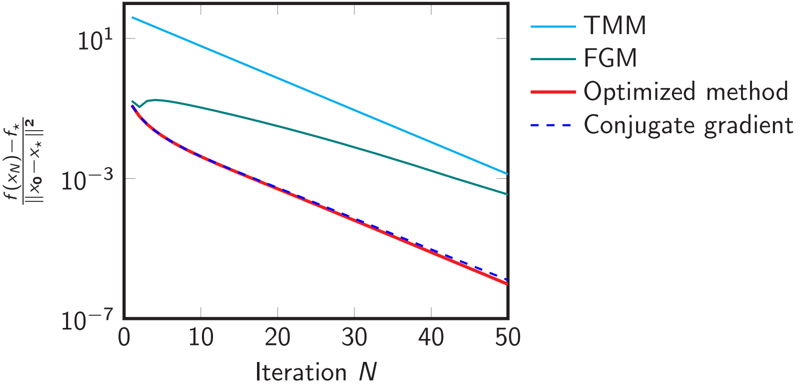
- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- Lower complexity bound (numerically generated).



Numerical example I

Worst-case performance $\frac{f(w_N)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

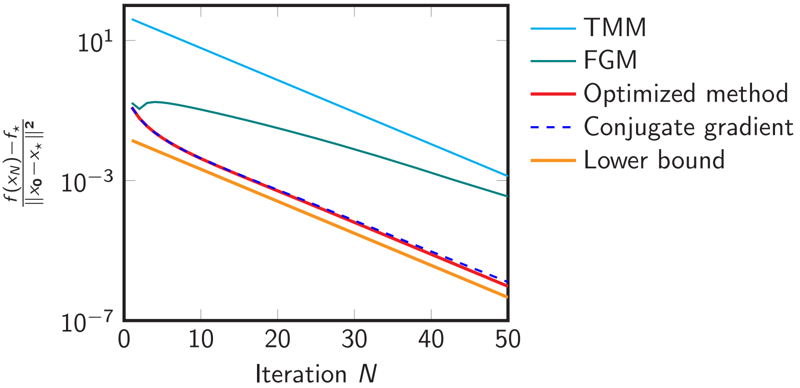
- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- Lower complexity bound (numerically generated).



Numerical example I

Worst-case performance $\frac{f(w_N)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- Lower complexity bound (numerically generated).



Example II: Information-Theoretic Exact Method (ITEM)

Optimal method for $\frac{\|z_N - z_\star\|^2}{\|z_0 - z_\star\|^2}$ is "Information-Theoretic Exact Method":[10]

[10] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization." Mathematical Programming 199(1).

Example II: Information-Theoretic Exact Method (ITEM)

Optimal method for $\frac{\|z_N - z_\star\|^2}{\|z_0 - z_\star\|^2}$ is "Information-Theoretic Exact Method":[10]

$$
\begin{aligned}
y_k &= (1-\beta_k) z_k + \beta_k \left(y_{k-1} - \frac{1}{L} \nabla f(y_{k-1}) \right) \\
z_{k+1} &= (1 - \tfrac{\mu}{L}\delta_k) z_k + \tfrac{\mu}{L}\delta_k \left(y_k - \frac{1}{\mu} \nabla f(y_k) \right),
\end{aligned}
$$

[10] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization." Mathematical Programming 199(1).

Example II: Information-Theoretic Exact Method (ITEM)

Optimal method for $\frac{\|z_N - z_\star\|^2}{\|z_0 - z_\star\|^2}$ is "Information-Theoretic Exact Method":[10]

$$
\begin{aligned}
y_k &= (1-\beta_k) z_k + \beta_k \left(y_{k-1} - \frac{1}{L} \nabla f(y_{k-1}) \right) \\
z_{k+1} &= (1 - \tfrac{\mu}{L}\delta_k) z_k + \tfrac{\mu}{L}\delta_k \left(y_k - \frac{1}{\mu} \nabla f(y_k) \right),
\end{aligned}
$$

where the sequences $\{\beta_k\}$ and $\{\delta_k\}$ depends on some external sequence

$$
A_{k+1} = \frac{(1+\frac{\mu}{L})A_k + 2\left(1 + \sqrt{(1+A_k)(1+\frac{\mu}{L}A_k)}\right)}{(1-\frac{\mu}{L})^2}, \quad k \geqslant 0,
$$

with $A_0 = 0$.

[10] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization." Mathematical Programming 199(1).

Example II: Information-Theoretic Exact Method (ITEM)

Optimal method for $\frac{\|z_N - z_\star\|^2}{\|z_0 - z_\star\|^2}$ is "Information-Theoretic Exact Method":[10]

$$y_k = (1-\beta_k) z_k + \beta_k \left(y_{k-1} - \frac{1}{L} \nabla f(y_{k-1}) \right)$$
$$z_{k+1} = (1 - \tfrac{\mu}{L}\delta_k) z_k + \tfrac{\mu}{L}\delta_k \left(y_k - \frac{1}{\mu} \nabla f(y_k) \right),$$

where the sequences $\{\beta_k\}$ and $\{\delta_k\}$ depends on some external sequence

$$A_{k+1} = \frac{(1+\frac{\mu}{L})A_k + 2\left(1 + \sqrt{(1+A_k)(1+\frac{\mu}{L}A_k)}\right)}{(1-\frac{\mu}{L})^2}, \quad k \geqslant 0,$$

with $A_0 = 0$. The (tight) guarantee is $\frac{\|z_N - z_\star\|^2}{\|z_0 - z_\star\|^2} \leqslant \frac{1}{1+\frac{\mu}{L}A_N} = O\left(\left(1 - \sqrt{\frac{\mu}{L}}\right)^{2N}\right)$. Matches exact lower bound.[11]

---

[10] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization." Mathematical Programming 199(1).

[11] Drori, T (2022). "On the oracle complexity of smooth strongly convex minimization." Journal of Complexity 68.

Example III: Projection-free online learning

[12] Elad Hazan (2016). "Introduction to Online Convex Optimization." Foundations and Trends in Optimization.

[13] Weibel, Gaillard, Koolen, T (2025). "Optimized projection-free algorithms for onlinelearning: construction and worst-case analysis."

Example III: Projection-free online learning

Online Frank–Wolfe algorithm[12,13]

Input: closed convex set $\mathcal{K}$, initial guess $x_1 \in \mathcal{K}$, sequence of costs $\ell_1, \ell_2, \ldots$.

For $t = 1, 2, \ldots$

Play x_t, pay cost $\ell_t(x_t)$, and observe $g_t = \nabla \ell_t(x_t)$.

$$\mathrm{dir}_t = \sum_{s=1}^{t} \eta_{t,s}\, g_s + \sum_{s=1}^{t-1} \beta_{t,s}\,(v_s - x_1)$$

$$v_t = \operatorname*{argmin}_{v \in \mathcal{K}} \langle \mathrm{dir}_t, v \rangle$$

$$x_{t+1} = x_1 + \sum_{s=1}^{t} \gamma_{t+1,s}\,(v_s - x_1),$$

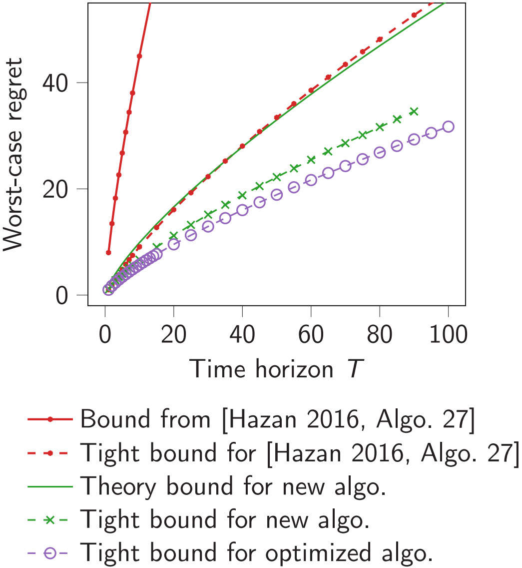
Target: good regret bounds $\rightarrow$ optimize (minimize) worst-case by appropriate choices $\eta_{t,s}, \beta_{t,s}, \gamma_{t,s}$.

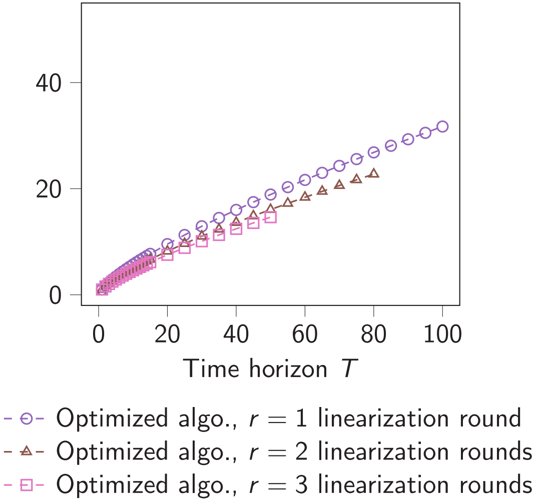
$$R_T(x_1, \ldots, x_T; x_\star) \triangleq \frac{1}{T} \sum_{t=1}^{T} \left\{ \ell_t(x_t) - \ell_t(x_\star) \right\} \leqslant \frac{1}{T} \sup_{x \in \mathcal{K}} \left\{ \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(x) \right\}.$$

[12] Elad Hazan (2016). "Introduction to Online Convex Optimization." Foundations and Trends in Optimization.

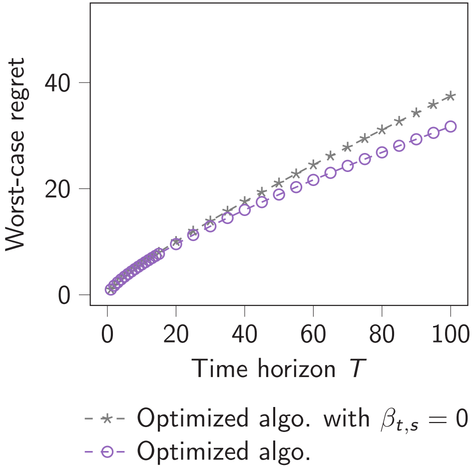
[13] Weibel, Gaillard, Koolen, T (2025). "Optimized projection-free algorithms for onlinelearning: construction and worst-case analysis."

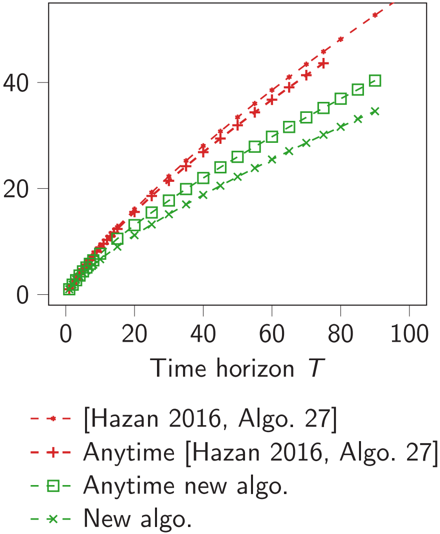
Numerically optimized online Frank-Wolfe





Numerically optimized online Frank-Wolfe





A few instructive examples

Design first-order methods via PEPs:

- Drori, Teboulle (2014). "Performance of first-order methods for smooth convex minimization: a novel approach." Mathematical Programming 145(1).
- Kim, Fessler (2016). "Optimized methods for smooth convex optimization." Mathematical programming 159.
- Van Scoy, Freeman, Lynch (2017). "The fastest known globally convergent first-order method for minimizing strongly convex functions." IEEE Control Systems Magazine 39(3).
- Kim, Fessler (2021). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions." Journal of Optimization Theory and Applications 188(1).
- Altschuler, Parrilo (2023). "Acceleration by Stepsize Hedging I: Multi-Step Descent and the Silver Stepsize Schedule." Preprint.

A few instructive examples

Design first-order methods via PEPs:

- ◇ Drori, Teboulle (2014). "Performance of first-order methods for smooth convex minimization: a novel approach." Mathematical Programming 145(1).
- ◇ Kim, Fessler (2016). "Optimized methods for smooth convex optimization." Mathematical programming 159.
- ◇ Van Scoy, Freeman, Lynch (2017). "The fastest known globally convergent first-order method for minimizing strongly convex functions." IEEE Control Systems Magazine 39(3).
- ◇ Kim, Fessler (2021). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions." Journal of Optimization Theory and Applications 188(1).
- ◇ Altschuler, Parrilo (2023). "Acceleration by Stepsize Hedging I: Multi-Step Descent and the Silver Stepsize Schedule." Preprint.

... including "brutal" examples:

- ◇ Grimmer (2024). "Provably faster gradient descent via long steps." SIAM Journal on Optimization 34(3).
- ◇ Gupta, Van Parijs, Ryu (2024). "Branch-and-Bound Performance Estimation Programming: A Unified Methodology for Constructing Optimal Methods." Mathematical Programming 204(1).

A few references

◇ T, Bach (2019). "Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions." Conference on Learning Theory (COLT).

◇ Drori, T (2020). "Efficient first-order methods for convex minimization: a constructive approach." Mathematical Programming 184(1).

◇ Drori, T (2022). "On the oracle complexity of smooth strongly convex minimization." Journal of Complexity 68.

◇ Barré, T, Bach (2023). "Principled analyses and design of first-order methods with inexact proximal operators." Mathematical Programming 201(1).


◇ T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization." Mathematical Programming 199(1).

◇ Weibel, Gaillard, Koolen, T (2025). "Optimised projection-free algorithms for online learning: construction and worst-case analysis."



Constructive approach to performance analysis

Towards structured analyses

Towards optimal algorithms

Concluding remarks

Concluding remarks

Performance estimation's philosophy

Concluding remarks

Performance estimation's philosophy

- numerically allows obtaining tight bounds (rigorous baselines),
  - fast prototyping
  - worth checking before trying to prove a method works.

Concluding remarks

Performance estimation's philosophy

- numerically allows obtaining tight bounds (rigorous baselines),
  - fast prototyping
  - worth checking before trying to prove a method works.
- algebraic insights into performance analyses: principled approach,
  - analyses are dual feasible points,
  - analyses are linear combinations of certain specific inequalities.

Concluding remarks

Performance estimation's philosophy

- numerically allows obtaining tight bounds (rigorous baselines),
  - fast prototyping
  - worth checking before trying to prove a method works.
- algebraic insights into performance analyses: principled approach,
  - analyses are dual feasible points,
  - analyses are linear combinations of certain specific inequalities.

Byproducts:

- computer-assisted design of analyses,
- computer-assisted design of numerical methods,
- step towards reproducible theory
  - validation & benchmark tool for proofs (also for reviews ☺),
  - complements existing open-source initiatives.

Take-home messages

Optimization can be seen as the science of proving inequalities

...including complexity bounds for numerical methods.

Powerful framework for designing methods and guarantees.

Thanks! Questions?

PerformanceEstimation/Performance-Estimation-Toolbox on Github

PerformanceEstimation/PEPit on Github