Constructive approaches to the analysis and design of first-order methods for optimization

Adrien Taylor

DIPopt workshop – November 2023

François Glineur

Julien Hendrickx

Etienne de Klerk

Ernest Ryu

Carolina Bergeling

Pontus Giselsson

Francis Bach

Jérôme Bolte

Yoel Drori

Alexandre d'Aspremont

Mathieu Barré

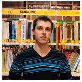
Radu Dragomir

Bryan Van Scoy

Laurent Lessard

Aymeric Dieuleveut

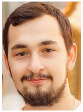
Céline Moucer



Baptiste Goujaud

Sebastian Banert

Manu Uphadyaya



Eduard Gorbunov

Gauthier Gidel

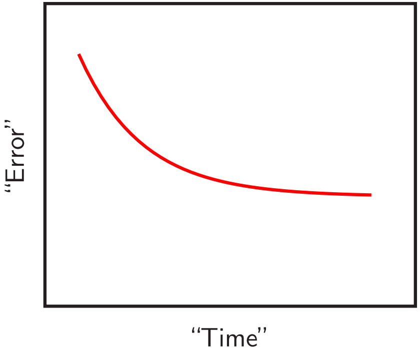
Many optimization schemes: usages depend on application requirements (target precision, time budget, memory budget,...).

Many optimization schemes: usages depend on application requirements (target precision, time budget, memory budget,...).

Can we predict their behaviors?

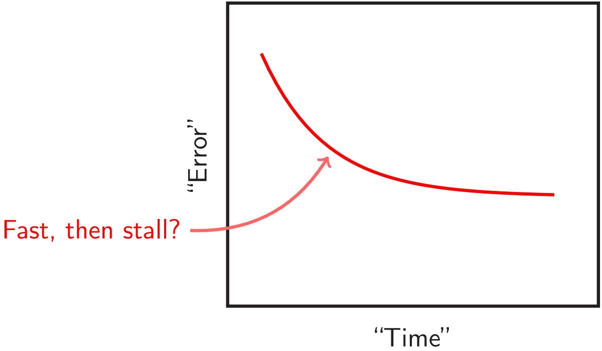
Many optimization schemes: usages depend on application requirements (target precision, time budget, memory budget,...).

Can we predict their behaviors?



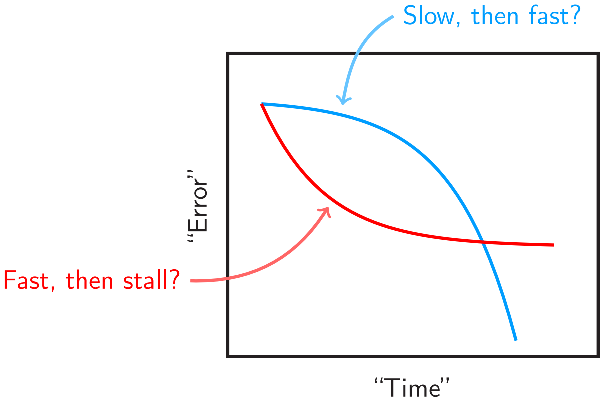
Many optimization schemes: usages depend on application requirements (target precision, time budget, memory budget,...).

Can we predict their behaviors?



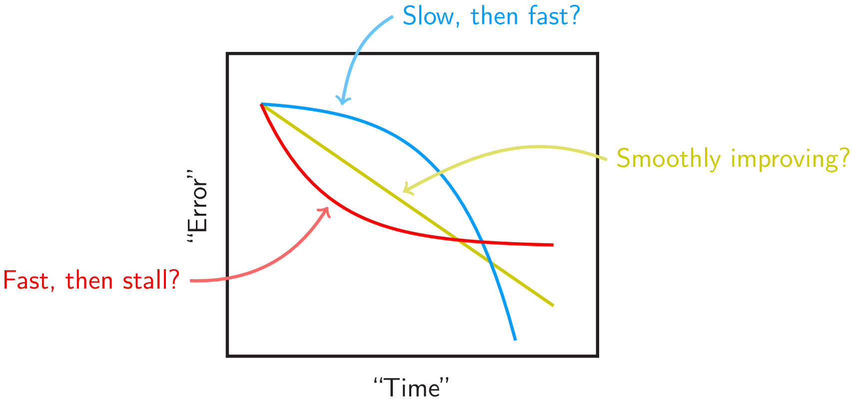
Many optimization schemes: usages depend on application requirements (target precision, time budget, memory budget,...).

Can we predict their behaviors?



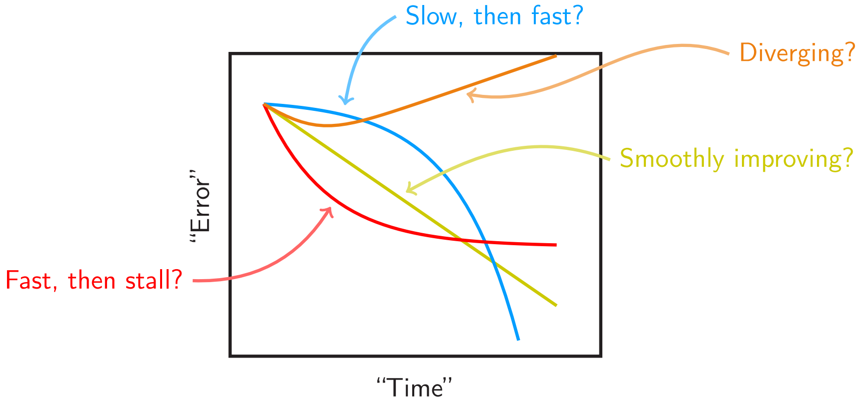
Many optimization schemes: usages depend on application requirements (target precision, time budget, memory budget,...).

Can we predict their behaviors?



Many optimization schemes: usages depend on application requirements (target precision, time budget, memory budget,...).

Can we predict their behaviors?



How to show that an algorithm works?

How to show that an algorithm works?

Here: principled approach to worst-case analysis.

Important inspiration & reference:

- ◇ Drori, and Teboulle ('14). "Performance of first-order methods for smooth convex minimization: a novel approach."

Important inspiration & reference:

- ◇ Drori, and Teboulle ('14). "Performance of first-order methods for smooth convex minimization: a novel approach."

First part of the presentation:

- ◇ T., Hendrickx, Glineur ('17). "Smooth strongly convex interpolation and exact worst-case performance of first-order methods."
- ◇ T., Hendrickx, Glineur ('17). "Exact worst-case performance of first-order methods for composite convex optimization."
- ◇ T., Hendrickx, Glineur ('17). "Performance estimation toolbox (PESTO): Automated worst-case analysis of first-order optimization methods."
- ◇ Goujaud, Moucer, et al. ('22). "PEPit: computer-assisted worst-case analyses of first-order optimization methods in Python."

Important inspiration & reference:

- ◇ Drori, and Teboulle ('14). "Performance of first-order methods for smooth convex minimization: a novel approach."

First part of the presentation:

- ◇ T., Hendrickx, Glineur ('17). "Smooth strongly convex interpolation and exact worst-case performance of first-order methods."
- ◇ T., Hendrickx, Glineur ('17). "Exact worst-case performance of first-order methods for composite convex optimization."
- ◇ T., Hendrickx, Glineur ('17). "Performance estimation toolbox (PESTO): Automated worst-case analysis of first-order optimization methods."
- ◇ Goujaud, Moucer, et al. ('22). "PEPit: computer-assisted worst-case analyses of first-order optimization methods in Python."

Second part

- ◇ Drori, T ('20). "Efficient first-order methods for convex minimization: a constructive approach."
- ◇ Drori, T ('22). "On the oracle complexity of smooth strongly convex minimization."
- ◇ T, Drori ('23). "An optimal gradient method for smooth strongly convex minimization."

Informal introduction: `https://francisbach.com/computer-aided-analyses/`.

Example: minimize differentiable $f : \mathbb{R}^d \to \mathbb{R}$:

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

where f is L-smooth and μ-strongly convex ($0 \leqslant \mu \leqslant L < \infty$).

Example: minimize differentiable $f : \mathbb{R}^d \to \mathbb{R}$:

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

where f is L-smooth and μ-strongly convex ($0 \leqslant \mu \leqslant L < \infty$).

Use gradient descent:

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

Example: minimize differentiable $f : \mathbb{R}^d \to \mathbb{R}$:

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

where f is L-smooth and μ-strongly convex ($0 \leqslant \mu \leqslant L < \infty$).

Use gradient descent:

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

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

Example: minimize differentiable $f : \mathbb{R}^d \to \mathbb{R}$:

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

where f is L-smooth and μ-strongly convex ($0 \leqslant \mu \leqslant L < \infty$).

Use gradient descent:

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

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

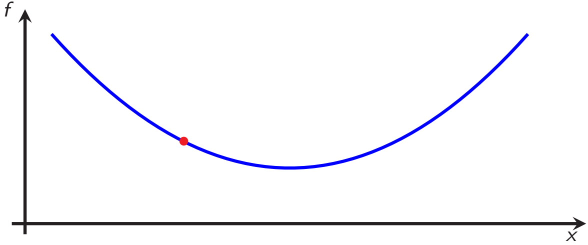
Examples: how small should $f(x_N) - f(x_\star)$, $\|\nabla f(x_N)\|$, $\|x_N - x_\star\|$ be?

About the assumptions

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

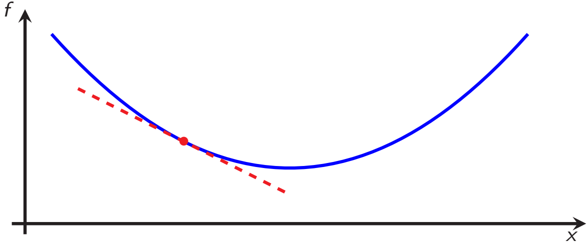
About the assumptions

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

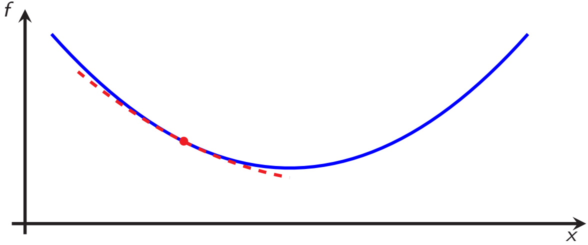
A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$:



(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 $\forall x, y \in \mathbb{R}^d$:

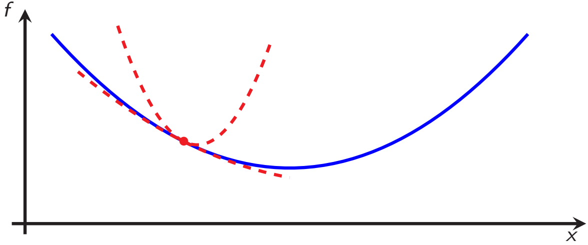


(1) (Convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle$,

(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 $\forall x, y \in \mathbb{R}^d$:



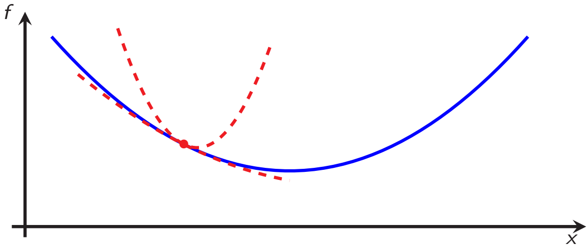
(1) (Convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle$,

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

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

Toy example, take I: find τ such that:

$$\|x_{k+1} - x_\star\|^2 \leqslant \tau \|x_k - x_\star\|^2,$$

for all

Toy example, take I: find τ such that:

$$\|x_{k+1} - x_\star\|^2 \leqslant \tau \|x_k - x_\star\|^2,$$

for all

- L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,\mathrm{L}}$),
- x_{k+1} generated by gradient step $x_{k+1} = x_k - h\nabla f(x_k)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$.

Toy example, take I: find τ such that:

$$\|x_{k+1} - x_\star\|^2 \leqslant \tau \|x_k - x_\star\|^2,$$

for all

- L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$),
- x_{k+1} generated by gradient step $x_{k+1} = x_k - h\nabla f(x_k)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$.

$$\|x_{k+1} - x_\star\|^2$$

Toy example, take I: find τ such that:

$$\|x_{k+1} - x_\star\|^2 \leqslant \tau \|x_k - x_\star\|^2,$$

for all

- $\diamond$ L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,\mathrm{L}}$),
- $\diamond$ x_{k+1} generated by gradient step $x_{k+1} = x_k - h\nabla f(x_k)$,
- $\diamond$ $x_\star = \underset{x}{\operatorname{argmin}}\, f(x)$.

$$\|x_{k+1} - x_\star\|^2 = \|x_k - x_\star\|^2 - 2h\langle \nabla f(x_k); x_k - x_\star\rangle + h^2\|\nabla f(x_k)\|^2$$

Toy example, take I: find τ such that:

$$\|x_{k+1} - x_\star\|^2 \leqslant \tau \|x_k - x_\star\|^2,$$

for all

- $\diamond$ L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,\mathrm{L}}$),
- $\diamond$ x_{k+1} generated by gradient step $x_{k+1} = x_k - h\nabla f(x_k)$,
- $\diamond$ $x_\star = \operatorname{argmin}_x f(x)$.

$$\|x_{k+1} - x_\star\|^2 = \|x_k - x_\star\|^2 - 2h\langle \nabla f(x_k); x_k - x_\star\rangle + h^2\|\nabla f(x_k)\|^2$$

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

Toy example, take I: find τ such that:

$$\|x_{k+1}-x_\star\|^2 \leqslant \tau\|x_k-x_\star\|^2,$$

for all

- $\diamond$ L-smooth and μ-strongly convex function f (notation $f\in\mathcal{F}_{\mu,\mathrm{L}}$),
- $\diamond$ x_{k+1} generated by gradient step $x_{k+1}=x_k-h\nabla f(x_k)$,
- $\diamond$ $x_\star=\underset{x}{\operatorname{argmin}}\ f(x)$.

$$\|x_{k+1}-x_\star\|^2 = \|x_k-x_\star\|^2 - 2h\langle\nabla f(x_k);\, x_k-x_\star\rangle + h^2\|\nabla f(x_k)\|^2$$

$\Big\downarrow$ inequality (1&2)

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

Toy example, take I: find τ such that:

$$\|x_{k+1} - x_\star\|^2 \leqslant \tau \|x_k - x_\star\|^2,$$

for all

- $\diamond$ L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,\mathrm{L}}$),
- $\diamond$ x_{k+1} generated by gradient step $x_{k+1} = x_k - h\nabla f(x_k)$,
- $\diamond$ $x_\star = \underset{x}{\operatorname{argmin}}\, f(x)$.

$$\|x_{k+1} - x_\star\|^2 = \|x_k - x_\star\|^2 - 2h\langle \nabla f(x_k);\, x_k - x_\star\rangle + h^2 \|\nabla f(x_k)\|^2$$

$\Big\downarrow$ inequality (1&2)

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

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

Toy example, take I: find τ such that:

$$\|x_{k+1} - x_\star\|^2 \leqslant \tau \|x_k - x_\star\|^2,$$

for all

- $\diamond$ L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,\mathrm{L}}$),
- $\diamond$ x_{k+1} generated by gradient step $x_{k+1} = x_k - h\nabla f(x_k)$,
- $\diamond$ $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$.

$$\|x_{k+1} - x_\star\|^2 = \|x_k - x_\star\|^2 - 2h\langle \nabla f(x_k);\, x_k - x_\star\rangle + h^2\|\nabla f(x_k)\|^2$$

$\downarrow$ inequality (1&2)

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

$\downarrow$ if $0 \leqslant h \leqslant \frac{2}{L+\mu}$

$$\leqslant (1 - h\mu)^2\|x_k - x_\star\|^2.$$

Legitimate questions:

Legitimate questions:

- ◇ anything improvable? Realistic analyses?

Legitimate questions:

- ◇ anything improvable? Realistic analyses?
- ◇ How to choose the right inequalities to combine?

Legitimate questions:

- ◇ anything improvable? Realistic analyses?
- ◇ How to choose the right inequalities to combine?
- ◇ Why studying this specific quantity?

Legitimate questions:

- ◇ anything improvable? Realistic analyses?
- ◇ How to choose the right inequalities to combine?
- ◇ Why studying this specific quantity?
- ◇ How to study other quantities, e.g., $f(x_k) - f(x_\star)$?

Legitimate questions:

- $\diamond$ anything improvable? Realistic analyses?
- $\diamond$ How to choose the right inequalities to combine?
- $\diamond$ Why studying this specific quantity?
- $\diamond$ How to study other quantities, e.g., $f(x_k) - f(x_\star)$?
- $\diamond$ Unique way to arrive to the desired result?

Legitimate questions:

- $\diamond$ anything improvable? Realistic analyses?
- $\diamond$ How to choose the right inequalities to combine?
- $\diamond$ Why studying this specific quantity?
- $\diamond$ How to study other quantities, e.g., $f(x_k) - f(x_\star)$?
- $\diamond$ Unique way to arrive to the desired result?
- $\diamond$ How likely are we to find such proofs in more complicated cases?

Take-home messages

Worst-cases are solutions to optimization problems.

Take-home messages

Worst-cases are solutions to optimization problems.

Sometimes, those optimization problems are tractable.

Take-home messages

Worst-cases are solutions to optimization problems.

Sometimes, those optimization problems are tractable.

Often tractable for first-order methods in convex optimization!

Take-home messages

Worst-cases are solutions to optimization problems.

Sometimes, those optimization problems are tractable.

Often tractable for first-order methods in convex optimization!

Acceleration/optimal methods by optimizing worst-cases.

Example

Software

Step-size optimization

Concluding remarks

Example

Software

Step-size optimization

Concluding remarks

Convergence rate of a gradient step

Convergence rate of a gradient step

Toy example, take II: What is the smallest τ such that:

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

for all

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

Convergence rate of a gradient step

Toy example, take II: What is the smallest τ such that:

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

for all

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

First: let's compute τ!

Convergence rate of a gradient step

Toy example, take II: What is the smallest τ such that:

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

for all

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

First: let's compute τ!

$$\tau(\mu, L, h) = \max_{f, x_0, x_1, x_\star} \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

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

Convergence rate of a gradient step

Toy example, take II: What is the smallest τ such that:

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

for all

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

First: let's compute τ!

$$\tau(\mu, L, h) = \max_{f, x_0, x_1, x_\star} \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

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

$$x_1 = x_0 - h\nabla f(x_0) \qquad \text{Algorithm}$$

Convergence rate of a gradient step

Toy example, take II: What is the smallest τ such that:

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

for all

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

First: let's compute τ!

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

Convergence rate of a gradient step

Toy example, take II: What is the smallest τ such that:

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

for all

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

First: let's compute τ!

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

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

Convergence rate of a gradient step

Toy example, take II: What is the smallest τ such that:

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

for all

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

First: let's compute τ!

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

Variables: f, x_0, x_1, $x_\star$; parameters: μ, L, h.

Sampled version

Sampled version

- Performance estimation problem:

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

Sampled version

- Performance estimation problem:

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

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

Sampled version

- Performance estimation problem:

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

- Variables: f, x_0, x_1, $x_\star$.
- Sampled version: f is only used at x_0 and $x_\star$ (no need to sample other points)

Sampled version

- Performance estimation problem:

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

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

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

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

- Variables: f, x_0, x_1, $x_\star$.
- Sampled version: f is only used at x_0 and $x_\star$ (no need to sample other points)

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

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

$$x_1 = x_0 - hg_0$$

$$g_\star = 0.$$

Sampled version

- Performance estimation problem:

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

- Variables: f, x_0, x_1, $x_\star$.
- Sampled version: f is only used at x_0 and $x_\star$ (no need to sample other points)

$$\begin{aligned} \max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad & \frac{\|x_1 - x_0\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases} \\ & x_1 = x_0 - hg_0 \\ & g_\star = 0. \end{aligned}$$

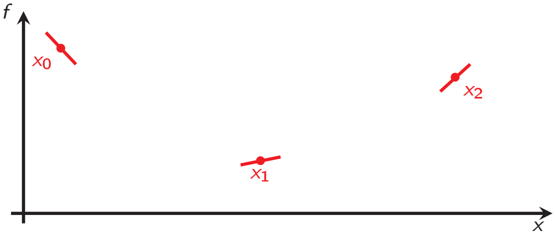
- Variables: x_0, x_1, $x_\star$, g_0, $g_\star$, f_0, $f_\star$.

Smooth strongly convex interpolation (or extension)

Consider an index set S, and its associated values $\{(x_i, g_i, f_i)\}_{i \in S}$ with coordinates x_i, (sub)gradients g_i and function values f_i.

Smooth strongly convex interpolation (or extension)

Consider an index set S, and its associated values $\{(x_i, g_i, f_i)\}_{i \in S}$ with coordinates x_i, (sub)gradients g_i and function values f_i.

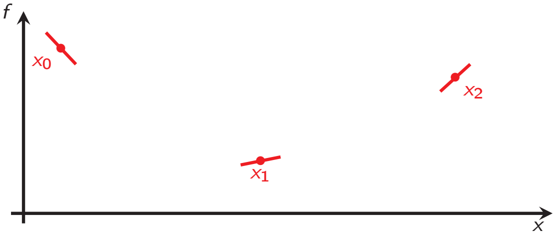


? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that

$$f(x_i) = f_i, \quad \text{and} \quad g_i \in \partial f(x_i), \qquad \forall i \in S.$$

Smooth strongly convex interpolation (or extension)

Consider an index set S, and its associated values $\{(x_i, g_i, f_i)\}_{i\in S}$ with coordinates x_i, (sub)gradients g_i and function values f_i.



? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that

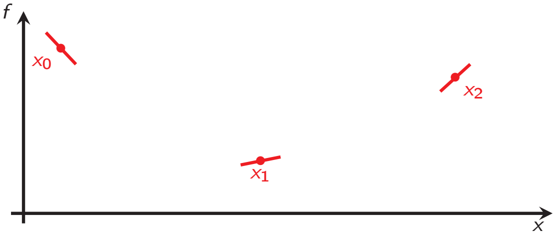
$$f(x_i) = f_i, \quad \text{and} \quad g_i \in \partial f(x_i), \qquad \forall i \in S.$$

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

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

Smooth strongly convex interpolation (or extension)

Consider an index set S, and its associated values $\{(x_i, g_i, f_i)\}_{i\in S}$ with coordinates x_i, (sub)gradients g_i and function values f_i.



? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that

$$f(x_i) = f_i, \quad \text{and} \quad g_i \in \partial f(x_i), \qquad \forall i \in S.$$

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

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

- 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

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

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

$$x_1 = x_0 - h g_0$$

$$g_\star = 0,$$

Replace constraints

- Interpolation conditions allow removing red constraints

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

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

$$x_1 = x_0 - hg_0$$

$$g_\star = 0,$$

- 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

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

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

$$x_1 = x_0 - hg_0$$

$$g_\star = 0,$$

◇ 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); but still non-convex quadratic problem.

Semidefinite lifting

Semidefinite lifting

◇ Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

Semidefinite lifting

- Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star \rangle \\ \langle g_0, x_0 - x_\star \rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

- previous problem can be reformulated as a 2×2 SDP

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

Semidefinite lifting

◇ Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

◇ previous problem can be reformulated as a 2×2 SDP

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

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

Semidefinite lifting

- Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star \rangle \\ \langle g_0, x_0 - x_\star \rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

- previous problem can be reformulated as a 2×2 SDP

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

(using an an 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

- Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star \rangle \\ \langle g_0, x_0 - x_\star \rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

- previous problem can be reformulated as a 2×2 SDP

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

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

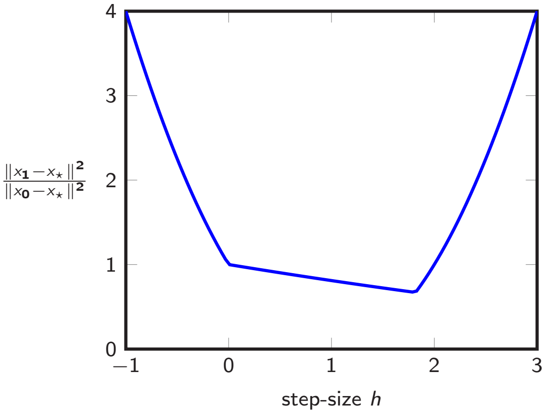
- Assuming $x_0, x_\star, g_0 \in \mathbb{R}^d$ with $d \geqslant 2$, same optimal value as original problem!
- For $d = 1$ same as original problem by adding $\text{rank}(G) \leqslant 1$.

Solving the SDP...

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

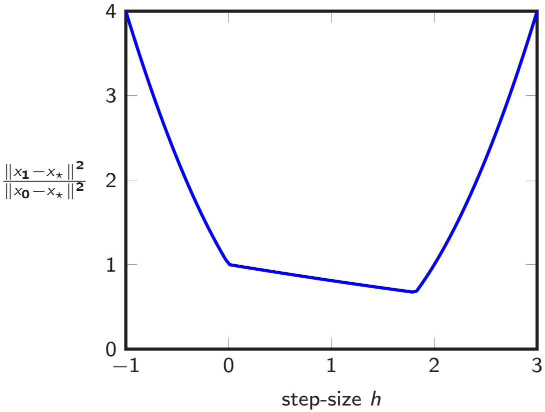
Solving the SDP...

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



Solving the SDP...

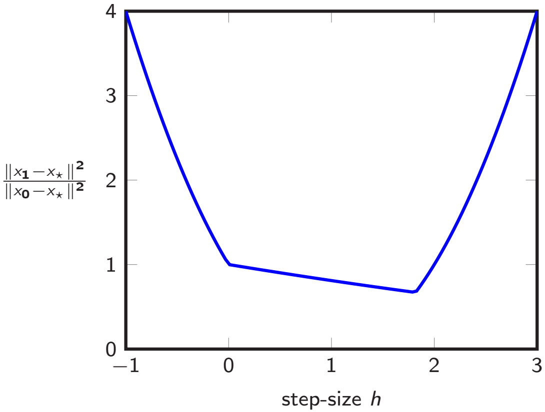
Fix $L = 1$, $\mu = .1$ and solve the SDP for a few values of h.



◇ Observation: numerics match $\max\{(1-hL)^2,(1-h\mu)^2\}$.

Solving the SDP...

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



◇ Observation: numerics match $\max\{(1-hL)^2, (1-h\mu)^2\}$.

◇ We recover the celebrated $\frac{2}{L+\mu}$ as the optimal step-size.

Dual problem

◇ Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \tau$$

$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & h - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ h - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - h^2 \end{bmatrix} \succcurlyeq 0$$

$$0 = \lambda_1 - \lambda_2.$$

Dual problem

- Dual problem is

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

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

Dual problem

◇ Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2 \geqslant 0} \tau$$

$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & h - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ h - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - h^2 \end{bmatrix} \succcurlyeq 0$$

$$0 = \lambda_1 - \lambda_2.$$

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

◇ Direct consequence: for any $\tau \geqslant 0$ we have

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

$$\Uparrow$$

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

Dual problem

- Dual problem is

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

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

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

Dual problem

◇ Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \tau$$

$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & h - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ h - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - h^2 \end{bmatrix} \succcurlyeq 0$$

$$0 = \lambda_1 - \lambda_2.$$

◇ Weak duality: any dual feasible point ≡ valid worst-case convergence rate ($\Uparrow$).

◇ Direct consequence: for any $\tau \geqslant 0$ we have

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

$$\Uparrow$$

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

◇ Strong duality holds (existence of a Slater point): any valid worst-case convergence rate ≡ valid dual feasible point ($\Downarrow$) .

Dual problem

- Dual problem is

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

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

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

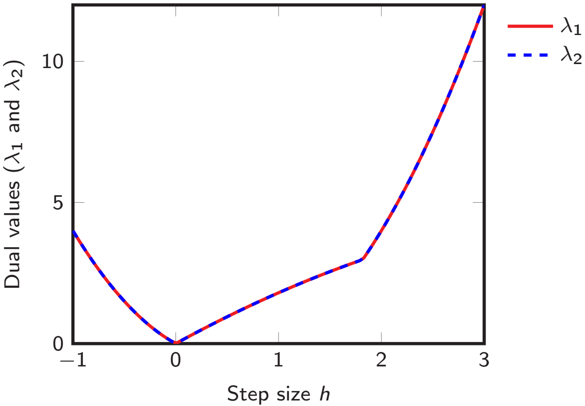
- Strong duality holds (existence of a Slater point): any valid worst-case convergence rate $\equiv$ valid dual feasible point ($\Downarrow$) : hence "$\Updownarrow$".

Dual solutions

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

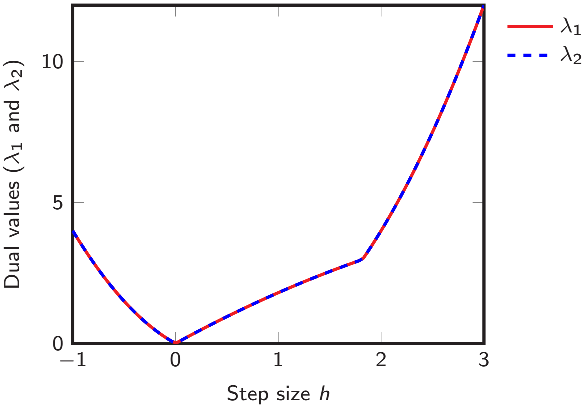
Dual solutions

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



Dual solutions

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



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

Recovering a “standard” proof

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

Recovering a "standard" proof

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

$$\begin{aligned} f_0 \geqslant f_\star \quad & + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\ & + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_1 \\ f_\star \geqslant f_0 \quad & + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\ & + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_2 \end{aligned}$$

Recovering a "standard" proof

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

$$\begin{aligned}
f_0 \geqslant f_\star \quad & +\tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_1 \\
f_\star \geqslant f_0 \quad & +\langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_2
\end{aligned}$$

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

Recovering a "standard" proof

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

$$\begin{aligned} f_0 \geqslant f_\star \quad & + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\ & + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_1 = 2h(1-\mu h) \\ f_\star \geqslant f_0 \quad & + \langle \nabla f(x_0), x_\star - x_0 \rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\ & + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_2 = 2h(1-\mu h) \end{aligned}$$

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

Recovering a "standard" proof

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

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

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

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

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

Recovering a "standard" proof

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

$$\begin{aligned} f_0 \geqslant f_\star \quad & + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\ & + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_1 = 2h(1-\mu h) \\ f_\star \geqslant f_0 \quad & + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\ & + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 \qquad : \lambda_2 = 2h(1-\mu h) \end{aligned}$$

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

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

Recovering a "standard" proof

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

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

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

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

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

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

Recovering a "standard" proof

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

$$\begin{aligned}
f_0 \geqslant f_\star \quad & +\tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_1 = 2h(1-\mu h) \\
f_\star \geqslant f_0 \quad & +\langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_2 = 2h(1-\mu h)
\end{aligned}$$

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

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

leading to $\|x_1 - x_\star\|^2 \leqslant (1 - \frac{\mu}{L})^2 \|x_0 - x_\star\|^2$

Recovering a "standard" proof

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

$$
\begin{aligned}
f_0 \geqslant f_\star \quad & +\tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_1 = 2h(1-\mu h) \\
f_\star \geqslant f_0 \quad & +\langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 && : \lambda_2 = 2h(1-\mu h)
\end{aligned}
$$

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

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

leading to $\|x_1 - x_\star\|^2 \leqslant (1 - \frac{\mu}{L})^2 \|x_0 - x_\star\|^2$

Recovering a "standard" proof

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

$$
\begin{aligned}
f_0 \geqslant f_\star \quad & + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 & : \lambda_1 = 2h(1-\mu h) \\
f_\star \geqslant f_0 \quad & + \langle \nabla f(x_0), x_\star - x_0\rangle + \tfrac{1}{2L}\|\nabla f(x_0)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}\nabla f(x_0)\right\|^2 & : \lambda_2 = 2h(1-\mu h)
\end{aligned}
$$

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

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

leading to $\|x_1 - x_\star\|^2 \leqslant (1 - \frac{\mu}{L})^2\|x_0 - x_\star\|^2$ (tight).

What did we do, so far?

Summary:

What did we do, so far?

Summary:

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

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

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

What did we do, so far?

Summary:

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

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

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

What did we do, so far?

Summary:

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

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

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

What did we do, so far?

Summary:

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

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

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

What did we do, so far?

Summary:

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

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

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

... what happens beyond gradient descent for smooth strongly convex minimization?

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.

What's next?

What's next?

- ◇ More iterations?

What's next?

- More iterations?
- Other types of problems?

  Non-smooth convex functions, non-convex smooth functions, indicator functions, monotone operators, etc.

What's next?

- More iterations?
- Other types of problems?

  Non-smooth convex functions, non-convex smooth functions, indicator functions, monotone operators, etc.
- Other types of methods?

  Projections, proximal operators, linear optimization oracles (Frank-Wolfe), mirror descent, approximate versions, momentum, etc.

What's next?

- More iterations?
- Other types of problems?

  Non-smooth convex functions, non-convex smooth functions, indicator functions, monotone operators, etc.
- Other types of methods?

  Projections, proximal operators, linear optimization oracles (Frank-Wolfe), mirror descent, approximate versions, momentum, etc.
- Human-readable/simpler proofs?

  Specialized PEPs looking for Lyapunov functions.

What's next?

- More iterations?
- Other types of problems?

  Non-smooth convex functions, non-convex smooth functions, indicator functions, monotone operators, etc.
- Other types of methods?

  Projections, proximal operators, linear optimization oracles (Frank-Wolfe), mirror descent, approximate versions, momentum, etc.
- Human-readable/simpler proofs?

  Specialized PEPs looking for Lyapunov functions.
- Step-size optimization?

  Optimize worst-case performance.

Example

Software

Step-size optimization

Concluding remarks

Avoiding semidefinite programming modeling steps?

Avoiding semidefinite programming modeling steps?

Baptiste Goujaud

Céline Moucer

Aymeric Dieuleveut

Julien Hendrickx

François Glineur

- ◇ Matlab version: Performance Estimation Toolbox (PESTO) available at

  github.com/PerformanceEstimation/Performance-Estimation-Toolbox

- ◇ Python version: PEPit available at

  github.com/PerformanceEstimation/PEPit/

Packages contain more than 75 examples!

A few examples

Algorithms for solving:

$$\min_x f(x)$$

with f convex and L-smooth.

A few examples

Algorithms for solving:

$$\min_x f(x)$$

with f convex and L-smooth.

We compare: gradient descent vs. heavy-ball vs. Nesterov's acceleration

A few examples

Algorithms for solving:

$$\min_x f(x)$$

with f convex and L-smooth.

We compare: gradient descent vs. heavy-ball vs. Nesterov's acceleration

- ◇ in terms of worst-cases $\frac{f(x_k)-f(x_\star)}{\|x_0-x_\star\|^2}$,

A few examples

Algorithms for solving:

$$\min_x f(x)$$

with f convex and L-smooth.

We compare: gradient descent vs. heavy-ball vs. Nesterov's acceleration

- in terms of worst-cases $\frac{f(x_k)-f(x_\star)}{\|x_0-x_\star\|^2}$,
- in terms of worst-cases $\frac{\|\nabla f(x_k)\|^2}{\|x_0-x_\star\|^2}$,

A few examples

Algorithms for solving:

$$\min_x f(x)$$

with f convex and L-smooth.

We compare: gradient descent vs. heavy-ball vs. Nesterov's acceleration

- ◇ in terms of worst-cases $\frac{f(x_k)-f(x_\star)}{\|x_0-x_\star\|^2}$,
- ◇ in terms of worst-cases $\frac{\|\nabla f(x_k)\|^2}{\|x_0-x_\star\|^2}$,
- ◇ in terms of worst-cases $\min_{0\leqslant i\leqslant k} \frac{\|\nabla f(x_i)\|^2}{\|x_0-x_\star\|^2}$.

A few examples

Proximal point algorithm for (maximal) monotone inclusion:

$$\text{find } x : \; 0 \in A(x)$$

with $A : \mathbb{R}^d \to 2^{\mathbb{R}^d}$ maximal monotone.

A few examples

Proximal point algorithm for (maximal) monotone inclusion:

$$\text{find } x: \; 0 \in A(x)$$

with $A: \mathbb{R}^d \to 2^{\mathbb{R}^d}$ maximal monotone.

What is the worst-case $\frac{\|x_{k+1}-x_k\|^2}{\|x_0-x_\star\|^2}$ when $x_{i+1} = J_A(x_i)$?

Current library of examples within PESTO/PEPit

Includes... but not limited to

- subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,

Current library of examples within PESTO/PEPit

Includes... but not limited to

- ◇ subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- ◇ proximal point algorithm,
- ◇ projected and proximal gradient, accelerated/momentum versions,

Current library of examples within PESTO/PEPit

Includes... but not limited to

- ◇ subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- ◇ proximal point algorithm,
- ◇ projected and proximal gradient, accelerated/momentum versions,
- ◇ steepest descent, greedy/conjugate gradient methods,
- ◇ Douglas-Rachford/three operator splitting,
- ◇ Frank-Wolfe/conditional gradient,

Current library of examples within PESTO/PEPit

Includes... but not limited to

- ◇ subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- ◇ proximal point algorithm,
- ◇ projected and proximal gradient, accelerated/momentum versions,
- ◇ steepest descent, greedy/conjugate gradient methods,
- ◇ Douglas-Rachford/three operator splitting,
- ◇ Frank-Wolfe/conditional gradient,
- ◇ inexact gradient/fast gradient,
- ◇ Krasnoselskii-Mann and Halpern fixed-point iterations,
- ◇ mirror descent/Bregman gradient/"NoLips",

Current library of examples within PESTO/PEPit

Includes... but not limited to

- ◇ subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- ◇ proximal point algorithm,
- ◇ projected and proximal gradient, accelerated/momentum versions,
- ◇ steepest descent, greedy/conjugate gradient methods,
- ◇ Douglas-Rachford/three operator splitting,
- ◇ Frank-Wolfe/conditional gradient,
- ◇ inexact gradient/fast gradient,
- ◇ Krasnoselskii-Mann and Halpern fixed-point iterations,
- ◇ mirror descent/Bregman gradient/"NoLips",
- ◇ stochastic methods: Point-SAGA, SAGA, SGD and variants.

Current library of examples within PESTO/PEPit

Includes... but not limited to

- subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- proximal point algorithm,
- projected and proximal gradient, accelerated/momentum versions,
- steepest descent, greedy/conjugate gradient methods,
- Douglas-Rachford/three operator splitting,
- Frank-Wolfe/conditional gradient,
- inexact gradient/fast gradient,
- Krasnoselskii-Mann and Halpern fixed-point iterations,
- mirror descent/Bregman gradient/"NoLips",
- stochastic methods: Point-SAGA, SAGA, SGD and variants.

... contain most of the recent PEP-related advances (including by other groups).

Current library of examples within PESTO/PEPit

Includes... but not limited to

- ◇ subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- ◇ proximal point algorithm,
- ◇ projected and proximal gradient, accelerated/momentum versions,
- ◇ steepest descent, greedy/conjugate gradient methods,
- ◇ Douglas-Rachford/three operator splitting,
- ◇ Frank-Wolfe/conditional gradient,
- ◇ inexact gradient/fast gradient,
- ◇ Krasnoselskii-Mann and Halpern fixed-point iterations,
- ◇ mirror descent/Bregman gradient/"NoLips",
- ◇ stochastic methods: Point-SAGA, SAGA, SGD and variants.

... contain most of the recent PEP-related advances (including by other groups).

Among others, see works by Drori, Teboulle, Kim, Fessler, Ryu, Lieder, Lessard, Recht, Packard, Van Scoy, Cyrus, Gu, Yang, etc.

Back to legitimate questions:

- ⋄ anything improvable? Realistic analyses?

Back to legitimate questions:

- ◇ anything improvable? Realistic analyses?
- ◇ How to choose the right inequalities to combine?

Back to legitimate questions:

- ◇ anything improvable? Realistic analyses?
- ◇ How to choose the right inequalities to combine?
- ◇ Why studying this specific quantity ($\|x_k - x_\star\|$)?

Back to legitimate questions:

- ◇ anything improvable? Realistic analyses?
- ◇ How to choose the right inequalities to combine?
- ◇ Why studying this specific quantity ($\|x_k - x_\star\|$)?
- ◇ How to study other quantities, e.g., $f(x_k) - f(x_\star)$?

Back to legitimate questions:

- ◇ anything improvable? Realistic analyses?
- ◇ How to choose the right inequalities to combine?
- ◇ Why studying this specific quantity ($\|x_k - x_\star\|$)?
- ◇ How to study other quantities, e.g., $f(x_k) - f(x_\star)$?
- ◇ Unique way to arrive to the desired result?

Back to legitimate questions:

- ◇ anything improvable? Realistic analyses?
- ◇ How to choose the right inequalities to combine?
- ◇ Why studying this specific quantity ($\|x_k - x_\star\|$)?
- ◇ How to study other quantities, e.g., $f(x_k) - f(x_\star)$?
- ◇ Unique way to arrive to the desired result?
- ◇ How likely are we to find such proofs in more complicated cases?

Recap'

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, etc.),

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, etc.),
- ☹ SDPs typically become prohibitively large in a variety of scenarios,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, etc.),
- ☹ SDPs typically become prohibitively large in a variety of scenarios,
- ☹ transient behavior VS. asymptotic behavior: might be hard to distinguish with small N,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, 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,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, 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 ('20). "On the convergence of the Halpern-iteration".

A few instructive examples

Worst-case analysis for fixed-point iterations:

- Lieder ('20). "On the convergence of the Halpern-iteration".

Analysis of the proximal-point algorithm for monotone inclusions:

- Gu, Yang ('19). "Optimal nonergodic sublinear convergence rate of the proximal point algorithm for maximal monotone inclusion problems".

A few instructive examples

Worst-case analysis for fixed-point iterations:

- ◇ Lieder ('20). "On the convergence of the Halpern-iteration".

Analysis of the proximal-point algorithm for monotone inclusions:

- ◇ Gu, Yang ('19). "Optimal nonergodic sublinear convergence rate of the proximal point algorithm for maximal monotone inclusion problems".

Application to nonconvex optimization:

- ◇ Abbaszadehpeivasti, de Klerk, Zamani ('21). "The exact worst-case convergence rate of the gradient method with fixed step lengths for L-smooth functions".
- ◇ Rotaru, Glineur, Patrinos ('22). "Tight convergence rates of the gradient method on hypoconvex functions".

A few instructive examples

Worst-case analysis for fixed-point iterations:

- ⋄ Lieder ('20). "On the convergence of the Halpern-iteration".

Analysis of the proximal-point algorithm for monotone inclusions:

- ⋄ Gu, Yang ('19). "Optimal nonergodic sublinear convergence rate of the proximal point algorithm for maximal monotone inclusion problems".

Application to nonconvex optimization:

- ⋄ Abbaszadehpeivasti, de Klerk, Zamani ('21). "The exact worst-case convergence rate of the gradient method with fixed step lengths for L-smooth functions".
- ⋄ Rotaru, Glineur, Patrinos ('22). "Tight convergence rates of the gradient method on hypoconvex functions".

Applications to distributed optimization:

- ⋄ Sundararajan, Van Scoy, Lessard ('19). "Analysis and design of first-order distributed optimization algorithms over time-varying graphs."
- ⋄ Colla, Hendrickx ('23). "Automatic performance estimation for decentralized optimization."

A few instructive examples—shameless advertisement

Applications to mirror descent + lower complexity bound

- ⋄ Dragomir, T., d'Aspremont, Bolte ('21). "Optimal complexity and certification of Bregman first-order methods."

A few instructive examples—shameless advertisement

Applications to mirror descent + lower complexity bound

- Dragomir, T., d'Aspremont, Bolte ('21). "Optimal complexity and certification of Bregman first-order methods."

Applications to adaptive methods

- Barré, T., d'Aspremont ('20). "Complexity Guarantees for Polyak Steps with Momentum."
- Das Gupta, Freund, Sun, T ('23). "Nonlinear conjugate gradient methods: worst-case convergence rates via computer-assisted analyses."

A few instructive examples—shameless advertisement

Applications to mirror descent + lower complexity bound

- ◇ Dragomir, T., d'Aspremont, Bolte ('21). "Optimal complexity and certification of Bregman first-order methods."

Applications to adaptive methods

- ◇ Barré, T., d'Aspremont ('20). "Complexity Guarantees for Polyak Steps with Momentum."
- ◇ Das Gupta, Freund, Sun, T ('23). "Nonlinear conjugate gradient methods: worst-case convergence rates via computer-assisted analyses."

Lyapunov functions (compact proofs) & counter-examples

- ◇ Lessard, Recht, Packard ('16). "Analysis and design of optimization algorithms via integral quadratic constraints."
- ◇ T, Bach ('19). "Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions."
- ◇ Upadhyaya, Banert, T, Giselsson ('23). "Automated tight Lyapunov analysis for first-order methods."
- ◇ Goujaud, Dieuleveut, T ('23). "Counter-examples in first-order optimization: a constructive approach."

Poster

Nizar
Bousselmi

Julien
Hendrickx

François
Glineur

→ Bousselmi, Hendrickx, Glineur ('23). "Interpolation Conditions for Linear Operators and applications to Performance Estimation Problems."

Example

Software

Step-size optimization

Concluding remarks

Creating new algorithms

Smooth (strongly) convex minimization with more than gradient descent?

Creating new algorithms

Smooth (strongly) convex minimization with more than gradient descent?

$$x_1 = x_0 - h_{1,0} \nabla f(x_0)$$

Creating new algorithms

Smooth (strongly) convex minimization with more than gradient descent?

$$x_1 = x_0 - h_{1,0}\nabla f(x_0)$$
$$x_2 = x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1)$$

Creating new algorithms

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0) \\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1) \\
&\vdots
\end{aligned}
$$

Creating new algorithms

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0)\\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1)\\
&\vdots\\
x_N &= x_{N-1} - h_{N,0}\nabla f(x_0) - \ldots - h_{N,N-1}\nabla f(x_{N-1})
\end{aligned}
$$

Creating new algorithms

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0) \\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1) \\
&\vdots \\
x_N &= x_{N-1} - h_{N,0}\nabla f(x_0) - \ldots - h_{N,N-1}\nabla f(x_{N-1})
\end{aligned}
$$

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

Creating new algorithms

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0) \\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1) \\
&\vdots \\
x_N &= x_{N-1} - h_{N,0}\nabla f(x_0) - \ldots - h_{N,N-1}\nabla f(x_{N-1})
\end{aligned}
$$

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

- pick a performance criterion, for instance

$$
\frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2},
$$

Creating new algorithms

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0)\\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1)\\
&\vdots\\
x_N &= x_{N-1} - h_{N,0}\nabla f(x_0) - \ldots - h_{N,N-1}\nabla f(x_{N-1})
\end{aligned}
$$

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

- ⋄ pick a performance criterion, for instance

$$
\frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2},
$$

- ⋄ solve the minimax:

$$
\min_{\{h_{i,j}\}_{i,j}} \max_{f\in\mathcal{F},\{x_i\}} \frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}.
$$

Solution to inner maximization via $N \times N$ SDP.

Design problem

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

$$\min_{\{h_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

Design problem

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

$$\min_{\{h_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

- brutal approaches
  - Das Gupta, Van Parys, Ryu ('23) "Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods."

Design problem

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

$$\min_{\{h_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

- ⋄ brutal approaches
  - – Das Gupta, Van Parys, Ryu ('23) "Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods."
- ⋄ convex relaxations,

Design problem

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

$$\min_{\{h_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

- ⋄ brutal approaches
  - – Das Gupta, Van Parys, Ryu ('23) "Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods."
- ⋄ convex relaxations,
- ⋄ analogies (e.g., with conjugate gradient methods).

Primal problem ($N = 1$)

Primal problem ($N = 1$)

Recall primal problem, with step-size optimization

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

Primal problem ($N = 1$)

Recall primal problem, with step-size optimization

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

"Simple" minimization problem by dualizing inner maximization.

Primal problem ($N = 1$)

Recall primal problem, with step-size optimization

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

"Simple" minimization problem by dualizing inner maximization.

Dualize inner maximization $\rightarrow \min\min$.

Optimizing the step-sizes ($N = 1$)

Optimizing the step-sizes ($N = 1$)

For $N = 1$, optimizing over step-size $h_{1,0}$ remains convex!

Optimizing the step-sizes ($N = 1$)

For $N = 1$, optimizing over step-size $h_{1,0}$ remains convex!

Indeed:

$$\min_{\tau,\lambda\geqslant 0} \quad \tau$$
$$\text{subject to} \quad \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & h_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ h_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - h_{1,0}^2 \end{bmatrix} \succcurlyeq 0.$$

Optimizing the step-sizes ($N = 1$)

For $N = 1$, optimizing over step-size $h_{1,0}$ remains convex!

Indeed:

$$\min_{\tau,\lambda\geqslant 0,h_{1,0}} \tau$$

$$\text{subject to } \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & h_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ h_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - h_{1,0}^2 \end{bmatrix} \succcurlyeq 0.$$

Optimizing the step-sizes ($N = 1$)

For $N = 1$, optimizing over step-size $h_{1,0}$ remains convex!

Indeed:

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

Optimize $h_{1,0}$ "for free" (linear SDP via Schur complement):

$$\begin{aligned} \min_{\tau,\lambda\geqslant 0,h_{1,0}} \quad & \tau \\ \text{subject to} \quad & \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & -\frac{\lambda(\mu+L)}{2(L-\mu)} & 1 \\ -\frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} & -h_{1,0} \\ 1 & -h_{1,0} & 1 \end{bmatrix} \succcurlyeq 0. \end{aligned}$$

Optimizing the step-sizes ($N = 2$)

When $N = 2$, the problem becomes

$$\min_{\substack{\tau, \lambda_1, \ldots, \lambda_6 \geqslant 0 \\ \{h_{i,j}\}}} \tau$$

subject to

Optimizing the step-sizes ($N = 2$)

When $N = 2$, the problem becomes

$$\min_{\substack{\tau, \lambda_1, \ldots, \lambda_6 \geqslant 0 \\ \{h_{i,j}\}}} \tau$$

subject to

$$\begin{bmatrix} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_5 \\ -\lambda_1 + \lambda_3 + \lambda_4 - \lambda_6 \end{bmatrix} = 0,$$

Optimizing the step-sizes ($N = 2$)

When $N = 2$, the problem becomes

$$\begin{aligned} \min_{\substack{\tau, \lambda_1, \ldots, \lambda_6 \geqslant 0 \\ \{h_{i,j}\}}} \; & \tau \\ \text{subject to } & \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{1,2} & S_{2,2} & S_{2,3} \\ S_{1,3} & S_{2,3} & S_{3,3} \end{bmatrix} \succcurlyeq 0 \\ & \begin{bmatrix} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_5 \\ -\lambda_1 + \lambda_3 + \lambda_4 - \lambda_6 \end{bmatrix} = 0, \end{aligned}$$

Optimizing the step-sizes ($N = 2$)

When $N = 2$, the problem becomes

$$
\begin{aligned}
\min_{\substack{\tau,\lambda_1,\ldots,\lambda_6 \geqslant 0 \\ \{h_{i,j}\}}} \; & \tau \\
\text{subject to } & \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{1,2} & S_{2,2} & S_{2,3} \\ S_{1,3} & S_{2,3} & S_{3,3} \end{bmatrix} \succcurlyeq 0 \\
& \begin{bmatrix} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_5 \\ -\lambda_1 + \lambda_3 + \lambda_4 - \lambda_6 \end{bmatrix} = 0,
\end{aligned}
$$

for some $S_{1,1}, S_{1,2}, \ldots, S_{3,3}$ (functions of $\tau, \lambda_1, \ldots, \lambda_6$ and $\{h_{i,j}\}$).

Optimizing the step-sizes ($N = 2$)

When $N = 2$, the problem becomes

$$\min_{\substack{\tau,\lambda_1,\ldots,\lambda_6\geqslant 0\\ \{h_{i,j}\}}} \tau$$

$$\text{subject to } \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{1,2} & S_{2,2} & S_{2,3} \\ S_{1,3} & S_{2,3} & S_{3,3} \end{bmatrix} \succcurlyeq 0$$

$$\begin{bmatrix} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_5 \\ -\lambda_1 + \lambda_3 + \lambda_4 - \lambda_6 \end{bmatrix} = 0,$$

for some $S_{1,1}, S_{1,2}, \ldots, S_{3,3}$ (functions of $\tau, \lambda_1, \ldots, \lambda_6$ and $\{h_{i,j}\}$).

In particular

$$S_{1,2} = -\frac{L\lambda_3 - 2(L-\mu)h_{2,0} + \mu\lambda_1 + L\mu(\lambda_2+\lambda_5)h_{1,0}}{L-\mu}$$

$$S_{2,2} = \frac{-2(\mu\lambda_6 + L\lambda_4)h_{1,0} - 2(L-\mu)h_{2,0}^2 + L\mu(\lambda_2+\lambda_4+\lambda_5+\lambda_6)h_{1,0}^2 + \lambda_1+\lambda_3+\lambda_4+\lambda_6}{L-\mu}$$

Optimizing the step-sizes ($N = 2$)

When $N = 2$, the problem becomes

$$\begin{aligned} \min_{\substack{\tau,\lambda_1,\ldots,\lambda_6 \geqslant 0 \\ \{h_{i,j}\}}} \; & \tau \\ \text{subject to } & \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{1,2} & S_{2,2} & S_{2,3} \\ S_{1,3} & S_{2,3} & S_{3,3} \end{bmatrix} \succcurlyeq 0 \\ & \begin{bmatrix} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_5 \\ -\lambda_1 + \lambda_3 + \lambda_4 - \lambda_6 \end{bmatrix} = 0, \end{aligned}$$

for some $S_{1,1}, S_{1,2}, \ldots, S_{3,3}$ (functions of $\tau, \lambda_1, \ldots, \lambda_6$ and $\{h_{i,j}\}$).

In particular

$$\begin{aligned} S_{1,2} &= -\frac{L\lambda_3 - 2(L-\mu)h_{2,0} + \mu\lambda_1 + L\mu(\lambda_2+\lambda_5)h_{1,0}}{L-\mu} \\ S_{2,2} &= \frac{-2(\mu\lambda_6 + L\lambda_4)h_{1,0} - 2(L-\mu)h_{2,0}^2 + L\mu(\lambda_2+\lambda_4+\lambda_5+\lambda_6)h_{1,0}^2 + \lambda_1+\lambda_3+\lambda_4+\lambda_6}{L-\mu} \end{aligned}$$

LMI convex in some step-sizes ($h_{2,0}$ and $h_{2,1}$) but not in the others.

Numerical examples I

Example for $L = 1$ and $\mu = .1$

Numerical examples I

Example for $L = 1$ and $\mu = .1$

- ◇ For $N = 1$, we reach $\frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.6694$ with step-sizes

$$[h^\star_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

Numerical examples I

Example for $L = 1$ and $\mu = .1$

- For $N = 1$, we reach $\frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.6694$ with step-sizes

$$[h^\star_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

- For $N = 2$, we reach $\frac{\|x_2 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.3769$ with

$$[h^\star_{i,j}] = \begin{bmatrix} 1.5466 & \\ 0.2038 & 2.4961 \end{bmatrix}.$$

Numerical examples I

Example for $L = 1$ and $\mu = .1$

- For $N = 1$, we reach $\frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.6694$ with step-sizes

$$[h^\star_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

- For $N = 2$, we reach $\frac{\|x_2 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.3769$ with

$$[h^\star_{i,j}] = \begin{bmatrix} 1.5466 & \\ 0.2038 & 2.4961 \end{bmatrix}.$$

- For $N = 3$, we reach $\frac{\|x_3 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.1932$ with

$$[h^\star_{i,j}] = \begin{bmatrix} 1.5466 & & \\ 0.1142 & 1.8380 & \\ 0.0642 & 0.4712 & 2.8404 \end{bmatrix}.$$

Numerical examples I

Example for $L = 1$ and $\mu = .1$

- For $N = 1$, we reach $\frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.6694$ with step-sizes

$$[h^\star_{i,j}] = \begin{bmatrix}1.8182\end{bmatrix}.$$

- For $N = 2$, we reach $\frac{\|x_2 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.3769$ with

$$[h^\star_{i,j}] = \begin{bmatrix}1.5466 & \\ 0.2038 & 2.4961\end{bmatrix}.$$

- For $N = 3$, we reach $\frac{\|x_3 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.1932$ with

$$[h^\star_{i,j}] = \begin{bmatrix}1.5466 & & \\ 0.1142 & 1.8380 & \\ 0.0642 & 0.4712 & 2.8404\end{bmatrix}.$$

- For $N = 4$, we reach $\frac{\|x_4 - x_\star\|^2}{\|x_0 - x_\star\|^2} \leqslant 0.0944$ with

$$[h^\star_{i,j}] = \begin{bmatrix}1.5466 & & & \\ 0.1142 & 1.8380 & & \\ 0.0331 & 0.2432 & 1.9501 & \\ 0.0217 & 0.1593 & 0.6224 & 3.0093\end{bmatrix}.$$

Numerical examples II

What about different performance measure? Example $\frac{f(x_N)-f_\star}{f(x_0)-f_\star}$ and $L=1$, $\mu=.1$.

Numerical examples II

What about different performance measure? Example $\frac{f(x_N)-f_\star}{f(x_0)-f_\star}$ and $L=1$, $\mu=.1$.

- For $N=1$, we obtain $\frac{f(x_1)-f_\star}{f(x_0)-f_\star} \leqslant 0.6694$ with step-size

$$[h_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

Numerical examples II

What about different performance measure? Example $\frac{f(x_N)-f_\star}{f(x_0)-f_\star}$ and $L=1$, $\mu=.1$.

◇ For $N=1$, we obtain $\frac{f(x_1)-f_\star}{f(x_0)-f_\star} \leqslant 0.6694$ with step-size

$$[h_{i,j}] = \begin{bmatrix}1.8182\end{bmatrix}.$$

◇ For $N=2$, we obtain $\frac{f(x_2)-f_\star}{f(x_0)-f_\star} \leqslant 0.3554$ with

$$[h_{i,j}] = \begin{bmatrix}2.0095 & \\ 0.4229 & 2.0095\end{bmatrix}.$$

Numerical examples II

What about different performance measure? Example $\frac{f(x_N)-f_\star}{f(x_\mathbf{0})-f_\star}$ and $L = 1$, $\mu = .1$.

- For $N = 1$, we obtain $\frac{f(x_\mathbf{1})-f_\star}{f(x_\mathbf{0})-f_\star} \leqslant 0.6694$ with step-size

$$[h_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

- For $N = 2$, we obtain $\frac{f(x_\mathbf{2})-f_\star}{f(x_\mathbf{0})-f_\star} \leqslant 0.3554$ with

$$[h_{i,j}] = \begin{bmatrix} 2.0095 & \\ 0.4229 & 2.0095 \end{bmatrix}.$$

- For $N = 3$, we obtain $\frac{f(x_\mathbf{3})-f_\star}{f(x_\mathbf{0})-f_\star} \leqslant 0.1698$ with

$$[h_{i,j}] = \begin{bmatrix} 1.9470 & & \\ 0.4599 & 2.2406 & \\ 0.1705 & 0.4599 & 1.9470 \end{bmatrix}.$$

Numerical examples II

What about different performance measure? Example $\frac{f(x_N)-f_\star}{f(x_0)-f_\star}$ and $L=1$, $\mu=.1$.

- For $N=1$, we obtain $\frac{f(x_1)-f_\star}{f(x_0)-f_\star} \leqslant 0.6694$ with step-size
$$[h_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$
- For $N=2$, we obtain $\frac{f(x_2)-f_\star}{f(x_0)-f_\star} \leqslant 0.3554$ with
$$[h_{i,j}] = \begin{bmatrix} 2.0095 & \\ 0.4229 & 2.0095 \end{bmatrix}.$$
- For $N=3$, we obtain $\frac{f(x_3)-f_\star}{f(x_0)-f_\star} \leqslant 0.1698$ with
$$[h_{i,j}] = \begin{bmatrix} 1.9470 & & \\ 0.4599 & 2.2406 & \\ 0.1705 & 0.4599 & 1.9470 \end{bmatrix}.$$
- For $N=4$, we obtain $\frac{f(x_4)-f_\star}{f(x_0)-f_\star} \leqslant 0.0789$ with
$$[h_{i,j}] = \begin{bmatrix} 1.9187 & & & \\ 0.4098 & 2.1746 & & \\ 0.1796 & 0.5147 & 2.1746 & \\ 0.0627 & 0.1796 & 0.4098 & 1.9187 \end{bmatrix}.$$

Numerical examples III

Worst-case performance $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

Numerical examples III

Worst-case performance $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ 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 examples III

Worst-case performance $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ 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 (numerically generated),

Numerical examples III

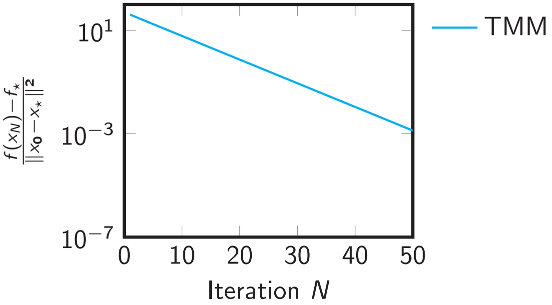
Worst-case performance $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

- 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 (numerically generated),
- Lower complexity bound (numerically generated).

Numerical examples III

Worst-case performance $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

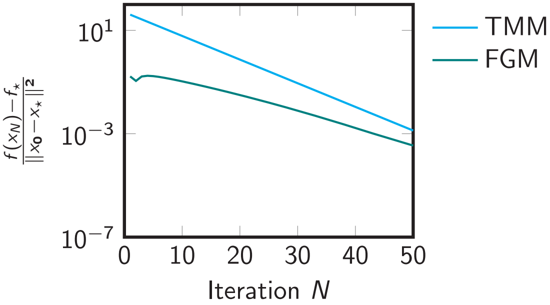
- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method (numerically generated),
- Lower complexity bound (numerically generated).



Numerical examples III

Worst-case performance $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

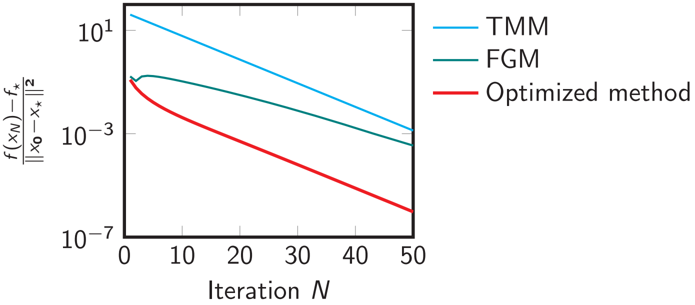
- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method (numerically generated),
- Lower complexity bound (numerically generated).



Numerical examples III

Worst-case performance $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

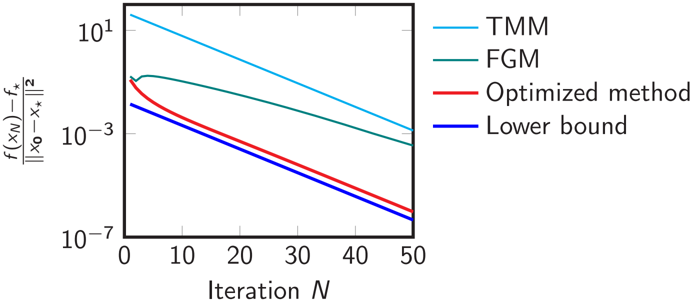
- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method (numerically generated),
- Lower complexity bound (numerically generated).



Numerical examples III

Worst-case performance $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ 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 (numerically generated),
- Lower complexity bound (numerically generated).



Analytical solutions

- ◇ It turns out that for $\frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}$, we can also solve the minimax in closed-form.

Analytical solutions

- ◇ It turns out that for $\frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}$, we can also solve the minimax in closed-form.
- ◇ The method referred to as "Information-Theoretic Exact Method" (ITEM)

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

for some sequences $\{\beta_k\}$, $\{\delta_k\}$ (depending on μ, L, and k).

Analytical solutions

- It turns out that for $\frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}$, we can also solve the minimax in closed-form.
- The method referred to as "Information-Theoretic Exact Method" (ITEM)

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

  for some sequences $\{\beta_k\}$, $\{\delta_k\}$ (depending on μ, L, and k).
- The worst-case guarantee matches exactly a lower complexity bound.

Analytical solutions

- ◇ It turns out that for $\frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}$, we can also solve the minimax in closed-form.
- ◇ The method referred to as "Information-Theoretic Exact Method" (ITEM)

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

for some sequences $\{\beta_k\}$, $\{\delta_k\}$ (depending on μ, L, and k).

- ◇ The worst-case guarantee matches exactly a lower complexity bound.
- ◇ Worst-case guarantee of order

$$\frac{\|z_N - z_\star\|^2}{\|z_0 - z_\star\|^2} = O\left(\left(1 - \sqrt{\tfrac{\mu}{L}}\right)^{2N} \right).$$

Analytical solutions

- It turns out that for $\frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}$, we can also solve the minimax in closed-form.
- The method referred to as "Information-Theoretic Exact Method" (ITEM)

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

  for some sequences $\{\beta_k\}$, $\{\delta_k\}$ (depending on μ, L, and k).
- The worst-case guarantee matches exactly a lower complexity bound.
- Worst-case guarantee of order

$$\frac{\|z_N - z_\star\|^2}{\|z_0 - z_\star\|^2} = O\left(\left(1 - \sqrt{\tfrac{\mu}{L}} \right)^{2N} \right).$$

- The proof is "simple"!

A few observations/limitations

Were we lucky? Some pieces are missing!

A few observations/limitations

Were we lucky? Some pieces are missing!

- ⋄ Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?

A few observations/limitations

Were we lucky? Some pieces are missing!

- Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?
- Why/when can the optimal method be expressed efficiently? (eg. using second order recursions)

A few observations/limitations

Were we lucky? Some pieces are missing!

- ◇ Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?
- ◇ Why/when can the optimal method be expressed efficiently? (eg. using second order recursions)

The situation seems quite involved in general, apart from a few cases

A few observations/limitations

Were we lucky? Some pieces are missing!

- ◇ Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?
- ◇ Why/when can the optimal method be expressed efficiently? (eg. using second order recursions)

The situation seems quite involved in general, apart from a few cases

- ◇ $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$: optimized gradient method (OGM, Kim & Fessler '16),

A few observations/limitations

Were we lucky? Some pieces are missing!

- ◇ Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?
- ◇ Why/when can the optimal method be expressed efficiently? (eg. using second order recursions)

The situation seems quite involved in general, apart from a few cases

- ◇ $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu = 0$: optimized gradient method (OGM, Kim & Fessler '16),
- ◇ $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori '21),

A few observations/limitations

Were we lucky? Some pieces are missing!

- Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?
- Why/when can the optimal method be expressed efficiently? (eg. using second order recursions)

The situation seems quite involved in general, apart from a few cases

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$: optimized gradient method (OGM, Kim & Fessler '16),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori '21),
- $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu=0$: OGM for gradient (OGM-G, Kim & Fessler '21).

A few observations/limitations

Were we lucky? Some pieces are missing!

- Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?
- Why/when can the optimal method be expressed efficiently? (eg. using second order recursions)

The situation seems quite involved in general, apart from a few cases

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$: optimized gradient method (OGM, Kim & Fessler '16),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori '21),
- $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu=0$: OGM for gradient (OGM-G, Kim & Fessler '21).

Relation to quadratics? When specifying f to be quadratic, similar known methods

A few observations/limitations

Were we lucky? Some pieces are missing!

- ⋄ Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?
- ⋄ Why/when can the optimal method be expressed efficiently? (eg. using second order recursions)

The situation seems quite involved in general, apart from a few cases

- ⋄ $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$: optimized gradient method (OGM, Kim & Fessler '16),
- ⋄ $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori '21),
- ⋄ $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu=0$: OGM for gradient (OGM-G, Kim & Fessler '21).

Relation to quadratics? When specifying f to be quadratic, similar known methods

- ⋄ $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$ (via Chebyshev polynomials),

A few observations/limitations

Were we lucky? Some pieces are missing!

- Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?
- Why/when can the optimal method be expressed efficiently? (eg. using second order recursions)

The situation seems quite involved in general, apart from a few cases

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$: optimized gradient method (OGM, Kim & Fessler '16),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori '21),
- $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu=0$: OGM for gradient (OGM-G, Kim & Fessler '21).

Relation to quadratics? When specifying f to be quadratic, similar known methods

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$ (via Chebyshev polynomials),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$ (via Chebyshev polynomials), asymptotically Polyak's Heavy-Ball

A few observations/limitations

Were we lucky? Some pieces are missing!

- ◇ Why/when are optimal step-sizes $\{h^\star_{i,j}\}$ independent of horizon N?
- ◇ Why/when can the optimal method be expressed efficiently? (eg. using second order recursions)

The situation seems quite involved in general, apart from a few cases

- ◇ $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu = 0$: optimized gradient method (OGM, Kim & Fessler '16),
- ◇ $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori '21),
- ◇ $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu = 0$: OGM for gradient (OGM-G, Kim & Fessler '21).

Relation to quadratics? When specifying f to be quadratic, similar known methods

- ◇ $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu = 0$ (via Chebyshev polynomials),
- ◇ $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$ (via Chebyshev polynomials), asymptotically Polyak's Heavy-Ball
- ◇ see e.g.: A. Nemirovsky's "Information-based complexity of convex programming." (lecture notes, 1995)

A few instructive examples

Design first-order methods via PEPs:

- Kim, Fessler ('16). "Optimized methods for smooth convex optimization".

A few instructive examples

Design first-order methods via PEPs:

- ⋄ Kim, Fessler ('16). "Optimized methods for smooth convex optimization".
- ⋄ Van Scoy, Freeman, Lynch ('17). "The fastest known globally convergent first-order method for minimizing strongly convex functions".

A few instructive examples

Design first-order methods via PEPs:

- ◇ Kim, Fessler ('16). "Optimized methods for smooth convex optimization".
- ◇ Van Scoy, Freeman, Lynch ('17). "The fastest known globally convergent first-order method for minimizing strongly convex functions".
- ◇ Kim ('21). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions".

A few instructive examples

Design first-order methods via PEPs:

- ◇ Kim, Fessler ('16). "Optimized methods for smooth convex optimization".
- ◇ Van Scoy, Freeman, Lynch ('17). "The fastest known globally convergent first-order method for minimizing strongly convex functions".
- ◇ Kim ('21). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions".

... including "brutal" examples:

- ◇ Gupta, Van Parijs, Ryu ('23). "Branch-and-Bound Performance Estimation Programming: A Unified Methodology for Constructing Optimal Methods".

A few instructive examples

Design first-order methods via PEPs:

- ◇ Kim, Fessler ('16). "Optimized methods for smooth convex optimization".
- ◇ Van Scoy, Freeman, Lynch ('17). "The fastest known globally convergent first-order method for minimizing strongly convex functions".
- ◇ Kim ('21). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions".

... including "brutal" examples:

- ◇ Gupta, Van Parijs, Ryu ('23). "Branch-and-Bound Performance Estimation Programming: A Unified Methodology for Constructing Optimal Methods".
- ◇ Grimmer ('23). "Provably faster gradient descent via long steps."
- ◇ Altschuler, Parrilo ('23). "Acceleration by Stepsize Hedging I: Multi-Step Descent and the Silver Stepsize Schedule."

Example

Software

Step-size optimization

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 proofs: principled approach,
  - proofs are dual feasible points,
  - proofs 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 proofs: principled approach,
  - proofs are dual feasible points,
  - proofs are linear combinations of certain specific inequalities.

Byproducts:

- ◇ computer-assisted design of proofs,
- ◇ computer-assisted design of numerical methods,
- ◇ step towards reproducible theory
  - validation & benchmark tool for proofs (also for reviews ☺).

Concluding remarks

Difficulties:

Concluding remarks

Difficulties:

- ◇ suffers from standard caveats of worst-case analyses,
- ◇ closed-form solutions might be involved.

Concluding remarks

Difficulties:

- ◇ suffers from standard caveats of worst-case analyses,
- ◇ closed-form solutions might be involved.

A few open directions:

Concluding remarks

Difficulties:

- ◇ suffers from standard caveats of worst-case analyses,
- ◇ closed-form solutions might be involved.

A few open directions:

- ◇ non-Euclidean algorithms (mirror descent-type), what

Concluding remarks

Difficulties:

- suffers from standard caveats of worst-case analyses,
- closed-form solutions might be involved.

A few open directions:

- non-Euclidean algorithms (mirror descent-type), what
- adaptative algorithms, high-order, beyond worst-cases,

Concluding remarks

Difficulties:

- ◇ suffers from standard caveats of worst-case analyses,
- ◇ closed-form solutions might be involved.

A few open directions:

- ◇ non-Euclidean algorithms (mirror descent-type), what
- ◇ adaptative algorithms, high-order, beyond worst-cases,
- ◇ many open setups: bi-level optimization, multi-objective optimization, etc.

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