# Performance Estimation of First-Order Methods: Extensions and Recent results

François Glineur
Université catholique de Louvain (UCLouvain)

*Center for Operations Research and Econometrics* and *Information and Communication Technologies, Electronics and Applied Mathematics Institute*

joint work with many collaborators, including
Nizar Bousselmi, Julien M. Hendrickx, Yassine Kamri,
Panos Patrinos (KU Leuven), Teodor Rotaru
and Adrien B. Taylor (INRIA, Paris)
Funding: FNRS/FRIA, KUL GlobalParternship, EU MSCA

LNMB conference on the Mathematics of Operations Research
Jan. 17, 2023

# Performance estimation of an optimization method

Recall goal of a PEP (Performance Estimation Problem):
compute the worst-case behaviour

Example:

# Performance estimation of an optimization method

Recall goal of a PEP (Performance Estimation Problem):
compute the worst-case behaviour

- of a given optimization method
  (considering a fixed number of iterations)

Example: after computing $N$ steps of the (unconstrained) gradient method with fixed step-size $\frac{1}{L}$

# Performance estimation of an optimization method

Recall goal of a PEP (Performance Estimation Problem):
compute the worst-case behaviour

- of a given optimization method
  (considering a fixed number of iterations)
- applied to any function belonging to a given class
  (e.g. convex functions, possibly with some additional regularity property such as smoothness)

Example: after computing $N$ steps of the (unconstrained) gradient method with fixed step-size $\frac{1}{L}$ applied to a convex function $f$ with an $L$-Lipschitz gradient and minimizer $x^*$

# Performance estimation of an optimization method

Recall goal of a PEP (Performance Estimation Problem):
compute the worst-case behaviour

- of a given optimization method
  (considering a fixed number of iterations)
- applied to any function belonging to a given class
  (e.g. convex functions, possibly with some additional regularity property such as smoothness)
- from any starting point
  (possibly satisfying some condition w.r.t. a minimizer)

Example: after computing $N$ steps of the (unconstrained) gradient method with fixed step-size $\frac{1}{L}$ applied to a convex function $f$ with an $L$-Lipschitz gradient and minimizer $x^*$ from an initial iterate satisfying $\|x_0 - x_*\| \leq R$,

# Performance estimation of an optimization method

Recall goal of a PEP (Performance Estimation Problem):
compute the worst-case behaviour

- of a given optimization method
  (considering a fixed number of iterations)
- applied to any function belonging to a given class
  (e.g. convex functions, possibly with some additional regularity property such as smoothness)
- from any starting point
  (possibly satisfying some condition w.r.t. a minimizer)
- for a given performance criteria
  (e.g. objective accuracy, distance to solution, gradient norm)

Example: after computing $N$ steps of the (unconstrained) gradient method with fixed step-size $\frac{1}{L}$ applied to a convex function $f$ with an $L$-Lipschitz gradient and minimizer $x^*$ from an initial iterate satisfying $\|x_0 - x_*\| \leq R$, what is the worst (largest) possible objective function accuracy for the last iterate $f(x_N) - f(x_*)$ ?

# Output of a performance estimation problem

For a given PEP (Performance Estimation Problem) we will

# Output of a performance estimation problem

For a given PEP (Performance Estimation Problem) we will

- compute the exact value of the performance criteria's worst-case = optimal value of PEP problem

# Output of a performance estimation problem

For a given PEP (Performance Estimation Problem) we will

- compute the exact value of the performance criteria's worst-case = optimal value of PEP problem
- obtain an independently-checkable proof that this worst-case value is a valid (upper) bound on the performance criteria = dual multiplier of PEP problem

# Output of a performance estimation problem

For a given PEP (Performance Estimation Problem) we will

- compute the exact value of the performance criteria's worst-case = optimal value of PEP problem
- obtain an independently-checkable proof that this worst-case value is a valid (upper) bound on the performance criteria = dual multiplier of PEP problem
- identify an explicit function (and starting point) achieving this worst-case value = primal solution of PEP problem + interpolation

# Output of a performance estimation problem

For a given PEP (Performance Estimation Problem) we will

- compute the exact value of the performance criteria's worst-case = optimal value of PEP problem
- obtain an independently-checkable proof that this worst-case value is a valid (upper) bound on the performance criteria = dual multiplier of PEP problem
- identify an explicit function (and starting point) achieving this worst-case value = primal solution of PEP problem + interpolation
- all three steps can be done either numerically or analytically

# Output of a performance estimation problem

For a given PEP (Performance Estimation Problem) we will

- compute the exact value of the performance criteria's worst-case = optimal value of PEP problem
- obtain an independently-checkable proof that this worst-case value is a valid (upper) bound on the performance criteria = dual multiplier of PEP problem
- identify an explicit function (and starting point) achieving this worst-case value = primal solution of PEP problem + interpolation
- all three steps can be done either numerically or analytically

For a large class of first-order methods applied to convex compositive optimization problems, these can be computed exactly using a semidefinite programming (SDP) problem.

# Two ingredients to formulate PEP as convex/SDP prob.

# Two ingredients to formulate PEP as convex/SDP prob.

1. explicit interpolation inequalities for studied class of functions.

   For example, smooth convex interpolation:

   Set $\{(x_i, g_i, f_i)\}_{i\in S}$ is interpolable by a function $f \in \mathcal{F}_{0,L}$

   $\Leftrightarrow$ there exists a convex function $f$ with $L$-Lipschitz gradient such that $f(x_i) = f_i$ and $\nabla f(x_i) = g_i$ for all $i \in S$

   $\Leftrightarrow$ $f_j \geq f_i + {g_i}^T(x_j - x_i) + \frac{1}{2L}||g_i - g_j||_2^2$ for all $i, j \in S$

# Two ingredients to formulate PEP as convex/SDP prob.

1. explicit interpolation inequalities for studied class of functions.

   For example, smooth convex interpolation:

   Set $\{(x_i, g_i, f_i)\}_{i\in S}$ is interpolable by a function $f \in \mathcal{F}_{0,L}$

   $\Leftrightarrow$ there exists a convex function $f$ with $L$-Lipschitz gradient such that $f(x_i) = f_i$ and $\nabla f(x_i) = g_i$ for all $i \in S$

   $\Leftrightarrow$ $f_j \geq f_i + g_i^T(x_j - x_i) + \frac{1}{2L}||g_i - g_j||_2^2$ for all $i, j \in S$

2. optimization method defined as constraint over $\{(x_i, g_i, f_i)\}_{i\in S}$

   For example, gradient step with constant stepsize :

   $$x_{k+1} = x_k - \tfrac{h}{L}\nabla f(x_k) \quad \Leftrightarrow \quad x_{k+1} = x_k - \tfrac{h}{L} g_k$$

   $$\Leftrightarrow \|x_{k+1} - (x_k - \tfrac{h}{L} g_k)\|^2 = 0$$

# Two ingredients to formulate PEP as convex/SDP prob.

1. explicit interpolation inequalities for studied class of functions.

   For example, smooth convex interpolation:

   Set $\{(x_i, g_i, f_i)\}_{i\in S}$ is interpolable by a function $f \in \mathcal{F}_{0,L}$
   $\Leftrightarrow$ there exists a convex function $f$ with $L$-Lipschitz gradient such that $f(x_i) = f_i$ and $\nabla f(x_i) = g_i$ for all $i \in S$
   $\Leftrightarrow$ $f_j \geq f_i + g_i^T(x_j - x_i) + \frac{1}{2L}||g_i - g_j||_2^2$ for all $i, j \in S$

2. optimization method defined as constraint over $\{(x_i, g_i, f_i)\}_{i\in S}$

   For example, gradient step with constant stepsize :

   $$x_{k+1} = x_k - \tfrac{h}{L}\nabla f(x_k) \quad \Leftrightarrow \quad x_{k+1} = x_k - \tfrac{h}{L} g_k$$
   $$\Leftrightarrow \|x_{k+1} - (x_k - \tfrac{h}{L} g_k)\|^2 = 0$$

3. All conditions above must be convex in variables $f_i$ and products between $x_i$ and $g_i$ (elements of their Gram matrix) Ideally they must also be SDP-representable

# What you obtain when you solve a PEP

1. In all cases:
   numerical value for worst-case performance
   numerical description of worst-case function (interpolate primal)
   numerical proof of worst-case performance (dual multipliers)

# What you obtain when you solve a PEP

1. In all cases:
   numerical value for worst-case performance
   numerical description of worst-case function (interpolate primal)
   numerical proof of worst-case performance (dual multipliers)

2. If you are lucky/clever: explicit formulas for some/all of above
   $\rightarrow$ this requires fitting/guessing algebraic expressions

# What you obtain when you solve a PEP

1. In all cases:
   numerical value for worst-case performance
   numerical description of worst-case function (interpolate primal)
   numerical proof of worst-case performance (dual multipliers)

2. If you are lucky/clever: explicit formulas for some/all of above
   $\rightarrow$ this requires fitting/guessing algebraic expressions

3. In the best case: a rigorous mathematical proof
   $\rightarrow$ this requires a proof with (often tedious) reformulation of worst-case rate as inequality involing sums-of-squares

   Note: by theorem, all valid proofs must be writable as sum-of-squares inequalities based on combinations of necessary and sufficient interpolation inequalities

# Gradient method, $L$-smooth function, constant step-size $\frac{h}{L}$

Worst-case rate for final iterate accuracy, for any $h \in [0, 2]$

$$\max f(x_N) - f^* = \frac{LR^2}{2} \max\left(\frac{1}{2Nh+1}, (1-h)^{2N}\right)$$

We actually know

- analytical expression for worst-case performance
- analytical expression of worst-case function (Huber/quadratic)
- analytical expression of dual multipliers

but rigorous sum-of-squares proof currently known only for $h \leq 1.5$

# Extra step: constants in analytical rates

Our performance estimation problem is parameterized by

- Lipschitz constant $L$ of gradient (smoothness constant)
- Distance $R$ between starting point $x_0$ and minimizer $x_*$

and worst-case final iterate accuracy appears to be always of type

$$f(x_N) - f(x_*) \leq w_*(L, R, N) = \frac{LR^2}{\text{expr(N)}}$$

# Extra step: constants in analytical rates

Our performance estimation problem is parameterized by

- Lipschitz constant $L$ of gradient (smoothness constant)
- Distance $R$ between starting point $x_0$ and minimizer $x_*$

and worst-case final iterate accuracy appears to be always of type

$$f(x_N) - f(x_*) \leq w_*(L, R, N) = \frac{LR^2}{\text{expr(N)}}$$

To avoid having to solve for every value of $L$ and $R$
we use homogeneity properties

# Extra step: exploiting homogeneity

Theorem: worst-case final iterate accuracy is always of type

$$f(x_N) - f(x_*) \leq w_*(L, R, N) = \frac{LR^2}{\text{expr(N)}}$$

- if $L \to \lambda L$ with $\lambda > 0$, worst-case is also multiplied by $\lambda$ (proof: scale $f \to \lambda f$)
- if $R \to \lambda R$ with $\lambda > 0$, worst-case is also multiplied by $\lambda$ (proof: right-scaling $f \to \lambda f(\cdot/\lambda)$)

hence worst-case $w_*(L, R, N)$ is proportional to $B$ and $R$

# Extra step: exploiting homogeneity

Theorem: worst-case final iterate accuracy is always of type

$$f(x_N) - f(x_*) \leq w_*(L, R, N) = \frac{LR^2}{\text{expr(N)}}$$

- if $L \to \lambda L$ with $\lambda > 0$, worst-case is also multiplied by $\lambda$ (proof: scale $f \to \lambda f$)
- if $R \to \lambda R$ with $\lambda > 0$, worst-case is also multiplied by $\lambda$ (proof: right-scaling $f \to \lambda f(\cdot/\lambda)$)

hence worst-case $w_*(L, R, N)$ is proportional to $B$ and $R$

# Extra step: exploiting homogeneity

Theorem: worst-case final iterate accuracy is always of type

$$f(x_N) - f(x_*) \leq w_*(L, R, N) = \frac{LR^2}{\text{expr(N)}}$$

- if $L \to \lambda L$ with $\lambda > 0$, worst-case is also multiplied by $\lambda$ (proof: scale $f \to \lambda f$)
- if $R \to \lambda R$ with $\lambda > 0$, worst-case is also multiplied by $\lambda$ (proof: right-scaling $f \to \lambda f(\cdot/\lambda)$)

hence worst-case $w_*(L, R, N)$ is proportional to $B$ and $R$

$\to$ hence only solve problem for $L = R = 1$, find worst-case value $w_*(1, 1, N)$ so that for general $L$ and $R$ we will have

$$f(x_N) - f(x_*) \leq w_*(L, R, N) = LR \cdot w_*(1, 1, N)$$

Only remaining parameter is number of steps $N$ (plus possibly other algorithmic/function class parameters)

# Outline

- Performance estimation
  - Recap from yesterday
  - Gradient method for smooth convex functions
- Beyond smooth convex functions
  - Nonsmooth functions: subgradient method
  - Strongly convex, non-convex/hypoconvex
- Software toolboxes: PESTO and PEPit
- Beyond (fixed-step) gradient methods
  - Projected gradient method
  - Methods using inexact gradient
  - Methods involving linear mappings
- A few reflections and open questions

# Nonsmooth optimization: subgradient method

First-order methods can deal with nonsmooth convex functions using the concept of subgradient

$$g \in \partial f(x) \quad \Leftrightarrow \quad f(y) \geq f(x) + g^T(y - x) \text{ for all } y$$

Sugradient method is then simply

$$x_{i+1} = x_i - hg_i \text{ for some } g_i \in \partial f(x_i)$$

Worst-case rates on objective accuracy require

- Distance $R$ from initial iterate $x_0$ to minimizer $x^*$
- Bound $B$ on maximum norm of any subgradient $g \in \partial f(x)$

# Interpolation conditions for nonsmooth convex functions

We need explicit conditions for the following

| |
|---|
| there exists proper and convex $f$ with $B$-bounded subgradients such that $f(x_i) = f_i$ and $g_i \in \partial f(x_i)$ for all $i \in I = \{*, 0, 1, \dots N\}$ |

i.e. given values of $x_i$, $f_i$ and $g_i$ we need to guarantee existence of $f$

We use the well-known (and easy to show) equivalence

$$\text{there exists proper and convex } f \text{ satisfying}$$
$$f(x_i) = f_i \text{ and } g_i \in \partial f(x_i) \text{ for every } i \in I$$
$$\Leftrightarrow$$
$$f_j \geq f_i + g_i(x_j - x_i) \text{ for every } i, j \in I$$

which can be extended to deal with $B$-bounded subgradients

# Interpolation conditions for nonsmooth convex functions

We need explicit conditions for the following

> there exists proper and convex $f$ with $B$-bounded subgradients such that $f(x_i) = f_i$ and $g_i \in \partial f(x_i)$ for all $i \in I = \{*, 0, 1, \ldots N\}$

i.e. given values of $x_i$, $f_i$ and $g_i$ we need to guarantee existence of $f$

We use the well-known (and easy to show) equivalence

$$\text{there exists proper and convex } f \text{ satisfying}$$
$$f(x_i) = f_i \text{ and } g_i \in \partial f(x_i) \text{ for every } i \in I$$
$$\Leftrightarrow$$
$$f_j \geq f_i + g_i(x_j - x_i) \text{ for every } i, j \in I$$

which can be extended to deal with $B$-bounded subgradients

# Interpolation conditions for nonsmooth convex functions

We need explicit conditions for the following

> there exists proper and convex $f$ with $B$-bounded subgradients such that $f(x_i) = f_i$ and $g_i \in \partial f(x_i)$ for all $i \in I = \{*, 0, 1, \dots N\}$

i.e. given values of $x_i$, $f_i$ and $g_i$ we need to guarantee existence of $f$

We use the well-known (and easy to show) equivalence

there exists proper and convex $f$ with $B$-bounded subgradients s.t.

$$f(x_i) = f_i \text{ and } g_i \in \partial f(x_i) \text{ for every } i \in I$$

$$\Leftrightarrow$$

$$f_j \geq f_i + g_i^T (x_j - x_i) \text{ for every } i, j \in I$$

$$\|g_i\| \leq B \text{ for every } i \in I$$

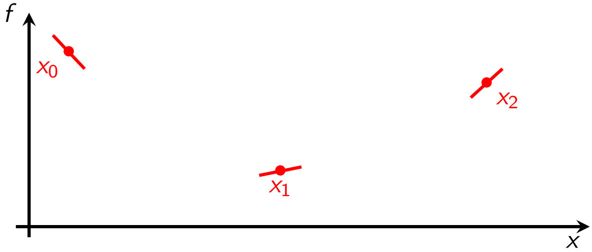
Leads to a convex formulation (using $g_i^T g_i \leq B^2$)

# Illustration: nonsmooth convex interpolation problem

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

- Is there $f \in \mathcal{F}_{0,\infty}$ (proper, closed, convex) s.t.

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

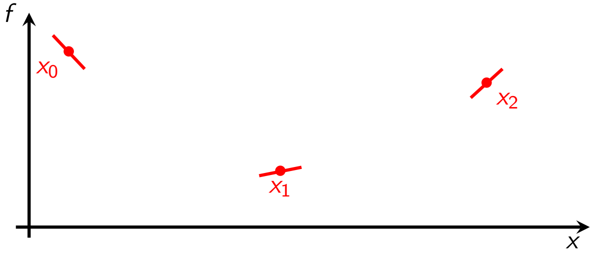


# Illustration: nonsmooth convex interpolation problem

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

- Is there $f \in \mathcal{F}_{0,\infty}$ (proper, closed, convex) s.t.

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



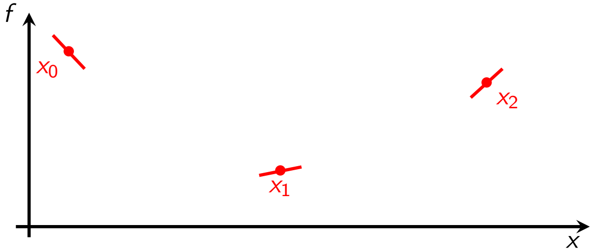

- We want necessary and sufficient conditions for existence of $f$

# Illustration: nonsmooth convex interpolation problem

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

- Is there $f \in \mathcal{F}_{0,\infty}$ (proper, closed, convex) s.t.

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



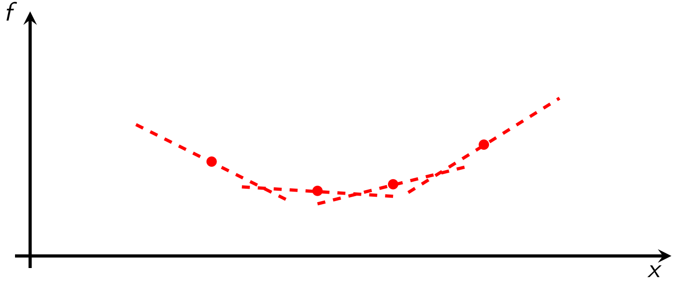

- We want necessary and sufficient conditions for existence of $f$
- These conditions will appear as a constraints in our PEP formulation

# Convex interpolation

Conditions for $\{(x_i, g_i, f_i)\}_{i\in S}$ to be interpolable by a function $f \in \mathcal{F}_{0,\infty}$ (proper, closed and convex function) ?

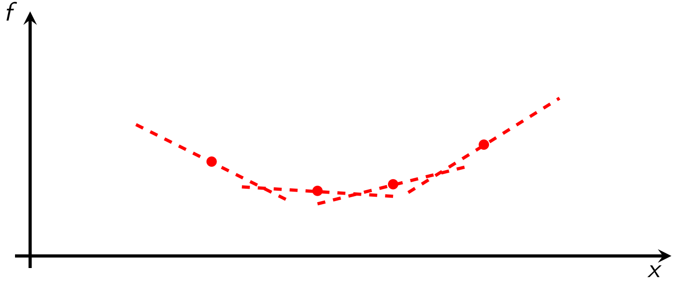
# Convex interpolation

Conditions for $\{(x_i, g_i, f_i)\}_{i\in S}$ to be interpolable by a function $f \in \mathcal{F}_{0,\infty}$ (proper, closed and convex function) ?



# Convex interpolation

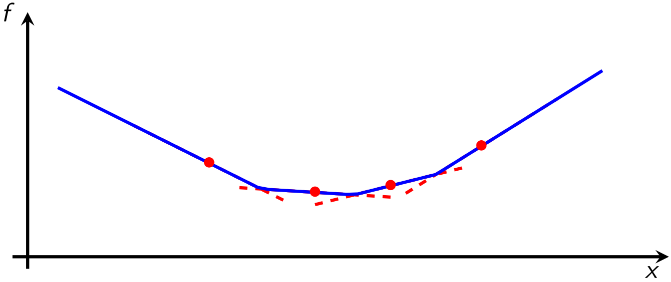
Conditions for $\{(x_i, g_i, f_i)\}_{i \in S}$ to be interpolable by a function $f \in \mathcal{F}_{0,\infty}$ (proper, closed and convex function) ?




Conditions $f_i \geq f_j + g_j^T(x_i - x_j)$ is necessary

# Convex interpolation

Conditions for $\{(x_i, g_i, f_i)\}_{i \in S}$ to be interpolable by a function $f \in \mathcal{F}_{0,\infty}$ (proper, closed and convex function) ?




Conditions $f_i \geq f_j + g_j^T(x_i - x_j)$ is necessary and sufficient

Explicit construction of a (piecewise linear) interpolating function:

$$f(x) = \max_j \left\{ f_j + g_j^T(x - x_j) \right\},$$

Not unique.

# Results: average iterate

Worst-case for fixed-step subgradient method

$$x_{i+1} = x_i - h(\tfrac{R}{B})g_i$$

applied to convex function with $B$-bounded sugradients

- For average value of iterates $\hat{f}_N = \frac{f(x_0)+f(x_1)+\ldots+f(x_N)}{N+1}$, tight worst-case is

$$\hat{f}_N - f(x_*) \leq \begin{cases} BR\left(1 - \frac{N}{2}h\right) & \text{when } h \leq \frac{1}{N+1} \\ BR\left(\frac{1}{2}h + \frac{1}{2N+2}\frac{1}{h}\right) & \text{when } h \geq \frac{1}{N+1} \end{cases}$$

  (recovers a well-known result for large $h$)

- Optimal constant step-size is then $h^* = \frac{1}{\sqrt{N+1}}$ (always in the second case) leading to tight worst-case

$$\hat{f}_N - f(x_*) \leq \frac{BR}{\sqrt{N+1}}$$

# Results: last iterate

- Define sequence $\{s_N\}_{N\geq 0}$ with $s_0 = 1, s_{i+1} = s_i + \frac{1}{s_i}\ \forall i \geq 0$ ($\rightarrow s_1 = 2,\ s_2 = \frac{5}{2} = 2.5,\ s_3 = \frac{29}{10} = 2.9$, etc.)
- Sequence $s_N$ grows like $\sqrt{2N + \frac{1}{2}\log(N) + 2}$ (no closed form)

# Results: last iterate

- Define sequence $\{s_N\}_{N\geq 0}$ with $s_0 = 1, s_{i+1} = s_i + \frac{1}{s_i}\ \forall i \geq 0$ ($\rightarrow s_1 = 2,\ s_2 = \frac{5}{2} = 2.5,\ s_3 = \frac{29}{10} = 2.9$, etc.)
- Sequence $s_N$ grows like $\sqrt{2N + \frac{1}{2}\log(N) + 2}$ (no closed form)
- For value of last iterate $f(x_N)$, tight worst-case is

$$f(x_N) - f(x_*) \leq BR\Big((\tfrac{1}{2}s_N^2 - N)h + (2s_N^2)^{-1}\frac{1}{h}\Big) \text{ when } h \geq \frac{1}{s_N}$$

(another less interesting regime holds for smaller $h$)

# Results: last iterate

- Define sequence $\{s_N\}_{N\geq 0}$ with $s_0 = 1, s_{i+1} = s_i + \frac{1}{s_i}\ \forall i \geq 0$ ($\rightarrow s_1 = 2,\ s_2 = \frac{5}{2} = 2.5,\ s_3 = \frac{29}{10} = 2.9$, etc.)
- Sequence $s_N$ grows like $\sqrt{2N + \frac{1}{2}\log(N) + 2}$ (no closed form)
- For value of last iterate $f(x_N)$, tight worst-case is

$$f(x_N) - f(x_*) \leq BR\Big((\tfrac{1}{2}s_N^2 - N)h + (2s_N^2)^{-1}\frac{1}{h}\Big) \text{ when } h \geq \frac{1}{s_N}$$

  (another less interesting regime holds for smaller $h$)
- Optimal step is then $h^* = 1/\sqrt{s_N^2(s_N^2 - 2N)}$ and corresponding worst-case value is

$$f(x_N) - f(x_*) \leq BR \cdot \Theta\Big(\sqrt{\frac{\log(N+1)}{N+1}}\Big)$$

  apparently new (previous results only on average/best iterate)

# Dealing with smooth strongly convex functions

Strongly convex functions $\Leftrightarrow$ lower bound $\mu > 0$ on curvature

To tackle smooth strongly convex functions (class $\mathcal{F}_{\mu,L}$) we only need one new ingredient: suitable interpolation conditions

Correct necessary and sufficient conditions are given by following Theorem [Taylor, Hendrickx, G. 2016]

Set $\{(x_i, g_i, f_i)\}_{i \in S}$ is $\mathcal{F}_{\mu,L}$-interpolable if and only

$$f_i - f_j - g_j^\top (x_i - x_j) \geq \frac{1}{2(1-\mu/L)} \left( \frac{1}{L} \|g_i - g_j\|_2^2 \cdots \right.$$
$$\left. + \mu \|x_i - x_j\|_2^2 - 2\frac{\mu}{L}(g_j - g_i)^\top (x_j - x_i) \right)$$

holds for every pair of indices $i \in I$ and $j \in S$

(generalizes conditions for smooth convex interpolation)

# Gradient method for smooth strongly convex functions

$$x_{i+1} = x_i - \frac{h}{L}\nabla f(x_i)$$

Linear convergence for all performance criteria, with same rate
$\rho = \max\{(1 - Lh)^2, (1 - \mu h)^2\}$ [Taylor, Hendrickx, G, 2018]

$$\begin{aligned} \|x_k - x_*\|^2 &\leq \rho^k \|x_0 - x_*\|^2 \\ \|\nabla f(x_k)\|^2 &\leq \rho^k \|\nabla f(x_0)\|^2 \\ f(x_k) - f(x_*) &\leq \rho^k \big(f(x_0) - f(x_*)\big) \end{aligned}$$

- All results with fully rigorous PEP-type mathematical proofs (pure linear rates $\rightarrow$ sufficient to prove them for one step)

# Gradient method for smooth strongly convex functions

$$x_{i+1} = x_i - \frac{h}{L}\nabla f(x_i)$$

Linear convergence for all performance criteria, with same rate $\rho = \max\{(1 - Lh)^2, (1 - \mu h)^2\}$ [Taylor, Hendrickx, G, 2018]

$$\begin{aligned} \|x_k - x_*\|^2 &\leq \rho^k \|x_0 - x_*\|^2 \\ \|\nabla f(x_k)\|^2 &\leq \rho^k \|\nabla f(x_0)\|^2 \\ f(x_k) - f(x_*) &\leq \rho^k \big(f(x_0) - f(x_*)\big) \end{aligned}$$

- All results with fully rigorous PEP-type mathematical proofs (pure linear rates $\rightarrow$ sufficient to prove them for one step)
- Only previously known in some special cases

# Gradient method for smooth strongly convex functions

$$x_{i+1} = x_i - \frac{h}{L}\nabla f(x_i)$$

Linear convergence for all performance criteria, with same rate $\rho = \max\{(1-Lh)^2, (1-\mu h)^2\}$ [Taylor, Hendrickx, G, 2018]

$$\begin{aligned} \|x_k - x_*\|^2 &\leq \rho^k \|x_0 - x_*\|^2 \\ \|\nabla f(x_k)\|^2 &\leq \rho^k \|\nabla f(x_0)\|^2 \\ f(x_k) - f(x_*) &\leq \rho^k \big(f(x_0) - f(x_*)\big) \end{aligned}$$

- All results with fully rigorous PEP-type mathematical proofs (pure linear rates $\rightarrow$ sufficient to prove them for one step)
- Only previously known in some special cases
- Optimal steplength is $h^* = \frac{2}{L+\mu}$ with $\rho^* = \left(\frac{L-\mu}{L+\mu}\right)^2$

# Gradient method for smooth strongly convex functions

$$x_{i+1} = x_i - \frac{h}{L}\nabla f(x_i)$$

Linear convergence for all performance criteria, with same rate
$\rho = \max\{(1 - Lh)^2, (1 - \mu h)^2\}$ [Taylor, Hendrickx, G, 2018]

$$\begin{aligned}
\|x_k - x_*\|^2 &\leq \rho^k \|x_0 - x_*\|^2 \\
\|\nabla f(x_k)\|^2 &\leq \rho^k \|\nabla f(x_0)\|^2 \\
f(x_k) - f(x_*) &\leq \rho^k \big(f(x_0) - f(x_*)\big)
\end{aligned}$$

- All results with fully rigorous PEP-type mathematical proofs (pure linear rates $\rightarrow$ sufficient to prove them for one step)
- Only previously known in some special cases
- Optimal steplength is $h^* = \frac{2}{L+\mu}$ with $\rho^* = \left(\frac{L-\mu}{L+\mu}\right)^2$
- Worst-case functions are 1D quadratics $\frac{\mu}{2}x^2$ and $\frac{L}{2}x^2$

# Gradient method for smooth strongly convex functions

$$x_{i+1} = x_i - \frac{h}{L}\nabla f(x_i)$$

Linear convergence for all performance criteria, with same rate $\rho = \max\{(1-Lh)^2, (1-\mu h)^2\}$ [Taylor, Hendrickx, G, 2018]

$$\begin{aligned} \|x_k - x_*\|^2 &\leq \rho^k \|x_0 - x_*\|^2 \\ \|\nabla f(x_k)\|^2 &\leq \rho^k \|\nabla f(x_0)\|^2 \\ f(x_k) - f(x_*) &\leq \rho^k \left(f(x_0) - f(x_*)\right) \end{aligned}$$

- All results with fully rigorous PEP-type mathematical proofs (pure linear rates $\rightarrow$ sufficient to prove them for one step)
- Only previously known in some special cases
- Optimal steplength is $h^* = \frac{2}{L+\mu}$ with $\rho^* = \left(\frac{L-\mu}{L+\mu}\right)^2$
- Worst-case functions are 1D quadratics $\frac{\mu}{2}x^2$ and $\frac{L}{2}x^2$
- When $\mu = 0$ we obtain $\rho = 1$, i.e. no convergence, which is tight (!) $\rightarrow$ other convergence results needed

# Further results: mixed performance criteria

- Smooth strongly convex case, with condition number $\kappa = \mu/L$
  [Taylor, Hendrickx, G, 2018]

$$\max\ f(x_N) - f_* = \frac{LR^2}{2} \max\left( \frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-2N}}, (1 - h)^{2N} \right)$$

# Further results: mixed performance criteria

- Smooth strongly convex case, with condition number $\kappa = \mu/L$
  [Taylor, Hendrickx, G, 2018]

$$\max\ f(x_N) - f_* = \frac{LR^2}{2} \max\left( \frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-2N}}, (1 - h)^{2N} \right)$$

- Residual gradient norm, strongly convex case

$$\max\ \|\nabla f(x_N)\|_2 = LR \max\left( \frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-N}}, |1 - h|^N \right)$$

# Further results: mixed performance criteria

- Smooth strongly convex case, with condition number $\kappa = \mu/L$ [Taylor, Hendrickx, G, 2018]

$$\max\ f(x_N) - f_* = \frac{LR^2}{2} \max\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h)^{-2N}}, (1-h)^{2N}\right)$$

- Residual gradient norm, strongly convex case

$$\max\ \|\nabla f(x_N)\|_2 = LR \max\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h)^{-N}}, |1-h|^N\right)$$

- Simple $1D$ piecewise linear-quadratic solutions in all cases (different for each case)

# Further results: mixed performance criteria

- Smooth strongly convex case, with condition number $\kappa = \mu/L$ [Taylor, Hendrickx, G, 2018]

$$\max\ f(x_N) - f_* = \frac{LR^2}{2} \max\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h)^{-2N}}, (1-h)^{2N}\right)$$

- Residual gradient norm, strongly convex case

$$\max\ \|\nabla f(x_N)\|_2 = LR \max\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h)^{-N}}, |1-h|^N\right)$$

- Simple $1D$ piecewise linear-quadratic solutions in all cases (different for each case)
- All results lead to optimal step-sizes

# Further results: mixed performance criteria

- Smooth strongly convex case, with condition number $\kappa = \mu / L$
  [Taylor, Hendrickx, G, 2018]

  $$\max\ f(x_N) - f_* = \frac{LR^2}{2} \max\left( \frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-2N}}, (1-h)^{2N} \right)$$

- Residual gradient norm, strongly convex case

  $$\max\ \|\nabla f(x_N)\|_2 = LR \max\left( \frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-N}}, |1-h|^{N} \right)$$

- Simple $1D$ piecewise linear-quadratic solutions in all cases (different for each case)
- All results lead to optimal step-sizes
- Smooth convex case is recovered exactly as the limit $\mu \to 0$ (expression for worst-case is continuous in $\mu$)

# Dealing with smooth nonconvex functions

Smooth nonconvex actually quite similar to smooth convex:

smooth nonconvex functions $\Leftrightarrow$ curvature must belong to $[-L, L]$

To tackle smooth nonconvex functions (class $\mathcal{F}_{-L,L}$) we only need one new ingredient: suitable interpolation conditions

It turns out that interpolation conditions for smooth strongly convex functions in class $\mathcal{F}_{\mu,L}$ also work, with exactly the same expressions, when $\mu$ is negative !

Hence smooth nonconvex interpolations conditions are obtained simply by taking $\mu = -L$

# Dealing with nonconvex and hypoconvex functions

We can even interpolate smoothly between smooth convex ($\mathcal{F}_{0,L}$) and smooth nonconvex ($\mathcal{F}_{-L,L}$) with all values of $\mu \in ]-L, 0[$

This leads to the class of hypoconvex functions $\mathcal{F}_{\mu,L}$ for any $\mu < 0$

Necessary and sufficient interpolation conditions are:
[Taylor, 2017] [Rotaru, Glineur, Patrinos, 2022] [Abbaszadehpeivasti, de Klerk, Zamani, 2022]
Set $\{(x_i, g_i, f_i)\}_{i \in S}$ is $\mathcal{F}_{\mu,L}$-interpolable if and only

$$f_i - f_j - g_j^\top (x_i - x_j) \geq \frac{1}{2(1-\mu/L)} \left( \frac{1}{L} \|g_i - g_j\|_2^2 \cdots \right.$$
$$\left. + \mu \|x_i - x_j\|_2^2 - 2\frac{\mu}{L}(g_j - g_i)^\top (x_j - x_i) \right)$$

holds for every pair of indices $i \in I$ and $j \in S$

(again generalizes conditions for smooth convex and smooth strongly convex interpolation)

# Gradient method for smooth nonconvex/hypoconvex

Example of (fully analytical) worst-case rate [Abbaszadehpeivasti, de Klerk, Zamani, 2022], [Rotaru, Glineur, Patrinos, 2022]

Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ be a smooth hypoconvex function, with $L > 0$ and $\mu \leq 0$, and let $\kappa := \frac{\mu}{L}$.

Consider $N$ iterations of the gradient method $x_{i+1} = x_i - \frac{h_i}{L}\nabla f(x_i)$ where stepsizes $h_i$ are less than $\bar{h}(\kappa) := \frac{3}{1+\kappa+\sqrt{1-\kappa+\kappa^2}} \in [\frac{3}{2}, 2)$ (hence non-constant step-sizes are allowed)

Then we have

$$\min_{0 \leq i \leq N} \left\{ \|\nabla f(x_i)\|^2 \right\} \leq \frac{2L\left[f(x_0) - f(x_N)\right]}{\sum_{i=0}^{N-1} p(h_i, \kappa)}$$

# Gradient method for smooth nonconvex/hypoconvex

$$\min_{0 \le i \le N} \left\{ \|\nabla f(x_i)\|^2 \right\} \le \frac{2L\big[f(x_0) - f(x_N)\big]}{\sum_{i=0}^{N-1} p(h_i, \kappa)}$$

where each of the $N$ terms in denominator is given by

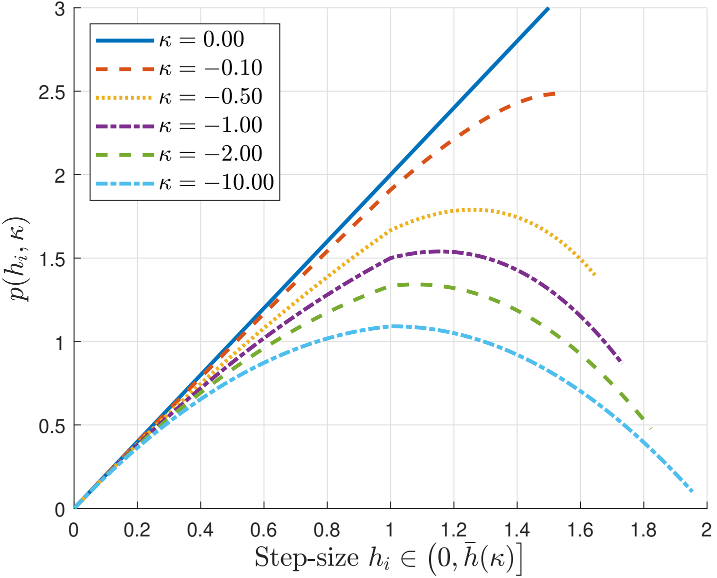
$$p(h_i, \kappa) = \begin{cases} 2h_i - h_i^2 \frac{-\kappa}{1-\kappa} & \text{if } h_i \in \big(0, 1\big] \\ \frac{h_i(2-h_i)(2-\kappa h_i)}{2-(1+\kappa)h_i} & \text{if } h_i \in \big[1, \bar{h}(\kappa)\big] \end{cases}$$

Additionally, if $f$ is bounded from below

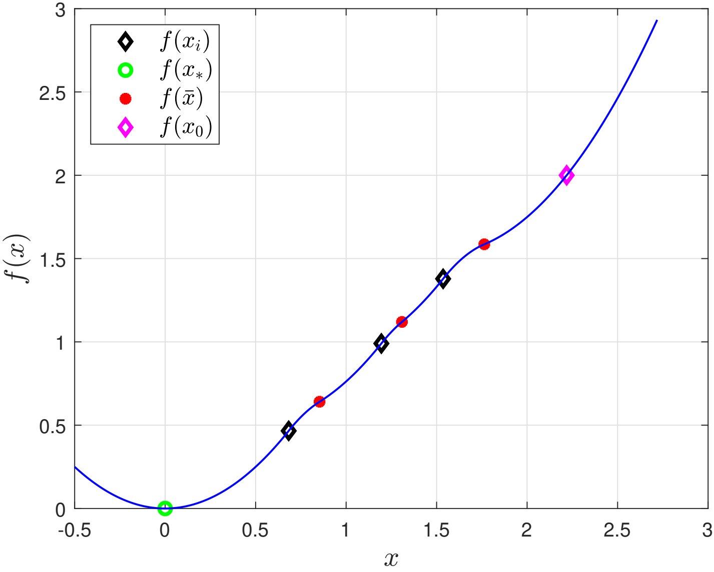
$$\min_{0 \le i \le N} \left\{ \|\nabla f(x_i)\|^2 \right\} \le \frac{2L\big[f(x_0) - f_*\big]}{1 + \sum_{i=0}^{N-1} p(h_i, \kappa)}$$

(even covers the smooth convex case in the limit $\mu \to 0$)

# Terms $p(h_i, \kappa)$ in the denominator vs. stepsize $h_i$



# Example of a worst-case function




$N = 3,\ f_0 - f_* = 2,\ L = 2,\ \kappa = -2,\ h_0 = 1,\ h_1 = 0.5,\ h_2 = 0.75$

# Outline

Performance estimation

- Recap from yesterday
- Gradient method for smooth convex functions

Beyond smooth convex functions

- Nonsmooth functions: subgradient method
- Strongly convex, non-convex/hypoconvex

Software toolboxes: PESTO and PEPit

Beyond (fixed-step) gradient methods

- Projected gradient method
- Methods using inexact gradient
- Methods involving linear mappings

A few reflections and open questions

# Toolboxes

[Hendrickx, G, Goujaud, Moucer, Taylor]

Please visit https://github.com/PerformanceEstimation

## PerformanceEstimation

This organization regroups works/packages/toolboxes related to performance estimation problems.

If you find some content useful, please don't hesitate to give feedbacks and or to star the content!

### Current packages

- **PESTO**: allows a quick access to performance estimation problems in Matlab.
- **PEPit**: allows a quick access to performance estimation problems in Python.

### Education

- Informal introduction to PEPs: **here**.
- Practical exercises: **here**.

### Events

If you organize a PEP-event, we would be happy to list your event below.

Upcoming events:

- February 2023: **PEP-talks**.

# Example: subgradient method with $h_k = \frac{1}{\sqrt{N+1}}$

```
P = pep();

param.R        = 1;
F              = P.DeclareFunction(
                 'ConvexBoundedGradient',param);
[xstar,fstar] = F.OptimalPoint();

x0 = P.StartingPoint();
P.InitialCondition((x0-xstar)^2<=1);

N=5;
x=x0;
for i=1:N
    [g,f] = F.oracle(x);
    x     = x - 1/sqrt(N+1)*g;
end

xN = x;
fN=F.value(xN);
P.PerformanceMetric(fN-fstar);

P.solve()
disp(double(fN - fstar))
```

# Contains numerous introductory examples

(tries to keep up with litterature)

- 1. Unconstrained convex minimization
  - 1.1. Gradient descent
  - 1.2. Subgradient method
  - 1.3. Subgradient method under restricted secant inequality and error bound
  - 1.4. Gradient descent with exact line search
  - 1.5. Conjugate gradient
  - 1.6. Heavy Ball momentum
  - 1.7. Accelerated gradient for convex objective
  - 1.8. Accelerated gradient for strongly convex objective
  - 1.9. Optimized gradient
  - 1.10. Optimized gradient for gradient
  - 1.11. Robust momentum
  - 1.12. Triple momentum
  - 1.13. Information theoretic exact method
  - 1.14. Cyclic coordinate descent
  - 1.15. Proximal point
  - 1.16. Accelerated proximal point
  - 1.17. Inexact gradient descent
  - 1.18. Inexact gradient descent with exact line search
  - 1.19. Inexact accelerated gradient
  - 1.20. Epsilon-subgradient method
  - 1.21. Gradient descent for quadratically upper bounded convex objective
  - 1.22. Gradient descent with decreasing step sizes for quadratically upper bounded convex objective
  - 1.23. Conjugate gradient for quadratically upper bounded convex objective
  - 1.24. Heavy Ball momentum for quadratically upper bounded convex objective

# Contains numerous introductory examples

(tries to keep up with litterature)

# Contains numerous introductory examples

(tries to keep up with litterature)

- 5. Monotone inclusions and variational inequalities
  - 5.1. Proximal point
  - 5.2. Accelerated proximal point
  - 5.3. Optimal Strongly-monotone Proximal Point
  - 5.4. Douglas Rachford Splitting
  - 5.5. Three operator splitting
  - 5.6. Optimistic gradient
  - 5.7. Past extragradient
- 6. Fixed point
  - 6.1. Halpern iteration
  - 6.2. Optimal Contractive Halpern iteration
  - 6.3. Krasnoselskii-Mann with constant step-sizes
  - 6.4. Krasnoselskii-Mann with increasing step-sizes
- 7. Potential functions
  - 7.1. Gradient descent Lyapunov 1
  - 7.2. Gradient descent Lyapunov 2
  - 7.3. Accelerated gradient method
- 8. Inexact proximal methods
  - 8.1. Accelerated inexact forward backward
  - 8.2. Partially inexact Douglas Rachford splitting
  - 8.3. Relatively inexact proximal point
- 9. Adaptive methods
  - 9.1. Polyak steps in distance to optimum
  - 9.2. Polyak steps in function value

# Contains numerous introductory examples

(tries to keep up with litterature)

- 10. Low dimensional worst-cases scenarios
  - 10.1. Inexact gradient
  - 10.2. Non-convex gradient descent
  - 10.3. Optimized gradient
  - 10.4. Frank Wolfe
  - 10.5. Proximal point
  - 10.6. Halpern iteration
  - 10.7. Alternate projections
  - 10.8. Averaged projections
  - 10.9. Dykstra
- 11. Continuous-time models
  - 11.1. Gradient flow for strongly convex functions
  - 11.2. Gradient flow for convex functions
  - 11.3. Accelerated gradient flow for strongly convex functions
  - 11.4. Accelerated gradient flow for convex functions
- 12. Tutorials
  - 12.1. Contraction rate of gradient descent

# Outline

Performance estimation

- Recap from yesterday
- Gradient method for smooth convex functions

Beyond smooth convex functions

- Nonsmooth functions: subgradient method
- Strongly convex, non-convex/hypoconvex

Software toolboxes: PESTO and PEPit

Beyond (fixed-step) gradient methods

- Projected gradient method
- Methods using inexact gradient
- Methods involving linear mappings

A few reflections and open questions

# Projected gradient and proximal methods

We can actually handle with little extra effort

- first-order methods for smooth constrained convex optimization i.e. express projection steps in our formulation

$$x_{k+1} = P_C\left[x_k - \frac{1}{L}\nabla f(x_k)\right]$$

- proximal algorithms i.e. express proximal steps

$$x_{k+1} = \mathrm{prox}_L f(x_k) = \arg\min_u \left(f(u) + \frac{1}{2L}\|u - x_k\|^2\right)$$

- composite minimization: $\min f(x) + h(x)$ where $f$ is smooth and $h$ is proximable, using proximal gradient method

$$x_{k+1} = \mathrm{prox}_L h(x_k - \tfrac{1}{L}\nabla f(x_k))$$

# Projected gradient and proximal methods

$$x_+ = \mathrm{prox}_L f(x) = \arg\min_u \left( f(u) + \frac{1}{2L}\|u - x\|^2 \right)$$

- Key idea: proximal steps can be formulated as

  $$x_+ = \mathrm{prox}_L f(x) \quad \Leftrightarrow \quad x_+ - \tfrac{1}{L} g_+ = x \text{ and } g_+ \in \partial f(x_+)$$

  which is a linear condition involving iterates and oracle outputs

- Proximal gradient for composite optimization $\min f(x) + h(x)$ can be decomposed in two successive, independent steps: $x_+ = \mathrm{prox}_L h(x - \frac{1}{L}\nabla f(x))$ is equivalent

  $$y = x - \tfrac{1}{L}\nabla f(x) \quad \text{then} \quad x_+ = \mathrm{prox}_L h(y)$$

- Projected gradient = proximal gradient using indicator function of set $C$ for nonsmooth term $h$
- Requires corresponding interpolation conditions (e.g. for indicator function $\mathbb{I}_C(x)$ of a convex set)
- Linear rates unchanged in smooth strongly convex case!

# Methods using inexact gradient

Instead of computing $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$ with exact gradient assume gradient is computed inexactly with bounded error

$$\tilde{g}_k \approx \nabla f(x_k) \text{ such that } \|\tilde{g}_k - \nabla f(x_k)\| \leq \Delta$$

Key technique: rewrite step with inexact gradient

$$x_{k+1} = x_k - \tilde{g}_k$$

as

$$x_{k+1} = x_k - \nabla f(x_k) - \big(\tilde{g}_k - \nabla f(x_k)\big)$$

and then observe it is equivalent to

$$L(x_k - \nabla f(x_k) - x_{k+1}) = \tilde{g}_k - \nabla f(x_k)$$

which can be written using the assumption on the error as

$$L\|x_k - \nabla f(x_k) - x_{k+1}\| \leq \Delta$$

Leads to a convex SDP formulation (after squaring both sides)

# Methods involving linear mappings

[Bousselmi, Hendrickx, Glineur, 2023]

We want to minimize $g(Mx)$ (alone/in composite objective) where $M$ is a linear mapping with some characteritcs

- $M$ symmetric and constraints on minimum/maximum eigenvalues $[\mu, L]$
- $M$ non-symmetric (possibly rectangular) with maximum singular value $S$

A gradient step on $F(x) = g(Mx)$ requires gradient

$$\nabla F(x) = M^T \nabla g(Mx)$$

which can be decomposed as three successive operations:

$$\begin{aligned} y &= Mx \\ u &= \nabla g(y), \\ v &= M^T u = \nabla F(x). \end{aligned}$$

# Gradient of function composed with linear mapping

In order to compute worst-case for methods involving

$$\begin{aligned} y &= Mx \\ u &= \nabla g(y), \\ v &= M^T u = \nabla F(x). \end{aligned}$$

we need to interpolate the following

$$\begin{aligned} y_i &= Mx_i, \\ u_i &= \nabla g(y_i), \\ v_i &= M^T u_i = \nabla F(x_i). \end{aligned}$$

New expressions: $y_i = Mx_i$ and $v_i = M^T u_i$

Requires interpolability of two sequences $\{x_i, y_i\}$ and $\{u_i, v_i\}$ by a linear mapping $M$ and its transpose $M^T$

# Interpolation theorem for linear mappings

For simplicity of notation we represent sequence $\{x_i\}$ as a matrix $X$ (columns are $x_i$), same for $\{y_i\}$, $\{u_i\}$, $\{v_i\}$

Let $X \in \mathbb{R}^{m \times N_1}$, $Y \in \mathbb{R}^{n \times N_1}$, $U \in \mathbb{R}^{m \times N_2}$ and $V \in \mathbb{R}^{n \times N_2}$.
$(X, Y, U, V)$ is $\mathbb{R}_S^{m \times n}$-matrix-interpolable if, and only if,

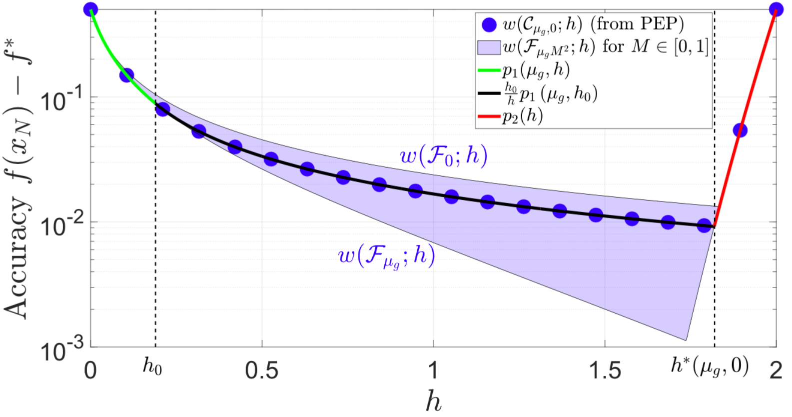
$$\begin{cases} X^T V = Y^T U, \\ Y^T Y \preceq S^2 X^T X, \\ V^T V \preceq S^2 U^T U. \end{cases}$$

(where $\preceq$ denotes the Lowner = positive semidefinite order)

Moreover, if $U = X$ and $V = Y$ (resp. $V = -Y$), the interpolant matrix can be chosen symmetric (resp. skew-symmetric).

[Bousselmi, Hendrickx, Glineur, 2023]

# Example result



# Outline

- Performance estimation
  - Recap from yesterday
  - Gradient method for smooth convex functions
- Beyond smooth convex functions
  - Nonsmooth functions: subgradient method
  - Strongly convex, non-convex/hypoconvex
- Software toolboxes: PESTO and PEPit
- Beyond (fixed-step) gradient methods
  - Projected gradient method
  - Methods using inexact gradient
  - Methods involving linear mappings
- A few reflections and open questions

# Reflections

- Automated procedure to compute worst-case rates
- But: is a PEP proof the final goal?

- More than once, these steps actually happened
  1. Numerical rate computed, analytical expression guessed/identified/confirmed
  2. Using insight provided by worst-case/proof, a closer look at rate/proof provides further intuition and new ideas
  3. Result can now be derived in standard way (and its “PEP” origin becomes unnoticeable!)

- Even when such simplifications are not found, shouldn't the goal of mathematical optimization theory first and foremost to increase our understanding?

# To conlude: a few open questions

- Some rates are known explicitly (including multipliers) but no proof available

  $$\max f(x_N) - f^* = \frac{LR^2}{2} \max\left(\frac{1}{2Nh+1}, (1-h)^{2N}\right) \text{ for } h > 1.5$$

  (should not underestimate difficulty, even with explicit expressions: original proof by Drori and Teboulle for gradient method with $h \leq 1$ required three dense pages of matrix analysis/algebra)

- Can we design first-order methods using PEP?

  Main difficulty: considering method coefficients to be variable (e.g. stepsizes $h_k$) destroys convexity

  Attempts try (and succed in) solving resulting nonconvex SDP [Das Gupta, Van Parys, Ryu, 2022]

  Is there a non obvious convex formulation for method design?

# To conlude: a few open questions

- Some classes lack necessary and sufficient interpolation conditions

  Example: convex functions with coordinate smoothness
  More generally: intersections of two or more classes
  (grouping conditions is necessary, but not always sufficient)

- How come worst-case functions are univariate for so many methods (or in rare cases two-dimensional)?
  Is there a fundamental reason for that?

- Can we go beyond first-order methods with fixed steps?
  Nonlinear stepsize rules: some recent success for nonlinear conjugate gradient [Das Gupta, Freud, Sun, Taylor, 2023]

  What about second-order methods? Interior-point methods?
  Higher order/tensor method?

# Thank you again for your attention!

References (strongly convex + constrained/proximal/composite):

Exact worst-case convergence rates of the proximal gradient method for composite convex minimization, Adrien B. Taylor, Julien M. Hendrickx, François Glineur, J. of Optim. Theory and Applic. (2018) vol. 178 (2)

Exact Worst-case Performance of First-order Methods for Composite Convex Optimization, Adrien B. Taylor, Julien M. Hendrickx, François Glineur, SIAM Journal on Optimization, 27(3), 1283–1313 (2017)

# Thank you again for your attention!

References (software toolboxes):

PEPit: computer-assisted worst-case analyses of first-order optimization methods in Python, Baptiste Goujaud, Céline Moucer, François Glineur, Julien M. Hendrickx, Adrien B. Taylor and Aymeric Dieuleveut, preprint arXiv:2201.04040 2022

Performance estimation toolbox (PESTO): Automated worst-case analysis of first-order optimization methods., Adrien B. Taylor, Julien M. Hendrickx, François Glineur, 2017 IEEE 56th Annual Conference on Decision and Control (CDC) (pp. 1278-1283)

https://github.com/PerformanceEstimation

# Thank you again for your attention!

References (nonconvex/hypoconvex cases):

Convex Interpolation and Performance Estimation of First-order Methods for Convex Optimization, Adrien B. Taylor, PhD thesis, Université catholique de Louvain, 2017

The exact worst-case convergence rate of the gradient method with fixed step lengths for $L$-smooth functions, Hadi Abbaszadehpeivasti, Etienne de Klerk and Moslem Zamani, Optimization Letters 16, no. 6 (2022), pp. 1649-1661.

Tight convergence rates of the gradient method on smooth hypoconvex functions, Teodor Rotaru, François Glineur, Panagiotis Patrinos, preprint arxiv 2203.00775

# Thank you again for your attention!

Reference (linear mappings):

Performance Estimation of First-Order Methods involving Linear Mappings, Nizar Bousselmi, Julien M. Hendrickx, François Glineur, to appear soon on arxiv.

References (method design, nonlinear stepsizes):

Branch-and-Bound Performance Estimation Programming: A Unified Methodology for Constructing Optimal Optimization Methods, Shuvomoy Das Gupta, Bart P.G. Van Parys, Ernest K. Ryu, preprint arxiv 2203.07305

Nonlinear conjugate gradient methods: worst-case convergence rates via computer-assisted analyses, Shuvomoy Das Gupta, Robert M. Freund, Xu Andy Sun, Adrien Taylor, preprint arxiv 2301.01530