Interpolation Constraints for Computing Worst-Case Bounds in Performance Estimation Problems

Anne Rubbens, Nizar Bousselmi, Sébastien Colla, and Julien M. Hendrickx

CDC - 2023

Typical Optimization Analysis Problem

For every function f in class $\mathcal{F}$

e.g. Convex L-Smooth

If I apply the algorithm (…)

e.g. Gradient Descent

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

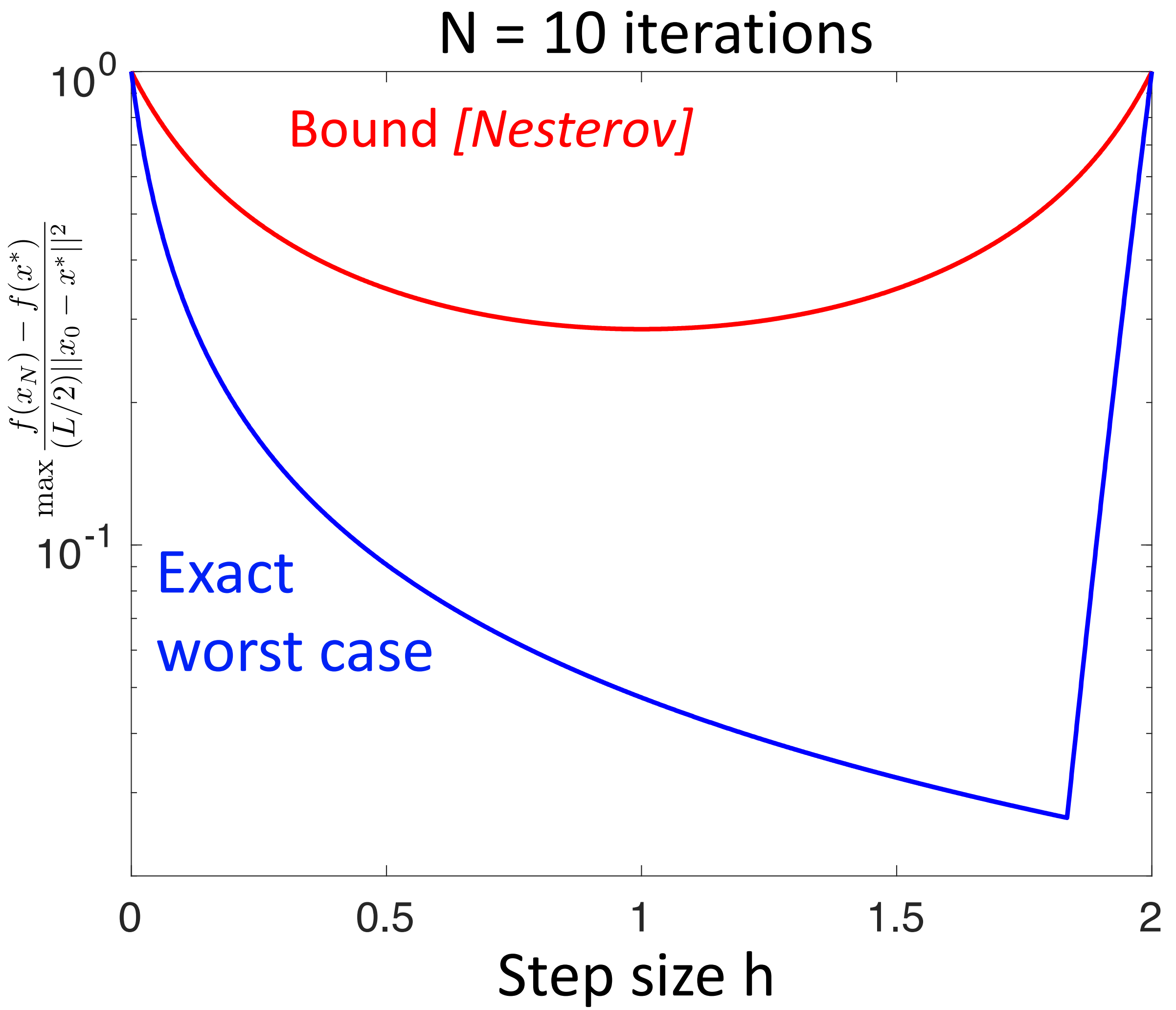
After N steps, some guarantee/bound holds

e.g.

$$f(x_N) - f(x^*) \leq \frac{(\ldots)}{N} ||x_0 - x^*||_2^2$$

Quality of guarantee = proxy for theoretical quality of algorithm

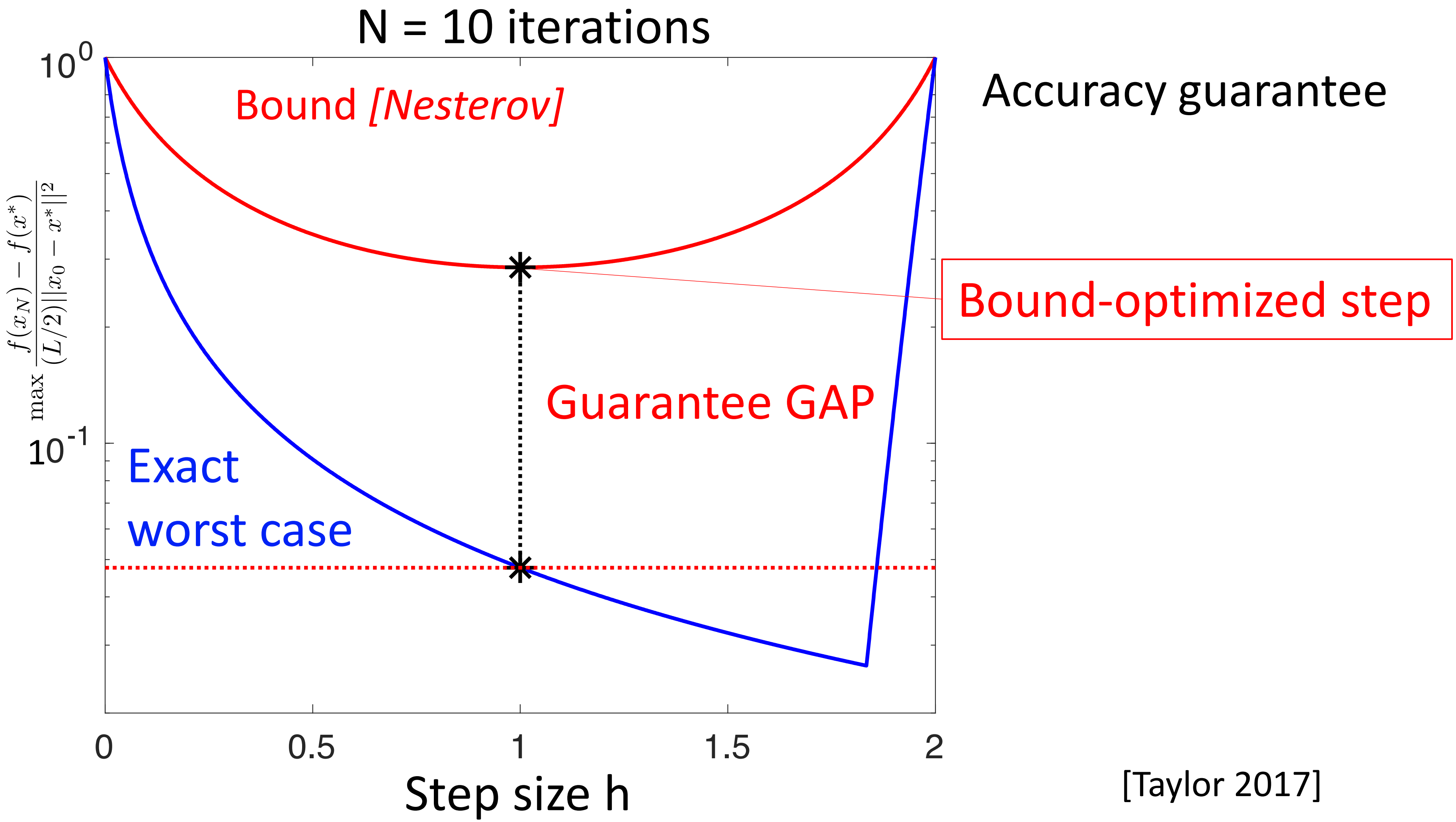
Importance of “good” bound: Gradient Descent



Accuracy guarantee

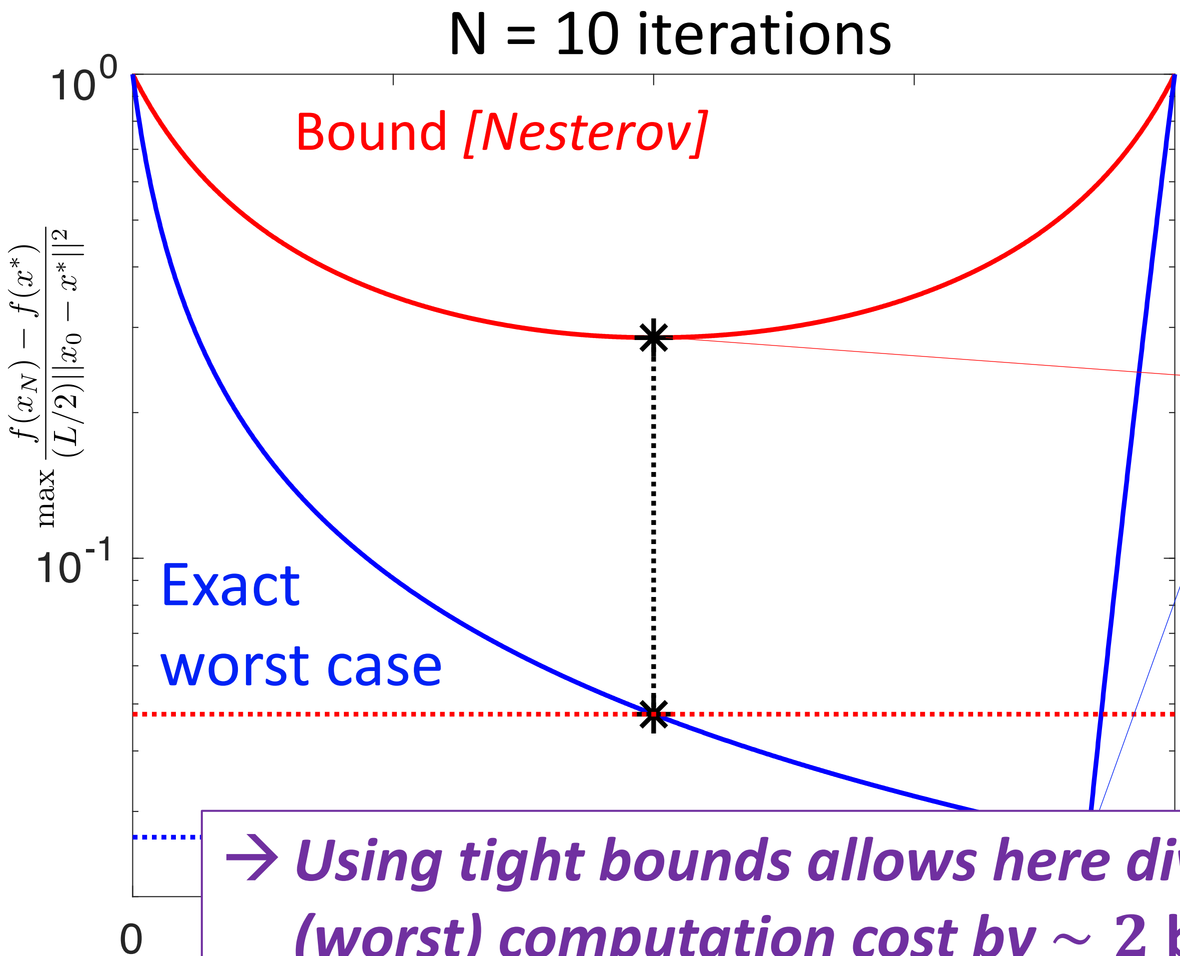
[Taylor 2017]

Importance of “good” bound: Gradient Descent



[Taylor 2017]

Importance of “good” bound: Gradient Descent



Accuracy guarantee

Bound-optimized step

Exact worst-case based Optimal step

Performance gap ~ 2

→ ***Using tight bounds allows here dividing actual (worst) computation cost by*** $\sim$ **2 by optimal tuning**

+ even more important for algorithm design

Abstract View of Optimization Analysis

For every function f in class $\mathcal{F}$ e.g. Convex L-Smooth

If I apply the algorithm (…) e.g. Gradient Descent

« Objects » invovled

$x_0, \nabla f(x_0), f(x_0)$
$x_1, \nabla f(x_1), f(x_1)$
(…)
$x_N, \nabla f(x_N), f(x_N)$
$x^*, \nabla f(x^*), f(x^*)$

Relations between objects

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

Conceptual assumption on function

e.g. f is convex L-smooth
Or $0 \preccurlyeq \nabla^2 f \preccurlyeq LI$

Goal: bound, e.g. relation between $f(x_N), f(x^*), ||x_0 - x^*||^2$

Not directly actionable !

- No ***direct*** involvement of the « objects »
- Concepts to which no direct access (2nd derivative etc.)
  (Even notion that $\nabla f(x_i)$ is the gradient of f)

➔ Need translation into algebraic relations between « objects » at stake

« Objects » invovled

$x_0, \nabla f(x_0), f(x_0)$
$x_1, \nabla f(x_1), f(x_1)$
(…)
$x_N, \nabla f(x_N), f(x_N)$
$x^*, \nabla f(x^*), f(x^*)$

Relations between objects

Algorithm

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

Conceptual assumption on function

e.g. f is convex L-smooth
Or $0 \preccurlyeq \nabla^2 f \preccurlyeq LI$

Goal: bound, e.g. relation between $f(x_N), f(x^*), ||x_0 - x^*||^2$

Issue prominent in optimization, but present for many other questions

Abstract structure of analysis

Every* analysis can be separated in two steps
(not always explicitly distinguished)

1) Translation

Conceptual assumptions

↓

Algebraic relations
between objects invovled
$(x_k, \nabla f(x_k), f(x_k))$

(Sometimes intermediate algebraic relations with other objects)

2) Algebraic Combination of

- Algebraic translations assumptions
- Algorithm description
- …

➔ ***desired bound***

- Mathematical skills
- Automated (optimized) techniques
  PEP, IQC, potential methods…

If translation conservative, then a priori conservative bound
<u>no matter how good the combination is</u>

Examples of « translations »

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{1}{2L} ||\nabla f(x) - \nabla f(y)||^2 \qquad \forall \boldsymbol{x}, \boldsymbol{y}$$

f L-Smooth Convex

$$\begin{cases} f(y) \geq f(x) + \nabla f(x)^T (y - x) \\ ||\nabla f(x) - \nabla f(y)||^2 \leq L^2 ||x - y||^2 \end{cases} \qquad \forall \boldsymbol{x}, \boldsymbol{y}$$

$$\begin{cases} f(y) \geq f(x) + \nabla f(x)^T (y - x) \\ f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} ||x - y||^2 \end{cases} \qquad \forall \boldsymbol{x}, \boldsymbol{y}$$

No Algebraic Equivalence

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

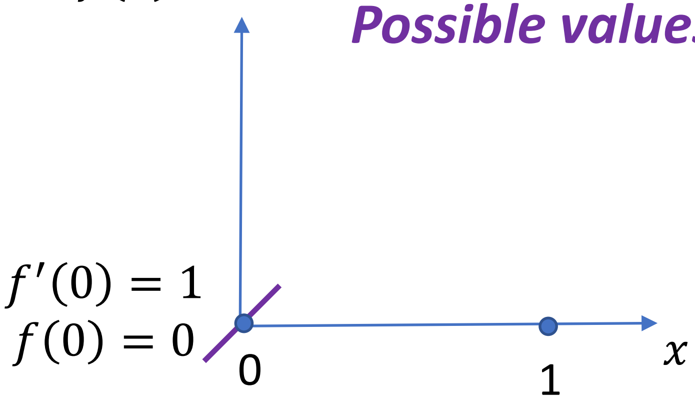
Algebraic expressions not equivalent!

« Translations » only equivalent if applied to all (x,y), not to finite set of values

$$\begin{cases} f(y) \geq f(x) + \nabla f(x)^T (y - x) \\ ||\nabla f(x) - \nabla f(y)||^2 \leq L^2 ||x - y||^2 \end{cases}$$

$$\begin{cases} f(y) \geq f(x) + \nabla f(x)^T (y - x) \\ f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} ||x - y||^2 \end{cases}$$

Some Translations « Stronger »



***f* is 1-smooth**

Possible values for $f(1), f'(1)$?

<u>*Translation 1:*</u>

$$f(x_j) \geq f(x_i) + \nabla f(x_i)^T (x_j - x_i)$$
$$||\nabla f(x_i) - \nabla f(x_j)||^2 \leq ||x_i - x_j||^2$$

→ $f(1) \leq 2$

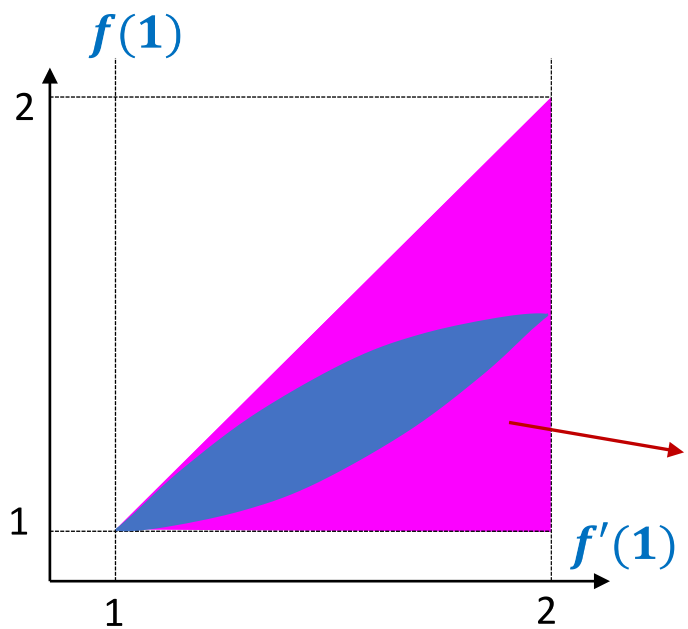
→ $f(1) - f'(1) \geq -1$

<u>*Translation 2:*</u>

$$f(x_j) \geq f(x_i) + \nabla f(x_i)^T (x_j - x_i) + \frac{1}{2}||\nabla f(x_j) - \nabla f(x_i)||^2$$

→ $f(1) \leq$ **1.5**

→ $f(1) - f'(1) \geq -0.5$



Bounds obtained by translation 1 valid for magenta zone, corresponding ***to no valid function***

→ Conservative (unless very lucky)

Impact on Optimization Bounds

Gradient descent $x_{k+1} = x_k - \frac{h}{L}\nabla f(x_k)$, 10 steps,

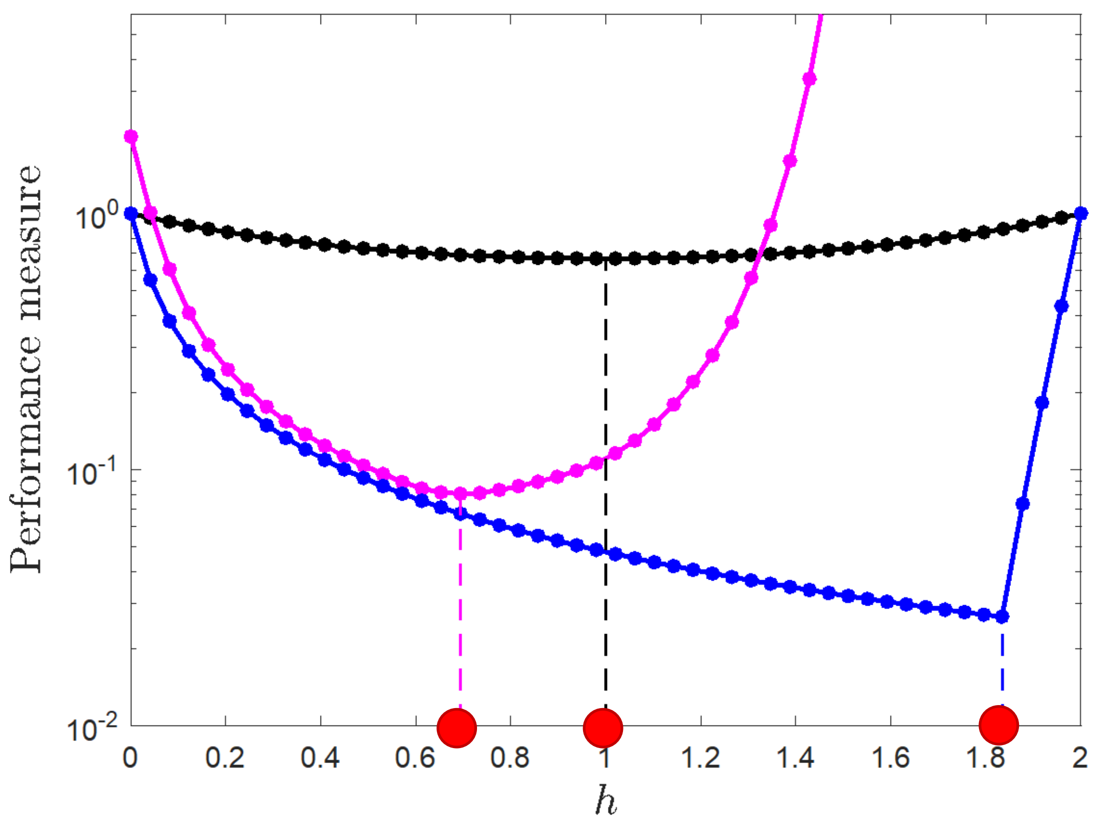
L-smooth convex functions

Bound from [Introductory Lectures on Convex Optimization Nesterov, 98]

Best bound with Translation 1

Best bound with Translation 2

Lead to ***very different optimized step-sizes***



Strongest Translation: Interpolation

« Definition »: An *interpolation contraint* for a class $\mathcal{F}$ is an algebraic relation on (finite) sequences (x_i, g_i, f_i) s.t.

P satisfied $\Leftrightarrow$ $\exists f \in \mathcal{F}$ s.t. $\begin{cases} f(x_i) = f_i \\ \nabla f(x_i) = g_i \end{cases}$

(generalization to higher order etc.)

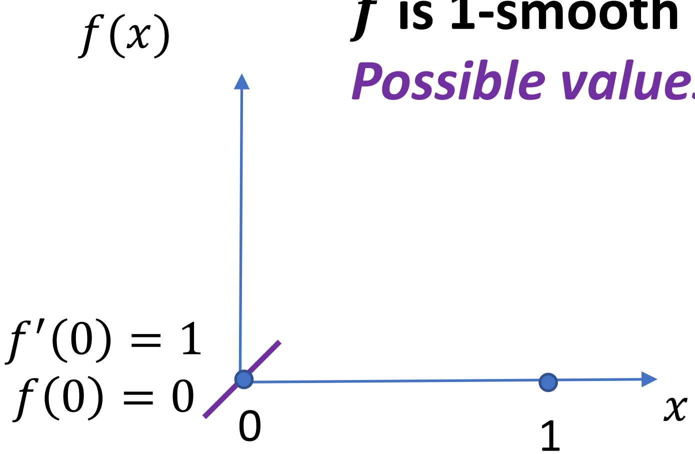
Example Interpolation constraint for L-smooth convex functions

$$f_j \geq f_i + g_i^T(x_j - x_i) + \frac{L}{2}||g_i - g_j||^2 \qquad \forall, i, j$$

Theorem: A claim involving the function class $\mathcal{F}$ and points x_i is ***true if and only if*** *it is implied by interpolation constraint* on the (x_i, g_i, f_i)

➔ *Lossless description*

Some Translations « Stronger »

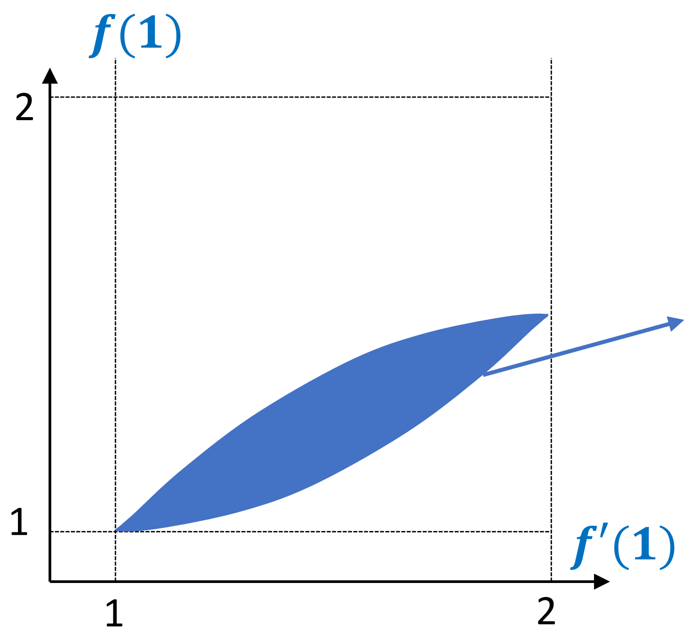


f is 1-smooth

Possible values for $f(1), f'(1)$?

*Translation 2 is an **interpolation constraint***

$$f(x_j) \geq f(x_i) + \nabla f(x_i)^T (x_j - x_i) + \frac{1}{2} ||\nabla f(x_j) - \nabla f(x_i)||^2$$



If $f(0) = 0, f'(0) = 1$

- Every convex L-smooth function leads to a blue point
- Every blue point corresponds to at least one convex L-smooth function

→ ***Tight description (and tight bounds)***

Example: necessity and sufficiency of Interpolation constraint

Any claim involving
- L-smooth convex functions
- Two points x_1, x_2 and a minimum x^*

Can be ***proved or disproved*** based on

+ other elements of the claim

$$f_2 \geq f_1 + g_1^T(x_2 - x_1) + \frac{L}{2}||g_1 - g_2||^2$$

$$f_1 \geq f_2 + g_2^T(x_1 - x_2) + \frac{L}{2}||g_2 - g_1||^2$$

$$f^* \geq f_1 + g_1^T(x^* - x_1) + \frac{L}{2}||g_1||^2$$

$$f^* \geq f_2 + g_2^T(x^* - x_2) + \frac{L}{2}||g_2||^2$$

$$f_2 \geq f^* + \frac{L}{2}||g_2||^2$$

$$f_1 \geq f^* + \frac{L}{2}||g_1||^2$$

Exploiting $\nabla f(x^*) = 0$

➔ Never any need to use other relations
And will allow ***tight results***

Tight Bound on gradient descent

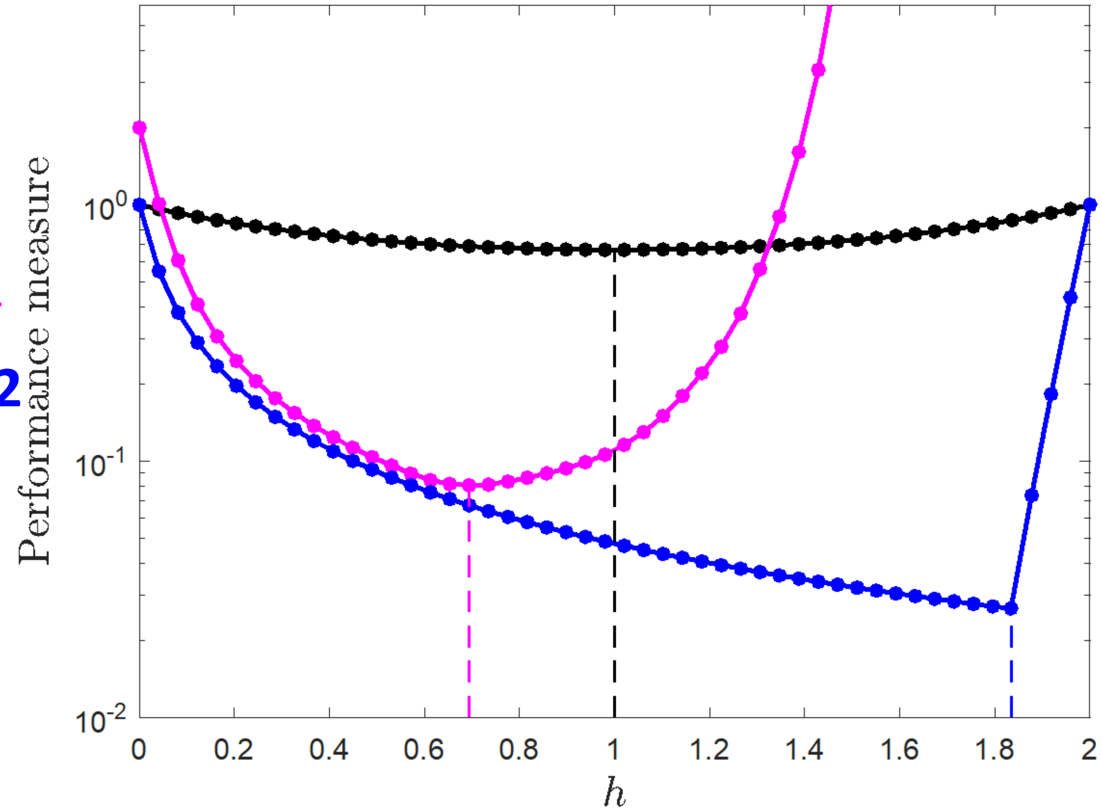
Gradient descent $x_{k+1} = x_k - \frac{h}{L}\nabla f(x_k)$, 10 steps,

L-smooth convex functions

Bound from [Introductory Lectures on Convex Optimization Nesterov, 98]

Best bound with Translation 1

Best bound with Translation 2
→ Tight worst-case bound



Interpolation Constraints and SDP

Interpolation constraint for convex L-smooth functions

$$f_j \geq f_i + g_i^T(x_j - x_i) + \frac{L}{2}||g_i - g_j||^2 \qquad \forall, i, j$$

<u>Nonlinear</u>, but ***<u>linear in</u>*** function values and scalar products of vectors

$$f_j \geq f_i + (g_i^T x_j) - (g_i^T x_i) + \frac{L}{2}(g_i^T g_i) - L(g_i^T g_j) + \frac{L}{2}(g_j^T g_j)$$

Same for many important classes of functions

→ ***SDP re-formulation*** *for analysis****,*** *if* ***dimension*** x_i, g_i ***left free***

Classes with Interpolation Constraints

- Lipschitz
- Smooth (Strongly) Convex – (strongly) convex
- Smooth
- Indicators
- Bounded domains
- Bounded subdifferential
- Weakly convex bounded « gradient »
- Certain classes of operators
- (…)

And allows representing implicit operations,
e.g. proximal operator defined by $\nabla f(x) + x - v = 0$

\+ ***Toolboxes*** to optimally exploit them

- PESTO (CDC 17), Matlab
- PEPiT (submitted), Python

Linear Operators in Many Problems

<u>*Ex:*</u> $\min_x f(x) + h(Mx)$ $\min_x f(x) \text{ s.t. } \mathrm{Mx} = \mathrm{b}$

→ <u>*Algorithms with linear operators*</u>:

- ***Chambolle-Pock***
- ADMM
- Primal-Dual fixed points
- …

$$\begin{cases} x_{k+1} = prox_{\tau f(.)}(x_i - \tau \boldsymbol{M^T u_i}) \\ u_{i+1} = prox_{\sigma g(.)}(u_i + \sigma \boldsymbol{M(2x_{i+1} - x_i)}) \end{cases}$$

Assumptions on classes of functions $\mathcal{F}, \mathcal{H}$ ***<u>and on operators</u> :***

- Bounded norm/ singular value $||M|| \leq L$
- Symmetric and eigenvalues $\lambda \in [\mu, L]$
- (…)

Representation of Linear Operators

Direct explicit representation (e.g. $\mu I \preccurlyeq M \preccurlyeq LI$) ***not necessarily convenient***

- Product and power of variables : $y = Mx$, or $y' = M^2 x$ …
- No direct interface with SDP formulations for functions
- Dimension dependent

➔ conditions on y_i, x_i, such that $\mathrm{y_i} = \mathrm{Mx_i}$ for M in desired class

<u>Simple Attempt:</u>

If $\begin{cases} y_i = Mx_i \\ ||M||_2 \leq L \end{cases}$ Then $||y_i||_2^2 \leq L^2 ||x_i||_2^2$

But, if applied to multiple couples (x_i, y_i) ***no guarantee of a common*** $\boldsymbol{M}$ s.t.

- $y_1 = Mx_1$
- $y_2 = Mx_2$
- (…)

Interpolation for Linear Operators

Theorem *[Bousselmi, J.H. Glineur 23]:* Given vectors x_i, y_i, u_j, v_j

There exists M s.t.

$||M|| \leq L$

$\boldsymbol{y_i = M x_i} \quad \forall i$

$\boldsymbol{v_j = M^T u_j} \quad \forall j$

If and only if

$$\begin{cases} X^T V = Y^T U \\ Y^T Y \preccurlyeq L^2 X^T X \\ V^T V \preccurlyeq L^2 U^T U \end{cases}$$

With

$X = (x_1 | x_2 | \dots | x_n)$ $\quad U = (u_1 | u_2 | \dots | u_m)$

$Y = (y_1 | y_2 | \dots | y_n)$ $\quad V = (v_1 | v_2 | \dots | v_m)$

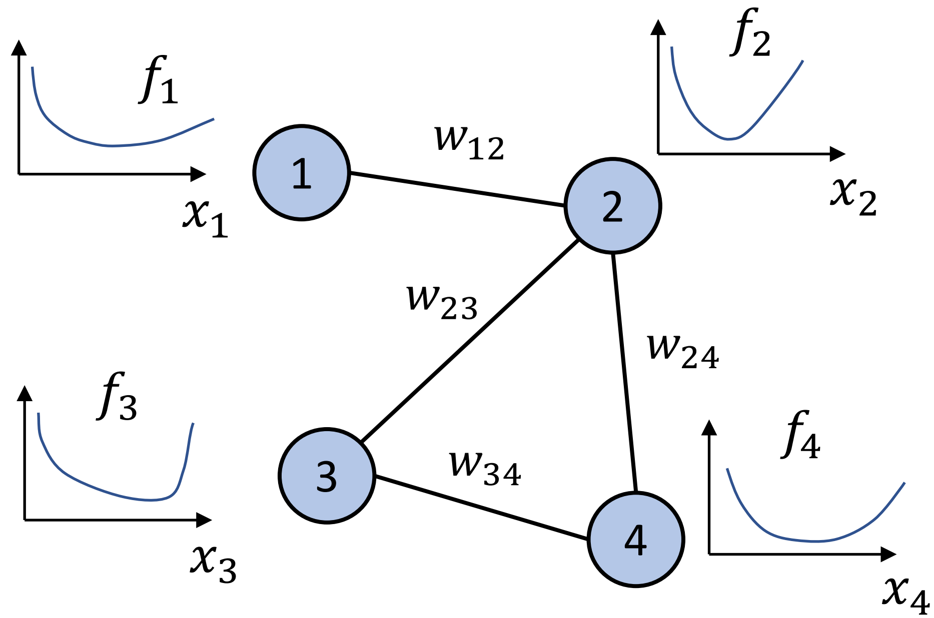
+ similar result for symmetric matrices with eigenvalue in a given range

→ Easy interface with interpolation condition on functions

→ Automated analysis of algorithms involving operators

Decentralized Optimization

$$\min_{x_1,\dots,x_N} \sum_i f_i(x_i)$$
$$\text{s.t. } x_i = x_j \quad \forall\,(i,j) \text{ neighbors}$$



Decentralization

- Local function: f_i
- Local copy of x: x_i

Example : Decentralized Gradient Descent (DGD)

For each iteration k

$$y_i^k = \sum_j w_{ij}\, x_j^k \qquad \textit{Consensus / Averaging step}$$

$$x_i^{k+1} = y_i^k - \alpha \nabla f_i(x_i^k) \qquad \textit{Local gradient step}$$

Present in many decentralized algorithm

Interpolating Consensus Steps

$$y_i^k = \sum_j w_{ij} x_i^k$$

$$\Leftrightarrow Y^k = (W \otimes I) X^k$$

x_i^k, y_i^k value of agent i at time k

W <u>*symmetric*</u> stochastic ($\sum w_{ij} = 1 \; w_{ij} \geq 0$)
Non-principal eigenvalues in $[-\lambda, \lambda]$

No interpolation conditions known for $(W \otimes I)$. But interpolation for

$$Y^k = \left(\left(\frac{1}{N} 11^T \otimes I \right) + \widetilde{W} \right) X^k$$

- $\widetilde{W}$ symmetric, $\widetilde{W}\mathbf{1} = 0$
- Eigenvalues of $\widetilde{W}$ in $[-\lambda, \lambda]$

[Colla, Hendrickx 24]

Average preserving

No Kronecker structure
→ Mixes different agent coordinates
in Space orthogonal to consensus

Interpolating Consensus Steps

$$y_i^k = \sum_j w_{ij} x_i^k$$

$$\Leftrightarrow Y^k = (W \otimes I) X^k$$

x_i^k, y_i^k value of agent i at time k

W *symmetric* stochastic ($\sum w_{ij} = 1 \; w_{ij} \geq 0$)
Non-principal eigenvalues in $[-\lambda, \lambda]$

No interpolation conditions known for $(W \otimes I)$. But interpolation for

$$Y^k = \left(\left(\frac{1}{N} 11^T \otimes I \right) + \widetilde{W} \right) X^k$$

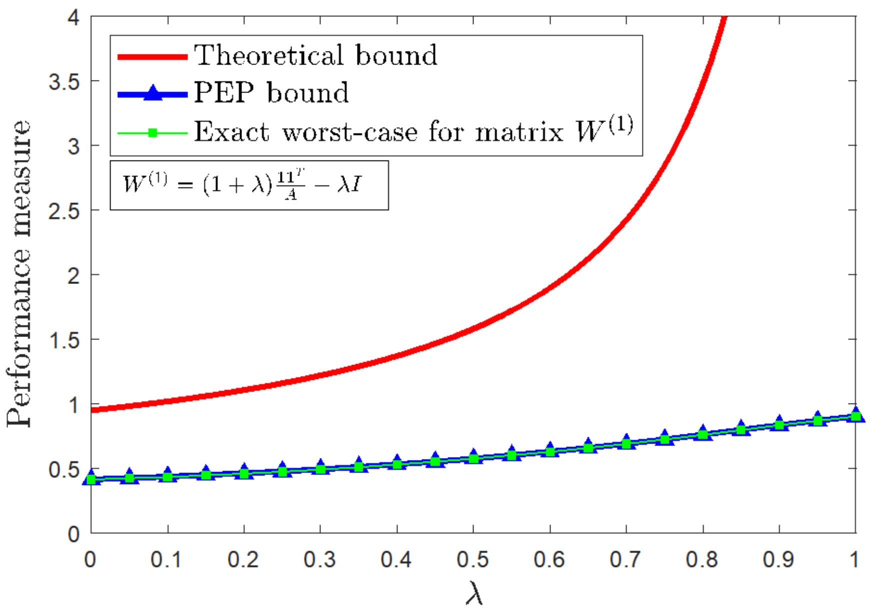
- $\widetilde{W}$ symmetric, $\widetilde{W}\mathbf{1} = 0$
- Eigenvalues in $[-\lambda, \lambda]$

[Colla, Hendrickx 24]

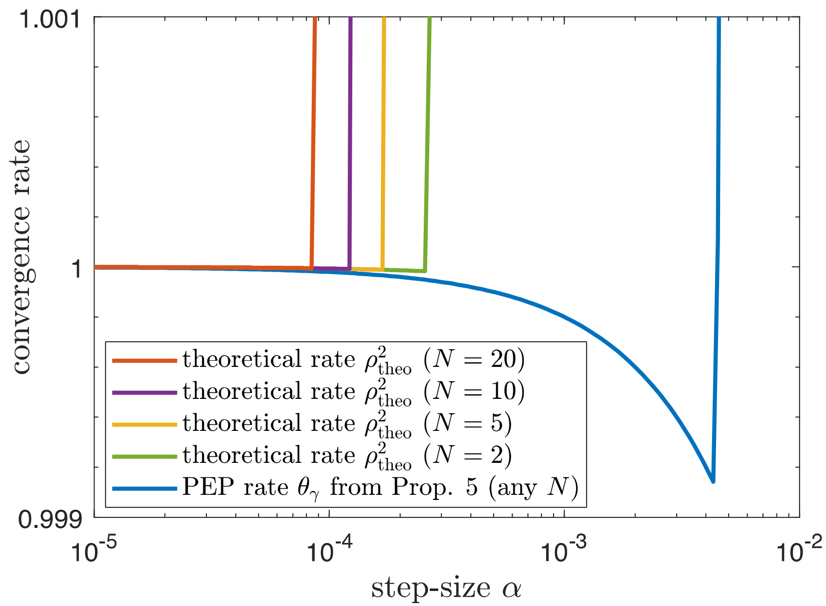
Relaxed Assumptions:

- Non-negative w_{ij} ***Assumed but never actually used in proofs***
- Agent values multiplied by ***scalar*** w_{ij} ***Unclear if actually used in proofs***
  → ***homogenous*** and ***independent*** treatment of coordinates

Examples of applications



Decentralized Gradient Descent



DIGing *[Nedic, Olshevsky, Shi, 2016]*

$\lambda = .9, \mu = .1$

Conclusions on Interpolation Constraints

- Lossless finite algebraic translation of conceptual assumptions
- When available, use them (and nothing else)
- Toolboxes for easy use (PESTO, PEPit)

Perspectives

- Systematic way of finding them (work in progress)
- Algebra of interpolation constraint
- Degree of conservatism of non-interpolation constraints?
- Multiple classes, second order etc.

Question: should we use classes for which there is no tractable algebraic constraints?

⇔ No tractable tight algebraic description