An introduction to semidefinite programming performance estimation for the gradient method

Etienne de Klerk

July, 2026

Tilburg University

The gradient method of Cauchy

The gradient descent method of Cauchy

Input: $f : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{x}_0 \in \mathbb{R}^n$, number of steps N.

for $k = 0, \ldots, N-1$

Select step length $t_k \in \mathbb{R}$

$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k)$

Augustin-Louis Cauchy (1789–1857)

(Studied the gradient descent method in 1847.)

History

Cauchy described his method in the short note:

A. Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. C. R. Acad. Sci. Paris, 25:536–538, 1847.

It was motivated by a problem in astronomy: to compute an orbit of a heavenly body as a nonlinear least squares problem:

> *'[to solve] the [algebraic] equations representing the motion of this body, taking as unknowns the elements of the orbit themselves. Then there are six such unknowns.'*

History: convergence analysis

Cauchy did not do any careful convergence analysis:

> *'If the new value of [the residual] is not a minimum, one can deduce, again proceeding in the same way, a [new] value still smaller; and, so continuing, smaller and smaller values will be found, which will converge to a minimal value. [The minimum residual] will always be obtained by the above method, provided that the [starting point is] suitably chosen.'*

To obtain more precise convergence results, one needs to restrict the class of objective functions ...

Some classes of objective functions

L-smooth convex functions

- An L-smooth function is differentiable and

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n,$$

  for some (known) Lipschitz constant $L > 0$.
- A differential function is convex if and only if

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

- Equivalently, $\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq 0 \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Exercise:

Show that a convex, L-smooth function f satisfies

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{1}{2L} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|^2 \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

Functions of bounded curvature

- The function f has a maximum curvature $0 \leq L < \infty$ if

$$\mathbf{x} \mapsto \tfrac{L}{2}\|\mathbf{x}\|^2 - f(\mathbf{x}) \text{ is convex ...}$$

- ... and minimum curvature $\mu \in \mathbb{R}$ if

$$\mathbf{x} \mapsto f(\mathbf{x}) - \tfrac{\mu}{2}\|\mathbf{x}\|^2 \text{ is convex.}$$

If $\mu > 0$ we say f is μ-strongly convex.

Notation:

If f has a maximum curvature L and minimum curvature $\mu \leq L$, we write $f \in \mathcal{F}_{\mu,L}$.

Functions of bounded curvature: exercises

Exercises (calculus)

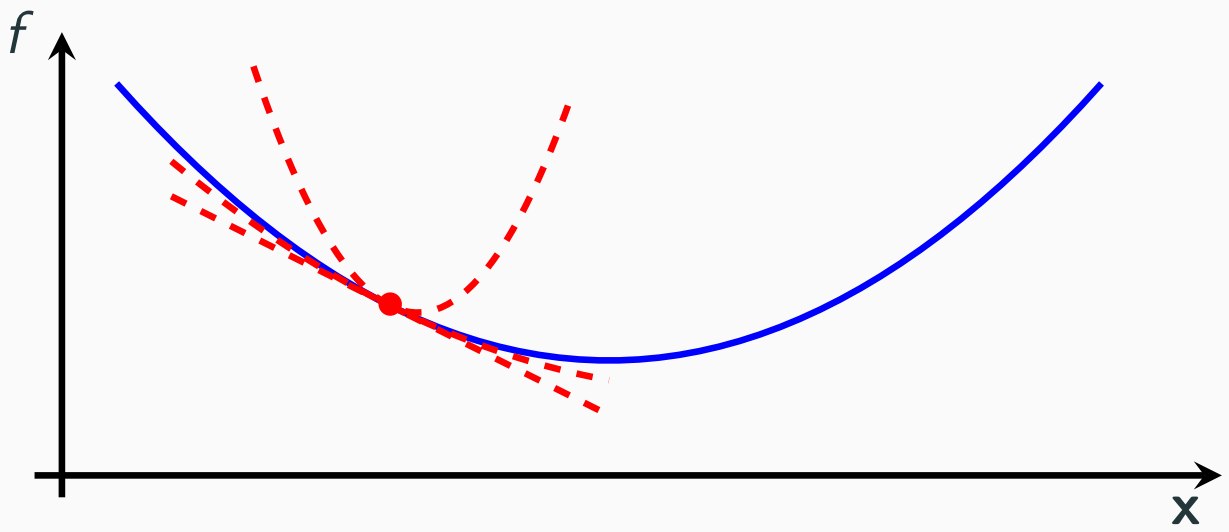
Assume $f \in \mathcal{F}_{\mu,L}$ with $\mu \leq L$. Show that

1. If $\mu = 0$ and $L > 0$, then f is convex and L-smooth.
2. If $L > 0$ and $\mu = -L$, then f is L-smooth.
3. If f is twice continuously differentiable, the spectrum of $\nabla^2 f(\mathbf{x})$ is contained in the interval $[\mu, L]$ for all $\mathbf{x} \in \mathbb{R}^n$.

Background reading:

Yu. Nesterov. *Lectures on convex optimization*, 2nd ed. Springer Optimization and Its Applications 137, Springer Nature, 2018.

Summary: function class $\mathcal{F}_{\mu,L}$ $(0 \le \mu \le L < \infty)$



(1) (Convexity) $f(\mathbf{x}) \ge f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$

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

(2) (L-smoothness) $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|$

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

Smooth, strongly convex functions: $\mathcal{F}_{\mu,L}$ with $0 \leq \mu < L$.

Exercise:

One has $f \in \mathcal{F}_{\mu,L}$ with $0 \leq \mu < L$, if and only if the following holds for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$:

$$\begin{aligned} f(\mathbf{x}) \quad \geq \quad & f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \\ & + \frac{1}{2(1-\mu/L)} \left(\frac{1}{L} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|^2 + \mu \|\mathbf{x} - \mathbf{y}\|^2 \right. \\ & \left. - 2\frac{\mu}{L} \langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \right). \end{aligned}$$

- When $\mu = 0$, we recover the result in the exercise on Slide 4.

Back to performance analysis

Worst-case performance estimation

Basic idea:

Find the worst input $f \in \mathcal{F}_{\mu,L}$ and $\mathbf{x}_0$ for the gradient descent algorithm, for given N, $0 < \mu \leq L$, and step sizes t_k $(k = 0, 1, \ldots, N-1)$.

Main tool: semidefinite programming (SDP) performance estimation, introduced in:

Y. Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

We have questions

- **Q:** Worst in which sense?

  A: Distance to optimality after N steps seems logical.
- **Q:** But, if $\mathbf{x}_\star$ is optimal, which one should we choose:

$$f(\mathbf{x}_N) - f(\mathbf{x}_\star), \ \|\mathbf{x}_N - \mathbf{x}_\star\|, \text{ or } \|\nabla f(\mathbf{x}_N)\|?$$

  A: Let's start with the function values.
- **Q:** Which step lengths do we use in gradient descent?

  A: Let's start with exact linesearch.
- We start by analysing one step ($N = 1$) to keep things simple.

Gradient descent with exact linesearch

Gradient descent with exact line search

Input: $f : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{x}_0 \in \mathbb{R}^n$.

for $k = 0, 1, \ldots$

$$t_k = \operatorname{argmin}_{t \in \mathbb{R}} f\left(\mathbf{x}_k - t \nabla f(\mathbf{x}_k)\right)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k)$$

Exercise

Show that, due to exact linesearch, we have

$$\langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle = 0 \quad k = 0, 1, \ldots$$

and

$$\langle \nabla f(\mathbf{x}_{k+1}), \nabla f(\mathbf{x}_k) \rangle = 0 \quad k = 0, 1, \ldots$$

Worst-case performance as an optimization problem

Performance estimation problem for first gradient step $\mathbf{x}_0 \to \mathbf{x}_1$:

$$\max\ f(\mathbf{x}_1) - f(\mathbf{x}_\star)$$

s.t.	$f \in \mathcal{F}_{\mu,L}$	Function class
	$\mathbf{x}_\star$ optimal for f	Optimal point
	$t_0 = \operatorname{argmin}_{t\in\mathbb{R}} f\left(\mathbf{x}_0 - t\nabla f(\mathbf{x}_0)\right)$	
	$\mathbf{x}_1 = \mathbf{x}_0 - t_0\nabla f(\mathbf{x}_0)$	Algorithm
	$f(\mathbf{x}_0) - f(\mathbf{x}_\star) \leq R$	Initial distance

Variables:

$f, \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_\star, \nabla f(\mathbf{x}_0)$.

Fixed parameters

$0 < \mu \leq L$, $R > 0$.

How do we deal with the constraint $f \in \mathcal{F}_{\mu,L}$?

We first introduce variables, indexed by $k \in S := \{0, 1, \star\}$:

- $\mathbf{g}_k$ corresponding to $\nabla f(\mathbf{x}_k)$
- f_k corresponding to $f(\mathbf{x}_k)$.

Q: When do given values $\mathbf{x}_k$, $\mathbf{g}_k \in \mathbb{R}^n$ and $f_k \in \mathbb{R}$ $(k \in S)$ satisfy

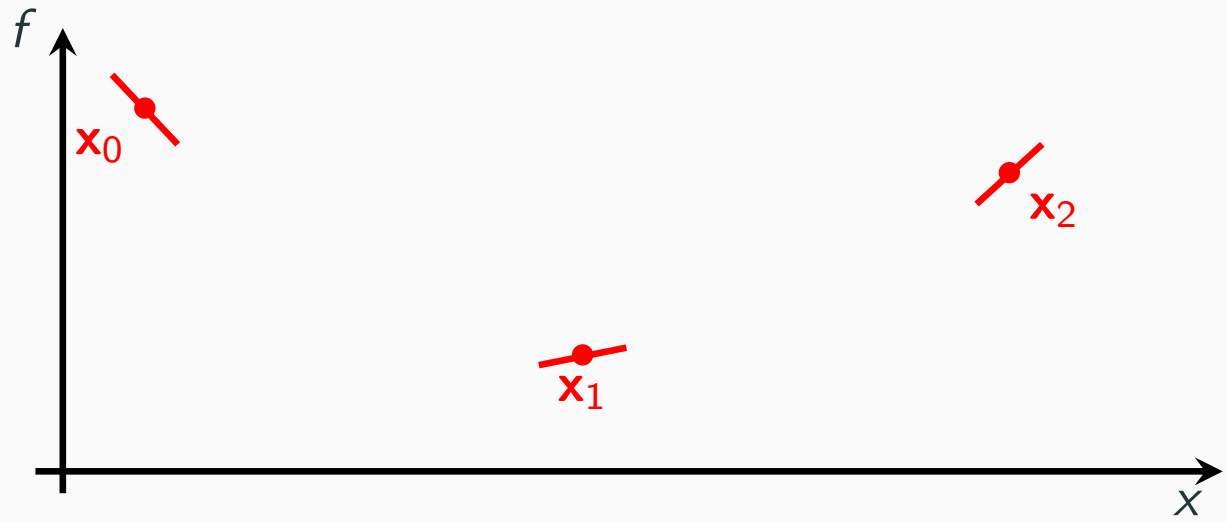
$$\mathbf{g}_k = \nabla f(\mathbf{x}_k) \text{ and } f_k = f(\mathbf{x}_k)\ \forall k \in S \text{ for some } f \in \mathcal{F}_{\mu,L}?$$

We call this the smooth strongly convex interpolation problem ...

Interpolation theorem

Smooth strongly convex interpolation problem

In general, consider an index set S, and given values $\{(\mathbf{x}_k, \mathbf{g}_k, f_k)\}_{i \in S}$ where $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{g}_k \in \mathbb{R}^n$ and $f_k \in \mathbb{R}$.



$$?\exists\ f \in \mathcal{F}_{\mu,L}\colon\ f(\mathbf{x}_k) = f_k, \quad \text{and} \quad \mathbf{g}_k = \nabla f(\mathbf{x}_k), \qquad \forall k \in S.$$

If yes, we say $\{(\mathbf{x}_i, \mathbf{g}_k, f_k)\}_{k \in S}$ is $\mathcal{F}_{\mu,L}$-interpolable.

Smooth Strongly Convex Interpolation Problem (ctd.)

Theorem (Taylor, Hendrickx, and Glineur (2015))

The following statements are equivalent:

1. $\{(\mathbf{x}_k, \mathbf{g}_k, f_k)\}_{k \in S}$ *is* $\mathcal{F}_{\mu,L}$*-interpolable;*
2. $\forall i, j \in S$*:*

$$\begin{aligned} & f_i - f_j - \langle \mathbf{g}_j, \mathbf{x}_i - \mathbf{x}_j \rangle \\ \geq \quad & \frac{1}{2(1-\mu/L)} \left(\frac{1}{L} \|\mathbf{g}_i - \mathbf{g}_j\|^2 + \mu \|\mathbf{x}_i - \mathbf{x}_j\|^2 - 2\frac{\mu}{L} \langle \mathbf{g}_j - \mathbf{g}_i, \mathbf{x}_j - \mathbf{x}_i \rangle \right). \end{aligned}$$

Reference

A.B. Taylor, J.M. Hendrickx, and F. Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161, 307–345 (2017).

Final formulation of worst-case performance problem

Worst-case problem reformulation

- Parameters: $L \geq \mu > 0$, $R > 0$;
- Variables: $\{(\mathbf{x}_k, \mathbf{g}_k, f_k)\}_{k \in S}$ $(S = \{\star, 0, 1\})$.

Performance estimation problem:

$$\max\ f_1 - f_\star$$

$$\begin{aligned}
\text{s.t. } & \{(\mathbf{x}_k, \mathbf{g}_k, f_k)\}_{k \in S} \ \ \mathcal{F}_{\mu,L}\text{-interpolable} && \text{Functional class}\\
& \mathbf{g}_\star = \mathbf{0} && \text{Optimal point}\\
& \langle \mathbf{x}_1 - \mathbf{x}_0, \mathbf{g}_1 \rangle = 0,\ \langle \mathbf{g}_0, \mathbf{g}_1 \rangle = 0 && \text{Algorithm (relaxation)}\\
& f_0 - f_\star \leq R && \text{Initial distance}
\end{aligned}$$

Explicit reformulation

$$\max f_1 - f_\star$$

subject to

$$\begin{aligned}
f_i &\geq f_j + \langle \mathbf{g}_j, \mathbf{x}_i - \mathbf{x}_j \rangle \\
&\quad + \frac{1}{2(1-\mu/L)} \left(\frac{1}{L} \|\mathbf{g}_i - \mathbf{g}_j\|^2 + \mu \|\mathbf{x}_i - \mathbf{x}_j\|^2 - 2\frac{\mu}{L} \langle \mathbf{g}_j - \mathbf{g}_i, \mathbf{x}_j - \mathbf{x}_i \rangle \right) \\
&\quad \forall i, j \in \{0, 1, \star\} \\
0 &= \langle \mathbf{g}_0, \mathbf{g}_1 \rangle \\
0 &= \langle \mathbf{g}_1, \mathbf{x}_1 - \mathbf{x}_0 \rangle \\
f_0 - f_\star &\leq R \\
\mathbf{g}_\star &= \mathbf{0}.
\end{aligned}$$

Explicit reformulation: remarks

- We may assume $\mathbf{x}_\star = \mathbf{0}$, $f_\star = 0$, $f_0 \geq 0$, and $f_1 \geq 0$ (why?).
- After eliminating the zero variables, all the constraints are linear in the variables f_0, f_1, and in the entries of the (positive semidefinite) Gram matrix:

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{12} & x_{22} & x_{23} & x_{24} \\ x_{13} & x_{23} & x_{33} & x_{34} \\ x_{14} & x_{24} & x_{34} & x_{44} \end{pmatrix} := \begin{pmatrix} \langle \mathbf{x}_0, \mathbf{x}_0 \rangle & \langle \mathbf{x}_0, \mathbf{x}_1 \rangle & \langle \mathbf{x}_0, \mathbf{g}_0 \rangle & \langle \mathbf{x}_0, \mathbf{g}_1 \rangle \\ \langle \mathbf{x}_0, \mathbf{x}_1 \rangle & \langle \mathbf{x}_1, \mathbf{x}_1 \rangle & \langle \mathbf{x}_1, \mathbf{g}_0 \rangle & \langle \mathbf{x}_1, \mathbf{g}_1 \rangle \\ \langle \mathbf{x}_0, \mathbf{g}_0 \rangle & \langle \mathbf{x}_1, \mathbf{g}_0 \rangle & \langle \mathbf{g}_0, \mathbf{g}_0 \rangle & \langle \mathbf{g}_0, \mathbf{g}_1 \rangle \\ \langle \mathbf{x}_0, \mathbf{g}_1 \rangle & \langle \mathbf{x}_1, \mathbf{g}_1 \rangle & \langle \mathbf{g}_0, \mathbf{g}_1 \rangle & \langle \mathbf{g}_1, \mathbf{g}_1 \rangle \end{pmatrix}.$$

- Thus the reformulation becomes a semidefinite programming problem (SDP) with positive semidefinite matrix variable (x_{ij}).
- We write $(x_{ij}) \succeq 0$ to indicate the matrix is positive semidefinite.

SDP problems

For suitable data (c_{ij}), $\mathbf{c}$, $(a_{ij}^{(k)})$, $\mathbf{a}_k$, the performance estimation problem is of the form (where $[m] := \{1, \dots, m\}$):

$$\max_{(x_{ij}) \succeq 0, \mathbf{f} \geq \mathbf{0}} \left\{ \sum_{i,j=1}^{n'} c_{ij} x_{ij} + \langle \mathbf{c}, \mathbf{f} \rangle \;\middle|\; \sum_{i,j=1}^{n'} a_{ij}^{(k)} x_{ij} + \langle \mathbf{a}_k, \mathbf{f} \rangle \leq b_k \;\; k \in [m] \right\}.$$

Dual SDP problem

Defining $A_k = \left(a_{ij}^{(k)}\right)$ and $A^\top = [\mathbf{a}_1, \dots, \mathbf{a}_m]$:

$$\min_{\mathbf{y} \in \mathbb{R}^m, \mathbf{y} \geq \mathbf{0}} \left\{ \mathbf{b}^\top \mathbf{y} \;\middle|\; A^\top \mathbf{y} \geq \mathbf{c}, \; \sum_{k=1}^{m} y_k A_k - C \succeq 0 \right\}.$$

NB: y_k is the Lagrange multiplier of $\sum_{i,j=1}^{n} a_{ij}^{(k)} x_{ij} + \langle \mathbf{a}_k, \mathbf{f} \rangle \leq b_k$.

Weak duality

Exercise:

Show that, if $(x_{ij}) \succeq 0$ and $\mathbf{f} \geq \mathbf{0}$ are primal feasible, and $\mathbf{y} \geq \mathbf{0}$ dual feasible, then:

$$\begin{aligned} \mathbf{b}^\top \mathbf{y} &\geq \sum_{k=1}^{m} y_k \left(\sum_{i,j=1}^{n} a_{ij}^{(k)} x_{ij} + \langle \mathbf{a}_k, \mathbf{f} \rangle \right) \\ &\geq \sum_{i,j=1}^{n'} c_{ij} x_{ij} + \langle \mathbf{c}, \mathbf{f} \rangle. \end{aligned}$$

Thus an upper bound on the SDP optimal value is obtained by aggregating the constraints with the dual multipliers y_k.

Duality exercise

Exercise:

For the performance estimation SDP on slide 17.:

1. Write down the primal and dual SDP problems explicitly.
2. Show that both primal and dual problems satisfy the Slater condition so that strong duality holds.

The SDP performance estimation problem: results

The SDP bound

Theorem

The optimal value of the SDP

$$\begin{aligned} & \max \ f_1 - f_\star \\ \text{s.t.} \ & \{(\mathbf{x}_i, \mathbf{g}_i, f_i)\}_{i\in\{0,1,\star\}} \ \ \mathcal{F}_{\mu,L}\text{-interpolable} \\ & \mathbf{g}_\star = \mathbf{0} \\ & \langle \mathbf{x}_1 - \mathbf{x}_0, \mathbf{g}_1 \rangle = 0, \ \langle \mathbf{g}_0, \mathbf{g}_1 \rangle = 0 \\ & f_0 - f_\star \leq R \end{aligned}$$

is

$$f_1 - f_\star = \left(\frac{L-\mu}{L+\mu}\right)^2 (f_0 - f_\star).$$

Proof sketch

Aggregate the constraints

$$f_0 \geq f_1 + \langle \mathbf{g}_1, \mathbf{x}_0 - \mathbf{x}_1 \rangle + \tfrac{1}{2(1-\mu/L)} \left(\tfrac{1}{L} \|\mathbf{g}_0 - \mathbf{g}_1\|^2 + \mu \|\mathbf{x}_0 - \mathbf{x}_1\|^2 - 2\tfrac{\mu}{L} \langle \mathbf{g}_1 - \mathbf{g}_0, \mathbf{x}_1 - \mathbf{x}_0 \rangle \right)$$
$$f_\star \geq f_0 + \langle \mathbf{g}_0, \mathbf{x}_\star - \mathbf{x}_0 \rangle + \tfrac{1}{2(1-\mu/L)} \left(\tfrac{1}{L} \|\mathbf{g}_\star - \mathbf{g}_0\|^2 + \mu \|\mathbf{x}_\star - \mathbf{x}_0\|^2 - 2\tfrac{\mu}{L} \langle \mathbf{g}_0 - \mathbf{g}_\star, \mathbf{x}_0 - \mathbf{x}_\star \rangle \right)$$
$$f_\star \geq f_1 + \langle \mathbf{g}_1, \mathbf{x}_\star - \mathbf{x}_1 \rangle + \tfrac{1}{2(1-\mu/L)} \left(\tfrac{1}{L} \|\mathbf{g}_\star - \mathbf{g}_1\|^2 + \mu \|\mathbf{x}_\star - \mathbf{x}_1\|^2 - 2\tfrac{\mu}{L} \langle \mathbf{g}_1 - \mathbf{g}_\star, \mathbf{x}_1 - \mathbf{x}_\star \rangle \right)$$
$$0 = \langle \mathbf{g}_0, \mathbf{g}_1 \rangle$$
$$0 = \langle \mathbf{g}_1, \mathbf{x}_1 - \mathbf{x}_0 \rangle,$$

with multipliers:

$$y_1 = \frac{L-\mu}{L+\mu}, \quad y_2 = 2\mu \frac{(L-\mu)}{(L+\mu)^2}, \quad y_3 = \frac{2\mu}{L+\mu}, \quad y_4 = \frac{2}{L+\mu}, \quad y_5 = 1.$$

Proof sketch (ctd.)

Resulting inequality:

$$\begin{aligned} f_1 - f_\star \quad \leq \quad & \left(\tfrac{L-\mu}{L+\mu}\right)^2 (f_0 - f_\star) \\ & - \tfrac{\mu L(L+3\mu)}{2(L+\mu)^2} \left\| \mathbf{x}_0 - \tfrac{L+\mu}{L+3\mu}\mathbf{x}_1 - \tfrac{2\mu}{L+3\mu}\mathbf{x}_\star - \tfrac{3L+\mu}{L^2+3\mu L}\mathbf{g}_0 - \tfrac{L+\mu}{L^2+3\mu L}\mathbf{g}_1 \right\|^2 \\ & - \tfrac{2L\mu^2}{L^2+2L\mu-3\mu^2} \left\| \mathbf{x}_1 - \mathbf{x}_\star - \tfrac{(L-\mu)^2}{2\mu L(L+\mu)}\mathbf{g}_0 - \tfrac{L+\mu}{2\mu L}\mathbf{g}_1 \right\|^2. \end{aligned}$$

Last two terms nonpositive, so

$$f_1 - f_\star \leq \left(\frac{L-\mu}{L+\mu}\right)^2 (f_0 - f_\star).$$

Finding the correct multipliers

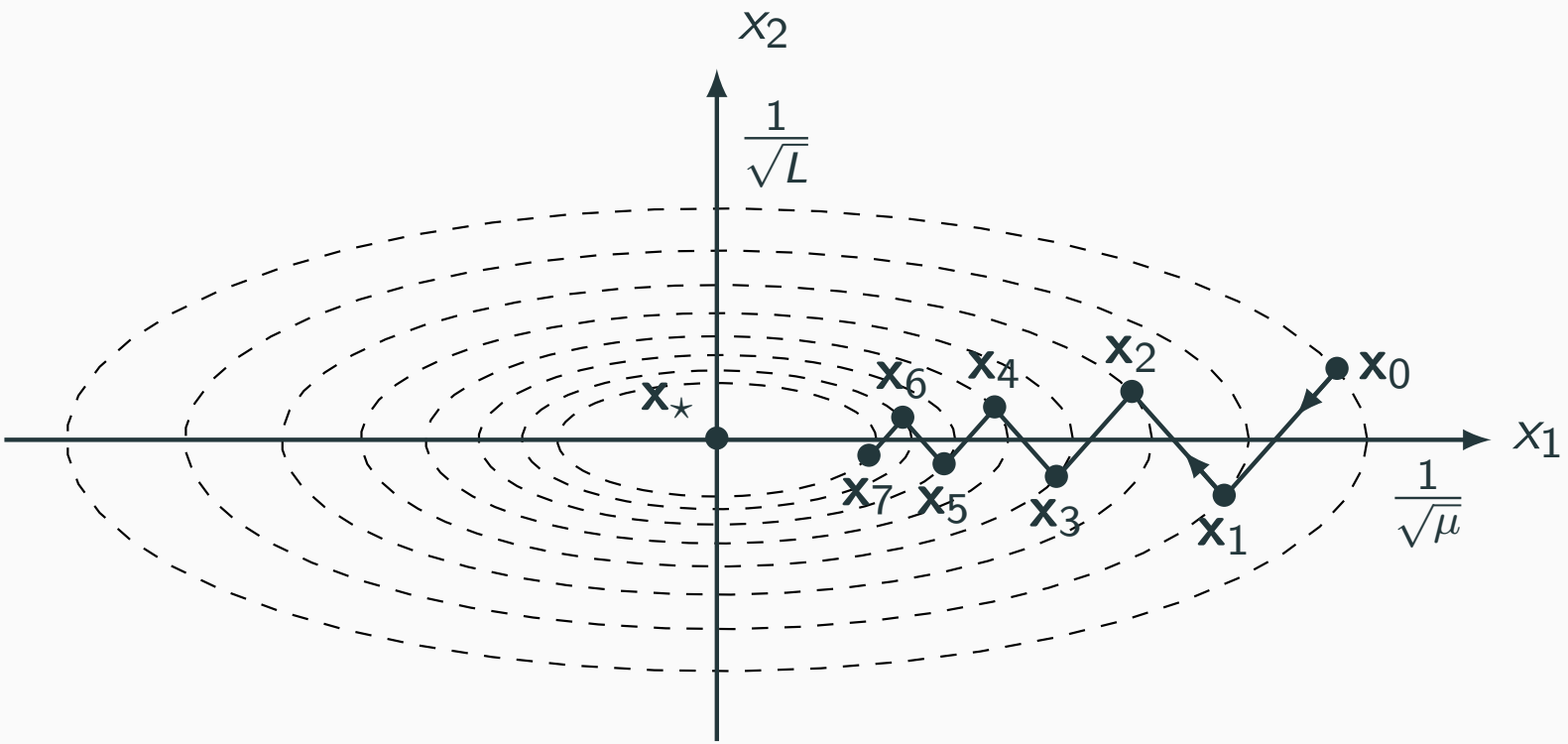
- The proof is easy to check, given the right multipliers $y_1, \ldots, y_5$.
- The multipliers were first guessed by solving the performance estimation SDP numerically for various choices of μ, L, R, ...
- .. where we may assume $L = R = 1$ (why?) ...
- ... and noting the optimal dual multipliers.
- Finally, the correctness of the inequality was verified through (symbolic) calculation.

Quadratic example

The SDP bound is tight, due to the following quadratic example.

$$f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n} \lambda_i x_i^2,$$

$0 < \mu = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n = L$, $\mathbf{x}_0 = (1/\mu, 0, \ldots, 0, 1/L)^\top$.



Quadratic example (ctd.)

Exercise:

The iterates for the quadratic example satisfy:

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_\star) = \left(\frac{L-\mu}{L+\mu}\right)^2 (f(\mathbf{x}_k) - f(\mathbf{x}_\star)) \quad k = 0, 1, \ldots$$

Thus, the quadratic example exhibits the worst possible convergence rate for $f \in \mathcal{F}_{\mu,L}$.

Linear convergence

We say gradient descent with exact line search has a linear rate of convergence for $f \in \mathcal{F}_{\mu,L}$ with $0 < \mu \leq L$, since:

$$\limsup_{k\to\infty} \frac{f(\mathbf{x}_{k+1}) - f(\mathbf{x}_\star)}{f(\mathbf{x}_k) - f(\mathbf{x}_\star)} \leq \left(\frac{L-\mu}{L+\mu}\right)^2 \in (0,1).$$

From exact line search to fixed step sizes

Gradient descent with fixed step sizes

- We may also analyze the fixed step size

$$t_k = \frac{2}{\mu + L} \; \forall k$$

  in the same way as for exact line search.

- We need to replace the exact line search conditions:

$$\langle \mathbf{x}_1 - \mathbf{x}_0, \mathbf{g}_1 \rangle = 0, \; \langle \mathbf{g}_0, \mathbf{g}_1 \rangle = 0,$$

  by the fixed step conditions:

$$\mathbf{x}_1 = \mathbf{x}_0 - \frac{2}{\mu + L} \mathbf{g}_0.$$

Gradient descent with fixed step sizes (ctd.)

In the line-search proof (Slide 23) we added the zero terms:

$$\frac{2}{\mu + L}\langle \mathbf{g}_0, \mathbf{g}_1 \rangle + 1 \cdot \langle \mathbf{g}_1, \mathbf{x}_1 - \mathbf{x}_0 \rangle = -\left\langle \mathbf{g}_1, \mathbf{x}_0 - \frac{2}{\mu + L}\mathbf{g}_0 - \mathbf{x}_1 \right\rangle.$$

The fixed step condition $\mathbf{x}_1 = \mathbf{x}_0 - \frac{2}{\mu+L}\mathbf{g}_0$ therefore also guarantees that the expression is zero.

Conclusion:

Gradient descent with fixed step lengths $\frac{2}{\mu+L}$ has the same linear rate of convergence as when using exact line-search if $0 < \mu \leq L$.

Reference:

E. de Klerk, F. Glineur, and A.B. Taylor. On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions. *Optimization Letters* 11, 1185–1199 (2017).

When are the SDP performance bounds guaranteed to be tight?

Tightness in the fixed-step case

Consider the performance problem for the fixed step length $\frac{2}{\mu+L}$.

- Parameters: $L \geq \mu > 0$, $R > 0$;
- Variables: $\{(\mathbf{x}_k, \mathbf{g}_k, f_k)\}_{k \in S}$ $(S = \{\star, 0, 1\})$.

Performance estimation problem:

$$\max \; f_1 - f_\star$$

$$\text{s.t.} \quad \{(\mathbf{x}_k, \mathbf{g}_k, f_k)\}_{k \in S} \quad \mathcal{F}_{\mu,L}\text{-interpolable} \qquad \text{Functional class}$$

$$\mathbf{g}_\star = \mathbf{0} \qquad \text{Optimal point}$$

$$\mathbf{x}_1 = \mathbf{x}_0 - \frac{2}{\mu + L}\mathbf{g}_0 \qquad \text{Algorithm (exact)}$$

$$f_0 - f_\star \leq R \qquad \text{Initial distance}$$

This reformulation is exact. (Why?)

Tightness in the fixed-step case

Consider again the last step to reformulate as an SDP:

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{12} & x_{22} & x_{23} & x_{24} \\ x_{13} & x_{23} & x_{33} & x_{34} \\ x_{14} & x_{24} & x_{34} & x_{44} \end{pmatrix} := \begin{pmatrix} \langle \mathbf{x}_0, \mathbf{x}_0 \rangle & \langle \mathbf{x}_0, \mathbf{x}_1 \rangle & \langle \mathbf{x}_0, \mathbf{g}_0 \rangle & \langle \mathbf{x}_0, \mathbf{g}_1 \rangle \\ \langle \mathbf{x}_0, \mathbf{x}_1 \rangle & \langle \mathbf{x}_1, \mathbf{x}_1 \rangle & \langle \mathbf{x}_1, \mathbf{g}_0 \rangle & \langle \mathbf{x}_1, \mathbf{g}_1 \rangle \\ \langle \mathbf{x}_0, \mathbf{g}_0 \rangle & \langle \mathbf{x}_1, \mathbf{g}_0 \rangle & \langle \mathbf{g}_0, \mathbf{g}_0 \rangle & \langle \mathbf{g}_0, \mathbf{g}_1 \rangle \\ \langle \mathbf{x}_0, \mathbf{g}_1 \rangle & \langle \mathbf{x}_1, \mathbf{g}_1 \rangle & \langle \mathbf{g}_0, \mathbf{g}_1 \rangle & \langle \mathbf{g}_1, \mathbf{g}_1 \rangle \end{pmatrix}.$$

- **NB:** the rank of (x_{ij}) is at most n. (Why?)
- For exactness of the SDP bound we need therefore to add the constraint $\operatorname{rank}(x_{ij}) \leq n$.
- When $n \geq 4$, this constraint is redundant.

Performance estimation for $\mathcal{F}_{0,L}$ with fixed steps

Fixed-step h/L with $h \in [0, 2]$.

Performance problem for the fixed step length $\frac{h}{L}$ and N steps:

- Parameters: $h \in [0, 2]$, $L > 0$, $R > 0$, $N \in \mathbb{N}$;
- Variables: $\{(\mathbf{x}_k, \mathbf{g}_k, f_k)\}_{k \in S}$ $(S = \{\star, 0, \dots, N-1\})$.

Performance estimation problem:

$$\max\ f_1 - f_\star$$

$$\text{s.t.}\ \ \{(\mathbf{x}_k, \mathbf{g}_k, f_k)\}_{k \in S}\ \ \mathcal{F}_{0,L}\text{-interpolable} \qquad \text{Functional class}$$

$$\mathbf{g}_\star = \mathbf{0} \qquad \text{Optimal point}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{h}{L}\mathbf{g}_k\ \ (k = 0, \dots, N-1) \qquad \text{Algorithm}$$

$$\|\mathbf{x}_0 - \mathbf{x}_\star\| \leq R \qquad \text{Initial distance}$$

This reformulation is again exact.

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

Tight worst-case bound for any $h \in [0, 2]$:

$$\max f(\mathbf{x}_N) - f_\star \leq \frac{LR^2}{2} \max\left(\frac{1}{2Nh+1}, (1-h)^{2N}\right)$$

$h \leq 1$ (6 pages of tedious algebra)

Y. Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

$h \leq 2$

T. Rotaru, F. Glineur, and P. Patrinos. Exact worst-case convergence rates of gradient descent: a complete analysis for all constant stepsizes over nonconvex and convex functions. *Mathematical Programming*, 2026.

Improvements when allowing different step lengths

Can one speed up convergence by better step length choice?

- For $\mathcal{F}_{0,L}$ and constant step length $1/L$, we get the tight bound:

$$\boxed{\max f(\mathbf{x}_N) - f_\star \leq \frac{LR^2}{2}\frac{1}{2N+1} = \mathcal{O}\left(\frac{1}{N}\right)}$$

- This is called a sublinear rate of convergence, since

$$\limsup_{k\to\infty} \frac{f(\mathbf{x}_{k+1}) - f_\star}{f(\mathbf{x}_k) - f_\star} = 1,$$

  in the worst case.

- It is possible to get a better sublinear rate of about $\mathcal{O}\left(\frac{1}{N^{1.2716}}\right)$, by using the so-called Silver step-size schedule ...

Silver step-size schedule

Let $\nu(k)$ denote the 2-adic valuation of $k \in \mathbb{N}$, i.e. the position (counting from zero) of the first 1 in the binary expansion of k.

Example

$$\nu(10) = \nu(1010_2) = 1,\ \nu(16) = \nu(10000_2) = 4.$$

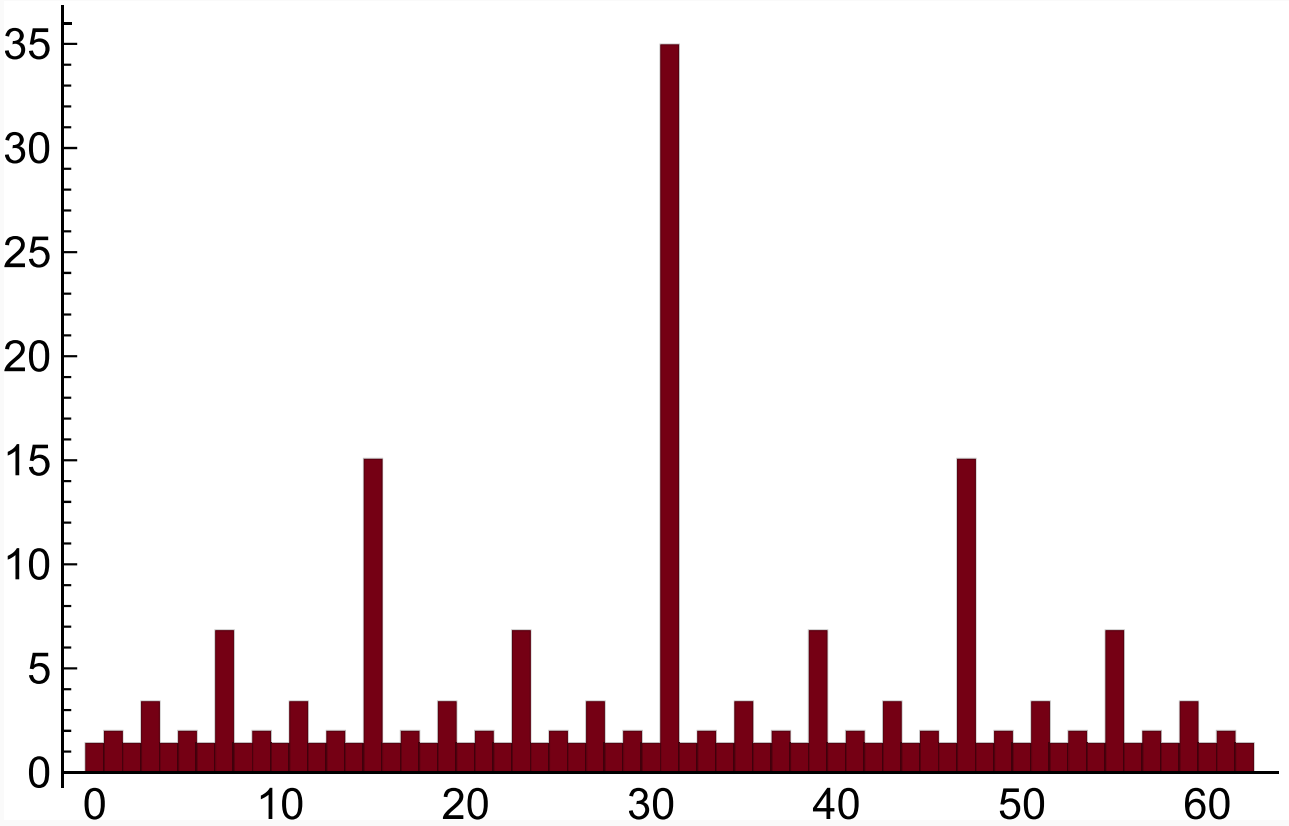
The Silver ratio is the number $\rho = 1 + \sqrt{2}$.

Silver step sizes for $\mathcal{F}_{0,L}$

$$t_k := \frac{1 + \rho^{\nu(k+1)-1}}{L} \qquad (k = 0, 1, \ldots, N-1)$$

Thus $t_0 = \frac{1+\rho^{-1}}{L} = \frac{\sqrt{2}}{L}$, $t_1 = \frac{2}{L}$, $t_2 = \frac{1+\rho^{-1}}{L}$, $t_3 = \frac{2+\sqrt{2}}{L}$, ...

Silver step-sizes $t_k = h_k/L$



Bar plot of the values h_k for some $k \in \mathbb{N}$.

Silver step-size convergence rate

Theorem

For $f \in \mathcal{F}_{0,L}$, the gradient descent method with Silver step sizes guarantees:

$$\max f(\mathbf{x}_N) - f_\star \leq \frac{L\|\mathbf{x}_0 - \mathbf{x}_\star\|^2}{2N^{\log_2 \rho}} \approx \frac{L\|\mathbf{x}_0 - \mathbf{x}_\star\|^2}{2N^{1.2716}}$$

Reference:

Altschuler, J.M., Parrilo, P.A. Acceleration by stepsize hedging: Silver Stepsize Schedule for smooth convex optimization. *Mathematical Programming* 213, 1105–1118 (2025).

The $\mathcal{O}(1/N^{1.2716})$ rate is in between the fixed-step rate $\mathcal{O}(1/N)$ and the best possible rate of $\mathcal{O}(1/N^2)$ for first order methods [Nesterov (1983)].

Silver step-size convergence rate (ctd.)

- For $\mathcal{F}_{\mu,L}$ with $0 < \mu < L$ the Silver step schedule depends on μ and L and on N that should be a power of 2.
- For $N = 2$, the steps are given by

$$t_0 = \frac{2}{\mu + \sqrt{L^2 + (L-\mu)^2}}$$
$$t_1 = \frac{2}{2L + \mu - \sqrt{L^2 + (L-\mu)^2}}.$$

- The Silver step schedule converges superlinearly if $\mu > 0$, i.e.

$$\limsup_{k \to \infty} \frac{f(\mathbf{x}_{k+1}) - f(\mathbf{x}_\star)}{f(\mathbf{x}_k) - f(\mathbf{x}_\star)} = 0.$$

Silver step-size convergence rate (ctd.)

The Silver step-size schedule in the strongly convex case was studied in:

Altschuler, J.M., Parrilo, P.A. Acceleration by Stepsize Hedging I: Multi-Step Descent and the Silver Stepsize Schedule. Journal of the ACM, Volume 72, Issue 2 Article No.: 12, Pages 1 - 38, 2025.

In the computer session, we will see how to "discover" the Silver step sizes for $N = 2$ though SDP performance estimation.

Computer session

Computer session

- Adapted from tutorial given at SIAM Optimization conference in Edinburgh by François Glineur and Adrien Taylor.
- Uses the PEPit Python software.
- Older Matlab version also available, called PESTO.

PEPit reference:

B. Goujaud, C. Moucer, F. Glineur, J. Hendrickx, A. Taylor, A. Dieuleveut. "PEPit: computer-assisted worst-case analyses of first-order optimization methods in Python." Math. Prog. Comp. 16, 337–367 (2024).

Many thanks to François Glineur and Adrien Taylor!

Computer session: links

- Use your browser to open (or scan the QR code below):
  `https://github.com/PerformanceEstimation/Tutorial-Europt-2026`
- Click on Part 1 in left-hand-side menu.
- Click on `Europt2026_Tutorial_Part1.ipynb`
- Click on Open in Colab (requires a Google account).