Accelerated gradient descent for large-scale optimization

On Gradient descent solving quadratic problems

Andersen Ang

ECS, Uni. Southampton, UK
andersen.ang@soton.ac.uk

Homepage angms.science

Version: March 13, 2025

First draft: November 12, 2018
Guest lecture of MARO 201 - Advanced Optimization
Faculte Polytechnique de Mons
Unversite de Mons

Content

Table of Contents

Introduction

- Theme: solve

$$\boxed{\begin{array}{l}(\mathcal{P})\text{: given } \boldsymbol{A} \in \mathbb{R}^{m\times n}, \boldsymbol{b} \in \mathbb{R}^m\text{, find } \boldsymbol{x} \in \mathbb{R}^n \text{ by solving} \\ \displaystyle\operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) \;\coloneqq\; \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2.\end{array}}$$

- 1st-order optimality condition $\nabla f(\boldsymbol{x}) = \boldsymbol{0}$ gives

$$\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top\boldsymbol{b} \;=\; \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) \;=\; \boldsymbol{A}\boldsymbol{x}-\boldsymbol{b} \;=\; \boldsymbol{0} \in \mathbb{R}^m.$$

  we denote $\nabla f(\overline{\boldsymbol{x}})$ the gradient vector of f in the standard basis in $\mathbb{R}^n$ with respect to the variable $\boldsymbol{x}$ at the point $\overline{\boldsymbol{x}}$

- Solving $(\mathcal{P})$ is the same as solving linear inverse problem: given $\boxed{\begin{array}{l}\boldsymbol{A} \in \mathbb{R}^{m\times n} \\ \boldsymbol{b} \in \mathbb{R}^m\end{array}}$, solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.

- Linear algebra solution: $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ if $\boldsymbol{A}^{-1}$ exists.

This lecture: constrained convex quadratic problem

$(\mathcal{P})$: given $\boldsymbol{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$, find $\boldsymbol{x} \in \mathcal{C} \subset \mathbb{R}^n$ by solving

$$\operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) \;:=\; \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

- We focus on the following **simple convex set** $\mathcal{C}$
  - $\mathcal{C} = \mathbb{R}^n$ (no constraint), $(\mathcal{P})$ = least squares (LS).
  - $\mathcal{C} = \mathbb{R}^n_+$ (nonnegative constraint), $(\mathcal{P})$ = Nonnegative LS (NNLS).

- Some other $\mathcal{C}$ (not the focus) :
  - $\mathcal{C} = \left\{\boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \|\boldsymbol{x}\|_2 \leq \epsilon\right\}$ (l_2-norm constraint).
  - $\mathcal{C} = \left\{\boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \|\boldsymbol{x}\|_1 \leq \epsilon\right\}$ (l_1-norm constraint).
  - $\mathcal{C} = \left\{\boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \|\boldsymbol{x}\|_0 \leq \epsilon\right\}$ (l_0-norm constraint).
  - $\mathcal{C} = \left\{\boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \|\boldsymbol{x}\|_{\mathrm{TV}} \leq \epsilon\right\}$ (Total variation constraint).
  - $\mathcal{C} = \left\{\boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \boldsymbol{C}\boldsymbol{x} \leq \boldsymbol{d}\right\}$ (polytope constraint).

Why study $(\mathcal{P})$

$(\mathcal{P})$: given $\boldsymbol{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$, find $\boldsymbol{x} \in \mathcal{C} \subset \mathbb{R}^n$ by solving

$$\operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) \ := \ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

1. **$(\mathcal{P})$ is easy to understand.**
   We will look at **gradient descent**, and the **accelerated variants** through the lens of $(\mathcal{P})$.

2. **$(\mathcal{P})$ is applicable to wide range of situations.**
   We see $(\mathcal{P})$ when we take **2nd-order Taylor's approximation** of a model.

3. **$(\mathcal{P})$ is easy to generalize/modify.**

Facts: basic properties of quadratic problem $(\mathcal{P})$

- $(\mathcal{P})$ is a convex problem: local minima $\rightarrow$ global minima
  - $\mathcal{C}$ is a convex set [assumption]
  - cost function f is convex
- $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x} + c$, where $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A},\ \boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b},\ c = \frac{1}{2}\|\boldsymbol{b}\|_2^2$.
- $(\mathcal{P})$ has a unique global minimum $\boldsymbol{x}^*$
  - if $\boldsymbol{A}$ full rank [assumption] $\implies$ $\boldsymbol{Q}$ is positive definite $\iff$ $\lambda_{\min}(\boldsymbol{Q}) > 0$.
- Gradient $\nabla f(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b}$
- f is L-smooth with $L = \|\boldsymbol{Q}\|_2 = \sqrt{\lambda_{\max}(\boldsymbol{Q}^\top \boldsymbol{Q})} = \lambda_{\max}(\boldsymbol{Q}) = \sigma_{\max}(\boldsymbol{A})$.
  - A function $g(x)$ is L-smooth if the gradient ∇g is L-Lipschitz : $\|\nabla g(a) - \nabla g(b)\| \leq L\|a - b\|\ \forall a, b \in \mathrm{dom} g$
  - $\|\boldsymbol{Q}\|_2 = \sqrt{\lambda_{\max}((\boldsymbol{A}^\top \boldsymbol{A})^2)} = \sqrt{\lambda_{\max}^2(\boldsymbol{A}^\top \boldsymbol{A})} = \lambda_{\max}(\boldsymbol{A}^\top \boldsymbol{A}) = \sigma_{\max}(\boldsymbol{A})$

$$\begin{aligned} \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 &= \|(\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}) - (\boldsymbol{Q}\boldsymbol{y} - \boldsymbol{p})\|_2 \\ &= \|\boldsymbol{Q}(\boldsymbol{x} - \boldsymbol{y})\|_2 \\ &\overset{\text{oni}}{\leq} \|\boldsymbol{Q}\|_2 \|\boldsymbol{x} - \boldsymbol{y}\|_2. \end{aligned}$$

oni: operator norm inequality $\|\boldsymbol{A}\boldsymbol{x}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{x}\|$.

Table of Contents

Iterative algorithm to solve $(\mathcal{P})$

- Iterative solver: produce a sequence $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots\}$ such that

$$f(\boldsymbol{x}_{k+1}) \;\leq\; f(\boldsymbol{x}_k) \text{ for all } k. \qquad \text{(Descent condition)}$$

- How: starting with an initial guess $\boldsymbol{x}_0$, perform the iteration/update

$$\underbrace{\boldsymbol{x}_{k+1}}_{\text{new variable}} = \underbrace{\boldsymbol{x}_k}_{\text{current variable}} + \underbrace{t}_{\text{stepsize}} \cdot \underbrace{\Delta}_{\text{direction}}.$$

  until a stopping condition is met.

- Two remarks
  1. Δ is called "direction" but it can have magnitude larger than 1.
     The true "stepsize" is $t\|\Delta\|$, the true "direction" is $\Delta\|\Delta\|^{-1}$.
  2. We focus on algo., but choosing $\boldsymbol{x}_0$ also affect the performance of algo. $\rightarrow$ a topic on its own.

Stopping condition (not the focus here)

Type	order	Stop if
Variable change	$-$	$\mathrm{dist}(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}) \leq \epsilon$
Functional value	0th	$f(\boldsymbol{x}_k) \leq \epsilon$
Successive functional value	0.5th	$\left\lvert f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k-1}) \right\rvert \leq \epsilon$
Gradient value	1st	$\lVert \nabla f(\boldsymbol{x}_k) \rVert_2 \leq \epsilon$
Hessian value	2nd	$\lambda_{\min}\Big(\nabla^2 f(\boldsymbol{x}_k)\Big) \geq \epsilon > 0$

Which one to use and what are their pros & cons will take another hour to explain.

I call successive functional value as "0.5th" order because $\left\lvert f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k-1}) \right\rvert$ resembles $\lVert \nabla f \rVert$: by the fact that f is convex and L-smooth, we have that for $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)$, its holds

$$f(\boldsymbol{x}_{k+1}) \geq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), t_k \nabla f(\boldsymbol{x}_k) \rangle + \frac{L}{2} \lVert t_k \nabla f(\boldsymbol{x}_k) \rVert_2^2$$

Hence

$$\left\lvert f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) \right\rvert \geq \Big(t_k + \frac{L}{2} t_k^2 \Big) \lVert \nabla f(\boldsymbol{x}_k) \rVert_2^2$$

Gradient descent (GD)

- GD uses $\Delta = -\nabla f(\boldsymbol{x})$ so we have $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t\nabla f(\boldsymbol{x}_k)$.

- Choices of stepsize t
  1. Optimal stepsize / exact line search: $t_{\text{Exact}} = \operatorname{argmin}_{t \geq 0} f(\boldsymbol{x} + t\Delta)$.
     Practically the best but can be expensive to compute.
  2. Line search stepsize: backtracking / sufficient descent / Armijo Rule / Wolfe condition.
     Practically okay and commonly used.
  3. Constant stepsize: t is fix.
     Practically not the best, but theoretically easier to analyze.
  4. * Constant stepsize using Lipschitz constant: $t_L = \dfrac{1}{L(f)}$.
     A common choice of stepsize for theorist.
  5. Diminishing stepsize: $t_{k+1} = t_k \cdot \theta$, $\theta < 1$, or $t_k = \dfrac{1}{k}$.
     Good for stochastic gradient.

  Our focus: 1 and 4.

GD with optimal stepsize on $(\mathcal{P})$: $\operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x}$

$$t_{\text{Exact}} = \operatorname*{argmin}_{t\geq 0} f\Big(\boldsymbol{x} - t\nabla f(\boldsymbol{x})\Big) = \frac{\nabla f(\boldsymbol{x})^\top \nabla f(\boldsymbol{x})}{\nabla f(\boldsymbol{x})^\top \boldsymbol{Q}\nabla f(\boldsymbol{x})} = \frac{\|\nabla f(\boldsymbol{x})\|_2^2}{\|\nabla f(\boldsymbol{x})\|_{\boldsymbol{Q}}^2}.$$

Derivation: reading exercise

$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x}$. To get $t_{\text{Exact}} = \operatorname*{argmin}_{t\geq 0} g(t) = f(\boldsymbol{x} - t\nabla f(\boldsymbol{x}))$, set $\dfrac{\partial g(t)}{\partial \boldsymbol{x}} = 0$.

$g(t) = \dfrac{1}{2}(\boldsymbol{x} - t\nabla f(\boldsymbol{x}))^\top \boldsymbol{Q}(\boldsymbol{x} - t\nabla f(\boldsymbol{x})) - \boldsymbol{p}^\top(\boldsymbol{x} - t\nabla f(\boldsymbol{x}))$, ignore terms without t:

$$\begin{aligned}
g(t) &= \tfrac{1}{2}\Big(- \boldsymbol{x}^\top \boldsymbol{Q} t\nabla f(\boldsymbol{x}) - t\nabla f(\boldsymbol{x})^\top \boldsymbol{Q}\boldsymbol{x} + t^2 \nabla f(\boldsymbol{x})^\top \boldsymbol{Q}\nabla f(\boldsymbol{x})\Big) + t\boldsymbol{p}^\top \nabla f(\boldsymbol{x}) \\
&= -\boldsymbol{x}^\top \boldsymbol{Q} t\nabla f(\boldsymbol{x}) + \frac{1}{2}t^2 \nabla f(\boldsymbol{x})^\top \boldsymbol{Q}\nabla f(\boldsymbol{x}) + t\boldsymbol{p}^\top \nabla f(\boldsymbol{x}) \\
\frac{\partial g(t)}{\partial t} &= -\boldsymbol{x}^\top \boldsymbol{Q}\nabla f(\boldsymbol{x}) + t\nabla f(\boldsymbol{x})^\top \boldsymbol{Q}\nabla f(\boldsymbol{x}) + \boldsymbol{p}^\top \nabla f(\boldsymbol{x}) \\
&= -(\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p})^\top \nabla f(\boldsymbol{x}) + t\nabla f(\boldsymbol{x})^\top \boldsymbol{Q}\nabla f(\boldsymbol{x}) \\
&= -\nabla f(\boldsymbol{x})^\top \nabla f(\boldsymbol{x}) + t\nabla f(\boldsymbol{x})^\top \boldsymbol{Q}\nabla f(\boldsymbol{x}).
\end{aligned}$$

Finally, $\dfrac{\partial g(t)}{\partial \boldsymbol{x}} = 0 \implies \nabla f(\boldsymbol{x})^\top \nabla f(\boldsymbol{x}) = t\nabla f(\boldsymbol{x})^\top \boldsymbol{Q}\nabla f(\boldsymbol{x})$. □

GD algo. with t_{Exact} on $(\mathcal{P})$: $\underset{\boldsymbol{x}}{\operatorname{argmin}}\ f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x},\ \boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A},\ \boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$

$$
\begin{aligned}
\text{GD update:} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - t_{\text{Exact}} \nabla f(\boldsymbol{x}_k) \\
&= \boldsymbol{x}_k - \frac{\|\nabla f(\boldsymbol{x}_k)\|_2^2}{\|\nabla f(\boldsymbol{x}_k)\|_{\boldsymbol{Q}}^2} \nabla f(\boldsymbol{x}_k) \\
&= \boldsymbol{x}_k - \frac{\|\boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{p}\|_2^2}{\|\boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{p}\|_{\boldsymbol{Q}}^2} (\boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{p})
\end{aligned}
$$

Algorithm 1: GD (with t_{Exact}) for $(\mathcal{P})$

Result: A sol. $\boldsymbol{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$

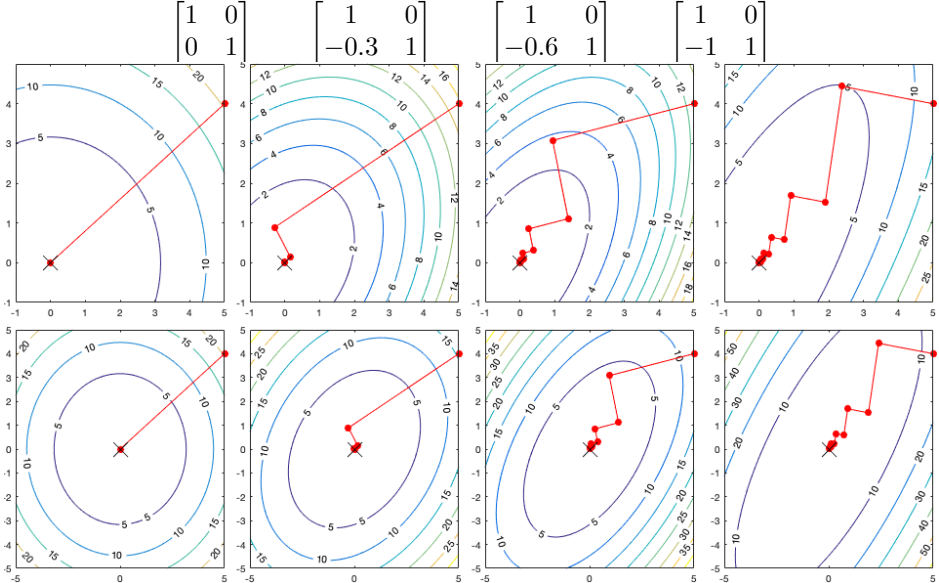
while stopping condition is not met **do**

  $\boldsymbol{g} = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}$

  $\boldsymbol{x} = \boldsymbol{x} - \frac{\|\boldsymbol{g}\|_2^2}{\|\boldsymbol{g}\|_{\boldsymbol{Q}}^2} \boldsymbol{g}$

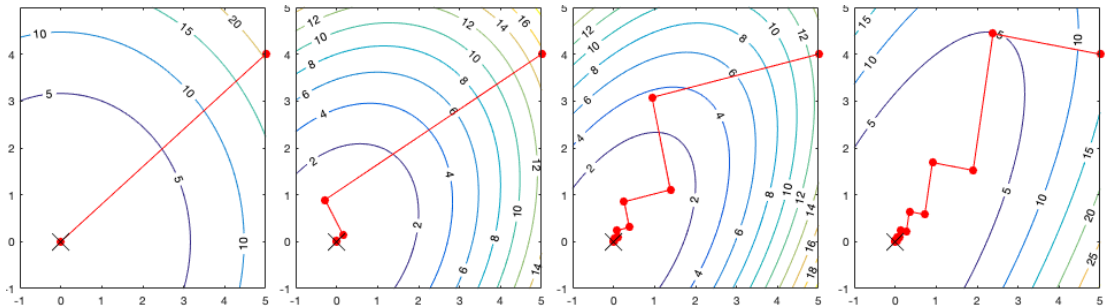
end

An example in $\mathbb{R}^2$: $\boldsymbol{x}^{\text{True}} = [0\ 0]^\top, \boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{x}^{\text{True}}, \boldsymbol{x}_0 = [5\ 4]^\top$, 30 iterations, four $\boldsymbol{A}$



Observations

- If the level set of $f(\boldsymbol{x})$ is circular, GD goes to $\boldsymbol{x}^*$ very fast. (In fact, in 1 step)
- When the level set of $f(\boldsymbol{x})$ is elliptic, GD zigzags (and slow).



Questions

- Why zigzags? Where does this zigzag come from?
- How to deal with it: how to improve GD?

Remarks on GD with t_{Exact}

- GD update with t_{Exact}:

$$\boldsymbol{x} = \boldsymbol{x} - \frac{\|\nabla f(\boldsymbol{x})\|_2^2}{\|\nabla f(\boldsymbol{x})\|_{\boldsymbol{Q}}^2} \nabla f(\boldsymbol{x}) = \boldsymbol{x} - \frac{(\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p})^\top (\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p})}{(\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p})^\top \boldsymbol{Q} (\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p})} (\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}).$$

- The step size is not constant, it depends on $\boldsymbol{x}$
  $\rightarrow$ hard to analyze.

- The step size contains many vector/matrix-vector/matrix products
  $\rightarrow$ not suitable on problem with big m, n.

- We now consider GD with *fixed step size* t_L.

GD alg. with fixed $t_L = \dfrac{1}{L}$ on $(\mathcal{P})$: $\operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x}, \quad \boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}, \quad \boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$

$$\text{GD update: } \boldsymbol{x}^+ \;=\; \boldsymbol{x} - \frac{1}{\|\boldsymbol{Q}\|_2}\nabla f(\boldsymbol{x}) \;=\; \boldsymbol{x} - \frac{1}{\|\boldsymbol{Q}\|_2}(\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}) \;=\; \cdots$$

we can further simplify this expression, come back to this later.

Algorithm 2: GD for $(\mathcal{P})$

Result: A solution $\boldsymbol{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$

Pre-compute $L = \|\boldsymbol{Q}\|_2$

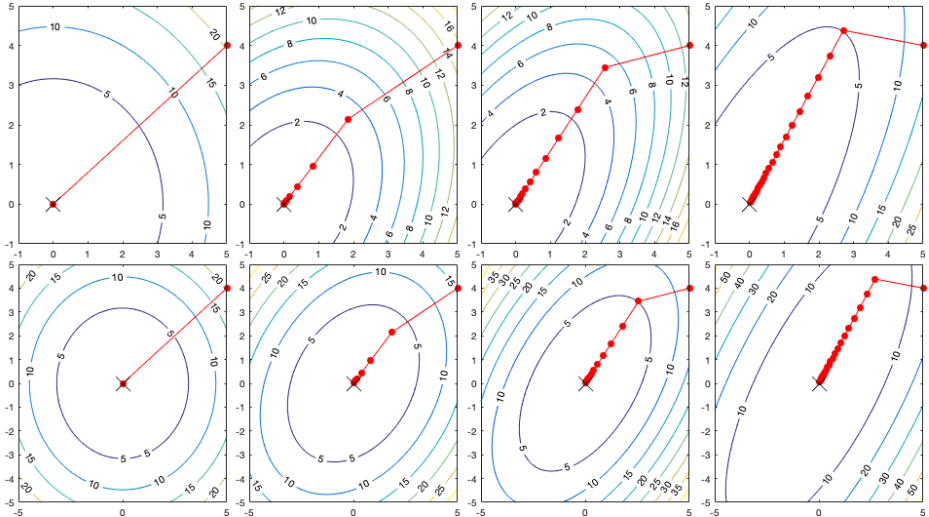
while stopping condition is not met **do**

  $\boldsymbol{x} = \boldsymbol{x} - \dfrac{1}{L}(\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p})$.

end

Same example in $\mathbb{R}^2$, but with stepsize t_L

Using t_L has slower convergence, but more applicable to big problem.

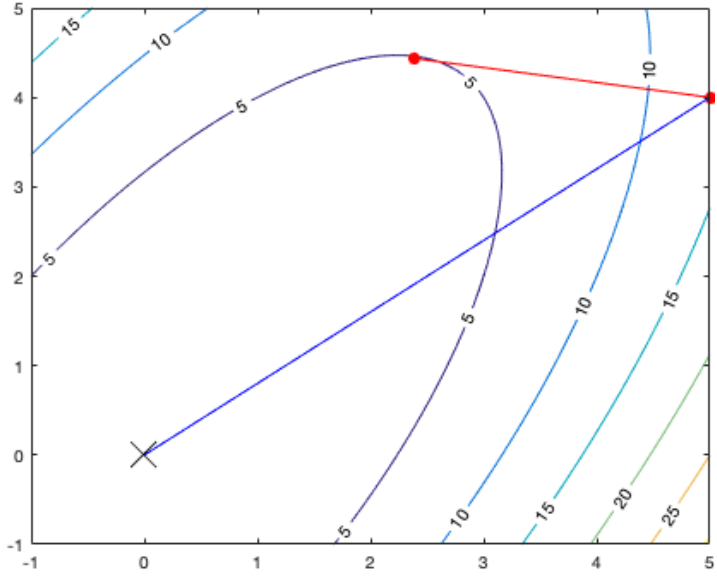


Test your understanding: why here t_L is fixed but the distances between consecutive points are changing? (Answer in p.8)

Table of Contents

GD is "stupid"

Figure: the first iteration of GD for the elliptic case.

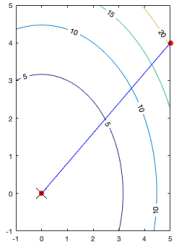


Question: Why GD moves along the red line, not the blue line?

Deeper understanding of GD: Why GD moves along the red line not the blue line?

- Two more questions:
  - Where does the expression $\boldsymbol{x} = \boldsymbol{x} - t\nabla f(\boldsymbol{x})$ "come from"?
  - What does the expression $\boldsymbol{x} = \boldsymbol{x} - t\nabla f(\boldsymbol{x})$ "actually do"?
- Answers :
  - It comes from a **local quadratic model** (denote as F) of f.
  - It minimizes that **local quadratic model** F.
- GD is stupid because GD is "local"
  - GD uses **local** info. (which is F) to makes **local decision** – so it follows the red path.
  - Blue path is a **global decision** that needs **global info**.
  - Being a local decision maker, there is no guarantee the decision made by GD will be as good as the global optimal one.
  - GD is a *greedy algorithm*.

Question: then why GD made such a good move here?
Answer: it just happens **when f is "nice"**, the local decision made by GD is **coincidentally** as good as the global one.



The local model F that GD minimizes

- GD step takes local info $\boldsymbol{x}_k$ to minimize a local model F of f

$$\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k - t\nabla f(\boldsymbol{x}_k) \;=\; \operatorname*{argmin}_{\boldsymbol{x}} F(\boldsymbol{x}; \boldsymbol{x}_k).$$

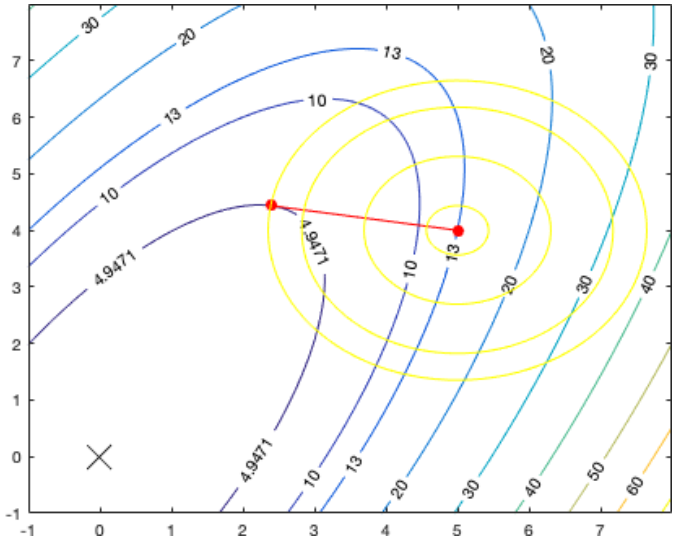
- Such local model F :
  - Takes $\boldsymbol{x}_k$ as parameter
  - Is a 2nd-order function of $\boldsymbol{x}$
  - Expression of $F(\boldsymbol{x}; \boldsymbol{x}_k) = f(\boldsymbol{x}_k) + (\boldsymbol{x} - \boldsymbol{x}_k)^\top \nabla f(\boldsymbol{x}_k) + \dfrac{1}{2t_k}\|\boldsymbol{x} - \boldsymbol{x}_k\|_2^2$.
    How come: $\dfrac{\partial F}{\partial \boldsymbol{x}} = 0$ yields $\boldsymbol{x}_k - t\nabla f(\boldsymbol{x}_k)$.
  - Equivalent expression of F

$$F(\boldsymbol{x}; \boldsymbol{x}_k) = \frac{1}{2t_k}\left\|\boldsymbol{x} - \Big(\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)\Big)\right\|_2^2.$$

  - Note that F is **spherical**.

GD makes progress based on the local spherical model

For t_{Exact}:



For t_L : a fixed factor $\frac{1}{L}$ is used.

GD zigzaging theorem

Question: why GD zigzags with t_{Exact}?
Answer: in fact this always occurs.

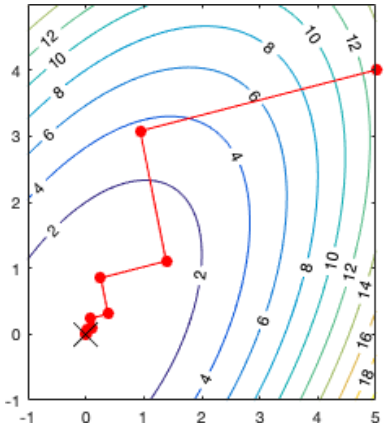
Theorem (reading exercise) Consecutive gradient directions with t_{Exact} are orthogonal to each other: $\nabla f(\boldsymbol{x}_{k+1}) \perp \nabla f(\boldsymbol{x}_k)$.

Proof: $\nabla f(\boldsymbol{x}_{k+1}) \perp \nabla f(\boldsymbol{x}_k)$ means $\langle \nabla f(\boldsymbol{x}_{k+1}), \nabla f(\boldsymbol{x}_k) \rangle = 0$.

To show this, recall t_{Exact} minimizes $g(t) = f(\boldsymbol{x}_k - t\nabla f(\boldsymbol{x}_k))$.

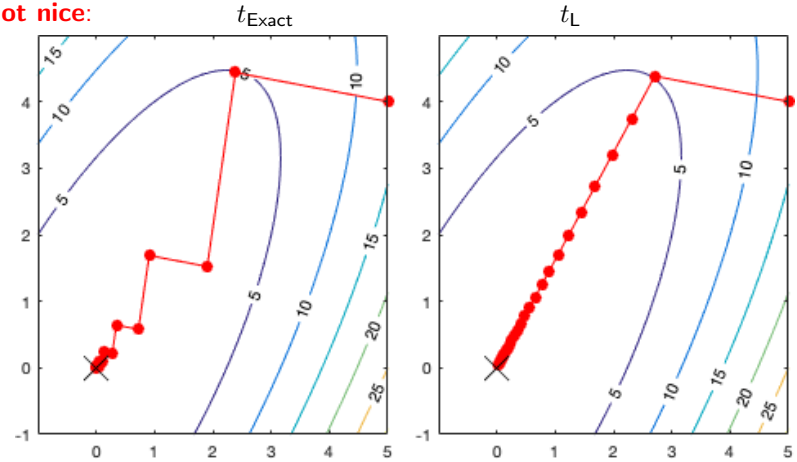
Consider $\dfrac{\partial g}{\partial t} = 0$, we have

$$\begin{aligned} \frac{\partial g}{\partial t} &\overset{\text{chain rule}}{=} \Big\langle -\nabla f(\boldsymbol{x}_k), \nabla f\Big(\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)\Big) \Big\rangle \\ &= -\Big\langle \nabla f(\boldsymbol{x}_k), \nabla f\big(\boldsymbol{x}_{k+1}\big) \Big\rangle \\ &= 0. \quad \square \end{aligned}$$



The problem of gradient descent

When f is not nice:



To improve GD:

1. On dynamic step size t_{Exact}: reduce the zigzag.
2. On fixed step size t_L: "move more" when appropriate.

Summary so far

$(\mathcal{P})$: given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, find $\boldsymbol{x} \in \mathbb{R}^n$ by solving

$$\boldsymbol{x} \coloneqq \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x},$$

where $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $L = \|\boldsymbol{Q}\|_2$.

Gradient Descent algorithm $\boldsymbol{x} = \boldsymbol{x} - t\nabla f(\boldsymbol{x})$

- $t_{\text{Exact}} = \dfrac{\|\nabla f(\boldsymbol{x})\|_2^2}{\|\nabla f(\boldsymbol{x})\|_{\boldsymbol{Q}}^2}$ or $t_L = \dfrac{1}{L}$.
- GD is a local decision maker.
- GD = minimizing a local quadratic model F of f at point $\boldsymbol{x}_k$.
- When level sets of f is elliptic:
  - GD zigzags with t_{Exact}.
  - GD is slow with t_L.

Table of Contents

Acceleration by extrapolation

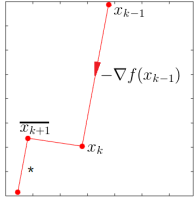
- We will look at two extrapolation schemes:
  - Polyak's Heavy Ball Method (HBM)
  - Nesterov's acceleration
- Idea: add "momentum" to the current iterate.
- Momentum = the "previous history".

$$\begin{aligned} \text{HBM} \qquad & \boldsymbol{x}_{\text{updated}} + \text{previous momentum} \\ = \quad & \overline{\boldsymbol{x}_{k+1}} + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}),\ \beta_k \geq 0 \\ = \quad & \overline{\boldsymbol{x}_{k+1}} - \beta_k t_{k-1} \nabla f(\boldsymbol{x}_{k-1}) \end{aligned}$$

Push $\boldsymbol{x}_{\text{updated}}$ along $-\nabla f(\boldsymbol{x}_{k-1})$ for $*$ amount $* = \beta_k t_{k-1} \|\nabla f(\boldsymbol{x}_{k-1})\|$.

- Other momentum (not the focus)
  - Adam (a variable metric gradient method)
  - Donald G. Anderson acceleration

Nesterov (2018, 2025)



Three update schemes

- **Normal gradient**

$$\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)$$

Move the point $\boldsymbol{x}_k$ in the direction $-\nabla f(\boldsymbol{x}_k)$ for $t_k \|\nabla f(\boldsymbol{x})\|$ amount.

- **Polyak's Heavy Ball Method**

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

Perform a GD, move the updated-$\boldsymbol{x}$ in the direction of the previous step for $\beta_k \|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|$ amount.

- **Nesterov's acceleration**

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

Move the not-yet-updated-$\boldsymbol{x}$ in the direction of the previous step for $\beta_k \|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|$ amount, perform a GD on the shifted-$\boldsymbol{x}$, then move the updated-$\boldsymbol{x}$ in the direction of the previous step for $\beta_k \|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|$ amount.

Algorithm 3: HBM for $(\mathcal{P})$, with step size t_{Exact}

Result: A solution $\boldsymbol{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\beta \geq 0$

while stopping condition is not met **do**

$$\boldsymbol{g} = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}$$

$$\boldsymbol{x} = \boldsymbol{x} - \frac{\|\boldsymbol{g}\|_2^2}{\|\boldsymbol{g}\|_{\boldsymbol{Q}}^2}\boldsymbol{g} + \beta(\boldsymbol{x} - \boldsymbol{x}^-)$$

end

Algorithm 4: HBM for $(\mathcal{P})$, with step size t_L

Result: A solution $\boldsymbol{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\beta \geq 0$

while stopping condition is not met **do**

$$\boldsymbol{g} = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}$$

$$\boldsymbol{x} = \boldsymbol{x} - \frac{1}{\|\boldsymbol{Q}\|_2}\boldsymbol{g} + \beta(\boldsymbol{x} - \boldsymbol{x}^-)$$

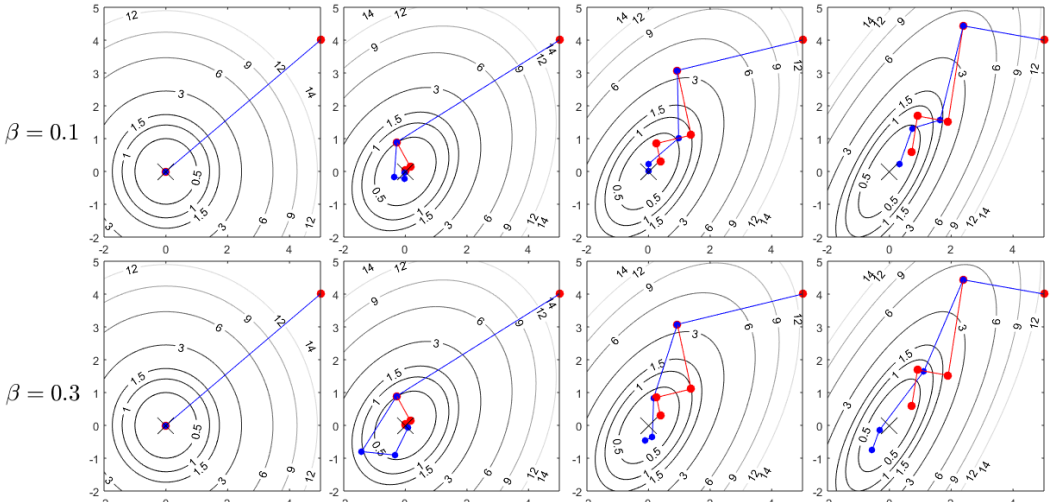
end

- Fixed β is used here.
- When $\beta = 0$, HBM reduces to GD

HBM with t_{Exact}

Same example set up, 4 iterations.

Red: normal gradient. Blue: HBM with different fixed β.

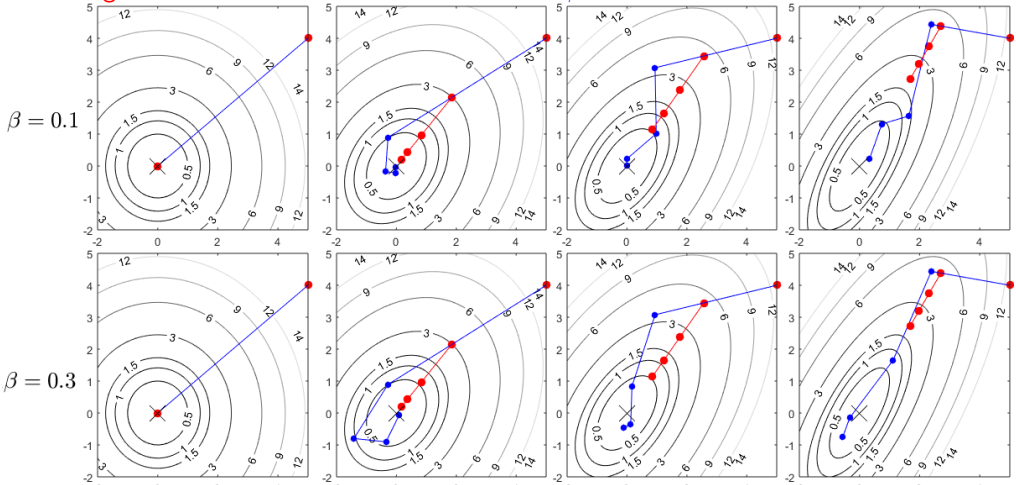


Observation: for nice f (with spherical level sets), GD is already good enough and HBM adds a little effect. However, for bad f (with elliptic level sets), HBM is better in some cases.

HBM with t_L

Same example set up, 4 iterations.

Red: normal gradient. Blue: HBM with different fixed β.



Observation: same. If nice f (spherical lv. sets), GD is already good enough. If bad f (with elliptic lv. sets), HBM is better in some cases.

If f is nice, GD doesn't need acceleration.
If f is nice, GD doesn't need acceleration.
If f is nice, GD doesn't need acceleration.

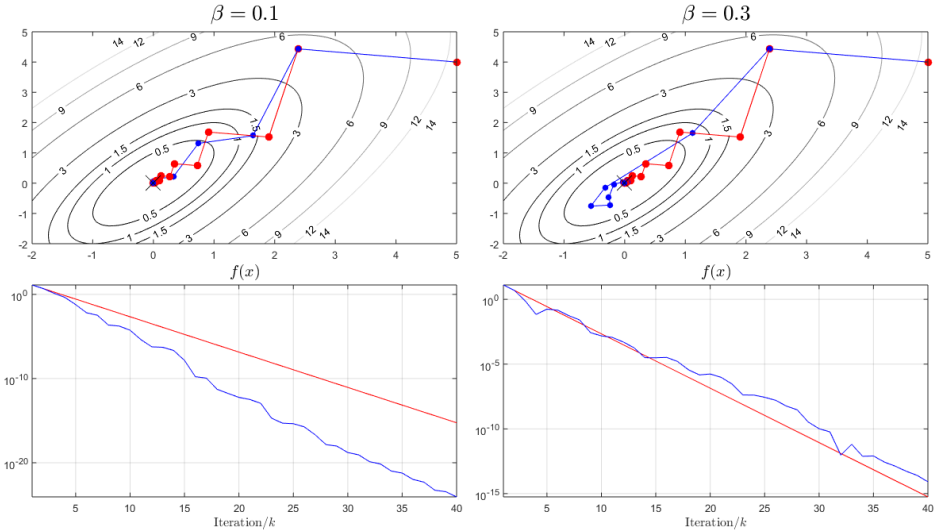
This is so important so I repeated 3 times.

The message: do not use acceleration blindly, for some problems GD don't need acceleration. Using acceleration blindly doesn't make you look cool. Knowing when to use it makes you look cool.

Effect of different β on HBM (on elliptic f)

In HBM, we need to guess a good β. A bad β gives bad effect.

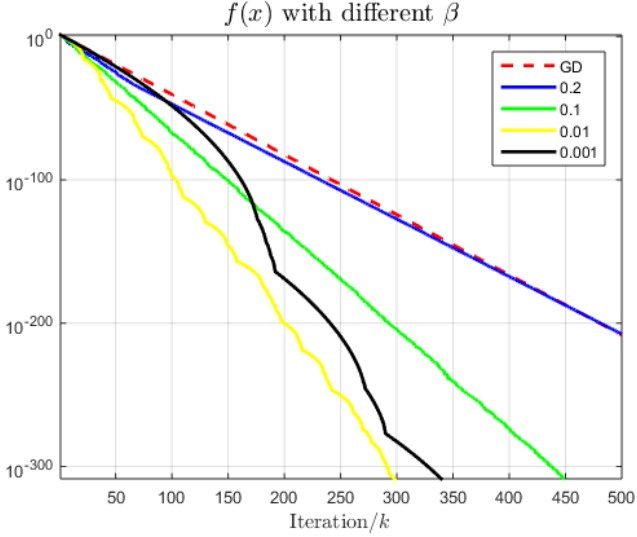
GD $\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)$ HBM $\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k) + \beta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$



Effect of different β on HBM, more iterations

Question: the smaller β, the better HBM?

Answer: No.



Is there a way to find best β? What about dynamic β?

Nesterov's acceleration

$$
\begin{array}{rl}
\text{GD} & \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k) \\
\text{HBM} & \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k) + \beta_k (\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \\
\text{Nesterov} & \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k + \beta_k (\boldsymbol{x}_k - \boldsymbol{x}_{k-1})) + \beta_k (\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \\
\text{Nesterov-2} & \boldsymbol{y}_{k+1} = \boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k) \\
& \boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1} + \beta_k (\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)
\end{array}
$$

- We saw HBM with fixed β.
- Nesterov gave the update scheme with *close-form formula* for β_k (in 1983)

$$
\alpha_1 \in [0, 1], \ \alpha_{k+1} = \frac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}, \ \beta_k = \frac{\alpha_k (1 - \alpha_k)}{\alpha_k^2 + \alpha_{k+1}}.
$$

  Note: β_k is not fix, it is a function of α_1. We need to guess α_1.

- How to get Nesterov-2 from Nesterov : set $\boldsymbol{x}_{-1} = \boldsymbol{x}_0$, $\boldsymbol{y}_{-1} = \boldsymbol{y}_0$

Algorithm 5: Nesterov's accelerated gradient for $(\mathcal{P})$

Result: A solution $\boldsymbol{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\alpha_1 \in (0\ 1)$

while stopping condition is not met **do**

  Compute $\nabla f(\boldsymbol{x}_k)$ and step size t

  Compute $\alpha_{k+1} = \dfrac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}$, $\beta_k = \dfrac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$

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

  $\boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1} + \beta_k(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)$

end

Table of Contents

Constrained problem and projected gradient descent

$(\mathcal{P})$: given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, find $\boldsymbol{x} \in \mathcal{C} \subset \mathbb{R}^n$ by solving

$$\boldsymbol{x} := \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{C}} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

The basic GD becomes *Projected GD*.

Algorithm 6: PGD for $(\mathcal{P})$, with step size t

Result: A solution $\boldsymbol{x} \in \mathcal{C}$ that approximately solves $(\mathcal{P})$

1 **Initialization** Set $\boldsymbol{x}_0 \in \mathcal{C}$
2 **while** stopping condition is not met **do**
3 Compute $\nabla f(\boldsymbol{x})$ and step size t_k
4 Compute gradient update $\boldsymbol{y} = \boldsymbol{x} - t_k \nabla f(\boldsymbol{x})$
5 Compute projection $\boldsymbol{x} = P_{\mathcal{C}}(\boldsymbol{y})$
6 **end**

Gradient and projection in 1 line: $\boldsymbol{x} = P_{\mathcal{C}}\Big(\boldsymbol{x} - t_k \nabla f(\boldsymbol{x})\Big)$.

Note: now the distance traveled between consecutive points is not $t\|\nabla f(\boldsymbol{x})\|$ but $\|\boldsymbol{x} - P_{\mathcal{C}}(\boldsymbol{x} - t_k \nabla f(\boldsymbol{x}))\|$.

Nonnegative least squares

$(\mathcal{P})$: given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, find $\boldsymbol{x} \in \mathbb{R}^n_+$ by solving $\boldsymbol{x} := \underset{\boldsymbol{x} \in \mathbb{R}^n_+}{\operatorname{argmin}} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$.

Algorithm 7: PGD for $(\mathcal{P})$, with step size t

Result: A solution $\boldsymbol{x} \in \mathbb{R}^n_+$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n_+$

while stopping condition is not met **do**

  Compute $\nabla f(\boldsymbol{x})$ and step size t_k

  Compute projected gradient update $\boldsymbol{x} = [\boldsymbol{x} - t_k \nabla f(\boldsymbol{x})]_+ = \max(\boldsymbol{x}, 0)$.

end

Algorithm 8: Nesterov's accelerated projected gradient for $(\mathcal{P})$

Result: A solution $\boldsymbol{x} \in \mathbb{R}^n_+$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n_+$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\alpha_1 \in (0\ 1)$

while stopping condition is not met **do**

  Compute $\nabla f(\boldsymbol{x}_k)$ and step size t

  Compute $\alpha_{k+1} = \dfrac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}$, $\beta_k = \dfrac{\alpha_k(1 - \alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$

  $\boldsymbol{y}_{k+1} = [\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)]_+$

  $\boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1} + \beta_k(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)$

end

PGD vs Accelerated PG on NNLS

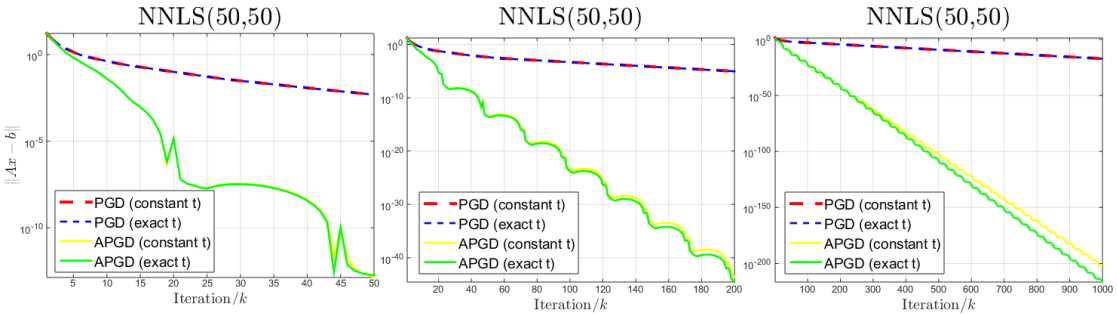


Figure: $m = n = 50$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{x}^* = 0$, $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\alpha_1 = 0.9$

- APGD is much faster.
- PGD is *monotonic*, APGD is not.
- Choice of step size makes small difference.
- Important: APGD and PGD have a very different convergence rate
  - the y-axis is in log-scale.
  - the curve of PGD and that of APGD have different slopes.

Other β_k schemes

- Nesterov's parameters looks so complicated

$$\alpha_{k+1} = \frac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2},\ \beta_k = \frac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$$

- Another Nesterov's parameters

$$\alpha_{k+1}^2 = (1-\alpha_{k+1})\alpha_k^2 + \kappa^{-1}\alpha_{k+1},\ \beta_k = \frac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$$

- Yet another Nesterov's parameters

$$\alpha_{k+1} = \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2},\ \beta_k = \frac{1-\alpha_k}{\alpha_{k+1}}.$$

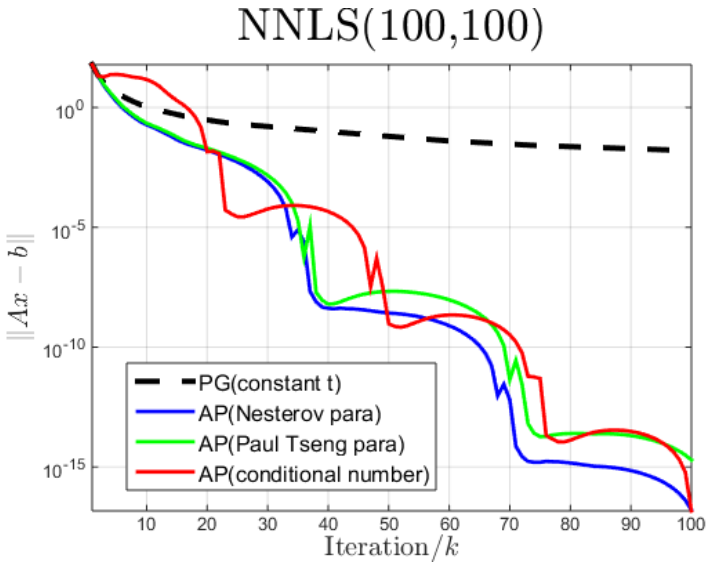
- Paul Tseng parameter

$$\beta_k = \frac{k-1}{k+2},\ \text{or}\ \frac{1}{k+2},$$

- Using conditional number

$$\beta_k = \beta = \frac{1-\sqrt{\kappa'}}{1+\sqrt{\kappa'}},\quad \kappa' = \frac{1}{\kappa},\ \kappa = \frac{\sigma_{\max}(\boldsymbol{Q})}{\sigma_{\min}(\boldsymbol{Q})} = \frac{\lambda_{\max}(\boldsymbol{Q})}{\lambda_{\min}(\boldsymbol{Q})}$$

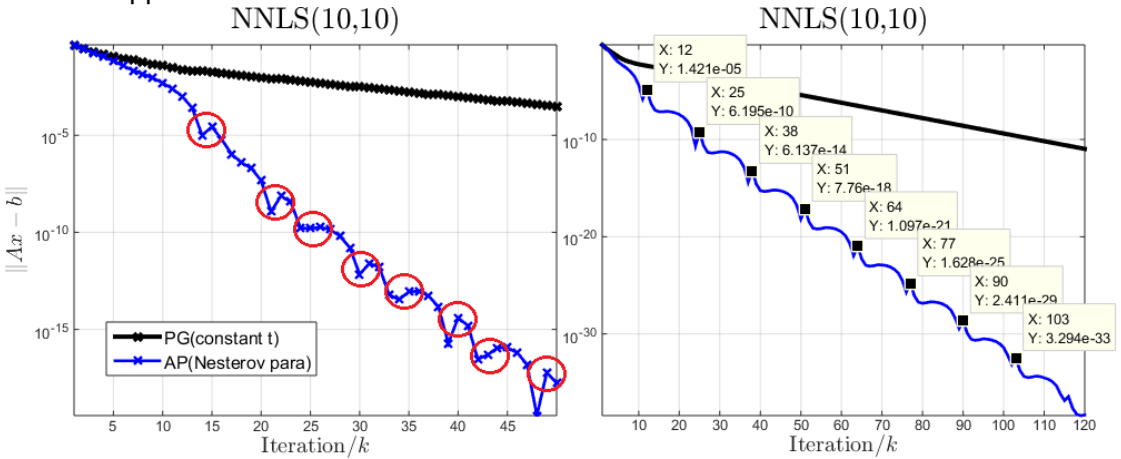
Different β on NNLS



$m = n = 100$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{x}^* = 0$, $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\alpha_1 = 0.9$

Table of Contents

Nesterov's ripple



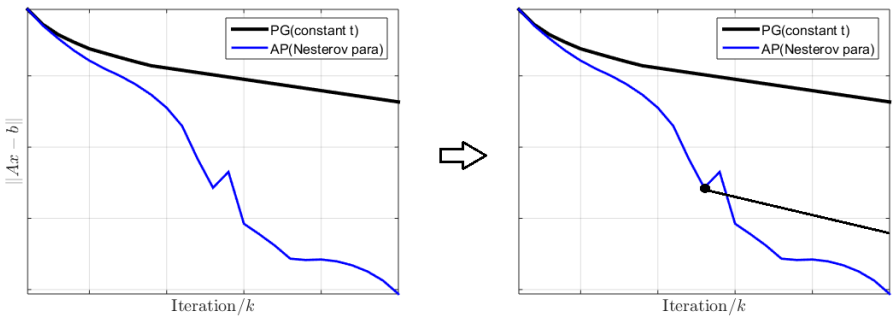
$m = n = 10$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{x}^* = 0$, $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\alpha_1 = 0.9$

- Function values of GD always go down.
- Function values of APGD sometimes go up.
- APGD exhibits some periodic behavior.

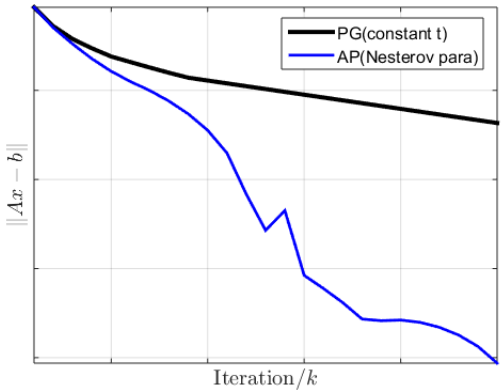
Acceleration with restarts

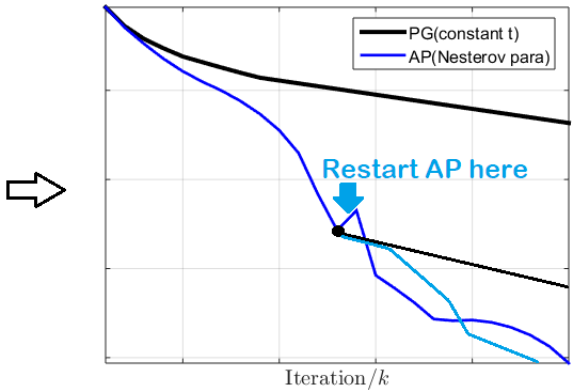
- When you run GD $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t\nabla f(\boldsymbol{x}_k)$, you always get $f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k)$.
  Why :
  - GD is the global minimizer of a local approximator F of f at $\boldsymbol{x}_k$.
    More technical, see theorem 2 in `https://angms.science/doc/CVX/CVX_GD_Convergence.pdf`
  - GD is a local decision maker, a *greedy algorithm*.
- We "twist" the Accelerated GD algorithm:

Go back to normal GD when function value ↑



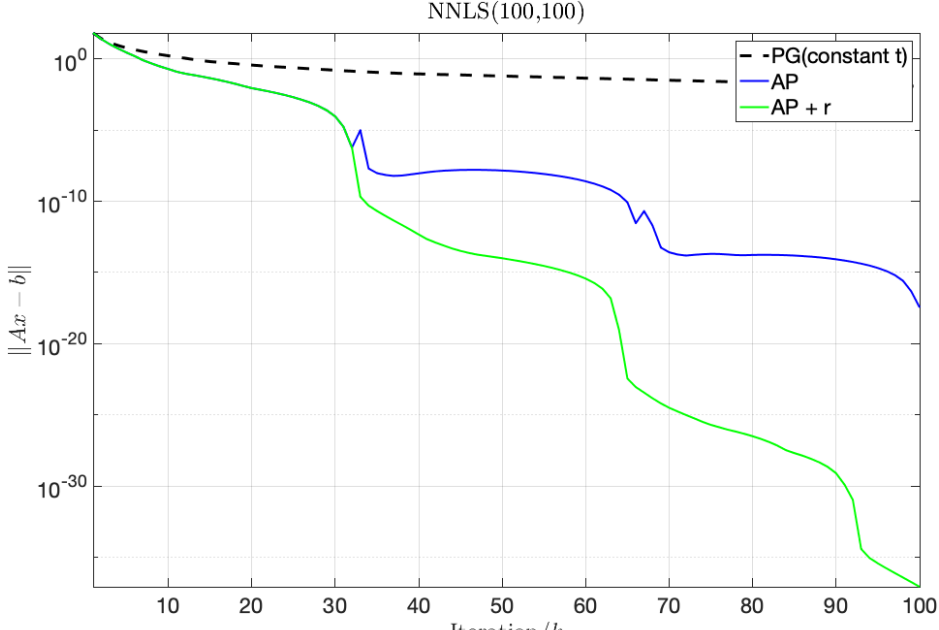
Acceleration with restarts





But go back to GD = go back to a slow algorithm!
→ Re-run acceleration right after back to GD.

Effect of restarts on APGD



Algorithm 9: APGD (using Nesterov's parameter) for $(\mathcal{P})$, no restart

Result: A solution $\boldsymbol{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n_+$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\alpha_1 \in (0\ 1)$

while stopping condition is not met **do**

  Compute $\nabla f(\boldsymbol{x}_k)$, step size t_k

  Compute $\alpha_{k+1} = \dfrac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}$, $\beta_k = \dfrac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$

  $\boldsymbol{y}_{k+1} = [\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)]_+$

  $\boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1} + \beta_k(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)$

end

Algorithm 10: APGD (using Nesterov's parameter) for $(\mathcal{P})$ with restart

Result: A solution $\boldsymbol{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n_+$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\alpha_1 \in (0\ 1)$

while stopping condition is not met **do**

  Compute $\nabla f(\boldsymbol{x}_k)$, step size t_k

  Compute $\alpha_{k+1} = \dfrac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}$, $\beta_k = \dfrac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$

  $\boldsymbol{y}_{k+1} = [\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)]_+$

  $\boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1} + \beta_k(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)$

  If error increase **do**

    $\boldsymbol{x}_{k+1} = [\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)]_+$ (go back to gradient descent)

    $\boldsymbol{y}_{k+1} = \boldsymbol{x}_{k+1},\ \alpha_k = \alpha_1$ (reset all parameters)

  end if

end

Algorithm 15: APGD (using Paul Tseng's parameter) for $(\mathcal{P})$ with restart

Result: A solution $\boldsymbol{x}$ that approximately solves $(\mathcal{P})$

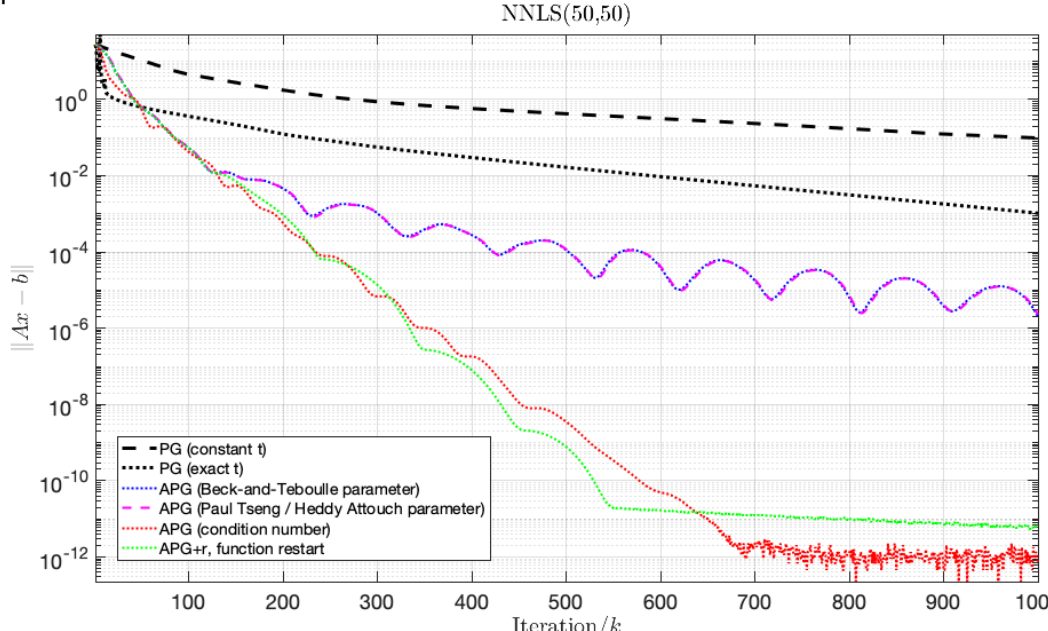
Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n_+$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\hat{k} = 0$
while stopping condition is not met **do**
  Compute $\nabla f(\boldsymbol{x}_k)$, step size t_k
  Compute $\beta_k = \dfrac{(k-\hat{k})-1}{(k-\hat{k})+2}$
  $\boldsymbol{y}_{k+1} = [\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)]_+$
  $\boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1} + \beta_k(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)$
  If error increase **do**
    $\boldsymbol{x}_{k+1} = [\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)]_+$ (go back to gradient descent)
    $\boldsymbol{y}_{k+1} = \boldsymbol{x}_{k+1}$, $\hat{k} = \hat{k} + k$ (reset all parameters)
  end if
end

Algorithm 16: APGD (using Conditional number) for $(\mathcal{P})$ with restart

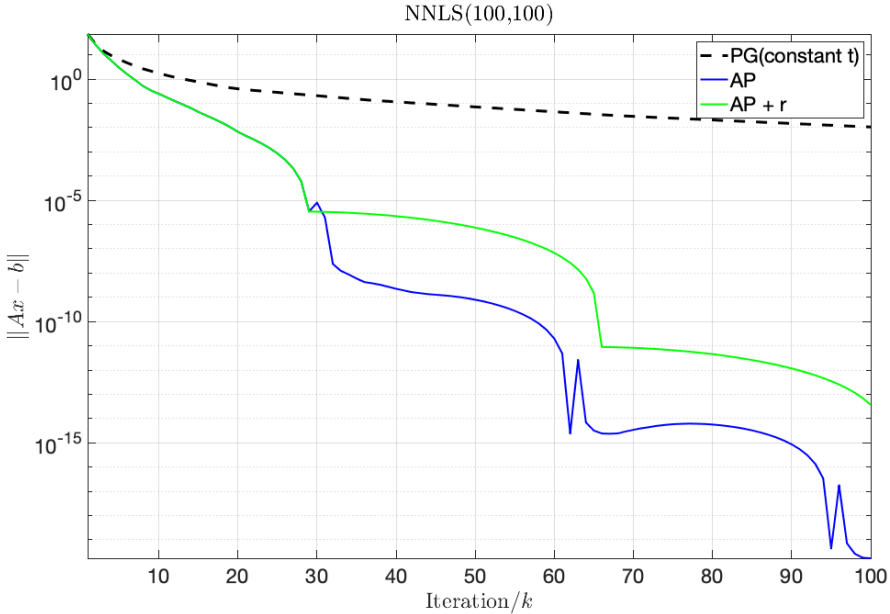
Result: A solution $\boldsymbol{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\boldsymbol{x}_0 \in \mathbb{R}^n_+$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\beta = \dfrac{1-\sqrt{\kappa'}}{1+\sqrt{\kappa'}}$
while stopping condition is not met **do**
  Compute $\nabla f(\boldsymbol{x}_k)$, step size t_k
  $\boldsymbol{y}_{k+1} = [\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)]_+$
  $\boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1} + \beta(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)$
  If error increase **do**
    $\boldsymbol{x}_{k+1} = [\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)]_+$ (go back to gradient descent)
    $\boldsymbol{y}_{k+1} = \boldsymbol{x}_{k+1}$ (reset all parameters)
  end if
end

Comparing all schemes



Warning: restart is not "always good"



Summary so far

$(\mathcal{P})$: given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, find $\boldsymbol{x} \in \mathcal{C}$ by solving

$$\boldsymbol{x} \coloneqq \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x},$$

$\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $L = \|\boldsymbol{Q}\|_2$.

Algorithm: Gradient Descent $\boldsymbol{x} = \boldsymbol{x} - t\nabla f(\boldsymbol{x})$

- Minimizes a local model of f, slow if level sets of f is elliptic.

Acceleration : add momentum $\beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$

- HBM $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t\nabla f(\boldsymbol{x}_k) + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$
- Nesterov $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t\nabla f(\boldsymbol{x}_k + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})) + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$
- Nesterov's parameter $\alpha_{k+1} = \dfrac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}$, $\beta_k = \dfrac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$
- Other parameters $\beta = \dfrac{k-1}{k+2}$, $\beta = \dfrac{1-\sqrt{\kappa'}}{1+\sqrt{\kappa'}}$
- Adaptive restarts

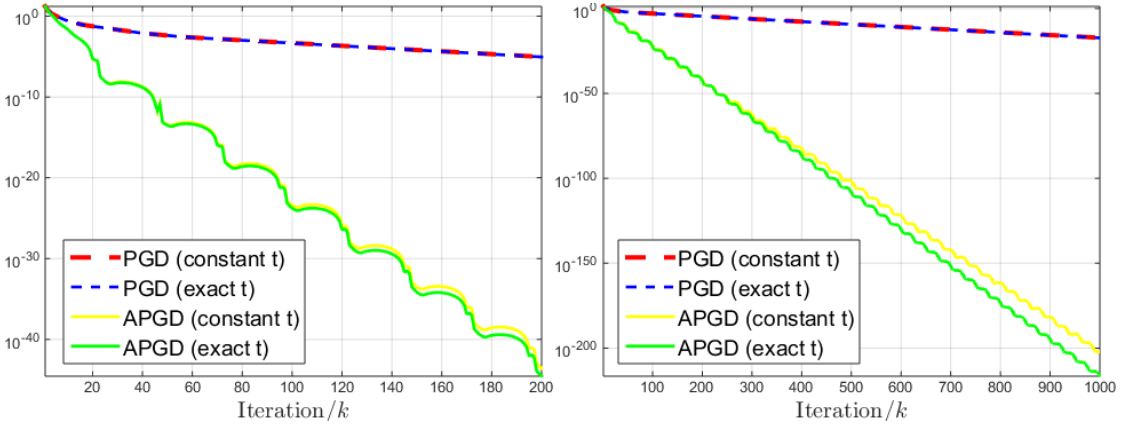
Why the acceleration works?

Pourquoi l'acceleration fonctionne?

Table of Contents

Convergence rate

Recall the observation : different slope $\Longrightarrow$ different convergence rate



We are going to study the convergence properties of GD.

- For simplicity, consider the model $(\mathcal{P})$ with no constraint.
- The theory applies for general any smooth convex f.

Convergence condition of GD on $(\mathcal{P})$

$(\mathcal{P})$: given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, find $\boldsymbol{x} \in \mathbb{R}^n$ by solving

$$\boldsymbol{x} \coloneqq \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x},$$

where $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, assumes $\boldsymbol{A}$ is full rank.

Theorem:

If stepsize $t_k > 0$ fulfills:

$$\max\left\{|1 - t_k \lambda_{\max}(\boldsymbol{Q})|, |1 - t_k \lambda_{\min}(\boldsymbol{Q})|\right\} < 1 \text{ for all } k \in \mathbb{N},$$

the sequence $\{\boldsymbol{x}_k\}_{k \in \mathbb{N}}$ produced by GD iteration

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

converges to the minimizer $\boldsymbol{x}^*$ of $(\mathcal{P})$.

Why: reading exercise in the next three pages.

Convergence condition of gradient descent ... 1/2

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x},\ \nabla f(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p} \text{ and } \begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k) \\ &= \boldsymbol{x}_k - t_k(\boldsymbol{Q}\boldsymbol{x}_k - \boldsymbol{p}) = \boldsymbol{x}_k - t_k\boldsymbol{Q}\boldsymbol{x}_k + t_k\boldsymbol{p} \\ &= (\boldsymbol{I}_n - t_k\boldsymbol{Q})\boldsymbol{x}_k + t_k\boldsymbol{p} \end{aligned}$$

Recall 1st-order optimality condition $\nabla f(\boldsymbol{x}^*) = 0$ so $\boldsymbol{Q}\boldsymbol{x}^* - \boldsymbol{p} = 0$.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= (\boldsymbol{I}_n - t_k\boldsymbol{Q})\boldsymbol{x}_k + t_k\boldsymbol{Q}\boldsymbol{x}^* \\ \boldsymbol{x}_{k+1} - \boldsymbol{x}^* &= (\boldsymbol{I}_n - t_k\boldsymbol{Q})\boldsymbol{x}_k + t_k\boldsymbol{Q}\boldsymbol{x}^* - \boldsymbol{x}^* \\ &= (\boldsymbol{I}_n - t_k\boldsymbol{Q})\boldsymbol{x}_k + (t_k\boldsymbol{Q} - \boldsymbol{I}_n)\boldsymbol{x}^* \\ &= (\boldsymbol{I}_n - t_k\boldsymbol{Q})(\boldsymbol{x}_k - \boldsymbol{x}^*) \\ \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 &= \|(\boldsymbol{I}_n - t_k\boldsymbol{Q})(\boldsymbol{x}_k - \boldsymbol{x}^*)\|_2 \\ &\overset{\text{oni}}{\leq} \|\boldsymbol{I}_n - t_k\boldsymbol{Q}\|_2 \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2. \ \text{(oni : operator norm inequality)} \end{aligned}$$

Let $\mu\boldsymbol{I}_n \preceq \lambda_{\min}(\boldsymbol{Q})\boldsymbol{I}_n \preceq \boldsymbol{Q} \preceq \lambda_{\max}(\boldsymbol{Q})\boldsymbol{I}_n \preceq L\boldsymbol{I}_n$, then

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 \leq \max\left\{|1 - t_kL|, |1 - t_k\mu|\right\}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2$$

Derivation

$$\begin{array}{rrcccl} & \mu\boldsymbol{I}_n & \preceq & \boldsymbol{Q} & \preceq & L\boldsymbol{I}_n \\ \iff & -L\boldsymbol{I}_n & \preceq & -\boldsymbol{Q} & \preceq & -\mu\boldsymbol{I}_n \\ \iff & -t_kL\boldsymbol{I}_n & \preceq & -t_k\boldsymbol{Q} & \preceq & -t_k\mu\boldsymbol{I}_n \\ \iff & \boldsymbol{I}_n - t_kL\boldsymbol{I}_n & \preceq & \boldsymbol{I}_n - t_k\boldsymbol{Q} & \preceq & \boldsymbol{I}_n - t_k\mu\boldsymbol{I}_n \\ \iff & (1 - t_kL)\boldsymbol{I}_n & \preceq & \boldsymbol{I}_n - t_k\boldsymbol{Q} & \preceq & (1 - t_k\mu)\boldsymbol{I}_n \\ \iff & \|(1 - t_kL)\boldsymbol{I}_n\|_2 & \leq & \|\boldsymbol{I}_n - t_k\boldsymbol{Q}\|_2 & \leq & \|(1 - t_k\mu)\boldsymbol{I}_n\|_2 \\ \iff & |1 - t_kL| \cdot \|\boldsymbol{I}_n\|_2 & \leq & \|\boldsymbol{I}_n - t_k\boldsymbol{Q}\|_2 & \leq & |1 - t_k\mu| \cdot \|\boldsymbol{I}_n\|_2 \\ \iff & |1 - t_kL| & \leq & \|\boldsymbol{I}_n - t_k\boldsymbol{Q}\|_2 & \leq & |1 - t_k\mu| \end{array}$$

So $\|\boldsymbol{I}_n - t_k\boldsymbol{Q}\|_2 \leq \max\left\{|1 - t_kL|, |1 - t_k\mu|\right\}$.

Convergence of gradient descent ... 2/2

$$
\begin{aligned}
\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 &\leq \max\left\{|1 - t_k L|, |1 - t_k \mu|\right\} \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2 \\
&\leq \max\left\{|1 - t_k L|, |1 - t_k \mu|\right\} \Big(\max\left\{|1 - t_{k-1} L|, |1 - t_{k-1} \mu|\right\} \|\boldsymbol{x}_{k-1} - \boldsymbol{x}^*\|_2 \Big) \\
&\quad \vdots \\
&\leq \Big(\prod_k \max\left\{|1 - t_k L|, |1 - t_k \mu|\right\} \Big) \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2.
\end{aligned}
$$

Hence GD converge for $(\mathcal{P})$ if

$$
\max\left\{|1 - t_k L|, |1 - t_k \mu|\right\} < 1 \quad \forall k. \quad \square
$$

Convergence of GD on $(\mathcal{P})$ with fix stepsize

$$\begin{aligned}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\|_2 &\leq \Big(\prod_k \max\big\{|1-t_kL|,|1-t_k\mu|\big\}\Big)\|\boldsymbol{x}_0-\boldsymbol{x}^*\|_2\\ &= \Big(\max\big\{|1-tL|,|1-t\mu|\big\}\Big)^k\|\boldsymbol{x}_0-\boldsymbol{x}^*\|_2.\end{aligned}$$

Case A $\quad \|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\|_2 \leq |1-tL|^k\|\boldsymbol{x}_0-\boldsymbol{x}^*\|_2.$
Case B $\quad \|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\|_2 \leq |1-t\mu|^k\|\boldsymbol{x}_0-\boldsymbol{x}^*\|_2.$

No matter which case, we need $|\cdot|<1$.
Case A $|1-tL|<1 \iff 1-tL<1$ or $1-tL>-1$

$$\begin{aligned}&\iff && -tL<0 \quad \text{or} \quad -tL>-2\\ &\iff && t>0 \qquad \text{or } t<\frac{2}{L}\\ &\iff && 0<t<\frac{2}{L}\end{aligned}$$

Similarly we get $0<t<\dfrac{2}{\mu}$ for case B.

Convergence of GD on $(\mathcal{P})$ for fix t

$(\mathcal{P})$: given full rank $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, find $\boldsymbol{x} \in \mathbb{R}^n$ by solving

$$\boldsymbol{x} \coloneqq \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x},$$

where $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $L = \|\boldsymbol{Q}\|_2$.

We have

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 \leq \Big(\max\big\{|1 - tL|, |1 - t\mu|\big\}\Big)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2.$$

Suppose case A occurs

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 \leq |1 - tL|^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2.$$

We have

Theorem (Convergence of GD with fixed step size, case A)
For fixed step size

$$0 < t < \frac{2}{L},$$

the sequence $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ produced by GD converges to $\boldsymbol{x}^*$ of $(\mathcal{P})$.

Convergence of GD on $(\mathcal{P})$ with $t = \dfrac{1}{L}$

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 \leq \Big(\max\big\{|1 - tL|, |1 - t\mu|\big\}\Big)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2.$$

If $t = \dfrac{1}{L}$

$$\begin{aligned} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 &\leq \left(\max\left\{ \left|1 - \frac{L}{L}\right|, \left|1 - \frac{\mu}{L}\right| \right\} \right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2 \\ &= \left(1 - \frac{1}{\kappa}\right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2, \text{ where } \kappa = \frac{L}{\mu} > 1 \end{aligned}$$

- If $\kappa = 1$, $1 - \dfrac{1}{\kappa} = 0$, and GD converge in 1 step.
- If $\kappa = 9999$, $1 - \dfrac{1}{\kappa} = 0.999...$, and GD converge very slowly.

Convergence of GD on $(\mathcal{P})$ with the "best" t

The "best" $t = \dfrac{2}{L+\mu}$

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 \leq \left(1 - \frac{2}{\kappa+1}\right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2.$$

Why: put $t = \dfrac{2}{L+\mu}$ into $\max\{|1-tL|, |1-t\mu|\}$:

$$\max\left\{\left|1 - \frac{2L}{L+\mu}\right|, \left|1 - \frac{2\mu}{L+\mu}\right|\right\} = \max\left\{\left|\frac{-L+\mu}{L+\mu}\right|, \left|\frac{L-\mu}{L+\mu}\right|\right\} = \frac{L-\mu}{L+\mu} = \frac{\kappa-1}{\kappa+1} = 1 - \frac{2}{\kappa+1}.$$

- If $\kappa = 1$, $1 - \dfrac{2}{\kappa+1} = 0$, and GD converge in 1 step
- If $\kappa = 9999$, $1 - \dfrac{2}{\kappa+1} = 0.999...$, and GD converge very slow

  Compared with $t = \dfrac{1}{L}$ with $\left(1 - \dfrac{1}{\kappa}\right)^k$: let $\kappa = 2$, we have $\dfrac{1}{2^k}$ vs $\dfrac{1}{3^k}$.

Summary on the convergence rate of GD on $(\mathcal{P})$

Theorem (Convergence condition of GD)
If step size $t_k > 0$ satisfies the following condition

$$\max\left\{|1 - t_k\lambda_{\max}(\boldsymbol{Q})|, |1 - t_k\lambda_{\min}(\boldsymbol{Q})|\right\} < 1 \ \forall k \in \mathbb{N},$$

the sequence $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ produced by GD converges to the minimizer $\boldsymbol{x}^*$.

And the theorem with different stepsize t :

- If $t = \dfrac{1}{L}$ then

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 \leq \left(1 - \frac{1}{\kappa}\right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2.$$

- If $t = \dfrac{2}{L+\mu}$ then

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 \leq \left(1 - \frac{2}{\kappa+1}\right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2.$$

Table of Contents

Convergence rate of GD on general smooth f

- Previously we studied the convergence rate of GD on minimizing

$$f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

  That is, we restrict f to be the least squares function.
- Now we consider general f that is smooth and convex.
- We now study the convergence rate of GD on

$$\operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x})$$

  where
  - f is convex
  - f is L-smooth: the gradient of f is L-Lipschitz

$$\|\nabla f(\boldsymbol{a}) - \nabla f(\boldsymbol{b})\| \leq L\|\boldsymbol{a} - \boldsymbol{b}\|.$$

A very important inequality for L-smooth f

- If f is L-smooth: the gradient of f is L-Lipschitz

$$\|\nabla f(\boldsymbol{a}) - \nabla f(\boldsymbol{b})\| \leq L\|\boldsymbol{a} - \boldsymbol{b}\|.$$

  Then

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

- Why and how: see next two pages (reading exercise).

Proof (1/2)

We show for $L > 0$, $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$ implies

$$\left|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \left\langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \right\rangle\right| \leq \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2.$$

Recall from calculus $G(b) - G(a) = \int_a^b g(\theta)d\theta$. Next, a smart step, let $g(\theta)$ as $g(\tau) = \langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})), \boldsymbol{y} - \boldsymbol{x} \rangle$ be a function in τ and $d\theta = d\tau$. Consider the definite integral of $g(\tau)$ from 0 to 1, let $G(b) = f(\boldsymbol{y})$ and $G(a) = f(\boldsymbol{x})$, hence

$$\begin{aligned} f(\boldsymbol{y}) - f(\boldsymbol{x}) &= \int_0^1 \Big\langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})), \boldsymbol{y} - \boldsymbol{x} \Big\rangle d\tau \\ &= \int_0^1 \Big\langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}) + \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \Big\rangle d\tau. \end{aligned}$$

As $\nabla f(\boldsymbol{x})$ is independent of τ, can take out from the integral

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) = \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \int_0^1 \Big\langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \Big\rangle d\tau.$$

The idea is to create the term $\langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle$ so that we can move it to the left and get $\left|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \left\langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \right\rangle\right|$.

Proof (2/2)

$$\begin{array}{rcl} |f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle| & = & |\int_0^1 \langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \, d\tau| \\ & \leq & \int_0^1 \left| \langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \right| d\tau \\ & \overset{\text{c.s.}}{\leq} & \int_0^1 \|\nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x})\| \cdot \|\boldsymbol{y} - \boldsymbol{x}\| d\tau. \end{array}$$

c.s. means Cauchy – Schwarz inequality.

Now look at $\|\nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x})\|$, this is exactly where we can apply the Lipschitz gradient inequality

$$\|\nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x})\| \leq L\|\tau(\boldsymbol{y} - \boldsymbol{x})\| \leq L|\tau|\|\boldsymbol{y} - \boldsymbol{x}\| = L\tau\|\boldsymbol{y} - \boldsymbol{x}\|$$

where $\|\tau(\boldsymbol{y} - \boldsymbol{x})\| = |\tau|\|\boldsymbol{y} - \boldsymbol{x}\|$ as norm is non-negative. Note that the integral range is from 0 to 1 so the absolute sign in τ can be removed.

Lastly

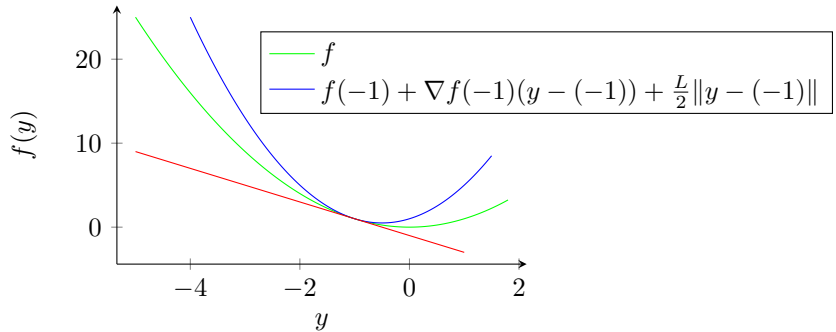
$$\left|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \left\langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \right\rangle\right| \leq \int_0^1 L\tau d\tau \cdot \|\boldsymbol{y} - \boldsymbol{x}\|^2 = \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2.$$

Remove the absolute value sign gives

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2. \quad \square$$

Meaning of L-smoothness: quadratic upper bound

A function f is L-**smooth** if for any two points $\boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom} f$,
$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2$



Interpretation: f is globally bounded above by a quadratic function.
i.e. f cannot be "grow too fast" than the quadratic upper bound.

Polyak-Lojasiewicz inequality

- A function $f : \mathbb{R}^n \to \mathbb{R}$ that is differentiable (i.e. $\nabla f(\boldsymbol{x})$ exists for $\boldsymbol{x} \in \mathrm{dom} f$) satisfies Polyak-Lojasiewicz (PL) inequality if there exists a positive scalar $\mu > 0$ such that

$$\frac{1}{2}\|\nabla f(\boldsymbol{x})\|^2 \geq \mu\big(f(\boldsymbol{x}) - f^*\big)$$

  for all $\boldsymbol{x} \in \mathrm{dom} f$, where $f^* = f(\boldsymbol{x}^*)$ and $\boldsymbol{x}^*$ is a minimizer of f.

- It means the gradient grows faster than a quadratic function (scaled by a scalar $\mu > 0$) as we move $\boldsymbol{x}$ away from $\boldsymbol{x}^*$.

- Why an inequality with crazy name suddenly jumps out from nowhere: because it is USEFUL!

- Good news: this inequality is true for many many many functions you see in daily life. So you can assume the f you work with is PL.

Poof of linear convergence of GD

$$\begin{aligned} f \text{ has } L\text{-Lips. grad} \quad f(\boldsymbol{y}) &\leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 \\ \text{GD update} \quad \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k) \end{aligned}$$

- Put $\boldsymbol{y} = \boldsymbol{x}_{k+1}$, $\boldsymbol{x} = \boldsymbol{x}_k$ in the first inequality, then plug in the second equation gives the *descent lemma*

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) - \frac{1}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2.$$

- PL inequality: $-\frac{1}{2}\|\nabla f(\boldsymbol{x}_k)\|^2 \leq -\mu\big(f(\boldsymbol{x}_k) - f^*\big)$, so

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) - \frac{\mu}{L}\big(f(\boldsymbol{x}_k) - f^*\big).$$

- Subtract both side by f^*

$$f(\boldsymbol{x}_{k+1}) - f^* \leq f(\boldsymbol{x}_k) - \frac{\mu}{L}\big(f(\boldsymbol{x}_k) - f^*\big) - f^* = \big(1 - \frac{\mu}{L}\big)\big(f(\boldsymbol{x}_k) - f^*\big).$$

- Recursion:

$$f(\boldsymbol{x}_{k+1}) - f^* \leq \big(1 - \frac{\mu}{L}\big)^k \big(f(\boldsymbol{x}_0) - f^*\big).$$

Remarks

- The proof also applies to optimal stepsize, since

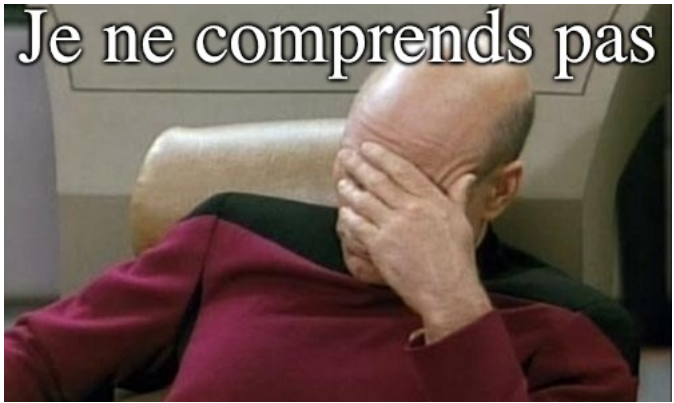
$$f(\boldsymbol{x}_{k+1}) = \min_{\alpha} f\Big(\boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)\Big) \leq f\Big(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)\Big),$$

  where the $\leq$ is by definition of the optimal stepsize.

- Another approach to show the convergence of GD for smooth-f is to impose f is μ-strongly convex. The proof is longer and more complicated.

Table of Contents

The convergence analysis of GD is so hard-core!



The convergence analysis of AGD is even more hard-core.

Convergence rates

- (One of the) Definition:

$$\rho \coloneqq \lim_{k \to \infty} \frac{f(\boldsymbol{x}_{k+1}) - f^*}{f(\boldsymbol{x}_k) - f^*}.$$

  What is it: limit of ratio of successive errors.

- Convergence rate
  - $\rho = 1$: sublinear convergence rate
  - $\rho \in]0, 1[$: linear convergence rate
  - $\rho = 0$: super-linear convergence rate

- Quadratic convergence rate

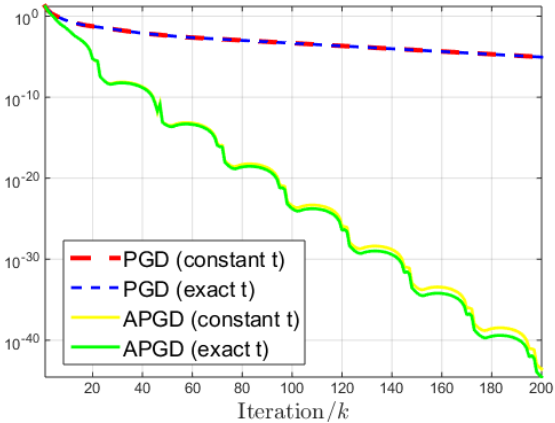
$$\lim_{k \to \infty} \frac{f(\boldsymbol{x}_{k+1}) - f^*}{\Big(f(\boldsymbol{x}_k) - f^*\Big)^2} < M,$$

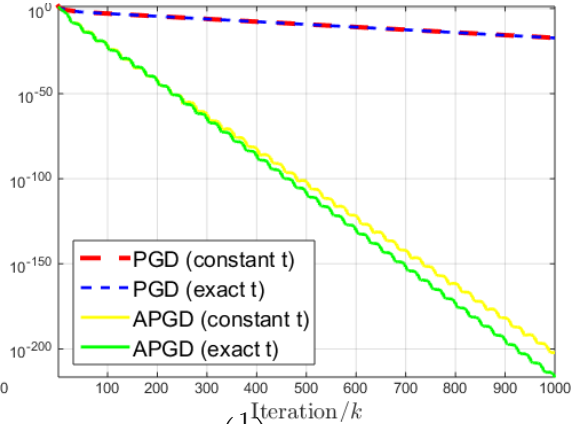
  for some constant $M > 0$.

- Other definition: $\|\boldsymbol{x}_k - \boldsymbol{x}^*\|$, $\|\nabla f(\boldsymbol{x}_k)\|$.

Convergence rate

Recall the observation : different slope $\implies$ different convergence rate





In general, for GD, the objective function decrease in the order of $\mathcal{O}\left(\frac{1}{k}\right)$.

But for Accelerated gradient, it drops in the order of $\mathcal{O}\left(\frac{1}{k^2}\right)$!

And it is *optimal*: you can never do better than $\mathcal{O}\left(\frac{1}{k^2}\right)$, if you only use gradient information!

Convergence rate of accelerated gradient is a order higher

We have the following rate of GD on solving $(\mathcal{P})$ for the best $t = \dfrac{2}{L+\mu}$:

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 \leq \left(1 - \frac{2}{\kappa+1}\right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2.$$

It can be shown, for accelerated gradient (e.g. the heavy ball method), we have

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 \leq \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2.$$

Example: $\kappa = 2$, we have $\dfrac{1}{3^k} = 0.3333^k$ vs 0.1716^k.
For 4 iterations :
GD : 0.33, 0.11, 0.03, 0.01
AGD : 0.17, 0.02, 0.0049, 0.0008

Let's make it even easier to understand

Original problem :

$(\mathcal{P})$: given full rank $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, find $\boldsymbol{x} \in \mathbb{R}^n$ by solving

$$\boldsymbol{x} \coloneqq \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x},$$

where $\boldsymbol{Q} = \boldsymbol{A}^\top \boldsymbol{A}$, $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$, $L = \|\boldsymbol{Q}\|_2$.

To make it easier to understand, set $\boldsymbol{b} = 0$

$(\mathcal{P}')$: given symmetric positive definite $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$, find $\boldsymbol{x} \in \mathbb{R}^n$ by solving

$$\boldsymbol{x} \coloneqq \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}.$$

Again, let $\lambda_{\max}(\boldsymbol{Q}) = L$ and $\lambda_{\min}(\boldsymbol{Q}) = \mu$.

Nesterov's accelerated gradient on $(\mathcal{P}')$

$(\mathcal{P}')$: given sym. p.d. $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$, find $\boldsymbol{x} \in \mathbb{R}^n$ by $\boldsymbol{x} \coloneqq \operatorname*{argmin}_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$.

The update steps in Nesterov's AGD with step size $t = \frac{1}{L}$ and fixed β

$$\boldsymbol{y}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\boldsymbol{Q}\boldsymbol{x}_k \qquad \boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1} + \beta(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)$$

In one line :

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \frac{1}{L}\boldsymbol{Q}\Big(\boldsymbol{x}_k + \beta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})\Big) + \beta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \\
&= \Big((1+\beta)\boldsymbol{x}_k - \beta\boldsymbol{x}_{k-1}\Big) - \frac{1}{L}\boldsymbol{Q}\Big((1+\beta)\boldsymbol{x}_k - \beta\boldsymbol{x}_{k-1}\Big) \\
&= \Big(\boldsymbol{I}_n - \frac{1}{L}\boldsymbol{Q}\Big)\Big((1+\beta)\boldsymbol{x}_k - \beta\boldsymbol{x}_{k-1}\Big) \\
&\overset{\text{e.d.}}{=} \Big(\boldsymbol{V}\boldsymbol{V}^\top - \frac{1}{L}\boldsymbol{V}\Lambda\boldsymbol{V}^\top\Big)\Big((1+\beta)\boldsymbol{x}_k - \beta\boldsymbol{x}_{k-1}\Big) \\
\boldsymbol{V}^\top\boldsymbol{x}_{k+1} &= \Big(\boldsymbol{V}^\top - \frac{1}{L}\Lambda\boldsymbol{V}^\top\Big)\Big((1+\beta)\boldsymbol{x}_k - \beta\boldsymbol{x}_{k-1}\Big) \\
&= \Big(\boldsymbol{I}_n - \frac{1}{L}\Lambda\Big)\Big((1+\beta)\boldsymbol{V}^\top\boldsymbol{x}_k - \beta\boldsymbol{V}^\top\boldsymbol{x}_{k-1}\Big) \\
\boldsymbol{w}_{k+1} &= \Big(\boldsymbol{I}_n - \frac{1}{L}\Lambda\Big)\Big((1+\beta)\boldsymbol{w}_k - \beta\boldsymbol{w}_{k-1}\Big).
\end{aligned}$$

Nesterov's accelerated gradient method on $(\mathcal{P}')$

$$\boldsymbol{w}_{k+1} = \Big(\boldsymbol{I}_n - \frac{1}{L}\Lambda\Big)\Big((1+\beta)\boldsymbol{w}_k - \beta\boldsymbol{w}_{k-1}\Big),$$

$$\begin{bmatrix} w_{k+1}^{[1]} \\ \vdots \\ w_{k+1}^{[n]} \end{bmatrix} = \left(\begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} - \frac{1}{L}\begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}\right)\left((1+\beta)\begin{bmatrix} w_k^{[1]} \\ \vdots \\ w_k^{[n]} \end{bmatrix} - \beta\begin{bmatrix} w_{k-1}^{[1]} \\ \vdots \\ w_{k-1}^{[n]} \end{bmatrix}\right)$$

That is, we have the decoupled element-wise expression

$$\begin{aligned} w_{k+1}^{[i]} &= \big(1 - \frac{\lambda_i}{L}\big)\Big((1+\beta)w_k^{[i]} - \beta w_{k-1}^{[i]}\Big) \\ &= \big(1 - \frac{\lambda_i}{L}\big)(1+\beta)w_k^{[i]} - \big(1 - \frac{\lambda_i}{L}\big)\beta w_{k-1}^{[i]}, \quad i = 1, 2, ..., n \end{aligned}$$

(* Recall AGD reduces to GD if $\beta = 0$, this expression also applies to GD)

Second-order dynamical system

$$w_{k+1}^{[i]} = \big(1 - \frac{\lambda_i}{L}\big)(1+\beta)w_k^{[i]} - \big(1 - \frac{\lambda_i}{L}\big)\beta w_{k-1}^{[i]}, \quad i = 1, 2, ..., n$$

The characteristic equation is

$$r^2 = \big(1 - \frac{\lambda_i}{L}\big)(1+\beta)r - \big(1 - \frac{\lambda_i}{L}\big)\beta.$$

The value $\beta = \beta^*$ that the equation has repeated roots are

$$\beta_{i=1,2}^* = \frac{1 - \sqrt{\lambda_i/L}}{1 + \sqrt{\lambda_i/L}}, \quad r(\beta^*) = 1 - \sqrt{\lambda_i/L}.$$

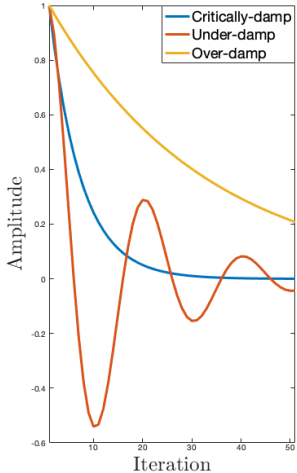
If $\beta \leq \beta^*$ the equation has two distinctive real roots.
If $\beta > \beta^*$, the equation has two complex roots.

The characteristic equation $r^2 = \left(1 - \frac{\lambda_i}{L}\right)(1+\beta)r - \left(1 - \frac{\lambda_i}{L}\right)\beta$.

At the value β^*, the equation has repeated roots

$$\beta^*_{i=1,2} = \frac{1 - \sqrt{\lambda_i/L}}{1 + \sqrt{\lambda_i/L}}, \quad r(\beta^*) = 1 - \sqrt{\lambda_i/L}.$$

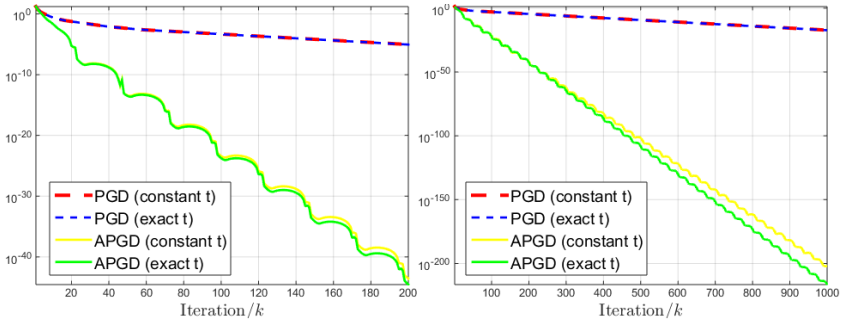
- $\beta = \beta^*$
  - Best β
  - Best amount of momentum
  - System is critically damped
- $\beta \leq \beta^*$
  - β too small
  - Not enough momentum
  - System is over-damped
- $\beta > \beta^*$
  - β too high
  - Too much momentum
  - System is under-damped
  - System is oscillatory



The acceleration comes from damping!!

The periodic ripples is due to $\beta > \beta^*$

When $\beta > \beta^*$, the system is under-damped $\implies$ the periodic ripples



In fact we have

$$w_k^{[i]} = c_1^{[i]} \Big(\beta\big(1 - \frac{\lambda_i}{L}\big)\Big)^{k/2} \cos(k\psi^{[i]} - c_2^{[i]})$$

where $c_{1,2}$ are some unimportant constants and

$$\psi^{[i]} = \cos^{-1}\left(\big(1 - \frac{\lambda_i}{L}\big)\frac{1+\beta}{2}\sqrt{\beta\big(1 - \frac{\lambda_i}{L}\big)}\right) \approx \sqrt{\frac{\lambda_i}{L}}, \quad \psi^{[i]} \approx \sqrt{\frac{\lambda_{\min}}{L}} = \sqrt{\kappa^{-1}}$$

(Detail derivations : take home exercise)

Table of Contents

Nesterov's accelerated gradient

- One of the form of Nesterov's accelerated gradient using Paul Tseng's parameter[1]

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{y}_k - t_k \nabla f(\boldsymbol{y}_k) \\ \boldsymbol{y}_{k+1} &= \boldsymbol{x}_{k+1} + \frac{k}{k+3}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \end{aligned}$$

- Theorem: if f is convex and L-smooth, picking stepsize $t_k = \frac{1}{L}$, the Nesterov's accelerated gradient has the convergence rate as

$$f(\boldsymbol{x}_k) - f^* \leq \frac{c\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{(k+1)^2},$$

  where c is some (unimportant) constant.

- The accelerated gradient actually associated to an ODE

$$\ddot{\boldsymbol{X}} + \frac{3}{\tau}\dot{\boldsymbol{X}} + \nabla f(\boldsymbol{X}) = 0,$$

  which we will show it in the coming slides.

[1] This is NOT the one proposed by Nesterov in 1983.

Derive the ODE ... (1/4)

- Accelerated gradient (constant stepsize)

$$\boldsymbol{x}_{k+1} = \boldsymbol{y}_k - t\nabla f(\boldsymbol{y}_k), \quad \boldsymbol{y}_{k+1} = \boldsymbol{x}_{k+1} + \beta_k(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)$$

so $\boldsymbol{y}_k = \boldsymbol{x}_k + \beta_{k-1}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$, put it to the gradient step gives

$$\begin{array}{rrcl} & \boldsymbol{x}_{k+1} & = & \boldsymbol{x}_k + \beta_{k-1}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) - t\nabla f(\boldsymbol{y}_k) \\ \iff & \boldsymbol{x}_{k+1} - \boldsymbol{x}_k & = & \beta_{k-1}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) - t\nabla f(\boldsymbol{y}_k). \end{array}$$

Why do this: 1) to cancel the $\boldsymbol{y}_k$ and 2) get difference terms such as $\boldsymbol{x}_{k+1} - \boldsymbol{x}_{k-1}$.

- We now have

$$\frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{\sqrt{t}} = \beta_{k-1}\frac{\boldsymbol{x}_k - \boldsymbol{x}_{k-1}}{\sqrt{t}} - \sqrt{t}\nabla f(\boldsymbol{y}_k). \qquad \text{(ode1)}$$

Our goal: derive an ODE from (ode1).
How: link discrete iteration with continuous time.

- Let $k = \dfrac{\tau}{\sqrt{t}}$: discrete iteration $= \dfrac{\text{continuous time}}{\text{stepsize}}$ so

$$\boldsymbol{X}(\tau) = \boldsymbol{X}(k\sqrt{t}) \approx \boldsymbol{x}_k, \quad \boldsymbol{X}(\tau + 1\cdot\sqrt{t}) = \boldsymbol{X}\big((k+1)\sqrt{t}\big) \approx \boldsymbol{x}_{k+1},$$

and the "$\approx$" becomes "$=$" if we take $\lim_{t\to 0}$.
We also have

$$\frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{\sqrt{t}} \approx \frac{\boldsymbol{X}(\tau + \sqrt{t}) - \boldsymbol{X}(\tau)}{\sqrt{t}}, \qquad \frac{\boldsymbol{x}_k - \boldsymbol{x}_{k-1}}{\sqrt{t}} \approx \frac{\boldsymbol{X}(\tau) - \boldsymbol{X}(\tau - \sqrt{t})}{\sqrt{t}}$$

Derive the ODE ... (2/4)

- Recall Taylor's series

$$u(x_0 + \Delta x) = u(x_0) + \Delta x \frac{\partial u}{\partial x}\Big|_{x=x_0} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 u}{\partial^2 x}\Big|_{x=x_0} + o(\Delta x).$$

Let $u = \boldsymbol{X}$, $x_0 = \tau$, $\Delta x = \sqrt{t}$ and $\dfrac{\partial u}{\partial x} = \dfrac{\partial \boldsymbol{X}}{\partial \tau} = \dot{\boldsymbol{X}}$ then

$$\boldsymbol{X}(x_0 + \Delta x) = \boldsymbol{X}(\tau) + \sqrt{t}\dot{\boldsymbol{X}}(\tau) + \frac{t}{2}\ddot{\boldsymbol{X}}(\tau) + o(\sqrt{t}).$$

- So $\dfrac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{\sqrt{t}} \approx \dfrac{\boldsymbol{X}(\tau + \sqrt{t}) - \boldsymbol{X}(\tau)}{\sqrt{t}}$ becomes $\dfrac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{\sqrt{t}} = \dot{\boldsymbol{X}}(\tau) + \dfrac{\sqrt{t}}{2}\ddot{\boldsymbol{X}}(\tau) + o(\sqrt{t})$.
- Similarly, using Taylor's expansion on $u(x_0 - \Delta x)$ with $\Delta x = -\sqrt{t}$, we get

$$\boldsymbol{X}(\boldsymbol{x}_0 - \Delta) = \boldsymbol{X}(\tau) - \sqrt{t}\dot{\boldsymbol{X}}(\tau) + \frac{t}{2}\ddot{\boldsymbol{X}}(\tau) + o(\sqrt{t}),$$

so $\boldsymbol{x}_k - \boldsymbol{x}_{k-1} \approx \boldsymbol{X}(\tau) - \boldsymbol{X}(\tau - \sqrt{t})$ and

$$\frac{\boldsymbol{x}_k - \boldsymbol{x}_{k-1}}{\sqrt{t}} = \dot{\boldsymbol{X}}(\tau) - \frac{\sqrt{t}}{2}\ddot{\boldsymbol{X}}(\tau) + o(\sqrt{t}).$$

Put them into (ode1)

$$\dot{\boldsymbol{X}}(\tau) + \frac{\sqrt{t}}{2}\ddot{\boldsymbol{X}}(\tau) + o(\sqrt{t}) = \beta_{k-1}\left(\dot{\boldsymbol{X}}(\tau) - \frac{\sqrt{t}}{2}\ddot{\boldsymbol{X}}(\tau) + o(\sqrt{t})\right) - \sqrt{t}\nabla f(\boldsymbol{y}_k).$$

Derive the ODE ... (3/4)

- We have

$$\dot{\boldsymbol{X}}(\tau) + \frac{\sqrt{t}}{2}\ddot{\boldsymbol{X}}(\tau) + o(\sqrt{t}) = \beta_{k-1}\left(\dot{\boldsymbol{X}}(\tau) - \frac{\sqrt{t}}{2}\ddot{\boldsymbol{X}}(\tau) + o(\sqrt{t})\right) - \sqrt{t}\nabla f(\boldsymbol{y}_k).$$

  Rearrange

$$\frac{\sqrt{t}}{2}\left(1+\beta_{k-1}\right)\ddot{\boldsymbol{X}}(\tau) + \left(1-\beta_{k-1}\right)\dot{\boldsymbol{X}}(\tau) + \sqrt{t}\nabla f(\boldsymbol{y}_k) + o(\sqrt{t}) = 0.$$

- For $\boldsymbol{y}_k$, as $\boldsymbol{x}_k = \boldsymbol{y}_k$ in the long run, we can take $\boldsymbol{y}_k = \boldsymbol{X}(\tau)$.

$$\frac{\sqrt{t}}{2}\left(1+\beta_{k-1}\right)\ddot{\boldsymbol{X}}(\tau) + \left(1-\beta_{k-1}\right)\dot{\boldsymbol{X}}(\tau) + \sqrt{t}\nabla f\big(\boldsymbol{X}(\tau)\big) + o(\sqrt{t}) = 0.$$

- Hide τ in $\boldsymbol{X}$

$$\frac{\sqrt{t}}{2}\left(1+\beta_{k-1}\right)\ddot{\boldsymbol{X}} + \left(1-\beta_{k-1}\right)\dot{\boldsymbol{X}} + \sqrt{t}\nabla f(\boldsymbol{X}) + o(\sqrt{t}) = 0.$$

- What next: time to plug in β_k.

Derive the ODE ... (4/4)

- If $\beta_k = \dfrac{k}{k+3}$ then $\beta_{k-1} = \dfrac{k-1}{k+2}$ and

$$\frac{k-1}{k+2} = 1 - \frac{-3}{k+2} \overset{k \gg 2}{\approx} 1 - \frac{3}{k} \overset{k=\frac{\tau}{\sqrt{t}}}{=} 1 - \frac{3\sqrt{t}}{\tau}.$$

  Then $1 + \beta_{k-1} = 2 - \dfrac{3\sqrt{t}}{\tau}$, $1 - \beta_{k-1} = \dfrac{3\sqrt{t}}{\tau}$

$$\frac{\sqrt{t}}{2}\left(2 - \frac{3\sqrt{t}}{\tau}\right)\ddot{\boldsymbol{X}} + \frac{3\sqrt{t}}{\tau}\dot{\boldsymbol{X}} + \sqrt{t}\nabla f(\boldsymbol{X}) + o(\sqrt{t}) = 0.$$

- Divide the whole equation by $\sqrt{t}$

$$\left(1 - \frac{3\sqrt{t}}{2\tau}\right)\ddot{\boldsymbol{X}} + \frac{3}{\tau}\dot{\boldsymbol{X}} + \nabla f(\boldsymbol{X}) + o(\sqrt{t}) = 0.$$

  Take $\lim_{t \to 0}$ gives $\ddot{\boldsymbol{X}} + \dfrac{3}{\tau}\dot{\boldsymbol{X}} + \nabla f(\boldsymbol{X}) = 0.$

Review of the derivation

- Step-1. Nesterov's accelerated gradient gives

$$\frac{\boldsymbol{x}_{k+1}-\boldsymbol{x}_k}{\sqrt{t}}=\beta_{k-1}\frac{\boldsymbol{x}_k-\boldsymbol{x}_{k-1}}{\sqrt{t}}-\sqrt{t}\nabla f(\boldsymbol{y}_k).$$

- Step-2. Apply Taylor's approximation

$$\frac{\sqrt{t}}{2}\left(1+\beta_{k-1}\right)\ddot{\boldsymbol{X}}+\left(1-\beta_{k-1}\right)\dot{\boldsymbol{X}}+\sqrt{t}\nabla f(\boldsymbol{X})+o(\sqrt{t})=0.$$

or

$$\frac{1+\beta_{k-1}}{2}\ddot{\boldsymbol{X}}+\frac{1-\beta_{k-1}}{\sqrt{t}}\dot{\boldsymbol{X}}+\nabla f(\boldsymbol{X})+o(\sqrt{t})=0.$$

- Step-3. Plugin β_{k-1} and take limit $t\to 0$

$$\ddot{\boldsymbol{X}}+\frac{3}{\tau}\dot{\boldsymbol{X}}+\nabla f(\boldsymbol{X})=0.$$

About the $\beta_k = \dfrac{3}{k+3}$

- It is instrumental such that taking limit $t \to 0$ will not cancel $\ddot{\boldsymbol{X}}$ or blow up the ODE.

- Selection of $\beta_k = \dfrac{3}{k+3}$ is legit because it satisfies

$$\frac{1-\beta_{k+1}}{\beta_{k+1}^2} \leq \frac{1}{\beta_k^2}.$$

Convergence rate using ODE

- ODE
$$\ddot{\boldsymbol{X}}(\tau) + \frac{3}{\tau}\dot{\boldsymbol{X}}(\tau) + \nabla f\Big(\boldsymbol{X}(\tau)\Big) = 0.$$

- Standard ODE theory gives
$$f\Big(\boldsymbol{X}(\tau)\Big) - f^* \leq \mathcal{O}\Big(\frac{1}{\tau^2}\Big).$$

- As ODE $\iff$ Nesterov's accelerated gradient, so this (partially) explains
$$f(\boldsymbol{x}_k) - f^* \leq \frac{c\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{(k+1)^2}.$$

After stories / What is not discussed

- **The proof of convergence of AGD for general problem**
  `https://angms.science/doc/CVX/CVX_NAGD.pdf`
  `https://angms.science/doc/CVX/CVX_NAGDalpha.pdf`
  `https://angms.science/doc/CVX/fista_convergence.pdf`
- **Is AGD really always good?**
  AGD is not suitable when gradient has noise.
  In other words, AGD scarifies robustness for speed.
- **Extension to other gradient scheme**
  Accelerated Stochastic Gradient Descent
  Accelerated Primal-Dual Gradient
- **Extension to other model**
  What about $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_1$: non-smooth optimization.
  What about f non-convex: current research topics.
- **What actually leads to the acceleration?**
  ODE, Variational perspective, principle of least action, damped Lagrangian ... in the most recent research within last 7 years!

C'est bon bon. End of lecture