Lecture 7

7.1 Complexity of The Heavy Ball Method

7.1.1 Algorithm

We are solving the unconstrained minimization problem,

$$\min_{x \in \mathbb{R}^n} f(x), \tag{7.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and it has a Lipschitz continuous gradient.

In the last lecture, we have discussed the following optimization algorithm, called the *heavy ball method* or the gradient method with *momentum*, as applied to (7.1). We review it in the following algorithmic form. We assume that the Lipschitz constant $L > 0$ is known and we also fix the number of iterations $K \geq 1$ of the method.

Algorithm 7.1: *Heavy Ball Method.*

```
Initialization: $x_0 \in \mathbb{R}^n$, Lipschitz constant $L > 0$, momentum parameter $0 \leq \beta \leq 1$,
number of iterations $K \geq 1$. Set $s_0 = 0 \in \mathbb{R}^n$.
For $k = 0 \ldots K-1$ iterate:
    1. Compute new gradient and aggregate: $s_{k+1} := \beta s_k + \nabla f(x_k)$
    2. Perform a step: $x_{k+1} := x_k - \frac{1}{L} s_{k+1}$
Return a point $\bar{x}$ with the best desired accuracy measure.
```

The parameter $0 \leq \beta \leq 1$ corresponds to how fast we forget the history. Note that in general we do not have monotonicity in the gradient norms or in function values. Therefore, we have to decide which point to return, depending on the analysis of the method.

The gradient method. The classic gradient method is covered by setting $\beta := 0$ (no history) in Algorithm 7.1. In the last lecture, we have established few rates of convergence for the gradient method. We proved the rate in terms of the functional residual, for $k \geq 0$:

$$f(x_k) - f^\star \;\; \leq \;\; \tfrac{2LR^2}{k+4}, \qquad R \;\; := \;\; \|x_0 - x^\star\|. \tag{7.2}$$

In this case, we can use the last point $\bar{x} := x_k$ for the result of the algorithm. Using this rate, se also have showed the following convergence in terms of the gradient norm, for $k \geq 1$:

$$\min_{0 \leq i \leq k-1} \|\nabla f(x_i)\| \;\; \leq \;\; \tfrac{4LR}{k}. \tag{7.3}$$

In this case, the result point $\bar{x}$ is the one among candidates $\{x_0, \dots, x_{k-1}\}$ with the smallest gradient norm.

Then, we have discussed that for *strongly convex* functions, which satisfy the uniform bound for the eigenvalues of the Hessian,

$$\mu I \quad \preceq \quad \nabla^2 f(x) \quad \preceq \quad LI, \qquad x \in \mathbb{R}^n,$$

All accuracy measures become equivalent, and we have proved the *linear rate*:

$$f(x_k) - f^\star \quad \leq \quad \exp\Bigl(-k\tfrac{\mu}{L}\Bigr)(f(x_0) - f^\star).$$

Therefore, the gradient method needs $K = \frac{L}{\mu} \ln \frac{f(x_0) - f^\star}{\varepsilon}$ to solve the problem. The question is whether this dependence on the condition number L/μ is *the best we can achieve*? And the answer is *no*.

The heavy ball with restarts. In the last lecture, we also analyzed the case $\boxed{\beta := 1}$ of Algorithm 7.1. on $\underline{\textit{quadratic functions}}$. This is much more restricted class of functions than all convex smooth functions.

We have proved the following result (Theorem 6.2.4. from Lecture 6):

- Form *average points*, $\bar{x}_k := \frac{1}{k} \sum\limits_{i=1}^{k-1} x_i$, for $1 \leq k \leq K$.
- Then, we have for the *smallest gradient norm* $g_k := \min\{\|\nabla f(\bar{x}_1)\|, \dots, \|\nabla f(\bar{x}_k)\|\}$ we have the rate
$$g_k^2 \quad \leq \quad \tfrac{12L(f(x_0) - f^\star)}{k(k+1)} \tag{7.4}$$
for the heavy ball method with $\beta := 1$ on convex quadratic functions.

Note that (7.4) gives us $g_k = O(1/k)$, which is similar to the rate of the gradient method (7.3). However, the presence of $f(x_0) - f^\star$ in the right hand side of (7.4) is very important, as it allows to *restart our method*, a popular technique in optimization.

Assume that our quadratic function, $f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$ is *strongly convex*, that is $\mu = \lambda_{\min}(A) > 0$. Then, by strong convexity, we have an inequality (Proposition 6.1.6. from Lecture 6):

$$f(x) - f^\star \quad \leq \quad \tfrac{1}{2\mu}\|\nabla f(x)\|^2, \qquad x \in \mathbb{R}^n.$$

Therefore, denoting by $\bar{x}_K^\star$ the point with the smallest gradient norm among $\{\bar{x}_1, \dots, \bar{x}_K\}$, we obtain from (7.4):

$$f(\bar{x}_K^\star) - f^\star \quad \leq \quad \tfrac{6L}{\mu K^2} \cdot (f(x_0) - f^\star). \tag{7.5}$$

Let us choose

$$K \quad := \quad \sqrt{\tfrac{12L}{\mu}}. \tag{7.6}$$

Substituting this value into (7.5), we get

$$f(\bar{x}_K^\star) - f^\star \quad \leq \quad \tfrac{1}{2}(f(x_0) - f^\star).$$

In other words, performing (7.6) iterations of the heavy ball method, we halve the functional residual, which is a very good progress: to get from an arbitrary functional residual $f(x_0) - f^\star$ to a point $f(y_T) - f^\star \leq \varepsilon$ we only need $T := \log_2 \frac{f(x_0) - f^\star}{\varepsilon}$ restarts!

Theorem 7.1.1. *The total complexity of the heavy ball method with restarts is*

$$\sqrt{\tfrac{12L}{\mu}}\log_2\tfrac{f(x_0)-f^\star}{\varepsilon} \tag{7.7}$$

first-order oracle calls (matrix-vector products) to minimize a strongly convex quadratic function.

We see that the condition number $\sqrt{\frac{L}{\mu}}$ is much better in (7.7) than that one $\frac{L}{\mu}$ in the gradient method, as typically $L \gg \mu$.

It appears that this complexity is the *optimal one*, i.e. it is impossible to develop a faster first-order method that has a better dependence on the condition number.

A couple of final remarks on the heavy ball method. There are the following possible choices of the parameter β:

- $\beta := 0$ corresponds to the gradient descent;
- $\beta := 1$ which we have analyzed; we needed to do restarts every $K \approx \sqrt{\frac{L}{\mu}}$ iterations in order to achieve the optimal complexity;
- A different analysis can be applied, which suggests to choose $\beta \approx 1 - \sqrt{\frac{\mu}{L}}$. Then, no restarts are needed;
- In practice: a standard choice $\beta \approx 0.99$;

Explicit analysis. To see other options for choosing $\beta \in (0,1)$, let us consider one iteration of the heavy ball method,

$$x_{k+1} \;=\; x_k - \alpha\nabla f(x_k) + \beta(x_k - x_{k-1}), \tag{7.8}$$

on the quadratic function. Thus, the gradient is an affine mapping that can be written as $\nabla f(x) = A(x - x^\star)$. Substituting this formula into (7.8), and subtracting $x^\star$ from both sides, we obtain the recursion:

$$r_{k+1} \;=\; r_k - \alpha A r_k + \beta(r_k - r_{k-1}), \tag{7.9}$$

where $r_k := x_k - x^\star \in \mathbb{R}^n$. This recursion can be seen as a *linear dynamical system*, and our goal in choosing $\alpha > 0$ and $\beta \in (0,1)$ is to ensure the fastest decrease $r_k \to 0$. Dynamical system (7.9) can be written in the matrix form:

$$\begin{pmatrix} r_{k+1} \\ r_k \end{pmatrix} \;=\; C\begin{pmatrix} r_k \\ r_{k-1} \end{pmatrix} \tag{7.10}$$

where

$$C \;=\; \begin{pmatrix} (1+\beta)I - \alpha A & -\beta I \\ I & 0 \end{pmatrix} \;\in\; \mathbb{R}^{2n\times 2n}$$

is a fixed matrix that depends on our parameters. It is possible to show that choosing $\alpha := \frac{1}{L}$ and $\beta := 1 - \sqrt{\mu/L}$, the spectral radius $\rho(\cdot)$ of a non-symmetric matrix C is separated from 1:

$$\rho(C) \;\leq\; 1 - \tfrac{1}{2}\sqrt{\mu/L},$$

and it can be used that the linear dynamical system (7.10) converges to 0 with the desirable linear rate (see also [Pol87]).

Note that complexity (7.7) of the heavy ball method holds only for *quadratic functions* and our analysis was quite specific for them; there are plenty of other first-order methods that achieve the same optimal complexity on quadratic functions. The best algorithm among them for unconstrained quadratic minimization is the *conjugate gradient method* (see [Nem95])

It is not clear whether it is possible to generalized the heavy ball method upon quadratic functions, and it remains to be an open problem. A recent result [GTD25], which utilizes a computer-aided analysis, demonstrates that it is impossible to achieve such complexity for the heavy ball method on the general class of strongly convex smooth problems.

Next week, we will study another accelerated algorithm, called *Nesterov's fast gradient method* [Nes18], that achieves this goal.

7.2 Lower Bounds for Smooth Convex Optimization

Our goal is to study the lower complexity bounds for our problem class. We focus on *convex smooth functions* (not necessary strongly-convex). We saw that the rate of the gradient method was

$$f(x_k) - f^\star \quad \leq \quad O\Bigl(\tfrac{LR^2}{k}\Bigr), \qquad k \geq 1, \tag{7.11}$$

and due to our results of quadratic functions, we might expect that this it *not optimal.* Indeed, we can prove the following lower bound.

Theorem 7.2.1. *Let $L > 0$ be fixed. For any first-order optimization algorithm running for $k \geq 1$ iterations, there is a convex function $f : \mathbb{R}^n \to \mathbb{R}$ with $n \geq 2k+1$ with Lipschitz continuous gradient with constant L such that*

$$f(x_K) - f^\star \quad \geq \quad \tfrac{3LR^2}{16(k+1)^2}. \tag{7.12}$$

We see that (7.12) does not match (7.11) which might indicate nothing, it could be that any of these too bounds is not tight enough. However, as we will see the lower bound in (7.12) can be matched up to a numerical constant, showing that the gradient descent is not optimal on our problem class.

- This bounds holds for *high-dimensional problems*, which is the case for modern applications;
- Even if the dimension is small, it tells us something about behavior of the algorithm in the early stage.

7.2.1 Lower Bound for the Linear Span Methods

Let us simplify our goal as much as possible. First, we can assume that $\boxed{x_0 := 0}$, so we always start a method from the origin. If it is not the case, we can always apply the same method to a shifted function $\varphi(x) := f(x - x_0)$.

Then, we can also assume that $L :=$ const (why)?

It is easier and more instructive to consider a restricted class of first-order algorithms, that generate the next iterate within a *linear combination of the gradients* seen so far:

$$x_{k+1} \quad \in \quad \mathcal{L}_{k+1} \quad := \quad \mathrm{span}\Bigl\{ \nabla f(x_0), \ldots, \nabla f(x_k) \Bigr\}. \tag{7.13}$$

Note that both the gradient method, and the heavy ball method satisfy this assumption, as well as most of the standard optimization algorithms. The chain of linear spaces

$$\mathcal{L}_0 \quad \subseteq \quad \mathcal{L}_1 \quad \subseteq \quad \mathcal{L}_2 \quad \subseteq \quad \ldots, \tag{7.14}$$

is called *Krylov subspaces* of the method. We have,

$$\begin{aligned} \mathcal{L}_0 &:= \{0\} \\ \mathcal{L}_1 &:= \operatorname{span}\{\nabla f(x_0)\} \\ \mathcal{L}_2 &:= \operatorname{span}\{\nabla f(x_0), \nabla f(x_1)\}, \\ &\dots \ \text{etc.} \end{aligned}$$

In fact, even if condition (7.13) is not satisfied, it is possible to modify the construction of the lower bound, using a *resisting oracle*, to ensure that (7.13) holds for any first-order method.

But for now, under assumption (7.13), we can consider one objective function *for any algorithm*, which is the following quadratic function [Nes18], parameterized by an integer $k \geq 0$:

$$\begin{aligned} f_k(x) &:= \tfrac{1}{2}\Big[\sum_{i=1}^{k-1} (x^{(i)} - x^{(i+1)})^2 + \sum_{i=k}^{n} (x^{(i)})^2 \Big] - \langle b, x\rangle \\ &= \tfrac{1}{2}\Big[(x^{(1)} - x^{(2)})^2 + \ldots + (x^{(k-1)} - x^{(k)})^2 + (x^{(k)})^2 + \ldots + (x^{(n)})^2 \Big] - \langle b, x\rangle \\ &= \tfrac{1}{2}\|C_k x\|_2^2 - \langle b, x\rangle \;=\; \tfrac{1}{2}\langle C_k^\top C_k x, x\rangle - \langle b, x\rangle \end{aligned} \tag{7.15}$$

where the matrix $C_k \in \mathbb{R}^{n\times n}$ has the following block structure: $C_k = \begin{pmatrix} D_k & 0 \\ 0 & I_{n-k} \end{pmatrix}$,

$$D_k = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots \\ 0 & 1 & -1 & 0 & \dots \\ 0 & 0 & 1 & -1 & \dots \\ \dots & & & & \\ 0 & & \dots & & 1 \end{pmatrix} \in \mathbb{R}^{k\times k}.$$

Therefore, our objective can be written in the standard form

$$f_k(x) = \tfrac{1}{2}\langle A_k x, x\rangle - \langle b, x\rangle,$$

where $A_k := C_k^\top C_k \succeq 0$. An explicit formula for the matrix $A_k \in \mathbb{R}^{n\times n}$ is:

$$A_k = \begin{pmatrix} \Lambda_k & 0 \\ 0 & I_{n-k} \end{pmatrix},$$

where

$$\Lambda_k = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 \\ \dots & & & & & \\ 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{pmatrix}. \tag{7.16}$$

What is so specific about this function?

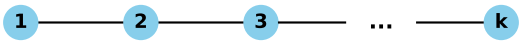

Figure 7.1: Chain graph with k nodes.

- It is simple enough that we can analyze it directly.
- The matrix A_k is *tridiagonal.*
- The matrix $\bar{\Lambda}_k := \Lambda_k - e_k e_k^\top$ is the Laplacian matrix of the *chain graph* (Fig. 7.1).

From (7.15) it is easy to see that $A_k \preceq 4I$. Indeed, for any $h \in \mathbb{R}^n$, we have

$$\begin{aligned} \langle \nabla^2 f_k(x)h, h\rangle &= \Big[\sum_{i=1}^{k-1} (h^{(i)} - h^{(i+1)})^2 + \sum_{i=k}^{n} (h^{(i)})^2 \Big] \\ &\leq \Big[\sum_{i=1}^{k-1} 2 \cdot (h^{(i)})^2 + 2 \cdot (h^{(i+1)})^2 + \sum_{i=k}^{n} (h^{(i)})^2 \Big] \\ &\leq 4\|h\|_2^2. \end{aligned}$$

Therefore, $f_k(\cdot)$ belongs to our class: $0 \preceq \nabla^2 f_k(\cdot) \preceq 4I$. For the linear term b, we use the first basis vector:

$$\boxed{b \;:=\; e_1 \;\in\; \mathbb{R}^n.}$$

Intuitively, this is "information" that we put at the first node of the graph. Then, each iteration of the gradient-based method can "propagate" this information maximum one node to the right, and to reach the end of the chain, we have to peform k iterations.

Let us compute the optimum $x_k^\star = \arg\min_{x \in \mathbb{R}^n} f_k(x)$. We differentiate (7.15) and see that a solution should satisfy the following linear system of equations:

$$\begin{aligned} x^{(1)} - x^{(2)} - 1 &= 0, \qquad i = 1 \;\; \text{(initial condition)} \\ 2x^{(i)} - x^{(i-1)} - x^{(i+1)} &= 0, \qquad \text{for} \;\; 2 \leq i \leq k, \\ 2x^{(k)} - x^{(k-1)} &= 0, \qquad i = k, \\ x^{(i)} &= 0, \qquad \text{for} \;\; k < i \leq n. \end{aligned}$$

We obtain that the following vector satisfies this equations:

$$\begin{aligned} (x_k^\star)^{(1)} &= k \\ (x_k^\star)^{(2)} &= k - 1 \\ &\ldots \\ (x_k^\star)^{(k)} &= 1 \\ (x_k^\star)^{(i)} &= 0, \qquad \text{for} \;\; i \geq k + 1. \end{aligned}$$

The optimum of the quadratic function f_k is given by

$$f_k^\star \quad = \quad f_k(x_k^\star) \quad = \quad \tfrac{1}{2}\langle Ax_k^\star, x_k^\star\rangle - \langle b, x_k^\star\rangle \quad = \quad -\tfrac{1}{2}\langle b, x_k^\star\rangle \quad = \quad -\tfrac{1}{2}(x_k^\star)^{(1)} \quad = \quad -\tfrac{k}{2}.$$

Let us also estimate the distance from the origin $x_0 = 0$ to the solution $x_k^\star$,

$$R_k^2 \quad := \quad \|x_0 - x_k^\star\|^2 \quad = \quad \sum_{i=1}^{n} \bigl[(x_k^\star)^{(i)}\bigr]^2 \quad = \quad \sum_{i=1}^{k} i^2 \quad = \quad \tfrac{k(k+1)(2k+1)}{6} \quad \leq \quad \tfrac{(k+1)^3}{3}.$$

This function has a very simple structure of the Krylov subspaces (7.14). Namely, we can control the number of non-zeros in iterations of the algorithm: after $k \geq 0$ iterations, the number of nonzeros in x_k is no more than k.

Proposition 7.2.2. *Assume that iterates $\{x_i\}_{i\geq 0}$ satisfy our assumption* (7.13)*. Then,*

$$\mathcal{L}_k \quad \subseteq \quad \mathbb{R}^{n,k} \quad := \quad \{x \in \mathbb{R}^n \;:\; x^{(i)} = 0 \;\; for \;\; i > k\}. \tag{7.17}$$

Proof. Let us prove (7.17) by induction.

- By our assumption $x_0 := 0$ and $\mathcal{L}_0 := \{0\} \quad = \quad \mathbb{R}^{n,0}$.
- Then, $\nabla f_k(x_0) = A0 - b = -e_1$ and we have that $x_1 \in \operatorname{span}\{\nabla f(x_0)\} = \operatorname{span}\{e_1\} \in \mathbb{R}^{n,1}$.
- Assume that $x_k \in \mathcal{L}_k \subseteq \mathbb{R}^{n,k}$. $\nabla f(x_k) = A_k x_k - e_1$. Since the matrix is tridiagonal, $\nabla f(x_k) \in \mathbb{R}^{n,k+1}$, which ensures that $\mathcal{L}_{k+1} \subseteq \mathbb{R}^{n,k+1}$.

□

Using this property, we prove that functions $f_k(\cdot)$ and $f_{k+p}(\cdot)$ are *informationally indistinguishable* for the method, where $p \geq 0$. This follows directly from the structure (7.15) of the objective.

Proposition 7.2.3. *Consider $f_k(x)$ and $f_{k+p}(x)$ for some $p \geq 0$. Then,*

$$f_k(x) \equiv f_{k+p}(x), \qquad x \in \mathbb{R}^{n,k}.$$

So these functions are informationally indistinguishable on $\mathbb{R}^{n,k}$.

Corollary 7.2.4. *For the output of the algorithm, $x_k \in \mathcal{L}_k \subseteq \mathbb{R}^{n,k}$, we have*

$$f(x_k) \quad = \quad f_{2k+1}(x_k) \quad = \quad f_k(x_k) \quad \geq \quad f_k^\star \quad = \quad -\tfrac{k}{2}.$$

At the same time, $f^\star = f_{2k+1}^\star = -\frac{2k+1}{2}$. Hence,

$$f(x_k) - f^\star \quad \geq \quad \tfrac{2k+1}{2} - \tfrac{k}{2} \quad = \quad \tfrac{k+1}{2}.$$

and

$$R \quad = \quad \|x_0 - x^\star\|^2 \quad = \quad \|x^\star\|^2 \quad \leq \quad \tfrac{2^3(k+1)^3}{3}.$$

Thus,

$$\tfrac{f(x_k)-f^\star}{\|x_0-x^\star\|^2} \quad \geq \quad \tfrac{k+1}{2R^2} \quad \geq \quad \tfrac{3(k+1)}{4(k+1)^3} \quad = \quad \tfrac{3}{4(k+1)^2}.$$

And this is the required bound (7.12) for $L = 4$.

Now, multiplying f by $\frac{L}{4}$ we obtain that

$$\tfrac{f(x_k)-f^\star}{\|x_0-x^\star\|^2} \quad \geq \quad \tfrac{3L}{16(k+1)^2},$$

which finishes the proof.

Therefore, the complexity K of any first-order method from our class to obtain $f(\bar{x}) - f^\star \leq \varepsilon$ is bounded as

$$K + 1 \quad \geq \quad \tfrac{1}{4}\sqrt{\tfrac{3LR^2}{\varepsilon}}. \tag{7.18}$$

7.2.2 General Case

What if an algorithm does not satisfy our assumption, $x_k \in \mathcal{L}_k$? It is possible to generalize our construction by performing a resisting oracle strategy, which rotates objective in a way that each iteration belongs to the Krylov subspace, and so $x_k \in \mathcal{L}_k$ is satisfied (see [Nem95]).

7.2.3 Strongly Convex Minimization

Using a similar reasoning, we can directly show a lower bound for the class $\mu I \preceq \nabla^2 f(x) \preceq LI$. However, instead let us study the following regularization technique, which is important on its own.

Assume that we want to minimize a convex function $f : \mathbb{R}^n \to \mathbb{R}$. However, we only have an algorithm that can minimize strongly convex functions. Then we can consider the regularized problem:

$$f_\mu(x) \quad = \quad f(x) + \tfrac{\mu}{2}\|x\|^2.$$

This function will be strongly convex with parameter $\mu > 0$. We can also assume that $\mu \leq L$, and thus f_μ will have a Lipschitz gradient with constant $2L$. Assume that we found an approximate solution to the regularized problem:

$$f_\mu(\bar{x}) - f_\mu^\star \quad \leq \quad \delta, \tag{7.19}$$

for some $\delta > 0$. Can we use it to obtain a good solution for the initial problem?

We notice that

- $f_\mu(\bar{x}) \geq f(\bar{x})$.
- At the same time,

$$f_\mu(\bar{x}) \quad \overset{(7.19)}{\leq} \quad f_\mu^\star + \delta \quad \leq \quad f_\mu(y) + \delta \quad = \quad f(y) + \tfrac{\mu}{2}\|y\|^2 + \delta,$$

  for any $y \in \mathbb{R}^n$. Let us substitute $y := x^\star$ (the solution to the original problem). We get

$$f(x^\star) \quad \geq \quad f_\mu(\bar{x}) - \tfrac{\mu}{2}\|x^\star\|^2 - \delta. \tag{7.20}$$

Hence,

$$f(\bar{x}) - f^\star \quad = \quad f(\bar{x}) - f(x^\star) \quad \leq \quad f_\mu(\bar{x}) - f(x^\star) \quad \overset{(7.20)}{\leq} \quad \tfrac{\mu}{2}\|x^\star\|^2 + \delta.$$

Therefore, by choosing $\delta := \frac{\varepsilon}{2}$ and $\mu := \frac{\varepsilon}{\|x^\star\|^2}$ we obtain the desired accuracy for the initial problem:

$$f(\bar{x}) - f^\star \quad \leq \quad \varepsilon.$$

Now, assume that the complexity of solving a strongly convex smooth function K with parameters μ and L is bounded as

$$K \quad < \quad c \cdot \sqrt{\tfrac{L}{\mu}} - 1,$$

with $c = \frac{\sqrt{3}}{4}$. This would mean that the complexity of solving a smooth convex function is

$$K \quad < \quad \sqrt{\tfrac{2L\|x^\star\|^2}{\varepsilon}} - 1,$$

which contradicts (7.18). Hence, we obtain the following.

Proposition 7.2.5. *The complexity of any first-order method minimizing smooth convex functions is lower bounded as*

$$K \quad \geq \quad c \cdot \sqrt{\tfrac{L}{\mu}} - 1.$$

This is the tight bound, up to a logarithmic term and a numerical constant. It is matched by the heavy ball method on quadratic functions (7.7).

Literature

[GTD25] Baptiste Goujaud, Adrien Taylor, and Aymeric Dieuleveut. Provable non-accelerations of the heavy-ball method. *Mathematical Programming*, pages 1–59, 2025.

[Nem95] Arkadi Nemirovski. *Information-based complexity of convex programming*. Lecture notes, 1995.

[Nes18] Yurii Nesterov. *Lectures on convex optimization*. Springer, 2018.

[Pol87] Boris T Polyak. Introduction to optimization. 1987.