Cornell ORIE 6365

Spring 2026

Lecture Notes on

Continuous Optimization: Algorithms and Complexity

Nikita Doikov

May 30, 2026

Preface

These lecture notes were developed for ORIE 6365 — Continuous Optimization: Algorithms and Complexity, taught at Cornell University in Spring 2026.

This is a graduate-level course on the theory and algorithms of continuous optimization. It prepares students for research in optimization theory and for developing advanced methods for applications in operations research, machine learning, and related domains. The main emphasis is on understanding different classes of optimization problems and their theoretical limitations. We rigorously study convergence rates and lower bounds for first-order and second-order optimization methods on both convex and non-convex problems, establishing the range of their applicability.

The only prerequisite is a good background in basic linear algebra and multivariate calculus; knowledge of probability theory is only needed for certain topics. Due to the lack of space, we do not cover as many applications as the field of optimization has to offer. Therefore, some previous exposure to introductory optimization (e.g., ORIE 6300) is highly recommended to motivate the theory.

The structure of this course is inspired by and largely based on the following sources, which we recommend for primary reading:

1. *Yurii E. Nesterov, Lectures on Convex Optimization, Springer, 2018.*
2. *Arkadi S. Nemirovski, Information-Based Complexity of Convex Programming (Lecture Notes).*

Chapters are extended by a few exercises that sometimes cover additional topics.

Contents

1 Optimization Problems and Complexity **3**
- 1.1 Introduction . . . 3
- 1.2 Complexity of Optimization Problems . . . 8
- 1.3 Grid Search Algorithm . . . 11
- 1.4 Lower Bound for Global Optimization . . . 12

2 Finding Stationary Points **17**
- 2.1 Using Local Information: Differentiable Functions . . . 17
- 2.2 Predictable Function Behavior: Smoothness . . . 21
- 2.3 Gradient Method for General Norms . . . 24
- 2.4 Adaptive Search for Gradient Method . . . 27
- 2.5 Stochastic Gradient Method . . . 29
- 2.6 Exercises . . . 32

3 Minimizing Differentiable Convex Functions **36**
- 3.1 Convex Functions . . . 37
- 3.2 Convergence Rates of Gradient Method . . . 42
- 3.3 Polyak's Heavy Ball Method . . . 48
- 3.4 Lower Bound for Smooth Convex Optimization . . . 53
- 3.5 Nesterov's Fast Gradient Method . . . 59
- 3.6 Applications: Machine Learning . . . 65
- 3.7 Fully Composite Problems . . . 69
- 3.8 Exercises . . . 73

4 Geometry of Non-Smooth Convex Minimization **77**
- 4.1 Convexity and Separation . . . 78
- 4.2 Binary Search Algorithm . . . 83
- 4.3 Ellipsoid Method . . . 85
- 4.4 Subgradient Method: Normalized Stepsizes . . . 92
- 4.5 Lower Bound for Non-Smooth Convex Optimization . . . 100
- 4.6 Adaptive Stepsizes for Stochastic Methods . . . 105
- 4.7 Smooth Stochastic Optimization II . . . 110
- 4.8 Mirror Descent and Accuracy Certificates . . . 113
- 4.9 Exercises . . . 122

5 Second-Order Methods **125**
- 5.1 Introduction . . . 125
- 5.2 Self-Concordant Functions and Local Convergence of Newton's Method . . . 129
- 5.3 Interior-Point Method . . . 139
- 5.4 Cubic Regularization of Newton's Method . . . 147
- 5.5 Quasi-Self-Concordant Functions and Gradient Regularization . . . 156
- 5.6 Contracting-Point Acceleration . . . 164
- 5.7 Exercises . . . 168

1. Optimization Problems and Complexity

In the first introductory part, we formally define the notion of *complexity* of an *optimization algorithm*, which we use throughout the course. To illustrate this concept, we use a classical example of the grid search algorithm, and show that it matches the lower complexity bound for the problem class of global optimization.

1.1 Introduction . . . 3
1.1.1 Basic Classification of Optimization Problems . . . 4
1.1.2 Types of Solutions . . . 5
1.1.3 Why Continuous? . . . 5
1.1.4 Examples . . . 6
1.2 Complexity of Optimization Problems . . . 8
1.2.1 Notion of Problem Class . . . 8
1.2.2 Oracles . . . 8
1.2.3 What is Optimization Algorithm? . . . 9
1.2.4 Stopping Conditions . . . 10
1.2.5 Definition of Complexity . . . 10
1.3 Grid Search Algorithm . . . 11
1.3.1 Global Optimization . . . 11
1.3.2 Grid Search . . . 11
1.4 Lower Bound for Global Optimization . . . 12
1.4.1 Packing Problem . . . 13
1.4.2 Resisting Oracle . . . 14
1.4.3 Lower Bound . . . 15

1.1 Introduction

In this course, we study efficient algorithms for solving optimization problems in the following general form:

$$\min_{x \in Q} f(x), \tag{1.1}$$

where $Q \subseteq \mathbb{R}^n$ is a given *feasible set* or *constraint set*, $x \in Q$ is called the vector of *optimization parameters* or the *decision variables*, and $f : Q \to \mathbb{R}$ is a target *objective function*, which we always assume to be continuous. The goal is to find a best possible point $x^\star \in Q$ that achieves the minimum (1.1).

Modern models that are used in practice typically include a large number of parameters, which means that the dimension n of the variable space is large (several thousands, millions, or even trillions of parameters). Therefore, using computers is the only possible way to find a solution. We will study the design of optimization algorithms, along with analysis of their *performance guarantees*, i.e. how fast the algorithm converges, or how many basic operations, such as gradient computations or matrix-vector products, are needed to obtain a desirable solution.

We also note that since $\max_x f(x) = -\min_x[-f(x)]$, we can always focus only on studying minimization problems, as maximization is trivially covered by putting minus in front of the objective.

1.1.1 Basic Classification of Optimization Problems

Depending on the structure of the objective and the constraint set in (1.1), we use the following standard classification of optimization problems:

Unconstrained optimization. This is the case when $Q \equiv \mathbb{R}^n$, i.e. there are no any constraints in the problem, and the decision variables can take any values in a finite-dimensional real vector space $\mathbb{R}^n$:

$$\min_{x \in \mathbb{R}^n} f(x). \tag{1.2}$$

Unconstrained optimization problems are commonly considered to be *simpler* than constrained ones, and we devote a significant part of this course to studying the unconstrained situation. However, as we will see, often the ideas from unconstrained optimization can be successfully applied to more general constrained problems, and these use cases will be thoroughly studied.

We call the problems of the form (1.2) as

- **Smooth optimization**, when the function $f : \mathbb{R}^n \to \mathbb{R}$ is sufficiently "smooth", at least differentiable (e.g. $f(x) = x^2, x \in \mathbb{R}$);
- **Non-smooth optimization**, when f is a non-differentiable function (e.g. $f(x) = |x|$).

Non-smooth optimization problems are usually regarded as more challenging, while for smooth problems we can ensure strictly positive progress in each iteration of the method.

Constrained optimization. When $Q \subset \mathbb{R}^n$ (strictly), we refer to (1.1) as a constrained optimization problem. Sometimes, the constraint set can be naturally induced as the *domain* of the objective function, i.e., $Q = \operatorname{dom} f$, and may even be omitted from the problem formulation. More often, constraints are given explicitly in the model based on natural physical observations. There are two important situations that are significantly different from the perspective of algorithmic design:

- **Simple constraints.** This means that the set Q is "explicit enough" and easily understandable. The classic example of simple constraints is a ball of a certain radius:

$$Q \;=\; \Big\{x \in \mathbb{R}^n \;:\; \|x - x_0\| \le R\Big\},$$

  where $\|\cdot\|$ is some standard norm (e.g., Euclidean norm $\|\cdot\|_2$ or $\|\cdot\|_\infty$-norm). Another example of simple constraints is the standard simplex:

$$\Delta_n \;=\; \Big\{x \in \mathbb{R}^n_+ \;:\; \langle e, x\rangle = 1\Big\}.$$

  We denote by $\mathbb{R}_+$ the set of nonnegative reals, and by $e = (1, \ldots, 1)^\top$ the vector of all ones. The notion of "simplicity" is informal, and it basically means that we can perform some standard operations with the set efficiently. One of the most important is the *projection* operation[1]:

$$\operatorname{proj}_Q(x) \;:=\; \operatorname*{argmin}_{y \in Q} \|y - x\|_2.$$

  For "simple" sets, projection onto them can be done efficiently, either with an explicit formula (as with projection onto the Euclidean ball, or onto a box), or with an efficient algorithm (as is the case for the simplex; projection onto Δ_n can be computed in $O(n \log n)$ time).

[1] We will always assume projection in the Euclidean norm $\|\cdot\|_2$, unless explicitly specified.

- **Functional constraints.** This situation is more difficult. The set Q is given in the following *general form*:
$$Q \;=\; \Big\{x \in \Omega \;:\; g_1(x) \le 0,\; \ldots,\; g_m(x) \le 0\Big\}, \tag{1.3}$$
where $\Omega \subseteq \mathbb{R}^n$ is possibly some simple set, or the whole vector space, and $g_i : \Omega \to \mathbb{R}$, $1 \le i \le m$ are given functions. Note that it is enough to use only "$\le$" in (1.3), as a constraint of the form $g(x) \ge 0$ can be rewritten as $-g(x) \le 0$, and a constraint of the form $g(x) = 0$ is equivalent to the pair of inequality constraints: $g(x) \le 0$ and $-g(x) \le 0$. Sets in the general form (1.3) can be quite difficult and require special attention when solving optimization problems.

Convex optimization. We call optimization problem (1.1) convex if the set Q is convex (that is, for any two points it contains the whole segment) and the objective f is a convex function (i.e., its epigraph is a convex set). Convex optimization problems play an outstanding role in the optimization theory and, without a doubt, are among the problem classes that admit the most efficient algorithms. Moreover, ideas obtained from the convex world help significantly in solving other non-convex problems. We will devote a significant amount of time to the theory and algorithms of Convex Optimization.

1.1.2 Types of Solutions

Any point that belongs to the constraint set, $x \in Q$, we call *feasible point*. And points outside of the set, $x \notin Q$ are called *infeasible*.

A point $x^\star \in Q$ is called *global solution* to problem (1.1), if

$$f(x^\star) \;\le\; f(x), \qquad x \in Q.$$

Finding a global solution is our **ideal goal**, which can be very difficult to achieve. We denote the optimal function value by $f^\star = f(x^\star)$. In some cases, it may be easier to compute $f^\star$ than $x^\star$, while in general, finding the optimal function can be equally difficult.

Note that we can easily have a unique global solution (e.g., $x^\star = 0$ for $f(x) = x^2$), many global solutions ($x^\star \in \mathbb{R}$ for $f(x) \equiv 0$) and no solutions at all ($f(x) = e^x$).

A point $\bar{x} \in Q$ is called a *local solution* to problem (1.1), if there exists its neighborhood $\bar{x} \in U \subseteq \mathbb{R}^n$ (i.e. U is an open set containing $\bar{x}$) such that

$$f(x^\star) \;\le\; f(x), \qquad x \in U \cap Q.$$

For general non-convex problems, finding a local solution is a much more tractable goal than finding a global one. For convex problems, these two concepts appear to coincide.

1.1.3 Why Continuous?

In this course, we are interested in solving *continuous* optimization problems. This means that the target objective f is a continuous function (in most cases, it will be even differentiable), and the variables $x \in Q \subseteq \mathbb{R}^n$ take a continuous range of values.

Intuitively, it is clear that continuous decisions are much easier than discrete ones. Imagine that we have to answer the following question: *Will it snow tomorrow?*, where x represents our answer. In the case of a discrete space, we might typically have two possible answers: $x = 1$ *(yes, it will snow)* or $x = 0$ *(no, it won't snow)*. Note that such an exact weather prediction is very hard—not

only for human intelligence, but for the most advanced AI systems as well. At the same time, if we allow for a *continuous space of answers*, $0 \le x \le 1$, where x represents the probability of snow, it is much easier to predict that tomorrow *there is a high chance of snow*, (e.g., $x \ge 0.8$).

These observations show that in real life, continuous decisions are often easier to make than discrete ones. It appears to be a very fruitful approach to work with a continuous space of parameters. Even if our original problem is discrete (such as finding a maximum cut in a graph), we can turn it into a much easier one by applying a *continuous relaxation*.

1.1.4 Examples

Let us show a few basic examples demonstrating that even the simplest optimization problems might be difficult to solve.

Constrained discrete problem. Consider one-dimensional space of variables $x \in \mathbb{R}$ and the following two quadratic functions:

$$g_1(x) \;=\; x^2 - 1, \qquad\qquad g_2(x) \;=\; 1 - x^2.$$

Note that both of them are not only continuous but infinitely differentiable, and might be considered analytically ideal. However, if we look at the constraint set given by two inequalities with these function,

$$\begin{aligned} Q \;&=\; \{x \in \mathbb{R} \;:\; g_1(x) \le 0,\; g_2(x) \le 0\} \\ &=\; \{x \in \mathbb{R} \;:\; x^2 \le 1,\; x^2 \ge 1\} \\ &=\; \{x \in \mathbb{R} \;:\; x^2 = 1\} \;=\; \{\pm 1\}, \end{aligned}$$

we obtain the discrete choice of x. Thus, a constrained optimization problem even with innocent-looking functional inequality constraints might be hard. This additionally explains why unconstrained optimization is typically considered to be simpler.

Linear programming. One of the most important examples of a non-trivial constrained problem that can be solved efficiently and is rich in applications is *linear programming*. This is the problem with the simplest possible functional constraints, $g_i(x) = \langle a_i, x\rangle - b_i$, that are affine functions, for given vectors $a_i \in \mathbb{R}^n$ and numbers $b_i \in \mathbb{R}$, $1 \le i \le m$. And the objective is a linear function: $f(x) = \langle c, x\rangle$, for some $c \in \mathbb{R}^n$. Any linear programming formulation can be cast into this form:

$$\min_{x \in \mathbb{R}^n} \Big\{ \langle c, x\rangle \;:\; \langle a_1, x\rangle \le b_1,\, \ldots,\, \langle a_m, x\rangle \le b_m \Big\}.$$

Forming the matrix $A \in \mathbb{R}^{n \times m}$ with vectors $\{a_i\}$ as its columns, and the vector $b = (b_1, b_2, \ldots, b_m)^\top \in \mathbb{R}^m$, we can write down the constraint set in the following matrix notation:

$$Q \;=\; \Big\{ x \in \mathbb{R}^n \;:\; A^\top x \le b \Big\}.$$

The set defined by affine inequalities is called a *polyhedron*.

Thus, an *instance* of the linear programming problem is given by the triple of *input data*: $\{A, b, c\}$. Linear programming is already non-trivial to solve. Luckily, there exist efficient *polynomial-time* algorithms for linear programming, which we will study in this course. It is notable that the ideas behind these methods can be successfully employed for more general classes of nonlinear optimization.

Feasibility problem. A closely related to optimization problems is the *feasibility problem*, that is, for a given set $Q \subseteq \mathbb{R}^n$ to find a point in this set:

$$x^\star \quad \in \quad Q. \tag{1.4}$$

For example, for $Q = \{x \in \mathbb{R}^n : Ax = b\} = \{x \in \mathbb{R}^n : Ax - b \leq 0,\, b - Ax \leq 0\}$, the feasibility problem is to solve the linear system: $Ax^\star = b$.

The feasibility problem (1.4) can be seen as a trivial instance of the optimization problem (1.1), when the objective function is zero: $f(x) \equiv 0$. However, we can also do the opposite, by turning *any* optimization problem into a sequence of feasibility problems.

For optimization problem $\min_{x \in Q} f(x)$ with an arbitrary objective, we can introduce an *extra variable* $t \in \mathbb{R}$ and *extra constraint* $f(x) \leq t$. Then, it is easy to see that our problem is equivalent to the following one with the univariate linear objective:

$$\min_{(x,t) \in Q'} t, \tag{1.5}$$

where $Q' = \{(x,t) \in \mathbb{R}^{n+1} \ : \ x \in Q,\ f(x) \leq t\}$ is the new constraint set with extended dimension. Then, to solve (1.5) we can run the simple *binary search* over t, finding the smallest value $t^\star$ such that $(x, t^\star) \in Q'$. Therefore, feasibility problems are in general as hard as optimization ones.

The most difficult problem in the world. Let $x^\star \in \mathbb{R}^n$ be a fixed point which is unknown to us. Set the following objective function:

$$f(x) \quad = \quad \begin{cases} 0, & x = x^\star; \\ 1, & \text{otherwise.} \end{cases} \tag{1.6}$$

Then, to solve the unconstrained problem, $\min_{x \in \mathbb{R}^n} f(x)$, means *to find* $x^\star$, which is arbitrary! Thus, the good news is that *any problem in the world can be encoded as optimization problem.* The language of optimization is indeed universal and can be applied in a vast variety of applications. However, it is impossible to hope to have an algorithm that solves problems like (1.6). We come to the pessimistic conclusion that, unfortunately, *optimization problems are generally unsolvable.*

Note that the objective function in (1.6) can be easily made continuous, as to show in the following simple exercise.

Exercise 1.1.1. Let $x^\star \in \mathbb{R}^n$ be fixed. For any $\varepsilon > 0$, construct a function $f_\varepsilon : \mathbb{R}^n \to \mathbb{R}$ such that

1. f_ε is continuous;
2. $f_\varepsilon(x^\star) = 0$;
3. $f_\varepsilon(x) = 1$ for all $x \in C_\varepsilon$, where C_ε is the closed complement of the Euclidean ball:

$$C_\varepsilon \quad = \quad \{x \in \mathbb{R}^n \ : \ \|x - x^\star\|_2 \geq \varepsilon\}.$$

Hint. Consider the case $n = 1$ first.

Therefore, the idea of trying to develop the most universal algorithm capable of solving any problem in the world is hopeless. Instead, we will investigate specific *problem classes* that we can solve with *efficient algorithms*. For each problem class, we will associate its corresponding *complexity*, that will describe the methods' efficiency.

1.2 Complexity of Optimization Problems

To understand the importance of working with problem classes, let us consider another extreme situation: assume that we have *one fixed* function $f_0 : \mathbb{R}^n \to \mathbb{R}$ which has global minimum $x^\star$. What is *the best algorithm* for solving this problem? Undoubtedly, the following method will be very efficient:

$$\boxed{\textbf{return } x^\star}$$

as it will solve the problem immediately. Moreover, it will work perfectly on *all functions* that have a minimum at $x^\star$. However, on any other function, this hard-coded method will fail.

1.2.1 Notion of Problem Class

What we want instead is not a single problem, but a *family of problems*, which we call a problem class $\mathcal{P}$. Examples include:

- $\mathcal{P} = \{f_0\}$ consisting of one function — *too small* to be interesting.
- $\mathcal{P} = \{f \text{ s.t. } f \in \mathcal{C}(\mathbb{R}^n)\}$ consisting of all continuous functions — *too large*, as we have seen in the previous section.
- $\mathcal{P} = \{f \text{ s.t. } f \text{ is convex and } \nabla f \text{ is Lipschitz}\}$ — already *much better*; we will study this problem class in more detail in the following lectures.
- $\mathcal{P} = \{(f, Q) \text{ s.t. } f \text{ is linear and } Q \text{ is a polyhedron}\}$ — *linear programming*, that is polynomially solvable.
- ...

Performance of an algorithm is measured over *all problems* from $\mathcal{P}$. Note the same algorithm can behave differently for different problem classes.

1.2.2 Oracles

Another important ingredient is the notion of *oracle*, which describes how exactly an algorithm can learn any information about the function. Indeed, if the algorithm somehow "knew" the entire function, it could simply output its minimum immediately. However, this is unrealistic, except for some rare and simple cases where we can calculate the minimum explicitly by hands, such as minimizing a univariate quadratic function.

The most common case is when an algorithm has access to so-called *black-box local* oracles, which is for any given $x \in \operatorname{dom} f$ returns a local information $\mathcal{O}(x)$ about the function in a small neighborhood of x. For example,

- **Zeroth-order oracle:** $\mathcal{O}(x) = \{f(x)\}$ returns only the function value at the given point.
- **First-order oracle:** $\mathcal{O}(x) = \{f(x), \nabla f(x)\}$ returns both the function value and the gradient vector (assuming that the function is differentiable).
- **Second-order oracle:** $\mathcal{O} = \{f(x), \nabla f(x), \nabla^2 f(x)\}$ returns the function value, the gradient, and the Hessian matrix.

From a theoretical perspective, the following oracles are also interesting:

- **pth-order oracle:** $\mathcal{O}(x) = \{f(x), \nabla f(x), \ldots, \nabla^p f(x)\}$ returns all derivatives of the objective up to the order $p \geq 1$.
- **Full local oracle:** $\mathcal{O}(x)$ returns all function values in a small neighborhood $U \subset \mathbb{R}^n$ of the point $x \in \mathbb{R}^n$: $\mathcal{O} = \{f(y) : y \in U\}$. Clearly, having all function values in an open neighborhood of x is sufficient to recover all derivatives of f.

Sometimes, even first-order information is unavailable or expensive to obtain exactly, as is often the case in stochastic problems or when the objective function is non-differentiable. At the same time, we, as algorithm designers, almost always have additional structural knowledge about the problem or its class, such as in linear programming, that helps in developing the best possible algorithms.

1.2.3 What is Optimization Algorithm?

To formally define an optimization algorithm, we associate it with three sequences of mappings:

- **Main iterates:** $(A_0, A_1, \ldots)$ — each rule A_k describes how to generate the next point x_{k+1} at every iteration $k \geq 0$ of the method, given all prior learned information about the objective. Hence, the main iterates of our method are as follows, for every $k \geq 0$:

$$x_{k+1} \;=\; A_k(I_k), \qquad \text{where} \qquad I_k \;=\; (\mathcal{O}(x_0), \ldots, \mathcal{O}(x_k)).$$

  The method starts with an initialization $x_0 \in Q$, which is part of the algorithm.
- **Stopping conditions:** $(S_0, S_1, \ldots)$ — each rule S_k decides at each iteration $k \geq 0$ whether to stop the method or continue its running, based on all prior information I_k.
- **How to form the result:** $(R_0, R_1, \ldots)$ — if we decide to stop at a certain iteration, R_k forms the output given all information we have seen so far.

Therefore, we define the following general algorithmic scheme:

Algorithm 1.1: *Formal Optimization Algorithm.*

Initialization: $x_0 \in Q$

For $k \geq 0$ **iterate:**

$I_k \;=\; \{\, \mathcal{O}(x_0), \ldots, \mathcal{O}(x_k) \,\}$ // collect information

If $S_k(I_k)$ **then** // stopping condition

  return $R_k(I_k)$ // return the result

$x_{k+1} = A_k(I_k)$ // compute next point

The return rule $R_k(I_k)$ is typically the simplest part of the algorithm. While in practice we might return the point with the smallest function value among $\{x_0, \ldots, x_k\}$ or a weighted average of the iterates, we can, without loss of generality, assume that the algorithm returns the latest point:

$$R_k(I_k) \;\equiv\; x_k.$$

Indeed, if this is not the case, we can modify the algorithm so that after $S_k(I_k)$ is triggered, it performs an extra "dumb" iteration. This iteration replaces $x_{k+1} = A_k(I_k)$ with $x_{k+1} = R_k(I_k)$ and then terminates, returning the last point. Such a modification would cost just one additional oracle call.

1.2.4 Stopping Conditions

Note that we formally allow the algorithm to perform an infinite number of iterations if the stopping condition is never triggered. However, in practice, we always run a method for a *finite number of iterations*. As a consequence, we cannot hope to obtain an exact solution $x^\star$ as an output. What we get instead is an approximation, $x_k \approx x^\star$, where x_k denotes the method's result.

The following measures of inexactness are the most popular for unconstrained minimization of a differentiable function, given a fixed *desired accuracy* $\varepsilon > 0$:

- **Small functional residual:** $f(x_k) - f^\star \leq \varepsilon$.
- **Small pointwise distance:** $\|x_k - x^\star\| \leq \varepsilon$.
- **Small gradient norm:** $\|\nabla f(x_k)\| \leq \varepsilon$.

It is important to be aware that both the choice of the inexactness measure and the required accuracy level $\varepsilon > 0$ are *included in the problem formulation*. Correspondingly, in our formal algorithmic scheme, we assume that the method's stopping condition $S_k(I_k)$ ensures that the returned point x_k satisfies the desired guarantee.

1.2.5 Definition of Complexity

We are ready to formally define the notion of *complexity* for an optimization algorithm, as applied to a specific problem class.

The formal definition of the problem class includes all three key elements, which we have already discussed:

- A family of problems $\mathcal{P}$, which describes both the type of objective and the constraints;
- A measure of inexactness and the required accuracy level $\varepsilon > 0$;
- An oracle $\mathcal{O}$, which is a type of information available to an algorithm.

The oracle (which is part of the problem formulation) typically determines the class of algorithms that we consider: for example, the class of all first-order methods, second-order methods, etc.

Definition 1.2.1. The *complexity* of an optimization algorithm on a problem is the *total number of oracle calls* required to solve the instance with a fixed accuracy $\varepsilon > 0$.

Note that the complexity can be $+\infty$ if the method is unable to solve the problem with the given accuracy in a finite number of iterations. Often, this notion of complexity is also called *oracle complexity*, *analytical complexity*, or *iteration complexity* of the method.

Definition 1.2.2. The *complexity* of an optimization algorithm on a problem class is the *maximum* over complexity on a problem p, over all $p \in \mathcal{P}$, with a fixed accuracy $\varepsilon > 0$.

In other words, for a fixed algorithm, we pick the "worst" problem within our problem class. Thus, this is often called the *worst-case* complexity.

1.3 Grid Search Algorithm

1.3.1 Global Optimization

To illustrate the new concept, we consider the following example of a problem class.

The problem:

$$\min_{x \in B} f(x), \tag{1.7}$$

where $B = \{x \in \mathbb{R}^n \ : \ \|x\|_\infty \leq R\}$ is a ball of radius $R > 0$ in ℓ_∞-norm: $\|x\|_\infty := \max_{1 \leq i \leq n} |x^{(i)}|$, and $f : B \to \mathbb{R}$ is a Lipschitz continuous function. That is, for some constant $L > 0$, it holds:

$$|f(y) - f(x)| \ \leq \ L\|y - x\|_\infty, \qquad x, y \in B. \tag{1.8}$$

Note that we can employ different norms in (1.8), but it is convenient to use ℓ_∞-norm as in the constraint set. From (1.8) it follows that the function is continuous and hence it achieves its global minimum $x^\star$ over compact set B.

Parameters of our problem class are:

- Dimension $n \geq 1$;
- Radius of the ball $R > 0$;
- Lipschitz constant $L > 0$.

For the accuracy measure we fix the *functional residual* condition. Thus we want to find a point $\bar{x} \in B$ that satisfies

$$f(\bar{x}) - f^\star \ \leq \ \varepsilon. \tag{1.9}$$

And we use *zeroth-order black-box* oracle: $x \mapsto \mathcal{O}(x) = \{f(x)\}$.

1.3.2 Grid Search

We consider the following simple algorithm, widely used, for example, for tuning hyperparameters in machine learning models.

The method generates a grid of points with a given density $p \geq 1$, and returns the best point among generated.

Algorithm 1.2: *Grid Search Algorithm.*

1. **Choose** $p \geq 1$ (an integer parameter of the method).
2. **Generate** p^n points,

   $$x_{(t_1,\ldots,t_n)} \ = \ \left[-\tfrac{p-1}{p} \cdot R + \tfrac{2R}{p} t_1 \ , \ \ldots \ , \ -\tfrac{p-1}{p} \cdot R + \tfrac{2R}{p} t_n\right],$$

   where $0 \leq t_i \leq p - 1$ for each coordinate $1 \leq i \leq n$.
3. **Find** the point $\bar{x}$ with the smallest function value among all generated points.
4. **Return** $\bar{x}$.

To implement step 3 the algorithm needs only access to the values of f, which is provided by the zeroth-order oracle. We can prove the following simple result about this method.

Theorem 1.3.1. *Let $\bar{x}$ be the output of the grid search algorithm. Then,*

$$f(\bar{x}) - f^\star \;\leq\; \tfrac{LR}{p}. \tag{1.10}$$

Consequently, to solve an instance of the problem from our problem class with an ε-accuracy in terms of the functional residual, it is enough to perform

$$K \;=\; \left(\lfloor \tfrac{LR}{\varepsilon} \rfloor + 1\right)^n. \tag{1.11}$$

zeroth-order oracle calls.

Proof. By the definition of our grid, we cover the entire box B with p^n small boxes, and we generated all centers of these small boxes in step 2 of the algorithm. The side length of the initial box is $2R$, while the side length of each small box is $\frac{2R}{p}$. Hence, the radius in ℓ_∞-norm of each small box is $\frac{R}{p}$.

Since we cover the entire B, there exists a small box that contains a global solution $x^\star$. We denote the center of this small box by $\hat{x}$. It remains to note that

$$f(\bar{x}) - f^\star \;=\; f(\bar{x}) - f(x^\star) \overset{\text{step 3}}{\leq} f(\hat{x}) - f(x^\star) \overset{(1.8)}{\leq} L\|\hat{x} - x^\star\|_\infty \;\leq\; \tfrac{LR}{p},$$

where in the last inequality we used that $x^\star$ belongs to the small box. This proves (1.10).

To establish (1.11), it is enough to choose $p := \lfloor \frac{LR}{\varepsilon} \rfloor + 1 \geq \frac{LR}{\varepsilon}$. Substituting this bound into (1.10) gives $f(\bar{x}) - f^\star \leq \varepsilon$, which justifies the sufficient number of oracle calls. □

1.4 Lower Bound for Global Optimization

In this section we establish a *lower complexity bound* on global minimization, that is, the minimal number of iterations required by *any* optimization algorithm from a given class to solve the problem. We will study zeroth-order methods and establish that the grid search algorithm from the previous section is *optimal*: thus, its upper complexity bound matches the lower bound, up to a constant.

We consider the following problem:

$$\min_{x \in B} f(x), \tag{1.12}$$

where B is a ball of radius $R > 0$ around origin in an *arbitrary norm* $\|\cdot\|$:

$$B \;=\; \{x \in \mathbb{R}^n \;:\; \|x\| \leq R\}, \tag{1.13}$$

and $f : B \to \mathbb{R}$ is a Lipschitz continuous function, with constant $L > 0$:

$$|f(y) - f(x)| \;\leq\; L\|y - x\|, \qquad x, y \in B. \tag{1.14}$$

The goal is to find a point $\bar{x} \in B$ that is an approximate global solution in terms of the functional residual:

$$f(\bar{x}) - f^\star \;\leq\; \varepsilon, \tag{1.15}$$

for a given $\varepsilon > 0$.

Note that in the previous section we fixed the ℓ_∞-norm, which is the simplest for the construction of the grid search. In contrast, our lower bound will work for arbitrary norms, where the choice of norm defines the underlying geometry of the problem.

1.4.1 Packing Problem

Establishing the desired lower bound is closely related to the following famous *packing problem*, which remains an active area of research. We will only need the very basic facts about this problem.

Suppose we have a set of K points in our large ball: $x_1, \ldots, x_K \in B$ and consider a set of small balls, each of radius $0 < r < R$, centered at these points:

$$b_i \;=\; \{x \in \mathbb{R}^n \;:\; \|x - x_i\| \le r\}, \qquad 1 \le i \le K.$$

We say that the set of balls $\{b_1, \ldots, b_K\}$ is a *packing* in B if

- Each $b_i \subseteq B$;
- The interiors of any two balls are *disjoint*: $\operatorname{int} b_i \cap \operatorname{int} b_j = \varnothing$, for $i \neq j$.

In other words, we "fill" the large ball B with small balls $\{b_1, \ldots, b_N\}$ without overlaps. We say that packing $\{b_1, \ldots, b_K\}$ is *maximal* if we cannot add a single ball of radius r without it overlapping one of the existing balls. See Figure 1.1 for an illustration. A maximal packing always exists, since any packing can be greedily extended to a maximal one.

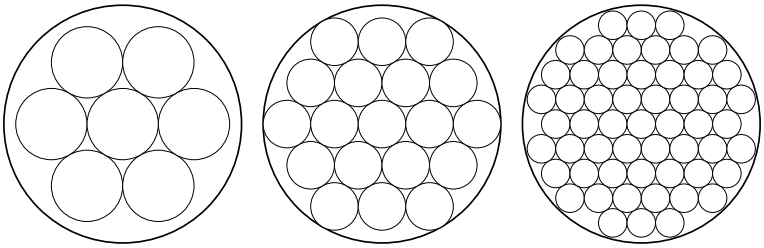

Figure 1.1: Packings of a unit Euclidean ball in $\mathbb{R}^2$.

We establish the following simple lower bound on the size of a maximal packing.

Proposition 1.4.1. *Let $\{b_1, \ldots b_K\}$ be a maximal packing. Then,*

$$K \;\ge\; \left(\tfrac{R-r}{2r}\right)^n. \tag{1.16}$$

Proof. Let us consider a slightly shrunken ball $\bar{B}$ of radius $R - r > 0$:

$$\bar{B} \;=\; \{x \in \mathbb{R}^n \;:\; \|x\| \le R - r\},$$

so that for any $x \in \bar{B}$, the small ball of radius r centered at x belongs entirely to B.

Since the packing $\{b_1, \ldots, b_K\}$ is maximal in B, for any point $x \in \bar{B}$, there exists some x_i $(1 \le i \le K)$, which is the center of the ball b_i, such that

$$\|x - x_i\| \;<\; 2r.$$

Otherwise, we could place a new ball of radius $r > 0$ centered at x and it would not overlap with any of $\{b_1, \ldots, b_K\}$, which contradicts the maximality of the packing.

Hence, the set of small balls with the same centers but twice the radius,

$$c_i \;=\; \{x \in \mathbb{R}^n \;: \|x - x_i\| \le 2r\}, \qquad 1 \le i \le K,$$

is a *covering* of $\bar{B}$:

$$\bar{B} \quad \subseteq \quad \bigcup_{1 \le i \le K} c_i. \tag{1.17}$$

Taking the volume of the left and the right-hand side in (1.17), we obtain

$$\mathrm{Vol}(\bar{B}) \quad \le \quad \sum_{i=1}^{K} \mathrm{Vol}(c_i) \quad = \quad K \cdot \mathrm{Vol}(c_1). \tag{1.18}$$

Let $\alpha := \mathrm{Vol}(\{x \in \mathbb{R}^n : \|x\| \le 1\})$ be the volume of the unit ball. Since in $\mathbb{R}^n$, the volume is a homogeneous function of degree n, we get:

$$\begin{aligned} \mathrm{Vol}(\bar{B}) &= \alpha \cdot (R-r)^n, \\ \mathrm{Vol}(c_1) &= \alpha \cdot (2r)^n. \end{aligned}$$

Substituting these values in (1.18), cancelling α, and rearranging the terms completes the proof. □

Remark 1.4.2. Let us consider the ball in ℓ_∞-norm, which is the box with side length $2R$. If we partition each side into $p \ge 1$ equal parts, where p is an integer parameter, we cover the entire box with $K = p^n$ small boxes, each of radius $r = \frac{R}{p}$ in the ℓ_∞-norm. Clearly, this gives us a maximal packing of size

$$K \quad = \quad \left(\tfrac{R}{r}\right)^n,$$

which sharpens the general lower bound provided by (1.16).

1.4.2 Resisting Oracle

We would like to establish a lower bound for the complexity of any zeroth-order algorithm applied to a problem from our class (1.12)–(1.15).

To derive the lower bound, it is useful to employ the idea of the so-called *resisting oracle*. We notice that at each iteration of a method, the oracle needs to return only the value of the objective function at a given point. It does not control the requested point, but it *controls the answer*. Therefore, the resisting oracle may always return the information that is least useful to the algorithm, forcing the algorithm to run as long as possible.

When playing this game, the resisting oracle must ensure only that in the end, when the algorithm returns a result, the oracle is able to *reveal at least one function* belonging to the problem class that is consistent with all previous answers.

In our current case, the resisting oracle is very simple:

Resisting oracle: always return $\mathcal{O}(x) \equiv \{0\}$.	(1.19)

Therefore, for every point that a method requests the function value, the function value will be zero. We can assume, without loss of generality, that the result of an algorithm after $k \ge 0$ is always the *last requested point* x_k, and thus $f(x_k) = 0$.

At the same time, after the method returns the result, we are able to construct a function which is consistent with all requested points, and that has $f^\star < 0$.

We denote by $K(R, r, n)$ a lower bound for the size of any maximal packing of n-dimensional ball B of radius R in a given norm $\|\cdot\|$, by small balls of radius $r < R$. From Proposition 1.4.1 we can take at least

$$K(R, r, n) \quad = \quad \left\lceil \left(\tfrac{R-r}{2r}\right)^n \right\rceil. \tag{1.20}$$

We use the lower bound on a maximal packing to prove the following result.

Proposition 1.4.3. *Let $0 < r < R$ be fixed. Consider any zeroth-order algorithm running for $k < K(R, r, n) - 1$ iterations. Then, there exists a Lipschitz continuous function f with Lipschitz constant $L > 0$, such that the result x_k of the algorithm when applied to this function satisfies:*

$$f(x_k) - f^\star \;=\; Lr. \tag{1.21}$$

Proof. Let $\{x_0, \ldots, x_k\} \subset B$ be the points generated by the algorithm when interacting with the resisting oracle (1.19). Thus, the result of the algorithm is $f(x_k) = 0$.

Now, let us pick an arbitrary maximal packing of B by balls of radius r. Its size is at least $K(R, r, n)$. Since $k+1 < K(R, r, n)$, there must exist at least one ball in the packing whose interior does not contain any of the points $\{x_0, \ldots, x_k\}$. We denote the center of such a ball by $x^\star$.

Introduce the function

$$f(x) \;=\; L \cdot \min\{0, \|x - x^\star\| - r\},$$

which possesses the following properties:

1. $f^\star = f(x^\star) = -Lr$.
2. For any x such that $\|x - x^\star\| \geq r$, $f(x) \equiv 0$. Hence, this function is consistent with outputs of the resisting oracle.
3. f is Lipschitz continuous with constant $L > 0$.

To check the last property, we need to verify (1.14), for any $x, y \in B$. Consider first the situation when both: $\|x - x^\star\| \geq r$ and $\|y - x^\star\| \geq r$. Then,

$$f(y) - f(x) \;=\; 0 \;\leq\; L\|y - x\|.$$

Now, assume that $\|x - x^\star\| \leq r$ and the other point y is arbitrary. Then,

$$\begin{aligned} f(y) - f(x) \;&=\; L \cdot \min\{0, \|y - x^\star\| - r\} - L \cdot (\|x - x^\star\| - r) \\ &\leq\; L \cdot (\|y - x^\star\| - \|x - x^\star\|) \;\leq\; L\|y - x\|, \end{aligned}$$

where we used triangle inequality in the last bound.

Therefore, f satisfies all the required properties, and (1.21) holds. □

1.4.3 Lower Bound

To establish the lower complexity bound, it remains to calibrate the radius $r > 0$ of the small ball according to the desired accuracy of solving the problem. Combining all elements, we arrive at the following result.

Theorem 1.4.4. *Let parameters $L > 0$, $R > 0$ and $\varepsilon > 0$ be fixed and assume that the target accuracy is sufficiently small: $\varepsilon < \frac{LR}{2}$. Then, the complexity K of any zeroth-order method on our problem class is bounded by*

$$K \;\geq\; \left\lceil \left(\tfrac{LR}{4\varepsilon} - \tfrac{1}{2} \right)^n \right\rceil - 1. \tag{1.22}$$

Proof. We set $r := \frac{2\varepsilon}{L} < R$ and assume that the method runs for $k < K(R, r, n) - 1$ iterations on any function from our problem class, where

$$K(R, r, n) \overset{(1.20)}{=} \left\lceil \left(\frac{R-r}{2r}\right)^n \right\rceil = \left\lceil \left(\frac{LR}{4\varepsilon} - \frac{1}{2}\right)^n \right\rceil.$$

Then, using Proposition 1.4.3, we conclude that there exist at least one function on which

$$f(x_k) - f^\star = Lr = 2\varepsilon,$$

which contradicts that the algorithm finds ε-solution. Hence $k \geq K(R, r, n) - 1$. □

We see that, up to numerical constants, the lower complexity bound of zeroth-order methods on our problem class is

$$K = \Omega\left(\left[\frac{LR}{\varepsilon}\right]^n\right). \tag{1.23}$$

At the same time, we saw in Theorem 1.3.1 that for the ℓ_∞-norm, this matches the complexity upper bound $O([\frac{LR}{\varepsilon}]^n)$ achieved by the grid search algorithm. This implies that the grid search is the *optimal method*, and in general, we cannot come up with a faster algorithm.

These news is rather pessimistic, as bound (1.23) is extremely large due to exponential dependence on dimension. Indeed, choosing $L = R = 1$, a very moderate accuracy $\varepsilon = 10^{-2}$, and $n \geq 50$, we obtain from (1.23) that

$$K \gtrsim 10^{100},$$

which is believed to be larger than the number of atoms in the observable universe. Therefore, from the computational perspective, it is impossible to solve the global optimization problem (high-dimensional and non-convex). Instead, we will first focus on a less ambitious goals: finding stationary points for smooth problems.

Lecture

For additional reading, we refer to Section 1.1 of [31] and Sections 1-2 of [27], the material on which these notes are largely based. Complexity theory in optimization was initially developed in the seminal book [29].

2. Finding Stationary Points

In this part, we analyze the performance of the gradient method, which uses *local* information about the function in order to generate a sequence converging to a *stationary point* of a possibly non-convex objective. We show that the gradient method converges at the same rate for an arbitrary choice of the norm, while its implementation and behavior do change if we change the norm. The standard gradient descent corresponds to the classic Euclidean norm. Finally, we analyze the stochastic gradient method, which has slower convergence due to noisy gradients.

2.1 Using Local Information: Differentiable Functions 17
2.1.1 Derivatives 18
2.1.2 Optimality Conditions 20
2.2 Predictable Function Behavior: Smoothness 21
2.2.1 Dual Space and Dual Norm 21
2.2.2 Functions with Lipschitz Gradient 22
2.3 Gradient Method for General Norms 24
2.3.1 Gradient Step 24
2.3.2 Progress of One Step 25
2.3.3 Convergence Rate to a Stationary Point 25
2.4 Adaptive Search for Gradient Method 27
2.4.1 Gradient Method: Summary 27
2.4.2 Adaptive Search 28
2.5 Stochastic Gradient Method 29
2.5.1 Stochastic Oracle 30
2.5.2 Stochastic Algorithm 30
2.5.3 Convergence Analysis 31
2.5.4 Stepsize Tuning 32
2.6 Exercises 32

2.1 Using Local Information: Differentiable Functions

We review well-known facts about differentiable functions, and discuss how to compute gradients and Hessians. Gradients are very important for optimization: they are used first as a primary search direction for optimization algorithms, and second as optimality conditions for solutions.

On vector spaces. We work with a finite-dimensional real vector space, typically denoted by $\mathbb{R}^n$. We denote by $\langle \cdot, \cdot \rangle$ the *standard inned product* for vectors in $\mathbb{R}^n$, for $x, y \in \mathbb{R}^n$: $\langle x, y \rangle = x^\top y = \sum_{i=1}^n x^{(i)} y^{(i)}$.

Sometimes decision variables can be more structured, such as *matrices* or *symmetric matrices*, or combinations of those, for example, all parameters of a neural network grouped by layers. Of course, every matrix or tensor can be reshaped into a vector, thus it is enough to be able to work with vectors. However, even though such a reshape is possible, in practice it is often convenient to keep the initial shape of objects, as for example, for symmetric matrices.

Exercise 2.1.1. Consider the space of symmetric matrices $\mathbb{S}^d = \{X \in \mathbb{R}^{d\times d} : X = X^\top\}$. Construct an explicit basis for $\mathbb{S}^d$ and prove that its dimension is $n = \frac{d(d+1)}{2}$.

For two matrices of the same size $X, Y \in \mathbb{R}^{n\times m}$, the inner product is $\mathrm{tr}(X^\top Y)$. It is immediate to check that this coincides with the standard inner product for vectors, viewed as if we were to reshape matrices into vectors in $\mathbb{R}^{nm}$:

$$\langle X, Y\rangle \;=\; \mathrm{tr}(X^\top Y) \;=\; \sum_{i=1}^{n}\sum_{i=1}^{m} X^{(i,j)}Y^{(i,j)}. \tag{2.1}$$

For the space of symmetric matrices $\mathbb{S}^d$, we use the same inner product induced by $\mathbb{R}^{d\times d}$, namely $\langle X, Y\rangle = \mathrm{tr}(XY)$. Note that it is *not equivalent* to forming two vectors in $\mathbb{R}^{d(d+1)/2}$ and taking their standard inner product, although that would also be a valid choice of inner product on $\mathbb{S}^d$. However, it is often more convenient to use the induced inner product (2.1) for symmetric matrices, as we will do in this course.

For a symmetric matrix $A \in \mathbb{S}^n$, we say that it is *positive-semidefinite* (notation: $A \succeq 0$) if

$$\langle Ah, h\rangle \;\geq\; 0, \qquad \forall h \in \mathbb{R}^n.$$

We say that A is *positive-definite* (notation: $A \succ 0$) if

$$\langle Ah, h\rangle \;>\; 0, \qquad \forall h \in \mathbb{R}^n \setminus \{0\}.$$

For two symmetric matrices $A, B \in \mathbb{S}^n$, we say $A \succeq B \Leftrightarrow A - B \succeq 0$.

It is known that for a symmetric matrix, all eigenvalues are *real* numbers, which we denote by $\lambda_1(A) \geq \ldots \geq \lambda_n(A)$. It holds: $A \succeq 0 \;\Leftrightarrow\; \lambda_i(A) \geq 0$ for all $1 \leq i \leq n$.

2.1.1 Derivatives

Consider $f : \mathbb{R}^n \to \mathbb{R}^m$. By definition, f is called *differentiable* at $x \in \mathbb{R}^n$ if there exists a linear operator $L : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$f(x+h) \;=\; f(x) + L[h] + o(\|h\|) \quad\Leftrightarrow\quad \lim_{\|h\|\to 0} \tfrac{\|f(x+h)-f(x)-L[h]\|}{\|h\|} \;=\; 0. \tag{2.2}$$

Since in finite-dimensional spaces all norms are topologically equivalent, it does not matter which norm to pick in the definition. It is easy to check that if such a linear operator L exists, then it is unique. It is called the *derivative* of f at x.

Commonly used **notations** for this linear operator are:

$$Df(x) \;\equiv\; df(x) \;\equiv\; f'(x) \;\equiv\; L.$$

In these notes, we will use $Df(x)$ to denote the derivative.

Thus, the derivative is the *best local approximation* of a function f at x by a linear function:

$$f(x+h) \;\approx\; f(x) + Df(x)[h].$$

Note that Df has two "arguments": x and h, and it is linear in h, but not in x.

Example 2.1.1 (Univariate Functions)**.** Let $f : \mathbb{R} \to \mathbb{R}$ be a univariate function. Then, $Df(x)[h] \equiv f'(x) \cdot h \in \mathbb{R}$, for $f'(x) \in \mathbb{R}$, $h \in \mathbb{R}$. In univariate calculus, h is sometimes denoted by 'dx'. Then, $f'(x)dx$ is called the *differential* of the function, $df = f'(x)dx$.

Example 2.1.2 (Squared Euclidean Norm)**.** Let $f(x) = \frac{1}{2}\|x - x_0\|_2^2 = \frac{1}{2}\langle x - x_0, x - x_0\rangle$. Then,

$$\begin{aligned} f(x+h) &= \tfrac{1}{2}\|x - x_0 + h\|^2 = \tfrac{1}{2}\|x - x_0\|^2 + \langle x - x_0, h\rangle + \tfrac{1}{2}\|h\|^2 \\ &= f(x) + \langle x - x_0, h\rangle + o(\|h\|). \end{aligned}$$

Therefore, $Df(x)[h] = \langle x - x_0, h\rangle$.

Example 2.1.3 (Frobenius Norm)**.** Let $f(X) = \frac{1}{2}\|X\|_F^2 = \frac{1}{2}\operatorname{tr}(X^\top X)$. Then,

$$\begin{aligned} f(X+H) &= \tfrac{1}{2}\operatorname{tr}((X+H)^\top (X+H)) = \tfrac{1}{2}\operatorname{tr}(X^\top X) + \operatorname{tr}(X^\top H) + \tfrac{1}{2}\operatorname{tr}(H^\top H) \\ &= f(X) + tr(X^\top H) + o(\|H\|). \end{aligned}$$

Therefore, $Df(X)[H] = \operatorname{tr}(X^\top H)$.

Gradients. In optimization, we work with functions that take real values: $f : \mathbb{R}^n \to \mathbb{R}$. Then, the *gradient* of f at x is a *unique vector* $\nabla f(x) \in \mathbb{R}^n$ such that

$$Df(x)[h] \equiv \langle \nabla f(x), h\rangle. \tag{2.3}$$

Note that the derivative $Df(x)$ on the left-hand side of (2.3) does not depend on a coordinate system, as our definition of the derivative is *coordinate-free*. Conversely, the right-hand side (2.3) depends on the choice of inner product. Hence, the gradient $\nabla f(x)$ *depends on a choice of the coordinate system*, and will change if we change the basis.

Example 2.1.4. From the previous examples, we immediately see that for $f(x) = \frac{1}{2}\|x - x_0\|_2^2$, the gradient is $\nabla f(x) = x - x_0$. For the matrix function $f(X) = \frac{1}{2}\|X\|_F^2$, the gradient is $\nabla f(X) = X$.

Directional derivative. Let $f : \mathbb{R}^n \to \mathbb{R}$. For any $h \in \mathbb{R}^n$, the *directional derivative* of f at point x *along direction* h is the derivative of the univariate function $\varphi(t) = f(x + th)$ at zero:

$$\frac{\partial f(x)}{\partial h} := \varphi'(0) = \lim_{t \to 0} \frac{f(x+th) - f(x)}{t}.$$

Directional derivative is a weaker notion that differentiability of a function. Even if a function has directional derivatives along any direction, it can be non-differentiable. However, if the function is differentiable and we know the derivative Df, it is very easy to compute the directional derivative.

Proposition 2.1.5. *For a differentiable function, it holds:*

$$\frac{\partial f(x)}{\partial h} = Df(x)[h] = \langle \nabla f(x), h\rangle. \tag{2.4}$$

Exercise 2.1.2. Check (2.4).

Therefore, we have *two principal ways* of computing the gradients:

1. *Coordinate-wise way.* We first compute all partial derivatives $\frac{\partial f(x)}{\partial x^{(i)}}$ for all coordinate directions $e_1, \ldots, e_n \in \mathbb{R}^n$. Then, we combine them into vector:

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x^{(1)}}, \ldots, \frac{\partial f(x)}{\partial x^{(n)}}\right)^\top \in \mathbb{R}^n. \tag{2.5}$$

   Theoretically, this approach is completely fine. Moreover, formula (2.5) can be used as the definition of the gradient. However, it can be computationally hard in practice, especially when working with matrix spaces.

2. *Coordinate-free way.* Think of $\nabla f(x)$ as of the *derivative representation.* We first compute the linear operator $Df(x)[h]$, as applied to an arbitrary direction h, and then find the unique vector $\nabla f(x)$ from the following equation: $Df(x)[h] \equiv \langle \nabla f(x), h \rangle$. Note that from linear algebra we know that such representation is always possible. Often, this approach of computing the gradient is much easier.

Note that $\nabla f(x)$ has always *the same shape* as the target variable x (e.g. a vector, a matrix, multiple tensors — layers in neural networks, etc.)

Second derivative. Assume that f is differentiable at *every point* x, and denote the new function $g(x) := Df(x)[h]$, for some fixed direction h. By definition, the derivative of g at x is defined by

$$g(x+u) \;=\; g(x) + Dg(x)[u] + o(\|u\|),$$

and Dg is called the *second derivative* of f. It is denoted by

$$D^2 f(x)[h,u] \;\equiv\; Dg(x)[u].$$

When f is sufficiently smooth (e.g. when $D^2 f(x)$ is continuous), it can be shown that the second derivative is symmetric: $D^2 f(x)[h,u] \equiv D^2 f(x)[u,h]$ and linear with respect to both "h" and "u".

A fundamental result from calculus is the following:

Proposition 2.1.6 (Taylor's Formula)**.** *For a twice differentiable function, it holds*

$$f(x+h) \;=\; f(x) + Df(x)[h] + \tfrac{1}{2} D^2 f(x)[h,h] + o(\|h\|^2).$$

Hessian matrix. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function. Then, the *Hessian* of f at x is a unique matrix $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$ such that

$$D^2 f(x)[h,u] \;\equiv\; \langle \nabla^2 f(x) h, u \rangle.$$

Therefore, the Hessian $\nabla^2 f(x)$ is the *representation* of the bilinear symmetric form $D^2 f(x)$, and it depends on the choice of the coordinate system. Since $D^2 f(x)$ is symmetric form, the Hessian is a symmetric matrix: $\nabla^2 f(x) \in \mathbb{S}^n$. Its entries can be computed coordinate-wise as:

$$[\nabla^2 f(x)]^{(i,j)} \;=\; \tfrac{\partial^2 f(x)}{\partial x^{(i)} \partial x^{(j)}}, \qquad 1 \le i,j \le n.$$

2.1.2 Optimality Conditions

Theorem 2.1.7 (Optimality Conditions)**.** *Let $x^\star$ be a local minimum of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. Then*

$$\nabla f(x^\star) \;=\; 0 \tag{2.6}$$

If function f is twice continuously differentiable, then additionally we have:

$$\nabla^2 f(x^\star) \;\succeq\; 0. \tag{2.7}$$

Proof. Assume $\nabla f(x^\star) \neq 0$. Take $h := -\alpha \nabla f(x^\star)$ with $\alpha > 0$, and consider

$$\begin{aligned} f(x^\star + h) &= f(x^\star) + \langle \nabla f(x^\star), h \rangle + o(\|h\|) \\ &= f(x^\star) - \alpha \|\nabla f(x^\star)\|_2^2 + o(\alpha). \end{aligned}$$

For sufficiently small α, we get: $f(x^\star + h) \leq f(x^\star) - \frac{\alpha}{2}\|\nabla f(x^\star)\|_2^2 < f(x^\star)$, which contradicts that $x^\star$ is a local minimum. Thus, we have proved (2.6).

To prove (2.7), we use Taylor's formula:

$$\begin{aligned} f(x^\star + h) &= f(x^\star) + \langle \nabla f(x^\star), h\rangle + \tfrac{1}{2}\langle \nabla^2 f(x^\star)h, h\rangle + o(\|h\|^2) \\ &\overset{(2.6)}{=} f(x^\star) + \tfrac{1}{2}\langle \nabla^2 f(x^\star)h, h\rangle + o(\|h\|^2). \end{aligned}$$

Assume $\nabla^2 f(x^\star) \not\succeq 0$. Then, there exists a direction u such that $\xi = \langle \nabla f(x^\star)u, u\rangle < 0$. Choose $h := \alpha u$ with $\alpha > 0$. We get, for sufficiently small α:

$$f(x^\star + h) = f(x^\star) + \tfrac{\alpha^2}{2}\xi + o(\alpha^2) \leq f(x^\star) + \tfrac{\alpha^2}{4}\xi < f(x^\star),$$

which contradicts that $x^\star$ is a local minimum. Thus, (2.7) is true. □

Positive definiteness of the Hessian serves as a *sufficient condition* for a point to be a strict local minimum.

Exercise 2.1.3. Let $\bar{x} \in \mathbb{R}^n$ and assume that it holds: $\nabla f(\bar{x}) = 0$ and $\nabla^2 f(\bar{x}) \succ 0$. Then, $\bar{x}$ is a strict local minimum of f, i.e. there exists a neighborhood $U \subset \mathbb{R}^n$ of $\bar{x}$ such that

$$f(\bar{x}) < f(x), \qquad x \in U \setminus \{\bar{x}\}.$$

2.2 Predictable Function Behavior: Smoothness

The key observation that we used to prove the first-order optimality condition is the following one: if at some point x the gradient is non-zero, $\nabla f(x) \neq 0$, then we can move in the direction of anti-gradient to improve the objective function value, for a sufficiently small $\alpha > 0$:

$$f(x - \alpha\nabla f(x)) = f(x) - \alpha\|\nabla f(x)\|_2^2 + o(\alpha) \leq f(x) - \tfrac{\alpha}{2}\|\nabla f(x)\|_2^2. \tag{2.8}$$

This observation is used in the core of the *gradient descent*, the most popular optimization algorithm. For a new point:

$$x^+ = x - \alpha\nabla f(x), \tag{2.9}$$

we can ensure $f(x^+) < f(x)$ when the "step-size" α is sufficiently small. But how small it should be? To implement the method and prove a reasonable rate of convergence, we seek a *quantitative characterization* of α that ensures (2.8). Clearly, it should be related to the behavior of f. In optimization, such a characterization is often called the objective *smoothness*.

2.2.1 Dual Space and Dual Norm

We want to be able to work with arbitrary norms, as the right choice can be crucial in applications.

Assume that we have a fixed norm $\|\cdot\|$ (not necessary Euclidean) in $\mathbb{R}^n$. We define the corresponding *dual norm* $\|\cdot\|_*$ as follows:

$$\|s\|_* := \max_{x:\|x\|\leq 1} \langle s, x\rangle = \max_{x:\|x\|=1} \langle s, x\rangle, \qquad s \in \mathbb{R}^n. \tag{2.10}$$

Exercise 2.2.1. Show that all properties of a norm hold for $\|\cdot\|_*$.

Defined this way, the dual norm automatically satisfies the Cauchy-Schwartz inequality:

$$|\langle s, x\rangle| \quad \leq \quad \|s\|_* \cdot \|x\|, \qquad x, s \in \mathbb{R}^n. \tag{2.11}$$

Example 2.2.1. Let the primal norm be Euclidean norm: $\|x\| := \|x\|_2 = \langle x, x\rangle^{1/2}$. Then, the dual norm is also Euclidean: $\|s\|_* := \|s\|_2$, which follows from the classical Cauchy-Schwartz inequality.

Example 2.2.2. Let $\|x\| := \langle Bx, x\rangle^{1/2}$, where $B = B^\top \succ 0$ is a fixed positive-definite matrix. Then, the dual norm is given by $\|s\|_* = \langle s, B^{-1}s\rangle^{1/2}$.

Example 2.2.3. Let $\|x\| := \|x\|_p$, for some $p \in [0, \infty]$, where $\|x\|_p := \left(\sum\limits_{i=1}^n |x^{(i)}|^p\right)^{1/p}$ for $p \geq 1$ and $\|x\|_\infty := \max\limits_{i=1}^n |x^{(i)}|$. Then, the dual norm is given by $\|s\|_* = \|s\|_q$ where $q \geq 1$ satisfies $\frac{1}{q} + \frac{1}{p} = 1$. The dual for $\|\cdot\|_\infty$ norm is $\|\cdot\|_1$ and vica versa.

While we use the *primal norm* $\|\cdot\|$ for vectors in our *primal space* $\mathbb{R}^n$, the *dual norm* $\|\cdot\|_*$ is used to measure the size of *linear forms* on $\mathbb{R}^n$, which are the elements of the *dual space*. The main example of a linear form for us is the derivative: $Df(x)[\cdot] \equiv \langle \nabla f(x), \cdot\rangle$.

The definition of the dual norm is very useful as we often have to employ bounds like this:

$$\langle \nabla f(x), h\rangle \quad \overset{(2.11)}{\leq} \quad \|\nabla f(x)\|_* \cdot \|h\|, \qquad x, h \in \mathbb{R}^n.$$

Every matrix $A \in \mathbb{R}^{n\times n}$ can be treated as a bilinear form: $(h, u) \mapsto \langle Ah, u\rangle$ for any $h, u \in \mathbb{R}^n$, and it is convenient to use the following *operator norm*, induced by the primal norm:

$$\|A\| \quad := \quad \max_{h:\|h\|\leq 1} \|Ah\|_* \quad = \quad \max_{\substack{h:\|h\|\leq 1 \\ u:\|u\|\leq 1}} \langle Ah, u\rangle.$$

This definition ensures that we have the following inequality: $\|Ah\|_* \leq \|A\| \cdot \|h\|$.

2.2.2 Functions with Lipschitz Gradient

We fix a primal norm $\|\cdot\|$ in our space (not necessary Euclidean). We say that a function $f : \mathbb{R}^n \to \mathbb{R}$ has Lipschitz continuous gradient with constant $L > 0$, with respect to this norm, if

$$\|\nabla f(y) - \nabla f(x)\|_* \quad \leq \quad L\|y - x\|, \qquad x, y \in \mathbb{R}^n. \tag{2.12}$$

The functions that satisfy (2.12) are often called *smooth functions* in optimization. Note that in the Euclidean case, we have the same Euclidean norm in the left- and right-hand sides of (2.12).

Intuitively, condition (2.12) says that if the points are close: $x \approx y$, then the gradients should also be uniformly close: $\nabla f(x) \approx \nabla f(y)$.

Note that L is a *global constant* as (2.12) should hold on the entire space $\mathbb{R}^n$. In case of constrained optimization, we can restrict (2.12) onto a given feasible set $Q \subset \mathbb{R}^n$.

For now, we consider the unconstrained optimization:

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

and use definition (2.12).

The following second-order characterization of smoothness is very important.

Theorem 2.2.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable. Then, the following statements are equivalent:*

- $\nabla f(\cdot)$ *is Lipschitz continuous with constant* $L > 0$.
- *For any* $x \in \mathbb{R}^n$, *we have*

$$\|\nabla^2 f(x)\| \quad \leq \quad L. \tag{2.13}$$

Remark 2.2.5. For the Euclidean norm, condition (2.13) is equivalent to:

$$-LI \quad \preceq \quad \nabla^2 f(x) \quad \preceq \quad LI$$

(all eigenvalues of the Hessian are in $[-L; L]$).

Proof. Assume that the gradient is Lipschitz, and choose an arbitrary direction $h \in \mathbb{R}^n$ of unit length, $\|h\| = 1$, and a small $\varepsilon > 0$. Then, by the definition of the derivative, we have:

$$\nabla^2 f(x)h \quad = \quad \tfrac{1}{\varepsilon}(\nabla f(x + \varepsilon h) - \nabla f(x)) + o(1).$$

Hence, taking the norm and using triangle inequality, we get

$$\begin{aligned} \|\nabla^2 f(x)h\|_* \quad &\leq \quad \tfrac{1}{\varepsilon}\|\nabla f(x + \varepsilon h) - \nabla f(x)\|_* + o(1) \\ &\overset{(2.12)}{\leq} \quad L + o(1). \end{aligned}$$

Taking the limit $\varepsilon \to 0$ we get $\|\nabla^2 f(x)h\|_* \leq L$. Since h is arbitrary we proved (2.13).

Now assume that (2.13) holds. Using the fundamental theorem of calculus, we have:

$$\begin{aligned} \|\nabla f(y) - \nabla f(x)\|_* \quad &= \quad \|\int_0^1 \nabla^2 f(x + \tau(y - x))(y - x)d\tau\|_* \\ &\leq \quad \int_0^1 \|\nabla^2 f(x + \tau(y - x))\|d\tau \cdot \|y - x\| \quad \overset{(2.13)}{\leq} \quad L\|y - x\|, \end{aligned}$$

which finishes the proof. □

Example 2.2.6 (Univariate Functions)**.** The derivative of the following univariate functions is Lipschitz continuous:

- $f(x) = a + bx + cx^2$.
- $f(x) = \sin(x)$.
- $f(x) = \ln(1 + e^x)$.

The derivative of the following functions *is not* Lipschitz continuous (globally):

- $f(x) = |x|^3$
- $f(x) = e^x$

Example 2.2.7 (Quadratic Function)**.** Let $f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$ for some $A = A^\top \succeq 0$, $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. Then, $L = \lambda_{\max}(A)$ (with respect to the Euclidean norm).

Theorem 2.2.8 (Global Model of the Function)**.** *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *have Lipschitz continuous gradient with constant* $L > 0$. *Then,*

$$|f(y) - f(x) - \langle \nabla f(x), y - x\rangle| \quad \leq \quad \tfrac{L}{2}\|y - x\|^2, \qquad x, y \in \mathbb{R}^n. \tag{2.14}$$

Proof. Using the fundamental theorem of calculus, we have

$$\begin{aligned} |f(y) - f(x) - \langle \nabla f(x), y - x \rangle| &= |\int_0^1 \langle \nabla f(x + \tau(y-x)) - \nabla f(x), y - x \rangle d\tau| \\ &\leq \int_0^1 |\langle \nabla f(x + \tau(y-x)) - \nabla f(x), y - x \rangle| d\tau \overset{(2.11)}{\leq} \int_0^1 \|\nabla f(x + \tau(y-x)) - \nabla f(x)\|_* d\tau \cdot \|y - x\| \\ &\overset{(2.12)}{\leq} \int_0^1 \tau d\tau \cdot L\|y - x\|^2 = \tfrac{L}{2}\|y - x\|^2, \end{aligned}$$

which is the required bound. □

2.3 Gradient Method for General Norms

2.3.1 Gradient Step

The main idea in the design and analysis of the gradient method is to use bound (2.14) as the *global upper approximation* of the objective. Staying at a point $x \in \mathbb{R}^n$, we fix a regularization constant $M > 0$ and approximate our objective $f(y)$ by the linear model augmented with quadratic regularizer:

$$f(y) \approx \Omega_M(x; y) := f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{M}{2}\|y - x\|^2, \qquad x, y \in \mathbb{R}^n.$$

By Theorem 2.2.8, we know that for a sufficiently large regularization parameter (at least, for $M \geq L$), this will be the *global model*: $f(y) \leq \Omega_M(x; y)$ for any $y \in \mathbb{R}^n$. One step of the gradient method consists in minimizing the model $\Omega_M(x; y)$ in y to obtain the next iterate:

$$x^+ = x_M^+(x) = \operatorname{argmin}_{y \in \mathbb{R}^n} \Big[\Omega_M(x; y) \Big]. \tag{2.15}$$

Note that a solution to subproblem (2.15) always exists, but may not be unique. If there are many solutions, we can pick any for x^+.

Example 2.3.1 (Euclidean Norm)**.** Let the norm be the standard Euclidean: $\|\cdot\| \equiv \|\cdot\|_2$. To compute x^+ we differentiate $g(y) \equiv \Omega_M(x; y)$ with respect to y:

$$\nabla g(y) = \nabla f(x) + M(y - x),$$

and set the gradient to zero $\nabla g(x^+) = 0$ which gives the unique solution:

$$x^+ = x - \tfrac{1}{M}\nabla f(x),$$

and the minimum of the model is

$$g^\star = \Omega_M(x; x^+) = f(x) - \tfrac{1}{2M}\|\nabla f(x)\|_2^2.$$

Therefore, for the Euclidean norm, computing the minimizer of (2.15) corresponds exactly to the classical gradient descent step (2.9) with step-size $\alpha = 1/M$. □

Example 2.3.2 (General Norm)**.** To solve the subproblem (2.15) for the case of a general norm, let us represent the displacement as follows:

$$y - x \;=\; \tau h,$$

where $h \in \mathbb{R}^n : \|h\| = 1$ and $\tau > 0$. Then, the subproblem becomes

$$\begin{aligned} \Omega_M(x; x^+) \;&=\; \min_{y \in \mathbb{R}^n} \Big[\Omega_M(x; y) \Big] \;=\; \min_{\tau > 0} \min_{h \in \mathbb{R}^n : \|h\| = 1} \Big[f(x) + \tau \langle \nabla f(x), h \rangle + \tfrac{M}{2} \tau^2 \Big] \\ &=\; \min_{\tau > 0} \Big[f(x) - \tau \|\nabla f(x)\|_* + \tfrac{M}{2} \tau^2 \Big] \;=\; f(x) - \tfrac{\|\nabla f(x)\|_*^2}{2M}. \end{aligned} \tag{2.16}$$

The optimum value is achieved for $x^+ - x = \tau^+ h^+$, where $\tau^+ = \frac{\|\nabla f(x)\|_*}{M}$ is the solution to a univariate quadratic minimization, and $h^+ \in \mathbb{R}^n$ is a vector of unit length such that

$$\langle \nabla f(x), h^+ \rangle \;=\; -\|\nabla f(x)\|_*.$$

Note that such h^+ always exists, but may not be unique. □

2.3.2 Progress of One Step

Now we have all ingredients to demonstrate the progress of one gradient step (2.15), when regularization parameter $M > 0$ is sufficiently large. We prove the following simple result, which is sometimes called *descent lemma* in the literature.

Proposition 2.3.3. *Let $M \geq L$. Then,*

$$f(x) - f(x^+) \;\geq\; \tfrac{1}{2M} \|\nabla f(x)\|_*^2. \tag{2.17}$$

Proof. Indeed, from Theorem 2.2.8 we have that

$$f(y) \;\leq\; f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{L}{2} \|y - x\|^2 \;\leq\; \Omega_M(x; y), \qquad x, y \in \mathbb{R}^n,$$

where in the last inequality we used that $M \geq L$. Now, plugging $y := x^+$ where x^+ is any solution to the subproblem (2.15), we get

$$f(x^+) \;\leq\; \Omega_M(x; x^+) \;\overset{(2.16)}{=}\; f(x) - \tfrac{1}{2M} \|\nabla f(x)\|_*^2,$$

which is the required progress. □

2.3.3 Convergence Rate to a Stationary Point

We consider the following algorithm.

Algorithm 2.1: *Gradient Method.*

Initialization: $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$

For $k \geq 0$ **iterate:**

1. If $\|\nabla f(x_k)\|_* \leq \varepsilon$ then
   return x_k
2. Choose $M_k > 0$
3. Perform the gradient step:

$$x_{k+1} \;=\; \operatorname*{argmin}_{y \in \mathbb{R}^n} \Big[\Omega_{M_k}(x_k; y) \;:=\; f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \tfrac{M_k}{2} \|y - x_k\|^2 \Big]$$

In step 2 of this method, we have to choose the regularization parameter $M_k > 0$. A natural choice, which is approved by the condition of Proposition 2.3.3 is the *constant step-size*: $M_k \equiv L$. Of course, for that we have to know the Lipschitz constant.

Another powerful rule is to simply ensure that at each step $k \geq 0$, we have the progress (2.17):

$$\begin{array}{c} \text{Choose } M_k > 0 \text{ s.t. for } x_k^+(M_k) := \operatorname*{argmin}_{y \in \mathbb{R}^n} \Omega_{M_k}(x_k; y) \text{ it holds} \\ f(x_k) - f(x_k^+(M_k)) \;\geq\; \tfrac{1}{2M_k} \|\nabla f(x_k)\|_*^2. \end{array} \tag{2.18}$$

Such condition can be achieved by an *adaptive search* procedure, that we discuss in the next section.

We prove the following convergence result for the gradient method.

Theorem 2.3.4. *Let f be bounded from below: $f^\star := \inf_{y \in \mathbb{R}^n} f(x) > -\infty$. Consider the sequence generated by the gradient method,*

$$x_{k+1} \;=\; x_k^+(M_k), \qquad k \geq 0.$$

for a sequence of regularization parameters $\{M_k\}_{k \geq 0}$.

Assume that all M_k satisfy the progress condition (2.18) *and are bounded from above: $M_k \leq M_\star$ for all $k \geq 0$. Then, it holds*

$$\tfrac{2M_\star (f(x_0) - f^\star)}{k} \;\geq\; \tfrac{1}{k} \sum_{i=0}^{k-1} \|\nabla f(x_i)\|_*^2 \;\geq\; \min_{0 \leq i \leq k-1} \|\nabla f(x_i)\|_*^2. \tag{2.19}$$

Proof. For every iteration, it holds:

$$f(x_i) - f(x_{i+1}) \overset{(2.18)}{\geq} \tfrac{1}{2M_k} \|\nabla f(x_i)\|_*^2 \;\geq\; \tfrac{1}{2M_\star} \|\nabla f(x_i)\|_*^2.$$

Summing up these inequalities for $0 \leq i \leq k-1$, we get

$$f(x_0) - f(x_k) \;\geq\; \tfrac{1}{2M_\star} \sum_{i=0}^{k-1} \|\nabla f(x_i)\|_*^2.$$

Using the bound: $f(x_k) \geq f^\star$ and multiplying both sides by $\frac{2M_\star}{k}$ completes the proof. □

We see that the gradient method makes the minimal gradient to converge to zero:

$$\min_{0 \le i \le k-1} \|\nabla f(x_i)\|_* \;\to\; 0, \qquad \text{with} \qquad k \to +\infty.$$

However, we do not ensure monotonicity of the sequence $\{\|\nabla f(x_k)\|_*\}_{k \ge 0}$, and it does not hold in general.

As a direct consequence of (2.19), we obtain the following complexity bound for our Algorithm 3.1.

Corollary 2.3.5. *To find a point $\bar{x} \in \mathbb{R}^n$ such that $\|\nabla f(\bar{x})\|_* \le \varepsilon$, the gradient method needs to perform*

$$K \;=\; \left\lfloor \tfrac{2M_\star (f(x_0) - f^\star)}{\varepsilon^2} \right\rfloor$$

first-order oracle calls, where $M_\star \ge M_k$, $k \ge 0$, is an upper bound on the regularization parameters.

In particular, choosing $M_k \equiv L$, we obtain the complexity:

$$K \;=\; \left\lfloor \tfrac{2L(f(x_0) - f^\star)}{\varepsilon^2} \right\rfloor. \tag{2.20}$$

In contrast to the complexity bound for global optimization proved in the previous chapter: $O((1/\varepsilon)^n)$, we see from (2.20) that

the complexity of the gradient method does not depend on the dimension n,

at least explicitly (it may depend on the dimension indirectly through parameters, such as the Lipschitz constant L). This explains why the gradient method is the most popular approach for solving huge-scale problems, when the dimension is extremely high ($n \to +\infty$).

2.4 Adaptive Search for Gradient Method

2.4.1 Gradient Method: Summary

We have studied the gradient method, which starts from some initialization $x_0 \in \mathbb{R}^n$ and then iterates, for $k \ge 0$:

$$x_{k+1} \;:=\; x_k^+(M_k),$$

where $\{M_k\}_{k \ge 0}$ is a sequence of regularization parameters, and $x_k^+(M)$ denotes a gradient step with respect to the given norm $\|\cdot\|$:

$$x_k^+(M) \;=\; \operatorname*{argmin}_{y \in \mathbb{R}^n} \Big\{ f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \tfrac{M}{2} \|y - x_k\|^2 \Big\}. \tag{2.21}$$

In case the norm is Euclidean, $\|\cdot\| = \|\cdot\|_2$, this simplifies to the classical update:

$$x_k^+(M) \;=\; x_k - \tfrac{1}{M} \nabla f(x_k), \tag{2.22}$$

and parameter $M > 0$ controls the step-size. For a non-Euclidean norm, the solution to (2.21) may be not unique, but always exists. We ensures the following progress in function value, for $M \ge L$:

$$f(x_k) - f(x_k^+(M)) \;\ge\; \tfrac{1}{2M} \|\nabla f(x_k)\|_*^2. \tag{2.23}$$

This is the key inequality to ensure convergence. It holds if we choose the *constant step-size*: $M_k \equiv L$, for every $k \ge 0$. But can we do better?

It appears that in practice it is much more efficient to use an *adaptive strategy* for choosing the regularization parameters, which does not require knowing the Lipschitz constant, and which can adaptively adjust to a local smoothness of the objective. Such an adaptive strategy is also often called a *line search*, as, in the Euclidean case, changing the value of M allows us to travel along the ray spanned by the gradient (2.22); this property no longer holds if we consider arbitrary norms.

2.4.2 Adaptive Search

To choose $M_k \geq 0$ adaptively, we can use the following algorithm, which is very useful in practice.

We start with an initial estimate of the regularization constant $M_0 > 0$, which can be arbitrary. Then, at each iteration $k \geq 0$, we double our current estimate M_k until (2.23) is satisfied. After each iteration, we also divide it by 2 to ensure both growth and decrease of the sequence $\{M_k\}_{k \geq 0}$.

Algorithm 2.2: *Gradient Method with Adaptive Search.*

Initialization: $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, $M_0 > 0$

For $k \geq 0$ **iterate:**

1. If $\|\nabla f(x_k)\|_* \leq \varepsilon$ then
   return x_k
2. **For** $t \geq 0$ **iterate:**
   - Set $M_k^+ := 2^t \cdot M_k$
   - Try gradient step: $x_k^+ := x_k^+(M_k^+)$
   - If $f(x_k) - f(x_k^+) \geq \frac{1}{2M_k^+}\|\nabla f(x_k)\|_*^2$ then **break** and **go to** step 3
3. Set $x_{k+1} = x_k^+$ and $M_{k+1} = \frac{1}{2}M_k^+$

This algorithm is well-define since the break condition will be satisfied at least when $M_k^+ \geq L$. In addition to using gradients, at each iteration $k \geq 0$ we might need to compute several function values (at least one). However, it is easy to show that the total number of oracle calls is well-bounded.

Proposition 2.4.1. *For each $k \geq 0$, we have*

$$M_k \;\leq\; M_\star \;:=\; \max\Big\{M_0,\, L\Big\}. \tag{2.24}$$

Moreover, during the first $k \geq 1$ iterations of the method, the total number of first-order oracle calls N_k of the type $\mathcal{O}(x) = \{f(x), \nabla f(x)\}$ is bounded as

$$N_k \;\leq\; 2k + \max\{0,\, 1 + \log_2 \tfrac{L}{M_0}\} \tag{2.25}$$

Proof. We prove (2.24) by induction. It obviously holds for $k = 0$. Assume that it holds for some $k \geq 0$ and consider one iteration of the algorithm. Let us denote by $t_k \geq 0$ the value of parameter t at step 2 that triggers the break. Thus,

$$M_{k+1} \;=\; 2^{t_k - 1} M_k. \tag{2.26}$$

If $t_k = 0$, then $M_{k+1} = \frac{1}{2}M_k \overset{(2.24)}{\leq} \frac{1}{2}M_\star < M_\star$. Otherwise, if $t_k > 0$, it means that the break condition was not satisfied for $2^{t_k-1}M_k \equiv M_{k+1}$, so $M_{k+1} < L \leq M_\star$. Thus, we have established (2.24) for all $k \geq 0$.

To show (2.25), we notice that

$$\begin{aligned} N_k \;&=\; \sum_{i=0}^{k-1}(1 + t_i) \;\overset{(2.26)}{=}\; \sum_{i=0}^{k-1}\big(2 + \log_2 \tfrac{M_{i+1}}{M_i}\big) \;=\; 2k + \log_2 \tfrac{M_k}{M_0} \\ &\overset{(2.24)}{\leq}\; 2k + \max\{0,\, 1 + \log_2 \tfrac{L}{M_0}\}, \end{aligned}$$

which completes the proof. □

Due to bound (2.24), and using Theorem 2.3.4 from the previous section, we ensure the same iteration complexity for Algorithm 2.2 as for the method with exact Lipschitz constant.

Corollary 2.4.2. *To find a point $\bar{x} \in \mathbb{R}^n$ such that $\|\nabla f(\bar{x})\|_* \leq \varepsilon$, Algorithm 2.2 needs to perform*

$$K \;=\; \left\lceil \frac{4 \max\{M_0, L\}(f(x_0) - f^\star)}{\varepsilon^2} \right\rceil$$

iterations, and the total number of oracle calls is bounded as in (2.25).

2.5 Stochastic Gradient Method

In applications, sometimes we do not have an access to the exact gradients $\nabla f(x)$, or it can be too expensive to compute them.

Example 2.5.1 (Expensive)**.** In machine learning, we are often interested in solving problems of the following form, called the *finite-sum* structure:

$$\min_{x \in \mathbb{R}^n} \Big[f(x) \;:=\; \tfrac{1}{N} \sum_{i=1}^{N} f_i(x) \Big], \tag{2.27}$$

where each $f_i : \mathbb{R}^n \to \mathbb{R}$ is a smooth function, $1 \leq i \leq N$, and N is typically a large number (i.e., number of entries in a dataset). Then, computing the exact gradient of f would involve computing the gradient over all data samples:

$$\nabla f(x) \;=\; \tfrac{1}{N} \sum_{i=1}^{N} \nabla f_i(x). \tag{2.28}$$

For large-scale datasets, an $O(N)$ per-iteration cost can be prohibitive. However, we can afford to compute a small subset of gradients from the finite sum (2.28).

Example 2.5.2 (Impossible)**.** Another class of problems is *purely stochastic optimization*, where the objective function is given in the integral form:

$$\min_{x \in \mathbb{R}^n} \Big[f(x) \;:=\; \mathbb{E}_\xi \big[F(x, \xi) \big] \Big], \tag{2.29}$$

where ξ is some random variable, and each $F(\cdot, \xi)$ is a smooth function. Of course, the finite-sum problem (2.27) can be casted as (2.29) by setting ξ to be a uniform distribution over $\{1, \ldots, N\}$. In general, it may be impossible for us to compute the full gradient $\nabla f(x)$, while we have access to $\nabla_x F(x, \xi)$ for any random sample ξ.

Example 2.5.3 (Unfavorable)**.** In some data analysis and machine learning problems, we may be restricted from using exact gradients $\nabla f(x)$ to protect user privacy. In such cases, we can *intentionally introduce noise* into the oracle information to prevent identifiability.

2.5.1 Stochastic Oracle

To cover such situations, we refine the formulation of our problem. We are still interested to minimize a smooth objective f: $\min_{x \in \mathbb{R}^n} f(x)$ that satisfies our previous conditions. However, instead of an access to exact gradients, we assume that, for any point $x \in \mathbb{R}^n$, we can sample a random variable ξ and get an access to *stochastic gradient* vector:

$$g(x; \xi) \quad \in \quad \mathbb{R}^n.$$

For example, solving finite-sum problems (2.27), ξ can be an index i of a function from the sum that we sample, and then $g(x; i) := \nabla f_i(x)$.

For simplicity, let us fix the standard Euclidean norm $\| \cdot \| := \| \cdot \|_2$ for the rest of this section. We assume that $g(x; \xi)$ satisfy the following properties.

1. $g(x; \xi)$ is an **unbiased estimator** of the true gradient $\nabla f(x)$:

$$\mathbb{E}_\xi [g(x; \xi)] \quad = \quad \nabla f(x), \qquad x \in \mathbb{R}^n.$$

2. $g(x; \xi)$ has a bounded **variance**:

$$\mathbb{E}_\xi \left[\|g(x; \xi) - \nabla f(x)\|^2 \right] \quad \leq \quad \sigma^2, \qquad x \in \mathbb{R}^n,$$

where $\sigma > 0$ is now a parameter of our problem class. Note that

$$\|g(x; \xi) - \nabla f(x)\|^2 \quad = \quad \|g(x; \xi)\|^2 - 2\langle g(x; \xi), \nabla f(x) \rangle + \|\nabla f(x)\|^2.$$

Thus, taking the expectation and rearranging the gradient norm, we obtain the bound for the second moment:

$$\mathbb{E}_\xi \left[\|g(x; \xi)\|^2 \right] \quad \leq \quad \sigma^2 + \|\nabla f(x)\|^2. \tag{2.30}$$

2.5.2 Stochastic Algorithm

We consider the following algorithm, which is gradient descent with substituted stochastic gradient instead of the full one. It is also often called *stochastic gradient descent* (SGD).

Algorithm 2.3: *Stochastic Gradient Method.*

Initialization: $x_0 \in \mathbb{R}^n$, regularization parameter $M > 0$, number of iterations $K \geq 1$.

For $k = 0 \dots K - 1$ **iterate:**

1. Sample ξ_k.
2. Compute stochastic gradient $g_k := g(x_k; \xi_k)$.
3. Update $x_{k+1} := x_k - \frac{1}{M} g_k$.

Sample $j \in \{0, \dots, K - 1\}$ uniformly at random and **return** $\bar{x}_K$.

We consider a constant parameter $M > 0$ and the key questions is *how we choose it?*

Note that we do not have a clear stopping condition because we lack access to $\|\nabla f(x)\|$ or even to the function value $f(x)$. Instead, we fix the number of iterations $K \geq 0$ for the method to perform and in the end return one of the points from the past, sampled uniformly at random.

2.5.3 Convergence Analysis

First, we investigate the progress of one random step.

Proposition 2.5.4. *Let $M > 0$. Then, for $x_{k+1} = x_k - \frac{1}{M} g_k$ with $g_k = g(x_k; \xi_k)$ we have*

$$\mathbb{E}_{\xi_k}\Big[f(x_k) - f(x_{k+1})\Big] \;\geq\; \tfrac{1}{M}\|\nabla f(x_k)\|^2 \cdot \Big(1 - \tfrac{L}{2M}\Big) - \tfrac{L}{2M^2}\sigma^2. \tag{2.31}$$

Proof. Using Lipschitzness of the gradient, we have

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \tfrac{L}{2}\|x_{k+1} - x_k\|^2 \\ &= f(x_k) - \tfrac{1}{M}\langle \nabla f(x_k), g_k\rangle + \tfrac{L}{2M^2}\|g_k\|^2. \end{aligned}$$

Rearranging the terms and taking the expectation, we obtain

$$\begin{aligned} \mathbb{E}_{\xi_k}\Big[f(x_k) - f(x_{k+1})\Big] &\geq \mathbb{E}_{\xi_k}\Big[\tfrac{1}{M}\langle \nabla f(x_k), g_k\rangle - \tfrac{L}{2M^2}\|g_k\|^2\Big] \\ &= \tfrac{1}{M}\|\nabla f(x_k)\|^2 - \tfrac{L}{2M^2}\mathbb{E}_{\xi_k}\Big[\|g_k\|^2\Big] \\ &\geq \tfrac{1}{M}\|\nabla f(x_k)\|^2 - \tfrac{L}{2M^2}\|\nabla f(x_k)\|^2 - \tfrac{L}{2M^2}\sigma^2 \\ &= \tfrac{1}{M}\|\nabla f(x_k)\|^2 \cdot \Big(1 - \tfrac{L}{2M}\Big) - \tfrac{L}{2M^2}\sigma^2, \end{aligned}$$

which is the required bound. □

Corollary 2.5.5. *Let $M \geq L$. We obtain*

$$\mathbb{E}_{\xi_k}\Big[f(x_k) - f(x_{k+1})\Big] \;\geq\; \tfrac{1}{2M}\|\nabla f(x_k)\|^2 - \tfrac{L}{2M^2}\sigma^2. \tag{2.32}$$

In case of no randomness ($\sigma = 0$), we recover the progress bound for the deterministic gradient method. However, when $\sigma > 0$, we cannot longer guarantee a positive progress of each iteration.

Now, we want to telescope (2.31) across K iterations to obtain the convergence rate.

Theorem 2.5.6. *Let $M \geq L$. Consider $K \geq 1$ iterations of Algorithm 2.3. Then,*

$$\tfrac{2M(f(x_0) - f^\star)}{K} + \tfrac{L}{M}\sigma^2 \;\geq\; \mathbb{E}\Big[\|\nabla f(\bar{x}_K)\|^2\Big]. \tag{2.33}$$

Proof. We denote by $\mathbb{E}[\cdot]$ the expectation w.r.t all $\{\xi_0, \ldots, \xi_{K-1}\}$ and the random choice of j for the output $\bar{x}_K$ and $\mathbb{E}_\xi[\cdot]$ with respect to all $\{\xi_0, \ldots, \xi_{K-1}\}$ only. Taking it for the both left and right hand sides of (2.32), we get

$$\mathbb{E}_\xi\Big[f(x_k) - f(x_{k+1})\Big] \;\geq\; \tfrac{1}{2M}\mathbb{E}_\xi\Big[\|\nabla f(x_k)\|^2\Big] - \tfrac{L}{2M^2}\sigma^2.$$

Telescoping this bound for the first K iterations, we get

$$\begin{aligned} f(x_0) - f^\star &\geq f(x_0) - \mathbb{E}_\xi[f(x_k)] \;\geq\; \tfrac{1}{2M}\sum_{i=0}^{K-1}\mathbb{E}_\xi\big[\,\|\nabla f(x_i)\|^2\,\big] - \tfrac{L}{M^2}\sigma^2 K \\ &= \tfrac{1}{2M}\mathbb{E}\Big[\|\nabla f(\bar{x}_K)\|^2\Big] - \tfrac{L}{M^2}\sigma^2 K. \end{aligned}$$

Rearranging the terms we obtain the required bound (2.33). □

2.5.4 Stepsize Tuning

Now, we need to choose $M > 0$ and $K \geq 1$ to ensure the following guarantee:

$$\mathbb{E}\Big[\|\nabla f(\bar{x}_K)\|^2\Big] \;\leq\; \varepsilon^2.$$

For that, using (2.33) it is enough to ensure that

1. $\frac{2M(f(x_0)-f^\star)}{K} \leq \frac{\varepsilon^2}{2}$ and
2. $\frac{L}{M}\sigma^2 \leq \frac{\varepsilon^2}{2}$.

The last inequality, together with our condition $M \geq L$ suggest the following choice of the regularization parameter:

$$\boxed{M \;:=\; L \cdot \max\Big\{1,\; \tfrac{2\sigma^2}{\varepsilon^2}\Big\}.} \tag{2.34}$$

Then, to ensure the first inequality, it is enough to choose the number of iterations sufficiently large:

$$\boxed{K \;:=\; 1 + \Big\lfloor \tfrac{4MF_0}{\varepsilon^2} \Big\rfloor.} \tag{2.35}$$

Combining these two choices together, and using Jensen's inequality, that is $(\mathbb{E}[\tau])^2 \leq \mathbb{E}[\tau^2]$, we obtain the following complexity bound.

Corollary 2.5.7. *To find a random point $\bar{x} \in \mathbb{R}^n$ such that $\mathbb{E}\Big[\|\nabla f(\bar{x})\|\Big] \leq \varepsilon$, it is enough to perform*

$$K \;=\; O\Big(L(f(x_0) - f^\star) \cdot \Big[\tfrac{1}{\varepsilon^2} + \tfrac{\sigma^2}{\varepsilon^4}\Big]\Big) \tag{2.36}$$

stochastic first-order oracle calls.

The upper bound (2.36) is matched, up to a numerical constant, by a lower complexity bound for our problem class in both for stochastic and deterministic cases. Consequently, the gradient method is *optimal* for finding a stationary point for functions with Lipschitz continuous gradients.

2.6 Exercises

Exercise 2.6.1. Let $\|\cdot\|$ be an arbitrary fixed norm on $\mathbb{R}^n$. Consider the function, $\rho : \mathbb{R}^n \to \mathbb{R}$, defined by

$$\rho(s) \;:=\; \max_{x \in \mathbb{R}^n \,:\, \|x\| \leq 1} \langle s, x \rangle.$$

- Show that $\rho(s) = \max_{x \in \mathbb{R}^n \,:\, \|x\|=1} \langle s, x \rangle$, i.e. the maximum of the linear form $\langle s, \cdot \rangle$ over the unit ball is always achieved on its boundary (the unit sphere).

- Show that $\rho(s)$ satisfies the standard axioms of a norm:

  1. *Positive definiteness:* $\rho(s) \geq 0$ for all $s \in \mathbb{R}^n$, and $\rho(s) = 0 \;\Leftrightarrow\; s = 0$.
  2. *Homogeneity:* $\rho(ts) = |t|\rho(s)$ for any $s \in \mathbb{R}^n$ and $t \in \mathbb{R}$.
  3. *Triangle inequality:* $\rho(s + y) \leq \rho(s) + \rho(y)$ for all $s, y \in \mathbb{R}^n$.

  Therefore, the *dual norm* $\|s\|_* := \rho(s)$ is a well-defined norm on $\mathbb{R}^n$.

Exercise 2.6.2. Consider the logistic loss function, $\ell(t) = \ln(1 + e^t)$ for $t \in \mathbb{R}$. Show that $\ell(\cdot)$ is convex and that its derivative $\ell'(\cdot)$ is Lipschitz continuous on $\mathbb{R}$. Compute the smallest possible Lipschitz constant $L > 0$ (w.r.t. the standard Euclidean norm, i.e., the absolute value $|\cdot|$ on $\mathbb{R}$).

Exercise 2.6.3. Consider the following function, often called *soft-max* or *log-sum-exp*:

$$f(x) \;=\; \ln\Big(\sum_{i=1}^{n} \exp(x^{(i)})\Big), \qquad x \in \mathbb{R}^n.$$

Show that $0 \preceq \nabla^2 f(x) \preceq I$, for any $x \in \mathbb{R}^n$, where $I \in \mathbb{R}^{n\times n}$ is the identity matrix; i.e., all eigenvalues of $\nabla^2 f(x)$ lie in the interval $[0, 1]$. Therefore, f is convex and has a Lipschitz continuous gradient with constant $L = 1$ (w.r.t. the Euclidean norm).

Exercise 2.6.4. Let $B = B^\top \succ 0$ be a symmetric positive definite matrix, $B \in \mathbb{R}^{n\times n}$. Consider the following generalized Euclidean norm: $\|x\| := \langle Bx, x\rangle^{1/2}$ for $x \in \mathbb{R}^n$.

- Show that the dual norm is given by $\|s\|_* = \langle s, B^{-1}s\rangle^{1/2}$ for $s \in \mathbb{R}^n$.
- Provide an explicit formula for the step x^+ of the gradient method with respect to this norm, for $M > 0$:
$$x^+ \;:=\; \operatorname*{argmin}_{y\in\mathbb{R}^n}\Big\{f(x) + \langle \nabla f(x), y - x\rangle + \tfrac{M}{2}\|y - x\|^2\Big\}.$$

Exercise 2.6.5. Let $f : \mathbb{R}^n \to \mathbb{R}$ have a Lipschitz continuous gradient with constant $L_f > 0$ with respect to the standard Euclidean norm in $\mathbb{R}^n$. Let $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$ be fixed parameters, and define the function (an affine change of variable):

$$F(y) \;=\; f(Ay + b), \qquad y \in \mathbb{R}^m.$$

- Consider the standard Euclidean norm in $\mathbb{R}^m$, express the Lipschitz constant L_F of the gradient ∇F in terms of L_f and given parameters.
- Fix $B = A^\top A$ and assume $B \succ 0$. Consider the generalized Euclidean norm in $\mathbb{R}^m$ defined by $\|y\| := \langle By, y\rangle^{1/2}$. Show that the Lipschitz constant of ∇F with respect to this norm is exactly L_f.

Exercise 2.6.6. Consider $f : \mathbb{S}^n \to \mathbb{R}$ where $\mathbb{S}^n = \{X \in \mathbb{R}^{n\times n} : X = X^\top\}$ is the space of symmetric $n \times n$ matrices. We equip $\mathbb{S}^n$ with the spectral norm:

$$\|X\| \;:=\; \max_{u\in\mathbb{R}^n:\|u\|_2=1} |\langle Xu, u\rangle| \;=\; \max_{1\le i\le n} |\lambda^{(i)}(X)| \;=:\; \|\lambda(X)\|_\infty, \qquad X \in \mathbb{S}^n,$$

where $\lambda(X) = (\lambda^{(1)}(X), \ldots, \lambda^{(n)}(X)) \in \mathbb{R}^n$ is the vector of eigenvalues of X.

- Compute the dual norm $\|Y\|_* := \max_{X\in\mathbb{S}^n:\|X\|\le 1} \langle Y, X\rangle$, where $\langle Y, X\rangle = \operatorname{tr}(YX)$ is the standard inner product on $\mathbb{S}^n$.
- Provide an explicit formula for a step X^+ of the gradient method with respect to this norm, for $M > 0$:
$$X^+ \;:=\; \operatorname*{argmin}_{Y\in\mathbb{S}^n}\Big\{f(X) + \langle \nabla f(X), Y - X\rangle + \tfrac{M}{2}\|Y - X\|^2\Big\}.$$

Exercise 2.6.7. Fix the standard Euclidean norm $\|\cdot\|_2$ on $\mathbb{R}^n$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth function and denote by $L > 0$ the Lipschitz constant of $\nabla f(\cdot)$. Assume f is bounded from below: $f^\star := \inf_{x \in \mathbb{R}^n} f(x) > -\infty$. Let $e_1, \ldots, e_n \in \mathbb{R}^n$ be the standard basis. Consider the following randomized algorithm for finding a stationary point of f:

Algorithm 2.4: *Coordinate Descent.*

Initialization: $x_0 \in \mathbb{R}^n$, the Lipschitz constant $L > 0$, number of iterations $K \geq 1$.

For $k = 0 \ldots K-1$ **iterate:**

1. Sample coordinate $i_k \in \{1, \ldots, n\}$ uniformly at random.
2. Compute the i_k-th partial derivative and form the vector $g_k := \frac{\partial f}{\partial x^{(i_k)}}(x_k) \cdot e_{i_k} \in \mathbb{R}^n$.
3. Update $x_{k+1} := x_k - \frac{1}{L} g_k$.

Sample $j \in \{0, \ldots, K-1\}$ uniformly at random and **return** $\bar{x}_K := x_j$.

We denote by $\bar{x}_K$ the result of the algorithm after running for $K \geq 1$ iterations. Note that $\bar{x}_K$ is a random vector. We let $\mathbb{E}[\cdot]$ denote the expectation with respect to all randomness in the method.

- Show the following progress of each step, $0 \leq k \leq K-1$:
$$\mathbb{E}\Big[f(x_k) - f(x_{k+1}) \,|\, x_k\Big] \quad \geq \quad \tfrac{1}{2nL}\|\nabla f(x_k)\|_2^2,$$
where $\mathbb{E}[\cdot \,|\, x_k]$ is the conditional expectation, when point x_k is fixed.
- Show that
$$\mathbb{E}\Big[\|\nabla f(\bar{x}_K)\|_2^2\Big] \quad \leq \quad \tfrac{2nL(f(x_0)-f^\star)}{K}.$$
- Show that for a given $\varepsilon > 0$, it is enough to set $K := \lfloor \frac{2nL(f(x_0)-f^\star)}{\varepsilon^2} \rfloor + 1$ in order to obtain $\mathbb{E}[\|\nabla f(\bar{x}_K)\|_2] \leq \varepsilon$. Compare this complexity with that one for the standard gradient descent.

Exercise 2.6.8. Assume that the gradient $\nabla f(\cdot)$ of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is Hölder continuous of degree $0 \leq \nu \leq 1$ with constant $H_\nu > 0$, with respect to the standard Euclidean norm:
$$\|\nabla f(y) - \nabla f(x)\|_2 \quad \leq \quad H_\nu \|y - x\|_2^\nu, \qquad x, y \in \mathbb{R}^n.$$
- Show that, for any $x, y \in \mathbb{R}^n$:
$$f(y) \quad \leq \quad f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{H_\nu}{1+\nu}\|y - x\|_2^{1+\nu}. \tag{2.37}$$
- Consider the method based on minimizing the global upper bound in (2.37), starting from some initialization $x_0 \in \mathbb{R}^n$, for $k \geq 0$:
$$x_{k+1} \quad = \quad \operatorname*{argmin}_{y \in \mathbb{R}^n} \Big[f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \tfrac{H_\nu}{1+\nu}\|y - x_k\|_2^{1+\nu}\Big]. \tag{2.38}$$
Show that each step can be written in the form $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$ and provide an explicit formula for the step-size $\alpha_k > 0$.
- What is the first-order oracle complexity of this method to find a stationary point $\bar{x}$ such that $\|\nabla f(\bar{x})\|_2 \leq \varepsilon$, for $0 < \nu < 1$?

Literature

For additional reading on stochastic gradient methods, see [13] and [24]. For the lower complexity bound on finding stationary points of smooth functions, see [5], which establishes that the bound (2.36) is *optimal* for both deterministic and stochastic optimization within our problem class.

3. Minimizing Differentiable Convex Functions

In this part of the course, we study the convergence of gradient-based methods on differentiable convex functions. It turns out that convexity is a natural sufficient condition (and, in a certain formal sense, a necessary condition) to ensure that a stationary point is the global minimum. Moreover, due to convexity, we obtain faster rates for the basic gradient method and encounter the phenomenon of *acceleration*. We discuss Polyak's *heavy ball method*, or the gradient method with *momentum*, with its optimal rate on quadratic functions, and Nesterov's *fast gradient method*, which is optimal for much broader classes of smooth convex functions.

3.1 Convex Functions . . . 37
3.1.1 Motivation . . . 37
3.1.2 Univariate Convex Functions . . . 37
3.1.3 Maximizing Convex Functions . . . 38
3.1.4 Multivariate Convex Functions . . . 39
3.1.5 Differentiable Convex Functions . . . 40
3.1.6 Global Optimality . . . 41
3.2 Convergence Rates of Gradient Method . . . 42
3.2.1 Smooth Convex Functions . . . 42
3.2.2 Convergence Rate . . . 43
3.2.3 Minimizing Gradient Norm . . . 45
3.2.4 Strongly Convex Smooth Functions . . . 46
3.3 Polyak's Heavy Ball Method . . . 48
3.3.1 Quadratic Functions . . . 48
3.3.2 Heavy Ball Method . . . 49
3.3.3 Optimal Complexity of the Heavy Ball Method . . . 50
3.4 Lower Bound for Smooth Convex Optimization . . . 53
3.4.1 Lower Bound for the Linear Span Methods . . . 54
3.4.2 Strongly Convex Minimization . . . 58
3.5 Nesterov's Fast Gradient Method . . . 59
3.5.1 Review: Strongly Convex Functions . . . 59
3.5.2 Analysis of Similar Triangles . . . 59
3.5.3 Fast Gradient Method . . . 62
3.5.4 The Parameter Choice . . . 63
3.5.5 Strongly Convex Optimization . . . 64
3.6 Applications: Machine Learning . . . 65
3.6.1 Generalized Linear Models . . . 65
3.6.2 Nonlinear Models: Neural Networks . . . 68
3.7 Fully Composite Problems . . . 69
3.7.1 Motivation . . . 69
3.7.2 Fully Composite Formulation . . . 69
3.7.3 Composite Fast Gradient Method . . . 71
3.7.4 Analysis . . . 72
3.8 Exercises . . . 73

3.1 Convex Functions

3.1.1 Motivation

We have studied the following problem classes so far.

1. **Global minimization of smooth functions:**

$$\mathcal{F} \;=\; \Big\{ f : \mathbb{R}^n \to \mathbb{R} \text{ s.t. } f \text{ is continuous / smooth} \Big\}.$$

   The goal: finding a *global solution* $\bar{x}$ s.t. $f(\bar{x}) - f^\star \leq \varepsilon$. We saw that these problems are too hard to be efficiently solvable in general. The optimal algorithm was the simplest grid search.

2. **Finding stationary points of smooth functions.** The same class $\mathcal{F}$, but the goal is much less ambitious: finding $\bar{x}$ s.t. $\|\nabla f(\bar{x})\| \leq \varepsilon$. We have analyzed two algorithms for this class: gradient method and stochastic gradient method, which both possess a dimension-free complexity bounds.

Now, we have two options. *Option 1:* to try finding something in between these classes, which is a difficult (but interesting) path. By using higher-order smoothness, we are able to achieve better complexities to get a stationary point than that one of the gradient method. However, checking whether a point is a local minimum, local maximum, or a saddle point might be NP-hard in general.

Option 2: to find a smaller problem class $\mathcal{F}' \subset \mathcal{F}$ for which the initial goal of finding a global solution is feasible. For example, we want the following property to hold: whenever it holds $\nabla f(\bar{x}) = 0$ then $\bar{x}$ is a global solution, so every stationary point is a global minimum. It appears that such path essentially leads us to *convex functions*.

Convex functions play a central role in optimization theory, as they provide examples of a broad range of globally solvable problem classes. We review the basic facts about convex functions that are most useful for the analysis of optimization algorithms. Convexity is an important concept in mathematics with a rich, well-developed theory, and a wide variety of applications across different domains besides optimization. There are plenty of excellent courses and textbooks on convex analysis, which we recommend for further reading.

3.1.2 Univariate Convex Functions

Let $f : \mathbb{R} \to \mathbb{R}$ be a *continuous* function, as we almost always assume in this course.[1] We say that f is *convex* if for any two points $x, y \in \mathbb{R}$:

$$f\big(\tfrac{x+y}{2}\big) \;\leq\; \tfrac{1}{2}\big(f(x) + f(y)\big). \tag{3.1}$$

That is, the value of the function at the midpoint of any interval never exceeds the average of the values at the endpoints.

This simple property, coupled with the fact that we require it *for any two points* $x, y \in \mathbb{R}$, leads to many consequences. We leave the proof of the following facts to the reader.

Proposition 3.1.1. *For any $x, y \in \mathbb{R}$ and $0 \leq \lambda \leq 1$, it holds*

$$f(\lambda x + (1-\lambda)y) \;\leq\; \lambda f(x) + (1-\lambda) f(y). \tag{3.2}$$

[1] Otherwise, it would not be called Continuous Optimization.

Thus, *midpoint* convexity (3.1) leads to a similar inequality along the *entire* segment when f is continuous. Geometrically, this means that the chord connecting any two points $(x, f(x))$ and $(y, f(y))$ always lies on or above the graph of f. The value $\lambda x + (1-\lambda)y$ is called a *convex combination* of x and y for $0 \le \lambda \le 1$.

Functions that satisfy (3.1) but not (3.2) are pathologically rare cases; in fact, these definitions are equivalent as soon as the function is measurable (which is true for any continuous function). Moreover, it can be shown that convex functions in the sense of (3.2) are locally *Lipschitz continuous* on the interior of their domain.

Proposition 3.1.2 (Jensen's inequality)**.** *Let $x_1, \dots, x_N \in \mathbb{R}$ be a finite set of points and let $\lambda \in \Delta_N$ be from the standard simplex, $\Delta_N = \{\lambda \in \mathbb{R}^N_+ \,:\, \langle e, \lambda\rangle = 1\}$, where $e \in \mathbb{R}^N$ is the vector of all ones. Then*

$$f\Big(\sum_{i=1}^N \lambda_i x_i\Big) \;\le\; \sum_{i=1}^N \lambda_i f(x_i). \tag{3.3}$$

Inequality (3.3) can be generalized beyond discrete distributions, by taking a limit to an infinite amount of points. The following inequality is very useful when analyzing stochastic algorithms.

Proposition 3.1.3 (Jensen's inequality for expectations)**.** *Let ξ be a random variable taking values in $\mathbb{R}$. Then*

$$f\big(\mathbb{E}[\xi]\big) \;\le\; \mathbb{E}\big[f(\xi)\big]. \tag{3.4}$$

Of course, it is straightforward to see that (3.4) $\Rightarrow$ (3.3) $\Rightarrow$ (3.2) $\Rightarrow$ (3.1). What is more interesting is that the reverse implications also hold, making them equivalent.

We also say that f is *concave*, if $-f$ is convex.

3.1.3 Maximizing Convex Functions

Convex functions are meant to be minimized. But what if we try to maximize them? It is easy to see that a maximum of a convex function over a set is always at the boundary. Interestingly, this property applied to a function with *all* affine perturbations *defines convexity*. In other words, if we "tilt" a convex function by adding to it a linear slope, we can force the highest point to lie at one of the endpoints of any segment, and this is the characteristic property of convexity.

Theorem 3.1.4. *The following conditions are equivalent:*

- *$f : \mathbb{R} \to \mathbb{R}$ is convex.*
- *For any segment $[x, y] \in \mathbb{R}$ and for any affine function $t \mapsto at + b$ it holds*

$$\max_{t\in[x,y]} \Big\{ f(t) - at - b \Big\} \;=\; \max\Big\{ f(x) - ax - b,\; f(y) - ay - b \Big\}. \tag{3.5}$$

Proof. Let f be convex. Then, since any $t \in [x, y]$ can be represented as the convex combination: $t = \lambda x + (1-\lambda)y$, for some $0 \le \lambda \le 1$, we have

$$\begin{aligned} f(t) + at + b &\overset{(3.2)}{\le} \lambda\Big[f(x) - ax - b\Big] + (1-\lambda)\Big[f(y) - ay - b\Big] \\ &\le \max\Big\{ f(x) - ax - b,\; f(y) - ay - b \Big\}, \end{aligned}$$

which is (3.5).

Now assume that (3.5) holds for any $a, b \in \mathbb{R}$. Let us verify that for any two fixed $x, y \in \mathbb{R}$, $x \neq y$, and for all $0 \leq \lambda \leq 1$ it holds:

$$\begin{aligned} & f(\lambda x + (1-\lambda)y) \;\; \leq \;\; \lambda f(x) + (1-\lambda) f(y) \\ & \Leftrightarrow \\ & f(y + \lambda(x-y)) - f(y) - \lambda(f(x) - f(y)) \;\; \leq \;\; 0 \\ & \Leftrightarrow \\ & f(t) - f(y) - (t-y) \cdot \tfrac{f(x)-f(y)}{x-y} \;\; \leq \;\; 0, \end{aligned} \tag{3.6}$$

where $t \equiv y + \lambda(x-y) \in [x, y]$. Now, we set $a := \frac{f(x)-f(y)}{x-y}$ and $b := f(y) - y \cdot a$, and consider the following perturbation of f by an affine function:

$$\varphi(t) \;\; := \;\; f(t) - at - b \;\; = \;\; f(t) - f(y) - (t-y) \cdot \tfrac{f(x)-f(y)}{x-y},$$

and by assumption (3.5), we have

$$\varphi(t) \;\; \leq \;\; \max\Big\{\varphi(x),\, \varphi(y)\Big\} \;\; = \;\; 0,$$

which proves (3.6). □

Thus, we see that *affine functions* play a fundamental role in the theory of convexity. Note that affine functions are uniquely identified as those and only those that are simultaneously convex and concave.

Proposition 3.1.5. *The following conditions are equivalent:*

- *f is both convex and concave, i.e. f preserves convex combination, for any $x, y \in \mathbb{R}$:*

$$f(\lambda x + (1-\lambda)y) \;\; = \;\; \lambda f(x) + (1-\lambda) f(y), \qquad 0 \leq \lambda \leq 1.$$

- *f is affine, i.e. $f(t) = at + b$, for some $a, b \in \mathbb{R}$.*

Exercise 3.1.1. Prove Proposition 3.1.5.

3.1.4 Multivariate Convex Functions

A multivariate function $f : \mathbb{R}^n \to \mathbb{R}$ is called *convex* if its restriction to any segment is convex. So, for any two $x, y \in \mathbb{R}^n$ and $0 \leq \lambda \leq 1$ it holds:

$$f(\lambda x + (1-\lambda)y) \;\; \leq \;\; \lambda f(x) + (1-\lambda) f(y). \tag{3.7}$$

All the properties discussed for the univariate case are naturally inherited by general convex functions. As previously, we say that the function f is *concave* if $-f$ is convex. The only functions that are both convex and concave are *affine functions*:

$$f(x) \;\; = \;\; \langle a, x \rangle + b, \qquad x \in \mathbb{R}^n,$$

for some $a \in \mathbb{R}^n, b \in \mathbb{R}$.

Thus, we know that the maximum of a convex function over any compact set $Q \subset \mathbb{R}^n$ is achieved on its boundary:

$$\max_{x \in Q} f(x) \quad = \quad \max_{x \in \partial Q} f(x),$$

and the same applies to the minimum of a concave function. In particular, we conclude that the minimum of a *linear function* over any compact set is always achieved on the boundary:

$$\min_{x \in Q} \langle a, x \rangle \quad = \quad \min_{x \in \partial Q} \langle a, x \rangle.$$

When Q is non-compact, we must ensure that the minimum actually exists. This fundamental fact is used in linear programming (where Q is a polyhedron and the minimum is attained at a vertex) and in more general settings such as semidefinite programming (where Q is the intersection of the cone of positive semidefinite matrices with an affine set).

3.1.5 Differentiable Convex Functions

For now, let us assume that $f : \mathbb{R}^n \to \mathbb{R}$ is several times differentiable convex function defined on the whole space (unconstrained minimization). We consider more general non-differentiable convex functions later in the course. We have the following important inequalities, that serve as equivalent definitions of convexity.

Theorem 3.1.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable. Then, the following statements are equivalent:*

- *f is convex* (3.7).
- *The linear approximation of f is its global lower bound:*

$$f(y) \quad \geq \quad f(x) + \langle \nabla f(x), y - x \rangle, \qquad x, y \in \mathbb{R}^n. \tag{3.8}$$

- *The gradient mapping is monotone:*

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \quad \geq \quad 0, \qquad x, y \in \mathbb{R}^n. \tag{3.9}$$

- *The Hessian is positive semidefinite:*

$$\nabla^2 f(x) \quad \succeq \quad 0, \qquad x \in \mathbb{R}^n. \tag{3.10}$$

Proof. Let f be convex. Then, for any $x, y \in \mathbb{R}^n$ and $0 < \alpha < 1$ we have

$$\begin{aligned} f((1-\alpha)x + \alpha y) \quad &\leq \quad (1-\alpha) f(x) + \alpha f(y) \\ \Leftrightarrow \\ f(y) \quad &\geq \quad f(x) + \tfrac{1}{\alpha}\big(f(x + \alpha(y-x)) - f(x)\big). \end{aligned}$$

Taking the limit $\alpha \to 0$ gives (3.8).

At the same time, substituting into (3.8) pairs of points (x, x_α) and (y, x_α), where $x_\alpha = (1-\alpha)x + \alpha y$, we get:

$$\begin{aligned} f(x) \quad &\geq \quad f(x_\alpha) + \langle \nabla f(x_\alpha), x - x_\alpha \rangle \quad = \quad f(x_\alpha) + \alpha \langle \nabla f(x_\alpha), y - x \rangle, \\ f(y) \quad &\geq \quad f(x_\alpha) + \langle \nabla f(x_\alpha), y - x_\alpha \rangle \quad = \quad f(x_\alpha) + (1-\alpha) \langle \nabla f(x_\alpha), x - y \rangle. \end{aligned}$$

Multiplying the first by $(1-\alpha)$ and the second by α, gives (3.7) after summation. Thus, we showed that (3.7) and (3.8) are equivalent.

To obtain (3.9), we only need to sum up a pair of inequalities (3.8), swapping the roles of x and y.

To show (3.10) we use the definition of the Hessian. For an arbitrary unit direction $h \in \mathbb{R}^n$, $\|h\| = 1$, and a sufficiently small $\varepsilon > 0$, we have:

$$\nabla f(x + \varepsilon h) \;\; = \;\; \nabla f(x) + \varepsilon \nabla^2 f(x) h + o(\varepsilon).$$

Hence, multiplying this vector equation by h and rearranging the terms, we get:

$$\langle \nabla^2 f(x) h, h \rangle \;\; = \;\; \tfrac{1}{\varepsilon^2} \langle \nabla f(x + \varepsilon h) - \nabla f(x), \varepsilon h \rangle + o(1) \;\; \overset{(3.9)}{\geq} \;\; o(1).$$

Taking the limit $\varepsilon \to 0$ proves (3.10).

Finally, using Taylor's formula, we get that (3.10) implies (3.8):

$$f(y) - f(x) - \langle \nabla f(x), y - x \rangle \;\; = \;\; \int\limits_0^1 (1-\tau) \langle \nabla^2 f(x + \tau(y-x))(y-x), y - x \rangle d\tau \;\; \overset{(3.10)}{\geq} \;\; 0,$$

which completes the proof. □

Second-order condition $\nabla^2 f(x) \succeq 0$ is useful for checking whether a function is convex.

Example 3.1.7. The following univariate functions are convex:

- $f(x) = e^x$
- $f(x) = -\ln x$, for $x > 0$
- $f(x) = x \ln x$, for $x > 0$
- $f(x) = \ln(1 + e^x)$
- $f(x) = |x|^p$, for $p \geq 1$

3.1.6 Global Optimality

We come to the following important implication in convex optimization.

Corollary 3.1.8. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable and convex. Then,*

$$\nabla f(x^\star) \;\; = \;\; 0 \qquad \Leftrightarrow \qquad x^\star \text{ is a global minimum.} \tag{3.11}$$

Proof. Indeed, substituting $x := x^\star$ into (3.8) gives

$$f(y) \;\; \geq \;\; f(x^\star), \qquad y \in \mathbb{R}^n.$$

The other direction is already known as *optimality condition* for local minimum. □

It is interesting that seeking for a functional class that satisfy (3.11), together with some simple natural conditions, we necessarily comes to the class of *convex functions*.

Theorem 3.1.9. *Let $\mathcal{F} \subset \{f : \mathbb{R}^n \to \mathbb{R},$ differentiable$\}$ be a <u>maximal</u> class of functions such that:*

1. *For any $f \in \mathcal{F}$: $\nabla f(\bar{x}) = 0 \Rightarrow \bar{x}$ is a global minimum.*

2. *If $f_1, f_2 \in \mathcal{F}$ then $f_1 + f_2 \in \mathcal{F}$.*

3. *Any affine function belongs to our problem class: $\langle a, x\rangle + b \in \mathcal{F}$, for any $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.*

Then, $\mathcal{F} \equiv$ convex differentiable functions.

Proof. Let $f \in \mathcal{F}$. Fix $x \in \mathbb{R}^n$. Denote

$$\varphi(y) \;=\; f(y) - \langle \nabla f(x), y\rangle \;\in\; \mathcal{F}.$$

Then,

$$\nabla\varphi(y) \;=\; \nabla f(y) - \nabla f(x).$$

Hence $\nabla\varphi(x) = 0$ and x is a global minimum of φ:

$$\varphi(y) = f(y) - \langle \nabla f(x), y\rangle \;\geq\; \varphi(x) = f(x) - \langle \nabla f(x), x\rangle, \qquad \forall y \in \mathbb{R}^n.$$

This means that $f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle$ for any $x, y \in \mathbb{R}^n$. Hence f is convex. □

3.2 Convergence Rates of Gradient Method

3.2.1 Smooth Convex Functions

Now, we couple both our assumptions together. We assume that $f : \mathbb{R}^n \to \mathbb{R}$ is both convex and smooth, i.e. its gradient is Lipschitz continuous, for any $x, y \in \mathbb{R}^n$: $\|\nabla f(y) - \nabla f(x)\|_* \leq L\|y - x\|$.

We have the following theorem.

Theorem 3.2.1. *The following conditions are equivalent:*

1. *f is convex and smooth.*

2. *For any $x, y \in \mathbb{R}^n$, it hods: $0 \leq f(y) - f(x) - \langle \nabla f(x), y - x\rangle \leq \frac{L}{2}\|y - x\|^2$.*

3. *For any $x, y \in \mathbb{R}^n$, it hods: $0 \leq \langle \nabla f(y) - \nabla f(x), y - x\rangle \leq L\|y - x\|^2$.*

4. *For any $x, h \in \mathbb{R}^n$, it hods: $0 \leq \langle \nabla^2 f(x)h, h\rangle \leq L\|h\|^2$ (for twice continuously differentiable functions).*

5. *For any $x, y \in \mathbb{R}^n$, it holds: $f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2L}\|\nabla f(y) - \nabla f(x)\|_*^2$.*

6. *For any $x, y \in \mathbb{R}^n$, it holds: $\langle \nabla f(y) - \nabla f(x), y - x\rangle \geq \frac{1}{L}\|\nabla f(y) - \nabla f(x)\|_*^2$.*

Proof. *1* $\Rightarrow$ *2* $\Rightarrow$ *3* $\Rightarrow$ *4* are trivial. At the same time, *1* follows from *4* by already established second-order characterizations of convexity and smoothness (see Theorem 3.1.6 and Theorem 2.2.4).

Let us prove *5*. First consider $x = x^\star$, which is

$$f(y) - f^\star \;\geq\; \tfrac{1}{2L}\|\nabla f(y)\|_*^2. \tag{3.12}$$

Note that such progress is satisfied for one gradient step $y \mapsto y^+(L)$ (Proposition 2.3.3)! Hence, since $f^\star \leq f(y^+(L))$, (3.12) is established for any $y \in \mathbb{R}^n$.

For an arbitrary x, define the tilted function $\varphi(y) = f(y) - \langle \nabla f(x), y \rangle$ that has a minimum at x. Clearly, $\varphi(\cdot)$ belongs to our problem class, and we already established inequality (3.12):

$$\varphi(y) - \varphi^\star \quad \geq \quad \tfrac{1}{2L} \|\nabla \varphi(y)\|_*^2.$$

Substituting the expression for $\varphi(\cdot)$, we get

$$f(y) - \langle \nabla f(x), y \rangle + f(x) + \langle \nabla f(x), x \rangle \quad \geq \quad \tfrac{1}{2L} \|\nabla f(y) - \nabla f(x)\|_*^2,$$

which proves *5*. To get *6* from 5 we just need to sum it up two times, swapping the role of x and y.

Finally, having *6*, we immediately conclude that f is convex, and by Cauchy-Schwarz inequality,

$$\tfrac{1}{L} \|\nabla f(y) - \nabla f(x)\|_*^2 \quad \leq \quad \langle \nabla f(y) - \nabla f(x), y - x \rangle \quad \leq \quad \|\nabla f(y) - \nabla f(x)\|_* \cdot \|y - x\|,$$

which proves that f is smooth. □

3.2.2 Convergence Rate

Let us study the convergence of the gradient method on our new problem class. By the previous analysis, we have the following progress of one step, using the constant step size (see Proposition 2.3.3):

$$f(x_k) - f(x_{k+1}) \quad \geq \quad \tfrac{1}{2L} \|\nabla f(x_k)\|_*^2. \tag{3.13}$$

At the same time, by convexity, we have

$$f^\star \quad \geq \quad f(x_k) + \langle \nabla f(x_k), x^\star - x_k \rangle,$$

or, rearranging the terms,

$$\begin{aligned} F_k \quad &:= \quad f(x_k) - f^\star \quad \leq \quad \langle \nabla f(x_k), x_k - x^\star \rangle \\ &\leq \quad \|\nabla f(x_k)\|_* \cdot \|x_k - x^\star\| \quad \leq \quad \|\nabla f(x_k)\|_* \cdot D, \end{aligned} \tag{3.14}$$

where we denote by D a global bound for all the iterates:

$$\|x_k - x^\star\| \quad \leq \quad D, \qquad k \geq 0. \tag{3.15}$$

For example, noticing that the method is *monotone* in function value (see (3.13)), we conclude that all iterates belong to the initial sublevel set:

$$x_k \quad \in \quad \mathcal{S}_0 \quad := \quad \Big\{ x \in \mathbb{R}^n \;:\; f(x) \leq f(x_0) \Big\}.$$

Hence, denoting

$$D \quad := \quad \sup_{x \in \mathcal{S}_0} \|x - x^\star\|$$

and assuming that $D < +\infty$, we ensure (3.15). In some other cases, as for example, in the gradient method for the Euclidean norm, we can explicitly show that the distance to the solution is non-increasing and bounded.

Combining (3.13) with (3.14) we get the recursion:

$$F_k - F_{k+1} \quad \geq \quad \tfrac{1}{2LD^2} F_k^2. \tag{3.16}$$

Notice that in the imaginary case of *continuous* time, the corresponding dynamic $t \mapsto F_t$ can be analyzed in a differential form:

$$-\dot{F}_t \quad \geq \quad cF_t^2, \tag{3.17}$$

where $c > 0$ is a constant, $\dot{F}_t$ is the time derivative, which becomes a finite difference in the discrete-time case. Hence, we obtain

$$\frac{d}{dt}\left[\frac{1}{F_t}\right] \quad = \quad -\frac{\dot{F}_t}{F_t^2} \quad \overset{(3.17)}{\geq} \quad c,$$

and, after integrating, $F_t^{-1} \geq F_0^{-1} + ct$. Thus, $F_t = O(1/t)$, and we use these observations to analyze (3.16). We have, for every $k \geq 0$:

$$\frac{1}{F_{k+1}} - \frac{1}{F_k} \quad = \quad \frac{F_k - F_{k+1}}{F_{k+1} \cdot F_k} \quad \geq \quad \frac{1}{2LD^2} \cdot \frac{F_k^2}{F_k F_{k+1}} \quad \geq \quad \frac{1}{2LD^2}.$$

Telescoping, we get

$$\frac{1}{F_k} \quad \geq \quad \frac{1}{F_0} + \frac{k}{2LD^2} \geq \frac{k+4}{2LD^2},$$

where we used that $F_0 = f(x_0) - f^\star = f(x_0) - f(x^\star) - \langle \nabla f(x^\star), x_0 - x^\star \rangle \leq \frac{L}{2}\|x_0 - x^\star\|^2 \leq \frac{LD^2}{2}$. We have proved the following convergence result.

Theorem 3.2.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and smooth. For the iterates $\{x_k\}_{k \geq 0}$ generated by the gradient method, it holds:*

$$f(x_k) - f^\star \quad \leq \quad \frac{2LD^2}{k+4}, \qquad k \geq 0.$$

As a consequence, we see that the gradient method converges to the global solution, and to find a point x_k such that $f(x_k) - f^\star \leq \varepsilon$ it is enough to perform

$$k \quad = \quad \left\lfloor \frac{2LD^2}{\varepsilon} \right\rfloor + 1 \tag{3.18}$$

first-order oracle calls. In the following lectures we will discuss the optimality of this complexity bound and whether it can be improved.

On the choice of the norm. We have established the global complexity (3.18) for the gradient method on the class of convex smooth functions $f : \mathbb{R}^n \to \mathbb{R}$, which can be characterized by the following second-order condition:

$$0 \quad \leq \quad \langle \nabla^2 f(x) h, h \rangle \quad \leq \quad L\|h\|^2, \qquad \forall x, h \in \mathbb{R}^n, \tag{3.19}$$

where $\|\cdot\|$ is an arbitrary norm on $\mathbb{R}^n$.

A non-standard choice of norm might be useful for the design of optimization algorithms. For example, an important algorithm for training deep learning architectures, called *Adam* [23], can be viewed as a stabilized version of the stochastic gradient method under the $\|\cdot\|_\infty$-norm [2].

At the same time, the choice of the $\|\cdot\|_1$-norm for the primal space in the gradient method leads to *greedy coordinate descent*.

Therefore, a proper choice of norm can enable desirable features for these algorithms. Other examples include problems with explicitly given geometry (such as the space of probability distributions), when the choice of the corresponding norm is very natural.

However, the Euclidean norm remains the most important choice for the design and analysis of the optimization methods. Unless explicitly specified,

from now on in this chapter, we will focus on the Euclidean norm $\boxed{\|\cdot\| \equiv \|\cdot\|_2}$

The condition (3.19) under the Euclidean norm reads as:

$$0 \;\preceq\; \nabla^2 f(x) \;\preceq\; LI, \qquad x \in \mathbb{R}^n. \tag{3.20}$$

And the gradient method for the Euclidean norm has the following simplest form, starting from some $x_0 \in \mathbb{R}^n$, we iterate:

$$x_{k+1} \;=\; x_k - \tfrac{1}{L}\nabla f(x_k), \qquad k \geq 0, \tag{3.21}$$

where we assume the fixed step size $1/L$ for now. Assuming that a minimum $x^\star$ exists, let us look at the distance to the solution. We have

$$\begin{aligned} \tfrac{1}{2}\|x_{k+1} - x^\star\|^2 &= \tfrac{1}{2}\|x_k - x^\star - \tfrac{1}{L}\nabla f(x_k)\|^2 \\ &= \tfrac{1}{2}\|x_k - x^\star\|^2 + \tfrac{1}{L}\Big(\tfrac{1}{2L}\|\nabla f(x_k)\|^2 + 2\langle x^\star - x_k, \nabla f(x_k)\rangle\Big) \\ &\overset{(3.20)}{\leq} \tfrac{1}{2}\|x_k - x^\star\|^2 + \tfrac{1}{L}\Big(f(x^\star) - f(x_k)\Big) \;\leq\; \tfrac{1}{2}\|x_k - x^\star\|^2. \end{aligned} \tag{3.22}$$

Hence, in the Euclidean case, we have proved that the iterates of the gradient method remains bounded,

$$\|x_k - x^\star\| \;\leq\; \|x_0 - x^\star\|, \qquad \forall k \geq 0. \tag{3.23}$$

Note that $x^\star$ here can be an *arbitrary minimizer*.

Repeating the previous reasoning we obtain a similar convergence rate, but with the explicit distance $R := \|x_0 - x^\star\|$ from the initial point to any fixed solution.

Theorem 3.2.3. *Assume that a minimum $x^\star$ exist. On convex smooth functions* (3.20)*, the gradient method* (3.21) *has the following rate of convergence:*

$$f(x_k) - f^\star \;\leq\; \tfrac{2LR^2}{k+4}, \qquad k \geq 0. \tag{3.24}$$

Corollary 3.2.4. *In order to find a point x_k such that $f(x_k) - f^\star \leq \varepsilon$, it is enough to perform*

$$k \;=\; O\Big(\tfrac{LR^2}{\varepsilon}\Big) \tag{3.25}$$

first-order oracle calls.

3.2.3 Minimizing Gradient Norm

How good this result as compared to what we have seen before for the gradient method?

First of all, notice that the convergence rate (3.24) is in terms of the *last point*, which is the most natural candidate for the output of the algorithm.

In Section 2.3, we have already proved that to reach $\|\nabla f(\bar{x})\| \leq \varepsilon$, it is enough to perform

$$K \;=\; O\Big(\tfrac{L(f(x_0) - f^\star)}{\varepsilon^2}\Big) \tag{3.26}$$

iterations of the gradient method. The complexity $O(1/\varepsilon^2)$ seems much worse than that one in (3.25). At the same time, technically, complexity bounds (3.25) and (3.26) are not *directly*

comparable as they refer to different accuracy measures, and the role of the letter "ε" is different in them!

It appears that using convexity, we can improve the dependence on ε in (3.26) for the gradient norm minimization. Let us consider $2K$ iterations of the gradient method, where $K \geq 1$ is fixed. From the analysis of the gradient norm minimization (Theorem 2.3.4 in Section 2.3.3), we have:

$$\min_{K \leq i \leq 2K-1} \|\nabla f(x_i)\|^2 \;\leq\; \tfrac{2L(f(x_K)-f^\star)}{K} \overset{(3.24)}{\leq} \tfrac{4L^2R^2}{K(K+4)} \;=\; O\Big(\Big[\tfrac{LR}{K}\Big]^2\Big). \tag{3.27}$$

Therefore, we have obtained the following result.

Proposition 3.2.5. *In order to find a point $\bar{x}$ such that $\|\nabla f(\bar{x})\| \leq \varepsilon$, the gradient method* (3.21) *on smooth convex functions* (3.20) *needs*

$$K \;=\; O\Big(\tfrac{LR}{\varepsilon}\Big)$$

iterations (first-order oracle calls).

The convergence in terms of the gradient norm is stronger, as it leads to the convergence in terms of the function value. The gradient norm is also easier to use in practice as the stopping criterion for the algorithms.

Finally, we might ask what is the convergence rate in terms of the *distance to the solution* $\|x_k - x^\star\| \to 0$? Unfortunately, in general for convex functions, we cannot guarantee such convergence, even if $\|\nabla f(x_k)\| \to 0$ and $f(x_k) \to f^\star$. However, there is a specific class of functions that enables us to guarantee convergence in terms of the distance between points.

3.2.4 Strongly Convex Smooth Functions

Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable. We say that f is *strongly convex* and *smooth*, if the eigenvalues of the Hessian are both uniformly bounded from above and separated from zero:

$$\mu I \;\preceq\; \nabla^2 f(x) \;\preceq\; LI, \qquad x \in \mathbb{R}^n, \tag{3.28}$$

where $0 < \mu \leq L$ are parameters of our problem class. Of course, if $\mu = 0$ in (3.28) than we obtain the class of convex smooth functions, that we discussed before.

The problems that satisfy (3.28) are among the most important in optimization and possesses the fastest rates of convergence. As before, we can formulate the conditions (3.28) in terms of the gradients and in terms of the function values. Indeed, using the fundamental theorems of calculus,

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \;=\; \int_0^1 \langle \nabla^2 f(x + \tau(y-x))(y-x), y-x \rangle d\tau$$

and

$$f(y) - f(x) - \langle \nabla f(x), y - x \rangle \;=\; \int_0^1 (1-\tau) \langle \nabla^2 f(x + \tau(y-x))(y-x), y-x \rangle d\tau,$$

we get the following equivalent characterization of our new problem class.

Theorem 3.2.6. *The following conditions, that hold for any $x, y \in \mathbb{R}^n$, are equivalent to* (3.28)*:*

$$\mu \|y - x\|^2 \;\leq\; \langle \nabla f(y) - \nabla f(x), y - x \rangle \;\leq\; L \|y - x\|^2 \tag{3.29}$$

and

$$\tfrac{\mu}{2} \|y - x\|^2 \;\leq\; f(y) - f(x) - \langle \nabla f(x), y - x \rangle \;\leq\; \tfrac{L}{2} \|y - x\|^2. \tag{3.30}$$

Now we obtain the perfect symmetry in our inequalities. Geometrically, inequality (3.30) means that at each point $x \in \mathbb{R}^n$, we have both global upper and lower quadratic models of our function.

Note that if we ignore the upper inequality in (3.28), (3.29), and (3.30), we obtain the class of just *strongly convex functions* (not necessary smooth).

Exercise 3.2.1. Show that every strongly convex function f (with respect to the Euclidean norm), can be represented as $f(x) \equiv \varphi(x) + \frac{\mu}{2}\|x\|^2$, where $\varphi(\cdot)$ is a convex function.

Let us look at some consequences of (3.30). Plugging $x := x^\star$, we get

Proposition 3.2.7. *For a strongly convex smooth function $f : \mathbb{R}^n \to \mathbb{R}$, it holds:*

$$\frac{\mu}{2}\|y - x^\star\|^2 \;\leq\; f(y) - f^\star \;\leq\; \frac{L}{2}\|y - x^\star\|^2, \qquad y \in \mathbb{R}^n. \tag{3.31}$$

Therefore, the functional residual and the distance to the solution becomes comparable. If we have a convergence in terms of the functional residual, $f(x_k) - f^\star$, bound (3.31) also leads to a convergence in terms of the distance: $\|x_k - x^\star\| \to 0$ with the same rate.

Note the the lower inequality in (3.31) also proves that the solution $x^\star$ to a strongly convex optimization problem always *exist* and *unique*. This is not the case for convex functions, when $\mu = 0$.

Now, let us rearrange the terms in the lower inequality in (3.30). We obtain:

$$f(y) \;\geq\; f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{\mu}{2}\|y - x\|^2,$$

that holds for any $x, y \in \mathbb{R}^n$. Minimizing the left-hand and the right-hand sides independently, we obtain

$$f^\star \;\geq\; f(x) - \tfrac{1}{2\mu}\|\nabla f(x)\|^2.$$

Or, rearranging the terms, we obtain the following very important inequality.

Proposition 3.2.8. *For strongly convex functions, it holds*

$$\tfrac{1}{2\mu}\|\nabla f(x)\|^2 \;\geq\; f(x) - f^\star. \tag{3.32}$$

Let us apply the new inequality for the analysis of the gradient method. For one gradient step, we have

$$f(x_k) - f(x_{k+1}) \;\geq\; \tfrac{1}{2L}\|\nabla f(x_k)\|^2 \overset{(3.32)}{\geq} \tfrac{\mu}{L}(f(x_k) - f^\star), \tag{3.33}$$

or, for the functional residual $F_k := f(x_k) - f^\star$, we have

$$F_{k+1} \overset{(3.33)}{\leq} \Bigl(1 - \tfrac{\mu}{L}\Bigr)F_k \;\leq\; \exp\Bigl(-\tfrac{\mu}{L}\Bigr)F_k. \tag{3.34}$$

The ratio $\frac{L}{\mu} \geq 1$ is very important and called the *condition number* of the function. Applying inequality (3.34) for k steps of the method, we prove the following result.

Theorem 3.2.9. *For the iterations of the gradient method on strongly convex smooth functions, we have the <u>linear rate</u> of convergence:*

$$f(x_k) - f^\star \;\leq\; \exp(-k\tfrac{\mu}{L})(f(x_0) - f^\star). \tag{3.35}$$

Therefore, to reach $f(x_K) - f^\star \leq \varepsilon$ it is enough to perform

$$K \;=\; \tfrac{L}{\mu}\ln\tfrac{f(x_0)-f^\star}{\varepsilon} \overset{(3.30)}{\leq} \tfrac{L}{\mu}\ln\tfrac{LR^2}{2\varepsilon} \tag{3.36}$$

iterations of the algorithm.

3.3 Polyak's Heavy Ball Method

3.3.1 Quadratic Functions

Let us consider the problem of unconstrained minimization of the quadratic function,

$$\min_{x \in \mathbb{R}^n} \Big\{ f(x) \;=\; \tfrac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle \Big\}, \tag{3.37}$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ is given data. We assume that $A = A^\top \succeq 0$ (why?). Actually, without loss of generality, we can always assume that $A \succ 0$ (strictly). However, the smallest eigenvalue can be tiny. In this case the problem is called *ill-conditioned*. Unfortunately, ill-conditioned problems are the most frequent in practice.

Computing the gradient, we get

$$\nabla f(x) \;=\; Ax - b, \tag{3.38}$$

and at the solution $x^\star$ should solve the linear system:

$$\nabla f(x^\star) \;=\; Ax^\star - b \;=\; 0 \qquad \Leftrightarrow \qquad b \;=\; Ax^\star. \tag{3.39}$$

Therefore, any optimization method for minimizing (3.37) automatically provides us with an algorithm for solving linear systems with symmetric matrices (3.39). In fact, this approach remains the most efficient way to solve large-scale systems, when the dimension n is huge. According to (3.38), to compute the gradient vector at a given point, it requires to perform one *matrix-vector* product. If the matrix A is *sparse*, it can be done efficiently even for a very large dimension n.

From (3.39), we have another representation of the gradient, using the optimal solution:

$$\nabla f(x) \;=\; A(x - x^\star). \tag{3.40}$$

Computing the Hessian, we observe that it is constant,

$$\nabla^2 f(x) \;\equiv\; A.$$

Therefore, quadratic function (3.37) is strongly convex and smooth with

$$0 \;<\; \mu \;=\; \lambda_{\min}(A) \;\le\; \lambda_{\max}(A) \;=\; L.$$

It is easy to check that the Taylor expansions hold exactly for quadratic functions.

Proposition 3.3.1. *For quadratic functions, it holds:*

$$\begin{aligned} f(y) - f(x) - \langle \nabla f(x), y - x\rangle \;&\equiv\; \tfrac{1}{2}\langle \nabla f(y) - \nabla f(x), y - x\rangle \\ &\equiv\; \tfrac{1}{2}\langle \nabla^2 f(x)(y - x), y - x\rangle \\ &\equiv\; \tfrac{1}{2}\langle A(y - x), y - x\rangle \;\equiv\; \tfrac{1}{2}\|y - x\|_A^2, \end{aligned} \tag{3.41}$$

where $\|y - x\|_A$ stands for the generalized Euclidean norm with matrix $A = A^\top \succ 0$.

Exercise 3.3.1. Check (3.41).

From (3.41), we obtain the following interesting formula,

$$f(y) - f(x) \;=\; \tfrac{1}{2}\langle \nabla f(y) + \nabla f(x), y - x\rangle, \qquad x, y \in \mathbb{R}^n. \tag{3.42}$$

3.3.2 Heavy Ball Method

We discuss a faster method for minimizing a quadratic function, developed by B.T. Polyak [37]. Instead of performing a simple gradient step, we add some *inertia* to the method. Each iteration reads as follows:

$$x_{k+1} \;=\; x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}), \qquad k \geq 1. \tag{3.43}$$

where $\alpha > 0$ is a step-size and $0 \leq \beta \leq 1$ is an extra parameter. This algorithm is called the heavy ball method. The dynamical system (3.43) corresponds to a discretized version of the motion of a body ("the heavy ball") in a potential field $\nabla f(\cdot)$ under the force of *friction*, where β is a parameter of this force ($\beta = 0$ correspond to no friction).

Another common interpretation of this algorithm is the gradient method with *momentum*. Indeed, iterations (3.43) can be rewritten as follows, starting from some initialization $x_0 \in \mathbb{R}^n$ and $s_0 = 0$, we update, for $k \geq 0$:

$$\begin{aligned} s_{k+1} &= \beta s_k + \nabla f(x_k), \\ x_{k+1} &= x_k - \tfrac{1}{L} s_{k+1}, \end{aligned} \tag{3.44}$$

and $0 \leq \beta \leq 1$ has an interpretation of *momentum parameter* (how fast we forget the history), and we use the constant step-size $1/L$ in front of s_{k+1}.

Exercise 3.3.2. Check that iterations (3.44) and (3.43) are equivalent.

This is a very popular technique in machine learning, where it helps the method to behave more stable, especially when the gradients are stochastic and noisy, and the objective landscape is non-convex.

Let us analyze algorithm (3.44) for the quadratic case. We consider the simplest choice of momentum parameter, $\boxed{\beta := 1}$. Hence, we have

$$s_{k+1} \;=\; \sum_{i=0}^{k} \nabla f(x_i). \tag{3.45}$$

The consequence of the fact that the gradient mapping $\nabla f(\cdot)$ is affine is that

$$\tfrac{1}{k} s_k \;=\; \tfrac{1}{k} \sum_{i=0}^{k-1} \nabla f(x_i) \;=\; \nabla f(\bar{x}_k), \qquad \text{where} \qquad \bar{x}_k \;:=\; \tfrac{1}{k} \sum_{i=0}^{k-1} x_i. \tag{3.46}$$

Let us substitute direction from (3.44) into the global quadratic upper bound:

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \tfrac{L}{2} \|x_{k+1} - x_k\|^2 \\ &\overset{(3.44)}{=} f(x_k) - \tfrac{1}{L} \langle \nabla f(x_k), s_{k+1} \rangle + \tfrac{1}{2L} \|s_{k+1}\|^2 \\ &= f(x_k) - \tfrac{1}{L} \langle s_{k+1} - s_k, s_{k+1} \rangle + \tfrac{1}{2L} \|s_{k+1}\|^2 \\ &= f(x_k) - \tfrac{1}{2L} \|s_{k+1}\|^2 + \tfrac{1}{L} \langle s_k, s_{k+1} \rangle. \end{aligned}$$

Therefore, we get the following "progress" of each step.

Proposition 3.3.2. *For every $k \geq 0$:*

$$f(x_k) - f(x_{k+1}) \;\geq\; \tfrac{1}{2L} \|s_{k+1}\|^2 - \tfrac{1}{L} \langle s_k, s_{k+1} \rangle. \tag{3.47}$$

The last term has an interpretation of the "correlation" between partial sums of the gradients.

We substitute two consequitive points into equation (3.42), which is valid only for quadratic functions:

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\overset{(3.42)}{=} \tfrac{1}{2}\langle \nabla f(x_k) + \nabla f(x_{k+1}), x_k - x_{k+1}\rangle \\ &= \tfrac{1}{2L}\langle s_{k+2} - s_k, s_{k+1}\rangle = \tfrac{1}{2L}\big(\langle s_{k+2}, s_{k+1}\rangle - \langle s_k, s_{k+1}\rangle\big). \end{aligned}$$

Rearranging the terms, we express next correlation using the previous one and the function difference,

$$f(x_k) - f(x_{k+1}) + \tfrac{1}{2L}\langle s_k, s_{k+1}\rangle = \tfrac{1}{2L}\langle s_{k+1}, s_{k+2}\rangle. \tag{3.48}$$

Telescoping this inequality, and using that $s_0 = 0$, we get the following bound on the correlations.

Proposition 3.3.3. *For every $k \geq 0$:*

$$f(x_0) - f^\star \geq f(x_0) - f(x_k) \overset{(3.48)}{=} \tfrac{1}{2L}\langle s_k, s_{k+1}\rangle. \tag{3.49}$$

Now, we can substitute bound (3.49) into (3.47). We obtain:

$$f(x_k) - f(x_{k+1}) \geq \tfrac{1}{2L}\|s_{k+1}\|^2 - 2(f(x_0) - f^\star).$$

Telescoping this inequality, we get

$$f(x_0) - f^\star \geq f(x_0) - f(x_k) \geq \tfrac{1}{2L}\sum_{i=1}^{k}\|s_i\|^2 - 2k(f(x_0) - f^\star).$$

Theorem 3.3.4. *For the iterations of the heavy ball method* (3.44)*, with $\beta = 1$, it holds:*

$$2L(1+2k)(f(x_0) - f^\star) \geq \sum_{i=1}^{k}\|s_i\|^2 \overset{(3.46)}{=} \sum_{i=1}^{k} i^2 \Big\|\nabla f(\bar{x}_i)\Big\|^2. \tag{3.50}$$

Therefore, denoting the smallest gradient among averaged points, $g_k := \min\Big\{\|\nabla f(\bar{x}_1)\|, \ldots, \|\nabla f(\bar{x}_k)\|\Big\}$, we have

$$g_k^2 \leq \frac{2L(1+2k)(f(x_0)-f^\star)}{\sum_{i=1}^{k} i^2} = \frac{12L(f(x_0)-f^\star)}{k(k+1)} = O\Big(\Big[\frac{L(f(x_0)-f^\star)}{k^2}\Big]\Big). \tag{3.51}$$

Exercise 3.3.3. Compare the convergence rate (3.51) with that of the gradient method in (3.27).

3.3.3 Optimal Complexity of the Heavy Ball Method

We write down the heavy ball method in the momentum form (3.44) as the following algorithm. For simplicity, 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 3.1: *Gradient Method with Momentum.*

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 3.1. In Section 3.2, we have established few rates of convergence for the gradient method. We proved the rate in terms of the functional residual, for $k \geq 0$:

$$f(x_k) - f^\star \;\leq\; \tfrac{2LR^2}{k+4}, \qquad R \;:=\; \|x_0 - x^\star\|. \tag{3.52}$$

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 $k \geq 1$:

$$\min_{0 \leq i \leq k-1} \|\nabla f(x_i)\| \;\leq\; \tfrac{4LR}{k}. \tag{3.53}$$

In this case, the result point $\bar{x}$ is the one among candidates $\{x_0, \ldots, x_{k-1}\}$ 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,

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

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

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

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*.

Complexity of the heavy ball with restarts. In the previous section, we have analyzed the case $\boxed{\beta := 1}$ of Algorithm 3.1. on *quadratic functions*. This is much more restricted class of functions than all convex smooth functions.

We have proved the following result (see Theorem 3.3.4):

- 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* $g_k := \min\{\|\nabla f(\bar{x}_1)\|, \ldots, \|\nabla f(\bar{x}_k)\|\}$ we have the rate
$$g_k^2 \;\leq\; \tfrac{12L(f(x_0)-f^\star)}{k(k+1)} \tag{3.54}$$
for Algorithm 3.1 with $\beta := 1$ on convex quadratic functions.

Note that (3.54) gives us $g_k = O(1/k)$, which is similar to the rate of the gradient method (3.53). However, the presence of $f(x_0) - f^\star$ in the right hand side of (3.54) 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 the strong convexity, we have the inequality (see Proposition 3.2.8):

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

Therefore, denoting by $\bar{x}_K^\star$ the point with the smallest gradient norm among $\{\bar{x}_1, \ldots, \bar{x}_K\}$, we obtain from (3.54):

$$f(\bar{x}_K^\star) - f^\star \quad \leq \quad \tfrac{6L}{\mu K^2} \cdot (f(x_0) - f^\star). \tag{3.55}$$

Let us choose

$$K \quad := \quad \sqrt{\tfrac{12L}{\mu}}. \tag{3.56}$$

Substituting this value into (3.55), we get

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

In other words, performing (3.56) 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 3.3.5. *The total complexity of the heavy ball method with restarts is*

$$\sqrt{\tfrac{12L}{\mu}} \log_2 \tfrac{f(x_0)-f^\star}{\varepsilon} \tag{3.57}$$

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 (3.57) 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 (see below), 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,

$$x_{k+1} \quad = \quad x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}), \tag{3.58}$$

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 (3.58), and subtracting $x^\star$ from both sides, we obtain the recursion:

$$r_{k+1} \quad = \quad r_k - \alpha A r_k + \beta(r_k - r_{k-1}), \tag{3.59}$$

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 (3.59) can be written in the matrix form:

$$\begin{pmatrix} r_{k+1} \\ r_k \end{pmatrix} \quad = \quad C \begin{pmatrix} r_k \\ r_{k-1} \end{pmatrix} \tag{3.60}$$

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 (3.60) converges to 0 with the desirable linear rate (see also [37]).

Note that complexity (3.57) 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 [27])

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 [15], 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.

Further in the course, we will study another accelerated algorithm, called *Nesterov's fast gradient method* [31], that achieves this goal.

3.4 Lower Bound 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

$$f(x_k) - f^\star \;\leq\; O\Big(\tfrac{LR^2}{k}\Big), \qquad k \geq 1, \tag{3.61}$$

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

Theorem 3.4.1. *Let $L > 0$, $R > 0$ and $K \geq 1$ be fixed. For any first-order optimization algorithm running for K iterations, there is a convex function $f : \mathbb{R}^n \to \mathbb{R}$ with $n \geq 2K + 1$ such that ∇f is Lipschitz continuous with constant L and $\|x_0 - x^\star\| = R$, where $x^\star$ is a minimizer of f, so that*

$$f(x_K) - f^\star \;\geq\; \tfrac{3LR^2}{2^6(K+1)^2}. \tag{3.62}$$

We see that (3.62) does not match (3.61) 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 (3.62) 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* ($n \geq 2K + 1$), which is the case for modern applications;
- Even if the dimension n is small, it tells us something about behavior of the algorithm in the early stage.

3.4.1 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 $L := \text{const}$ (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:

$$x_{k+1} \;\in\; \mathcal{L}_{k+1} \;:=\; \text{span}\Big\{\nabla f(x_0), \ldots, \nabla f(x_k)\Big\}. \tag{3.63}$$

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

$$\mathcal{L}_0 \;\subseteq\; \mathcal{L}_1 \;\subseteq\; \mathcal{L}_2 \;\subseteq\; \ldots, \tag{3.64}$$

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

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

In fact, even if condition (3.63) is not satisfied, it is possible to modify the construction of the lower bound, using a *resisting oracle*, to ensure that (3.63) holds for any first-order method, and the construction of the lower bound would work.

But for now, under assumption (3.63), we can consider one objective function *for any algorithm*, which is the following quadratic function [31], parameterized by an integer $k \geq 0$:

$$\begin{aligned}
f_k(x) \;&:=\; \tfrac{1}{2}\Big[\sum_{i=1}^{k-1}(x^{(i)} - x^{(i+1)})^2 + \sum_{i=k}^{n}(x^{(i)})^2\Big] - \langle b, x\rangle \\
&= \tfrac{1}{2}\Big[(x^{(1)} - x^{(2)})^2 + \ldots + (x^{(k-1)} - x^{(k)})^2 + (x^{(k)})^2 + \ldots + (x^{(n)})^2\Big] - \langle b, x\rangle \\
&= \tfrac{1}{2}\|C_k x\|_2^2 - \langle b, x\rangle \;=\; \tfrac{1}{2}\langle C_k^\top C_k x, x\rangle - \langle b, x\rangle
\end{aligned} \tag{3.65}$$

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

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

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 Λ_k is the following tridiagonal matrix:

$$\Lambda_k \;=\; \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 \\ \dots & & & & & \\ 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{pmatrix}. \tag{3.66}$$

What is so specific about this function?

- 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. 3.1).

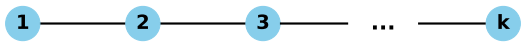

Figure 3.1: Chain graph with k nodes.

From (3.65) it is easy to see that $A_k \preceq 4I$. Indeed, for any $h \in \mathbb{R}^n$, we have

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

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 $x_k^\star = \operatorname{argmin}_{x \in \mathbb{R}^n} f_k(x)$. We differentiate (3.65) and see that a solution should satisfy the following linear system of equations:

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

We obtain that the following vector satisfies this equations:

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

The optimum of the quadratic function f_k is given by

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

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

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

This function has a very simple structure of the Krylov subspaces (3.64). Namely, we can control the number of non-zeros in iterations of the algorithm: after $k \ge 0$ iterations, the number of nonzeros in x_k is no more than k.

Proposition 3.4.2. *Assume that iterates $\{x_i\}_{i \ge 0}$ satisfy our assumption* (3.63)*. Then,*

$$\mathcal{L}_k \subseteq \mathbb{R}^{n,k} := \{x \in \mathbb{R}^n \ : \ x^{(i)} = 0 \ \textit{for} \ i > k\}. \tag{3.67}$$

Proof. Let us prove (3.67) by induction.

- By our assumption $x_0 := 0$ and $\mathcal{L}_0 := \{0\} = \mathbb{R}^{n,0}$.
- 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 $p \ge 0$. This follows directly from the structure (3.65) of the objective.

Proposition 3.4.3. *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 $\mathbb{R}^{n,k}$.

We run a method for a fixed number of k iterations, on a function $\boxed{f(x) := f_{2k+1}(x)}$, starting from $0 \in \mathbb{R}^n$. Tridiagonal structure of the matrix ensures that $x_k \in \mathbb{R}^{n,k}$ (the space where only first k components are non-zero):

$$\begin{array}{rcl} x_0 & = & [0 \;\; 0 \;\; 0 \;\; \cdots \;\; 0] \\ x_1 & = & [\star \;\; 0 \;\; 0 \;\; \cdots \;\; 0] \\ x_2 & = & [\star \;\; \star \;\; 0 \;\; \cdots \;\; 0] \\ x_3 & = & [\star \;\; \star \;\; \star \;\; \cdots \;\; 0] \\ & \cdots & \end{array}$$

And since f_k and f_{2k+1} are informationally indistinguishable for the method for the first k iterations, we have the lower bound on the best possible function value.

Corollary 3.4.4. *For the output of the algorithm, $x_k \in \mathcal{L}_k \subseteq \mathbb{R}^{n,k}$, we have*

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

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

$$f(x_k) - f^\star \;\; \geq \;\; \tfrac{2k+1}{2} - \tfrac{k}{2} \;\; = \;\; \tfrac{k+1}{2}. \tag{3.68}$$

and

$$R \;\; = \;\; \|x_0 - x^\star\|^2 \;\; = \;\; \|x_{2k+1}^\star\|^2 \;\; \leq \;\; \tfrac{2^3(k+1)^3}{3}. \tag{3.69}$$

Therefore,

$$\tfrac{f(x_k)-f^\star}{\|x_0-x^\star\|^2} \overset{(3.68)}{\geq} \tfrac{k+1}{2R^2} \overset{(3.69)}{\geq} \tfrac{3(k+1)}{2^4(k+1)^3} \;\; = \;\; \tfrac{3}{2^4(k+1)^2}.$$

and this is the lower bound!

Now, for an arbitrary Lipschitz constant $L > 0$, we can take a multiplied function:

$$\varphi(x) \;\; := \;\; \tfrac{L}{4} f(x),$$

for which we will have:

$$\tfrac{\varphi(x_k)-\varphi^\star}{R^2} \;\; \geq \;\; \tfrac{3L}{2^6(k+1)^2}.$$

This completes the proof of the following theorem.

Theorem 3.4.5. *Let $L > 0$ and $K \geq 1$ be fixed. Then, for any first-order optimization algorithm, such that*

$$x_{k+1} \;\; \in \;\; span\Big\{\nabla f(x_0), \ldots, \nabla f(x_k)\Big\},$$

there is a convex function $f : \mathbb{R}^n \to \mathbb{R}$ with $n \geq 2K + 1$ with L-Lipschitz gradient, such that

$$f(x_K) - f^\star \geq \;\; \tfrac{3L\|x_0-x^\star\|^2}{2^6(K+1)^2}. \tag{3.70}$$

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

$$K + 1 \;\; \geq \;\; \tfrac{1}{8}\sqrt{\tfrac{3L\|x_0-x^\star\|^2}{\varepsilon}}. \tag{3.71}$$

Remark 3.4.6. Not that in Theorem 3.4.5 we do not specify the value of $\|x_0 - x^\star\|$ in the right hand side of (3.70): we just say that there exists a function with *some* $\|x_0 - x^\star\|$ such that (3.70) is true. However, it is actually possible to show that for any desirable $R > 0$ there is a function such that $\|x_0 - x^\star\| = R$ for its minimum $x^\star$ and that (3.70) holds — see Exercise 3.8.1.

General case. What if an algorithm does not satisfy our assumption, $x_k \in \mathcal{L}_k$? It is possible to generalize this 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 [27]).

3.4.2 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:

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

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:

$$f_\mu(\bar{x}) - f_\mu^\star \;\; \leq \;\; \delta, \tag{3.72}$$

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,

$$f_\mu(\bar{x}) \;\; \overset{(3.72)}{\leq} \;\; f_\mu^\star + \delta \;\; \leq \;\; f_\mu(y) + \delta \;\; = \;\; f(y) + \tfrac{\mu}{2}\|y\|^2 + \delta,$$

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

$$f(x^\star) \;\; \geq \;\; f_\mu(\bar{x}) - \tfrac{\mu}{2}\|x^\star\|^2 - \delta. \tag{3.73}$$

Hence,

$$f(\bar{x}) - f^\star \;\; = \;\; f(\bar{x}) - f(x^\star) \;\; \leq \;\; f_\mu(\bar{x}) - f(x^\star) \;\; \overset{(3.73)}{\leq} \;\; \tfrac{\mu}{2}\|x^\star\|^2 + \delta.$$

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

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

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

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

with $c = \frac{\sqrt{3}}{8}$. This would mean that the complexity of solving a smooth convex function is

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

which contradicts (3.71). Hence, we obtain the following.

Proposition 3.4.7. *The complexity of any first-order method minimizing smooth convex functions is lower bounded as*

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

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 (3.57).

3.5 Nesterov's Fast Gradient Method

Before, we analyzed methods built on some "physical" or "geometrical" intuition. Such methods are easy to describe by analogy to some known phenomena. However it is difficult to analyze them, after the method is already rigidly fixed. Now, we try a different approach. We will immediately start to look into the essence of what we want to achieve, and that would lead us to the development of a method. One of the crucial components of the analysis is the strong convexity of the regularizer that we use in the method, a concept that we review first.

3.5.1 Review: Strongly Convex Functions

We say that a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is *strongly convex* with a constant $\mu > 0$, if

$$f(y) \quad \geq \quad f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{\mu}{2}\|y - x\|^2, \qquad \forall x, y \in \mathbb{R}^n. \tag{3.74}$$

if we have two functions f_1, and f_2 that satisfy (3.74) with constants $\mu_1, \mu_2 \geq 0$, then their sum $f(\cdot) = f_1(\cdot) + f_2(\cdot)$ satisfy (3.74) with constant $\mu = \mu_1 + \mu_2$. Therefore, the sum of a convex function with a strongly convex one will always give us a strongly convex function.

The most important example of strongly convex function is the squared Euclidean norm:

$$d(x) \quad = \quad \tfrac{1}{2}\|x\|^2 \quad = \quad \tfrac{1}{2}\langle x, x \rangle.$$

Let us check (3.74) directly. We have

$$\begin{aligned} d(y) - d(x) - \langle \nabla d(x), y - x \rangle &= \tfrac{1}{2}\|y\|^2 - \tfrac{1}{2}\|x\|^2 - \langle x, y - x \rangle \\ &= \tfrac{1}{2}\|y - x + x\|^2 - \tfrac{1}{2}\|x\|^2 - \langle x, y - x \rangle \\ &= \tfrac{1}{2}\|y - x\|^2 + \tfrac{1}{2}\|x\|^2 + \langle x, y - x \rangle - \tfrac{1}{2}\|x\|^2 - \langle x, y - x \rangle \\ &= \tfrac{1}{2}\|y - x\|^2. \end{aligned}$$

Therefore, for the squared Euclidean norm, inequality (3.74) is satisfied as equation, with $\mu = 1$.

As a direct consequence of these observations, let us consider a regularized objective

$$g(y) \quad := \quad f(y) + \tfrac{1}{2}\|y - x_0\|^2,$$

where f is an arbitrary differentiable convex function. Hence, g is strongly convex with constant $\mu = 1$, and by (3.74), for the optimum $x_g^\star := \operatorname{argmin}_{y \in \mathbb{R}^n} g(y)$ we have

$$g(y) \quad \geq \quad g(x_g^\star) + \langle \nabla g(x_\star), y - x_\star \rangle + \tfrac{\mu}{2}\|y - x^\star\|^2 \quad = \quad g^\star + \tfrac{\mu}{2}\|y - x_g^\star\|^2. \tag{3.75}$$

Therefore, by strong convexity, we obtain a strengthening (3.75) of a trivial inequality: $g(y) \geq g^\star$ that is the definition of $g^\star$.

3.5.2 Analysis of Similar Triangles

We are interested in solving an unconstrained optimization problem,

$$\min_{x \in \mathbb{R}^n} f(x), \tag{3.76}$$

where f is convex and it has a Lipschitz gradient. These are the main two inequalities that characterize our problem class and that we will employ:

$$0 \;\leq\; f(y) - f(x) - \langle \nabla f(x), y - x \rangle \;\leq\; \tfrac{L}{2}\|y - x\|^2, \qquad x, y \in \mathbb{R}^n.$$

We fix a sequence $A_k > 0$ of growing coefficients that will give us the "rate" of the method. The idea is to prove the following inequality:

$$f(x_k) - f^\star \;\leq\; \tfrac{\|x_0 - x^\star\|^2}{2A_k}, \tag{3.77}$$

where A_k specifies exactly the *rate of convergence*. To achieve the optimal rate matching (3.70), we hope to have $A_k \approx \frac{k^2}{L}$.

Instead of (3.77), let us try to ensure a bit more general goal, the same inequality but for an arbitrary $x \in \mathbb{R}^n$:

$$f(x_k) - f(x) \;\leq\; \tfrac{\|x_0 - x\|^2}{2A_k},$$

which is equivalent to

$$\varphi_k(x) \;:=\; \tfrac{1}{2}\|x_0 - x\|^2 + A_k f(x) \;\geq\; A_k f(x_k), \qquad x \in \mathbb{R}^n. \tag{3.78}$$

Notice that in the left hand side of (3.78) we have a value of a strongly convex function at arbitrary point x (as a sum of a convex function $A_k f(\cdot)$ and strongly convex $\frac{1}{2}\|x_0 - \cdot\|^2$), and in the right hand side we have a constant term. Thus, from (3.78) we want to achieve that

$$\varphi_k^\star \;=\; \min_{x \in \mathbb{R}^n} \varphi_k(x) \;\geq\; A_k f(x_k). \tag{3.79}$$

However, by strong convexity, we know an improved inequality:

$$\varphi_k(x) \;\geq\; \varphi_k^\star + \tfrac{1}{2}\|x_{\varphi_k}^\star - x\|^2 \;\overset{(3.79)}{\geq}\; A_k f(x_k) + \tfrac{1}{2}\|x_{\varphi_k}^\star - x\|^2.$$

Therefore, we refine our goal. Now we want to construct a sequence of growing coefficients A_k (growing as fast as possible), and two sequences of points $\{x_k\}_{k \geq 0}$ and $\{v_k\}_{k \geq 0}$ such that, for any $k \geq 0$:

$$\tfrac{1}{2}\|x_0 - x\|^2 + A_k f(x) \;\geq\; \tfrac{1}{2}\|v_k - x\|^2 + A_k f(x_k), \qquad x \in \mathbb{R}^n. \tag{3.80}$$

Clearly, if (3.80) is satisfied, than we achieve our initial goal (3.77), by plugging $x := x^\star$. Note that in (3.80), the point v_k is not necessarily equal to $x_{\varphi_k}^\star$, but rather a substitute for it.

First, let us check how to start. We can assume that $A_0 = 0$, and $x_0 = v_0$. Then (3.80) is trivially satisfied.

Now, assume that (3.80) is satisfied for some $k \geq 0$. We want to "increase the rate", by setting $A_{k+1} = A_k + a_{k+1}$, where $a_{k+1} > 0$ is some positive coefficient. Thus, we have

$$\begin{aligned} \tfrac{1}{2}\|x_0 - x\|^2 + A_{k+1} f(x) \;&=\; \tfrac{1}{2}\|x_0 - x\|^2 + a_{k+1} f(x) + A_k f(x) \\ &\overset{(3.80)}{\geq}\; \tfrac{1}{2}\|v_k - x\|^2 + a_{k+1} f(x) + A_k f(x_k). \end{aligned}$$

Now, when we have a sum of two function values, it is natural to use *convexity*. Denote $\gamma_k = \frac{a_{k+1}}{A_{k+1}} = \frac{a_{k+1}}{A_k + a_{k+1}}$. We have:

$$a_k f(x) + A_k f(x_k) \;=\; A_{k+1}\Big(\gamma_k f(x) + (1 - \gamma_k) f(x_k)\Big) \;\geq\; A_{k+1} f(y),$$

where $y := \gamma_k x + (1-\gamma_k)x_k$.

We can continue the bound by using the global linear model, by convexity again:

$$f(y) \quad \geq \quad f(y_k) + \langle \nabla f(y_k), y - y_k \rangle,$$

where y_k is some point that we have to choose. We have a flexibility in the choice of y_k, but one natural candidate is

$$y_k \quad = \quad \gamma_k v_k + (1-\gamma_k)x_k,$$

as we would have then $y - y_k = \gamma_k(x - v_k)$.

We obtained,

$$\begin{aligned} \tfrac{1}{2}\|x_0 - x\|^2 + A_{k+1}f(x) \quad &\geq \quad \tfrac{1}{2}\|v_k - x\|^2 + A_{k+1}\Big[f(y_k) + \langle \nabla f(y_k), y - y_k\rangle\Big] \\ &= \quad \tfrac{1}{2}\|v_k - x\|^2 + A_{k+1}\Big[f(y_k) + \gamma_k\langle \nabla f(y_k), x - v_k\rangle\Big] \\ &\equiv \quad m_k(x). \end{aligned}$$

We minimize the right hand side in x to obtain the next auxiliary point, $v_{k+1} = \operatorname{argmin}_x m_k(x)$. This leads us to the following update:

$$v_{k+1} \quad = \quad v_k - a_{k+1}\nabla f(y_k).$$

Hence, by strong convexity of $m_k(\cdot)$, we get

$$\tfrac{1}{2}\|x_0 - x\|^2 + A_{k+1}f(x) \quad \geq \quad \tfrac{1}{2}\|v_{k+1} - x\|^2 + m_k^\star,$$

where

$$m_k^\star \quad = \quad m_k(v_{k+1}) \quad = \quad \tfrac{1}{2}\|v_k - v_{k+1}\|^2 + A_{k+1}\Big[f(y_k) + \gamma_k\langle \nabla f(y_k), v_{k+1} - v_k\rangle\Big].$$

To finish the proof, we need to make it possible that $m_k^\star \geq A_{k+1}f(x_{k+1})$. How we can do that?

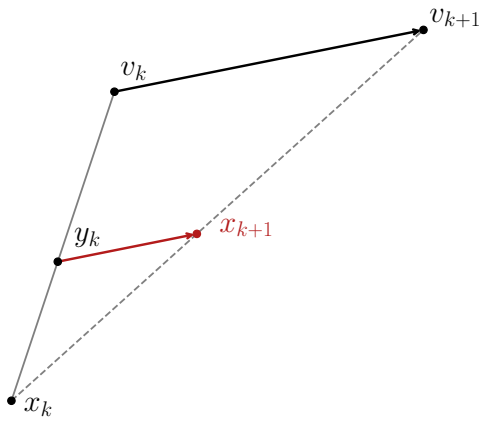

Figure 3.2: Update of similar triangles in the fast gradient method.

We choose $x_{k+1} := \gamma_k v_{k+1} + (1-\gamma_k)x_k$, thus $x_{k+1} - y_k$ is parallel to $v_{k+1} - v_k$ (see Fig. 3.2). Therefore,

$$\begin{aligned} m_k^\star &= \tfrac{1}{2\gamma_k^2}\|x_{k+1} - y_k\|^2 + A_{k+1}\Big[f(y_k) + \langle \nabla f(y_k), x_{k+1} - y_k\rangle\Big] \\ &= A_{k+1}\Big[\tfrac{1}{2\gamma_k^2 A_{k+1}}\|x_{k+1} - y_k\|^2 + \langle \nabla f(y_k), x_{k+1} - y_k\rangle + f(y_k)\Big] \\ &\geq A_{k+1} f(x_{k+1}), \end{aligned}$$

as soon as $\frac{1}{\gamma_k^2 A_{k+1}} = \frac{A_{k+1}^2}{a_{k+1}^2 A_{k+1}} = \frac{a_{k+1}+A_k}{a_{k+1}^2} \geq L$. Thus, we have established (3.80) for all $k \geq 0$.

3.5.3 Fast Gradient Method

We come to the following algorithmic scheme of the optimization method.

Algorithm 3.2: *Fast Gradient Method.*

Initialization: $x_0 \in \mathbb{R}^n$. Set $v_0 = x_0$ and $A_0 = 0$. Fix $K \geq 1$.

For $k = 0 \dots K-1$ **iterate:**

1. Choose a new coefficient $a_{k+1} > 0$. Set $A_{k+1} := A_k + a_{k+1}$ and $\gamma_k := \frac{a_{k+1}}{A_{k+1}}$
2. Compute the gradient $\nabla f(y_k)$ at the intermediate point $y_k := \gamma_k v_k + (1-\gamma_k)x_k$
3. Update $v_{k+1} = v_k - a_{k+1}\nabla f(y_k)$
4. Set a new point from the triangle rule: $x_{k+1} := \gamma_k v_{k+1} + (1-\gamma_k)x_k$

Return x_K

In this method, for simplicity we fix the number of iterations K and use it as a stopping condition for the algorithm. A more advanced stopping condition would include a computation of an *accuracy certificate* for a solution, that would guarantee a small function residual for the output. We study how to compute accuracy certificates later in the course.

Note that in this algorithm, the only unspecified parameter is a sequence $\{a_k\}_{k\geq 1}$ of positive coefficients, that we have to choose. Then, the growth of

$$A_k = \sum_{i=1}^k a_i$$

defines the rate of convergence.

With our analyses, we have established the following result.

Theorem 3.5.1. *Let $a_{k+1} > 0$ be chosen such that $\frac{a_{k+1}+A_k}{a_{k+1}^2} \geq L$. Then, we have*

$$\tfrac{1}{2}\|v_k - x\|^2 + A_k(f(x_k) - f(x)) \leq \tfrac{1}{2}\|x_0 - x\|^2, \qquad x \in \mathbb{R}^n. \tag{3.81}$$

Substituting $x := x^\star$, we obtain

$$f(x_k) - f^\star \leq \tfrac{\|x_0 - x^\star\|^2}{2A_k}, \qquad k \geq 1. \tag{3.82}$$

3.5.4 The Parameter Choice

We need to establish the following inequality:

$$\frac{A_{k+1}}{a_{k+1}^2} \quad \geq \quad L. \tag{3.83}$$

Note that as larger a_{k+1} than the faster the rate of convergence of the method.

Solving quadratic equation. Let us choose $a_{k+1} > 0$ such that inequality (3.83) is satisfied as equation. That is:

$$A_{k+1} \quad = \quad a_{k+1} + A_k \quad = \quad La_{k+1}^2. \tag{3.84}$$

This is quadratic equation in a_{k+1}, which has an explicit formula for a (positive) solution:

$$\boxed{a_{k+1} \quad := \quad \frac{1}{2L} \cdot \left(1 + \sqrt{1 + 4A_k L}\right).} \tag{3.85}$$

Note that in the basic gradient descent we choose $a_{k+1} \approx \frac{1}{L}$. Therefore, formula (3.85) is more aggressive, allowing for larger exploratory steps of the accelerated method.

We need to figure out the rate of convergence. We have

$$\begin{aligned} \sqrt{A_{k+1}} - \sqrt{A_k} \quad &= \quad \frac{A_{k+1} - A_k}{\sqrt{A_{k+1}} + \sqrt{A_k}} \quad = \quad \frac{a_{k+1}}{\sqrt{A_{k+1}} + \sqrt{A_k}} \\ &\overset{(3.84)}{=} \quad \frac{\sqrt{A_{k+1}}}{\sqrt{L}(\sqrt{A_{k+1}} + \sqrt{A_k})} \quad \geq \quad \frac{\sqrt{A_{k+1}}}{2\sqrt{LA_{k+1}}} \quad = \quad \frac{1}{2\sqrt{L}}. \end{aligned}$$

Thus, telescoping this inequality, we obtain

$$\sqrt{A_k} \quad \geq \quad \sqrt{A_0} + \frac{k}{2\sqrt{L}} \quad = \quad \frac{k}{2\sqrt{L}}.$$

Hence, $A_k \geq \frac{k^2}{4L}$ and we obtain the following optimal rate for the fast gradient method.

Corollary 3.5.2. *Let a_{k+1} be chosen according to* (3.85) *in Algorithm 8.1. Then, the rate of convergence is:*

$$f(x_k) - f^\star \quad \leq \quad \frac{2L\|x_0 - x^\star\|^2}{k^2}, \qquad k \geq 1. \tag{3.86}$$

Remark 3.5.3. This rate matches the lower bound (3.70) up to a numerical constant. Therefore, the fast gradient method is *optimal* for our problem class.

Remark 3.5.4. The complexity of the fast gradient method to obtain $f(x_K) - f^\star \leq \varepsilon$ is

$$K \quad = \quad \left\lfloor \sqrt{\frac{2L\|x_0 - x^\star\|^2}{\varepsilon}} \right\rfloor + 1$$

first-order oracle calls. This is much better than $O(\frac{1}{\varepsilon})$ of the gradient method.

Predefined growth. There are other possibilities in choosing sequence a_{k+1} in order to satisfy (3.83). While solving the quadratic equation (3.84) provides us with the best exact formula, in some of more sophisticated situations (such as stochastic accelerated methods, or second-order acceleration), it is more convenient to specify the growth of coefficients explicitly.

For example, let us specify

$$\boxed{a_k \quad := \quad \frac{1}{2L} k.}$$

Then, we have

$$A_k \;=\; \tfrac{1}{2L}\sum_{i=1}^{k} i \;=\; \tfrac{k(k+1)}{4L},$$

which is the required rate of convergence. It is easy to check that inequality (3.83) is also satisfied for this choice.

Adaptive search. So far, we discussed accelerated schemes with a fixed Lipscthiz constant $L > 0$. However, by analogy with the gradient method, it is possible to employ a simple adaptive search that estimates parameter L adaptively over iterations. The key inequality comes from the last part of our proof, where we required the following condition to hold:

$$\begin{aligned} &\tfrac{1}{2\gamma_k^2 A_{k+1}}\|x_{k+1}-y_k\|^2 + \langle \nabla f(y_k), x_{k+1}-y_k\rangle + f(y_k) \;\geq\; f(x_{k+1}) \\ &\qquad \Leftrightarrow \\ &f(y_k) - f(x_{k+1}) \;\geq\; \langle \nabla f(y_k), y_k - x_{k+1}\rangle - \tfrac{1}{2\gamma_k^2 A_{k+1}}\|x_{k+1}-y_k\|^2 \\ &\qquad\qquad = \;\tfrac{a_{k+1}^2}{2A_{k+1}}\|\nabla f(y_k)\|^2 \;\equiv\; \tfrac{1}{2L_k}\|\nabla f(y_k)\|^2, \end{aligned} \tag{3.87}$$

where $L_k := \frac{A_{k+1}}{a_{k+1}^2}$. As soon as inequality (3.87) holds, for every $k \geq 0$, we get the desired recursion (3.81) satisfied.

Exercise 3.5.1. Develop a version of the fast gradient method with the adaptive search of L, which achieves the optimal convergence rate as in (3.86), up to a numerical constant. What is the total complexity of the resulting algorithm in terms of the gradient and function value computations?

3.5.5 Strongly Convex Optimization

We have proved the following rate of convergence, for the fast gradient method as applied to a convex function f with Lipschitz continuous gradient, $x_k = \mathrm{FGM}_k(f, x_0)$ for $k \geq 0$ iterations, we have

$$f(x_k) - f^\star \;\leq\; \tfrac{2L\|x_0 - x^\star\|^2}{k^2}, \qquad k \geq 1.$$

Now, assume that the objective is additionaly strongly convex with parameter $\mu > 0$. Then, we have the following bound:

$$\tfrac{\mu}{2}\|x_0 - x^\star\|^2 \;\leq\; f(x_0) - f^\star.$$

Combining these two inequalities together, we get:

$$f(x_k) - f^\star \;\leq\; \tfrac{4L}{\mu k^2}(f(x_0) - f^\star).$$

Let us set

$$K \;:=\; \sqrt{\tfrac{8L}{\mu}} \tag{3.88}$$

and run the fast gradient method for this number of iterations. As a result we halve the functional residual:

$$f(x_K) - f^\star \;\leq \tfrac{1}{2}(f(x_0) - f^\star).$$

Now, we can restart this procedure again (starting a new fresh version of the fast gradient method from the output of the previous run). We perform the following iterations, starting from $y_0 := x_0$:

$$\boxed{y_{t+1} \quad = \quad \mathrm{FGM}_K(f, y_t), \qquad t \geq 0,} \tag{3.89}$$

and we need $T = \log_2 \frac{f(x_0)-f^\star}{\varepsilon}$ restarts in order to obtain obtain $f(y_T) - f^\star \leq \varepsilon$.

Theorem 3.5.5. *The total complexity of the fast gradient method with restarts is*

$$\sqrt{\tfrac{8L}{\mu}} \log_2 \tfrac{f(x_0)-f^\star}{\varepsilon} \tag{3.90}$$

first-order oracle calls to minimize a strongly convex smooth function.

This is the same optimal dependence on the condition number $\sqrt{\frac{L}{\mu}}$ as we obtained for the heavy ball method. However, the fast gradient method works for a larger class of all smooth functions (not necessary quadratic).

Using a more advanced reasoning it is possible to obtain the same complexity on the strongly convex functions without restarts. See Exercise 3.8.7 for developing a direct version of the fast gradient method for strongly convex functions (without restarts), that achieves the optimal dependence (3.90) on the problem class parameters.

Note that to perform restarts, we need to know the condition number (3.88), which is not a trivial knowledge in practice.

- In a direct version of the fast gradient method that takes into account strong convexity, we still need to know the constant of strong convexity $\mu \geq 0$.
- We may avoid knowing L by performing an adaptive search, analogously to that one for the basic gradient method.

3.6 Applications: Machine Learning

3.6.1 Generalized Linear Models

Consider the following objective, for a loss function $\ell : \mathbb{R} \to \mathbb{R}$:

$$f(x) \quad = \quad \tfrac{1}{m} \sum_{i=1}^{m} \ell(\langle a_i, x \rangle - b_i) \tag{3.91}$$

and the following optimization problem to train our model:

$$\min_{x \in \mathbb{R}^n} \Big[f(x) + \psi(x) \Big],$$

where $\psi(\cdot)$ is usually some regularizer. $a_1, \ldots, a_m \in \mathbb{R}^n$ and $b_1, \ldots, b_m \in \mathbb{R}$ are given data, and $x \in \mathbb{R}^n$ represents parameters of the model that we want to learn.

Loss functions. We assume loss $\ell(\cdot)$ to be convex and differentiable. Classical examples of loss functions are:

- *Quadratic loss:* $\ell(t) = \frac{1}{2}t^2$, then the objective f is a quadratic multivariate function. It has the Lipschitz derivative with constant $L_\ell = 1$.

- *Logistic loss:* $\ell(t) = \log(1 + e^t)$. It has the Lipschitz derivative with constant $L_\ell = \frac{1}{4}$.
- *Huber loss:*

$$\ell(t) \quad = \quad \begin{cases} \frac{1}{2\delta}t^2, & -\delta \le \tau \le \delta, \\ |t| - \frac{\delta}{2}, & \text{otherwise}, \end{cases}$$

  which is a smooth approximation of the absolute valuer $|t|$, and it has the Lipschitz derivative with constant $L_\ell = \frac{1}{\delta}$.

Choice of the norm. It is convenient to analyze objective (3.91) in a matrix form. Denote by $g : \mathbb{R}^m \to \mathbb{R}$ the separable loss:

$$g(y) \quad = \quad \frac{1}{m} \sum_{i=1}^{m} \ell(y^{(i)}).$$

Then, g has a Lipschitz gradient with respect to the standard Euclidean norm, with constant $L_g = L_\ell$. We can form our data into the matrix $A \in \mathbb{R}^{m \times n}$ and the vector $b \in \mathbb{R}^m$. Then, our objective (3.91) is

$$\begin{aligned} f(x) \quad &= \quad g(Ax + b), \\ \nabla f(x) \quad &= \quad A^\top \nabla g(Ax + b). \end{aligned} \tag{3.92}$$

We have two basic choices for the norm:

- Fix the standard Euclidean norm in $\mathbb{R}^n$, $\|x\| = \langle x, x \rangle^{1/2}$. Then, $L = L_\ell \cdot \|A\|^2$ with respect to this norm. Then, the main step of the fast gradient method reads as:

$$v_{k+1} \quad = \quad v_k - a_{k+1} \nabla f(y_k),$$

  and it requires to perform two matrix-vector products per iteration (one with matrix A and another with matrix $A^\top$).

- Fix the generalized Euclidean norm with $B = A^\top A \in \mathbb{R}^{n \times n}$, that is $\|x\| = \langle Bx, x \rangle^{1/2}$. When the number of data is large ($m \gg n$), we have $B \succ 0$. In practice, we can always use a regularized matrix, $B = A^\top A + \delta I$ for a small $\delta > 0$.

  The Lipschitz constant is much smaller with respect to this norm: $L = L_\ell$ (for $\delta = 0$), and in all cases above it is just an absolute constant that does not depend on the data! It is easy to check that the fast gradient method can be deduced entirely identical for a generalized Euclidean norm. Then, the main iteration of the fast gradient method is the *preconditioned* gradient step:

$$v_{k+1} \quad = \quad v_k - a_{k+1} B^{-1} \nabla f(y_k).$$

  Therefore, we have to invert the matrix B, but we need to do it only once in the beginning before running an algorithm. However, this preconditioning significantly improves overall performance and it also provides us with a clear way of choosing the Lipschitz constant.

Efficient implementation. Note that in the fast gradient method, we choose

$$y_k \quad = \quad \gamma_k v_k + (1 - \gamma_k) x_k, \tag{3.93}$$

where $\gamma_k \in (0, 1)$ is our parameter that depends on a_{k+1} and $A_{k+1} = A_k + a_{k+1}$ as follows: $\gamma_k = \frac{a_{k+1}}{A_{k+1}}$. At the same time, according to our theory, an optimal choice for a_{k+1} that solves the quadratic equation is:

$$a_{k+1} \quad = \quad \frac{1}{2L} \cdot \left(1 + \sqrt{1 + 4A_k L}\right).$$

Now imagine that we choose L in this formula adaptively, possible performing several different tries per iteration. Than, for each try of $\bar{L}$ we have to compute $\bar{a}_{k+1}$, $\bar{\gamma}_k$, and the corresponding gradient $\nabla f(\bar{y}_k)$ at the intermediate point (3.93). Each new computation of the gradient would involve two matrix-vector products, if implemented straightforwardly. However, notice that due to the structure (3.92),

$$\begin{aligned} \nabla f(\bar{y}_k) &= \nabla f(\bar{\gamma}_k v_k + (1-\bar{\gamma}_k)x_k) \\ &\overset{(3.92)}{=} A^\top \nabla g(\bar{\gamma}_k A v_k + (1-\bar{\gamma}_k)Ax_k + b). \end{aligned} \tag{3.94}$$

We see that we do not need to recompute $A\bar{y}_k$ each time: it is enough compute Av_k and Ax_k once, and then use them to construct $A\bar{y}_k$ for all different $\bar{\gamma}_k$. This will save us few matrix-vector products (overall, it can make the method 2-5 times faster!)

The same technique can be applied for a line search used in the classic gradient descent. Imagine we want to perform the classic gradient update: $x^+ = x - \frac{1}{\bar{L}}\nabla f(x)$ for different $\bar{L} > 0$, checking the following inequality:

$$f(x) - f(x^+) \geq \tfrac{1}{2\bar{L}}\|\nabla f(x)\|^2.$$

If implemented straightforwardly, we need to recompute $f(x^+)$ possibly several times per iteration. However, we notice that

$$f(x^+) \overset{(3.92)}{=} g(Ax^+ + b) = g(Ax - \tfrac{1}{\bar{L}}A\nabla f(x) + b).$$

Hence, having computed Ax and $A\nabla f(x)$ once, we can evaluate the new function value $f(x^+)$ very efficiently for many different values of $\bar{L}$. This implementation trick makes it really efficient when solving large-scale problems.

Regularizers. When training a model, we typically solve the following optimization problem,

$$\min_{x\in\mathbb{R}^n} \Big[f(x) + \psi(x) \Big], \tag{3.95}$$

where f is the main part of our objective, and ψ is some *simple regularizer*. The most common examples include:

- *ℓ_2-regularization.* $\psi(x) = \frac{\mu}{2}\|x\|^2$, where $\mu > 0$ is the regularization parameter.

  This makes our objective *strongly convex* and therefore we will always have a unique global solution, and our methods will exhibit fast *linear convergence rates*. We do not need to change anything in the gradient method, as it will automatically adjust to strong convexity. On the contrary, we need to modify the fast gradient method, taking $\mu > 0$ into account (either performing restarts, or a modified choice of parameters).

- *ℓ_1-regularization.* $\psi(x) = \lambda\|x\|_1$, where $\lambda > 0$. This is popular to induce desired sparsity in the solution $x^\star$. Note that full objective (3.95) is still convex, but it becomes non-differentiable. Therefore we cannot technically apply our smooth methods anymore (as we simply cannot compute gradients). Later in the course we will study general methods that can be applied for non-smooth convex optimization. However, typically these methods are much slower than those ones for smooth convex optimization (the complexity becomes $O(\frac{1}{\varepsilon^2})$ instead of $O(\frac{1}{\varepsilon^{1/2}})$, where ε is the target accuracy in terms of the functional residual).

  Luckily, there is a very efficient approach to properly *modify the step* of the basic and the fast gradient methods, which will ensure the same fast convergence rates. We will study this modification later in this lecture in the most general form.

- *Simple constraints.* A related to the previous case, a typical situation is when there are additional *simple* convex constraints $Q \subset \mathbb{R}^n$ that we want to induce into our problem (e.g. that the parameter variable lie in a given ball, a box, a simplex, etc.). This can be modeled by the following artificial function (which is still convex):

$$\psi(x) \quad = \quad \begin{cases} 0, & x \in Q, \\ +\infty, & \text{otherwise.} \end{cases}$$

Then, problem (3.95) becomes constrained optimization problem:

$$\min_{x \in Q} f(x).$$

The modification that will follow from our general construction will be simply to add *projection* after the main step of the fast gradient method:

$$v_{k+1} \quad = \quad \pi_Q(v_k - a_{k+1} \nabla f(y_k)). \tag{3.96}$$

Everything else (including the fast rates of convergence!) remain the same. "Simplicity" of the set Q means that we can actually perform projection (3.96) efficiently.

3.6.2 Nonlinear Models: Neural Networks

In non-linear models (e.g. deep neural networks), we replace the affine part $Ax - b$ in our objective by a nonlinear operator. Therefore, the objective gets the following form:

$$f(x) \quad = \quad \frac{1}{m} \sum_{i=1}^{m} \ell(m_i(x)), \tag{3.97}$$

where $m_i(\cdot)$ represent the output of the model on i-th training example, for a given value of parameters. In case of the simplest neural networks, it has the form of composition of $T \geq 1$ linear layers:

$$m_i(x) \quad = \quad \langle x^{(T)}, \, \sigma(\dots X^{(3)} \sigma(X^{(2)} \sigma(X^{(1)} a_i))) \rangle,$$

where $X^{(1)}, \dots, X^{(T-1)}$ are matrices of parameters of appropriate shapes, and $x^{(T)}$ is a vector (if we want the result of the last layer to be a number), which are all stacked together into

$$x \quad = \quad \left[X^{(1)}, X^{(2)}, \dots, X^{(T-1)}, x^{(T)} \right],$$

and $\sigma(\cdot)$ is a point-wise nonlinear function (typically, either *relu activation* $\sigma(t) = \max\{0, t\}$, *sigmoid*, $\sigma(t) = \frac{1}{1+e^{-t}}$, or *hyperbolic tangent*, $\sigma(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$.

In case where $T = 1$ (one layer), we recover the linear models. Note that, in general ($T \geq 2$), the objective in (3.97) is non-convex and therefore no longer belongs to our problem class. However, both Nesterov's accelerated method and Polyak's heavy-ball method are widely applied in practice and remain the de facto standards for incorporating momentum into training algorithms.

Investigating the properties of objective functions such as (3.97) (e.g., *hidden convexity*) and the dynamics of first-order algorithms applied to them (*overparametrization*, *implicit bias*, *the edge of stability*) remains an active area of research in theoretical deep learning.

3.7 Fully Composite Problems

3.7.1 Motivation

Now, we have a perfect picture about the problem class of smooth convex functions (at least in high dimension $n \to \infty$). We discussed the lower bounds for this problem class and the fast gradient method that is *optimal* (thus its upper bound on the complexity matches the lower bound up to a numerical constant).

This class is reach enough, as it includes, for example, objectives of the form (3.91). Now, before switching to some entirely different situations, we might ask the following question:

how much we can extend our current problem class such that the fast gradient method still works?

We will still work with convex objectives, as convexity was one of the crucial building blocks of the analysis of the fast gradient method. However, what we want is to include at least problems with simple constraints and simple regularizers. As a result, we might allow our objective to be *non-smooth*, but the non-smoothness should be controlled explicitly by our structure.

We want to be abstract enough, as to be able to cover as biggest mathematical formulation of the optimization problem as possible, for which the fast gradient method is still applied. Besides practical importants (as to cover many interesting applications), such generalization is very instructive: as the formulation becomes more abstract, we can use only the very basic mathematical operations (convexity, monotonicity), and it allows us to distinguish the most important steps of the analysis.

3.7.2 Fully Composite Formulation

Now we consider the following optimization problem

$$\min_{x \in Q} \varphi(x), \tag{3.98}$$

where φ has the following *fully composite structure*:

$$\varphi(x) \;=\; F(x, f_1(x), \ldots, f_m(x)), \qquad x \in Q \subseteq \mathbb{R}^n.$$

Set Q is convex (that is, with any two points $x, y \in Q$ it contains entire segment between them: $\lambda x + (1-\lambda)y \in Q$ for any $0 \le \lambda \le 1$).

Smooth components. Now, instead of one main objective function $f : \mathbb{R}^n \to \mathbb{R}$ we have m "smooth components" $f_1, \ldots, f_m$. Each of these functions $f_i : Q \to \mathbb{R}$ is convex and has a Lipschitz continuous gradient with constant $L^{(i)} > 0$, for every $1 \le i \le m$:

$$\|\nabla f_i(x) - \nabla f_i(y)\| \;\le\; L^{(i)}\|x - y\|, \qquad x, y \in Q.$$

We combine all Lipschitz constants into one vector $L = [L^{(1)}, \ldots, L^{(m)}] \in \mathbb{R}^m$. We can stack all these functions into one vector function $f : Q \to \mathbb{R}^m$:

$$f(x) \;=\; \Big[f_1(x), \ldots, f_m(x)\Big]^\top \;\in\; \mathbb{R}^m.$$

These functions are the "difficult parts" of the problem. Thus, we only assume a black-box access to the first-order oracles. With abuse of notation, we denote by $\nabla f(x) \in \mathbb{R}^{m \times n}$ the Jacobian of the mapping f at point $x \in Q \subseteq \mathbb{R}^n$, that is simply composed by the gradient of all f_i:

$$\nabla f(x) \;\; = \;\; \begin{bmatrix} \nabla f_1(x)^\top \\ \nabla f_2(x)^\top \\ \cdots \\ \nabla f_m(x)^\top \end{bmatrix} \;\; \in \;\; \mathbb{R}^{m \times n}.$$

Composite component. With the vector notation, we can write down our target objective as

$$\min_x \Big[\varphi(x) \;\; = \;\; F(x, f(x)), \Big] \tag{3.99}$$

where $F : Q \times \mathbb{R}^m \to \mathbb{R}$ is the "outer" or "composite" component of the objective. The main assumption about F is that it is *simple enough*. Namely, we assume that we can solve the following optimization subproblems efficiently:

$$\min_x F(x, Ax + b) + \tfrac{\alpha}{2}\|x\|^2, \tag{3.100}$$

with an arbitrary affine mapping. That is the linearized version of the original problem.

Moreover, we will need the following formal assumptions about F:

1. F is a *jointly convex* function on $Q \times \mathbb{R}^m$
2. F is *monotone* in the second argument: for any $u, v \in \mathbb{R}^m$ s.t. $u \leq v$ (coordinate-wise), it holds: $F(x, u) \leq F(x, v)$ for any $x \in Q$. Monotonicity ensures that φ as the composition of F and f will be convex.
3. We also assume that F is Lipschitz in second argument:

   $$|F(x, u) - F(x, v)| \;\; \leq \;\; M\|u - v\|, \qquad x \in Q, u, v \in \mathbb{R}^m,$$

   for some constant $M > 0$.

Examples.

1. *Classical unconstrained minimization*: we have only one function $f_1(x)$ and $F(x, u) \equiv u^{(1)}$. Then,

   $$\varphi(x) \;\; = \;\; F(x, f(x)) \;\; \equiv \;\; f_1(x).$$

   The subproblem (3.100) that we require to be able to solve efficiently is the simplest quadratic minimization, as for computing the gradient step, for a certain $a \in \mathbb{R}^n$ and $\alpha > 0$:

   $$\min_x \Big\{ \langle a, x \rangle + \tfrac{\alpha}{2}\|x\|^2 \Big\}$$

2. *Non-smooth regularization*: set $F(x, u) \equiv u^{(1)} + \psi(x)$, where ψ is a given convex (possibly non-smooth) function. Then,

   $$\varphi(x) \;\; = \;\; F(x, f(x)) \;\; \equiv \;\; f(x) + \psi(x).$$

   This example covers both constrained minimization and ℓ_1-regularization. The subproblem (3.100) becomes:

   $$\min_x \Big\{ \langle a, x \rangle + \tfrac{\alpha}{2}\|x\|^2 + \psi(x) \Big\}$$

   and is often called as the *proximal operator* for ψ.

3. *Max-type problems*: let $F(x,u) \equiv \max\limits_{1\le i\le m} u^{(i)}$. Then,

$$\varphi(x) \;=\; F(x,f(x)) \;\equiv\; \max_{1\le i\le m} f_i(x). \tag{3.101}$$

Note that such functions will always be convex (for convex components f_i), but non-smooth. Such objective might appear, for example, if we want to solve the *feasibility problem* with a convex set given by functional inequalities:

$$x^\star \;\in\; Q \;=\; \Big\{x\in\mathbb{R}^n \;:\; f_1(x)\le 0,\dots,f_m(x)\le 0\Big\}. \tag{3.102}$$

Then, we can aim to solve (3.102) by minimizing objective of the form (3.101). As we saw in the first lecture, feasibility problems are very general and are equivalent to minimization problems.

In our method as applied to solve (3.101), we require to compute a solution to the following non-smooth convex subproblem with the explicit structure:

$$\min_x\Big\{\max_{1\le i\le m}\Big[\langle a_i,x\rangle - b_i\Big] + \tfrac{\alpha}{2}\|x\|^2\Big\}.$$

There are efficient solvers that can be applied to solve this subproblem in general.

3.7.3 Composite Fast Gradient Method

Now, we present the fully composite version of the fast gradient method, as to solve problems of the form (3.99). As before, we generate two sequences of points: the main sequence $\{x_k\}_{k\ge 0}$ and the sequence of auxiliary points $\{v_k\}_{k\ge 0}$, both starting from the same initialization, $x_0 = v_0 \in \operatorname{dom}\varphi$. We use a sequence of growing coefficients $\{A_k\}_{k\ge 0}$ starting from $A_0 = 0$.

Algorithm 3.3: *Fully Composite Fast Gradient Method.*

Initialization: $x_0 \in \operatorname{dom}\varphi$. Set $v_0 = x_0$ and $A_0 = 0$. Fix $K \ge 1$.

For $k = 0 \dots K-1$ **iterate:**

1. Choose a new coefficient $a_{k+1} > 0$. Set $A_{k+1} := A_k + a_{k+1}$ and $\gamma_k := \frac{a_{k+1}}{A_{k+1}}$
2. Compute the function values $f(y_k) \in \mathbb{R}^m$ and the Jacobian $\nabla f(y_k) \in \mathbb{R}^{m\times n}$ at the intermediate point $y_k := \gamma_k v_k + (1-\gamma_k)x_k$
3. Compute the new auxiliary point v_{k+1} by solving the following linearized subproblem:

$$v_{k+1} \;=\; \operatorname*{argmin}_x\Big[F(x, f(y_k)+\nabla f(y_k)(x-y_k)) + \tfrac{1}{2a_{k+1}}\|x-v_k\|^2\Big]$$

4. Set a new point from the triangle rule: $x_{k+1} := \gamma_k v_{k+1} + (1-\gamma_k)x_k$

Return x_K

3.7.4 Analysis

Surprisingly, the analysis of Algorithm 3.3 almost identically repeats the analysis of the basic version of the fast gradient method. We only need to be careful when working with the composite outer part $F(\cdot,\cdot)$.

Our goal is to prove by induction the following inequality, for any $k \geq 0$:

$$\frac{1}{2}\|x - x_0\|^2 + A_k\varphi(x) \quad \geq \quad \frac{1}{2}\|x - v_k\|^2 + A_k\varphi(x_k), \qquad x \in \operatorname{dom}\varphi. \tag{3.103}$$

It obviously holds for $k = 0$. Assume that it holds for some $k \geq 0$ and consider one step of the method. We have:

$$\begin{aligned}
\frac{1}{2}\|x - x_0\|^2 + A_{k+1}\varphi(x) &= \frac{1}{2}\|x - x_0\|^2 + a_{k+1}\varphi(x) + A_k\varphi(x_k) \\
&\overset{(3.103)}{\geq} \frac{1}{2}\|x - v_k\|^2 + a_{k+1}\varphi(x) + A_k\varphi(x_k) \\
&= \frac{1}{2}\|x - v_k\|^2 + a_{k+1}F(x, f(x)) + A_k\varphi(x_k) \\
&\geq \frac{1}{2}\|x - v_k\|^2 + a_{k+1}F(x, f(y_k) + \nabla f(y_k)(x - y_k)) + A_k\varphi(x_k),
\end{aligned} \tag{3.104}$$

where in the last inequality we used convexity of each f_i, in a vector component-wise form:

$$f(x) \quad \geq \quad f(y_k) + \nabla f(y_k)(x - y_k) \quad \in \quad \mathbb{R}^m,$$

and the monotonicity of F in the second argument.

Note that by definition, $v_{k+1} = \operatorname{argmin}_x m_k(x)$ is the minimum of the strongly convex model from the right hand side of (3.104):

$$m_k(x) \quad := \quad \frac{1}{2}\|x - v_k\|^2 + a_{k+1}F(x, f(y_k) + \nabla f(y_k)(x - y_k)) + A_k\varphi(x_k).$$

Hence, we have

$$m_k(x) \quad \geq \quad \frac{1}{2}\|x - v_{k+1}\|^2 + m_k^\star,$$

where

$$\begin{aligned}
m_k^\star &= m_k(v_{k+1}) \\
&= \frac{1}{2}\|v_{k+1} - v_k\|^2 + a_{k+1}F(v_{k+1}, f(y_k) + \nabla f(y_k)(v_{k+1} - y_k)) + A_kF(x_k, f(x_k)) \\
&\overset{(*)}{\geq} \frac{1}{2}\|v_{k+1} - v_k\|^2 + a_{k+1}F(v_{k+1}, f(y_k) + \nabla f(y_k)(v_{k+1} - y_k)) \\
&\qquad + A_kF(x_k, f(y_k) + \nabla f(y_k)(x_k - y_k)) \\
&\overset{(**)}{\geq} \frac{1}{2}\|v_{k+1} - v_k\|^2 + A_{k+1}F(x_{k+1}, f(y_k) + \nabla f(y_k)(x_{k+1} - y_k)) \\
&= \frac{1}{2\gamma_k^2}\|x_{k+1} - y_k\|^2 + A_{k+1}F(x_{k+1}, f(y_k) + \nabla f(y_k)(x_{k+1} - y_k)),
\end{aligned}$$

where we used in $(*)$ again convexity of f and the monotonicity of F, and in $(**)$ we used the joint convexity of F.

Now, let us use that components of f have the Lipschitz continuous gradients, in a vector component-wise form:

$$0 \;\leq\; f(x_{k+1}) - f(y_k) - \nabla f(y_k)(x_{k+1} - y_k) \;\leq\; \tfrac{\|x_{k+1}-y_k\|^2}{2} L \;\in\; \mathbb{R}^m, \tag{3.105}$$

and the fact that $F(\cdot,\cdot)$ is Lipschitz in its second argument:

$$F(x_{k+1}, u_1) \;\leq\; F(x_{k+1}, u_2) + M\|u_1 - u_2\|, \qquad u_1, u_2 \in \mathbb{R}^m. \tag{3.106}$$

Hence, we get

$$\begin{aligned} \varphi(x_{k+1}) \;&=\; F(x_{k+1}, f(x_{k+1})) \\ &\overset{(3.106)}{\leq}\; F(x_{k+1}, f(y_k) + \nabla f(y_k)(x_{k+1} - y_k)) + M\|\delta\|, \end{aligned}$$

where

$$\|\delta\| \;:=\; \|f(x_{k+1}) - f(y_k) - \nabla f(y_k)(x_{k+1} - y_k)\| \overset{(3.105)}{\leq} \tfrac{\|x_{k+1}-y_k\|^2}{2}\|L\|.$$

Therefore, combining these observations together, it is sufficient to choose $\frac{1}{\gamma_k^2 A_{k+1}} \geq M\|L\|$ in order to ensure:

$$m_k^\star \;\geq\; A_{k+1}\varphi(x_{k+1}).$$

Thus, we established (3.103) for all $k \geq 1$.

By plugging $x := x^\star$ into (3.103) and choosing a_{k+1} by solving the corresponding quadratic equation, we prove the following result.

Theorem 3.7.1. *Let at each iteration of Algorithm 3.3. we choose*

$$a_{k+1} \;:=\; \tfrac{1}{2\alpha}\Big(1 + \sqrt{1 + 4A_k\alpha}\Big), \quad \textit{where} \quad \alpha \;:=\; M\|L\|.$$

Then,

$$\varphi(x_k) - \varphi^\star \;\leq\; \tfrac{2\alpha\|x_0 - x^\star\|^2}{k^2}, \qquad k \geq 1.$$

Therefore, the fully composite version of the fast gradient method exhibits the same rate of $O(1/k^2)$ as the version for unconstrained minimization from the previous lecture. However, the cost of each step becomes more expensive as it requires solving a non-trivial subproblem with the composite outer part $F(\cdot,\cdot)$.

3.8 Exercises

On Lower Bound

Exercise 3.8.1. In Theorem 3.4.1, we allow the parameters $L > 0$ (the Lipschitz constant), and $K \geq 1$ (the number of iterations) to be arbitrarily fixed. However, another important parameter, that is the initial distance to the solution $\|x_0 - x^\star\|$, appears on the right hand side of (3.62) without any specification. In this problem, we aim to prove the following more general result:

Theorem. Let $L > 0$, $R > 0$, and $K \geq 1$ be fixed. Then, for any first-order optimization algorithm, such that

$$x_{k+1} \;\in\; \operatorname{span}\Big\{\nabla f(x_0), \ldots, \nabla f(x_k)\Big\}, \tag{3.107}$$

running for K iterations, there exists a convex function $f : \mathbb{R}^n \to \mathbb{R}$ with $n \geq 2K + 1$ such that

1. ∇f is Lipschitz with constant L;
2. $\|x_0 - x^\star\| = R$, where $x^\star$ is the minimizer of f;
3. For the output x_K of the algorithm, the following holds:

$$f(x_K) - f^\star \;\geq\; c \cdot \tfrac{LR^2}{(K+1)^2}, \tag{3.108}$$

where $c > 0$ is an absolute numerical constant that depends neither on L, K, nor R.

- Consider the following quadratic objective:

$$f_k(x) \;:=\; \tfrac{\alpha}{2}\Big[\sum_{i=1}^{k-1}(x^{(i)} - x^{(i+1)})^2 \,+\, \sum_{i=k}^{n}(x^{(i)})^2\Big] - \beta x^{(1)}, \qquad x \in \mathbb{R}^n,$$

where $\alpha > 0$ and $\beta > 0$ are parameters. Compute the Lipschitz constant of the gradient $L > 0$ and the squared norm of the solution: $\|x_k^\star\|_2^2$, where $x_k^\star = \operatorname{argmin}_{x \in \mathbb{R}^n} f_k(x)$, for this objective.

- Prove the enhanced theorem.

On Composite Problems

Exercise 3.8.2. Show that function $\varphi(x) := F(x, f(x))$ is convex under the assumptions of the fully composite framework.

Exercise 3.8.3. Consider iterations of the fully composite version of the *basic gradient method*. Starting with some $x_0 \in Q$, we iterate, for $k \geq 0$:

$$x_{k+1} \;=\; \operatorname*{argmin}_{y \in Q}\Big\{F(y, f(x_k) + \nabla f(x_k)(y - x_k)) + \tfrac{\alpha}{2}\|y - x_k\|^2\Big\}, \tag{3.109}$$

where $\alpha > 0$ is a fixed regularization parameter, and $\nabla f(x_k) \in \mathbb{R}^{m \times n}$ denotes the Jacobian of the mapping f computed at point x_k. We aim to prove the convergence rate for this process. Note that due to the composite structure, our previous reasoning for the gradient method is not applicable here, and we need to come up with another proof technique.

- Using the properties of the fully composite problem and definition (3.109) of one step, show that

$$\varphi(x_{k+1}) \;\leq\; \varphi(y) + \tfrac{\alpha}{2}\|y - x_k\|^2, \qquad \forall y \in Q, \tag{3.110}$$

as soon as $\alpha \geq M\|L\|$.

- Substituting $y := \gamma x^\star + (1 - \gamma)x_k$ for $\gamma \in [0, 1]$ into (3.110), show that

$$\varphi(x_{k+1}) - \varphi^\star \;\leq\; (1 - \gamma)(\varphi(x_k) - \varphi^\star) + \tfrac{\alpha D^2}{2}\gamma^2, \tag{3.111}$$

for some $D > 0$, assuming that the initial sublevel set $\mathcal{F}_0 = \Big\{x \in Q : \varphi(x) \leq \varphi(x_0)\Big\}$ is bounded.

- Minimizing the right hand side on (3.111) in $\gamma \in [0, 1]$, show the progress of each iterate in terms of the functional residual $\varphi_k := \varphi(x_k) - \varphi^\star$.
- Show that $\varphi_k = O(1/k)$.

Hint. Consider the case of unconstrained smooth minimization $\varphi(x) \equiv f_1(x)$ first, as the new analysis should recover the rate of the classical gradient descent.

Exercise 3.8.4. Let $F(x,u) := u^{(1)} + \psi(x)$, where ψ is the indicator of a given convex set $Q \subseteq \mathbb{R}^n$:

$$\varphi(x) \;=\; \begin{cases} 0, & x \in Q \\ +\infty, & \text{otherwise.} \end{cases}$$

Show that each iteration (3.109) can be represented as follows:

$$x_{k+1} \;=\; \pi_Q(x_k - \tfrac{1}{\alpha}\nabla f(x_k)),$$

where $\pi_Q(x) := \operatorname{argmin}_{y \in Q} \|y - x\|$ is the projection onto set Q in the Euclidean norm.

Exercise 3.8.5. Let $F(x,u) := u^{(1)} + \lambda\|x\|_1$, for $\lambda > 0$, where $\|x\|_1 := \sum_{i=1}^n |x^{(i)}|$. Thus we are interested to minimize a smooth convex function $f : \mathbb{R}^n \to \mathbb{R}$ with ℓ_1 regularizer:

$$\min_{x \in \mathbb{R}^n} \Bigl\{ \varphi(x) \;\equiv\; f(x) + \lambda\|x\|_1 \Bigr\}.$$

Provide an explicit formula for one step $x_k \mapsto x_{k+1}$ of method (3.109).

Exercise 3.8.6. Let $F(x,u) := u^{(1)} + \frac{\lambda}{3}\|x\|_2^3$, for $\lambda > 0$. That is, the problem of minimizing a smooth convex function $f : \mathbb{R}^n \to \mathbb{R}$ with cubic regularization:

$$\min_{x \in \mathbb{R}^n} \Bigl\{ \varphi(x) \;\equiv\; f(x) + \tfrac{\lambda}{3}\|x\|_2^3 \Bigr\}.$$

Provide an explicit formula for one step $x_k \mapsto x_{k+1}$ of method (3.109).

On Accelerated Method

Exercise 3.8.7. Consider the problem of unconstrained minimization:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a strongly convex differentiable function with the Lipschitz continuous gradient. That is, for any $x, y \in \mathbb{R}^n$ it holds

$$\tfrac{\mu}{2}\|y - x\|^2 \;\le\; f(y) - f(x) - \langle \nabla f(x), y - x \rangle \;\le\; \tfrac{L}{2}\|y - x\|^2,$$

where $0 < \mu \le L$ are two parameters, that we assume to be known. We aim to develop a modification of the fast gradient method on this problem class that achieves the optimal complexity of

$$O\Bigl(\sqrt{\tfrac{L}{\mu}} \log \tfrac{f(x_0) - f^\star}{\varepsilon}\Bigr) \tag{3.112}$$

first-order oracle calls to reach $f(x_k) - f^\star \le \varepsilon$, without restarts.

We prove by induction the following inequality, for any $k \ge 0$:

$$\tfrac{\beta_0}{2}\|x - x_0\|^2 + A_k f(x) \;\ge\; \tfrac{\beta_k}{2}\|x - v_k\|^2 + A_k f(x_k), \qquad x \in \mathbb{R}^n, \tag{3.113}$$

for two sequences of points $\{x_k\}_{k \ge 0}$ and $\{v_k\}_{k \ge 0}$, starting from $x_0 = v_0$, and for two sequences $\{\beta_k\}_{k \ge 0}$, $\{A_k\}_{k \ge 0}$ of increasing non-negative coefficients.

We denote $a_{k+1} := A_{k+1} - A_k > 0$ and $\gamma_k := \frac{a_{k+1}}{A_{k+1}} \in (0, 1]$. The indermediate points are denoted by

$$y_k \;:=\; \gamma_k v_k + (1 - \gamma_k) x_k.$$

- Assuming that (3.113) holds for some $k \geq 0$, show that

$$\frac{\beta_0}{2}\|x - x_0\|^2 + A_{k+1}f(x) \quad \geq \quad m_k(x),$$

where

$$m_k(x) \quad := \quad \frac{\beta_k}{2}\|x - v_k\|^2 + a_{k+1}\Big[f(y_k) + \langle \nabla f(y_k), x - y_k\rangle + \frac{\mu}{2}\|x - y_k\|^2\Big] + A_k f(x_k).$$

- Find the formula for $v_{k+1} = \operatorname{argmin}_x m_k(x)$ and $\beta_{k+1} > \beta_k$ such that

$$m_k(x) \quad \geq \quad \frac{\beta_{k+1}}{2}\|x - v_{k+1}\|^2 + m_k(v_{k+1}), \qquad x \in \mathbb{R}^n.$$

- Show that choosing $x_{k+1} = \gamma_k v_{k+1} + (1 - \gamma_k)x_k$, it is possible to ensure

$$m_k(v_{k+1}) \quad \geq \quad A_{k+1}\Big[f(y_k) + \langle \nabla f(y_k), x_{k+1} - y_k\rangle + \frac{\beta_k}{2\gamma_k^2 A_{k+1}}\|x_{k+1} - y_k\|^2\Big].$$

- Show how to choose $A_{k+1} > A_k$ such that $m_k(v_{k+1}) \geq A_{k+1}f(x_{k+1})$ and thus prove (3.113).
- Obtain the optimal complexity (3.112) for the iterations of this algorithm.
- Write down the iterations of this algorithm using only one sequence of points. Compare it with the iterations of the heavy ball method:

$$z_{k+1} \quad = \quad z_k - \alpha\nabla f(z_k) - \beta(z_k - z_{k-1}).$$

Literature

For an additional reading on convexity, we refer to [20, 3, 36, 41]. Theorem 3.1.4 is from [20], and Theorem 3.1.9 is from [31]. See Section 2.1.5 in [31] for the analysis of the gradient method on smooth convex functions.

See Section 3.2.1 in [37] for the direct convergence analysis of the heavy ball method on strongly convex quadratic functions and the tuning of the momentum parameter $0 \leq \beta \leq 1$.

Our analysis in Section 3.3.2 is inspired by [25], in which the authors establish the convergence of the heavy ball method with restarts for non-convex optimization.

The fast gradient method was developed in [30]. The modern versions of this algorithm can be found in [31].

4. Geometry of Non-Smooth Convex Minimization

In this part of the course, we are moving on to study general problems with convex components that are not necessarily smooth. The geometry of such problems is directly linked to the notion of *convex sets*, which we will review first. Then, we will study two fundamental methods in convex optimization that are both based on the idea of *separation*: the ellipsoid method and the subgradient method.

The ellipsoid method, which belongs to a broader class of cutting-plane schemes, can be viewed as a generalization of the binary search algorithm to the multidimensional case. This method is crucially important in establishing the fundamental theoretical result that *convex optimization* is generally solvable in *polynomial time*.

At the same time, the subgradient method, which can be somewhat misleadingly regarded as "a non-smooth version of gradient descent", relies on similar geometric concepts based on separation. Moreover, the subgradient method is optimal for large-scale non-smooth convex optimization and has inspired many advancements in optimization related to stochasticity and the non-Euclidean geometry of problems, which we study at the end of this part.

4.1 Convexity and Separation . . . 78
4.1.1 Convex Sets . . . 78
4.1.2 Separation Theorem and Subgradients . . . 80
4.1.3 Optimality Condition for Additive Composite Minimization . . . 82
4.2 Binary Search Algorithm . . . 83
4.3 Ellipsoid Method . . . 85
4.3.1 Separation Oracle . . . 86
4.3.2 Cutting Plane Scheme . . . 87
4.3.3 Notion of Size . . . 88
4.3.4 Ellipsoid Method . . . 90
4.4 Subgradient Method: Normalized Stepsizes . . . 92
4.4.1 Optimality Condition for Constrained Minimization . . . 93
4.4.2 Subgradient Method . . . 94
4.4.3 Analysis via Functional Growth . . . 97
4.5 Lower Bound for Non-Smooth Convex Optimization . . . 100
4.5.1 Lower Bound . . . 101
4.5.2 Overview of the Non-Smooth Convex Optimization . . . 104
4.6 Adaptive Stepsizes for Stochastic Methods . . . 105
4.6.1 Stochastic Subgradient Method . . . 107
4.6.2 Adaptive Stepsizes . . . 107
4.7 Smooth Stochastic Optimization II . . . 110
4.7.1 Variance Reduction via Minibatching . . . 112
4.8 Mirror Descent and Accuracy Certificates . . . 113
4.8.1 Application Example: Min-Max Problems . . . 113
4.8.2 Arbitrary Regularizers: Bregman Divergence . . . 117
4.8.3 Mirror Descent . . . 118
4.8.4 Accuracy Certificates . . . 121
4.9 Exercises . . . 122

4.1 Convexity and Separation

4.1.1 Convex Sets

We say that a set $Q \subseteq \mathbb{R}^n$ is *convex*, if for any two points $x, y \in Q$, the whole segment between these points belong to the set:

$$\lambda x + (1-\lambda)y \quad \in \quad Q.$$

Basic properties.

1. *Intersection.* Let $Q_1, Q_2 \subseteq \mathbb{R}^n$ are convex. Then, $Q_1 \cap Q_2$ is also convex. More generally, let $\{Q_\alpha \subseteq \mathbb{R}^n\}$ be *any family* of convex sets indexed by some α. Then their intersection

$$Q \quad = \quad \bigcap_\alpha Q_\alpha$$

is convex. Indeed, let $x, y \in Q$ and $0 \le \lambda \le 1$. Consider arbitrary index α. Then, $x, y \in Q_\alpha$ and due to convexity of Q_α, we have $x_\lambda := \lambda x + (1-\lambda)y \in Q_\alpha$. Therefore, $x_\lambda \in Q$.

2. *The convex hull* of a set $X \subseteq \mathbb{R}^n$ is the smallest convex set that contains X:

$$\mathrm{conv}(X) \quad = \quad \bigcap_\alpha Q_\alpha, \qquad \text{for all convex } \; Q_\alpha \subseteq \mathbb{R}^n \; \text{ s.t. } \; X \subseteq Q_\alpha.$$

3. *Scaling* of a convex set by a positive scalar, $a > 0$,

$$aQ \quad = \quad \{ax \; : \; x \in Q\}$$

is convex for convex Q. This is a particular case of the following construction.

4. *Affine image* of a convex set is a convex set. Let $\mathcal{A}(x) := Ax + b$, for some $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Then,

$$\mathcal{A}(Q) \quad = \quad \Big\{\mathcal{A}(x) \; : \; x \in Q\Big\} \quad \subseteq \quad \mathbb{R}^m$$

is convex. Indeed, let $x, y \in \mathcal{A}(Q)$ and $0 \le \lambda \le 1$. Thus, for some $\bar{x}, \bar{y} \in Q$ we have $x = \mathcal{A}(\bar{x})$ and $y = \mathcal{A}(\bar{y})$. By convexity of Q we have that $\bar{x}_\lambda := \lambda\bar{x} + (1-\lambda)\bar{y} \in Q$, and therefore, since affine mapping preserves convex combinations, we have

$$x_\lambda \quad := \quad \lambda x + (1-\lambda)y \quad = \quad \lambda\mathcal{A}(\bar{x}) + (1-\lambda)\mathcal{A}(\bar{y})$$

$$= \quad \mathcal{A}(\lambda\bar{x} + (1-\lambda)\bar{y}) \quad = \quad \mathcal{A}(\bar{x}_\lambda).$$

Hence, $x_\lambda \in \mathcal{A}(Q)$.

Examples of convex sets.

1. *Hyperplane*: $Q = \{x \in \mathbb{R}^n \; : \; \langle a, x\rangle = b\}$ and *half-space*: $Q = \{x \in \mathbb{R}^n \; : \; \langle a, x\rangle \le b\}$, for $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, are convex sets.

2. *Affine subspace*: $Q = \big\{x \in \mathbb{R}^n \; : \; Ax = b\big\}$, for any $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. It is convex as intersection of the hyperplanes.

3. *Polyhedron*: $Q = \{x \in \mathbb{R}^n \ : \ Ax \leq b\}$ is convex as it is a finite intersection of half-spaces. This set is fundamental to linear programming. At the same time, a central fact of convex analysis is that *any* closed convex set can be represented as the intersection (possibly infinite) of half-spaces.

4. *Ball in any norm*:
$$Q \ = \ \Big\{x \in \mathbb{R}^n \ : \ \|x\| \leq R\Big\}.$$
Indeed, let $x, y \in Q$. Then, for $x_\lambda = \lambda x + (1-\lambda)y$ with $0 \leq \lambda \leq 1$ we have
$$\|x_\lambda\| \ \leq \ \lambda\|x\| + (1-\lambda)\|y\| \ \leq \ \lambda R + (1-\lambda)R \ = \ R.$$

5. *Ellipsoid:*
$$Q \ = \ \Big\{ \ x \in \mathbb{R}^n \ : \ \langle H(x-x_0), x-x_0\rangle \leq 1 \ \Big\}.$$
for some $H = H^\top \succ 0$.

   - It is an image of the unit Euclidean ball under affine transformation:
$$Q \ = \ \Big\{Bu + x_0 \ : \ u \in \mathbb{R}^n \ \text{ s.t. } \langle u, u\rangle \leq 1\Big\},$$
     where $B := H^{-1/2}$.

6. *Cone of positive definite matrices*: $\mathbb{S}^n_+ = \Big\{X \in \mathbb{S}^n \ : \ X \succeq 0\Big\}$.
   Indeed, let $X, Y \in \mathbb{S}^n_+$. Then $\lambda X \in \mathbb{S}^n_+$ (for any $\lambda \geq 0$) and $X+Y \in \mathbb{S}^n_+$ by the basic properties of eigenvalues. Hence, $\lambda x + (1-\lambda)y \in \mathbb{S}^n_+$ as well, for $0 \leq \lambda \leq 1$. So $\mathbb{S}^n_+$ is a *convex cone*.

7. *Semidefinite programming*. Q is an intersection of $\mathbb{S}^n_+$ cone with affine hyperplanes:
$$Q \ = \ \Big\{X \in \mathbb{S}^n_+ \ : \ \langle A_1, X\rangle = b_1, \ldots, \langle A_m, X\rangle = b_m\Big\}.$$
   If we additionally restrict matrix X to be diagonal, we get the feasible set in *linear programming*:
$$Q \ = \ \Big\{x \in \mathbb{R}^n_+ \ : \ \langle a_1, x\rangle = b_1, \ldots, \langle a_m, x\rangle = b_m\}.$$

Epigraph of convex function. We recall that a function $f : \operatorname{dom} f \to \mathbb{R}$, where $\operatorname{dom} f \subseteq \mathbb{R}^n$, is convex, if
$$f(\lambda x + (1-\lambda)y) \ \leq \ \lambda f(x) + (1-\lambda)f(y), \qquad x, y \in \operatorname{dom} f, \ \ 0 \leq \lambda \leq 1. \tag{4.1}$$
This definition implies that the domain of a convex function must be a *convex set*, otherwise, the left hand side in (4.1) is not defined. Note that this general definition works even if the function is non-differentiable.

Now, we can look at the following set, called *epigraph*:
$$\operatorname{epi} f \ = \ \Big\{(x,t) \in Q \times \mathbb{R} \ : \ f(x) \leq t\Big\}. \tag{4.2}$$

Proposition 4.1.1. *Function f is convex $\Leftrightarrow$ $\operatorname{epi} f$ is a convex set.*

Exercise 4.1.1. Prove Proposition 4.1.1.

Therefore, we can associate any convex function with a convex set, its epigraph. And this is a systematic way to generate convex sets: we can take any convex function we already know, and we obtain a non-trivial convex set (4.2).

In non-smooth convex optimization, we rather view functions through the lens of convex sets and their underlying geometry.

4.1.2 Separation Theorem and Subgradients

The most important geometrical facts about convex sets are the so-called *separation theorems*. There are many variations of these. For our purposes, it is convenient to use the following version, which we state without the proof (see, e.g. [20]).

Theorem 4.1.2. *Let $Q \subseteq \mathbb{R}^n$ be convex and assume that its interior is non-empty: $\operatorname{int} Q \neq \varnothing$. Let $x \in \mathbb{R}^n$ do not belong to the interior of Q: $x \notin \operatorname{int} Q$. Then, x can be separated from Q by a linear function, i.e. there exists $\ell \in \mathbb{R}^n$, $\ell \neq 0$:*

$$\langle \ell, x \rangle \quad \geq \quad \langle \ell, y \rangle, \qquad y \in Q. \tag{4.3}$$

When separation occurs at a boundary point $x \in \partial Q$, we call such hyperplane *supporting* to the set Q. Supporting hyperplanes provide us with the main search directions for algorithms in convex optimization. Another consequence of Theorem 4.1.2 is the following statement.

Corollary 4.1.3. *Any closed (open) convex set $Q \subseteq \mathbb{R}^n$ is equal to the intersection of closed (open) hyperplanes containing it.*

Now, we can look at a convex function through the lens of their epigraphs. Let $f : Q \to \mathbb{R}$ be convex and consider $x \in Q \subset \mathbb{R}^n$. Then, point $(x, f(x))$ belongs to the boundary of the epigraph: $(x, f(x)) \in \partial \operatorname{epi} f$. Hence, there exists a supporting hyperplane such that it separates epigraph. Such hyperplane is called the subgradient.

Definition 4.1.1. We say that a vector $g \in \mathbb{R}^n$ is a subgradient of f at point x if

$$f(y) \quad \geq \quad f(x) + \langle g, y - x \rangle, \qquad y \in \operatorname{dom} f.$$

The set of all subgradients is denoted by $\partial f(x)$ and is called subdifferential of f. We denote by $f'(x) \in \partial f(x)$ any particular selection of a subgradient.

Note that by this definition we might have several subgradients at the same point, which happens when the function is non-differentiable. It might also be the case that there are not subgradients at all: $\partial f(x) = \varnothing$. However, it appears that such unfortunate situations might only happen at the boundary of our domain.

In this course, we will always assume that $f : Q \to \mathbb{R}$ where $Q \subseteq \mathbb{R}^n$ is *open set*. In such situations a subgradient always exists for any $x \in Q$.

Theorem 4.1.4. *Let $f : Q \to \mathbb{R}$ be a convex function defined on an open set $Q \subseteq \mathbb{R}^n$. Then, for any $x \in Q$ we have $\partial f(x) \neq \varnothing$.*

Proof. Consider the point $y = (x, f(x)) \in \partial \operatorname{epi} f \subseteq \mathbb{R}^{n+1}$ from the boundary of the epigraph. By the separation theorem, there exists a non-zero vector $[\ell_0, \ell]^\top \in \mathbb{R}^{n+1}$ where $\ell_0 \in \mathbb{R}$ and $\ell \in \mathbb{R}^n$, such that

$$\ell_0(f(x) - t) + \langle \ell, x - y \rangle \quad \overset{(4.3)}{\geq} \quad 0, \qquad \forall y \in Q \ \text{ and } \ \forall t \geq f(y). \tag{4.4}$$

Substituting $y := x$ and $t > f(x)$ we have $\ell_0(f(x) - t) \geq 0$. Therefore, we conclude that $\ell_0 \leq 0$.

Let us prove that $\ell_0 < 0$ (strictly). Assume that $\ell_0 = 0$ and take $y := x + \varepsilon \frac{\ell}{\|\ell\|}$ for a sufficiently small $\varepsilon > 0$ so that $y \in Q$. We have

$$\langle \ell, x - y \rangle \quad = \quad -\varepsilon \|\ell\| \quad < \quad 0,$$

which contradicts (4.4). Therefore, $\ell < 0$. Dividing inequality (4.4) by it and rearranging the terms, we get, for $t := f(y)$:

$$f(y) \quad \geq \quad f(x) + \langle \tfrac{\ell}{\ell_0}, y - x \rangle.$$

Thus, $\frac{\ell}{\ell_0} \in \partial f(x)$. □

Properties of Subdifferentials.

- *Sum of two convex functions.* Let $f(x) = \alpha f_1(x) + \beta f_2(x)$. Then $\partial f(x) = \alpha \partial f_1(x) + \beta \partial f_2(x)$.
- *Pointwise maximum* of any family of convex functions:

  $$f(x) \quad = \quad \max_{\alpha} f_\alpha(x),$$

  is convex as its epigraph is the intersection of convex sets:

  $$\operatorname{epi} f(x) \quad = \quad \bigcap_{\alpha} \operatorname{epi} f_\alpha.$$

  In general, we have:

  $$\partial f(x) \quad \supseteq \quad \operatorname{conv}\Big\{ \partial f_\alpha(x) \ : \ \alpha \text{ s.t. } f(x) = f_\alpha(x) \Big\}. \tag{4.5}$$

  Indeed, let us fix $x \in Q$ and an α s.t. the maximum is achieved: $f(x) = f_\alpha(x)$. Then, for any $y \in Q$ it holds:

  $$f(y) \quad \geq \quad f_\alpha(y) \quad \geq \quad f_\alpha(x) + \langle f'_\alpha(x), y - x \rangle \quad = \quad f(x) + \langle f'_\alpha(x), y - x \rangle.$$

  Hence, $f'_\alpha(x) \in \partial f(x)$. The exact equation in (4.5) holds, e.g., when the family $\{f_\alpha\}$ is *finite*.
- *Differentiable function.* Let $f : Q \to \mathbb{R}$ be differentiable and convex. Then,

  $$f(x + h) - f(x) \quad \geq \quad \langle g, h \rangle \quad \text{for} \quad g \in \partial f(x)$$

  and

  $$f(x + h) - f(x) \quad = \quad \langle \nabla f(x), h \rangle + o(\|h\|).$$

  Subtracting this equation from the inequality above, we get

  $$0 \quad \geq \quad \langle g - \nabla f(x), h \rangle + o(\|h\|), \qquad h \in \mathbb{R}^n.$$

  We conclude that $g = \nabla f(x)$. So $\partial f(x) = \{\nabla f(x)\}$.

Examples.

1. Let $f(x) = \|x\|_2 = \sqrt{\langle x, x \rangle}$. This function is differentiable everywhere except 0. Therefore, we have
$$\partial \|\cdot\|_2(x) \;=\; \{\nabla f(x)\} \;=\; \Big\{\tfrac{1}{\|x\|_2} x\Big\}, \qquad x \neq 0. \tag{4.6}$$
Computing the subdifferential at 0 means to find all vectors $s \in \mathbb{R}^n$ (subgradients) such that
$$\|x\|_2 \;\geq\; \langle s, x \rangle, \qquad x \in \mathbb{R}^n. \tag{4.7}$$
By Cauchy-Schwarz inequality, we know that any $s \in \mathbb{R}^n$ such that $\|s\|_2 \leq 1$ satisfies (4.7). At the same time, plugging in $x := s$ into (4.7) we ensure that this is also a necessary condition for s to be a subgradient. Hence, we justified that
$$\partial \|\cdot\|_2(0) \;=\; \Big\{s \in \mathbb{R}^n \;:\; \|s\|_2 \leq 1\Big\}. \tag{4.8}$$

2. The formula (4.8) works for an arbitrary norm $\|\cdot\|$:
$$\partial \|\cdot\|(0) \;=\; \Big\{s \in \mathbb{R}^n \;:\; \|s\|_* \leq 1\Big\}, \tag{4.9}$$
where $\|\cdot\|_*$ is the dual norm. However, formula (4.6) is no longer true, as $f(x) = \|x\|$ is, in general, might nor be differentiable (e.g. consider $\|\cdot\|_1$ or $\|\cdot\|_\infty$ norm).

3. Let $f(x) = \max\limits_{1 \leq i \leq m} \Big[\langle a_i, x \rangle - b_i\Big] = \max\limits_{1 \leq i \leq m} f_i(x)$, where $f_i(x) = \langle a_i, x \rangle - b_i$ is affine. We have $\nabla f_i(x) = a_i$ and
$$\partial f(x) \;=\; \mathrm{conv}\Big\{a_i \;:\; 1 \leq i \leq m \;\text{ s.t. }\; f(x) = \langle a_i, x \rangle - b_i\Big\}.$$

4. Let $f(X) = \lambda_{\max}(X)$, for $X \in \mathbb{S}^n$. It is convex as an (infinite) maximum of linear functions:
$$f(X) \;=\; \max_{u \in \mathbb{R}^n \,:\, \|u\| = 1} \langle Xu, u \rangle \;=\; \max_{u \in \mathbb{R}^n \,:\, \|u\| = 1} \mathrm{tr}(Xuu^\top)$$
From this representation, we immediately obtain a way to compute its subgradients:
$$\partial f(X) \;\supseteq\; \mathrm{conv}\Big\{uu^\top \;:\; u \in \mathbb{R}^n \text{ s.t. } Xu = \lambda_{\max}(X) u\Big\}.$$

4.1.3 Optimality Condition for Additive Composite Minimization

Consider the following problem of *additive composite optimization*, that often appears in practice:
$$\min_{x \in Q} \Big[F(x) \;=\; f(x) + \psi(x)\Big] \tag{4.10}$$
We can set $Q := \mathrm{dom}\, \psi$ and assume that ψ is a *general convex function* (possibly non-differentiable). At the same time, f is differentiable. We can directly prove the following optimality condition for a minimum of this problem.

Theorem 4.1.5. *A point $x^\star$ is a global minimum of* (4.10) *if and only if*
$$\langle \nabla f(x^\star), x - x^\star \rangle + \psi(x) \;\geq\; \psi(x^\star), \qquad x \in Q. \tag{4.11}$$

Proof. Indeed, if (4.11) holds, then, using convexity of f we have:

$$\begin{aligned} F(x) &= f(x) + \psi(x) \\ &\geq f(x^\star) + \langle \nabla f(x^\star), x - x^\star \rangle + \psi(x) \\ &\overset{(4.11)}{\geq} f(x^\star) + \psi(x^\star) = F(x^\star). \end{aligned}$$

Hence, $x^\star$ is the global minimum.

Now, assume that $x^\star$ is the global minimum of (4.10) and our goal is to prove (4.11). For a sufficiently small $\alpha > 0$, we have

$$\begin{aligned} \langle \nabla f(x^\star), x - x^\star \rangle + \psi(x) - \psi(x^\star) &= \tfrac{1}{\alpha}\Big[f(x^\star + \alpha(x - x^\star)) - f(x^\star)\Big] + \psi(x) - \psi(x^\star) + o(1) \\ &= \tfrac{1}{\alpha}\Big[F(x^\star + \alpha(x - x^\star)) - F(x^\star)\Big] + \tfrac{1}{\alpha}\Big[\alpha\psi(x) + (1-\alpha)\psi(x^\star) - \psi(x^\star + \alpha(x - x^\star))\Big] + o(1) \\ &\geq 0. \end{aligned}$$

□

Corollary 4.1.6. *We have proved that*

$$-\nabla f(x^\star) \in \partial\psi(x^\star).$$

In practice, it implies that the rule "set gradient to zero" works as well:

$$F'(x^\star) = \nabla f(x^\star) + \psi'(x^\star) = 0,$$

where $\psi'(x^\star) \in \partial\psi(x^\star)$ is some subgradient.

Corollary 4.1.7. *From the proof we see that if f is a non-convex differentiable function, and $x^\star$ is a local minimum of* (4.10)*, then* (4.11) *holds, as a necessary condition for local optimality.*

Corollary 4.1.8. *Let $\psi(x)$ be the indicator of a convex set Q:*

$$\psi(x) = \begin{cases} 0, & x \in Q \\ +\infty, & x \notin Q. \end{cases}$$

Then, our problem is $\min\limits_{x \in Q} f(x)$ and condition (4.11) *implies that:*

$$\langle \nabla f(x^\star), x - x^\star \rangle \geq 0, \qquad x \in Q.$$

Geometric interpretation of this inequality is that the gradient at the optimum, $\nabla f(x^\star)$ separates Q from the sublevel set $\mathcal{F} = \Big\{x \in \operatorname{dom} f \ : \ f(x) \leq f(x^\star)\Big\}$ — see Section 4.4.1 for an illustration.

4.2 Binary Search Algorithm

Problem class. We consider 1-dimensional optimization problem:

$$\min_{a \leq x \leq b} f(x),$$

where $f : [a, b] \to \mathbb{R}$ is a convex continuous function, and our feasible set $Q = [a, b]$ is the segment.

- Oracle: $x \mapsto (f(x), f'(x))$ where $f'(x)$ is some subgradient $f'(x) \in \partial f(x) \subseteq \mathbb{R}$.
- The goal: to find $\bar{x}$ s.t. $f(\bar{x}) - f^\star \leq \varepsilon$.

Algorithm. Let us analyze the simplest binary search algorithm, which is familiar to everyone.

We start with the initial segment $\ell_0 = a, r_0 = b$. Set $x_0 = \frac{\ell_0 + r_0}{2}$, the middle point, and compute $f'(x_0)$. By convexity, we have the following inequality,

$$f(y) \quad \geq \quad f(x_0) + f'(x_0)(y - x_0), \qquad y \in [a, b].$$

There are the following three options:

1. $f'(x_0) = 0$. Then x_0 is the desirable global minimum: $x_0 = x^\star$. However, in practice it is better to never conduct such exact check due to machine precision errors.

2. $f'(x_0) < 0$ (the function is decreasing at x_0). Thus for any $y \in [a, x_0]$ we have $f'(x_0)(y - x_0) \geq 0$ and hence
$$f(y) \quad \geq \quad f(x_0).$$

3. $f'(x_0) > 0$ (the function is increasing at x_0). Then, for any $y \in [x_0, b]$ we have
$$f(y) \quad \geq \quad f(x_0).$$
In both cases, we know how to switch to a smaller segment.

Algorithm 4.1: *Binary Search Algorithm.*

Initialization: $\ell_0 = a$, $r_0 = b$. Fix $K \geq 1$.

For $k = 0 \dots K - 1$ **iterate:**

1. Set $x_k = \frac{1}{2}(\ell_k + r_k)$
2. Compute $f'(x_k) \in \partial f(x_k) \subseteq \mathbb{R}$
3. **If** $f'(x_k) < 0$ then set $\ell_{k+1} = x_k$ and $r_{k+1} = r_k$ **else** set $\ell_{k+1} = \ell_k$ and $r_{k+1} = x_k$.

Return a point $\bar{x}_K$ among $\{x_0, \dots, x_K\}$ with the smallest function value: $f(\bar{x}_K) = \min\limits_{0 \leq i \leq K} f(x_i)$.

Note that returning the last point x_K is not a good idea in general, if we are interested in a small functional residual.

Analysis. The analysis of the method is based on the following simple observations, which are immediate to check:

Proposition 4.2.1. *Denote by $G_k := [\ell_k, r_k]$ our localization set. It holds $|G_k| = r_k - \ell_k = \frac{b-a}{2^k}$.*

Proposition 4.2.2. *For any solution $x^\star$, we have $x^\star \in G_k$.*

Proposition 4.2.3. *For any $y \in Q \setminus G_k$, we have $f(y) \geq f(\bar{x}_k)$ (the function value outside the localizer is always greater than the best seen point).*

Thus, from the construction of the binary search we immediately obtain very fast linear rate of decrease of the localizer set $|G_k| \to 0$ (Proposition 4.2.1). However, our initial goal was to establish convergence in terms of the functional residual. For that, we employ the following simple machinery.

We denote by V the *variation of the function* over our initial set $Q = [a, b]$:

$$V \;=\; \max_{x \in Q} f(x) - \min_{x \in Q} f(x) \;=\; \max_{x \in Q} f(x) - f^\star$$

For some $\gamma \in [0, 1]$ consider the *contraction* of the initial set:

$$Q_\gamma \;:=\; \gamma Q + (1 - \gamma) x^\star.$$

We have: $|Q_\gamma| = \gamma |Q| = \gamma(b - a)$. Let $1 \geq \gamma > 2^{-k}$. Then, there exists

$$y \;=\; \gamma z + (1 - \gamma) x^\star \;\in\; Q_\gamma, \;\; z \in Q,$$

such that $y \notin G_k$. Then, we get, by convexity:

$$f(\bar{x}_k) \;\leq\; f(y) \;\leq\; \gamma f(z) + (1 - \gamma) f^\star \;=\; \gamma (f(z) - f^\star) + f^\star \;\leq\; \gamma V + f^\star.$$

By taking the limit $\gamma \to 2^{-k}$ we prove the following theorem.

Theorem 4.2.4. *After $K \geq 0$ iterations of the binary search algorithm, it holds:*

$$f(\bar{x}_K) - f^\star \;\leq\; \tfrac{V}{2^K}.$$

We see that this is a very fast linear rate with a constant factor. In order to achieve $f(\bar{x}_K) - f^\star \leq \varepsilon$ it is enough to perform

$$K \;=\; \log_2 \tfrac{V}{\varepsilon}$$

oracle calls.

It appears that this complexity is *optimal* for the univariate case (there is no better algorithm than binary search in general for one-dimensional convex minimization). See Section 1 in [27]. In the next section, we study a generalization of the binary search to multivariate case, called the *ellipsoid method*.

4.3 Ellipsoid Method

Problem formulation. Our goal is to solve the following convex optimization problem,

$$\min_{x \in Q} f(x) \tag{4.12}$$

where $Q \subseteq \mathbb{R}^n$ is a convex set, and $f : Q \to \mathbb{R}$ is a convex function. We consider a general situation, when f might not be differentiable, and set Q can also be a general convex set with difficult geometry.

To quantify the complexity of this problem, we need some regularity assumptions:

- For objective function f, we denote by V its *variation* over the set Q:

$$V \;:=\; \max_{x \in Q} f(x) - \min_{x \in Q} \;=\; \max_{x \in Q} f(x) - f^\star,$$

  and we assume that V is bounded: $V < +\infty$.

- For set Q, we assume that it is *bounded* and has a *nonempty interior* $\operatorname{int} Q \neq \varnothing$. When solving unconstrained optimization problems, we can always introduce an auxiliary ball of sufficiently big radius to satisfy this assumption. Quantitatively, we assume that there exist $0 < r \leq R < +\infty$ and $\bar{x} \in \operatorname{int} Q$ such that:

$$B_r(\bar{x}) \quad \subseteq \quad Q \quad \subseteq \quad B_R(\bar{x}),$$

  where $B_\alpha(\bar{x}) := \{x \in \mathbb{R}^n \,:\, \|x - \bar{x}\|_2 \leq \alpha\}$ is the Euclidean ball of radius $\alpha \geq 0$.

  The ratio $\frac{R}{r} \geq 1$ is sometimes called the *asphericity* of Q, and can be seen as the "condition number" of the set.

Thus, the parameters V, r, R will describe the complexity of solving (4.12), but as we will see, the dependence on them is rather weak. The main complexity parameter will be the dimension n.

As before, our goal is to find a point $\bar{x} \in Q$ such that

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

4.3.1 Separation Oracle

We assume that we have an access to the following *separation oracle* to solve (4.12):

$$\mathcal{O}(x) \quad = \quad \begin{cases} f'(x) \quad \in \quad \partial f(x), & \text{if } \; x \in \operatorname{int} Q, \\ \\ s_Q(x), & \text{otherwise,} \end{cases} \tag{4.13}$$

where $s_Q(x) \in \mathbb{R}^n$, $s_Q(x) \neq 0$ is a vector that separates point x from Q. Thus, by the definition of $s_Q(x)$, it holds:

$$\langle s_Q(x), x - y \rangle \quad \geq \quad 0, \qquad y \in Q. \tag{4.14}$$

We know from the separation theorem that such a vector always exists for $x \notin \operatorname{int} Q$, but it may not be unique. A subgradient vector $f'(x) \in \partial f(x)$ is not uniquely defined. Therefore, we might have many options to actually implement the oracle $\mathcal{O}(x)$, and any particular selection works for us. Note that by the definition of the subgradient, we have:

$$\langle f'(x), x - y \rangle \quad \geq \quad f(x) - f(y), \qquad \forall x, y \in \operatorname{dom} f. \tag{4.15}$$

Without loss of generality, we can always assume that $f'(x) \neq 0$. Otherwise, if $f'(x) = 0$, inequality (4.15) implies that $x = x^\star$ is the global optimum and we can return as the result of a method.

At the same time, when $f'(x) \neq 0$, the subgradient also provides us with a *separation* of our space into two halves, and we know which half contains the optimum:

$$x^\star \quad \in \quad \Big\{ \; y \quad : \quad \langle f'(x), x - y \rangle \geq 0 \; \Big\}.$$

Geometrically, inequality $\langle f'(x), x - y \rangle \geq 0$ separates the sublevel set of the objective f at point x.

Example 4.3.1. Consider the set

$$Q \quad = \quad \{y \in \mathbb{R}^n \,:\, \langle a, y \rangle \leq b\},$$

for given $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, which is a half space. Assume $x \notin \operatorname{int} Q$. What will be a result of the separation oracle $s_Q(x)$? It is easy to see that

$$s_Q(x) \;:=\; a,$$

will do the job. Indeed, for $x \notin \operatorname{int} Q$ we have $\langle a, x\rangle \geq b$ At the same time, for any $y \in Q$ we have

$$\langle a, y\rangle \;\leq\; b \;\leq\; \langle a, x\rangle,$$

and condition (4.14) is satisfied.

Example 4.3.2. Using a separation oracle for a half space, it is very easy to implement the separation oracle for the polyhedron, which is the intersection of half spaces:

$$Q \;=\; \big\{y \in \mathbb{R}^n \;:\; Ay \leq b\big\} \;=\; \big\{y \in \mathbb{R}^n \;:\; \langle a_1, y\rangle \leq b_1, \ldots, \langle a_m, y\rangle \leq b_m\big\},$$

where $a_1, \ldots, a_m \in \mathbb{R}^n$ are rows of the matrix $A \in \mathbb{R}^{m\times n}$. For a given x, we need to go and check whether any of the inequalities is violated. If so, we return the vector a_i that correspond to the violated inequality. Thus, separation oracle for the linear programming can be implemented in $O(nm)$ arithmetical operations.

In a similar vein separation oracles can be developed for other types of standard convex sets.

4.3.2 Cutting Plane Scheme

The idea of cutting plane schemes in optimization is to use separation oracle to "cut" our search space into two halves and continue the search in one of them. This is a generalization of the binary search from the univariate case.

However, in one-dimensional case ($n = 1$), we usually have more or less one natural possibility:

- **The search region**, that we also call the the *localizer* G_k at iteration $k \geq 0$, is a segment:

$$G_k \;=\; [\ell_k, r_k], \qquad \ell_k < r_k.$$

  We ensure the invariant that $x^\star \in G_k$ (that is why it is called the localizer), and that

$$\operatorname{size}(G_k) \;:=\; r_k - \ell_k \;\to\; 0.$$

- **The next point**, x_k which defines where we access the oracle is the midpoint of the segment:

$$x_k \;=\; \tfrac{\ell_k + r_k}{2}.$$

Now, we want to generalize this construction to the multivariate case. For $n \geq 2$, we obtain great flexibility in how to define G_k and x_k, and there are many options that lead to particular implementations of the so-called *cutting plane scheme*.

The main obstacle in the general case $n \geq 2$ is that the "shape" of G_k might become quite difficult. For example, for a set of previous points $\{x_0, \ldots, x_k\}$ with known oracle information $g_i = \mathcal{O}(x_i)$, for $0 \leq i \leq k$, it is natural to construct the following set, which is the intersection of all separating half spaces:

$$G_{k+1} \;=\; \Big\{\, y \in \mathbb{R}^n \;:\; \langle g_0, x_0 - y\rangle \geq 0,\; \ldots,\; \langle g_k, x_k - y\rangle \geq 0 \,\Big\}. \tag{4.16}$$

Then, clearly $x^\star \in G_{k+1}$. However, defined this way, set (4.16) becomes a polyhedron and a question of just finding a point $x_{k+1} \in G_{k+1}$ (the feasibility problem) is equivalent to a linear programming problem. But we do not want "any" point: we rather want a point such that sizes would go to zero: $\text{size}(G_k) \to 0$, which becomes it even more difficult to decide on x_{k+1}.

The idea to overcome this is as follows: instead of adding unstructured cuts (4.16), at every iteration $k \geq 0$ we preserve a structure of the set to be "simple by rich enough", defined by $E_k \subseteq \mathbb{R}^n$. In our case E_k will be an Ellipsoid centered at x_k:

$$E_k \;\; := \;\; \Big\{ y \in \mathbb{R}^n \;:\; \langle A_k^{-1}(y - x_k), y - x_k \rangle \;\; \leq \;\; 1 \Big\},$$

which is given by a positive definite symmetric matrix $A_k = A_k^\top \succ 0$. Our invariant remains that a solution always belongs to our localizers:

$$x^\star \;\; \in \;\; E_k, \qquad k \geq 0.$$

Then, at each iterate we call the separation oracle at the center x_k, and perform a single cut of E_k:

$$G_{k+1} \;\; := \;\; \Big\{ \; y \in E_k \;\; : \;\; \langle g_k, x_k - y \rangle \geq 0 \; \Big\}.$$

Thus, $x^\star \in G_{k+1}$, but G_{k+1} is not an ellipsoid anymore. To keep a simple structure, we have to *find a new ellipsoid* E_{k+1} that contains the halved one:

$$E_{k+1} \;\; \supseteq \;\; G_{k+1},$$

at that constitute one step of the method.

To initialize the method we have to choose the initial ellipsoid E_0. Usually it is taken to be a large enough Euclidean ball, $A_0 := \frac{1}{R} I$ around some given point $x_0 \in Q$, such that the feasible set is entirely contained in it:

$$E_0 \;\; := \;\; B_R(x_0) \;\; \supseteq \;\; Q \;\; =: \;\; G_0,$$

such R exists by our assumption.

4.3.3 Notion of Size

Now, to be able to show that the method converges sufficiently fast, we want to ensure that our sets $\{E_k\}_{k \geq 0}$ are getting smaller and smaller in "size" with a certain good rate:

$$\text{size}(E_k) \;\; \to \;\; 0 \qquad \text{with} \qquad k \to +\infty. \tag{4.17}$$

For the ellipsoid method, we use the volume as a measure of the size, for any compact convex set with non-empty interior $K \subset \mathbb{R}^n$:

$$\text{size}(K) \;\; := \;\; (\text{Vol}(K))^{1/n}. \tag{4.18}$$

Another possible choice for a "size" is the diameter of the set in a given norm.

We require the following properties to be satisfied by a function $\text{size}(\cdot)$, which are obviously satisfied for the volume function (4.18).

1. *Monotonicity.* If $K_1 \subseteq K_2$ then: $\text{size}(K_1) \leq \text{size}(K_2)$.

2. *Homogeneity.* For any $\alpha \geq 0$ we have: $\text{size}(\alpha K) = \alpha \, \text{size}(K)$.

3. *Translation Invariance.* For any $x \in \mathbb{R}^n$ we have: $\operatorname{size}(K + x) = \operatorname{size}(K)$.

It appears that under these natural assumptions, we are able to relate the geometry of the localizer with the target objective function. We prove the following result.

Theorem 4.3.3. *Consider general cutting scheme, starting from some $E_0 \supseteq Q$, and preceding as follows, for $k \geq 0$:*

1. *Choose $x_k \in E_k$*
2. *Access the separation oracle: $g_k = \mathcal{O}(x_k)$*
3. *Find $E_{k+1} \supseteq \{ y \in E_k \ : \ \langle g_k, x_k - y \rangle \geq 0 \}$*

Assume that for some $k \geq 0$ we have a small relative size of E_k, for some $0 < \delta < 1$:

$$\operatorname{size}(E_k) \quad \leq \quad \delta \operatorname{size}(Q). \tag{4.19}$$

Set $\bar{x}_k$ to be the point with the smallest function value among strictly feasible points:

$$\bar{x}_k \quad := \quad \operatorname{argmin}\Big\{ f(y) \ : \ y \in \{x_0, \ldots, x_k\} \ \ s.t. \ \ y \in \operatorname{int} Q \Big\}. \tag{4.20}$$

Then

$$f(\bar{x}_k) - f^\star \quad \leq \quad \delta V. \tag{4.21}$$

Proof. Choose $\delta < \gamma \leq 1$ and denote the contracted set

$$Q_\gamma \quad := \quad \gamma Q + (1 - \gamma) x^\star \quad \subseteq \quad Q.$$

We have

$$\operatorname{size}(Q_\gamma) \quad = \quad \gamma \operatorname{size}(Q) \quad > \quad \delta \operatorname{size}(Q) \quad \overset{(4.19)}{\geq} \quad \operatorname{size}(E_k).$$

Hence, $Q_\gamma \not\subseteq E_k$, and we conclude that there exists a point $y = \gamma z + (1-\gamma) x^\star \in Q_\gamma$ for some $z \in Q$ such that

$$y \quad \notin \quad E_k.$$

By our construction,

$$E_k \quad \supseteq \quad \Big\{ y \in Q \quad : \quad \langle g_0, x_0 - y \rangle \geq 0, \ldots, \langle g_k, x_k - y \rangle \Big\}.$$

Therefore, there exists an index $0 \leq i \leq k$ such that one of the separation conditions is violated:

$$\langle g_i, x_i - y \rangle \quad < \quad 0. \tag{4.22}$$

Note that due to $y \in Q$, it cannot be a separation oracle from Q. Therefore, $x_i \in \int Q$ and $g_i = f'(x_i) \in \partial f(x_i)$. This reasoning, in particular, ensures that among points $\{x_0, \ldots, x_k\}$ there is at least one from $\operatorname{int} Q$ and $\bar{x}_k$ is well defined in (4.20).

Employing convexity of f, we have

$$f(\bar{x}_k) \quad \overset{(4.20)}{\leq} \quad f(x_i) \quad \overset{(4.22)}{<} \quad f(x_i) + \langle f'(x_i), y - x_i \rangle \quad \leq \quad f(y) \quad \leq \quad \gamma f(z) + (1 - \gamma) f^\star.$$

Rearranging the terms, we get:

$$f(\bar{x}_k) - f^\star \quad \leq \quad \gamma (f(z) - f^\star) \quad \leq \quad \gamma V.$$

Taking the limit $\gamma \to \delta$ completes the proof. □

4.3.4 Ellipsoid Method

We are ready to present the ellipsoid method.

As we see from the previous reasoning, all what we want to do is to construct a next set containing a part of the previous one:

$$E_{k+1} \quad \supseteq \quad \Big\{ y \in E_k \ : \ \langle g_k, x_k - y \rangle \geq 0 \Big\}, \tag{4.23}$$

and so that $\text{size}(E_k) \to 0$.

For that purpose we use ellipsoids and size to be the volume function (4.18). To ensure (4.23), we use the following geometric lemma.

Lemma 4.3.4. *Let $E \subseteq \mathbb{R}^n$ to be an ellipsoid given by*

$$E \quad = \quad \Big\{ y \in \mathbb{R}^n \ : \ \langle A^{-1}(y - x), y - x \rangle \leq 1 \Big\},$$

where $x \in \mathbb{R}^n$ is its center and $A = A^\top \succ 0$. Consider an arbitrary cut through the center of E given by vector $g \in \mathbb{R}^n$. Then, for

- $x^+ := x - \frac{1}{(n+1)\langle Ag, g \rangle^{1/2}} Ag$
- $A^+ := \frac{n^2}{n^2-1}\Big(A - \frac{2}{(n+1)\langle Ag, g \rangle} Agg^\top A\Big)$

we have the new ellipsoid $E^+ := \{ y \in \mathbb{R}^n \ : \ \langle (A^+)^{-1}(y - x^+), y - x^+ \rangle \leq 1 \}$ such that

1. *$E^+ \supseteq \{y \in E \ : \ \langle g, x - y \rangle \geq 0\}$ and*
2. *$\text{Vol}(E^+) \leq \exp\Big(-\frac{1}{2n}\Big) \text{Vol}(E)$.*

See, e.g., Section 2.2 in [4], or Section 3.2.8. in [31], for the proof of this lemma.

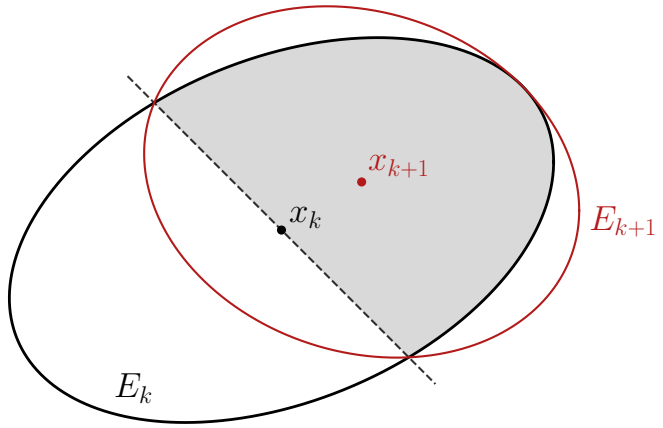

Figure 4.1: One iteration of the ellipsoid method.

Let us write down our method in the algorithmic form.

Algorithm 4.2: *Ellipsoid Method.*

Initialization: $x_0 \in \mathbb{R}^n$ and $R > 0$ such that $B_R(x_0) \supseteq Q$. Fix $K \geq 1$. Set $A_0 := \frac{1}{R} I$.

For $k = 0 \dots K-1$ **iterate:**

1. Access the separation oracle: $g_k = \mathcal{O}(x_k)$
2. Compute new point: $x_{k+1} = x_k - \frac{1}{(n+1)\langle A_k g_k, g_k \rangle^{1/2}} A_k g_k$
3. Update the matrix: $A_{k+1} = \frac{n^2}{n^2-1}\Big(A_k - \frac{2}{(n+1)\langle A_k g_k, g_k \rangle} A_k g_k g_k^\top A_k\Big)$

Return a point $\bar{x}_K := \operatorname{argmin}\{f(y) \ : \ y \in \{x_0, \dots, x_K\} \ \text{s.t.} \ y \in \operatorname{int} Q\}$.

Using our previous reasoning, we can prove the following complexity result for this algorithm.

Theorem 4.3.5. *Let $0 < \varepsilon < V$ be fixed. In order to achieve $f(\bar{x}_K) - f^\star \leq \varepsilon$ it is enough to perform*

$$K \quad = \quad \Big\lfloor 2n^2 \ln \tfrac{RV}{r\varepsilon} \Big\rfloor + 1 \tag{4.24}$$

iterations of Algorithm 4.2 (separation oracle calls).

Proof. From Lemma 4.3.4 we know that

$$\operatorname{Vol}(E_k) \quad \leq \quad \exp\Big(-\tfrac{k}{2n}\Big) \operatorname{Vol}(E_0). \tag{4.25}$$

Therefore, for our size$(\cdot)$ we have that

$$\operatorname{size}(E_k) \quad = \quad \operatorname{Vol}(E_k)^{1/n} \quad \overset{(4.25)}{\leq} \quad \exp\Big(-\tfrac{k}{2n^2}\Big) \operatorname{size}(E_0). \tag{4.26}$$

Let us choose $\delta := \frac{\varepsilon}{V} < 1$. Then, by Theorem 4.3.3, we have $f(\bar{x}_K) - f^\star \leq \varepsilon$ as soon as $\operatorname{size}(E_k) \leq \delta \operatorname{size}(Q)$. According to (4.26), to achieve this goal it is enough to have

$$\exp\Big(-\tfrac{k}{2n^2}\Big) \operatorname{size}(E_0) \quad \leq \quad \delta \operatorname{size}(Q) \quad \Leftrightarrow \quad k \quad \geq \quad 2n^2 \ln \tfrac{\operatorname{size}(E_0)}{\delta \operatorname{size}(Q)}.$$

It remains to use the upper bound: $\frac{\operatorname{size}(E_0)}{\operatorname{size}(Q)} \leq \frac{R}{r}$ to complete the proof. □

Discussion. We see that the oracle complexity of the ellipsoid method is $O(n^2 \ln \frac{RV}{r\varepsilon})$. Each iteration of the method can be implemented in $O(n^2)$ arithmetic operations (for matrix-vector operations) plus additional cost of perfoming the separation oracle. For linear programming, separation oracle can be easily implemented in $O(nm)$ operations (for dense data) which leads to the total complexity of

$$O\Big(n^3(n+m) \ln \tfrac{RV}{r\varepsilon}\Big). \tag{4.27}$$

This bound can be used to show the famous result of polynomial solvability of linear programming, as parameters of the problem (such as V, r, R) comes under logarithm, which leads to the polynomial-like dependence on the size of data input.

The complexity result of the ellipsoid method shows the very important fact that

convex optimization is generally solvable.

However, in practice the ellipsoid method is usually less efficient for linear or semidefinite programming than the methods that take into account the structure of the problem, such as *interior-point methods*, which we study in the last part of the course, unless the dimension n is small ($n \approx 10-20$). At the same time, the dependence on m in (4.27) is linear, so we can use the ellipsoid method for solving low-dimensional problems with huge numbers of constraints.

Compared with cheap gradient methods, we see that despite its excellent logarithmic dependence on the target accuracy, the ellipsoid method depends *explicitly on the dimension* n. Consequently, the method does not work as $n \to \infty$ (even in theory), as the formulas in Algorithm 11.1 prevent the method from performing steps in this limit. A modern modification of the ellipsoid method that remains valid as $n \to \infty$ was developed in [42].

Another, more theoretical version of the cutting plane scheme called the *center of gravity method*. At each iteration, we work with the cutting polytope (4.16) directly and define the next query point as the center of gravity of the set G_k:

$$x_{k+1} \;=\; \frac{1}{\mathrm{Vol}\, G_k} \int\limits_{G_k} y dy. \tag{4.28}$$

The oracle complexity of such method is

$$O(n \ln \tfrac{1}{\varepsilon})$$

separation oracle calls, which is the *optimal* dependence on n (matching the corresponding lower bound). However, this approach is completely unpractical, as each iteration requires computing an n-dimensional integral (4.28).

4.4 Subgradient Method: Normalized Stepsizes

We consider the following convex optimization problem:

$$\min_{x \in Q} f(x), \tag{4.29}$$

for a convex set $Q \subseteq \mathbb{R}^n$, and a convex (possibly non-differentiable) function $f : \mathbb{R}^n \to \mathbb{R}$, which, for simplicity of the presentation, we define over the whole space. In modern applications, the dimension n in problem (4.29) is large: $n \to \infty$ (*large-scale optimization*), so we cannot directly apply the heavy machinery of the ellipsoid method. Instead, we will analyze a non-smooth analog of the gradient method.

We assume that for any point $x \in Q$ we can compute a subgradient vector $f'(x) \in \partial f(x)$ that satisfies:

$$f(y) \;\geq\; f(x) + \langle f'(x), y - x \rangle, \qquad x, y \in Q.$$

For the set Q we assume a possibility of computing the projection, in the Euclidean norm:

$$\pi_Q(x) \;:=\; \operatorname*{argmin}_{y \in Q} \|y - x\|. \tag{4.30}$$

This is a different and more expensive operation than a separation oracle for Q. The possibility of computing projections (4.30) typically means that the set Q is *simple*. Thus, as we did when discussing the fully composite problems in Section 3.7, we assume that the main difficulty of solving problem (4.29) lies in the objective function f, but not in the constraints. However, it is possible to generalize the subgradient method to the case of using only a separation oracle for Q, when the set is specified by a number of black-box functional inequalities.

4.4.1 Optimality Condition for Constrained Minimization

Let us review an optimality condition for a point $x^\star$ to be a global minimum of (4.29). We have the following generalization of the first-order condition from unconstrained minimization. Let $Df(x^\star)[h]$ denote the directional derivative of f along the direction h:

$$Df(x)[h] \;:=\; \lim_{\alpha \to 0+} \tfrac{1}{\alpha}\big[f(x+\alpha h) - f(x)\big].$$

- For differentiable functions: $Df(x)[h] \equiv \langle \nabla f(x), h \rangle$.
- For convex functions, the directional derivative along any direction exists at all interior points of the domain. In this case, we have the following interesting relationship:

$$Df(x)[h] \;=\; \max\big\{\langle g, h\rangle \;:\; g \in \partial f(x)\big\}.$$

Proposition 4.4.1. *Let $x^\star$ be a constrained minimum of f over set Q. Then*

$$Df(x^\star)[x - x^\star] \;\geq\; 0, \qquad x \in Q. \tag{4.31}$$

Proof. Indeed, by the definition of the directional derivative, we have for a sufficiently small $\alpha > 0$:

$$Df(x^\star)[x - x^\star] \;=\; \tfrac{1}{\alpha}\big[f(x+\alpha h) - f(x)\big] + o(1) \;\geq\; o(1),$$

where $o(1)$ goes to zero when $\alpha \to 0$. Taking the limit complete the proof. □

Corollary 4.4.2. *For a constrained minimum of a differentiable function f, we have:*

$$\langle \nabla f(x^\star), x - x^\star \rangle \;\geq\; 0, \qquad x \in Q. \tag{4.32}$$

Remark 4.4.3. See also Theorem 4.1.3 from Section 4.1.3 for a more general optimality condition suitable for additive composite optimization, that generalizes (4.32).

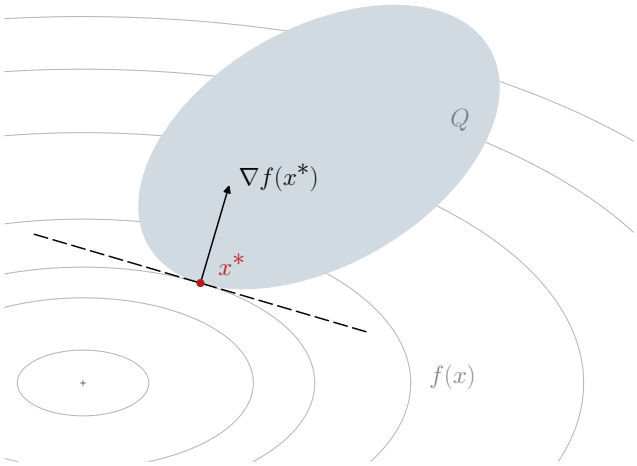

Figure 4.2: The constrained minimum of a function f over a set Q.

Subgradient step. We will use the optimality condition to analyze one step of the subgradient method. The method is very similar to the basic gradient method. At point $x \in Q$, we compute a subgradient $f'(x) \in \partial f(x)$ and perform the following update, for some $\eta > 0$:

$$x^+ \;=\; \operatorname*{argmin}_{y \in Q} \Big[\, m(y) \;:=\; \eta \langle f'(x), y - x \rangle + \tfrac{1}{2}\|y - x\|^2 \,\Big]. \tag{4.33}$$

Note that the function $m(\cdot)$ is differentiable and 1-strongly convex. Hence, we have

$$\begin{aligned} m(y) \;&\geq\; m(x^+) + \langle \nabla m(x^+), y - x^+ \rangle + \tfrac{1}{2}\|y - x^+\|^2 \\ &\overset{(4.32)}{\geq}\; m(x^+) + \tfrac{1}{2}\|y - x^+\|^2. \end{aligned}$$

Expanding the definition of the model gives:

$$\begin{aligned} \tfrac{1}{2}\|y - x\|^2 + \eta \langle f'(x), y - x \rangle \;&\geq\; \tfrac{1}{2}\|y - x^+\|^2 + \Big[\eta \langle f'(x), x^+ - x \rangle + \tfrac{1}{2}\|x^+ - x\|^2\Big] \\ &\geq\; \tfrac{1}{2}\|y - x^+\|^2 - \tfrac{\eta^2}{2}\|f'(x)\|^2. \end{aligned}$$

Thus, we have proved the following important lemma, which is in the core of analysis of all subgradient methods.

Lemma 4.4.4. *For any $y \in Q$, it holds*

$$\tfrac{1}{2}\|y - x\|^2 + \tfrac{\eta^2}{2}\|f'(x)\|^2 \;\geq\; \tfrac{1}{2}\|y - x^+\|^2 + \eta \langle f'(x), x - y \rangle. \tag{4.34}$$

Consequently, substituting $y := x^\star \in Q$ (any minimizer), we get:

$$\tfrac{1}{2}\|x^\star - x\|^2 + \tfrac{\eta^2}{2}\|f'(x)\|^2 \;\geq\; \tfrac{1}{2}\|x^\star - x^+\|^2 + \eta \langle f'(x), x - x^\star \rangle. \tag{4.35}$$

Note that for convex functions, the inner product in the right hand side of (4.35) is non-negative, and it is bounded by the functional residual:

$$\langle f'(x), x - x^\star \rangle \;\geq\; f(x) - f^\star \;\geq\; 0.$$

This reasoning can immediately lead to a convergence rate for the method. But first, let us choose stepsizes in a smart way.

4.4.2 Subgradient Method

In non-smooth optimization, the main information that is provided by a subgradient $f'(x) \in \partial f(x)$ is of *geometric nature*, as it gives us a certain *separation of the sublevel set* of f at x. At the same time, in contrast to smooth optimization, the magnitude $\|f'(x)\|$ does not reveal much information.

Example 4.4.5. Consider $f(x) = |x|$, $x \in \mathbb{R}$. Then, for any $x \neq 0$, we have $|f'(x)| = 1$, no matter how close we are to the optimum $x^\star = 0$.

Therefore, it is natural to *normalize* the subgradient direction: $\frac{f'(x)}{\|f'(x)\|}$, which happens also to equip our method with *universal* (problem-class independent) convergence rates. We consider the following algorithm.

Algorithm 4.3: *Subgradient Method.*

Initialization: $x_0 \in Q$. Fix $K \geq 1$ and positive parameters $\{\gamma_k\}_{k \geq 0}$.

For $k = 0 \dots K-1$ **iterate:**

1. Compute a subgradient: $f'(x_k) \in \partial f(x_k)$
2. Perform the normalized subgradient step:

$$x_{k+1} \;=\; \operatorname*{argmin}_{y \in Q} \Big[\tfrac{\gamma_k}{\|f'(x_k)\|} \langle f'(x_k), y - x_k \rangle \;+ \tfrac{1}{2}\|y - x_k\|^2 \;\Big] \;=\; \pi_Q\Big(x_k - \tfrac{\gamma_k}{\|f'(x_k)\|} f'(x_k)\Big)$$

Return a point $\bar{x}_K := \operatorname{argmin}\{f(y) \;:\; y \in \{x_0, \dots, x_K\}\}$ or the average $\frac{1}{K} \sum\limits_{i=0}^{K-1} x_i$.

Substituting our stepsize choice in the previous lemma, we obtain, for every $k \geq 0$:

$$\tfrac{\gamma_k^2}{2} + \tfrac{1}{2}\|x_k - x^\star\|^2 \;\;\geq\;\; \tfrac{1}{2}\|x_{k+1} - x^\star\|^2 + \gamma_k \Delta_k, \tag{4.36}$$

where

$$\Delta_k \;\;:=\;\; \tfrac{\langle f'(x_k), x_k - x^\star \rangle}{\|f'(x_k)\|}$$

is a certain *measure of optimality*. We also denote:

$$\bar{\Delta}_K \;\;:=\;\; \tfrac{1}{K} \sum_{i=0}^{K-1} \Delta_i \qquad \text{and} \qquad \Delta_K^\star \;\;:=\;\; \min_{0 \leq i \leq K-1} \Delta_i.$$

Clearly, we have $\bar{\Delta}_K \geq \Delta_K^\star$.

Telescoping inequality (4.36) for the first $K \geq 1$ iterations of the method, we get the following progress:

$$\tfrac{1}{2}\|x_0 - x^\star\|^2 + \tfrac{1}{2} \sum_{i=0}^{K-1} \gamma_i^2 \;\;\geq\;\; \sum_{i=0}^{K-1} \gamma_i \Delta_i \;\;\geq\;\; \Big(\sum_{i=0}^{K-1} \gamma_i \Big) \Delta_K^\star \tag{4.37}$$

Therefore, we obtain the following bound on our new accuracy measure:

$$\Delta_K^\star \;\;\leq\;\; \frac{\|x_0 - x^\star\|^2 + \sum\limits_{i=0}^{K-1} \gamma_i^2}{2 \sum\limits_{i=0}^{K-1} \gamma_i} \;\;=\;\; \varphi(\gamma_0, \dots, \gamma_{K-1}). \tag{4.38}$$

In principle, we want to choose $\{\gamma_k\}_{k \geq 0}$ such that the right hand side of (4.38) is as small as possible, i.e. to minimize it: $\varphi(\cdot) \to \min$. Notice that

- $\varphi(\cdot)$ is convex;
- $\varphi(\cdot)$ is symmetric in γ: its value is invariant to any permutation of $\{\gamma_0, \dots, \gamma_{K-1}\}$.

For a convex symmetric function, there is always exists a solution with all the same arguments,

$$\gamma_0^\star \;\;=\;\; \gamma_1^\star \;\;=\;\; \dots \;\; \gamma_{k-1}^\star \;\;=\;\; \gamma.$$

Therefore, a constant choice $\gamma > 0$ is able to give us the best bound in (4.38), when the number of iterations K is fixed. In practice, however, we might want to use a decreasing sequence, e.g. $\gamma_k = O(1/\sqrt{k})$, so not to fix K.

We denote by $R \geq \|x_0 - x^\star\|$ *any* upper bound for the distance from the initial point to any of the solutions. Using this bound in (4.37) along with the constant choice, $\gamma_i \equiv \gamma > 0$, leads to

$$\bar{\Delta}_K \overset{(4.37)}{\leq} \tfrac{\|x_0 - x^\star\|^2}{2K\gamma} + \tfrac{K\gamma}{2} \leq \tfrac{R^2}{2K} + \tfrac{K\gamma}{2}.$$

Minimizing the right-hand side in $\gamma > 0$, we obtain the optimal choice

$$\boxed{\gamma := \tfrac{R}{K^{1/2}}.} \tag{4.39}$$

Thus, we have proved the following result.

Theorem 4.4.6. *Let all γ_k in Algorithm 4.3 be chosen according to* (4.39)*. Then,*

$$\Delta_K^\star \leq \bar{\Delta}_K \leq \tfrac{R}{K^{1/2}}. \tag{4.40}$$

Lipschitz functions. The question is how to relate our quantities Δ_k with a standard accuracy measure, the functional residual $f(x_k) - f^\star$. The simplest reasoning is as follows.

Assume that our subgradients are bounded:

$$\|f'(x)\| \leq M, \qquad \forall x \in Q, \tag{4.41}$$

which means that the function f is Lipschitz:

$$|f(y) - f(x)| \leq M\|y - x\|, \qquad x, y \in Q.$$

Then, by convexity we immediately obtain the following relationship:

$$\Delta_k := \tfrac{\langle f'(x_k), x_k - x^\star \rangle}{\|f'(x_k)\|} \overset{(4.41)}{\geq} \tfrac{\langle f'(x_k), x_k - x^\star \rangle}{M} \geq \tfrac{f(x_k) - f^\star}{M}. \tag{4.42}$$

Corollary 4.4.7. *We have,*

$$f\Big(\tfrac{1}{K}\sum_{i=0}^{K-1} x_i\Big) - f^\star \leq \tfrac{1}{K}\sum_{i=0}^{K-1}[f(x_i) - f^\star] \overset{(4.42)}{\leq} M \cdot \bar{\Delta}_K \leq \tfrac{MR}{K^{1/2}}. \tag{4.43}$$

Note that we used a conservative choice of stepsizes that requires fixing the number of iterations. Therefore, technically, (4.43) is not a "rate" of convergence, as this bound is achieved only once, for the final output of the algorithm. If we wanted to achieve higher precision, we would need to rerun the method with a smaller stepsize γ. In Section 4.6, we discuss a more advanced approach using *adaptive stepsizes* that solves this issue, as they do not require fixing the number of iterations in advance, and they also work for stochastic problems.

We also assume we know a bound $R \geq \|x_0 - x^\star\|$. At the same time, it is easy to see that any choice of $R > 0$ in (4.39) will give us a similar bound. For example, in practice, we can set $R := 1$ if no other information is available. However, the closer R is to $\|x_0 - x^\star\|$, the faster the method converges.

In the next lecture, we prove that the rate $O\Big(\tfrac{MR}{K^{1/2}}\Big)$ is *optimal* for this problem class.

4.4.3 Analysis via Functional Growth

Note that to prove our bound (4.43) for the subgradient method on convex Lipschitz functions, it is enough to take much more conservative steps, using the same constant for each iteration in (4.33):

$$\eta \;:=\; \tfrac{\gamma}{M} \overset{(4.39)}{=} \tfrac{R}{MK^{1/2}}, \tag{4.44}$$

in contrast to using normalized stepsizes.

Exercise 4.4.1. Check that subgradient steps (4.33) with η given by (4.44) achieve the same bound (4.43) for the functional residual.

In this section, we provide a more advanced analysis that reveals the true power of normalized stepsizes: they enable us to prove convergence rates for problems well beyond the Lipschitz assumption.

Geometric interpretation. First, let us understand the geometric meaning of the quantity $\Delta_k := \frac{\langle f'(x_k), x_k - x^\star\rangle}{\|f'(x_k)\|}$. For simplicity, consider the case of unconstrained optimization, thus

$$Q \;\equiv\; \mathbb{R}^n.$$

Staying at point x_k, the subgradient $f'(x_k) \neq 0$ provides us with the supporting hyperplane:

$$L_k \;=\; \Big\{\; y \in \mathbb{R}^n \;:\; \langle f'(x_k), x_k - y\rangle = 0 \;\Big\}.$$

Let us look at the optimal solution $x^\star \in \mathbb{R}^n$ and find the projection of it onto L_k:

$$\min_{y \in L_k} \|y - x^\star\|. \tag{4.45}$$

We take vector $h := \Delta_k \frac{f'(x_k)}{\|f'(x_k)\|}$ and the perturbed solution $y^\star := x^\star + h$. We have

$$\begin{aligned}\langle f'(x_k), x_k - y^\star\rangle &= \langle f'(x_k), x_k - x^\star\rangle + \langle f'(x_k), h\rangle \\ &= \Delta_k \|f'(x_k)\| - \Delta_k \tfrac{\langle f'(x_k), f'(x_k)\rangle}{\|f'(x_k)\|} \;=\; 0,\end{aligned}$$

which concludes that $y^\star \in L_k$ is the solution to (4.45). Note that

$$\|y^\star - x^\star\| \;=\; \|h\| \;=\; \Delta_k.$$

Corollary 4.4.8. *Δ_k is the distance from $x^\star$ to the hyperplane L_k.*

Consider the localizing polyhedron:

$$G_{k+1} \;=\; \Big\{ y \in \mathbb{R}^n \;:\; \langle f'(x_0), x_0 - y\rangle \geq 0, \ldots, \langle f'(x_k), x_k - y\rangle \geq 0 \Big\}.$$

By convexity $x^\star \in G_{k+1}$. Note that $\Delta_k^\star$ is the minimal distance from $x^\star$ to one of the hyperplanes defining the polyhedron. Hence, $\Delta_k^\star$ is *the maximal radius of the Euclidan ball that is contained in the localizer G_{k+1}*:

$$\Delta_k^\star \;=\; \max\Big\{ r \geq 0 \;:\; B_r(x^\star) \subset G_{k+1} \Big\}$$

and the subgradient method manages to $\Delta_k^\star \to 0$. Note that such quantity can be used to define an approprite size$(\cdot)$ of a set (see previous lecture).

Convergence rate with functional growth. Define the following quantity, called *the growth of f at point $x^\star$*:

$$\omega_f(r) \;:=\; \max\Big\{f(x) - f(x^\star) \;:\; \|x - x^\star\| \le r\Big\}.$$

Clearly, $\omega_f(\cdot)$ is a nondecreasing function of r.

We can relate our measure of optimality Δ_k with the functional residual. By convexity, we know that

$$\Delta_k \;=\; \tfrac{\langle f'(x_k), x_k - x^\star\rangle}{\|f'(x_k)\|} \;\ge\; \tfrac{f(x_k) - f(x^\star)}{\|f'(x_k)\|}.$$

Now we present a more advanced reasoning. Consider the perturbation, as we fixed before:

$$h \;:=\; \tfrac{\Delta_k}{\|f'(x_k)\|} f'(x_k).$$

Then,

$$\begin{aligned}
\langle f'(x_k), x_k - x^\star\rangle &= \langle f'(x_k), x_k - x^\star - h\rangle + \langle f'(x_k), h\rangle \\
&\ge f(x_k) - f(x^\star + h) + \|f'(x_k)\|\Delta_k \\
&= f(x_k) - f^\star + (f^\star - f(x^\star + h)) + \|f'(x_k)\|\Delta_k.
\end{aligned}$$

Note that $\langle f'(x_k), x_k - x^\star\rangle = \|f'(x_k)\|\Delta_k$. Rearranging the terms, we obtain

$$f(x_k) - f^\star \;\le\; f(x^\star + h) - f^\star.$$

This inequality has a clear geometric meaning: due to convexity, the function value at $f(x_k)$ is better than that at $f(x^\star + h)$, as shown on Fig. 4.3. We established the following result.

Theorem 4.4.9. *For the result of the subgradient method, it holds:*

$$f(\bar{x}_K) - f^\star \;\le\; \omega_f(\Delta^\star_K), \qquad \text{and} \qquad \Delta^\star_K \;\le\; \tfrac{R}{K^{1/2}}. \tag{4.46}$$

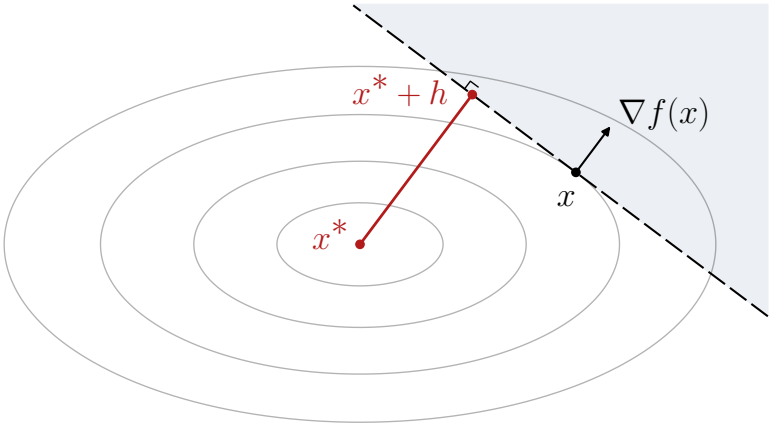

Figure 4.3: Point $x^\star + h$ is the projection of the unconstrained minimum $x^\star$ to the supporting hyperplane defined by a subgradient at x.

Example: Lipschitz functions.

Proposition 4.4.10. *Consider Lipschitz functions: $f(x+h) - f(x) \leq M\|h\|$. Then, we have*

$$\omega_f(r) \quad \leq \quad Mr. \tag{4.47}$$

Hence, we immediately recover the bound from the previous reasoning:

$$f(\bar{x}_K) - f^\star \quad \overset{(4.46)}{\leq} \quad \omega_f\Big(\tfrac{R}{K^{1/2}}\Big) \quad \overset{(4.47)}{\leq} \quad \tfrac{MR}{K^{1/2}}. \tag{4.48}$$

To obtain an ε-solution we need $\Big[\frac{MR}{\varepsilon}\Big]^2$ oracle calls.

Let us consider the following objective:

$$f(x) \quad = \quad \max_{1 \leq i \leq m} \Big[\langle a_i, x\rangle - b_i\Big].$$

Then

$$\begin{aligned} f(y) - f(x) \quad &= \quad \max_{1 \leq i \leq m} \Big[\langle a_i, x\rangle - b_i\Big] - \max_{1 \leq j \leq m} \Big[\langle a_j, y\rangle - b_j\Big] \\ &= \quad \max_{1 \leq i \leq m} \Big[\langle a_i, x\rangle - b_i - \max_{1 \leq j \leq m} \Big[\langle a_j, y\rangle - b_j\Big]\Big] \\ &\leq \quad \max_{1 \leq i \leq m} \Big[\langle a_i, x - y\rangle\Big] \quad \leq \quad \max_{1 \leq i \leq m} \|a_i\| \cdot \|x - y\|. \end{aligned}$$

Hence, $M = \max\limits_{1 \leq i \leq m} \|a_i\|$.

At every point x, a subgradient can be computed as $f'(x) := a_i$, where i it the active index.

Example: smooth functions.

Proposition 4.4.11. *Let f be differentiable and smooth. Then,*

$$\begin{aligned} \omega_f(r) \quad &= \quad \max\{f(x^\star + h) - f(x^\star) \;:\; \|h\| \leq r\} \\ &\leq \quad \max\{\langle \nabla f(x^\star), h\rangle + \tfrac{L}{2}\|h\|^2 \;:\; \|h\| \leq r\} \\ &\leq \quad \|\nabla f(x^\star)\| r + \tfrac{L}{2} r^2. \end{aligned} \tag{4.49}$$

Substituting this value into our bound, we get:

$$f(\bar{x}_K) - f^\star \quad \overset{(4.46)}{\leq} \quad \omega_f\Big(\tfrac{R}{K^{1/2}}\Big) \quad \overset{(4.49)}{\leq} \quad \tfrac{\|\nabla f(x^\star)\| R}{K^{1/2}} + \tfrac{Lr^2}{2K}.$$

Hence, in case of smooth functions and small $\|\nabla f(x^\star)\|$, our method automatically receives a faster rate of convergence than that one from (4.48). For unconstrained minimization, $\nabla f(x^\star) = 0$.

Example: maximum of smooth functions. Let

$$f(x) \;=\; \max_{1\le i\le m}\Big[f_i(x)\Big], \tag{4.50}$$

where each $f_i(x) = \frac{1}{2}\langle A_i x, x\rangle - \langle b_i, x\rangle$ is a convex quadratic function, or, more generally, each f_i is convex and has the Lipschitz gradient with constant L_i.

Then,

$$\begin{aligned} f_i(x) &\le f_i(x^\star) + \langle \nabla f_i(x^\star), x - x^\star\rangle + \tfrac{L_i}{2}\|x - x^\star\|^2 \\ &\le f_i(x^\star) + \|\nabla f_i(x^\star)\| \cdot \|x - x^\star\| + \tfrac{L_i}{2}\|x - x^\star\|^2. \end{aligned}$$

And we obtain that

$$\omega_f(r) \;\le\; \max_{1\le i\le m}\|\nabla f_i(x^\star)\|\cdot r + \max_{1\le i\le m} L_i \cdot \tfrac{r^2}{2}. \tag{4.51}$$

Corollary 4.4.12. *It holds:*

$$f(\bar{x}_K) - f^\star \overset{(4.46)}{\le} \omega_f\Big(\tfrac{R}{K^{1/2}}\Big) \overset{(4.51)}{\le} \tfrac{\max_{1\le i\le m}\|\nabla f_i(x^\star)\|R}{K^{1/2}} + \tfrac{\max_{1\le i\le m} L_i R^2}{2K}.$$

We see that exactly the same subgradient method with normalized steps will work on different subclasses of non-smooth convex problems, even though the objective in (4.50) is not Lipschitz.

4.5 Lower Bound for Non-Smooth Convex Optimization

We proved the following result for the subgradient method minimizing convex Lipschitz functions:

$$f(\bar{x}_K) - f^\star \;\le\; \tfrac{MR}{\sqrt{K}}. \tag{4.52}$$

In the first part of the course, we saw that in the smooth convex case, it is possible to have better rates (for the gradient method and for the accelerated fast gradient method). However, in non-smooth convex optimization, it appears that the rate (4.52) is optimal.

Problem class. To prove the lower bound, we restrict ourselves onto the following class of problems, which is obviously a particular case of our situation.

$$f^\star \;=\; \min_{x\in\mathbb{R}^n}\Big\{\, f(x) \;:\; \|x\| \le R \,\Big\}, \tag{4.53}$$

where f is convex and Lipschitz continuous:

$$f(y) - f(x) \;\le\; M\|y - x\|, \qquad \forall x, y.$$

The norm is the standard Euclidean, and $M > 0$ and $R > 0$ our two key complexity parameters.

We consider the class of *all first-order optimization methods*, starting from an arbitrary initialization x_1[1]. We associate an optimization method with a sequence of mappings:

$$\mathcal{A} \;=\; (A_1, A_2, \ldots)$$

that define the iteration process:

$$x_{k+1} \;=\; A_k(\mathcal{O}_f(x_1), \ldots, \mathcal{O}_f(x_k)), \quad k \ge 1,$$

where $\mathcal{O}_f(x) := (f(x), f'(x))$ for some arbitrary subgradient $f'(x) \in \partial f(x)$. Therefore, each mapping A_k, $k \ge 1$ takes all first-order oracle information available up to the current moment and return the next iterate. Without loss of generality, we assume that the result of the algorithm is the last generated point: x_K, where $K \ge 1$ is a fixed number of iterations.

[1] We start iterations from $k \ge 1$ in this lecture to keep our notation simpler.

4.5.1 Lower Bound

We prove the following theorem.

Theorem 4.5.1. *Let $M > 0$ and $R > 0$ be fixed. Then, for any first-order algorithm running for $K \geq 1$ iterations, there exists a function $f : \mathbb{R}^n \to \mathbb{R}$ with $n \geq K$ such that*

1. *f is convex and Lipschitz with constant M;*
2. *For the output of the algorithm on this function, it holds:*

$$f(x_K) - f^\star \quad \geq \quad \frac{MR}{2\sqrt{K}}.$$

Proof. Let $\delta > 0$ be a fixed parameter (it can be arbitrarily small).

We use a *resisting oracle* that will choose a set of numbers (orientations):

$$\xi_1, \ldots, \xi_K \quad \in \quad \{-1, 1\},$$

and a permutation of coordinates:

$$t \quad \mapsto \quad \sigma(t) \quad \in \quad \{1, 2, \ldots, n\}.$$

These parameters will be built along the oracle calls from the given algorithm, which is an arbitrary first-order method. We also denote by $e_i \in \mathbb{R}^n$, $1 \leq i \leq n$ the standard basis vectors.

We consider the following family of convex function, $x \in \mathbb{R}^n$:

$$f_k(x) \quad := \quad M \cdot \max_{1 \leq i \leq k} \Big[\, \xi_i \langle e_{\sigma(i)}, x \rangle - (i-1)\delta \, \Big]. \tag{4.54}$$

Note that

$$\langle e_{\sigma(i)}, x \rangle \quad = \quad x^{(\sigma(i))} \quad \in \quad \mathbb{R}$$

is the ith coordinate of a vector after permuting the coordinates.

It is clear that every $f_k(\cdot)$ is Lipschitz continuous with constant M.

1. Let us analyze its minimum. We have

$$\begin{aligned} f_k^\star \quad := \quad \min_{\|x\| \leq R} f_k(x) \quad &= \quad M \cdot \min_{\|x\| \leq R} \max_{1 \leq i \leq k} \Big[\, \xi_i \langle e_{\sigma(i)}, x \rangle - (i-1)\delta \, \Big] \\ &\leq \quad M \cdot \min_{\|x\| \leq R} \max_{1 \leq i \leq k} \xi_t \langle e_{\sigma(i)}, x \rangle \quad = \quad M \cdot \min_{\|x\| \leq R} \max_{1 \leq i \leq k} \langle e_{\sigma(i)}, x \rangle \quad = \quad -M\gamma, \end{aligned}$$

where we noticed that, due to symmetry in the problem, the minimum is achieved when all first k coordinates after the permutation are the same:

$$x^{(\sigma(1))} \quad = \quad x^{(\sigma(2))} \quad = \quad \ldots \quad = \quad x^{(\sigma(k))} \quad = \quad -\gamma,$$

for some $\gamma > 0$, while other coordinates are zero. The smallest value is achieved at the boundary of the ball, $\|x\|^2 = k\gamma^2 = R^2$, and we find that $\gamma := \frac{R}{\sqrt{k}}$. Hence,

$$f_k^\star \quad \leq \quad -\frac{MR}{\sqrt{k}}. \tag{4.55}$$

2. Now, we present a *resisting strategy* for choosing ξ_k and $\sigma(k)$. We pick them *adversarially* by following the following rules.

- At first step, the algorithm asks the oracle information at the initial point x_1. Let us pick

$$\sigma(1) \quad \in \quad \arg\max_{1 \le i \le n} |\langle e_i, x_1 \rangle|$$

  In other words, $\sigma(1)$ is an index of a maximal entry (in absolute value) among coordinates of x_1. Then, we specify $\xi_1 \in \{-1, 1\}$ in a way that

$$\xi_1 \langle e_{\sigma(1)}, x_1 \rangle \quad = \quad |\langle e_{\sigma(1)}, x_1 \rangle|,$$

  hence $\xi_1 = \mathrm{sign}(\langle e_{\sigma(1)}, x_1 \rangle) = \mathrm{sign}(x_1^{(\sigma(1))})$. So

$$f_1(x) \quad = \quad M \cdot \xi_1 \cdot \langle e_{\sigma(1)}, x \rangle$$

  is fully defined.

- For $1 \le k \le K-1$, assume that we have built $f_k(x)$. Let x_{k+1} be the point of the trajectory of $\mathcal{A}$ at iteration k applied to $f_k(x)$:

$$x_{k+1} \quad = \quad \mathcal{A}(\mathcal{O}_{f_k}(x_1), \ldots, \mathcal{O}_{f_k}(x_k)).$$

  Note that it can be arbitrary as we cannot control what the method returns.

  Let us choose as $\sigma(k+1)$ the index of a maximal element of x_{k+1} (in absolute value), except $\sigma(1), \ldots, \sigma(k)$. Thus,

$$\sigma(k+1) \quad \in \quad \operatorname*{argmax}_{1 \le i \le n \text{ s.t. } i \notin \{\sigma(1), \ldots, \sigma(k)\}} |\langle e_i, x_{k+1} \rangle|,$$

  and specify $\xi_{k+1} \in \{-1, 1\}$ such that

$$\xi_{k+1} \langle e_{\sigma(k+1)}, x_{k+1} \rangle \quad = \quad |\langle e_{\sigma(k+1)}, x_{k+1} \rangle|,$$

  i.e. $\xi_{k+1} = \mathrm{sign}(\langle e_{\sigma(k+1)}, x_{k+1} \rangle) = \mathrm{sign}(x_{k+1}^{(\sigma(k+1))})$. Thus we obtained the next $f_{k+1}(x)$.

3. We need to verify the following important fact, which is the outcome of our resisting strategy: for any $s < k$, functions $f_s(\cdot)$ and $f_k(\cdot)$ are *informationally indistinguishable* for any local oracle at x_s:

$$f_s(x) \quad \equiv \quad f_k(x), \qquad \forall x \quad \text{s.t.} \quad \|x - x_s\| \le \delta. \tag{4.56}$$

This implies that the first-order oracle information at x_s is identical for both functions, as the subdifferential is fully determined by function values in a neighbourhood of that point (for example, via the directional derivatives). For our method, it means that all oracle information received in the past remains consistent for subsequent functions in the "future":

$$\boxed{\mathcal{O}_{f_s}(x_s) \quad = \quad \mathcal{O}_{f_{s+1}}(x_s) \quad = \quad \mathcal{O}_{f_{s+2}}(x_s) \quad = \quad \ldots \quad = \quad \mathcal{O}_{f_k}(x_s)}$$

so running the algorithm for s iterations on f_s is the same as performing these iterations on f_k.

To prove (4.56), we note that

$$f_k(x) \quad = \quad \max\Big\{ f_s(x), \;\; M \cdot \max_{s < i \le k} \big[\, \xi_i \langle e_{\sigma(i)}, x \rangle - (i-1)\delta \, \big] \Big\}. \tag{4.57}$$

By the definition of $f_s(\cdot)$, due to our choice of $\sigma(s)$ and ξ_s, we have

$$\xi_s \langle e_{\sigma(s)}, x_s \rangle \;\; \geq \;\; \xi_i \langle e_{\sigma(i)}, x_s \rangle, \qquad \forall i > s.$$

Hence,

$$f_s(x_s) \;\; \geq \;\; M \cdot [\; \xi_i \langle e_{\sigma(i)}, x_s \rangle - (i-1)\delta \;] \;\; + \;\; M \cdot \delta, \qquad \forall i > s. \tag{4.58}$$

Due to the Lipschitz continuity, from (4.58), it holds for all x such that $\|x - x_s\| \leq \delta$ that

$$f_s(x) \;\; \geq \;\; M \cdot [\; \xi_t \langle e_{\sigma(t)}, x \rangle - (t-1)\delta \;], \qquad \forall i > s. \tag{4.59}$$

Applying this bound to (4.57) we conclude that (4.56) is true.

4. For the final function, we take $\boxed{f(x) := f_K(x)}$. Then, for the output x_K of the algorithm as applied to f, we have

$$\begin{aligned} f(x_K) \;\; &= \;\; f_K(x_K) \;\; \geq \;\; M \cdot [\, \xi_K \langle e_{\sigma(K)}, x_K \rangle - (K-1)\delta \,] \\ &= \;\; M \cdot [\, |x_K^{(\sigma(K))}| - (K-1)\delta] \;\; \geq \;\; -(K-1)\delta. \end{aligned} \tag{4.60}$$

Note that since $\delta > 0$ can be arbitrarily small, we can have $f(x_K) \approx 0$. At the same time, bound (4.55) shows that the exact minimum is strictly below zero. We finally obtain the following bound for the functional residual:

$$f(x_K) - f^\star \;\; \overset{(4.60),(4.55)}{\geq} \;\; \tfrac{MR}{\sqrt{K}} - \delta(K-1) \;\; \geq \;\; \tfrac{MR}{2\sqrt{K}},$$

for a sufficiently small $\delta \leq \frac{MR}{2(K-1)\sqrt{K}}$, which completes the proof. □

Remark 4.5.2. Note that the functions $f_k(\cdot)$ from the construction of the lower bound are very simple. Consider for simplicity $M = 1$ in (4.54). Then,

- For $k = 1$, we have

$$f_1(x) \;\; = \;\; \Big[\; \xi_1 \langle e_{\sigma(1)}, x \rangle \;\Big] \;\; = \;\; \pm x^{(\sigma(1))}.$$

- For $k = 2$, we have

$$\begin{aligned} f_2(x) \;\; &= \;\; \max\Big[\; \xi_1 \langle e_{\sigma(1)}, x \rangle,\; \xi_2 \langle e_{\sigma(2)}, x \rangle - \delta \;\Big] \\ &= \;\; \max\Big[\; \pm x^{(\sigma(1))},\; \pm x^{(\sigma(2))} - \delta \;\Big]. \end{aligned}$$

The signs and permutations are chosen in an adversarial way.

Let us visualize these functions. As $\delta > 0$ can be arbitrarily small, we can set it to zero for our visualization. Up to the permutation of coordinates, the first function is either

$$f_1(x) \;\; = \;\; x^{(1)} \qquad \text{or} \qquad f_1(x) \;\; = \;\; -x^{(1)},$$

where $x \in \mathbb{R}^n$ lives in high-dimensional space. Their graphs for $n = 2$ are shown in Fig. 4.4.

At the second iterate, up to the permutation of coordinates, we have four possible functions (see Fig. 4.5), depending on the selection of signs:

$$\begin{aligned} f_2(x) \;\; &= \;\; \max[x^{(1)}, x^{(2)}], & f_2(x) \;\; &= \;\; \max[-x^{(1)}, x^{(2)}] \\ f_2(x) \;\; &= \;\; \max[-x^{(1)}, -x^{(2)}], & f_2(x) \;\; &= \;\; \max[x^{(1)}, -x^{(2)}], \end{aligned}$$

and so on. Each time, we manage to place the minimum $x^\star$ at a significant distance from the query point.

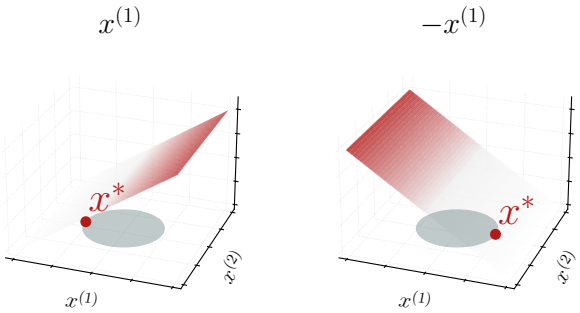

Figure 4.4: Two possible functions, $f_1(x) = x^{(1)}$ and $f_1(x) = -x^{(1)}$, to be selected at the first oracle call. The resisting strategy picks the function with the largest value at the requested point.

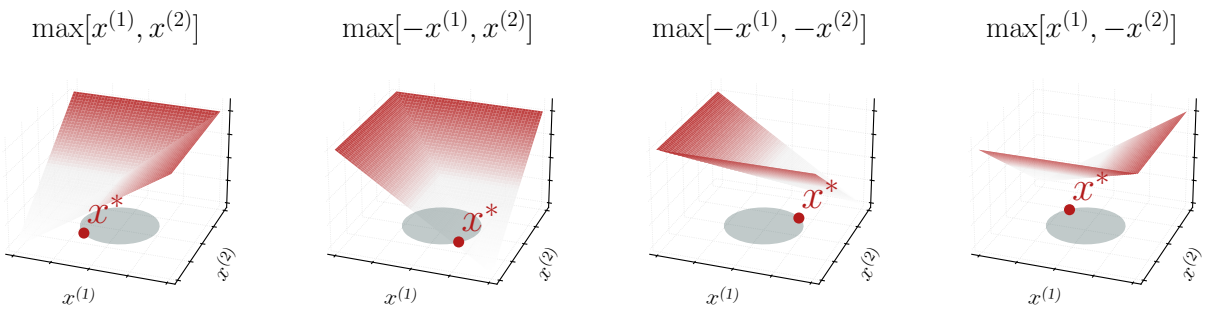

Figure 4.5: The four candidate functions after the resisting strategy at the second oracle call.

Remark 4.5.3. It is possible to set $\delta := 0$ in the worst-case function construction. However, in this case, the resisting oracle must be allowed to decide *which subgradient to return* when the query point is at a point where multiple components of the function are active.

By selecting $\delta > 0$, we ensure the same worst-case behavior even if the algorithm has access to the *entire subdifferential* $\partial f(x)$ at each query point.

4.5.2 Overview of the Non-Smooth Convex Optimization

Let us conclude by discussing the complexity landscape of black-box convex optimization. Depending on the problem dimension $n \geq 1$, there are different regimes with its own "optimal methods":

- $n = 1$: **univariate minimization** — *binary search*. While more practically efficient methods for univariate minimization exist, binary search remains optimal in terms of its oracle complexity and its practical simplicity.
- $n \geq 1$: **small dimension** — *cutting plane schemes*, which are generalizations of the binary search to higher dimensions. Allowing a method to perform $k \geq n$ iterations, the optimal complexity is
$$O(n \log \tfrac{MR}{\varepsilon}),$$
first-order oracle calls, which is achieved by the center of gravity method. However, this method is completely unpractical. The ellipsoid method, that we discussed in previous lecture, has slightly worse oracle complexity:
$$O(n^2 \log \tfrac{MR}{\varepsilon}). \tag{4.61}$$

At the same time, the ellipsoid method is much more practical and it is possible to implement. However, due to the arithmetic cost of each iteration of order $O(n^2)$ and rather conservative steps, the ellipsoid method seems suitable for problems of range $n \approx 10 - 20$.

- $n \to \infty$: **large-scale optimization** — *cheap gradient / subgradient methods.* When $k \leq n$, the complexity of the subgradient method,

$$O\Big(\Big[\frac{MR}{\varepsilon}\Big]^2\Big), \tag{4.62}$$

is optimal, matching the lower bound that we have just proved. Note that in contrast to cutting-plane schemes, the complexity bound (4.62) *does not depend on the dimension*[2]. This makes it favorable for solving problems of huge dimensions, where no other methods can work.

Comparing the complexity bounds (4.62) and (4.61), we see that the subgradiend method is superior than the ellipsoid method (in terms of oracle complexity), at least when

$$n \quad \geq \quad \Big[\frac{MR}{\varepsilon}\Big].$$

Therefore,

1. The subgradient method is *superior* when the dimension is extremely large.
2. Conversely, when the target accuracy ε is very small ($\varepsilon \to 0$), the subgradient method cannot be expected to solve the problem to such precision.
3. The dependence of large-scale first-order methods on ε improves significantly when the problem is smooth (e.g., the gradient method: $O(1/\varepsilon)$, the fast gradient method: $O(1/\varepsilon^{1/2})$), or additionally strongly convex.

Another crucial aspect defining the success of subgradient-type methods is their *low per-iteration cost* and *resilience to noise*. As we will see, the same analysis that we applied to the subgradient method is directly applicable to stochastic methods, such as stochastic gradient descent (SGD) and its adaptive variants.

Finally, note an unsatisfying gap in this picture: problems of **moderate dimension** $10 \leq n \leq 10^4$, which routinely appear in computational practice. In the final part of the course, we will discuss *interior-point methods* (IPMs), that close this gap.

Interior-point methods require knowledge of the precise structure of the problem (such as linear or quadratic programming). Thus, they cannot formally be categorized as black-box methods. However, such structural knowledge is often available, and the IPMs remain extremely efficient for solving moderate-size problems with high accuracy, while possessing polynomial-time complexity.

4.6 Adaptive Stepsizes for Stochastic Methods

Motivation. We discussed stochastic optimization previously in Section (2.5). We showed that for smooth (but possibly non-convex) problems, stochastic gradient method of the form

$$x_{k+1} \quad = \quad x_k - \tfrac{1}{M} g_k, \tag{4.63}$$

[2]At least explicitly. As we will see, it may depend on the dimension n, through parameters M and R.

where $g_k = g(\xi_k, x_k)$ is a stochastic unbiased estimate of the gradient $\nabla f(x_k)$ of our objective, will converge to a stationary point as soon as the step size parameter $M > 0$ is sufficiently small. Namely, we set

$$M \;:=\; L \cdot \max\Big\{1, \tfrac{2\sigma^2}{\varepsilon^2}\Big\}, \tag{4.64}$$

where L is the Lipschitz constant of the gradient, σ^2 is bound for the variance of g_k and $\varepsilon > 0$ is the target accuracy in terms of the gradient norm. Then, in order to reach a random point $E\big[\|\nabla f(\bar{x})\|\big] \leq \varepsilon$, we showed (see Theorem 4.2.6) that it is enough to perform

$$K \;=\; O\Big(L(f(x_0) - f^\star) \cdot \big[\tfrac{1}{\varepsilon^2} + \tfrac{\sigma^2}{\varepsilon^4}\big]\Big)$$

iterations (4.63). We have two goals now:

- **Study** stochastic methods for *convex problems*: having explored various methods and proof techniques in convex optimization, it is natural to expect that convexity will also benefit stochastic problems.
- **Develop** an *adaptive stepsize* rule: the constant rule in (4.64) depends on two parameters, L and σ, which are usually unknown in practice. Unlike determenistic methods, we cannot use an adaptive *line-search* to ensure progress of every step. Furthermore, a constant stepsize (4.64) seems too conservative, as it prevents the method from improving convergence when *local values* of L and σ are small.

It appears that ideas from non-smooth convex optimization work nicely in the stochastic case. Informally speaking, stochasticity can be treated as a form of "non-smoothness" in the problem. Thus, the subgradient methods developed initially for non-smooth convex problems are simple to generalize to stochastic settings.

Problem formulation. We consider the following convex optimization problem,

$$\min_{x \in Q} f(x),$$

where $Q \subseteq \mathbb{R}^n$ is a bounded convex set. The boundedness will be a crucial assumption for the analysis of our method. However, if the initial problem is over unbounded set, we can always introduce an additional simple constrain of the large enough Euclidean ball around the origin.

We denote the diameter of Q in the Euclidean norm by:

$$D \;:=\; \max_{x,y \in Q} \|x - y\|.$$

We assume that $f : Q \to \mathbb{R}$ is convex, possibly non-differentiable, and denote by M its Lipschitz constant.

Now we only have access to the *stochastic first-order oracle*, for any $x \in Q$ we assume we can sample a random variable ξ and compute a vector:

$$g(x; \xi) \;\in\; \mathbb{R}^n,$$

that is a stochastic substitute for a subgradient. We assume that

- This is unbiased estimator of a subgradient:

$$\mathbb{E}_\xi[g(x, \xi)] \;=\; f'(x) \qquad \text{for some} \qquad f'(x) \;\in\; \partial f(x).$$

- It has bounded variance:

$$\mathbb{E}_\xi\left[\|g(x;\xi) - \nabla f(x)\|^2\right] \;\leq\; \sigma^2,$$

for $\sigma > 0$, which is a parameter of our problem. As a consequence, we have (formula (2.30) in Section 2.5):

$$E_\xi\left[\|g(x,\xi)\|^2\right] \;\leq\; \sigma^2 + \|f'(x)\|^2 \;\leq\; \sigma^2 + M^2.$$

4.6.1 Stochastic Subgradient Method

For the deterministic problem, we have analyzed the following variant of the subgradient method with normalized stepsizes:

$$x_{k+1} \;=\; \pi_Q\Big(x_k - \tfrac{\gamma}{\|f'(x_k)\|} f'(x_k)\Big), \qquad \text{with} \qquad \gamma \;:=\; \tfrac{D}{\sqrt{K+1}}, \tag{4.65}$$

where $K \geq 1$ is a fixed number of iterations. This method gave us the optimal complexity for the non-smooth convex optimization.

In stochastic setting, it is a natural idea to replace $f'(x_k) \mapsto g_k$ in formula (4.65). We obtain a step in the following normalized stochastic direction:

$$x_{k+1} \;=\; \pi_Q\Big(x_k - \tfrac{\gamma}{\|g_k\|} g_k\Big),$$

which has the following drawbacks:

- It is more difficult to analyze the normalized random variable $\frac{g_k}{\|g_k\|}$ than the deterministic update in (4.65). We can also use instead the following update, which is easier and resembles our primary approach (4.63):

$$x_{k+1} \;=\; \pi_Q\Big(x_k - \eta g_k\Big), \tag{4.66}$$

- However, in both cases, the method becomes too chaotic, performing a lot of random fluctuations and we have to tune parameter $\eta > 0$ to be really small to ensure convergence. For update of (4.66) to work, we have to know variance σ to choose the step-size, as in (4.64).

- In the stepsize formula for (4.65), even in the deterministic case, we still need to fix the number of iterations $K > 0$ in advance and use it for the stepsize. If, after K iterations, we want to continue running our method, proposed γ will not work anymore.

4.6.2 Adaptive Stepsizes

We consider is a more advanced *adaptive* stepsize rule that solves all these problems at once. Namely, we perform the following algorithm.

Algorithm 4.4: *Stochastic Subgradient Method with Adaptive Stepsizes.*

Initialization: $x_0 \in Q$ and $S_0 = 0$.

For $k = 0 \dots K-1$ **iterate:**

1. Sample ξ_k and compute stochastic gradient $g_k := g(x_k, \xi_k)$
2. Update $S_{k+1} := S_k + \|g_k\|^2$ and set $\beta_k := \frac{\sqrt{S_{k+1}}}{D}$.
3. Perform the step:

$$x_{k+1} \;=\; \operatorname*{argmin}_{y \in Q} \Big[\langle g_k, y - x_k \rangle + \tfrac{\beta_k}{2} \|y - x_k\|^2 \Big]$$

Return $\bar{x}_K = \frac{1}{K} \sum\limits_{i=1}^{K} x_i$.

Remark 4.6.1. One step of this method is given by:

$$x_{k+1} \;=\; \pi_Q\Big(x_k - \tfrac{1}{\beta_k} g_k\Big), \qquad \text{where} \qquad \beta_k \;:=\; \tfrac{1}{D} \sqrt{\|g_0\|^2 + \ldots + \|g_k\|^2}, \tag{4.67}$$

while for deterministic normalized subgradient method (4.65) we had:

$$x_{k+1} \;=\; \pi_Q\Big(x_k - \tfrac{1}{\alpha_k} f'(x_k)\Big), \qquad \text{where} \qquad \alpha_k \;:=\; \tfrac{\|f'(x_k)\| \sqrt{K+1}}{D}.$$

Therefore, new more advanced formula (4.67) replaces the subgradient norm at the current point by the average of all (stochastic) subgradient norms:

$$\|f'(x_k)\| \sqrt{k+1} \;\approx\; \sqrt{S_{k+1}} \;=\; \sqrt{\|g_0\|^2 + \ldots \|g_k\|^2}.$$

Remark 4.6.2. Steps of the method are *independent* of a fixed number of iterations $K \geq 1$. We use K only as a stopping condition and to form the output. If after K iterations we decide to continue running the method, we can easily do that without any restarts.

Adaptive analysis. We want to show the convergence for this algorithm. Let us start our analyses the same way as we did for the subgradient method. We denote our model by

$$m_k(y) \;:=\; \langle g_k, y - x_k \rangle + \tfrac{\beta_k}{2} \|y - x_k\|^2,$$

and x_{k+1} is defined as the minimizer of this model over Q. Due to strong convexity of the model, we have that

$$m_k(y) \;\geq\; m_k(x_{k+1}) + \tfrac{\beta_k}{2} \|y - x_{k+1}\|^2.$$

We also notice that

$$\begin{aligned} m_k(x_{k+1}) \;&=\; \langle g_k, x_{k+1} - x_k \rangle + \tfrac{\beta_k}{2} \|x_{k+1} - x_k\|^2 \\ &\geq\; \min_{h \in \mathbb{R}^n} \Big[\langle g_k, h \rangle + \tfrac{\beta_k}{2} \|h\|^2 \Big] \;=\; -\tfrac{1}{2\beta_k} \|g_k\|^2. \end{aligned}$$

Hence, we get

$$\tfrac{1}{2\beta_k} \|g_k\|^2 + \tfrac{\beta_k}{2} \|y - x_k\|^2 \;\geq\; \tfrac{\beta_k}{2} \|y - x_{k+1}\|^2 + \langle g_k, x_k - y \rangle.$$

Further, we can substitute $y := x^\star$ (any minimizer for our problem). We get:

$$\frac{1}{2\beta_k}\|g_k\|^2 + \frac{\beta_k}{2}\|x^\star - x_k\|^2 \;\geq\; \frac{\beta_k}{2}\|x^\star - x_{k+1}\|^2 + \langle g_k, x_k - x^\star\rangle. \tag{4.68}$$

Notice that

$$\mathbb{E}_{\xi_k}\big[\langle g_k, x_k - x^\star\rangle\big] \;=\; \langle f'(x_k), x_k - x^\star\rangle \;\geq\; f(x_k) - f^\star.$$

Therefore, the right hand side of (4.68) gives us the progress of the method, and the left hand side contains an "error" of one step: $\frac{1}{2\beta_k}\|g_k\|^2$. However, to make (4.68) telescoping, we want to have β_{k+1} in the right hand side instead of β_k.

We can use the following simple but crucial observation, for $\beta_{k+1} \geq \beta_k$, using the boundedness of the feasible set:

$$\begin{aligned}\frac{\beta_{k+1}}{2}\|x^\star - x_{k+1}\|^2 &= \frac{\beta_k}{2}\|x^\star - x_{k+1}\|^2 + \frac{\beta_{k+1}-\beta_k}{2}\|x^\star - x_{k+1}\|^2 \\ &\leq \frac{\beta_k}{2}\|x^\star - x_{k+1}\|^2 + \frac{\beta_{k+1}-\beta_k}{2}D^2.\end{aligned}$$

Thus, we have established the following consequence for one step of the method.

Lemma 4.6.3. *Let $\beta_{k+1} \geq \beta_k$. Then,*

$$\frac{1}{2\beta_k}\|g_k\|^2 + \frac{\beta_{k+1}-\beta_k}{2}D^2 + \frac{\beta_k}{2}\|x^\star - x_k\|^2 \;\geq\; \frac{\beta_{k+1}}{2}\|x^\star - x_{k+1}\|^2 + \langle g_k, x_k - x^\star\rangle. \tag{4.69}$$

Now, we have two "error terms" in the left hand side of (4.69). For convenience, let us denote $\boxed{\beta_{-1} := 0}$. Then, we notice that, by the definition of β_k, for every $k \geq 0$:

$$\begin{aligned}(\beta_k - \beta_{k-1})D^2 &= (\sqrt{S_{k+1}} - \sqrt{S_k})D \;=\; \frac{S_{k+1}-S_k}{\sqrt{S_{k+1}}+\sqrt{S_k}} \cdot D \;=\; \frac{\|g_k\|^2}{\sqrt{S_{k+1}}+\sqrt{S_k}} \cdot D \\ &\geq \frac{\|g_k\|^2}{\sqrt{S_{k+1}}} \cdot \frac{D}{2} \;=\; \frac{\|g_k\|^2}{2\beta_k}.\end{aligned}$$

This important observations allows us to simplify the left hand side of (4.69).

Lemma 4.6.4. *Let β_k be chosen as in Algorithm 4.4. Then,*

$$D^2 \cdot \Big(\frac{\beta_{k+1}-\beta_k}{2} + \beta_k - \beta_{k-1}\Big) + \frac{\beta_k}{2}\|x^\star - x_k\|^2 \;\geq\; \frac{\beta_{k+1}}{2}\|x^\star - x_k\|^2 + \langle g_k, x_k - x^\star\rangle.$$

Telescoping this inequality for the first $K \geq 0$ iterations, we get

$$\begin{aligned}\sum_{i=0}^{K-1}\langle g_i, x_i - x^\star\rangle &\leq \frac{\beta_0}{2}\|x^\star - x_0\|^2 + D^2 \cdot \Big(\frac{\beta_K - \beta_0}{2} + \beta_{K-1} - \beta_{-1}\Big) \\ &\leq \tfrac{3}{2}D^2\beta_K.\end{aligned}$$

Let us take the expectations for the left and the right hand sides. For the left hand side, we have:

$$\begin{aligned}\mathbb{E}\Big[\sum_{i=0}^{K-1}\langle g_i, x_i - x^\star\rangle\Big] &= \mathbb{E}\Big[\sum_{i=0}^{K-1}\langle f'(x_i), x_i - x^\star\rangle\Big] \;\geq\; K \cdot \mathbb{E}\Big[\tfrac{1}{K}\sum_{i=0}^{K-1}\big[f(x_i) - f^\star\big]\Big] \\ &\geq K \cdot \mathbb{E}\big[f(\bar{x}_K) - f^\star\big].\end{aligned}$$

For the right hand side, we obtain, using Jensen's inequality for concave function $\sqrt{\cdot}$, that

$$\mathbb{E}\Big[\beta_K\Big] \;=\; \tfrac{1}{D}\mathbb{E}\Big[\sqrt{\sum_{i=0}^{K-1}\|g_i\|^2}\,\Big] \;\leq\; \tfrac{1}{D}\sqrt{\sum_{i=0}^{K-1}\mathbb{E}[\|g_i\|^2]} \;\leq\; \frac{\sqrt{K}\cdot\sqrt{\sigma^2+M^2}}{D} \;\leq\; \frac{\sqrt{K}(\sigma+M)}{D}.$$

Combining these two bounds together, we have proved the following theorem.

Theorem 4.6.5. *It holds,*

$$\mathbb{E}\Big[f(\bar{x}_K) - f^\star\Big] \quad \leq \quad \frac{3D^2\mathbb{E}[\beta_K]}{2K} \quad \leq \quad \frac{3(\sigma+M)D}{2\sqrt{K}}.$$

This is the same rate as for the deterministic subgradient method, where we replaced M by $\sigma + M$ in the complexity estimate. Note that this adaptive strategy work for the deterministic method as well ($\sigma = 0$), eliminating the need to fix the number of iterations $K \geq 0$ within the stepsize.

4.7 Smooth Stochastic Optimization II

We consider unconstrained minimization problem,

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex, and is gradient is Lipschitz coninuous:

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

Let us study the performance of the stochastic gradient method on this class of problems.

Which rate we can expect?

For simplicity, we consider unconstrained optimization, so $\nabla f(x^\star) = 0$.

Direct approach. Notice that

$$\|\nabla f(y)\| \quad = \quad \|\nabla f(y) - \nabla f(x^\star)\| \quad \leq \quad L\|y - x^\star\| \quad \leq \quad LD \quad =: \quad M,$$

where D is a distance $D \geq \|x_0 - x^\star\|$. If we can estimate D somehow, we can introduce a ball $Q := \{x : \|x - x_0\| \leq D\}$ and apply the method from the previous lecture, which will give us the rate

$$\mathbb{E}[f(\bar{x}_K) - f^\star] \quad \leq \quad \frac{LD^2}{\sqrt{k}} + \frac{\sigma D}{\sqrt{k}}, \tag{4.70}$$

where $\sigma > 0$ is the uniform bound on the variance of stochastic gradients.

It appears that the we cannot improve the last "variance term" in (4.70) for the general stochastic optimization problems. However, the first term can be improved. For the basic stochastic gradient method we can ensure the rate of:

$$\mathbb{E}[f(\bar{x}_K) - f^\star] \quad = \quad O\Big(\frac{LD^2}{k} + \frac{\sigma D}{\sqrt{k}}\Big), \tag{4.71}$$

and, for the accelerated stochastic gradient method, we can have the optimal rate:

$$\mathbb{E}[f(\bar{x}_K) - f^\star] \quad = \quad O\Big(\frac{LD^2}{k^2} + \frac{\sigma D}{\sqrt{k}}\Big). \tag{4.72}$$

In this note, we establish (4.71).

Useful inequality. Let us recall the following useful inequality, which is a consequence of convexity and smoothness (see Theorem 3.2.1 in Section 3.2):

$$f(y) - f(x) - \langle \nabla f(x), y - x\rangle \quad \geq \quad \frac{1}{2L}\|\nabla f(y) - \nabla f(x)\|^2, \qquad \forall x, y \in \mathbb{R}^n. \tag{4.73}$$

Main lemma. For simplicity, we analyze the method with a constant stepsize $\eta > 0$, while an employment of adaptive stepsizes, like in the previous lecture, is also possible.

Thus, we perform iterations, starting from some $x_0 \in \mathbb{R}^n$:

$$x_{k+1} \;=\; x_k - \eta g_k, \qquad k \geq 0,$$

where $g_k \in \mathbb{R}^n$ is a stochastic gradient.

Note that

$$\begin{aligned} \tfrac{1}{2}\|x_{k+1} - x^\star\|^2 &= \tfrac{1}{2}\|x_k - x^\star - \eta g_k\|^2 \\ &= \tfrac{1}{2}\|x_k - x^\star\|^2 + \tfrac{\eta^2}{2}\|g_k\|^2 - \gamma\langle g_k, x_k - x^\star\rangle. \end{aligned}$$

Rearranging the terms we get a familiar expression:

$$\tfrac{\gamma^2}{2}\|g_k\|^2 + \tfrac{1}{2}\|x_k - x^\star\|^2 \;=\; \tfrac{1}{2}\|x_{k+1} - x^\star\|^2 + \gamma\langle g_k, x_k - x^\star\rangle.$$

Now, we notice that

$$\mathbb{E}_{\xi_k}\|g_k - \nabla f(x_k)\|^2 \;=\; \mathbb{E}_{\xi_k}\|g_k\|^2 - \|\nabla f(x_k)\|^2,$$

thus

$$\mathbb{E}\|g_k - \nabla f(x_k)\|^2 \;=\; \mathbb{E}\|g_k\|^2 - \mathbb{E}\|\nabla f(x_k)\|^2.$$

At the same time, we have

$$\mathbb{E}_\xi\langle g_k, x_k - x^\star\rangle \;=\; \langle \nabla f(x_k), x_k - x^\star\rangle \;\overset{(4.73)}{\geq}\; f(x_k) - f^\star + \tfrac{1}{2L}\|\nabla f(x_k)\|^2.$$

Thus,

$$\mathbb{E}\langle g_k, x_k - x^\star\rangle \;\geq\; \mathbb{E}[f(x_k) - f^\star] + \tfrac{1}{2L}\mathbb{E}\|\nabla f(x_k)\|^2$$

Combining these observations together, we get the following progress of one step. Denote

$$R_k^2 \;:=\; \mathbb{E}\|x_k - x^\star\|^2.$$

Lemma 4.7.1. *We have:*

$$R_k^2 - R_{k+1}^2 + \tfrac{\gamma^2}{2}E\|\nabla f(x_k)\|^2 + \tfrac{\gamma^2}{2}\mathbb{E}\|\nabla f(x_k) - g_k\|^2 \;\geq\; \gamma\mathbb{E}[f(x_k) - f^\star] + \tfrac{\gamma}{2L}\|\nabla f(x_k)\|^2.$$

Corollary 4.7.2. *Consequently, for $\gamma \leq \frac{1}{L}$ we have:*

$$R_k^2 - R_{k+1}^2 + \tfrac{\gamma^2}{2}\mathbb{E}\|\nabla f(x_k) - g_k\|^2 \;\geq\; \gamma\mathbb{E}[f(x_k) - f^\star].$$

Convergence rate. We can bound the variance of the gradients at iteration by σ^2. We obtain the following inequality,

$$\mathbb{E}[f(x_k) - f^\star] \;\leq\; \tfrac{1}{2\gamma}R_k^2 - \tfrac{1}{2\gamma}R_{k+1}^2 + \tfrac{\gamma}{2}\sigma^2.$$

Telescoping and using Jensen's inequality, we have

$$\begin{aligned} \gamma k\mathbb{E}[f(\bar{x}_k) - f^\star] &\leq \gamma\mathbb{E}[\textstyle\sum_{i=0}^{k-1}(f(x_i) - f^\star)] \;\leq\; \tfrac{1}{2\gamma}(R_0^2 - R_k^2) + \tfrac{\gamma}{2}\sigma^2 k \\ &\leq \tfrac{1}{2\gamma}R_0^2 + \tfrac{\gamma}{2}\sigma^2 k. \end{aligned}$$

Let us minimize the right hand side with respect to γ. We get the optimal choice

$$\gamma^\star \;=\; \tfrac{R_0}{\sigma\sqrt{k}}.$$

Taking into account the other condition, $\gamma \le \frac{1}{L}$, we set $\boxed{\gamma := \min\{\tfrac{1}{L}, \tfrac{R_0}{\sigma\sqrt{k}}\}}$

For that choice we get

$$\mathbb{E}[f(\bar{x}_k) - f^\star] \;\le\; \max\{\tfrac{LR_0^2}{2k}, \tfrac{R_0\sigma}{2\sqrt{k}}\} + \tfrac{R_0\sigma}{\sqrt{k}} \;=\; O\Big(\tfrac{LR_0^2}{k} + \tfrac{\sigma R}{\sqrt{k}}\Big).$$

The oracle complexity is

$$O\Big(\tfrac{LR_0^2}{\varepsilon} + \big[\tfrac{\sigma R_0}{\varepsilon}\big]^2\Big).$$

4.7.1 Variance Reduction via Minibatching

Instead of one sample, we use $m \ge 1$ samples:

$$g_k \;:=\; \tfrac{1}{m}\sum_{i=1}^m g(x_k, \xi_{k,i})$$

It is clear that $\mathbb{E}g_k = \nabla f(x_k)$. What will be the variance of g_k?

Denote $\delta_i := g(x_k, \xi_{k,i}) - \nabla f(x_k)$. Hence, $\mathbb{E}\delta_i = 0$. Observe that

$$\begin{aligned}
\mathbb{E}\Big[\big\|g_k - \nabla f(x_k)\big\|^2\Big] \;&=\; \mathbb{E}\Big[\big\|\tfrac{1}{m}\sum_{i=1}^m \delta_i\big\|^2\Big] \;=\; \mathbb{E}\Big[\langle \tfrac{1}{m}\sum_{i=1}^m \delta_i, \tfrac{1}{m}\sum_{j=1}^m \delta_j\rangle\Big] \\
&=\; \tfrac{1}{m^2}\mathbb{E}\sum_{i=1}^m \|\delta_i\|^2 + \tfrac{2}{m^2}\sum_{i<j}\mathbb{E}\langle\delta_i, \delta_j\rangle \\
&=\; \tfrac{1}{m^2}\mathbb{E}\sum_{i=1}^m \|\delta_i\|^2 + \;\le\; \tfrac{1}{m}\sigma^2.
\end{aligned}$$

Therefore, we can reduce the value σ by $\sqrt{m}$, when using minibatch of size $m \ge 1$.

What is the total complexity of the method in terms of the *total sampled gradients*, where at each iteration sample m gradients. A natural choice would be to make the two complexity terms equal:

$$\tfrac{LR_0^2}{\varepsilon} \;\stackrel{?}{=}\; \tfrac{1}{m}\big[\tfrac{\sigma R_0}{\varepsilon}\big]^2,$$

which leads to the choice:

$$m \;:=\; 1 + \tfrac{\varepsilon}{LR_0^2}\cdot\big[\tfrac{\sigma R_0}{\varepsilon}\big]^2 \;=\; 1 + \tfrac{\sigma^2}{\varepsilon L}.$$

Performing this amount of samples each iteration, we ensure the same rate as for the deterministic method. At the same time, the total number of samples over all iterations is:

$$m\cdot O(\tfrac{LR_0^2}{\varepsilon}) \;=\; \Big(1 + \tfrac{\sigma^2}{\varepsilon L}\Big)\cdot O(\tfrac{LR_0^2}{\varepsilon}) \;=\; O\Big(\tfrac{LR_0^2}{\varepsilon} + \tfrac{\sigma^2 R_0^2}{\varepsilon^2}\Big).$$

So, the total number of samples remains the same. In practice, it is always useful use a small mini-batch ($m \approx 100$ or more).

4.8 Mirror Descent and Accuracy Certificates

This section, we study general techniques, that are built on top of the basic subgradient method that we have discussed few previous lectures. These techniques, called *mirror descent* or *dual averaging* are quite general and useful. However, to see their full power, we illustrate our developments with the problems of the following structure.

4.8.1 Application Example: Min-Max Problems

A general form of a convex optimization problem is as follows:

$$\min_{x \in Q} f(x), \tag{4.74}$$

where $Q \subseteq \mathbb{R}^n$ is a convex set and f is a convex function. We can apply first-order methods in a black-box manner directly to (4.74). However, in practice, we always know something more about the problem and the actual structure of the objective. Examples include:

- *Fully-composite problems* (see Section 3.7), where we can identify their *smooth components* and then apply efficient methods from smooth optimization, such as the fast gradient method. The main assumption is that non-smooth parts of the problem as *simple* (e.g. a non-smooth regularizer or simple constraints).
- *Finite-sum minimization.* When the function f in (4.74) is represented as a finite sum of smooth objectives, we can apply efficient stochastic methods with variance reduction, such as SVRG, that preserve fast rates of the deterministic methods.

Now, we consider another specific structure of the problem, called *min-max*, that is very popular in practice:

$$f(x) \;:=\; \max_{u \in \Omega} F(x, u), \tag{4.75}$$

where $\Omega \subset \mathbb{R}^m$ is a bounded convex set, and $F(\cdot, u)$ is convex for any u and $F(x, \cdot)$ is concave for any x.

Hence, our original problem is the min-max or *saddle point* problem:

$$\begin{aligned} \min_{x \in Q} f(x) &= \min_{x \in Q} \max_{u \in \Omega} F(x, u) \\ &\geq \max_{u \in \Omega} \min_{x \in Q} F(x, u) \;=:\; \max_{u \in \Omega} \varphi(u), \end{aligned}$$

where $\varphi(u) := \min_{x \in Q} F(x, u)$. We call the latter problem:

$$\max_{u \in \Omega} \varphi(u), \tag{4.76}$$

as the *adjoint* or *dual* problem to (4.74). Note that the same primal problem (4.74) can have different min-max representations of the objective, and therefore the corresponding adjoint problems (4.76) depends on this representation and is not uniquely defined.

In most cases, we have so called *strong duality* (e.g., when both sets are compact and F is convex-concave and continuous) so these two problems are mathematically equivalent and symmetric:

$$\min_{x \in Q} f(x) \;=\; \max_{u \in \Omega} \varphi(u).$$

However, in practice, one of these problems can be much easier to solve than the other, due to different dimensionality, $x \in Q \subseteq \mathbb{R}^n$ and $u \in \Omega \subseteq \mathbb{R}^m$ and specific structure of F.

Since our initial interest was the primal problem (4.74), we assume that we can efficiently compute the *first-order oracle* for f along with the following information, for any given point $x \in Q$:

- Function value: $f(x) := \max\limits_{u \in \Omega} F(x, u)$;
- A minimizer: $u(x) := \operatorname{argmax}\limits_{u \in \Omega} F(x, u)$;
- A subgradient: $f'(x) = F'_x(x, u(x)) \in \partial_x F(x, u(x))$;

where F'_x is a subgradient of F with respect to the first variable x, and $\partial_x F$ is the partial subdifferential (the set of all such subgradients). In practice, access to $u(x)$ is almost always available, once we assume an efficient way of computing the function value $f(x)$ itself.

It appears that having a method that solves the primal problem (4.74), we can also automatically generate solutions for the adjoint problem (4.76), that we will show in the next lecture.

Matrix games. The simplest example of the previous setting is the following objective,

$$F(x, u) \quad = \quad \langle Ax, u \rangle + \langle b, u \rangle + \langle c, x \rangle, \tag{4.77}$$

where $A \in \mathbb{R}^{m \times n}$ is a given matrix, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ are some vectors.

We assume that we have two players x and u. The *strategy* of every player belongs to the corresponding simplex:

$$x \in Q = \Delta_n = \{x \in \mathbb{R}^n_{\geq 0} : \sum_{i=1}^n x^{(i)} = 1\}$$

and

$$y \in \Omega = \Delta_m = \{y \in \mathbb{R}^m_{\geq 0} : \sum_{i=1}^m y^{(i)} = 1\}.$$

Therefore, our problem is the following one:

$$\min_{x \in \Delta_n} \max_{u \in \Delta_m} \Big[\langle Ax, u \rangle + \langle b, u \rangle + \langle c, x \rangle \Big]$$

It corresponds to choosing the best strategy of playing for the first player, which minimizes their loss under *any strategy* of the second player. In other words, we want to win as much as possible in the worst-case scenario, when our opponent plays the best (maximizing our loss).

Then, in the terminology of primal problem (4.74), our objective is

$$\begin{aligned} f(x) &= \max_{u \in \Delta_m} \Big[\langle Ax, u \rangle + \langle b, u \rangle \Big] + \langle c, x \rangle \\ &= \max_{u \in \Delta_m} \sum_{i=1}^m u^{(i)} \big[\langle a_i, x \rangle + b_i \big] + \langle c, x \rangle \qquad (4.78) \\ &= \max_{1 \leq i \leq m} \big[\langle a_i, x \rangle + b_i \big] + \langle c, x \rangle, \end{aligned}$$

where $a_1, \ldots, a_m \in \mathbb{R}^n$ are the rows of our matrix A.

Therefore, to compute the function value $f(x)$, we need to find the variable

$$u(x) \;=\; e_i \;\in\; \Delta_m$$

with $1 \leq i \leq m$ such that $f(x) = \langle a_i, x\rangle + b_i + \langle c, x\rangle$. The corresponding subgradient can be set as follows:

$$f'(x) \;=\; a_i.$$

Performance of the subgradient method. In previous lectures, we analyzed the following subgradient method that we can directly apply to primal problem (4.74),

$$x_{k+1} \;=\; \pi_Q\big(x_k - \eta_k f'(x_k)\big) \tag{4.79}$$

and proved the following convergence guarantee:

$$f(\bar{x}_k) - f^\star \;\leq\; \tfrac{MD}{\sqrt{K}}, \tag{4.80}$$

for example, using the normalized stepsizes, $\eta_k := \frac{\gamma}{\|f'(x_k)\|}$, or the constant $\eta_k \equiv \frac{\gamma}{M}$, where $\gamma := \frac{D}{\sqrt{K}}$ (Section 4.4), or, using the *adaptive stepsizes* (Section 4.6) that do not depend on the fixed number of iterations. Here,

$$D \;\geq\; \|x_0 - x^\star\|_2 \qquad \text{and} \qquad \|f'(x)\|_2 \;\leq\; M, \qquad \forall x \in Q, \quad \forall f'(x) \in \partial f(x). \tag{4.81}$$

We also proved that the rate of (4.80) is *optimal* (Section 4.5), and the function from the lower bound construction actually matches the form of our problem (4.78). Therefore, it seems like the end of story.

How can the convergence result (4.80) be further improved?

- We do not know **when to stop the method**? Even though we can use adaptive stepsizes, that do not use a fixed number of iterations K in the method, we do not have a *computable stopping condition* in the algorithm, which would ensure that

  $$f(\bar{x}_K) - f^\star \;\leq\; \varepsilon$$

  for a given accuracy $\varepsilon > 0$, unless we know $f^\star$.

- A related to the previous question: can we say anything about **solving the adjoint problem** (4.76)?

- Another crucial observation is that we **fix the geometry** by choosing $\|\cdot\|_2$ norm in the method (4.79): and this is the norm in which parameters M and R are measured in (4.81).

What can be wrong with $\|\cdot\|_2$ norm?

Example 4.8.1. Consider the objective (4.78) from the previous section. Hence $Q = \Delta_n$. Then,

$$\operatorname{diam}_{\|\cdot\|_2}(\Delta_n) \;=\; \|e_1 - e_2\|_2 \;=\; \sqrt{2}.$$

Now, assume that we have $f'(x) = a_i + c = (1, \ldots, 1)^\top \in \mathbb{R}^n$. Then

$$\|f'(x)\|_2 \;=\; \sqrt{n}.$$

Hence, $M_{\|\cdot\|_2} \geq \sqrt{n}$, and the convergence rate in (4.80) *does depend on the dimension!* If we increase n, the method will need significantly more time to solve the problem.

Example 4.8.2. For the same problem, let us choose $\|\cdot\|_1$ norm for the primal variables $x \in \Delta_n$. Then,

$$\mathrm{diam}_{\|\cdot\|_1}(\Delta_n) \;=\; \|e_1 - e_2\|_1 \;=\; 2,$$

which is an absolute constant again. However, to measure the size of the dual objects (subgradients), we use the dual norm, which is $\|\cdot\|_\infty$. In this norm, we have

$$\|f'(x)\|_\infty \;=\; \|(1,\ldots,1)\|_\infty \;=\; 1.$$

Therefore, $M_{\|\cdot\|_\infty} \approx 1$ and it is much better than $M_{\|\cdot\|_2}$.

Note that we always have $\|\cdot\|_\infty \le \|\cdot\|_2 \le \|\cdot\|_1$ and hence $M_{\|\cdot\|_\infty} \le M_{\|\cdot\|_2}$ always, while in a particular instance, $M_{\|\cdot\|_\infty}$ can be significantly better as in our example.

Why gradient method is not enough? We want to have a modification of the subgradient method that is more suitable to the problem geometry. We assume that we have a *primal space* of variables, which we denote by $\mathbb{R}^n$:

$$x \;\in\; Q \;\subseteq\; \mathbb{R}^n.$$

We use an arbitrary norm $\|\cdot\|$ in this space (not necessary Euclidean).

Then, we treat subgradients as linear forms: $\langle f'(x), \cdot\rangle$ which are *dual objects*, and we use the corresponding dual norm for them, defined by

$$\|s\|_* \;=\; \max_{h\in\mathbb{R}^n \,:\, \|h\|\le 1} \langle s, h\rangle.$$

Recall that we already saw the gradient method for arbitrary norms (Section 2.3). For a smooth objective (that is, the gradient of f is Lipschitz w.r.t. a fixed norm), we can perform:

$$x_{k+1} \;=\; \mathop{\mathrm{argmin}}_{y\in Q}\Big[f(x_k) + \langle \nabla f(x_k), y - x_k\rangle + \tfrac{L}{2}\|y - x_k\|^2\Big], \tag{4.82}$$

and we can establish global convergence rate of $O(1/k)$ in terms of the functional residual, for this process.

The first issue is the constraint set Q. When the norm is Euclidean, iteration (4.82) can be rewritten in terms of the projection operation. However, in general, for an arbitrary norm, this is a nontrivial subproblem.

The second issue is that this analysis worked only for *smooth function* (but possibly non-convex). Performing iterations of form (4.82), we can ensure a positive progress of each step:

$$f_k - f_{k+1} \;\ge\; \tfrac{1}{2LD^2} f_k^2, \tag{4.83}$$

leading to the desired rate. These iterations are based on *local relaxation*.

In non-smooth optimization ($L \to \infty$), we are not able to establish the local improvement for our objective (4.83). The methods of non-smooth convex optimization are based on the idea of building a *global model* of the objective (or building a *localizer set* that containts a solution). Our analysis was substantially based on the algebraic properties of the Euclidean norm, in particular, on

$$\textit{strong convexity of the regularizer } \tfrac{1}{2}\|x\|_2^2.$$

However, an arbitrary norm is not strongly convex in general.

Exercise 4.8.1. Consider $d(x) = \|x\|_1^2$, $x \in \mathbb{R}^n$, and show that it is not strongly convex for $n > 1$.

4.8.2 Arbitrary Regularizers: Bregman Divergence

The key idea of the *mirror descent* algorithm is to replace the square norm $\|\cdot\|^2$ by an arbitrary distance function $d : \operatorname{int} Q \to \mathbb{R}$.

The main assumption is that d is *simple* and at least *strictly convex* differentiable function. It is also convenient to define the following associated *Bregman Divergence*:

$$\beta_d(x;y) \;=\; d(y) - d(x) - \langle \nabla d(x), y - x\rangle \;>\; 0, \qquad x \neq y. \tag{4.84}$$

Strict convexity of d means that the inequality in (4.82) is strict: ">". Geometrically, $g(y) := \beta_d(x;y)$ is a "rotation" of our regularizer d such that it is minimum in x.

Example 4.8.3. Let $d(x) = \frac{1}{2}\|x\|_2^2$ (squared Euclidean norm). Then $x_0 = 0$. We have

$$\begin{aligned}\beta_d(x;y) &= \tfrac{1}{2}\|y\|_2^2 - \tfrac{1}{2}\|x\|_2^2 - \langle x, y - x\rangle \\ &= \tfrac{1}{2}\|y - x\|^2.\end{aligned}$$

This is exactly the regularized that we used in the construction of the subgradient method.

The most popular and important example is the following one.

Example 4.8.4. Let $d(x) = \sum\limits_{i=1}^n x^{(i)} \ln x^{(i)}$ (negative Entropy). Then,

$$x_0 \;=\; (\tfrac{1}{n}, \ldots, \tfrac{1}{n}) \;\in\; \Delta_n,$$

and $d(x_0) = -\ln n$. We have

$$[\nabla d(x)]^{(i)} \;=\; 1 + \ln x^{(i)}.$$

Hence, the Bregman divergence is equal to

$$\begin{aligned}\beta_d(x;y) &= \sum_{i=1}^n y^{(i)} \ln y^{(i)} - \sum_{i=1}^n x^{(i)} \ln x^{(i)} - \sum_{i=1}^n (1 + \ln x^{(i)})(y^{(i)} - x^{(i)}) \\ &= \sum_{i=1}^n y^{(i)} \ln y^{(i)} - \sum_{i=1}^n x^{(i)} \ln x^{(i)} - \sum_{i=1}^n \ln x^{(i)}(y^{(i)} - x^{(i)}) \\ &= \sum_{i=1}^n y^{(i)} \ln \tfrac{y^{(i)}}{x^{(i)}}.\end{aligned}$$

In statistics, it is called Kullback–Leibler or KL divergence between probability distributions x and y.

Note that the second derivative of d is the diagonal matrix:

$$[\nabla f(x)]^{(i,i)} \;=\; \tfrac{1}{x^{(i)}} \;>\; 0, \qquad x \;\in\; \operatorname{int}\Delta_n \;:=\; \Big\{x \in \mathbb{R}^n_{>0} \;:\; \sum_{i=1}^n x^{(i)} = 1\Big\}$$

Hence, we know that d is strictly convex and thus $\beta_d(x;y) > 0$ for $x \neq y$. It appears that we can improve this inequality. Indeed, for any $x \in \operatorname{int}\Delta_n$ and $h \in \mathbb{R}^n$:

$$\langle \nabla^2 d(x)h, h\rangle \;=\; \sum_{i=1}^n \frac{\left(h^{(i)}\right)^2}{x^{(i)}}.$$

Then, using Cauchy-Schwarz inequality, we observe that

$$\|h\|_1 \;=\; \sum_{i=1}^n |h^{(i)}| \;=\; \sum_{i=1}^n \frac{|h^{(i)}|}{\sqrt{x^{(i)}}} \cdot \sqrt{x^{(i)}} \;\leq\; \Big(\sum_{i=1}^n \frac{|h^{(i)}|^2}{x^{(i)}}\Big)^{1/2} \cdot \Big(x^{(i)}\Big)^{1/2} \;=\; \langle \nabla^2 d(x)h, h\rangle^{1/2}.$$

Hence, $d(\cdot)$ is strongly convex with respect to $\|\cdot\|_1$ norm, and we have

$$\beta_d(x;y) \;\geq\; \tfrac{1}{2}\|y - x\|_1^2.$$

Main lemma. To analyze methods with arbitrary regularizers, we need the following simple lemma.

Let $\psi : Q \to \mathbb{R}$ be a convex function and $d : Q \to \mathbb{R}$ is a convex regularizer, both defined on an open convex set $Q \subset \mathbb{R}^n$. We can assume for simplicity that both ψ and d are differentiable, which will be sufficient for the analysis of the mirror descent, while this assumption can be relaxed. Consider the regularized objective and denote its minimum by

$$x^+ \;:=\; \operatorname*{argmin}_{y \in Q} \Big[\, g(y) \;:=\; \psi(y) + d(y) \,\Big],$$

assuming that it exists.

Lemma 4.8.5. *We have*

$$g(y) \;\geq\; g(x^+) + \beta_d(x^+; y), \qquad \forall y \in Q. \tag{4.85}$$

Proof. Note that

$$\beta_g(x; y) \;=\; \beta_\psi(x; y) + \beta_d(x; y) \;\geq\; \beta_d(x; y).$$

Rearranging the left hand side, we get

$$g(y) \;\geq\; g(x) + \langle \nabla g(x), y - x \rangle + \beta_d(x; y).$$

Substituting $x := x^+$ and using the optimality condition: $\langle \nabla g(x^+), y - x^+ \rangle \geq 0$, for all $y \in Q$ (see Corollary 4.32 in Section 4.4.1), completes the proof. □

Remark 4.8.6. Note that the extra non-negative term $\beta_d(x^+; y)$ in (4.85) is an improvement of the trivial inequality: $g(y) \geq g(x^+)$. It is remarkable that we do not need any specific properties of g and d, such as strong convexity, to prove (4.85).

4.8.3 Mirror Descent

Problem formulation. Let us review our setting. We consider the following optimization problem,

$$\min_{x \in Q} f(x), \tag{4.86}$$

where $Q \subset \mathbb{R}^n$ is a convex set and $f : Q \to \mathbb{R}$ is a convex function.

We denote by $\|\cdot\|$ an arbitrary general norm for the primal space $\mathbb{R}^n$, and the corresponding dual norm $\|\cdot\|_*$ for measuring the subgradients.

Our main complexity parameter is constant $M > 0$:

$$\|f'(x)\|_* \;\leq\; M, \qquad \forall x \in \operatorname{int} Q, \;\; \forall f'(x) \in \partial f(x).$$

We believe that using a non-Euclidean norm can be better to capture geometry of the problem.

To perform the mirror descent method for an arbitrary norm, we introduce a *distance function*

$$d : \operatorname{int} Q \to \mathbb{R}$$

that measures distances between points in Q. We will use this function as a regularizer in our method. The main assumptions for this function is that it is

- *simple* — a regularizer should *help* us solving problem (4.86) rather than complicate the problem;

- at least *strictly convex*; ideally *strongly convex* with respect to our primal norm $\|\cdot\|$.

We denote the minimum of the distance function by

$$x_0 \;=\; \operatorname*{argmin}_{y\in Q} d(y), \tag{4.87}$$

the point from which we will start our method, a certain *center of set* Q. Thus, by default, $d(y)$ measures "how far a point $y \in \operatorname{int} Q$ from the center x_0". To measure distances between any two points $x, y \in \operatorname{int} Q$, we introduce the *Bregman divergence*:

$$\beta_d(x;y) \;:=\; d(y) - d(x) - \langle \nabla d(x), y - x\rangle.$$

Note that in general $\beta_d(x;y)$ is not symmetric: $\beta_d(x;y) \neq \beta_d(y;x)$. The property that d strictly convex implies that

$$\beta_d(x;y) \;>\; 0, \qquad x \neq y.$$

In our examples and analysis we will use that d is *strongly convex* :

$$\beta_d(x;y) \;\geq\; \tfrac{1}{2}\|x-y\|^2, \qquad \forall x, y \in \operatorname{int} Q.$$

This assumption is required to simply relate the distance function with the dual norm of the (sub)gradients, and it can be relaxed.

Mirror descent algorithm. The idea is to replace the squared Euclidean norm in the subgradient method by an arbitrary Bregman divergence. We obtain the following algorithm, called *mirror descent*. We iterate, for $k \geq 0$:

$$\boxed{x_{k+1} \;=\; \operatorname*{argmin}_{y\in Q}\Big[\eta\langle f'(x_k), y - x_k\rangle + \beta_d(x_k;y)\Big]} \tag{4.88}$$

where $\eta > 0$ is a constant step-size, starting from x_0 defined by (4.87).

Assume for a moment that $Q \equiv \mathbb{R}^n$ (unconstrained minimization). Then, the stationary condition of one mirror descent step gives

$$\eta f'(x_k) + \nabla d(x_{k+1}) - \nabla d(x_k) \;=\; 0,$$

or, rearranging the terms,

$$\nabla d(x_{k+1}) \;=\; \nabla d(x_k) - \eta g_k, \quad \text{where} \quad g_k \;=\; f'(x_k) \;\in\; \partial f(x_k).$$

This formula explains the name of the method, as we perform the "descent update" $-\eta g_k$ in the dual (mirror) space, compared to the classic gradient descent update in the primal space.

Example 4.8.7. Let $d(x) := \frac{1}{2}\|x\|_2^2$. Then, $\beta_d(x;y) = \frac{1}{2}\|y-x\|^2$ and one step of the method is the classic subgradient step with projection:

$$x_{k+1} \;=\; \pi_Q(x_k - \eta f'(x_k)).$$

Example 4.8.8. Let $Q = \Delta_n$ and $d(x) = \sum\limits_{i=1}^n x^{(i)} \ln x^{(i)}$ be the negative entropy. Then, one step of the method is as follows:

$$x_{k+1} \;=\; \operatorname*{argmin}_{y\in\Delta_n}\Big\{\eta\langle g_k, y - x_k\rangle + \sum_{i=1}^n y^{(i)} \ln \tfrac{y^{(i)}}{x_k^{(i)}}\Big\}.$$

It can be written explicitly as the *multiplicative weight update* (see Exercise 4.9.1):

$$x_{k+1}^{(i)} \quad = \quad \frac{x_k^{(i)} \exp\bigl(-\eta g_k^{(i)}\bigr)}{\sum_{j=1}^n x_k^{(j)} \exp\bigl(-\eta g_k^{(j)}\bigr)}$$

Note that we do not need to perform projection to the simplex as point x_{k+1} already belongs to it, due to normalization.

Convergence analysis. By our main Lemma, we have, for any $y \in Q$:

$$\begin{aligned}
\beta_d(x_k; y) + \eta\langle g_k, y - x_k\rangle &\geq \beta_d(x_k; x_{k+1}) + \eta\langle g_k, x_{k+1} - x_k\rangle + \beta_d(x_{k+1}; y) \\
&\geq \tfrac{1}{2}\|x_k - x_{k+1}\|^2 - \eta\|g_k\|_*\|x_{k+1} - x_k\| + \beta_d(x_{k+1}; y) \\
&\geq \min_{t>0}\Bigl\{\tfrac{t^2}{2} - \eta\|g_k\|_* t\Bigr\} + \beta_d(x_{k+1}; y) \\
&= -\tfrac{\eta^2\|g_k\|_*^2}{2} + \beta_d(x_{k+1}; y) \\
&\geq -\tfrac{\eta^2 M^2}{2} + \beta_d(x_{k+1}; y).
\end{aligned}$$

Therefore, for one step of the method, we have the following inequality:

$$\tfrac{\eta^2 M^2}{2} + \beta_d(x_k; y) - \beta_d(x_{k+1}; y) \quad \geq \quad \eta\langle g_k, x_k - y\rangle.$$

Telescoping this inequality for the first $k \geq 1$ iterations, we obtain

$$\begin{aligned}
\eta k \cdot \tfrac{1}{k}\sum_{i=0}^{k-1}\langle g_i, x_i - y\rangle &\leq k\tfrac{\eta^2 M^2}{2} + \beta_d(x_0; y) - \beta_d(x_{k+1}; y) \\
&\leq k \cdot \tfrac{\eta^2 M^2}{2} + \beta_d(x_0; y).
\end{aligned}$$

Now, let us define

$$\boxed{\mathrm{Gap}_K \quad := \quad \max_{y \in Q} \tfrac{1}{K}\sum_{i=0}^{K-1}\langle g_i, x_i - y\rangle} \tag{4.89}$$

and

$$\boxed{D^2 := 2 \cdot \max_{y \in Q} \beta_d(x_0; y).}$$

This is a "diameter" of the set Q measure by our distance function.

We have proved the following bound for the new accuracy measure.

Theorem 4.8.9. *For any $\eta > 0$:*

$$\mathrm{Gap}_K \quad \leq \quad \tfrac{D^2}{2\eta K} + \tfrac{\eta M^2}{2}.$$

By choosing $\boxed{\eta := \frac{D}{M\sqrt{K}}}$ *we obtain*

$$\mathrm{Gap}_K \quad \leq \quad \tfrac{MD}{\sqrt{K}}. \tag{4.90}$$

We can compute the new accuracy measure (4.89) within iterations of our method, as it requires the minimization of a linear function over the set Q. This subproblem is easier that computing one step of the method (4.88). Thus, (4.89) is the accuracy certificate that we can use to stop our method:

$$\mathrm{Gap}_K \quad \leq \quad \varepsilon,$$

at it serves us an upper bound for the functional residual.

4.8.4 Accuracy Certificates

How to relate Gap_K to the functional residual?

- **Convex functions:**

$$\begin{aligned} \mathrm{Gap}_K &\geq \tfrac{1}{K} \sum_{i=0}^{K-1} \langle f'(x_i), x_k - x^\star \rangle \geq \tfrac{1}{K} \sum_{i=0}^{K-1} \Big[f(x_i) - f^\star \Big] \\ &\geq f(x_k) - f^\star, \end{aligned}$$

  where $x_K := \frac{1}{K} \sum_{i=0}^{K-1} x_i$.

- **Online optimization:** in online optimization, we have a stream of functions: a function $f_k(x)$ at iteration k, and we observe the corresponding subgradient $g_k := f_k'(x_k)$. Note that our analysis of mirror descent *did not use anything* about vectors g_k, and thus the result of Theorem 4.8.9 holds in this setting as well. In online optimization, the quantity Gap_K is usually called *regret*.

- **Min-Max structure.** As in the previous lecture, consider the following min-max problem:

$$f^\star = \min_{x \in Q} \Big[f(x) := \max_{u \in \Omega} F(x,u) \Big],$$

  where $\Omega \subset \mathbb{R}^m$ is a bounded convex set, and F is a continuous function, convex in x and concave in u.

  Denote $u(x) := \operatorname{argmax}_{u \in \Omega} F(x,u)$. Then,

$$\begin{aligned} f(x) &= F(x, u(x)), \\ f'(x) &= F'_x(x, u(x)). \end{aligned}$$

  This is the primal problem. The corresponding adjoint / dual problem is

$$\varphi_\star = \max_{u \in \Omega} \Big[\varphi(u) := \min_{x \in Q} F(x,u) \Big],$$

  which can potentially be much harder (or easier) to solve than the primal one. We observe that $f^\star \geq \varphi_\star$.

  Then,

$$\begin{aligned} \mathrm{Gap}_K &:= \max_{y \in Q} \tfrac{1}{K} \sum_{i=0}^{K-1} \langle f'(x_i), x_i - y \rangle = \max_{y \in Q} \tfrac{1}{K} \sum_{i=0}^{K-1} \langle F'_x(x, u(x)), x_i - y \rangle \\ &\geq \max_{y \in Q} \tfrac{1}{K} \sum_{i=0}^{K-1} [F(x_i, u(x_i)) - F(y, u(x_i))] = \max_{y \in Q} \tfrac{1}{K} \sum_{i=0}^{K-1} [f(x_i) - F(y, u(x_i))] \\ &\geq \max_{y \in Q} \Big[f(\bar{x}_K) - F(y, \bar{u}_K) \Big] = f(\bar{x}_K) - \varphi(\bar{u}_K), \end{aligned}$$

where in the last inequality we used convexity of f and concavity of F with respect to the second argument (Jensen's inequality), denoting the average primal point:

$$\bar{x}_K \;=\; \tfrac{1}{K}\sum_{i=0}^{K-1} x_i$$

and the average dual point:

$$\bar{u}_K \;=\; \tfrac{1}{K}\sum_{i=0}^{K-1} u(x_i). \tag{4.91}$$

Hence, we can ensure convergence not only in terms of the primal residual, but in terms of the dual residual as well:

$$\begin{aligned} \tfrac{MD}{\sqrt{K}} \;\overset{(4.90)}{\geq}\; & \mathrm{Gap}_K \;\geq\; f(\bar{x}_K) - \varphi(\bar{u}_K) \\ =\; & \underbrace{f(\bar{x}_K) - f^\star}_{\geq 0} \;+\; \underbrace{\varphi_\star - \varphi(\bar{u}_K)}_{\geq 0} \;+\; \underbrace{f^\star - \varphi_\star}_{\geq 0}. \end{aligned} \tag{4.92}$$

Note that our algorithm is initially designed to a *primal problem only*, but automatically solves the dual problem as well, where the dual solution at each iteration can be computed by formula (4.91).

It is remarkable that by setting $K \to \infty$ in (4.92), we obtain an *algorithmic proof* of *strong duality* for our min-max problem:

$$\boxed{f^\star \;=\; \varphi_\star}$$

4.9 Exercises

Exercise 4.9.1. Consider the negative entropy distance function,

$$d(x) \;=\; \sum_{i=1}^{n} x^{(i)} \ln x^{(i)}, \qquad x \in \Delta_n,$$

defined on the standard simplex $\Delta_n = \{x \in \mathbb{R}^n_+ \,:\, \langle e, x\rangle = 1\}$, where $e = (1, \ldots, 1)^\top \in \mathbb{R}^n$.

- Show that $x_0 = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)^\top \in \mathbb{R}^n$ is the minimum of d over Δ_n.
- Consider a step of the mirror descent algorithm:

$$x_+ \;=\; \operatorname*{argmin}_{y \in \Delta_n} \Big\{ \eta \langle g, y - x\rangle + \beta_d(x; y) \Big\},$$

where $x \in \Delta_n$ is a current point, $g \in \mathbb{R}^n$ is the direction (the gradient), $\eta > 0$ is a step-size, and

$$\beta_d(x; y) \;=\; d(y) - d(x) - \langle \nabla d(x), y - x\rangle$$

is the Bregman divergence associated with d. Prove the following explicit formula for x_+:

$$x_+^{(i)} \;=\; \frac{x^{(i)} \exp\left(-\eta g^{(i)}\right)}{\sum_{j=1}^n x^{(j)} \exp(-\eta g^{(j)})}, \qquad 1 \leq i \leq n.$$

Exercise 4.9.2. Consider the space of symmetric matrices, $\mathbb{S}^n = \{X \in \mathbb{R}^{n\times n} \ : \ X = X^\top\}$, and the following open *spectrahedron* set, which is a semidefinite generalization of the standard simplex:

$$Q \ = \ \Big\{X \ \succ \ 0 \ : \ \langle I, X\rangle = 1\Big\} \ \subset \ \mathbb{S}^n.$$

Recall that for $\mathbb{S}^n$ we use the standard inner product $\langle X, Y\rangle = \mathrm{tr}(XY)$, and I is the identity matrix.

For a symmetric matrix $X \in \mathbb{S}^n$, with a spectral decomposition $X = U\,\mathrm{Diag}(\lambda_1, \ldots, \lambda_n)U^\top$ we define the *matrix exponent* by

$$\exp(X) \ := \ U\,\mathrm{Diag}(e^{\lambda_1}, \ldots, e^{\lambda_n})U^\top$$

and, when $\lambda_1, \ldots, \lambda_n > 0$ (so $X \succ 0$ is positive definite), the *matrix logarithm* by

$$\ln X \ := \ U\,\mathrm{Diag}(\ln\lambda_1, \ldots, \ln\lambda_n)U^\top.$$

A natural distance function for spectahedron Q is the Von Neumann entropy:

$$d(X) \ = \ \mathrm{tr}(X \ln X). \tag{4.93}$$

- Derive the expression for the gradient $\nabla d(X)$ and the Bregman divergence $\beta_d(X; Y)$ associated with distance function (4.93).
- Compute an explicit formula for the matrix version of the mirror descent:

$$X_+ \ = \ \operatorname*{argmin}_{Y\in Q}\Big\{\eta\langle G, Y - X\rangle + \beta_d(X; Y)\Big\}, \qquad X \in Q, \quad G \in \mathbb{S}^n, \quad \eta > 0.$$

Exercise 4.9.3. A general technique to solve non-smooth optimization problems is called *smoothing*. Consider the following non-smooth problem in a min-max form:

$$\min_{x\in\mathbb{R}^n}\Big\{ \ f(x) \ := \ \max_{1\le i\le m}\big[\langle a_i, x\rangle + b_i\big] \ = \ \max_{u\in\Delta_m}\Big[\ \sum_{i=1}^m u^{(i)}[\langle a_i, x\rangle + b_i] \ \Big] \ \Big\}, \tag{4.94}$$

where Δ_m is the standard simplex, and $a_i \in \mathbb{R}^n, b_i \in \mathbb{R}$ for $1 \le i \le m$ are given data.

We consider the *entropy smoothing* of f, that is, the following modified objective, for $\mu > 0$:

$$f_\mu(x) \ := \ \max_{u\in\Delta_m}\Big[\ \sum_{i=1}^m u^{(i)}[\langle a_i, x\rangle + b_i] - \mu d(u) \ \Big], \tag{4.95}$$

where $d(u) := \sum\limits_{i=1}^m u^{(i)} \ln u^{(i)}$.

- Show that, for all $u \in \Delta_m$, we have

$$-\ln m \ \le \ d(u) \ \le \ 0. \tag{4.96}$$

- Using (4.96), show the following bound on the approximation of $f(\cdot)$ by its smoothing $f_\mu(\cdot)$:

$$f_\mu(x) - \mu\ln m \ \le \ f(x) \ \le \ f_\mu(x), \qquad \forall x \in \mathbb{R}^n. \tag{4.97}$$

- Derive an explicit formula for $f_\mu(x)$ and its gradient $\nabla f_\mu(x)$.
- Show that $f_\mu(x)$ has a Lipschitz continuous gradient and derive the corresponding Lipschitz constant L_μ. How does the Lipschitz constant L_μ depend on $\mu > 0$?

- Consider applying the fast gradient method to minimizing $f_\mu(x)$, $x \in \mathbb{R}^n$. Assume that after $k \geq 1$ iterations we obtained a point x_k with the following guarantee, for some $\delta > 0$:

$$f_\mu(x_k) - f_\mu^\star \quad \leq \quad \delta.$$

Using (4.97), show how to choose parameters μ and δ in order to ensure an ε-accuracy for the original problem (4.94):

$$f(x_k) - f^\star \quad \leq \quad \varepsilon.$$

What is the iteration complexity k of the resulting method in terms of dependence on ε to find such a point? Compare it with that of the subgradient method as applied to the original function (4.94) directly.

Literature

The general ellipsoid method, the mirror descent, and the lower complexity bounds for optimization methods were developed in 1976 by Arkadi S. Nemirovski and David B. Yudin in their seminal book [29]. See also the lecture notes [27] for additional reading on these topics.

More advanced lower complexity bounds which incorporates different problem geometries (beyond the Euclidean) and various degrees of smoothness into the complexity bound are available in [18].

The subgradient method was discovered by Naum Z. Shor in 1962 [44]. Another powerful selection of stepsizes for the subgradient method includes Polyak's stepsizes [37, 19] and their recent modification suitable for constrained and composite optimization [32]. See also Section 7 in [27] and Section 3.2 in [31] for the design of subgradient methods for constrained problems with functional inequalities. An advanced analysis of the convergence rate for the last iterate x_k of the subgradient method, instead of $\bar{x}_k$, was developed recently in [47].

Our analysis of adaptive stepsizes for the subgradient method is based on recent work [43], which demonstrates that they work universally well for stochastic problems with varying degrees of smoothness, biased oracles, composite, and acceleration methods, while automatically supporting variance reduction. See also the references therein for additional reading. The adaptive stepsizes presented here are also referred to as *AdaGrad stepsizes*, as they resemble the popular AdaGrad algorithm (Adaptive Subgradient Method) [12] from machine learning.

5. Second-Order Methods

In the final part of the course, we study *second-order optimization* algorithms. These methods utilize the Hessian of the objective (or its approximation) to better capture the geometry of the problem. While second-order algorithms are typically more computationally expensive, they often possess superior convergence rates compared to first-order ones. Therefore, selecting an appropriate optimization algorithm requires a trade-off between iteration complexity and per-iteration cost; this choice depends on problem properties (such as dimension, sparsity, etc.) and the desired target accuracy.

5.1 Introduction . . . 125
5.1.1 Quadratic Taylor Approximation: Newton's Step . . . 127
5.1.2 Affine Invariance . . . 128
5.2 Self-Concordant Functions and Local Convergence of Newton's Method . . . 129
5.2.1 Definition and Basic Properties . . . 129
5.2.2 Self-Concordant Analysis . . . 133
5.2.3 Local Convergence of Newton's Method . . . 135
5.3 Interior-Point Method . . . 139
5.3.1 Self-Concordant Barriers . . . 140
5.3.2 Path-Following Scheme . . . 143
5.3.3 Interior-Point Algorithm . . . 146
5.4 Cubic Regularization of Newton's Method . . . 147
5.4.1 Cubic Regularization of Quadratic Model . . . 149
5.4.2 Local Quadratic Convergence for Strongly Convex Functions . . . 151
5.4.3 Non-Convex Quadratics and Strong Duality . . . 152
5.4.4 Solving Cubic Subproblem in Practice . . . 154
5.4.5 Main Inequalities for Cubic Newton Step . . . 154
5.4.6 Convergence to Second-Order Stationary Point . . . 156
5.5 Quasi-Self-Concordant Functions and Gradient Regularization . . . 156
5.5.1 Motivational Example: Smoothness of Loss Functions . . . 156
5.5.2 Quasi-Self-Concordant Functions . . . 158
5.5.3 Gradient Regularization of Newton's Method . . . 160
5.5.4 Global Linear Rate . . . 161
5.6 Contracting-Point Acceleration . . . 164
5.6.1 Contracting-Point Scheme . . . 164
5.6.2 Example: Acceleration of First-Order Methods . . . 166
5.7 Exercises . . . 168

5.1 Introduction

We will be solving an optimization problem of the form,

$$\min_{x \in Q} f(x), \tag{5.1}$$

and distinguish between two principal cases, for each developing corresponding second-order algorithms:

1. *Unconstrained optimization.* $Q = \mathbb{R}^n$, where $\mathbb{R}^n$ is the target vector space. Note that in this setting, the vector space can be readily replaced by an *affine space*, $Q = \{x \in \mathbb{R}^n \ : \ Ax = b\}$, which is useful if we need to cover the affine equality constraints.

   First-order methods. We allow to compute $f(x)$ and $\nabla f(x)$ and perform very simple operations, such as *summation of two vectors*:

   $$x^+ \quad = \quad x - \alpha \nabla f(x). \tag{5.2}$$

   Or, choosing a distance function d (which has to be *simple*), the mirror descent step:

   $$\nabla d(x^+) \quad = \quad \nabla d(x) - \alpha \nabla f(x).$$

   These operations can be generalized further by the framework of composite optimization, e.g. treating an additive composite regularizer $\psi(y)$ to the objective in (5.1) by steps:

   $$x^+ \quad = \quad \operatorname*{argmin}_y \Big[\ \langle \nabla f(x), y - x \rangle + \beta_d(x; y) + \psi(y) \ \Big],$$

   where $\beta_d(x; y)$ is the Bregman divergence. However, all operations remain simple and we typically hope to obtain an explicit formula for x^+.

   Second-order methods. The key assumption of second-order algorithms is

   $$\textit{We can solve linear systems}\text{: } Hx = g.$$

   Indeed, thanks to advances in linear algebra and the rapid development of numerical packages, we can use efficient linear algebra techniques (LU, QR, Cholesky, tridiagonal decomposition, SVD, ...). Note that efficient packages, such as LAPACK (integrated in Python and other languages), are able to solve linear systems of dimension $n \approx 2 \cdot 10^4$ in under a minute on a standard laptop (see Fig. 5.1). Along with time complexity, the main bottleneck for larger n becomes memory constraints (a dense $n \times n$ matrix of `float64` values with $n = 28000$ requires about 6 GB of RAM only to store it).

   An alternative, highly effective approach to solving a symmetric positive-definite linear system is to run a first-order method on a quadratic objective (e.g., the conjugate gradient method, or, in the case of constraints or non-smooth regularizers, the fast gradient method) — this scales well to very large values of n, as the first-order method requires only a procedure to compute Hessian-vector products, and we can perform only a few iterations of the solver to obtain an approximate solution. Further performance gains can be achieved if the linear system is *sparse* or *low-rank*.

   Thus, the main idea behind second-order algorithms is to rely more on efficient linear algebra rather than solely on the summation of vectors, as in first-order methods. We will also assume access to a second-order local oracle, computing $f(x)$, $\nabla f(x)$, and the Hessian $\nabla^2 f(x)$.

   We will study a few variants of Newton's method that achieve *provably better global rates* than those of the first-order methods, for convex and non-convex unconstrained optimization.

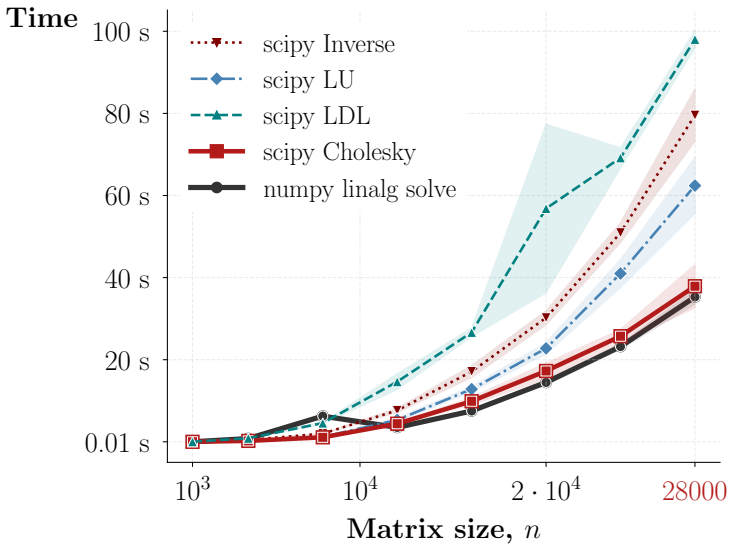

Figure 5.1: *Scaling of linear system solvers.* Solving a symmetric positive definite dense linear system in Python on a MacBook Pro 2024, using the following functions:
1. Inverting the matrix with `scipy.linalg.inv`
2. General system solver (LU decomposition) with `scipy.linalg.solve(assume_a='gen')`
3. Symmetric system solver (LDL decomposition) with `scipy.linalg.solve(assume_a='sym')`
4. Cholesky decomposition with `scipy.linalg.solve(assume_a='pos')`
5. Numpy default solver with `np.linalg.solve`.

2. *Structured constrained optimization.* In this case, $Q \subset \mathbb{R}^n$ is a complicated convex set with a specific structure, while the objective function is linear, $f(x) = \langle c, x \rangle$.

   Our main assumption is the possibility of constructing a *self-concordant barrier* F for the set Q (see Fig. 5.2), which we study in detail in the following lectures. It appears that such barriers can be constructed for practically all known classes of convex optimization problems (e.g., linear programming, quadratic programming, semidefinite programming, etc.), and, moreover, they can be minimized very efficiently with second-order methods, achieving polynomial-time complexity and excellent practical performance.

   This class of algorithms is known as *interior-point methods*, which places second-order algorithms in a unique position as an essential tool for the minimization of self-concordant barriers, and for the overall success of this framework.

5.1.1 Quadratic Taylor Approximation: Newton's Step

The idea of the classical Newton method is to use second-order (quadratic) Taylor expansion of the objective $f(y)$, around the current point $x \in \mathbb{R}^n$:

$$f(y) \;=\; f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{1}{2} \langle \nabla^2 f(x)(y - x), y - x \rangle + o(\|y - x\|^2).$$

Then, we choose the next point x^+ as a minimum of the second-order model:

$$x^+ \;=\; \operatorname*{argmin}_{y \in \mathbb{R}^n} \Big[\langle \nabla f(x), y - x \rangle + \tfrac{1}{2} \langle \nabla^2 f(x)(y - x), y - x \rangle \Big]. \tag{5.3}$$

Note that the minimum (5.3) might not exist at all, or if exists, it might not be unique.

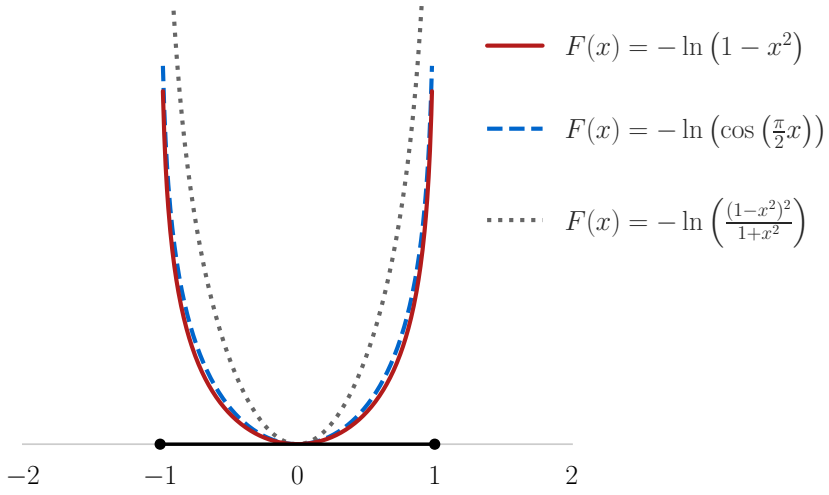

Figure 5.2: Self-concordant barriers for the segment $[-1, 1]$. See Section 5.3.1 for the definition.

For now, and for the next few lectures we might assume that $\nabla^2 f(x) \succ 0$, and then x^+ exists and unique as the minimizer of strongly convex model in (5.3).

The optimality condition for (5.3) states that x^+ should satisfy the linear equation:

$$\nabla f(x) + \nabla^2 f(x)(x^+ - x) \quad = \quad 0.$$

or, rearranging the terms, we obtain the classic Newton's step:

$$x^+ \quad = \quad x - \nabla^2 f(x)^{-1} \nabla f(x). \tag{5.4}$$

5.1.2 Affine Invariance

Let $x := Ay + b$ for an invertible matrix A and define new function

$$F(y) \quad = \quad f(Ay + b) \quad = \quad f(x).$$

Consider Newton's step (5.4) for the original objective, and Newton's step for the new function, from some point $y \in \mathbb{R}^n$:

$$y^+ \quad = \quad y - \nabla^2 F(y)^{-1} \nabla F(y)$$

It appears that Newton's step is *affine-invariant*:

Proposition 5.1.1. *Let $x = Ay + b$. Then, $x^+ = Ay^+ + b$.*

Proof. Note that

$$\nabla F(y) \quad = \quad A^\top \nabla f(Ay + b) \quad = \quad A^\top \nabla f(x),$$

$$\nabla^2 F(y) \quad = \quad A^\top \nabla^2 f(Ay + b) A \quad = \quad A^\top \nabla^2 f(x) A.$$

Then,

$$\begin{aligned} y^+ \quad &= \quad y - \nabla^2 F(y)^{-1} \nabla F(y) \\ &= \quad y - A^{-1} \nabla^2 f(x)^{-1} A^{-\top} A^\top \nabla f(x) \\ &= \quad y - A^{-1} \nabla^2 f(x)^{-1} \nabla f(x). \end{aligned}$$

Therefore,

$$Ay^+ + b \;=\; Ay + b - \nabla^2 f(x)^{-1}\nabla f(x) \;=\; x^+.$$

□

Hence, Newton's method is independent of the choice of coordinate system in $\mathbb{R}^n$. This property can also be seen directly from (5.3), as the value of Taylor polynomial does not depend on the actual choice of the inner product $\langle \cdot, \cdot \rangle$. If we change the coordinate system and run the method from the corresponding initial point, the result remains the same. In other words, classic Newton's method cannot be "accelerated" by finding a better coordinate system, unlike the gradient method.

Note however that the basic gradient step (5.2) is invariant to shifts and orthogonal transformations:

Proposition 5.1.2. *Let $x = Uy + b$, where $UU^\top = I$. Consider $F(y) = f(Uy+b)$ and two gradient steps $y^+ = y - \alpha\nabla F(y)$ and $x^+ = x - \alpha\nabla f(x)$. Then,*

$$x^+ \;=\; Uy^+ + b$$

Exercise 5.1.1. Prove this proposition. Show that the gradient method is not affine-invariant.

5.2 Self-Concordant Functions and Local Convergence of Newton's Method

5.2.1 Definition and Basic Properties

The main result about Newton's method is its *local quadratic convergence*: when the point is sufficiently close to the optimum $x \approx x^\star$, the method doubles known digits of the solution with every step. We will prove this result using a modern affine-invariant analysis.

We consider differentiable function $f : Q \to \mathbb{R}$, where $Q \subseteq \mathbb{R}^n$ is an open convex set. We assume that $\nabla^2 f(x) \succ 0$ everywhere on Q, so f is strictly convex.

Local Euclidean structure. The main component in the definition of self-concordant functions is the notion of the *local norm*. Note that at every point $x \in Q$, the Hessian of f defines the Euclidean structure on $\mathbb{R}^n$:

$$\langle u, v \rangle_x \;:=\; D^2 f(x)[u, v] \;=\; \langle \nabla^2 f(x) u, v \rangle, \tag{5.5}$$

where in the right hand side we use the standard inner product. In (5.5), the matrix $\nabla^2 f(x)$ depends on $\langle \cdot, \cdot \rangle$, while $D^2 f(x)$ and, consequently, $\langle \cdot, \cdot \rangle_x$ *do not depend on the coordinate system.*

Correspondingly, we can define the local norm, which is the norm generated by the Hessian of the objective:

$$\|h\|_x \;:=\; \langle h, h \rangle_x^{1/2} \;=\; \langle \nabla^2 f(x) h, h \rangle^{1/2}, \qquad x \in Q, h \in \mathbb{R}^n. \tag{5.6}$$

We will use this norm to characterize smoothness of the objective f.

Third derivative. To understand the behavior of (5.6), we fix a direction $h \in \mathbb{R}^n$ and study the quadratic form

$$g(x) \;=\; \|h\|_x^2 \;=\; \langle \nabla^2 f(x)h, h\rangle \;>\; 0, \tag{5.7}$$

as a function of x. By continuity, it is clear that

$$\text{when} \quad y \;\approx\; x \quad \text{then} \quad \nabla^2 f(y) \;\approx\; \nabla^2 f(x). \tag{5.8}$$

Our goal is to provide a *quantitative* and *affine-invariant* characterization of (5.8). For an arbitrary perturbation $u \in \mathbb{R}^n$, we consider

$$\varphi(t) \;=\; \|h\|_{x+tu}^2 \;=\; \langle \nabla^2 f(x+tu)h, h\rangle,$$

for small $t > 0$. Note that φ is a scalar function. Its derivative at zero,

$$\varphi'(0) \;=\; D^3 f(x)[h, h, u] \;\in\; \mathbb{R},$$

shows how fast the local norm (5.7) changes at x. It is known that $D^3 f(x)$ is a trilinear symmetric form, as soon as f is sufficiently differentiable.

Definition. We say that a function $f : Q \to \mathbb{R}$ is *self-concordant* with constant $M \geq 0$, if

$$D^3 f(x)[h, h, u] \;\leq\; M\|h\|_x^2 \|u\|_x, \qquad \forall h, u \in \mathbb{R}^n, x \in Q. \tag{5.9}$$

This inequality means that the third derivative is bounded by constant $M \geq 0$, but the "boundedness" is measured by the local norm at the same point x (which provides the *concordance*). As a result, the parameter M does not depend on the coordinate system.

If we subsitute $u := h$ (the same direction) into (5.9), we obtain the bound:

$$|D^3 f(x)[h, h, h]| \;\leq\; M\|h\|_x^3 \;=\; M\langle \nabla^2 f(x)h, h\rangle^{3/2}, \qquad \forall h \in \mathbb{R}^n, x \in Q. \tag{5.10}$$

However, it appears that the reverse implication holds as well! If the last inequality is satisfied, then (5.9) also holds. So, inequality (5.10) can be used as a definition of self-concordance. It is easier to check whether a function is self-concordant with the latter inequality. At the same time, our original definition (5.9) is better suitable for an analysis.

The equivalence of two definitions is a consequence of the following fact (see also Exercise 5.2.2).

Lemma 5.2.1. *Let $T[\cdot, \cdot, \cdot]$ be a trilinear symmetric form. Denote by $S = \{h : \langle h, h\rangle = 1\}$ the standard Euclidean unit sphere. Then,*

$$\max_{h,u \in S} T[h, h, u] \;=\; \max_{h \in S} T[h, h, h]. \tag{5.11}$$

Proof. Let $h^\star, u^\star \in S$ be any pair of maximizers (which clearly exists, as T is a continuous function and S is a compact set):

$$T^\star \;=\; \max_{h,u \in S} T[h, h, u] \;=\; T[h^\star, h^\star, u^\star]. \tag{5.12}$$

Denote $\theta = \langle h^\star, u^\star\rangle$. Without loss of generality we can assume $\theta \geq 0$. If $\theta = 1$ then (5.11) is proved. Hence, we assume that $0 \leq \theta < 1$.

Consider the linear form, $\langle \ell, u\rangle \equiv T[h^\star, h^\star, u]$. Thus, we have $\langle \ell, u\rangle \leq \langle \ell, u^\star\rangle = T^\star$, for all $u \in S$. By Cauchy-Schwartz inequality we conclude that $\ell = T^\star u^\star$. Thus, we get

$$T[h^\star, h^\star, h^\star] \;=\; \langle \ell, h^\star\rangle \;=\; \theta T^\star. \tag{5.13}$$

Similarly, consider the symmetric matrix A defined by the equation $\langle Ah, h\rangle \equiv T[u^\star, h, h]$, and we have $\langle Ah, h\rangle \le \langle Ah^\star, h^\star\rangle = T^\star$, for all $h \in S$. Thus, by the spectral theorem, we conclude that $h^\star$ is the eigenvector of the matrix corresponding to the maximal eigenvalue: $Ah^\star = T^\star h^\star$, and, therefore,

$$T[u^\star, u^\star, h^\star] \;=\; \langle Ah^\star, u^\star\rangle \;=\; \theta T^\star. \tag{5.14}$$

Denote $v^\star := \frac{h^\star + u^\star}{\|h^\star + u^\star\|_2} \in S$ and note that $\|u^\star + h^\star\|_2^2 \;=\; 2(1+\theta)$. Then,

$$\begin{array}{rcl} T[v^\star, v^\star, h^\star] & = & \frac{1}{2(1+\theta)}\Big(T[h^\star, h^\star, h^\star] + T[u^\star, u^\star, h^\star] + 2T[h^\star, h^\star, u^\star]\Big) \\ & \overset{(5.12),(5.13),(5.14)}{=} & \frac{2\theta+2}{2(1+\theta)}T^\star \;=\; T^\star. \end{array} \tag{5.15}$$

Hence, the new triplet $(v^\star, v^\star, h^\star)$ preserves the optimal value of T, while *shrinking the distance*:

$$\tfrac{1}{2}\|v^\star - h^\star\|_2^2 \;=\; \Big(1 - \sqrt{\tfrac{1+\theta}{2}}\Big) \;\overset{(*)}{\le}\; \Big(1 - \tfrac{1}{\sqrt{2}}\Big)\cdot(1-\theta) \;=\; \Big(1 - \tfrac{1}{\sqrt{2}}\Big)\tfrac{1}{2}\|h^\star - u^\star\|_2^2, \tag{5.16}$$

where $(*)$ follows from convexity of the function $\varphi(\theta) = 1 - \sqrt{\frac{1+\theta}{2}}$ for $0 \le \theta \le 1$.

Finally, consider the set of all maximizers of $T[h, h, u]$, which is a compact nonempty set:

$$\Omega \;=\; \Big\{(h,u) \in S \times S \;:\; T[h,h,u] \;=\; T^\star\Big\},$$

and a continuous function $\rho(h,u) \;=\; \|h - u\|_2$. Let $(h^\star, u^\star)$ be the minimizer of ρ over Ω. If $h^\star \ne u^\star$, by the previous reasoning, we can find a pair $(v^\star, h^\star)$ with a strictly smaller value of ρ, which contradicts that $(h^\star, u^\star)$ is the minimizer. Hence, $h^\star = u^\star$. □

Examples. First, let us start with the following three basic examples that show that the class of self-concordant functions is not empty and actually quite broad.

1. *Convex quadratic functions:* $f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$ is self-concordant with $\boxed{M = 0}$.

2. *Negative logarithm:* $f(x) = -\ln x$, for $x > 0$. Indeed,

   $$f'(x) \;=\; -\tfrac{1}{x}, \quad f''(x) \;=\; \tfrac{1}{x^2}, \quad f'''(x) \;=\; -\tfrac{2}{x^3}.$$

   Hence,

   $$|f'''(x)| \;=\; 2(f''(x))^{3/2},$$

   and we conclude that f is self-concordant with $\boxed{M = 2}$. Since logarithmic barriers play the key role in the theory of interior-point methods, self-concordant functions with $M = 2$ are often called *standard self-concordant*.

3. *Strongly convex functions with Lipschitz Hessian.* Let f have Lipschitz Hessian with constant $L > 0$ w.r.t a fixed norm, and let f be strongly convex with constant $\mu > 0$. Thus, for all $x, y \in Q$ and $h \in \mathbb{R}^n$:

   $$\|\nabla^2 f(x) - \nabla^2 f(y)\| \;\le\; L\|x - y\| \qquad \text{and} \qquad \mu\|h\|^2 \;\le\; \|h\|_x^2.$$

   Then,

   $$D^3 f(x)[h,h,h] \;\le\; L\|h\|^3 \;\le\; \tfrac{L}{\mu^{3/2}}\|h\|_x^3.$$

   Hence, we can take $M = \frac{L}{\mu^{3/2}}$. However, note that $L_{\|\cdot\|}$ and $\mu_{\|\cdot\|}$ depend on the choice of the norm $\|\cdot\|$, while M does not. Thus,

   $$\boxed{M \;\le\; \inf_{\|\cdot\|}\Big(\tfrac{L_{\|\cdot\|}}{\mu_{\|\cdot\|}^{3/2}}\Big).}$$

Logarithmic barrier for semidefinite cone. The fact that $f(x) = -\ln x$, for $x > 0$, is self-concordant can be generalized to the cone of positive-definite symmetric matrices,

$$\mathbb{S}^n_+ \;=\; \left\{ X \in \mathbb{R}^{n\times n} \;:\; X = X^\top \succeq 0 \right\} \;\subset\; \mathbb{S}^n.$$

Example 5.2.2. Define, for $X \in \operatorname{int}\mathbb{S}^n_+$, the logarithmic barrier:

$$f(X) \;=\; -\ln\det X \;=\; -\sum_{i=1}^n \ln\lambda_i(X). \tag{5.17}$$

Then,

$$\begin{aligned}
Df(X)[H] &= -\operatorname{tr}(X^{-1}H) \qquad \Rightarrow \qquad \nabla f(X) = -X^{-1},\\
D^2f(X)[H,H] &= \operatorname{tr}(X^{-1}HX^{-1}H) = \operatorname{tr}(S^2), \qquad \text{where} \qquad S = X^{-1/2}HX^{-1/2} \in \mathbb{S}^n.\\
D^3f(X)[H,H,H] &= -2\operatorname{tr}(X^{-1}HX^{-1}HX^{-1}H) = -2\operatorname{tr}(S^3).
\end{aligned}$$

It remains to notice that

$$\operatorname{tr}(S^3) \;=\; \sum_{i=1}^n \lambda_i(S)^3 \;\le\; \sum_{i=1}^n |\lambda_i(S)|^3 \;\le\; \Big(\sum_{i=1}^n |\lambda_i(S)|^2\Big)^{3/2} \;=\; \Big(\operatorname{tr}(S^2)\Big)^{3/2},$$

where we used the inequality $\|\cdot\|_3 \le \|\cdot\|_2$. Hence, $f(X)$ is self-concordant with constant $\boxed{M = 2}$.

Summation of self-concordant functions. For a sum of m functions:

$$f(x) \;=\; \sum_{i=1}^m f_i(x),$$

where each f_i, $1 \le i \le m$ is self-concordant with constant $M_i \ge 0$, we have

$$\begin{aligned}
D^3f(x)[u]^3 &= \sum_{i=1}^m D^3f_i(x)[u]^3 \overset{(5.10)}{\le} \sum_{i=1}^m M_i\Big(D^2f_i(x)[u]^2\Big)^{3/2}\\
&\le \max_{1\le i\le m} M_i \cdot \sum_{i=1}^m \Big(D^2f_i(x)[u]^2\Big)^{3/2} \le \max_{1\le i\le m} M_i \cdot \Big(\sum_{i=1}^m D^2f_i(x)[u]^2\Big)^{3/2}\\
&= \max_{1\le i\le m} M_i \cdot \Big(D^2f(x)[u]^2\Big)^{3/2},
\end{aligned}$$

where in the last inequality we used that $\|\cdot\|_{3/2} \le \|\cdot\|_1$. Thus, f is self-concordant with constant

$$\boxed{M \;=\; \max_{1\le i\le m} M_i.}$$

Example 5.2.3. The logarithmic barrier for $\mathbb{R}^n_{>0}$,

$$f(x) \;=\; -\sum_{i=1}^n \ln x^{(i)}, \tag{5.18}$$

is self-concordant with constant $\boxed{M = 2}$. Note that (5.18) can be viewed as a restriction of the logarithmic barrier (5.17) for the cone of positive definite matrices $\mathbb{S}^n_{\succ 0}$ onto the subset of diagonal matrices with positive entries, that is isomorphic to $\mathbb{R}^n_{>0}$.

Affine restrictions and affine substitutions do not affect self-concordance, as to show in the following exercise.

Exercise 5.2.1. Let $g(y) = f(Ay+b)$, where $A \in \mathbb{R}^{n\times m}$ and $b \in \mathbb{R}^n$. Show that g is self-concordant with the same constant $\boxed{M_g = M_f}$.

Exercise 5.2.2. Using affine-invariance and Lemma 5.2.1, show that the definition of self-concordance along one direction (5.10) implies the definition along two arbitrary directions (5.9).

Exercise 5.2.3. Let $g(x) = cf(x)$, for $c > 0$. What will be the constant of self-concordance M_g for g? Show that for $M_f > 0$, we can always choose c such that $M_g = 2$, so any self-concordant function can be made "standard self-concordant" after an appropriate rescaling.

Example 5.2.4. The logarithmic barrier for the polyhedron $\{\langle a_1, x\rangle < b_1, \dots, \langle a_m, x\rangle < b_m\}$:

$$f(x) = -\sum_{i=1}^{m} \ln(b_i - \langle a_i, x\rangle)$$

is self-concordant with constant $\boxed{M = 2}$.

5.2.2 Self-Concordant Analysis

Main lemma. Our goal is to compare two Hessians, $\nabla^2 f(x)$ and $\nabla^2 f(y)$, when the points are close to each other, $x \approx y$. The following lemma is the main consequence of self-concordance, which, in fact, can be substituted for its definition (see [40]). It underpins most of the other important results about self-concordant functions.

Lemma 5.2.5. *Let $x, y \in Q$ and denote $r := \|y - x\|_x$. Assume that x and y are close:*

$$r \;<\; \tfrac{2}{M}. \tag{5.19}$$

Then,

$$(1 - \tfrac{M}{2}r)^2 \nabla^2 f(x) \;\preceq\; \nabla^2 f(y) \;\preceq\; \tfrac{1}{(1-\frac{M}{2}r)^2}\nabla^2 f(x). \tag{5.20}$$

Proof. First, we prove (5.20) along direction $h := y - x$, that is

$$(1 - \tfrac{M}{2}r)r \;\le\; \|y - x\|_y \;\le\; \tfrac{r}{(1-\frac{M}{2}r)}. \tag{5.21}$$

For that, we consider a function

$$\varphi(t) \;:=\; D^2 f(x + t(y - x))[y - x]^2 \;=\; \|y - x\|^2_{x+t(y-x)}.$$

We have

$$|\varphi'(t)| \;=\; D^3 f(x + t(y - x))[y - x]^3 \;\le\; M\varphi(t)^{3/2}. \tag{5.22}$$

Hence,

$$\left|\tfrac{d}{dt}\left[\tfrac{-2}{\sqrt{\varphi(t)}}\right]\right| \;=\; \tfrac{|\varphi'(t)|}{\varphi(t)^{3/2}} \overset{(5.22)}{\le} M. \tag{5.23}$$

Therefore, by the fundamental theorem of calculus, for $0 \le t \le 1$:

$$\left|\tfrac{2}{\sqrt{\varphi(t)}} - \tfrac{2}{\sqrt{\varphi(0)}}\right| \;=\; \left|\int_0^t \tfrac{d}{d\tau}\tfrac{-2}{\sqrt{\varphi(\tau)}}\right| \;\le\; \int_0^t \left|\tfrac{d}{d\tau}\tfrac{-2}{\sqrt{\varphi(\tau)}}\right| \overset{(5.23)}{\le} tM. \tag{5.24}$$

Substituting $t = 1$ and using the definition of φ, we immediately obtain:

$$\Big|\tfrac{1}{\|y-x\|_y} - \tfrac{1}{\|y-x\|_x}\Big| \;\leq\; \tfrac{M}{2},$$

or, the corresponding pair of inequalities, the first one:

$$\tfrac{1}{\|y-x\|_y} \;\leq\; \tfrac{1}{\|y-x\|_x} + \tfrac{M}{2} \;=\; \tfrac{1}{r} + \tfrac{M}{2} \;=\; \tfrac{1+\frac{M}{2}r}{r} \;\leq\; \tfrac{1}{r(1-\frac{M}{2}r)}, \tag{5.25}$$

where we used that $(1-t)(1+t) = 1-t^2 \leq 1$ for $t := \frac{M}{2}r \overset{(5.19)}{<} 1$, and the second one:

$$\tfrac{1}{\|y-x\|_y} \;\geq\; \tfrac{1}{\|y-x\|_x} - \tfrac{M}{2} \;=\; \tfrac{1}{r} - \tfrac{M}{2} \;=\; \tfrac{1-\frac{M}{2}r}{r}. \tag{5.26}$$

Inequalities (5.25) and (5.26) together give (5.21).

Note that from (5.24), we also have, for $0 \leq t \leq 1$:

$$\tfrac{1}{\sqrt{\varphi(t)}} \;\geq\; \tfrac{1}{\sqrt{\varphi(0)}} - \tfrac{tM}{2} \;=\; \tfrac{1-tMr/2}{r} \qquad \Leftrightarrow \qquad \sqrt{\varphi(t)} \;\leq\; \tfrac{r}{1-tMr/2}. \tag{5.27}$$

Finally, to prove (5.20) along arbitrary direction h, we denote

$$\psi(t) \;:=\; D^2 f(x+t(y-x))[h]^2 \;=\; \|h\|^2_{x+t(y-x)}.$$

Differentiating it gives

$$\begin{aligned} |\psi'(t)| \;&=\; |D^3 f(x+t(y-x))[h,h,y-x]| \;\overset{(??)}{\leq}\; M\|h\|_{x+t(y-x)}\|y-x\|_{x+t(y-x)} \\ &=\; M\psi(t)\sqrt{\varphi(t)} \;\overset{(5.27)}{\leq}\; \tfrac{Mr}{1-tMr/2}\psi(t). \end{aligned} \tag{5.28}$$

Thus,

$$\Big|\tfrac{d}{dt}\ln\psi(t)\Big| \;=\; \Big|\tfrac{\psi'(t)}{\psi(t)}\Big| \;\overset{(5.28)}{\leq}\; \tfrac{Mr}{1-tMr/2} \;=\; 2\tfrac{d}{dt}\ln\Big(1 - t\tfrac{Mr}{2}\Big). \tag{5.29}$$

Integrating this inequality, we conclude:

$$\Big|\ln\tfrac{\|h\|_y}{\|h\|_x}\Big| \;=\; \Big|\ln\psi(1) - \ln\psi(0)\Big| \;\leq\; -2\ln(1-\tfrac{Mr}{2}),$$

or, equivalently,

$$\ln\Big(\,\big[1-\tfrac{Mr}{2}\big]^2\,\Big) \;\leq\; \ln\tfrac{\|h\|_y}{\|h\|_x} \;\leq\; \ln\Big(\,\big[1-\tfrac{Mr}{2}\big]^{-2}\,\Big)$$

which completes the proof. □

Dikin's ellipsoid. The previous lemma states that as long as $y \in Q$ belongs to *Dikin's ellipsoid*:

$$y \;\in\; \mathcal{E}_x \;:=\; \Big\{\, y \in \mathbb{R}^n \;:\; \|y-x\|_x < \tfrac{2}{M} \,\Big\}, \tag{5.30}$$

the Hessians $\nabla^2 f(y)$ and $\nabla^2 f(x)$ are *comparable* or *stable*: that is (5.20) holds.

Note that, in general, it might happen that $\mathcal{E}_x \not\subseteq Q$, as Q can be *any open convex set* on which f is defined. However, instead of taking an arbitrary set, it is natural to assume that Q is "as large as possible". Specifically, we define Q as the "domain" of f, meaning that the following property holds:

For any $y^\star \in \partial Q$ and any sequence $\{y_k\}_{k\geq 0}$ of points in Q such that $y_k \to y^\star$, we have

$$f(y_k) \;\to\; +\infty. \tag{5.31}$$

Thus, under condition (5.31), we assume either that $Q = \mathbb{R}^n$ is the whole space, or that f blows up at the boundary.

With this condition, we can show that self-concordance implies that with every point $x \in Q$, *the entire Dikin's ellipsoid* (5.30) belongs to it.

Proposition 5.2.6. *Let $Q \subseteq \mathbb{R}^n$ be an open set such that condition* (5.31) *holds for $f : Q \to \mathbb{R}$. Assume that f is self-concordant with constant $M > 0$. Then, for every $x \in Q$:*

$$\mathcal{E}_x \;\subseteq\; Q. \tag{5.32}$$

Proof. Assume (5.32) does not hold. Thus, there exists $y \in \mathcal{E}_x$ such that $y \notin Q$. Denote

$$t^\star \;=\; \inf\Big\{\, t \in (0,1] \;:\; x + t(y-x) \notin Q \,\Big\}.$$

By definition of $t^\star$, for any $\epsilon > 0$ there exists $y(\epsilon) = x + (t^\star - \epsilon)(y-x) \in Q$. Therefore, selecting a sequence of small positive $\epsilon_k \to 0$, we generate the sequence of points $y_k = y(\epsilon_k) \in Q$ such that $y_k \to y^\star := x + t^\star(y-x) \notin Q$. Hence, $y^\star \in \partial Q$, and by assumption (5.31) we have $f(y_k) \to +\infty$.

At the same time, $\|y_k - x\|_x = (t^\star - \epsilon_k)\|y - x\|_x < \|y-x\|_x < \frac{2}{M}$ and we conclude that the Hessian is uniformly bounded over segments $z = x + \alpha(y_k - x) \in Q$, $0 \leq \alpha \leq 1$:

$$\|\nabla^2 f(z)\| \;\overset{(5.20)}{\leq}\; \tfrac{1}{(1-\frac{M}{2}\|z-x\|_x)^2}\|\nabla^2 f(x)\| \;<\; L \;:=\; \tfrac{1}{(1-\frac{M}{2}\|y-x\|_x)^2}\|\nabla^2 f(x)\|.$$

Therefore, all function values $f(y_k)$ are uniformly bounded, which contradicts $f(y_k) \to +\infty$. □

5.2.3 Local Convergence of Newton's Method

In this lecture, we establish the local quadratic convergence of the classic Newton's method. Using self-concordant analysis, it is possible to provide an *affine-invariant characterization* of the *region of quadratic convergence*, which is essential for the design of interior-point methods.

We consider differentiable strictly convex function $f : Q \to \mathbb{R}$ defined on an open convex set $Q \subseteq \mathbb{R}^n$. We assume that Q is a natural *domain* of f, so the function blows up when approaching the boundary, and that f is self-concordant with constant $M > 0$ on this set (see previous section).

Dual local norm. We use the local norm $\|\cdot\|_x$, induced by the Hessian, for the primal vectors, for any $x \in Q$:

$$\|h\|_x \;:=\; \langle \nabla^2 f(x)h, h\rangle^{1/2}, \qquad h \in \mathbb{R}^n.$$

For the dual objects (gradients), we have to use the *dual norm* to it, which with abuse of notation we denote by the same symbol. Thus, for a linear form $\langle g, \cdot\rangle$, $g \in \mathbb{R}^n$, its local norm at $x \in Q$ is

$$\|\langle g, \cdot\rangle\|_x \;\equiv\; \|g\|_{x,*} \;\equiv\; \max_{h \in \mathbb{R}^n \,:\, \|h\|_x \leq 1} \langle g, h\rangle \;=\; \langle g, \nabla^2 f(x)^{-1} g\rangle \;=\; \|\nabla^2 f(x)^{-1/2} g\|_2,$$

where denotes the standard Euclidean norm for vectors, and the spectral norm for matrices.

The most important case for us is when $g := \nabla f(x)$. Denote,

$$\boxed{\lambda(x) \;:=\; \|\langle \nabla f(x), \cdot\rangle\|_x \;=\; \langle \nabla f(x), \nabla^2 f(x)^{-1} \nabla f(x)\rangle^{1/2},}$$

which is sometimes called *the Newton decrement*.

Newton's step. Consider one iteration of Newton's method:

$$x^+ \;=\; x - \nabla^2 f(x)^{-1}\nabla f(x) \qquad \Leftrightarrow \qquad \nabla f(x) + \nabla^2 f(x)(x^+ - x) \;=\; 0. \tag{5.33}$$

Then, the Newton decrement is equal to the length of Newton's step in local norm:

$$\|x^+ - x\|_x \;=\; \langle \nabla^2 f(x)(x^+ - x), x^+ - x\rangle^{1/2} \;=\; \langle \nabla f(x), \nabla^2 f(x)^{-1}\nabla f(x)\rangle^{1/2} \;=\; \lambda(x).$$

Therefore, when

$$\lambda(x) \;<\; \tfrac{2}{M}, \tag{5.34}$$

we conclude that $x^+ \in \mathcal{E}_x$, where

$$\mathcal{E}_x \;=\; \left\{ y \,:\, \|y - x\|_x < \tfrac{2}{M} \right\} \;\subseteq\; Q$$

is Dikin's ellipsoid (see Proposition 5.2.6 in the previous section). Hence, by preconditioning the gradient direction with the inverted Hessian (5.33), we remain in the domain without any auxiliary projections, under condition (5.34).

In the last lecture we have proved the following lemma, that is a key to analyze local behavior of the Newton's method.

Lemma 5.2.7. *Let $x \in Q$ and assume that $y \in \mathcal{E}_x$. Then, the Hessians are comparable:*

$$(1 - \tfrac{M}{2}\|y - x\|_x)^2 \nabla^2 f(x) \;\preceq\; \nabla^2 f(y) \;\preceq\; (1 - \tfrac{M}{2}\|y - x\|_x)^{-2} \nabla f(x). \tag{5.35}$$

We are ready to establish the main result about Newton's method.

Theorem 5.2.8. *Let*

$$\lambda(x) \;\leq\; \tfrac{1}{M}. \tag{5.36}$$

Then,

$$\lambda(x^+) \;\leq\; M\lambda(x)^2 \tag{5.37}$$

Proof. Denote $r := \lambda(x) \overset{(5.36)}{\leq} \frac{1}{M} < \frac{2}{M}$. Therefore, the condition of Lemma 5.2.7 is satisfied.

First, we can move from the new norm, to the old norm, using the lemma:

$$\lambda(x^+) \;=\; \|\nabla f(x^+)\|_{x^+} \overset{(5.35)}{\leq} \tfrac{1}{1 - \frac{M}{2}r}\|\nabla f(x^+)\|_x.$$

Then, using the definition of Newton's step and the main theorem of calculus, we get:

$$\begin{aligned} \nabla f(x^+) &\overset{(5.33)}{=} \nabla f(x^+) - \nabla f(x) - \nabla^2 f(x)(x^+ - x) \\ &= (G - H)(x^+ - x), \end{aligned}$$

where $G = \int\limits_0^1 \nabla^2 f(x + \tau(x^+ - x))d\tau$ and $H = \nabla^2 f(x)$. Hence, we obtain:

$$\begin{aligned} \|\nabla f(x^+)\|_x &= \|(G - H)(x^+ - x)\|_x \\ &= \|H^{-1/2}(G - H)H^{-1/2}H^{1/2}(x^+ - x)\|_2 \\ &\leq \|H^{-1/2}(G - H)H^{-1/2}\|_2 \cdot \|H^{1/2}(x^+ - x)\|_2 \\ &= \|H^{-1/2}(G - H)H^{-1/2}\|_2 \cdot r \end{aligned}$$

We have the following lower bounds:

$$G \;=\; \int\limits_0^1 \nabla^2 f(x+\tau(x^+-x))d\tau \;\overset{(5.35)}{\succeq}\; H\cdot\int\limits_0^1 (1-t\tfrac{M}{2}r)^2 dt \;=\; (1-\tfrac{Mr}{2}+\tfrac{1}{12}M^2r^2)H,$$

and

$$H^{-1/2}(G-H)H^{-1/2} \;\succeq\; \tfrac{Mr}{2}\Big(\tfrac{Mr}{6}-1\Big)I.$$

The corresponding upper bound is:

$$G \;\overset{(5.35)}{\preceq}\; H\int\limits_0^1 \tfrac{dt}{(1-tMr/2)^2} \;=\; \tfrac{1}{1-Mr/2}H,$$

and

$$H^{-1/2}(G-H)H^{-1/2} \;\preceq\; \Big(\tfrac{1}{1-Mr/2}-1\Big)I \;=\; \tfrac{Mr/2}{1-Mr/2}.$$

Thus,

$$\|H^{-1/2}(G-H)H^{-1/2}\|_2 \;\leq\; \tfrac{Mr}{2}\max\Big\{\tfrac{1}{1-Mr/2}, 1-\tfrac{Mr}{6}\Big\} \;\leq\; \tfrac{Mr/2}{1-Mr/2}.$$

Combining all ingredients together, we get:

$$\lambda(x^+) \;\leq\; \tfrac{M}{2-Mr}r^2 \;=\; \tfrac{M}{2-M\lambda(x)}\lambda(x)^2 \;\overset{(5.36)}{\leq}\; M\lambda(x)^2,$$

which completes the proof. □

We observe that inequality (5.37) leads to a very quick progress of the method:

Corollary 5.2.9. *Denote*

$$\delta_k \;=\; M\lambda(x_k).$$

We have

$$\delta_{k+1} \;\leq\; \delta_k^2.$$

Hence, after $k \geq 0$ iterations, starting from $\delta_0 = M\lambda(x_k) \leq \frac{1}{2}$ we obtain

$$\delta_k \;\leq\; \delta_0^{2^k} \;\leq\; \big(\tfrac{1}{2}\big)^{2^k}. \tag{5.38}$$

The convergence rate (5.38) is called *quadratic* convergence. It is very fast: with each iteration, the number of the correct digits in the solution doubles! To obtain a point x_k satisfying:

$$\lambda(x_k) \;\leq\; \varepsilon,$$

for a given $\varepsilon > 0$, it follows from (5.38) that it is sufficient to perform just

$$k \;=\; 1+\Big\lceil \log_2\log_2 \tfrac{1}{M\varepsilon}\Big\rceil$$

Newton step, provided that $x_0 \in \mathcal{Q}$, where

$$\mathcal{Q} \;:=\; \Big\{x \,:\, \lambda(x) \;=\; \langle\nabla f(x), \nabla^2 f(x)^{-1}\nabla f(x)\rangle^{1/2} \;\leq\; \tfrac{1}{2M}\Big\} \;\subseteq\; Q$$

is the *region of quadratic convergence.*

Note that set $\mathcal{Q}$ is affine-invariant as it does not depend on the choice of the coordinate system in our space. At the same time, if we fix any particular norm $\|\cdot\|$, and assume that function f is strongly convex with parameter $\mu > 0$ with respect to this norm:

$$\langle \nabla^2 f(x)h, h\rangle \;\geq\; \mu\|h\|^2, \qquad \forall h \in \mathbb{R}^n,\; x \in Q. \tag{5.39}$$

Then,

$$\lambda(x)^2 \;=\; \langle \nabla f(x), \nabla^2 f(x)^{-1}\nabla f(x)\rangle \;\leq\; \|\nabla f(x)\|_*\|\nabla^2 f(x)^{-1}\nabla f(x)\| \overset{(5.39)}{\leq} \tfrac{\lambda(x)}{\mu^{1/2}}\|\nabla f(x)\|_*,$$

and we obtain an upper bound on the Newton decrement:

$$\lambda(x) \;\leq\; \tfrac{1}{\mu^{1/2}}\|\nabla f(x)\|_*. \tag{5.40}$$

Assuming additionally that the Hessian is Lipschitz, for some constant $L > 0$:

$$\|\nabla^2 f(y) - \nabla^2 f(x)\| \;\leq\; L\|y - x\|, \qquad \forall x, y \in Q,$$

we can set the constant of self-concordance for f as $M = \frac{L}{\mu^{3/2}}$ (see Section 5.2.1).

Corollary 5.2.10. *For any norm $\|\cdot\|$, consider the region:*

$$\mathcal{G} \;=\; \mathcal{G}_{\|\cdot\|} \;=\; \Big\{ x \;:\; \|\nabla f(x)\|_* \;\leq\; \tfrac{\mu^2}{2L} \Big\}. \tag{5.41}$$

Then, all points from $\mathcal{G}$ are in the region of quadratic convergence of Newton's method:

$$\mathcal{G} \;\subseteq\; \mathcal{Q}.$$

Proof. Indeed, for any $x \in \mathcal{G}$:

$$\lambda(x) \overset{(5.40)}{\leq} \tfrac{1}{\mu^{1/2}}\|\nabla f(x)\|_* \overset{(5.41)}{\leq} \tfrac{\mu^{3/2}}{2L} \;=\; \tfrac{1}{2M},$$

thus $x \in \mathcal{Q}$. □

When using a fixed norm, it is much easier to prove directly that (5.41) is the region of quadratic convergence for Newton's method. However, we obtain it as a simple direct consequence of Theorem 5.2.8. Affine-invariant characterization of $\mathcal{Q}$ is crucial for analyzing the path-following scheme.

Bound for the distance to the solution. We have proved local quadratic convergence in terms of the quantity $\lambda(x)$. But what about other possible accuracy measures? We can think of the functional residual $f(x) - f^\star$, or distance to the solution, e.g. $\|x - x^\star\|_x$. It appears that, locally, *all these measures are equivalent* (see Theorem 5.2.1 in [31] for the exact bounds):

$$f(x) - f^\star \;\approx\; \|x - x^\star\|_x \;\approx\; \lambda(x). \tag{5.42}$$

So since we can make $\lambda(x)$ extremely small, we can also make any of these measures as small as we want.

Let us prove one inequality quantifying (5.42), which will be important for our further analysis of the path-following scheme.

Proposition 5.2.11. *Let for some point $x \in Q$ we have $\lambda(x) \leq \frac{1}{2M}$. Then, the minimizer $x^\star$ of f exists, and it holds:*

$$\|x - x^\star\|_x \;\leq\; 3\lambda(x). \tag{5.43}$$

Proof. Consider the closed ellipsoid $B := \{y \;:\; \|y - x\|_x \;\leq\; 3\lambda(x)\} \subset \mathcal{E}_x$. Our goal is to show $x^\star \in B$. We can assume $\lambda(x) \neq 0$, since otherwise $x = x^\star$ and the statement holds.

First, for any $y \in \mathcal{E}_x$, using the main theorem of calculus, we have

$$\begin{aligned} \langle \nabla f(y) - \nabla f(x), y - x \rangle &= \int\limits_0^1 \langle \nabla^2 f(x + \tau(y-x))(y-x), y - x \rangle d\tau \\ &\overset{(5.35)}{\geq} \|y - x\|_x^2 \cdot \int\limits_0^1 (1 - \tau \tfrac{M}{2} \|y - x\|_x)^2 d\tau \\ &\geq \|y - x\|_x^2 \cdot \int\limits_0^1 (1 - \tau)^2 d\tau \;=\; \tfrac{1}{3}\|y - x\|_x^2. \end{aligned} \tag{5.44}$$

Consider points from the boundary of the ellipsoid, $y \in S := \partial B$, thus $\|y - x\|_x = 3\lambda(x)$. Then, for such points:

$$\begin{aligned} \langle \nabla f(y), y - x \rangle &\overset{(5.44)}{\geq} \langle \nabla f(x), y - x \rangle + \tfrac{1}{3}\|y - x\|_x^2 \\ &\geq -\lambda(x)\|y - x\|_x + \tfrac{1}{3}\|y - x\|_x^2 \;=\; 0. \end{aligned} \tag{5.45}$$

Then, for any $z \in Q \setminus B$, there exists $\alpha \in (0, 1)$ such that $y = \alpha x + (1 - \alpha) z \in S$. By convexity, we obtain

$$f(z) \;\geq\; f(y) + \langle \nabla f(y), z - y \rangle \;=\; f(y) + \tfrac{\alpha}{1-\alpha} \langle \nabla f(y), y - x \rangle \overset{(5.45)}{\geq} f(y).$$

Therefore, the minimum of f over the compact set B is its global minimum, which proves the required statement. □

5.3 Interior-Point Method

We consider the convex optimization problem in the following form,

$$\min \Big\{ \langle c, x \rangle \;:\; x \in \bar{Q} \Big\}, \tag{5.46}$$

where $Q \subset \mathbb{R}^n$ is an open convex set, which, we assume to be bounded, for simplicity.

The idea of interior-point methods is to substitute constrained problem (5.46) by a *sequence of unconstrained minimization* subproblems. For that, we introduce a "barrier" function F defined on $\operatorname{dom} F = Q$, that prevents us from going outside the set. Then, we consider a family of objectives:

$$f_t(x) \;=\; t\langle c, x \rangle + F(x), \qquad t \geq 0. \tag{5.47}$$

We denote the minimum of (5.47) by

$$x_t^\star \;:=\; \operatorname*{argmin}_{x \in \operatorname{dom} F} \Big[\, t\langle c, x \rangle + F(x) \, \Big], \tag{5.48}$$

which is called the *central path*. The initial point of the central path,

$$x_0^\star \;=\; \operatorname*{argmin}_{x \in \operatorname{dom} F} F(x),$$

is called an *analytic center* of the set, and $x_t^\star \to x^\star$ with $t \to +\infty$, where $x^\star \in \partial Q$ is a solution to (5.46). Note that a solution to (5.46) is always at the boundary, while the central path belong to the interior of our feasible set:

$$x_t^\star \;\in\; Q, \qquad t \geq 0.$$

The idea of interior-point methods is to trace the central path, by approximately solving (5.48) for an increasing sequence of parameters t, starting from $t = 0$.

Since

$$\nabla^2 f_t(x) \;\equiv\; \nabla^2 F(x),$$

we see that the second derivative of f_t does not depend on t. It appears that the geometry induced by $\nabla^2 F(x)$ is crucial for being able to solve (5.47) efficiently.

Note that even though the subproblem in (5.48) is formally still a *constrained* minimization, with the right choice of the barrier F, we can use the plain Newton's method to solve it. In fact, after each change of $t \mapsto t^+ = t + \Delta$ in (5.47), we will use only *one Newton's step*, ensuring that every new point belongs to the local region of quadratic convergence around the central path. The "right" choice of F is described by the following formal definition of *self-concordant barriers*.

5.3.1 Self-Concordant Barriers

We say that a strictly convex differentiable function $F : Q \to \mathbb{R}$ is a *self-concordant barrier* for a convex open set Q with parameter $\theta > 0$ if the following three conditions are satisfied:

1. *Q is the domain of F*: for any sequence $\{x_k\}_{k \geq 0}$ such that $x_k \to \partial Q$ it holds $F(x_k) \to +\infty$.

2. *F is a standard self-concordant function* (the constant of self-concordance is $M = 2$; otherwise we can rescale the barrier):

$$D^3 F(x)[h, h, h] \;\leq\; 2\|h\|_x^3 \;\equiv\; 2\langle \nabla^2 F(x) h, h\rangle^{3/2}, \qquad \forall x \in Q,\; h \in \mathbb{R}^n. \tag{5.49}$$

3. *F is Lipschitz with respect to the local norm*[1]*:*

$$\|DF(x)\|_x^2 \;\equiv\; \langle \nabla F(x), \nabla^2 F(x)^{-1} \nabla F(x)\rangle \;\leq\; \theta, \qquad \forall x \in Q. \tag{5.50}$$

The first two conditions are already familiar to us. In particular, they imply that with every point $x \in Q$, the entire *Dikin's ellipsoid* belong to the domain (see Proposition 5.2.6 in Section 5.2.2 for the proof):

$$\mathcal{E}_x \;=\; \left\{ y \;\in\; \mathbb{R}^n \;:\; \|y - x\|_x < 1 \right\} \;\subseteq\; Q,$$

and one Newton's step $x \mapsto x^+ = x - \nabla^2 F(x)^{-1} g$, for some $g \in \mathbb{R}^n$, remains in the ellipsoid, as soon as

$$\|x^+ - x\|_x \;=\; \langle g, \nabla^2 F(x)^{-1} g\rangle^{1/2} \;<\; 1. \tag{5.51}$$

[1]Recall that we use the dual norm to measure the size of the linear form $DF(x)[\cdot] = \langle \nabla F(x), \cdot\rangle$. That is why the Hessian in (5.50) is inverted, while in (5.49) we use the primal local norm.

This explains why we can treat optimization subproblem (5.48) as unconstrained one: assuming that g is sufficiently small (5.51) and using the Hessian of the barrier as a preconditioner ensures the feasibility of all points *without the need to perform any projections*. If first-order methods are used to solve (5.48), this property is lost. Note that computing the projection onto Q is harder than solving the original linear minimization problem (5.46).

The third condition (5.50) of the definition is new. It enable the efficient tracing of the central path (5.48). The main result of the interior-point method theory we aim to establish is the following statement — Theorem 5.3.10: *we can solve* (5.46) *to within accuracy* $\varepsilon > 0$*, i.e., find a point* $\bar{x} \in Q$ *such that* $\langle c, \bar{x} - x^\star \rangle \leq \varepsilon$ *using*

$$K \;=\; O\Big(\sqrt{\theta} \ln \tfrac{1}{\varepsilon}\Big) \tag{5.52}$$

total number of Newton steps.

Therefore, the barrier parameter $\theta > 0$ describes the complexity of the problem (5.46).

Equivalent conditions. The barrier condition (5.50) is equivalent to the following global inequality:

$$2\langle \nabla F(x), u\rangle - \langle \nabla^2 F(x)u, u\rangle \;\leq\; \theta, \qquad \forall u \in \mathbb{R}^n,\; x \in Q. \tag{5.53}$$

Indeed, (5.50) is the maximum of the left-hand-side of (5.53) over $u \in \mathbb{R}^n$. Furthermore, the new condition (5.53) can be used as a definition in cases when $\nabla^2 F(x)$ has *degenerate directions*, implying that F is not strictly convex along them[2].

Finally, substituting $u := \tau h$ for $\tau > 0$ and $h \in \mathbb{R}^n$ in (5.53), and maximizing the left-hand-side over $\tau > 0$, which corresponds to *homogenization* of the inequality, we obtain the following equivalent global bound:

$$\max_{\tau > 0} \Big[\; 2\tau \langle \nabla F(x), h\rangle - \tau^2 \langle \nabla^2 F(x)h, h\rangle \;\Big] \;=\; \tfrac{\langle \nabla F(x), h\rangle^2}{\langle \nabla^2 F(x)h, h\rangle} \overset{(5.53)}{\leq} \theta, \tag{5.54}$$

where the maximum is achieved for $\tau^\star = \frac{\langle \nabla F(x), h\rangle}{\langle \nabla^2 F(x)h, h\rangle}$. The last condition is the most convenient for checking the third barrier property. It can be rewritten as follows:

$$\langle \nabla F(x), h\rangle^2 \;\leq\; \theta \langle \nabla^2 F(x)h, h\rangle, \qquad \forall h \in \mathbb{R}^n,\; x \in Q, \tag{5.55}$$

or

$$\nabla^2 F(x) \;\succeq\; \tfrac{1}{\theta} \nabla F(x) \nabla F(x)^\top. \tag{5.56}$$

Therefore, we can interpret the last barrier property as a certain form of "self-concordant strong convexity" with parameter $\mu_{\mathfrak{sc}} := \frac{1}{\theta}$, while the self-concordant parameter $L_{\mathfrak{sc}} := 2$ is a measure of "smoothness". Under this speculative interpretation, the complexity (5.52) resembles that of the fast gradient method for smooth and strongly convex functions:

$$O\Big(\sqrt{\tfrac{L_{\mathfrak{sc}}}{\mu_{\mathfrak{sc}}}} \ln \tfrac{1}{\varepsilon}\Big),$$

even though the analysis of path-following schemes looks very different.

We have the following important examples of self-concordant barriers. We already know that these functions are standard self-concordant. Therefore, we check only the new barrier condition (5.50).

[2]This might happen if Q contains a line: $\{x + \tau h \;:\; \tau \in \mathbb{R}\} \subseteq Q$ for some $x \in Q$ and direction $h \in \mathbb{R}^n$. In this case, self-concordance of F implies $\nabla^2 F(y)h \equiv 0$ *for all* $y \in Q$ along this direction h. Assuming that Q does not contain lines (e.g., it is bounded), we prevent this situation, ensuring $\nabla^2 F(y) \succ 0$ for all $y \in Q$.

Example 5.3.1. $F(x) = -\log x$ is self-concordant barrier for $\mathbb{R}_{>0}$ with $\boxed{\theta = 1}$. Indeed,

$$F'(x) \;=\; -\tfrac{1}{x}, \qquad F''(x) \;=\; \tfrac{1}{x^2}.$$

Hence, (5.50) is satisfied.

Example 5.3.2. $F(X) = -\log\det(X)$ is self-concordant barrier for $\mathbb{S}^n_{\succ 0}$ with $\boxed{\theta = n}$. We have,

$$\begin{aligned} DF(X)[H] &= \operatorname{tr}(X^{-1}H) = \operatorname{tr}(S), \quad \text{for} \quad S := X^{-1/2}HX^{-1/2}, \quad \text{and} \\ D^2F(x)[H,H] &= \operatorname{tr}(X^{-1}HX^{-1}H) = \operatorname{tr}(S^2). \end{aligned}$$

Therefore, condition (5.55) implies that we need to check

$$\operatorname{tr}(S)^2 \;=\; \Big[\sum_{i=1}^n \lambda_i(S)\Big]^2 \;\le\; \theta\Big[\sum_{i=1}^n \lambda_i(S)^2\Big] \;=\; \operatorname{tr}(S^2),$$

which is true for $\theta = n$ due to the standard inequality between norms: $\|\cdot\|_1 \le \sqrt{n}\|\cdot\|_2$.

Let us look at some simple properties of the self-concordant barriers.

Proposition 5.3.3. *Let $F_1, \dots, F_m$ be self-concordant barriers for $Q_1, \dots, Q_m$. Then,*

$$F(x) \;=\; \sum_{i=1}^m F_i(x)$$

is self-concordant barrier for

$$Q \;=\; \bigcap_{1\le i\le m} Q_i.$$

with

$$\boxed{\theta \;=\; \sum_{i=1}^m \theta_i.}$$

Proof. It follows immediately from the bound (5.53) and the fact that the maximum of a sum does not exceed the sum of the maxima. □

Example 5.3.4. The logarithmic barrier for the positive orthant $Q = \mathbb{R}^n_{>0}$ given as the sum of barriers for rays (5.3.1),

$$F(x) \;=\; -\sum_{i=1}^n \log x^{(i)},$$

is a self-concordant barrier with parameter $\boxed{\theta = n}$.

Note that, similarly to the definition of self-concordant functions, the definition of self-concordant barriers is *affine-invariant* (check it!), and thus the parameter of self-concordance θ is preserved under affine transformation:

Proposition 5.3.5. *Let $\mathcal{A}(x) = Ax + b$ be an affine transformation, and let $F : Q \to \mathbb{R}$ be a self-concordant barrier for Q with parameter θ_F. Then,*

$$\Phi(x) \;=\; F(\mathcal{A}(x)), \qquad x \in \Omega,$$

is a self-concordant barrier for the affine preimage of Q:

$$\Omega \;=\; \Big\{x \;:\; \mathcal{A}(x) \in Q\Big\},$$

with parameter $\boxed{\theta_\Phi = \theta_F}$.

Example 5.3.6. The logarithmic barrier for the polyhedron $\{x \in \mathbb{R}^n : \langle a_i, x\rangle \le b_i, 1 \le i \le m\}$:

$$F(x) \;=\; -\sum_{i=1}^{m} \ln(b_i - \langle a_i, x\rangle)$$

is a self-concordant barrier with parameter $\boxed{\theta = m}$.

Key property. It appears that the following property of self-concordant barriers is central for the interior-point methods theory. Functions that satisfy (5.57) are called *set-limited* [33].

Lemma 5.3.7. *For any $x, y \in Q$, we have*

$$\langle \nabla F(x), y - x\rangle \;\le\; \theta. \tag{5.57}$$

Proof. Consider $\varphi(t) := \langle \nabla F(x + t(y-x)), y - x\rangle$, for $t \in [0,1]$. Then, our goal is to show that $\varphi(0) \le \theta$. We have

$$\begin{aligned} \varphi'(t) \;&=\; \langle \nabla^2 F(x + t(y-x))(y-x), y - x\rangle \\ &\overset{(5.55)}{\ge}\; \tfrac{1}{\theta}\langle \nabla F(x + t(y-x)), y - x\rangle^2 \;=\; \tfrac{1}{\theta}\varphi(t)^2. \end{aligned} \tag{5.58}$$

Thus, $\varphi(t)$ is increasing with t. We can assume that $\varphi(0) > 0$ (otherwise, (5.57) trivially holds). Therefore, we conclude

$$\tfrac{1}{\varphi(0)} \;>\; \tfrac{1}{\varphi(0)} - \tfrac{1}{\varphi(1)} \;=\; \int_0^1 \tfrac{d}{dt}\Big[-\tfrac{1}{\varphi(t)}\Big] \;=\; \int_0^1 \tfrac{\varphi'(t)dt}{\varphi(t)^2} \;\overset{(5.58)}{\ge}\; \tfrac{1}{\theta},$$

which completes the proof. □

Geometric interpretation of inequality (5.57) is that normalized gradient direction $\frac{1}{\theta}\nabla F(x)$ belongs to the *polar* to Q at point $x \in Q$:

$$\tfrac{1}{\theta}\nabla F(x) \;\overset{(5.57)}{\in}\; P_Q(x) \;:=\; \Big\{ g \in \mathbb{R}^n \;:\; \langle g, y - x\rangle \le 1 \;\; \forall y \in Q\Big\}.$$

5.3.2 Path-Following Scheme

We consider the following constrained convex optimization problem,

$$\min_{x \in \bar{Q}} \langle c, x\rangle, \tag{5.59}$$

where $Q \subset \mathbb{R}^n$ is an open bounded convex set equipped with a *self-concordant barrier* $F : Q \to \mathbb{R}$. We denote by $\theta > 0$ the parameter of the barrier (see previous section), which describes the complexity of set Q. Note that the same set can be equipped with different self-concordant barriers, and ideally we would want to choose that one with the smallest possible barrier parameter θ.

For solving (5.59), we trace the *central path* $x_t^\star \in Q$, for $t \ge 0$, defined as the minimum of the following subproblem:

$$x_t^\star \;=\; \operatorname*{argmin}_{x \in \operatorname{dom} F}\Big[\; f_t(x) \;:=\; t\langle c, x\rangle + F(x) \;\Big], \tag{5.60}$$

and when $t \to +\infty$ the central path converges to a solution $x^\star \in \partial Q$ to (5.59): $x_t^\star \to x^\star$ (we prove a convergence rate for the central path in terms of the functional residual later in this section).

Note that for our family of perturbed objectives $f_t(\cdot)$, the gradients are different by a constant term, while all the Hessians are the same, for any $t \geq 0$ and $\Delta \geq 0$:

$$\begin{aligned} \nabla f_{t+\Delta}(x) &\equiv \nabla f_t(x) + \Delta c \equiv \nabla F(x) + (t+\Delta)c, \\ \nabla^2 f_{t+\Delta}(x) &\equiv \nabla^2 f_t(x) \equiv \nabla^2 F(x). \end{aligned}$$

Therefore, for any moment of time t, the local norm at any point $x \in Q$ remains the same for all t, and it is fully defined by the Hessian $\nabla^2 F(x)$ of the self-concordant barrier at this point.

Newton's step on perturbed problem. We are going to trace the central path approximately, utilizing the machinery of self-concordant functions, that we developed previously.

The optimality condition for the exact optimum in (5.60) is, for any $t \geq 0$:

$$\nabla f_t(x_t^\star) = tc + \nabla F(x_t^\star) = 0. \tag{5.61}$$

Let us assume that at a current fixed moment of time $t \geq 0$, we have a point $x \in Q$ that is an approximation of the central path, $x \approx x_t^\star$, under the following inexactness condition of a small gradient norm:

$$\|\nabla f_t(x)\|_x := \langle \nabla f_t(x), \nabla^2 F(x)^{-1} \nabla f_t(x) \rangle^{1/2} \leq \delta := \tfrac{1}{10}. \tag{5.62}$$

We fix $\delta = 1/10$ for simplicity of the presentation, while tighter constants can be obtained.

Consider an update of time,

$$\boxed{t^+ = t + \Delta > 0,}$$

for some small Δ (which can be either positive, if we increase t, or negative, if we want to trace the central path backwards in time), and one Newton's step from x for the new objective $f_{t^+}(\cdot)$:

$$\boxed{x^+ = x - \nabla^2 F(x)^{-1}(\nabla F(x) + t^+ c).}$$

It appears that if Δ is small, then we can ensure that x^+ will remain close to the central path with the same inexactness condition as in (5.62).

Theorem 5.3.8. *Let x satisfy (5.62) with $\delta = \frac{1}{10}$ and let*

$$|\Delta| \leq \tfrac{\gamma}{\|c\|_x}, \tag{5.63}$$

for $\gamma = \frac{1}{10}$ and $\|c\|_x := \langle c, \nabla F(x)^{-1} c \rangle^{1/2}$ is the dual norm of the target linear form. Then:

$$\|\nabla f_{t^+}(x^+)\|_{x^+} \leq \delta. \tag{5.64}$$

Proof. Denote by $\lambda(y)$ the local norm of the gradient of the *new function* f_{t^+} at point y. Thus,

$$\lambda(x) = \|\nabla f_{t^+}(x)\|_x = \|\nabla f_t(x) + \Delta c\|_x \overset{(5.63)}{\leq} \|\nabla f_t(x)\|_x + \gamma \overset{(5.62)}{\leq} \delta + \gamma. \tag{5.65}$$

Note that f_{t^+} is a standard self-concordant functions (the parameter of self-concordance is $M = 2$). Hence, by our choice of δ and γ, the point x lies in the region of local convergence of Newton's method: $\lambda(x) \leq \frac{1}{5} < \frac{1}{M} = \frac{1}{2}$ (see Theorem 5.2.8 in Section 5.2.3), and thus, after one Newton's, step we have

$$\lambda(x^+) = \|\nabla f_{t^+}(x^+)\|_{x^+} \leq 2\lambda(x)^2 \leq \tfrac{2}{25} < \delta,$$

which proves the required statement. □

Therefore, starting from a point x_0 within proximity to the analytic center: $\|\nabla f_0(x_0)\|_{x_0} = \|\nabla F(x_0)\|_{x_0} \leq \delta = \frac{1}{10}$, we can remain close the central path, ensuring the invariant (5.62), as soon as time t is updated not very fast.

The rate of updating time. Note that to prove the previous theorem, we did not use the third *barrier property* of F, which is the Lipschitzness of F with respect to the local norm. This property is crucial to establish a fast *linear rate* with which we are able to update the time.

Lemma 5.3.9. *Let* $\|\nabla f_t(x)\|_x \leq \delta = \frac{1}{10}$. *Then,*

$$\|c\|_x \;\leq\; \tfrac{1}{t}\Big(\tfrac{1}{10} + \sqrt{\theta}\Big). \tag{5.66}$$

Consequently, ensuring (5.63)*, we can update time with the linear rate:*

$$t^+ \;:=\; t + \tfrac{1}{10\|c\|_x} \;\overset{(5.66)}{\geq}\; t \cdot \Big(1 + \tfrac{1}{1+10\sqrt{\theta}}\Big) \;\geq\; t \cdot \exp\Big(\tfrac{1}{2(1+10\sqrt{\theta})}\Big). \tag{5.67}$$

Proof. Indeed, we have

$$t\|c\|_x \;=\; \|\nabla f_t(x) - \nabla F(x)\|_x \;\leq\; \|\nabla f_t(x)\|_x + \|\nabla F(x)\|_x \;\leq\; \delta + \sqrt{\theta},$$

which proves (5.66). □

A similar rate to (5.67) holds for tracing the central path backwards, replacing '+' by '−'.

Convergence rate of central path. We are ready to prove the global complexity for a method which traces the central path.

First, let us show that the exact central path $x_t^\star$ indeed convergences to the optimal solution in terms of the objective function value. For that, we observe that according to optimality condition (5.61), for any positive moment of time $t > 0$, it holds:

$$c \;\overset{(5.61)}{=}\; -\tfrac{1}{t}\nabla F(x_t^\star). \tag{5.68}$$

At the same time, by the set-limitedness of F (see Lemma 5.3.7 in Section 5.3.1), for any $x, y \in Q$, we have

$$\langle \nabla F(x), y - x\rangle \;\leq\; \theta. \tag{5.69}$$

Therefore,

$$\langle c, x_t^\star\rangle - \langle c, x^\star\rangle \;=\; \langle c, x_t^\star - x^\star\rangle \;\overset{(5.68)}{=}\; \tfrac{1}{t}\langle \nabla F(x_t^\star), x^\star - x_t^\star\rangle \;\overset{(5.69)}{\leq}\; \tfrac{\theta}{t}. \tag{5.70}$$

Now, let $x \approx x_t^\star$ be an approximate point that satisfies condition (5.62). Note that the small gradient norm implies that the distance between points is also small. In particular, using Proposition 5.2.11 from Section 5.2.3, under condition (5.62): $\|\nabla f_t(x)\|_x \leq \frac{1}{\delta} = \frac{1}{10} < \frac{1}{4} = \frac{1}{2M}$, we have[3]

$$\|x - x_t^\star\|_x \;\leq\; 3\|\nabla f_t(x)\|_x \;\leq\; \tfrac{3}{10}. \tag{5.71}$$

Thus,

$$\langle c, x - x_t^\star\rangle \;\overset{(5.71)}{\leq}\; \tfrac{3}{10}\|c\|_x \;\overset{(5.66)}{\leq}\; \tfrac{3}{10t}\Big(\tfrac{1}{10} + \sqrt{\theta}\Big), \tag{5.72}$$

and we obtain the bound on the functional residual at point x, as follows:

$$\langle c, x\rangle - \langle c, x^\star\rangle \;\overset{(5.72),(5.70)}{\leq}\; \tfrac{1}{t}\Big[\theta + \tfrac{3}{10}\Big(\tfrac{1}{10} + \sqrt{\theta}\Big)\Big] \;\leq\; \tfrac{2}{t}\Big[\theta + \tfrac{3}{100}\Big]. \tag{5.73}$$

[3] A more precise analysis can be used to show that $\|x - x_t^\star\|_x \leq \frac{\|\nabla f_t(x)\|_x}{1 - \|\nabla f_t(x)\|_x} \leq \frac{\delta}{1-\delta} = \frac{1}{9}$.

Hence, if we want to find a point $\bar{x}$ with an ε-accuracy in terms of the target residual:

$$\langle c, \bar{x} \rangle - \langle c, x^\star \rangle \;\leq\; \varepsilon, \tag{5.74}$$

according to (5.73), it is enough to trace the central path up to the moment:

$$t \;=\; \tfrac{2}{\varepsilon}\Big[\theta + \tfrac{3}{100}\Big]. \tag{5.75}$$

Assume that we start with some $t_1 > 0$, and update discrete timestamps $(t_1, t_2, t_3, \ldots)$ with the following linear rate, as in (5.67):

$$t_{k+1} \;\geq\; t_k \exp\Big(\tfrac{1}{2(1+10\sqrt{\theta})}\Big) \;\geq\; \ldots \;\geq\; t_1 \exp\Big(\tfrac{k}{2(1+10\sqrt{\theta})}\Big) \;\overset{(*)}{\geq}\; \tfrac{2}{\varepsilon}\Big[\theta + \tfrac{3}{100}\Big],$$

where $(*)$ holds as soon as

$$k \;\geq\; 2(1+10\sqrt{\theta}) \cdot \log\Big(\tfrac{2}{t_1 \varepsilon}\Big[\theta + \tfrac{3}{100}\Big]\Big). \tag{5.76}$$

5.3.3 Interior-Point Algorithm

We come to the following direct algorithm for solving problem (5.59). This method requires as initialization a point x_0 that is already close to the analytic center $x_0^\star := \operatorname{argmin}_{y \in Q} F(y)$. To find x_0 we can use an auxiliary path-following algorithm, which we discuss in the next section.

Algorithm 5.1: *Path-Following Interior-Point Method.*

Initialization: Fix $\delta = \gamma = \frac{1}{10}$. Choose $x_0 \in Q$ such that $\|\nabla F(x_0)\|_{x_0} \leq \delta$. Set $t_0 = 0$.

For $k \geq 0$ **iterate:**

1. Update time: $t_{k+1} = t_k + \frac{\gamma}{\|c\|_{x_k}}$, where $\|c\|_{x_k} := \langle c, \nabla^2 F(x_k)^{-1} c \rangle^{1/2}$
2. Perform Newton's step with perturbed gradient:
$$x_{k+1} \;=\; x_k - \nabla^2 F(x_k)^{-1}\Big(\nabla F(x_k) + t_{k+1} c\Big)$$
3. If $t_{k+1} \geq \frac{2}{\varepsilon}\Big[\theta + \frac{3}{100}\Big]$ then **return** x_k

According to previous observations, we have proved the following result:

Theorem 5.3.10. *For any $\varepsilon > 0$, Algorithm 5.1 stops and return as the result a solution:*

$$\langle c, x_k \rangle - \langle c, x^\star \rangle \;\leq\; \varepsilon,$$

after the following number of iterations (Newton's steps):

$$k \;=\; O\Big(\Big[1 + \sqrt{\theta}\Big] \cdot \log \tfrac{1+\theta}{t_1 \varepsilon}\Big). \tag{5.77}$$

An auxiliary path to the analytic center. To find the analytic center $x_0^\star$, we can trace an *auxiliary path* that starts from any feasible point $y_1 \in Q$ and leads to $x_0^\star$, as follows:

$$y_s^\star \;=\; \operatorname*{argmin}_y \Big[\, \bar{f}_s(y) \;:=\; -s\langle \nabla F(y_1), y\rangle + F(y) \,\Big], \qquad 1 \geq s \geq 0. \tag{5.78}$$

The optimality condition for (5.78) is $\nabla F(y_s^\star) = s\nabla F(y_1)$. Hence, $y_1^\star = y_1$ is the starting point of the auxiliary path, which we trace and $y_0^\star = x_0^\star$ is the endpoint that we wish to approach.

Using the same reasoning as above, it is easy to show that total complexity of this auxiliary traverse is of the same order (5.77).

5.4 Cubic Regularization of Newton's Method

The framework of self-concordant barriers and interior-point methods that we studied is very powerful, as it enables Newton's method to solve broad classes of convex structured problems with polynomial-time complexity.

The main drawback of this approach is that it requires a model designer to formulate the optimization problem in a specific form (linear programming, conic programming, etc.) In practice, we often encounter problems that are given in a *black-box* form, which may also be *non-convex*, and thus fail to satisfy the assumptions of the interior-point machinery.

Nevertheless, it is still possible to apply second-order methods in these cases by employing ideas from first-order optimization, thereby establishing superior convergence properties through the use of second-order information, i.e., the Hessians $\nabla^2 f(x)$.

Problem and assumptions. We consider the minimization problem:

$$\min_{x\in\mathbb{R}^n} f(x), \tag{5.79}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function, possibly non-convex. The ideas that we will discuss can be generalized to the constrained or even the fully-composite case (Section 3.7). However, the most important and the simplest setting to study is unconstrained minimization (5.79).

We assume that the objective f is bounded from below:

$$f^\star \;:=\; \inf_{x\in\mathbb{R}^n} f(x) > -\infty,$$

while our goal, in a general non-convex setting, is to find an *approximate stationary point* to f, as finding the global minimum is generally intractable from a complexity standpoint (Section 1.4).

We will see that by using the Hessians along with a stronger smoothness assumption on f, we can find a stationary point faster than with gradient descent. Moreover, we can additionally guarantee convergence to a *second-order stationary point* (i.e., points where the Hessian is nearly positive semidefinite), which helps to better distinguish between saddle points and local minima.

Let us fix a positive definite matrix $B = B^\top \succ 0$ (e.g., $B := I$, the identity matrix). We use it to define the *global norms* in our space, primal, dual and operator norms:

$$\|h\| \;:=\; \langle Bh, h\rangle^{1/2} \;=\; \|B^{1/2}h\|_2, \qquad h \in \mathbb{R}^n,$$

$$\|g\|_* \;:=\; \langle g, B^{-1}g\rangle^{1/2} \;=\; \|B^{-1/2}g\|_2, \qquad g \in \mathbb{R}^n,$$

and for a symmetric matrix $A = A^\top \in \mathbb{R}^{n \times n}$:

$$\begin{aligned} \|A\| &:= \max_{h \in \mathbb{R}^n \,:\, \|h\| \le 1} \|Ah\|_* = \max_{h \in \mathbb{R}^n \,:\, \|h\| \le 1} \langle Ah, h \rangle = \max_{u \in \mathbb{R}^n \,:\, \|u\|_2 \le 1} \langle B^{-1/2} A B^{-1/2} u, u \rangle \\ &= \max\Big\{ \lambda_{\max}(B^{-1/2} A B^{-1/2}), -\lambda_{\min}(B^{-1/2} A B^{-1/2}) \Big\}. \end{aligned}$$

Lipschitz Hessian. Our main assumption is that f has a Lipschitz continuous Hessian, for some constant $L \ge 0$:

$$\|\nabla^2 f(y) - \nabla^2 f(x)\| \le L \|y - x\|, \qquad x, y \in \mathbb{R}^n. \tag{5.80}$$

This condition is equivalent to

$$-L\|y - x\| B + \nabla^2 f(x) \preceq \nabla^2 f(y) \preceq \nabla^2 f(x) + L\|y - x\| B, \qquad x, y \in \mathbb{R}^n, \tag{5.81}$$

which can be seen as a variant of the Hessian stability (compare with Lemma 5.2.5 from Section 5.2.2 on self-concordant functions).

Taylor approximation bounds. Using assumption (5.80) we can characterize the global approximation error for the Taylor models, the linear model of the gradient:

$$\nabla f(y) \approx \nabla f(x) + \nabla^2 f(x)(y - x), \tag{5.82}$$

and the quadratic model of the function:

$$f(y) \approx Q(x; y) := f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{1}{2} \langle \nabla^2 f(x)(y - x), y - x \rangle. \tag{5.83}$$

Note that approximations (5.82) and (5.83) are *local* in their nature: they serve as good models when $x \approx y$.

In contrast, assumption (5.80) is *global*, as it holds for all $x, y \in \mathbb{R}^n$. Integrating bound (5.80) we obtain the following:

Lemma 5.4.1. *It holds, for any $x, y \in \mathbb{R}^n$:*

$$\begin{aligned} \|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y - x)\|_* &\le \tfrac{L}{2} \|y - x\|^2 \\ |f(y) - Q(x; y)| &\le \tfrac{L}{6} \|y - x\|^3. \end{aligned}$$

Proof. By the main theorem of calculus, we have, for any h such that $\|h\| \le 1$:

$$\begin{aligned} \langle \nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y - x), h \rangle &= \int_0^1 \langle \nabla^2 f(x + \tau(y - x))(y - x) - \nabla^2 f(x)(y - x), h \rangle d\tau \\ &\le \int_0^1 \|\nabla^2 f(x + \tau(y - x)) - \nabla^2 f(x)\| \cdot \|y - x\| d\tau \\ &\le L\|y - x\|^2 \int_0^1 \tau d\tau = \tfrac{L}{2} \|y - x\|^2. \end{aligned}$$

And, by the integral form of the Taylor theorem, we have:

$$\begin{aligned} |f(y) - Q(x; y)| &= \int_0^1 (1 - \tau) \langle \Big[\nabla^2 f(x + \tau(y - x)) - \nabla^2 f(x) \Big](y - x), y - x \rangle d\tau \\ &\le L\|y - x\|^3 \int_0^1 (1 - \tau)\tau d\tau = L\|y - x\|^3 \cdot (\tfrac{1}{2} - \tfrac{1}{3}) = \tfrac{L}{6} \|y - x\|^3. \end{aligned}$$

□

5.4.1 Cubic Regularization of Quadratic Model

From the previous lemma, we obtain the following *global upper model* of the objective function, around any point $x \in \mathbb{R}^n$:

$$\begin{aligned} f(y) &\leq \Omega_H(x;y) := Q(x;y) + \tfrac{H}{6}\|y-x\|^3 \\ &= f(x) + \langle \nabla f(x), y-x\rangle + \tfrac{1}{2}\langle \nabla^2 f(x)(y-x), y-x\rangle + \tfrac{H}{6}\|y-x\|^3, \end{aligned} \tag{5.84}$$

which holds *for all* $y \in \mathbb{R}^n$, when the regularization parameter $H \geq 0$ is sufficiently large. Indeed, by Lemma 5.4.1, inequality (5.84) holds uniformly for all $x, y \in \mathbb{R}^n$ at least when $H \geq L$.

Therefore, a natural idea for an optimization method is to minimize the upper model (5.84) to obtain the next iterate:

$$x^+ := \operatorname*{argmin}_{y \in \mathbb{R}^n} \Omega_H(x;y). \tag{5.85}$$

Note that $H := 0$ in (5.85) corresponds exactly to the *pure Newton step*. At the same time, when $H \geq L$, it follows from previous observations (5.84) that we can ensure *global progress* for each iterate; rearranging the terms, we have:

$$\begin{aligned} f(x) - f(x^+) &\overset{(5.84)}{\geq} -\Big[\langle \nabla f(x), x^+ - x\rangle + \tfrac{1}{2}\langle \nabla^2 f(x)(x^+ - x), x^+ - x\rangle + \tfrac{H}{6}\|x^+ - x\|^3\Big] \\ &\overset{(5.85)}{=} \max_{y \in \mathbb{R}^n}\Big\{\langle \nabla f(x), x - y\rangle - \tfrac{1}{2}\langle \nabla^2 f(x)(y-x), y-x\rangle - \tfrac{H}{6}\|y-x\|^3\Big\}, \end{aligned}$$

and the last expression is strictly positive[4], unless $\nabla f(x) = 0$, i.e., we are already at a stationary point. In the latter case, we either remain at the same point ($x^+ = x$) if $\nabla^2 f(x) \succeq 0$, or we *jump out of it* if there exists a direction with a negative quadratic form value, $\nabla^2 f(x) \not\succeq 0$ (a strict saddle point or a strict local maximum).

Thus, iterations of the form (5.85) seem very attractive. However, notice that finding x^+ is not trivial, as the model $\Omega_H(x;y)$, as a function of y, is generally non-convex and might possess isolated local minima, as well as multiple global minima[5]. It is not a priori clear whether x^+ can be computed efficiently, which is the main question that we should ask to a new approach.

The following observations are important for making iterations (5.85) practical:

1. Notably, when the initial objective in (5.79) is *convex*, we know that $\nabla^2 f(x) \succeq 0$ for all $x \in \mathbb{R}^n$. Therefore, $\Omega_H(x;y)$ is a strictly convex function of y with a unique global minimizer x^+.

   Hence, we can apply any already known method from convex optimization to find x^+ (e.g., we can run the composite version of the fast gradient method, treating the cubic regularizer as the composite term, or the interior-point method, which requires building a suitable self-concordant barrier for the epigraph of the cubic model).

2. The optimality condition for (5.85) is the following non-linear equation[6]:

$$\nabla_y \Omega_H(x;x^+) = \nabla f(x) + \nabla^2 f(x)(x^+ - x) + \tfrac{H}{2} r B(x^+ - x) = 0, \tag{5.86}$$

[4] Indeed, $y := x$ already results in a zero value, and a slight perturbation along the linear term will yield a positive value for the maximization subproblem.

[5] When $\nabla f(x) = 0$ and $\lambda_{\min}(\nabla^2 f(x)) < 0$, we can *jump out* of the stationary point x in multiple ways (see Exercise 5.4.1).

[6] It is nonlinear due to the presence of r, which depends on x^+.

where $r := \|x^+ - x\|$. Thus, the global minimum x^+ must satisfy (5.86), though not every stationary point is the global minimum. In practice, using a stationary point that satisfies (5.86) in an algorithm might already be sufficient to ensure progress. Moreover, instead of an exact solution to (5.86), we can relax this condition to an *approximate stationary point*:

$$\nabla_y \Omega_H(x; x^+) \quad \approx \quad 0,$$

and use, for example, the gradient descent algorithm to find such x^+.

3. Fortunately, it appears that we can *always compute* the global solution x^+ to (5.85) by using the structure of the subproblem and linear algebra techniques, such as the SVD of the Hessian. We discuss this approach further in more detail.

Exercise 5.4.1. Assume $\nabla f(x) = 0$ and $\lambda_{\min} := \lambda_{\min}(\nabla^2 f(x)) < 0$. Show that the set of all global minimizers of (5.85) consists of vectors

$$x^+ \quad = \quad x \pm \tau h,$$

where h is an eigenvector of the Hessian corresponding to the smallest eigenvalue: $\nabla^2 f(x) h = \lambda_{\min} h$, and $\tau > 0$ is a step-size that depends on $\lambda_{\min}$ and the regularization parameter $H > 0$.

An immediate consequence of the optimality condition (5.86) is the following important lemma. Note that it works even for $H := 0$ (the pure Newton step).

Lemma 5.4.2. *For any $H \geq 0$, it holds*

$$\|\nabla f(x^+)\|_* \quad \leq \quad \tfrac{L+H}{2} r^2. \tag{5.87}$$

Proof. Substituting the optimality condition into the bound on the gradient approximation in Lemma 5.4.1, we get

$$\|\nabla f(x^+) + \tfrac{H}{2} r B(x^+ - x)\|_* \quad \overset{(5.86)}{=} \quad \|\nabla f(x^+) - \nabla f(x) - \nabla^2 f(x)(x^+ - x)\|_* \leq \tfrac{L}{2} r^2.$$

Using triangle inequality gives (5.87). □

Inequality (5.87) allows us to compare the length of the step $r := \|x^+ - x\|$ with the norm of the gradient at the *new point*.

Remark 5.4.3. Notice that when performing the gradient method step with a parameter $H > 0$:

$$\bar{x} \quad = \quad x - \tfrac{1}{H} B^{-1} \nabla f(x),$$

we have $\|\nabla f(x)\|_* = H \|\bar{x} - x\|$. Assuming that the gradient is Lipschitz with constant L_1, we obtain a similar bound to (5.87), but with a different power:

$$\|\nabla f(\bar{x})\|_* \quad \leq \quad \|\nabla f(x)\|_* + L_1 \|\bar{x} - x\| \quad = \quad (L_1 + H) \|\bar{x} - x\|.$$

The difference in the power leads to different convergence rates.

5.4.2 Local Quadratic Convergence for Strongly Convex Functions

Before moving on to the general non-convex case, let us verify that the cubic regularization of the quadratic Taylor model will *preserve the local quadratic convergence* of the pure Newton method.

The local quadratic convergence is the most distinguishing feature of Newton's method, and we are definitely interested to keep it.

We assume that f is a strongly convex function, for a positive parameter $\mu > 0$:

$$\nabla^2 f(x) \;\succeq\; \mu B, \qquad x \in \mathbb{R}^n. \tag{5.88}$$

Let us multiply the optimality condition (5.86), which in this case defines the unique global minimum x^+, by $\langle \cdot, x^+ - x \rangle$. Rearranging the terms, we obtain

$$r\|\nabla f(x)\|_* \;\geq\; \langle \nabla f(x), x^+ - x\rangle \overset{(5.86)}{=} \langle \nabla f(x)(x^+ - x), x^+ - x\rangle + \tfrac{H}{2} r^3 \overset{(5.88)}{\geq} \mu r^2.$$

From this inequality we get the following bound.

Lemma 5.4.4. *For any $H \geq 0$, it holds:*

$$r \;\leq\; \tfrac{1}{\mu}\|\nabla f(x)\|_*. \tag{5.89}$$

It remains to combine (5.89) with (5.87):

$$\|\nabla f(x^+)\|_* \;\leq\; \tfrac{L+H}{2} r^2 \;\leq\; \tfrac{L+H}{2\mu^2}\|\nabla f(x)\|_*^2, \tag{5.90}$$

and this is the local quadratic convergence! As soon as the initial gradient is sufficiently small, the next gradient will be quadratically smaller, and we can estimate the region of the quadratic convergence as follows.

Theorem 5.4.5. *For any $H \geq 0$, the cubic Newton method converges quadratically in the local region:*

$$\mathcal{Q} \;:=\; \Big\{ x \;:\; \|\nabla f(x)\|_* \;\leq\; \tfrac{\mu^2}{L+H} \Big\}. \tag{5.91}$$

Thus, starting from $x_0 \in \mathcal{Q}$ and performing the iterations $x_{k+1} = \operatorname{argmin}_{y \in \mathbb{R}^n} \Omega_H(x_k; y)$, $k \geq 0$, we get

$$\|\nabla f(x_k)\|_* \;\leq\; \varepsilon$$

after the following number of steps:

$$k \;=\; 1 + \Big\lceil \log_2 \log_2 \tfrac{2\mu^2}{(L+H)\varepsilon} \Big\rceil. \tag{5.92}$$

Proof. Indeed,

$$\tfrac{L+H}{2\mu^2}\|\nabla f(x_k)\|_* \overset{(5.90)}{\leq} \Big(\tfrac{L+H}{2\mu^2}\|\nabla f(x_{k-1})\|_*\Big)^2 \;\leq\; \ldots \;\leq\; \Big(\tfrac{L+H}{2\mu^2}\|\nabla f(x_0)\|_*\Big)^{2^k} \;\leq\; \Big(\tfrac{1}{2}\Big)^{2^k},$$

which gives (5.92). □

Note that this result holds for any $H \geq 0$, including $H := 0$. Therefore, we automatically reestablish the local quadratic convergence of the pure Newton method. Compared to the self-concordant analysis (Section 5.2.3), the result of Theorem 5.4.5 is no longer affine-invariant — as is the case with the cubic Newton method — since we fix the coordinate system through the operator B.

It is remarkable that the region (5.91) of quadratic convergence is of the same order as covered by the general self-concordant theory for Newton's method on strongly convex functions with Lipschitz Hessian (see Corollary 5.2.10 in Section 5.2.3).

Hence, we see that the cubic regularization of Newton's method "does not harm" the best local quadratic approximation provided by Taylor's polynomial $Q(x; y)$.

5.4.3 Non-Convex Quadratics and Strong Duality

Let us discuss how the cubic subproblem (5.85) can be solved globally, even in the non-convex case. To this end, we consider the following more general problem using simplified notation:

$$\min_{y\in\mathbb{R}^n}\Big\{\; P(y) \;\;=\;\; \langle g,y\rangle + \tfrac{1}{2}\langle Ay,y\rangle + \varphi\big(\langle By,y\rangle\big)\;\Big\}, \tag{5.93}$$

where $g\in\mathbb{R}^n$ is an arbitrary vector and $A=A^\top\in\mathbb{R}^{n\times n}$ is an arbitrary symmetric matrix, not necessary positive semidefinite, representing correspondingly the gradient and the Hessian from the cubic model.

As before, we assume that $B=B^\top\succ 0$, and φ is a non-decreasing univariate convex function defined on $\mathbb{R}_{\geq 0}$ and representing the regularizer. The most interesting examples are as follows:

1. *Cubic regularization* is covered by the choice $\varphi(s):=\frac{H}{6}s^{3/2}$. Indeed, substituting it into (5.93) and using that $\|y\|:=\langle By,y\rangle^{1/2}$ gives

$$P(y) \;\;=\;\; \langle g,y\rangle + \tfrac{1}{2}\langle Ay,y\rangle + \tfrac{H}{6}\|y\|^3. \tag{5.94}$$

2. *Trust-region approach.* For a given parameter $r\geq 0$ (the trust-region radius), we set

$$\varphi(s) \;\;:=\;\; \begin{cases} 0, & s\leq r \\ +\infty, & \text{otherwise.}\end{cases}$$

Substituting it into (5.93) we obtain the trust-region subproblem:

$$\min_{y\in\mathbb{R}^n\,:\,\|y\|\leq r}\Big\{\; \langle g,y\rangle + \tfrac{1}{2}\langle Ay,y\rangle\;\Big\}. \tag{5.95}$$

Trust-region methods are another popular way to globalize Newton's steps, especially for non-convex problems. They work by restricting the Taylor quadratic polynomial to a ball around the current iterate, the region where we "trust" our second-order model. Cubic and trust-region subproblems can be seen as equivalent, up to the choice of the parameters H and r.

We mainly focus on cubic regularization, as our assumption on the Lipschitz continuity of the Hessian (5.80) immediately leads to the natural choice of the regularization parameter $H:=L$. Historically, trust-region methods have been used intensively in practice, with a wide variety of efficient solvers developed specifically for subproblem (5.95). Given the shared structure (5.93), these efficient solvers can be used for the cubic case as well, with an appropriate change of φ.

3. *Quartic regularization.* One can think of other choices of φ, such as $\varphi(s)=\frac{H}{24}s^2$, which leads to the quartic model:

$$P(y) \;\;=\;\; \langle g,y\rangle + \tfrac{1}{2}\langle Ay,y\rangle + \tfrac{H}{24}\|y\|^4.$$

Thus, the global minimum of this function can be found with the same effort as in the cubic case (5.94).

For $\varphi:\mathbb{R}_{\geq 0}\to\mathbb{R}\cup\{+\infty\}$, we seek to use the following adjoint representation:

$$\varphi(s) \;\;=\;\; \max_{\tau\geq 0}\Big[\;\tau s - \varphi^\star(\tau)\;\Big], \tag{5.96}$$

where $\varphi^\star(\tau)$ is a convex function. A fundamental fact from convex analysis is that the function $\varphi^\star$ defined by (5.96) can be found as the *convex conjugate* of φ:

$$\varphi^\star(\tau) \;=\; \max_{s\geq 0}\Big[\, \tau s - \varphi(s) \,\Big].$$

In other words, Fenchel-Moreau duality holds: $\varphi^{\star\star}(s) = \varphi(s)$, when φ is "sufficiently good".

It is easy to check that for the case of cubic regularization, we use the following conjugate pair:

$$\varphi(s) \;=\; \tfrac{H}{6}s^{3/2} \qquad \text{and} \qquad \varphi^\star(\tau) \;=\; \tfrac{2^4}{3H^2}\tau^3.$$

Then, we can write the primal dual pair of problems[7]:

$$\begin{aligned}
\min_{y\in\mathbb{R}^n} P(y) \;&=\; \min_{y\in\mathbb{R}^n}\Big\{ \langle g, y\rangle + \tfrac{1}{2}\langle Ay, y\rangle + \varphi\big(\langle By, y\rangle\big) \Big\} \\
&\overset{(5.96)}{=} \min_{y\in\mathbb{R}^n}\max_{\tau\geq 0}\Big\{ \langle g, y\rangle + \tfrac{1}{2}\langle Ay, y\rangle + \tfrac{\tau}{2}\langle By, y\rangle - \varphi^\star\big(\tfrac{\tau}{2}\big) \Big\} \qquad (5.97)\\
&\geq \max_{\tau\geq 0}\min_{y\in\mathbb{R}^n}\Big\{ \langle g, y\rangle + \tfrac{1}{2}\langle Ay, y\rangle + \tfrac{\tau}{2}\langle By, y\rangle - \varphi^\star\big(\tfrac{\tau}{2}\big) \Big\} \;=\; \max_{\tau\in\mathcal{W}} D(\tau),
\end{aligned}$$

where

$$D(\tau) \;:=\; \min_{y\in\mathbb{R}^n}\Big\{ \langle g, y\rangle + \tfrac{1}{2}\langle (A+\tau B)y, y\rangle \Big\} - \varphi^\star\big(\tfrac{\tau}{2}\big) \;=\; -\tfrac{1}{2}\langle g, (A+\tau B)^{-1}g\rangle - \varphi^\star\big(\tfrac{\tau}{2}\big),$$

is a concave univariate function defined on the (possibly open) ray:

$$\mathcal{W} \;:=\; \{\tau \geq 0 \;:\; A + \tau B \;\succ\; 0\} \;=\; \{\tau \geq 0 \;:\; \tau > -\lambda_{\min}(B^{-1/2}AB^{-1/2})\}.$$

In fact, one can show that the *strong duality* holds in this case:

$$\boxed{\min_{y\in\mathbb{R}^n} P(y) \;=\; \max_{\tau\in\mathcal{W}} D(\tau)} \qquad (5.98)$$

This is remarkable, as the problem in the left-hand-side of (5.98) is non-convex, while in the right-hand-side we have a simple maximization of a univariate concave function.

It appears that the primal problem albeit non-convex in its current form, possesses *hidden convexity*. This can be viewed as a consequence of some fundamental facts about interactions of quadratic forms — one can show that the joint image of two quadratic forms

$$U \;:=\; \Big\{ [u_1, u_2]^\top \;=\; \Big[\, \langle g, y\rangle + \tfrac{1}{2}\langle Ay, y\rangle,\; \langle By, y\rangle \,\Big]^\top \;:\; y \in \mathbb{R}^n \Big\} \;\subseteq\; \mathbb{R}^2,$$

is a convex set in two-dimensional space. Therefore, the original non-convex primal problem can be rewritten as *convex minimization*:

$$\min_{y\in\mathbb{R}^n} P(y) \;=\; \min_{u\in U}\Big\{ u_1 + \varphi(u_2) \Big\},$$

while being an implicit formulation.

[7] We could use inf and sup here instead of min and max to be a bit more precise.

5.4.4 Solving Cubic Subproblem in Practice

Consider, for simplicity, $B := I$, using the standard Euclidean norm $\|\cdot\|_2$ in the regularization. The generalization to arbitrary $B \succ 0$ is straightforward. To compute the cubic Newton step (5.85):

$$x \;\mapsto\; x^+ \;\;=\;\; \operatorname*{argmin}_{y\in\mathbb{R}^n} \Omega_H(x;y),$$

we perform the following steps:

1. Simplify the quadratic part, by computing the *eigenvalue* (or *tridiagonal*) decomposition of the Hessian $\nabla^2 f(x)$:
$$\nabla^2 f(x) \;\;=\;\; U\Lambda U^\top,$$
where the matrix U is orthogonal: $UU^\top = I$, and the matrix Λ is diagonal (or tridiagonal). This can be done in $\mathcal{O}(n^3)$ arithmetic operations.

2. Then, we can solve the *univariate dual problem*
$$\max_{\tau\geq 0}\Big\{ -\tfrac{1}{2}\langle \nabla f(x), (\nabla^2 f(x)+\tau I)^{-1}\nabla f(x)\rangle - \tfrac{2}{3H^2}\tau^3 \;\;:\;\; \tau \;>\; -\lambda_{\min}(\nabla^2 f(x)) \Big\},$$
e.g., by finding the root of the equation $D'(\tau^\star) = 0$. This gives us the following non-linear equation to solve, using the reparametrization $\tau^\star \equiv \frac{H}{2} r^\star$:
$$h(r^\star) \;\;=\;\; \|s(r^\star)\|_2 - r^\star \;\;=\;\; 0, \tag{5.99}$$
where
$$s(r) \;\;:=\;\; \big(\nabla^2 f(x) + \tfrac{H}{2} r I\big)^{-1}\nabla f(x).$$
Note that $\|s(r)\|_2$ is a monotonically decreasing convex function that we want to intersect with the identity function in order to find $r^\star$. For solving (5.99), we can employ either binary search or univariate Newton's method, which will use $\tilde{\mathcal{O}}(n^2)$ arithmetic operations, hiding logarithmic factors.

3. After finding the root $r^\star$ of $h(\cdot)$, one step of the cubic Newton method can be written in the following explicit form:
$$\boxed{x^+ \;\;=\;\; x - \Big(\nabla^2 f(x) + \tfrac{H}{2} r^\star I\Big)^{-1}\nabla f(x),}$$
unless we are in the rare *degenerate* case: $\frac{H}{2} r^\star = -\lambda_{\min}\big(\nabla^2 f(x)\big)$, which corresponds to the situation when the supremum of the dual problem is achieved at the boundary of the open ray $\mathcal{W}$. This situation should be handled separately.

5.4.5 Main Inequalities for Cubic Newton Step

Let us consider one step $x \mapsto x^+$ of the cubic Newton method as applied to $f : \mathbb{R}^n \to \mathbb{R}$, defined by

$$\begin{aligned} x^+ \;\;&=\;\; \operatorname*{argmin}_{y\in\mathbb{R}^n} \Omega_H(x;y) \\ &=\;\; \operatorname*{argmin}_{y\in\mathbb{R}^n}\Big\{ \langle \nabla f(x), y-x\rangle + \tfrac{1}{2}\langle \nabla^2 f(x)(y-x), y-x\rangle + \tfrac{H}{6}\|y-x\|^3 \Big\}, \end{aligned}$$

and satisfying the first-order optimality condition:

$$\nabla f(x) + \nabla^2 f(x)(x^+ - x) + \tfrac{Hr}{2} B(x^+ - x) \;=\; 0, \tag{5.100}$$

where $r := \|x^+ - x\| = \langle B(x^+ - x), x^+ - x\rangle^{1/2}$ and $B = B^\top \succ 0$ is the matrix that defines the generalized Euclidean norm. In the non-degenerate case, the optimality condition (5.100) is equivalent to the step written in the following classic form:

$$x^+ \;=\; x - \left(\nabla^2 f(x) + \tfrac{Hr}{2} B\right)^{-1} \nabla f(x).$$

We have the following main inequalities that involve r:

1. By Lemma 5.4.2, for any $H \geq 0$, it holds:

$$\|\nabla f(x^+)\|_* \;\leq\; \tfrac{L+H}{2} r^2. \tag{5.101}$$

2. By properties of the solution to the cubic subproblem, we know that:

$$\nabla^2 f(x) + \tfrac{Hr}{2} B \;\succeq\; 0. \tag{5.102}$$

   Using the Lipschitz continuity of the Hessian, we conclude that

$$\tfrac{Hr}{2} B \overset{(5.102)}{\succeq} -\nabla^2 f(x) \;\succeq\; -LrB.$$

   Rearranging the terms, we get, for any $H \geq 0$:

$$r \;\geq\; \tfrac{2}{H+2L} \mu(x^+), \tag{5.103}$$

   where $\mu(x^+) := -\lambda_{\min}(B^{-1/2} \nabla^2 f(x^+) B^{-1/2})$.

Now, let us choose $H \geq L$, and substitute the optimality condition (5.100) into the global upper bound on the objective function (see Lemma 5.4.1), which was the main motivation for defining the cubic step:

$$\begin{aligned}
f(x^+) \;&\leq\; f(x) + \langle \nabla f(x), x^+ - x\rangle + \tfrac{1}{2}\langle \nabla^2 f(x)(x^+ - x), x^+ - x\rangle + \tfrac{H}{6} r^3 \\
&\overset{(5.100)}{=}\; f(x) - \tfrac{1}{2}\langle \nabla^2 f(x)(x^+ - x), x^+ - x\rangle - \tfrac{H}{2} r^3 + \tfrac{H}{6} r^3 \\
&=\; f(x) - \tfrac{1}{2}\langle \left[\nabla^2 f(x) + \tfrac{Hr}{2} B\right](x^+ - x), x^+ - x\rangle - \tfrac{H}{12} r^3 \\
&\overset{(5.102)}{\leq}\; f(x) - \tfrac{H}{12} r^3.
\end{aligned}$$

Therefore, we have established the progress in the function value:

Lemma 5.4.6. *For any $H \geq L$, it holds:*

$$f(x) - f(x^+) \;\geq\; \tfrac{H}{12} r^3. \tag{5.104}$$

Now, using that $H \geq L$, we have:

$$r \overset{(5.101)}{\geq} \left(\tfrac{2}{H+L} \|\nabla f(x^+)\|_*\right)^{1/2} \;\geq\; \left(\tfrac{1}{H} \|\nabla f(x^+)\|_*\right)^{1/2}$$

and

$$r \overset{(5.103)}{\geq} -\tfrac{2}{H+2L} \mu(x^+) \;\geq\; -\tfrac{2}{3H} \mu(x^+).$$

Combining the progress (5.104) with these lower bounds on r, we obtain:

Theorem 5.4.7. *Let $H \geq L$. Then,*

$$f(x) - f(x^+) \;\geq\; \max\left\{ \tfrac{1}{12 H^{1/2}} \|\nabla f(x^+)\|_*^{3/2},\; \tfrac{2}{3^4 H^2} \mu(x^+)^3 \right\}. \tag{5.105}$$

5.4.6 Convergence to Second-Order Stationary Point

Consider iterations of the cubic Newton method, starting from some $x_0 \in \mathbb{R}^n$:

$$x_{k+1} \;=\; \operatorname*{argmin}_{y \in \mathbb{R}^n} \Omega_L(x_k; y), \qquad k \geq 0. \tag{5.106}$$

where we fix the regularization parameter as $H := L$, for simplicity.

Then, for each iteration we have:

$$f(x_k) - f(x_{k+1}) \;\overset{(5.105)}{\geq}\; p_{k+1} \;:=\; \max\Big\{ \tfrac{1}{12L^{1/2}} \|\nabla f(x_{k+1})\|_*^{3/2},\; \tfrac{2}{3^4 L^2} \mu(x_{k+1})^3 \Big\}.$$

Telescoping it for the first $k \geq 1$ iterations:

$$f(x_0) - f^\star \;\geq\; f(x_0) - f(x_k) \;\geq\; k \cdot \tfrac{1}{k} \sum_{i=1}^{k} p_i \;\geq\; k \cdot \min_{1 \leq i \leq k} p_i,$$

we ensure the decrease: $\min_{1 \leq i \leq k} p_i = \mathcal{O}(1/k)$.

Therefore, we obtain the following global convergence rates:

Theorem 5.4.8. *For the iterations of the cubic Newton method* (5.106)*, we have:*

$$\min_{1 \leq i \leq k} \|\nabla f(x_i)\|_* \;\leq\; \Big(\tfrac{12 L^{1/2} (f(x_0) - f^\star)}{k} \Big)^{2/3}$$

and

$$\min_{1 \leq i \leq k} \mu(x_i) \;\leq\; \Big(\tfrac{3^4 L^2}{2} \cdot \tfrac{f(x_0) - f^\star}{k} \Big)^{1/3}.$$

5.5 Quasi-Self-Concordant Functions and Gradient Regularization

5.5.1 Motivational Example: Smoothness of Loss Functions

Consider the following canonical problem in a separable form (e.g., training a generalized linear models such as the logistic regression):

$$\min_{x \in \mathbb{R}^n} \Big[f(x) \;:=\; \sum_{i=1}^{m} \ell(\langle a_i, x \rangle) \Big] \tag{5.107}$$

where $\ell : \mathbb{R} \to \mathbb{R}$ is a differentiable convex loss function. To which problem class does this objective belong?

For that, we look at the derivatives. The first derivative linear form:

$$\langle \nabla f(x), h \rangle \;=\; \sum_{i=1}^{m} \ell'(\langle a_i, x \rangle) \langle a_i, h \rangle, \qquad x, h \in \mathbb{R}^n, \tag{5.108}$$

and the second derivative quadratic form:

$$\|h\|_x^2 \;\equiv\; \langle \nabla^2 f(x) h, h \rangle \;=\; \sum_{i=1}^{m} \ell''(\langle a_i, x \rangle) \langle a_i, h \rangle^2, \qquad x, h \in \mathbb{R}^n. \tag{5.109}$$

First-order methods. Assuming that the loss has a uniformly bounded second derivative, $\ell''(t) \leq L_1$, for all $t \in \mathbb{R}$, we can bound the Hessian of f as follows:

$$\langle \nabla^2 f(x)h, h\rangle \;\leq\; L_1 \sum_{i=1}^m \langle a_i, h\rangle^2 \;\equiv\; L_1 \langle Bh, h\rangle \;\equiv\; L_1 \|h\|_B^2 \;\leq\; L_1 \|B\| \cdot \|h\|_2^2, \tag{5.110}$$

where $B := \sum_{i=1}^m a_i a_i^\top$ is a symmetric positive semidefinite matrix. Without loss of generality, we can assume $B \succ 0$. Indeed, if for a certain direction $h \in \mathbb{R}^n$ it holds that $Bh = 0$, then both $\langle \nabla f(x), h\rangle = 0$ and $\langle \nabla^2 f(x)h, h\rangle = 0$ simultaneously, and a method will be automatically restricted to move only along the subspace spanned by $\{a_1, \ldots, a_m\}$.

Therefore, we conclude that f has the Lipschitz gradient, and we can apply first-order methods to (5.107). For example, the complexity of the basic gradient method to find a point x_k such that $f(x_k) - f^\star \leq \varepsilon$, starting from an arbitrary $x_0 \in \mathbb{R}^n$ is

$$O\Big(\tfrac{L_1 \|x_0 - x^\star\|_B^2}{\varepsilon}\Big) \;\leq\; O\Big(\tfrac{L_1 \|B\| \cdot \|x_0 - x^\star\|_2^2}{\varepsilon}\Big) \tag{5.111}$$

first-order oracle calls, correspondingly, when using the norm $\|\cdot\|_B$ or $\|\cdot\|_2$ in the method. These complexities can be improved by extracting the square root with the fast gradient method.

Second-order methods. What about second-order methods? For them, we need to bound the third derivative, for any $x, h, u \in \mathbb{R}^n$:

$$\begin{aligned} |D^3 f(x)[h,h,u]| \;&=\; \Big| \sum_{i=1}^m \ell'''(\langle a_i, x\rangle) \langle a_i, h\rangle^2 \cdot \langle a_i, u\rangle \Big| \\ &\leq\; \max_{1 \leq i \leq m} |\langle a_i, u\rangle| \cdot \sum_{i=1}^m |\ell'''(\langle a_i, x\rangle)| \langle a_i, h\rangle^2 \\ &\leq\; \|u\|_B \cdot \sum_{i=1}^m |\ell'''(\langle a_i, x\rangle)| \langle a_i, h\rangle^2, \end{aligned} \tag{5.112}$$

where we estimated the ℓ_∞-norm by the Euclidean one:

$$\max_{1 \leq i \leq m} |\langle a_i, u\rangle| \;\leq\; \sqrt{\sum_{i=1}^m \langle a_i, u\rangle^2} \;=\; \langle Bu, u\rangle^{1/2} \;=:\; \|u\|_B.$$

Thus, assuming that $\ell'''(t) \leq L_2$, for all $t \in \mathbb{R}$, we get the following uniform bound on the third derivative:

$$|D^3 f(x)[h,h,u]| \;\leq\; L_2 \|u\|_B \cdot \sum_{i=1}^m \langle a_i, h\rangle^2 \;=\; L_2 \cdot \|u\|_B \cdot \|h\|_B^2.$$

The last inequality implies that the Hessian of f is Lipschitz continuous. Hence, we can apply the Cubic Newton method for this problem, that possesses the following global complexity, on convex functions (see exam questions):

$$O\Big(\Big[\tfrac{L_2 D^3}{\varepsilon}\Big]^{1/2}\Big), \tag{5.113}$$

where $D \geq \|x_0 - x^\star\|_B$ is the size of the initial sublevel set measured in $\|\cdot\|_B$ norm:

$$D \;:=\; \max\Big\{ \|x - x^\star\|_B \;:\; f(x) \leq f(x_0) \Big\}. \tag{5.114}$$

We see that the complexity of the cubic Newton (5.113) is better than (5.111) of the gradient methods, in terms of the dependence on ε.

It is possible to accelerate the cubic Newton further (see Exercise 5.7.4).

Let us consider the following popular examples of the loss function.

Example 5.5.1 (Logistic Loss)**.**

$$\ell(t) \;=\; \ln(1+e^t),$$

we have

$$L_1 \;=\; \tfrac{1}{4}, \qquad L_2 \;=\; \tfrac{1}{6\sqrt{3}}.$$

However, computing these constants, we may observe the following interesting relationship:

$$\ell'''(t) \;=\; \ell''(t)\cdot(1-2\ell'(t)) \;=\; \ell''(t)\cdot\bigl(1-\tfrac{2}{1+e^{-t}}\bigr).$$

Hence, for logistic loss, it holds

$$|\ell'''(t)| \;\le\; \ell''(t), \qquad t\in\mathbb{R}. \tag{5.115}$$

Example 5.5.2 (Exponential Loss)**.**

$$\ell(t) \;=\; e^t.$$

Note that $L_1 = L_2 = +\infty$ (globally), while (5.115) is satisfied as an exact equation.

Using (5.115) in our computations, we obtain:

$$|D^3 f(x)[h,h,u]| \overset{(5.112),(5.115)}{\le} \|u\|_B \sum_{i=1}^m \ell''(\langle a_i,x\rangle)\langle a_i,h\rangle^2 \;=\; \|u\|_B\cdot\|h\|_x^2,$$

where $\|\cdot\|_x$ is the local norm induced by the Hessian (5.109). These observations motivate our next definition.

5.5.2 Quasi-Self-Concordant Functions

We consider a differentiable convex function $f: Q\to\mathbb{R}$, where $Q\subseteq\mathbb{R}^n$ is an open convex set. Without loss of generality, we can assume that $\nabla^2 f(x)\succ 0$ everywhere on Q. As usual, we denote by

$$\|h\|_x \;:=\; \langle\nabla^2 f(x)h,h\rangle^{1/2}, \qquad h\in\mathbb{R}^n,$$

the local norm at $x\in Q$ induced by the Hessian, and by $\|h\|$ we denote a fixed global norm. The main example for us is when the global norm is induced by a fixed positive definite operator $B=B^\top\succ 0$:

$$\|h\| \;:=\; \langle Bh,h\rangle^{1/2}, \qquad h\in\mathbb{R}^n.$$

Definition. We say that a function $f: Q\to\mathbb{R}$ is *quasi-self-concordant* with constant $M\ge 0$, if

$$D^3 f(x)[h,h,u] \;\le\; M\|h\|_x^2\|u\|, \qquad \forall h,u\in\mathbb{R}^n, x\in Q. \tag{5.116}$$

The difference from classic self-concordant functions (see Section 5.2.1) is that we replace the local norm for u with the global fixed norm in the right-hand side of (5.116). Hence, the new problem class is no longer affine-invariant (the parameter M changes, if we change the coordinate system).

This definition can be seen as an intermediate problem class between self-concordant functions and the functions with Lipschitz continuous Hessians.

We see that logistic and exponential regression objectives satisfy assumption (5.116) with $\boxed{M=1}$ by choosing the matrix $B=A^\top A$ as in the previous section, where A is the matrix representing input data, or $\boxed{M=\|A\|}$ when $B:=I$. It is possible to show that they are not self-concordant in the classic sense:

Exercise 5.5.1. Show that both $\ell(t)=\ln(1+e^t)$ and $\ell(t)=e^t$ are not self-concordant on $\mathbb{R}$, i.e., in each of these cases, there is no constant $M\ge 0$ such that

$$|\ell'''(t)| \;\le\; M\bigl(\ell''(t)\bigr)^{3/2}, \qquad \forall t\in\mathbb{R}.$$

Main properties. Let us take an arbitrary direction $h \in \mathbb{R}^n$ and consider how the local norm of u changes between two given points x and y. In particular, we look at the function

$$g(t) \;=\; \ln \|h\|^2_{x+t(y-x)} \;=\; \ln\langle \nabla^2 f(x+t(y-x))h, h\rangle, \qquad t \in [0,1].$$

Then,

$$\left|g'(t)\right| \;=\; \left|\tfrac{D^3 f(x+t(y-x))[h]^2[y-x]}{\|h\|^2_{x+t(y-x)}}\right| \overset{(5.116)}{\leq} M\|y-x\|.$$

Therefore,

$$\left|\ln \tfrac{\|h\|^2_y}{\|h\|^2_x}\right| \;=\; |g(1)-g(0)| \;=\; |\int\limits_0^1 g'(t)dt| \;\leq\; M\|y-x\|.$$

Hence, taking the exponent:

$$\|h\|^2_x e^{-M\|y-x\|} \;\leq\; \|h\|^2_y \;\leq\; \|h\|^2_x e^{M\|y-x\|}.$$

We have established the following main lemma, which is an analog of the Hessian stability for the quasi-self-concordant functions (compare with Lemma 5.2.5 from Section 5.2.2 on self-concordant functions, and with that one (5.81) from Section 5.4 for the functions with Lipschitz Hessian):

Lemma 5.5.3. *For any $x, y \in \mathbb{R}^n$:*

$$\nabla^2 f(x) e^{-M\|y-x\|} \;\preceq\; \nabla^2 f(y) \;\preceq\; \nabla^2 f(x) e^{M\|y-x\|}. \tag{5.117}$$

Let us derive a consequence of our definition for the approximation of the gradient norm. For any direction $h \in \mathbb{R}^n$, s.t. $\|h\| \leq 1$, we have, using the Taylor theorem:

$$\begin{aligned}
\langle \nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x), h\rangle \;&=\; \int\limits_0^1 (1-t) D^3 f(x+t(y-x))[y-x]^2[h]dt \\
&\overset{(5.116)}{\leq}\; M\|h\| \cdot \int\limits_0^1 (1-t)\|y-x\|^2_{x+t(y-x)}dt \\
&\overset{(5.117)}{\leq}\; M\|y-x\|^2_x \cdot \int\limits_0^1 (1-t) e^{tM\|y-x\|}d\tau \\
&\equiv\; M\|y-x\|^2_x \cdot \varphi(M\|y-x\|).
\end{aligned}$$

By computing the integral, we obtain the following useful bound on the linear approximation of the gradient.

Lemma 5.5.4. *For any $x, y \in \mathbb{R}^n$:*

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x)\|_* \;\leq\; M\|y-x\|^2_x \cdot \varphi(M\|y-x\|), \tag{5.118}$$

where $\varphi(t) := \frac{e^t - t - 1}{t^2} \geq 0$ is a monotone convex function (see Fig. 5.3, left).

Integrating (5.118) once more yields global upper and lower approximation bounds for an objective function (see Fig. 5.3, right).

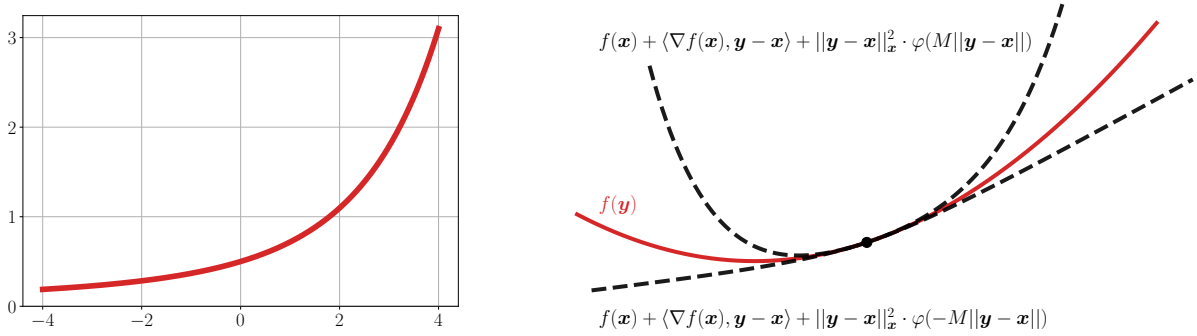

Figure 5.3: **Left:** The graph of $\varphi(t) = \frac{e^t - t - 1}{t^2}$ from the bound (5.118). **Right:** Global upper and lower models of a quasi-self-concordant function.

5.5.3 Gradient Regularization of Newton's Method

The cubic regularization of Newton's method is a very powerful approach as it works both for convex and non-convex functions, achieving superior rates to those of the first-order methods.

However, when solving convex problems, we can replace the nonlinear cubic subproblem by the quadratic regularization, which is easier to implement as each step requires solving only one linear system.

Recall that one step of the cubic Newton method with the regularization parameter $L \geq 0$ can be written in the following form (see Section 5.4.4):

$$x^+ \quad = \quad x - \left(\nabla^2 f(x) + \tfrac{Lr^\star}{2} B\right)^{-1} \nabla f(x), \tag{5.119}$$

where $r^\star$ is the solution of the nonlinear univariate equation, and we have $r^\star \approx \sqrt{\frac{1}{L}\|\nabla f(x^+)\|_*}$.

The idea of the *gradient regularization* is to replace the implicit regularization coefficient $r^\star$ with the current gradient norm $\|\nabla f(x)\|_*$, which can be easily computed at the current point x. Indeed, using the fact that $\nabla^2 f(x) \succeq 0$ for convex objectives, we can bound the length of the step from (5.119), as follows:

$$r^\star \quad := \quad \|x^+ - x\| \quad \overset{(5.119)}{=} \quad \left\| \left(\nabla^2 f(x) + \tfrac{Lr^\star}{2} B\right)^{-1} \nabla f(x) \right\| \quad \leq \quad \tfrac{2}{Lr^\star} \|\nabla f(x)\|_*. \tag{5.120}$$

Hence, we obtain the upper bound:

$$r^\star \quad \overset{(5.120)}{\leq} \quad \sqrt{\tfrac{2}{L}\|\nabla f(x)\|_*}, \tag{5.121}$$

which we can use in (5.119) instead of $r^\star$. It appears that such an approximation preserves the fast global rates of the cubic Newton method [26, 11], for convex functions.

In general, we can consider iterations of the form:

$$x^+ \quad = \quad x - \left(\nabla^2 f(x) + H\|\nabla f(x)\|_*^\alpha B\right)^{-1} \nabla f(x), \tag{5.122}$$

where $0 \leq \alpha \leq 1$ is some fixed power, and $H \geq 0$ is a regularization parameter. Then, $\alpha = 0$ implies that we regularize the Hessian by a constant matrix, while substituting the upper bound (5.121) into (5.119) corresponds to $\alpha = 1/2$ and $H = \sqrt{2L}$.

Among the possible powers, the most appealing is $\boxed{\alpha = 1}$, as it preserves the *local quadratic convergence* of Newton's method, to be shown in the following exercise. This is the choice that we will analyze further for quasi-self-concordant functions.

Exercise 5.5.2. Consider step (5.122) for $\alpha = 1$ and some fixed $H \geq 0$. Assume that the function f is strongly convex, with a Lipschitz continuous Hessian (with corresponding parameters μ and L). Show that

$$\|\nabla f(x^+)\|_* \quad \leq \quad \left(\frac{L}{2\mu^2} + \frac{H}{\mu}\right)\|\nabla f(x)\|_*^2.$$

Therefore, the method possesses local quadratic convergence. What will be the local rate of the method with arbitrary $0 \leq \alpha \leq 1$?

5.5.4 Global Linear Rate

Using $\alpha = 1$, iteration (5.122) can be rewritten as the solution to the linear system:

$$\nabla f(x) + \nabla^2 f(x)(x^+ - x) + H\|\nabla f(x)\|_* B(x^+ - x) \quad = \quad 0. \tag{5.123}$$

Taking the inner product with $x^+ - x$ and rearranging the terms, we get

$$\|x^+ - x\|_x^2 + H\|\nabla f(x)\|_*\|x^+ - x\|^2 \quad = \quad \langle \nabla f(x), x - x^+\rangle \quad \leq \quad \|\nabla f(x)\|_*\|x^+ - x\|. \tag{5.124}$$

Dropping either the first or the second term, which are nonnegative, we obtain the following bounds:

Lemma 5.5.5. *It holds:*

$$\|x^+ - x\| \quad \leq \quad \tfrac{1}{H}. \tag{5.125}$$

and

$$\|x^+ - x\|_x^2 \quad \leq \quad \|\nabla f(x)\|_*\|x^+ - x\|. \tag{5.126}$$

Consequently, by performing gradient regularization, we automatically ensure that the iterates remain within the ball of radius $\frac{1}{H}$ centered at x. Furthermore, by (5.126), we can also control the radius of the ball in the local norm (the the radius of Dikin's ellipsoid).

Progress of one step. Now, let us fix for simplicity $\boxed{H := M}$ — so we choose the regularization parameter to be exactly the constant of quasi-self-concordance. We combine the optimality condition (5.123) with our bound on the gradient approximation (5.118). Denote $g := \|\nabla f(x)\|_*$ and $r = \|x^+ - x\|$. First, note that

$$\varphi(M\|x^+ - x\|) \quad \leq \quad \varphi(\tfrac{M}{H}) \quad = \quad \varphi(1) \quad = \quad \rho = e - 2 \quad \approx \quad 0.718281828.$$

$$\begin{aligned} \|\nabla f(x^+) + MgB(x^+ - x)\|_* \quad &\leq \quad M\|x^+ - x\|_x^2 \cdot \varphi(M\|x^+ - x\|) \\ &\leq \quad \rho M g\|x^+ - x\|. \end{aligned}$$

Squaring both sides, we obtain:

$$g_+^2 + (Mgr)^2 + 2Mg\langle \nabla f(x^+), x^+ - x\rangle \quad \leq \quad \rho^2 (Mgr)^2, \tag{5.127}$$

where $g_+ := \|\nabla f(x_+)\|_*$. Using the fact that $\rho < 1$, we obtain the following progress of one iteration.

Theorem 5.5.6. *For one Newton's step with gradient regularization, we have:*

$$f(x) - f(x^+) \quad \geq \quad \langle \nabla f(x^+), x - x^+\rangle \quad \overset{(5.127)}{\geq} \quad \tfrac{1}{2Mg} g_+^2 \quad = \quad \tfrac{1}{2M}\left(\tfrac{\|\nabla f(x_+)\|_*}{\|\nabla f(x)\|_*}\right)^2 \|\nabla f(x)\|_* \tag{5.128}$$

Global linear rate. Let us derive the rate of convergence from (5.128). We perform the following simple iterations:

$$x_{k+1} \;=\; x_k - \Big(\nabla^2 f(x_k) + M\|\nabla f(x_k)\|_* B\Big)^{-1} \nabla f(x_k), \qquad k \geq 0, \tag{5.129}$$

starting from an arbitrary initialization $x_0 \in \mathbb{R}^n$.

Denote the functional residual as $F_k := f(x_k) - f^\star$ and the gradient norm as $g_k := \|\nabla f(x_k)\|_*$. By convexity, we have

$$g_k \;\geq\; \tfrac{F_k}{D}, \tag{5.130}$$

where D is the diameter of the initial sublevel set as in (5.114). Substituting this into the progress of one step, we get:

$$F_k - F_{k+1} \overset{(5.128)}{\geq} \tfrac{1}{2M}\Big(\tfrac{g_{k+1}}{g_k}\Big)^2 g_k \overset{(5.130)}{\geq} \tfrac{1}{2MD}\Big(\tfrac{g_{k+1}}{g_k}\Big)^2 F_k. \tag{5.131}$$

It remains to derive the convergence rate from the recurrence (5.131). Rearranging the terms in (5.131), we see that

$$F_{k+1} \;\leq\; \Big[1 - \tfrac{1}{2MD}\Big(\tfrac{g_{k+1}}{g_k}\Big)^2\Big] \cdot F_k \;\approx\; \Big[1 - \tfrac{1}{2MD}\Big] \cdot F_k.$$

Thus, we can expect a linear rate of decrease for the sequence F_k, which suggests that the *appropriate quantity to telescope*[8] is $\ln(F_k)$.

We know that $\ln(a)$ is a concave function. Hence, for any $a, b > 0$:

$$\ln(a) \;\leq\; \ln(b) + \tfrac{1}{b}(a-b) \qquad \Leftrightarrow \qquad \ln(b) - \ln(a) \;\geq\; \tfrac{1}{b}(b-a), \tag{5.132}$$

Therefore,

$$\ln(F_k) - \ln(F_{k+1}) \overset{(5.132)}{\geq} \tfrac{F_k - F_{k+1}}{F_k} \overset{(5.131)}{\geq} \tfrac{1}{2MD}\Big(\tfrac{g_{k+1}}{g_k}\Big)^2. \tag{5.133}$$

Telescoping this bound, and using the inequality between arithmetic and geometric means (that is, Jensen's inequality for concavity of the logarithm), we get:

$$\begin{aligned}
\ln \tfrac{F_0}{F_k} &\overset{(5.133)}{\geq} \tfrac{k}{2MD} \cdot \tfrac{1}{k} \sum_{i=0}^{k-1} \Big[\tfrac{g_{i+1}}{g_i}\Big]^2 \;\geq\; \tfrac{k}{2MD} \cdot \Big[\prod_{i=0}^{k-1} \tfrac{g_{i+1}}{g_i}\Big]^{2/k} \\
&= \tfrac{k}{2MD} \cdot \Big[\tfrac{g_k}{g_0}\Big]^{2/k} \;=\; \tfrac{k}{2MD} \cdot \exp\Big(\tfrac{2}{k} \ln \tfrac{g_k}{g_0}\Big) \\
&\overset{(*)}{\geq} \tfrac{k}{2MD} \cdot \Big(1 + \tfrac{2}{k} \ln \tfrac{g_k}{g_0}\Big) \overset{(5.130)}{\geq} \tfrac{k}{2MD} \cdot \Big(1 + \tfrac{2}{k} \ln \tfrac{F_k}{g_0 D}\Big),
\end{aligned} \tag{5.134}$$

where in $(*)$ we used that $e^t \geq 1 + t$ for all $t \in \mathbb{R}$, which follows from convexity of e^t.

Consider two cases.

[8] Note that in the continuous-time case, the recurrence (5.131) takes the form $-\dot{F}_t \geq c \cdot F_t$ for a constant $c \geq 0$, which can be integrated to:

$$\ln \tfrac{F_0}{F_t} \;=\; \ln F_0 - \ln F_t \;=\; \int_0^t \tfrac{d}{dt}\big[-\ln F_t\big] \;=\; \int_0^t -\tfrac{\dot{F}_t}{F_t} dt \;\geq\; ct \qquad \Rightarrow \qquad F_t = O(F_0 e^{-ct}).$$

Integrating in continuous time corresponds to telescoping discrete sequences.

1. Either $\frac{2}{k} \ln \frac{F_k}{g_0 D} \leq -\frac{1}{2}$, which is equivalent to the very fast rate with constant factor:

$$F_k \;\leq\; \exp(-k/4) g_0 D.$$

2. Otherwise, $\frac{2}{k} \ln \frac{F_k}{g_0 D} \geq -\frac{1}{2}$. Substituting this bound into (5.134), gives

$$\ln \tfrac{F_0}{F_k} \;\geq\; \tfrac{k}{4MD} \qquad \Leftrightarrow \qquad F_k \;\leq\; \exp\Bigl(-\tfrac{k}{4MD}\Bigr) F_0.$$

Finally, we combine these two bounds together to obtain the following convergence rate.

Theorem 5.5.7. *For iterations of Newton's method with gradient regularization* (5.129)*, we have the global linear rate:*

$$f(x_k) - f^\star \;\leq\; \exp\Bigl(-\tfrac{k}{4MD}\Bigr)(f(x_0) - f^\star) + \exp\Bigl(-\tfrac{k}{4}\Bigr) g_0 D. \tag{5.135}$$

Therefore, in order to obtain $f(x_k) - f^\star \leq \varepsilon$ it is enough to perform the following number of iterations (second-order oracle calls):

$$k \;=\; O\Bigl(MD \ln \tfrac{F_0}{\varepsilon} + \ln \tfrac{g_0 D}{\varepsilon}\Bigr). \tag{5.136}$$

To establish (5.135), we did not use any additional assumptions, such as strong or uniform convexity, other than our main assumption of quasi-self-concordance (5.116).

The complexity bound (5.136) is superior to those of the gradient methods (5.111) and the cubically regularized Newton method (5.113) in terms of the final dependence on the target accuracy $\varepsilon > 0$. Note that in each of these situations, we are discussing not only different methods but, more importantly, *different problem classes*. For the basic (non-accelerated) methods, we have the following complexity picture:

- *Convex functions with Lipschitz gradient:* $O(1/\varepsilon)$
- *Convex functions with Lipschitz Hessian:* $O(1/\varepsilon^{1/2})$
- *Quasi-self-concordant functions:* $O(\ln \frac{1}{\varepsilon})$

At the same time, a single objective function can belong to multiple problem classes simultaneously. Therefore, for a given problem, we are primarily interested in the best possible convergence rate among the available options. For example, for training logistic regression, the quasi-self-concordant framework appears to provide the best global complexity (5.136) among those considered.

While we used a constant choice for the regularization parameter in (5.129), which requires knowing the constant of quasi-self-concordance, we can instead perform a simple adaptive search. This is analogous to the adaptive search used in gradient methods. Such an adaptive search will ensure sufficient progress (5.128) at each iteration. In fact, it was shown in [9] that employing the adaptive search allows the Newton method with gradient regularization to *automatically* achieve the best convergence rate among all the problem classes listed above, yielding *super-universal* guarantees.

5.6 Contracting-Point Acceleration

We discuss a conceptual acceleration scheme that can be used to potentially accelerate *any* optimization algorithm, including sophisticated ones (e.g., stochastic methods, such as coordinate descent or methods with variance reduction, as well as second-order algorithms).

While direct acceleration (i.e., acceleration developed for a specific method) is usually preferable from a practical standpoint, the conceptual scheme that we will discuss is useful for quickly identifying the expected rate of convergence one should aim for, while remaining remarkably simple.

Problem formulation. We consider the minimization of a convex function $f : Q \to \mathbb{R}$ defined on an open convex set $Q \subseteq \mathbb{R}^n$:

$$\min_{x \in Q} f(x), \tag{5.137}$$

and we assume that a minimizer $x^\star$ exists.

Thus far, we do not assume any additional conditions on the objective, such as smoothness, although such conditions are often crucial for acceleration. Recall that the subgradient method is optimal for the black-box minimization of non-differentiable Lipschitz convex functions; therefore, acceleration is not possible for every combination of algorithm and problem class.

To solve (5.137), we fix a differentiable convex *regularizer* $d : Q \to \mathbb{R}$ and define the associated *Bregman divergence*:

$$\beta_d(x; y) \;\; := \;\; d(y) - d(x) - \langle \nabla d(x), y - x \rangle \;\; \geq \;\; 0,$$

which serves as a measure of the distance from y to x.

We use the following main fact about the Bregman divergence (see Lemma 4.8.5 in Section 4.8.2). For any convex $g : Q \to \mathbb{R}$, consider the solution to the regularized subproblem for a fixed center $v \in Q$:

$$v^+ \;\; := \;\; \operatorname*{argmin}_{y \in Q} \Big\{ g(y) + \beta_d(v; y) \Big\}$$

Then, it holds:

$$g(y) + \beta_d(v; y) \;\; \geq \;\; g(v^+) + \beta_d(v; v^+) + \beta_d(v^+; y), \qquad y \in Q. \tag{5.138}$$

Inequality (5.138) improves upon the trivial bound that holds by the definition of the minimum, due tp additional non-negative term $\beta_d(v^+; y) \geq 0$, which is very useful for the analysis.

5.6.1 Contracting-Point Scheme

In the accelerated method, we construct two sequences of points, both starting from some initialization $x_0 = v_0 \in Q$:

- An auxiliary sequence of *prox centers* $\{v_k\}_{k \geq 0}$,
- A sequence of *main iterates* $\{x_k\}_{k \geq 0}$.

We also have an increasing sequence of controlling parameters $\{A_k\}_{k \geq 0}$, starting from $A_0 = 0$. We denote the partial differences by:

$$a_{k+1} \;\; := \;\; A_{k+1} - A_k \;\; > \;\; 0 \qquad \Leftrightarrow \qquad A_k \;\; = \;\; \sum_{i=1}^{k} a_i,$$

and the *contracting coefficients*:

$$\gamma_k \;:=\; \tfrac{a_{k+1}}{A_{k+1}} \;\in\; (0,1].$$

Then, our goal is to ensure the following inequality, for any $k \geq 0$:

$$\beta_d(x_0;x) + A_k f(x) \;\geq\; \beta_d(v_k;x) + A_k f(x_k), \qquad x \in Q. \tag{5.139}$$

Note that plugging $x := x^\star$ into (5.139) and rearranging the terms, we obtain the following convergence rate:

$$f(x_k) - f^\star \;\leq\; \tfrac{\beta_d(x_0;x^\star)}{A_k}, \qquad k \geq 1, \tag{5.140}$$

and, therefore, we are interested to increase $A_k \to +\infty$ as fast as possible.

It is easy to check that inequality (5.139) holds for $k = 0$ due to our choices: $A_0 = 0$ and $x_0 = v_0$. Now, we assume that it holds for a current iteration $k \geq 0$ and see how we can propagate this inequality for the next iteration. We have,

$$\begin{aligned}
\beta_d(x_0;x) + A_{k+1} f(x) \;&=\; \beta_d(x_0;x) + A_k f(x) + a_{k+1} f(x) \\
&\overset{(5.139)}{\geq}\; \beta_d(v_k;x) + A_k f(x_k) + a_{k+1} f(x) \\
&\geq\; \beta_d(v_k;x) + A_{k+1} f\big(\gamma_k x + (1-\gamma_k) x_k\big),
\end{aligned} \tag{5.141}$$

where in the last inequality we used convexity of f. Let us denote by v_{k+1} the minimum of the right-hand side of (5.141):

$$v_{k+1} \;:=\; \operatorname*{argmin}_{x \in Q} \Big\{ A_{k+1} f\big(\gamma_k x + (1-\gamma_k) x_k\big) + \beta_d(v_k;x) \Big\}. \tag{5.142}$$

Applying the main Bregman divergence inequality (5.138), we obtain:

$$\begin{aligned}
\beta_d(x_0;x) + A_{k+1} f(x) \;&\overset{(5.141)}{\geq}\; \beta_d(v_k;x) + A_{k+1} f\big(\gamma_k x + (1-\gamma_k) x_k\big) \\
&\overset{(5.138)}{\geq}\; \beta_d(v_k;v_{k+1}) + A_{k+1} f\big(\gamma_k v_{k+1} + (1-\gamma_k) x_k\big) + \beta_d(v_{k+1};x) \\
&\geq\; \beta_d(v_{k+1};x) + A_{k+1} f(x_{k+1}),
\end{aligned}$$

where in the last inequality we dropped[9] the non-negative term $\beta_d(v_k;v_{k+1}) \geq 0$, and set the next main iterate as

$$x_{k+1} \;:=\; \gamma v_{k+1} + (1-\gamma_k) x_k.$$

Therefore, we established (5.139) for the next iteration, and thus proved it by induction for all $k \geq 0$.

We can write down contracting-point iterations in algorithmic form.

[9] To obtain the fastest possible rate, it is actually better to keep all the terms, which we omit here for simplicity.

Algorithm 5.2: *Contracting-Point Scheme for Acceleration.*

Initialization: $x_0 \in \mathbb{R}^n$. Choose regularizer $d(\cdot)$. Set $v_0 = x_0$ and $A_0 = 0$. Fix $K \geq 1$.

For $k = 0 \dots K-1$ **iterate:**

1. Choose a new coefficient $a_{k+1} > 0$. Set $A_{k+1} := A_k + a_{k+1}$ and $\gamma_k := \frac{a_{k+1}}{A_{k+1}}$
2. Form the contracted objective with Bregman regularization:
$$h_k(x) \;:=\; A_{k+1} f\big(\gamma_k x + (1-\gamma_k)x_k\big) + \beta_d(v_k; x)$$
3. Compute
$$v_{k+1} \;\approx\; \operatorname*{argmin}_{x \in Q} h_k(x)$$
4. Set a new point from the triangle rule: $x_{k+1} := \gamma_k v_{k+1} + (1-\gamma_k)x_k$

Return x_K

From our previous reasoning, we obtain the following convergence result.

Theorem 5.6.1. *Let v_{k+1} be the exact minimizer of $h_k(\cdot)$. Then, we have*

$$f(x_k) - f^\star \;\leq\; \tfrac{\beta_d(x_0; x^\star)}{A_k}, \qquad k \geq 1. \tag{5.143}$$

A similar convergence rate can be established when v_{k+1} is an approximate minimizer of $h_k(\cdot)$ with sufficient accuracy [10].

Note that the classic fast gradient method can be viewed as an instance of this scheme, where we use the Euclidean prox function $d(x) = \frac{1}{2}\|x\|_2^2$ and, in Step 3, additionally linearize the contracted objective $f(\gamma_k x + (1-\gamma_k)x_k)$ around the point v_k:

$$\begin{aligned} h_k(x) \;&=\; A_{k+1} f\big(\gamma_k x + (1-\gamma_k)x_k\big) + \tfrac{1}{2}\|x - v_k\|^2 \\ &\approx\; A_{k+1}\Big[f(y_k) + \gamma_k \langle \nabla f(y_k), x - v_k\rangle\Big] + \tfrac{1}{2}\|x - v_k\|^2, \end{aligned} \tag{5.144}$$

where $y_k := \gamma_k v_k + (1-\gamma_k)x_k$. In the fast gradient method, we then set v_{k+1} to be the minimizer of the right-hand side of (5.144).

5.6.2 Example: Acceleration of First-Order Methods

Let us consider an example of using the general contracting-point scheme to accelerate the basic gradient method.

The first crucial choice is to fix the regularizer $d(x)$. The simplest one is the square of the Euclidean norm:

$$d(x) \;:=\; \tfrac{1}{2}\|x\|_2^2,$$

which makes the Bregman divergence to be:

$$\beta_d(x; y) \;=\; \tfrac{1}{2}\|y - x\|_2^2.$$

At step 3 of the algorithm, we need to solve the following subproblem:

$$\min_{x \in Q}\Big\{\; h_k(x) \;:=\; g_k(x) + \tfrac{1}{2}\|x - v_k\|_2^2 \;\Big\}, \tag{5.145}$$

where

$$g_k(x) \;:=\; A_{k+1} f\big(\gamma_k x + (1-\gamma_k)x_k\big).$$

it the *contracted* objective, which gives the name to the whole scheme. Notice that

$$\nabla h_k(x) \;=\; a_{k+1}\nabla f\big(\gamma_k x + (1-\gamma_k)x_k\big) + (x - v_k),$$

$$\nabla^2 h_k(x) \;=\; \tfrac{a_{k+1}^2}{A_{k+1}}\nabla^2 f\big(\gamma_k x + (1-\gamma_k)x_k\big) + I.$$

Now, assume that f has the Lipschitz continuous gradient with constant L_f, with respect to the Euclidean norm. Then,

$$I \;\preceq\; \nabla^2 h_k(x) \;\preceq\; \Big(\tfrac{a_{k+1}^2}{A_{k+1}} L_f + 1\Big) I,$$

and we conclude that $h_k(\cdot)$ is *strongly convex* with parameter $\mu_k := 1$ and it has the Lipschitz gradient with parameter $L_k := \frac{a_{k+1}^2}{A_{k+1}} L_f + 1$.

We know that the basic gradient method, as applied to (5.145), will then exhibit a linear rate of convergence, and the main complexity factor will be the *condition number*:

$$\tfrac{L_k}{\mu_k} \;=\; \tfrac{a_{k+1}^2}{A_{k+1}} L_f + 1. \tag{5.146}$$

Therefore, by choosing a_{k+1} in a smart manner we can ensure that the condition number (5.146) is an absolute constant. For example, we can find a_{k+1} from the quadratic equation:

$$\boxed{\tfrac{a_{k+1}^2}{A_{k+1}} \;=\; \tfrac{1}{L_f},} \tag{5.147}$$

which makes $\frac{L_k}{\mu_k} = 2$. Note that equation (5.147) is exactly the one we used in deriving the rate of the fast gradient method (Section 3.84), which leads to the following rate of growth of the controlling coefficients:

$$A_k \;\geq\; \tfrac{k^2}{4L_f}. \tag{5.148}$$

We conclude that the resulting contracting-point scheme will have the accelerated optimal rate:

$$f(x_k) - f^\star \overset{(5.143),(5.148)}{\leq} \tfrac{2L_f \|x_0 - x^\star\|_2^2}{k^2}, \qquad k \geq 1, \tag{5.149}$$

while each subproblem in Step 3 can be solved in $\tilde{O}(1)$ iterations of the gradient method, where $\tilde{O}(\cdot)$ notation hides logarithmic factors.

Compared to direct acceleration, the fast gradient method performs exactly one gradient step per iteration, achieving the same optimal rate (5.149). An extra logarithmic factor seems to be a reasonable price to pay for the generality. Utilizing the same reasoning, we can obtain acceleration for second-order methods (see Exercise 5.7.4).

5.7 Exercises

On Self-Concordant and Quasi-Self-Concordant Functions

Exercise 5.7.1. Assume that $f : Q \to \mathbb{R}$ is *self-concordant* with constant $M_f \geq 0$.

- Let $g(y) = f(Ay + b)$, where $A \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^n$. Show that g is self-concordant with the same constant $M_g = M_f$.
- Let $g(x) = cf(x)$, for $c > 0$. What will be the constant of self-concordance M_g for g? Show that for $M_f > 0$, we can always choose c such that $M_g = 2$ (so the function g is "standard self-concordant" after an appropriate rescaling).

Exercise 5.7.2. Assume that $f : Q \to \mathbb{R}$ is *quasi-self-concordant* with constant $M_f \geq 0$. What will be the constant of quasi-self-concordance M_g after each of the following transformations, as in the previous exercise:

- Let $g(y) = f(Ay + b)$ (affine substitution)?
- Let $g(x) = cf(x)$ for $c > 0$ (scaling)?

Exercise 5.7.3. For each of the following univariate functions, indicate whether its either *self-concordant*, *quasi-self-concordant*, *both*, or *neither*. If yes, show a possible constant $M_f \geq 0$:

- $f(x) = x^4$, $x \in \mathbb{R}$;
- $f(x) = x^4 + x^2$, $x \in \mathbb{R}$;
- $f(x) = e^x$, $x \in \mathbb{R}$;
- $f(x) = \ln(1 + e^x)$, $x \in \mathbb{R}$;
- $f(x) = \frac{1}{x}$, $x > 0$;
- $f(x) = \frac{1}{x} + \frac{x^2}{2}$, $x > 0$.

On Acceleration of Cubic Newton

Exercise 5.7.4. Consider the problem of unconstrained minimization of a *convex* differentiable function:

$$\min_{x \in \mathbb{R}^n} f(x), \tag{5.150}$$

where f has a Lipschitz continuous Hessian with constant L_f, with respect to the standard Euclidean norm.

In the lectures, we proved the following progress for one step $x_k \mapsto x_{k+1}$ of the cubically regularized Newton's method:

$$f(x_k) - f(x_{k+1}) \quad \geq \quad \frac{1}{12 L_f^{1/2}} \|\nabla f(x_{k+1})\|^{3/2}. \tag{5.151}$$

- Prove that progress (5.151) on convex functions leads to the following rate in terms of the functional residual, for every $k \geq 1$:

$$f(x_k) - f^\star = O\Big(\frac{1}{k^2}\Big).$$

Consider the following prox function, $d(x) := \frac{1}{3}\|x\|^3 = \frac{1}{3}\langle x, x\rangle^{3/2}$.

- Show that d has a Lipschitz continuous Hessian and compute the corresponding constant L_d.

We say that a differentiable function φ is *uniformly convex* of degree $p \geq 2$ with a constant $\sigma > 0$, if the following inequality holds globally, for all $x, y \in \mathbb{R}^n$:

$$\varphi(y) \quad \geq \quad \varphi(x) + \langle \nabla\varphi(x), y - x\rangle + \tfrac{\sigma}{p}\|y - x\|^p. \tag{5.152}$$

- For uniformly convex functions (5.152), show the following bound on the functional residual:

$$\varphi(x) - \varphi^\star \quad \leq \quad C\|\nabla\varphi(x)\|^\alpha,$$

  and compute explicit formulas for parameters C and α using the parameters of uniform convexity p and σ.

  Hint: minimize the left- and the right-hand sides of (5.152) with respect to y independently.

It is known that $d(x) := \frac{1}{3}\|x\|^3$ is uniformly convex of degree $p = 3$ with constant $\sigma = \frac{1}{2}$ and you can use this fact without the proof. Consider the regularized objective

$$F(x) \quad := \quad f(x) + d(x), \tag{5.153}$$

where f is our original objective from (5.150) (convex; with a Lipschitz continuous Hessian).

- Show that F will be uniformly convex and have the Lipschitz Hessian.
- Consider applying the Cubic Newton method to minimizing F. Show that the method will have the *linear rate of convergence*, and express the complexity of finding a point $\bar{x}$ s.t. $F(\bar{x}) - F^\star \leq \varepsilon$.
- For any choice of $a_{k+1} > 0$, show what will be the complexity of minimizing $h_k(x)$ in Algorithm 5.2 by the Cubic Newton method, assuming, as previously, that f is convex with a Lipschitz continuous Hessian and $d(x) := \frac{1}{3}\|x\|^3$.
- Show how to choose a_{k+1} such that the complexity of minimizing $h_k(x)$ by the Cubic Newton is $\tilde{O}(1)$ – an absolute constant, up to logarithmic terms that depend on the target accuracy.
- Show that the corresponding growth of A_k, when substituted into (5.143), will lead to an accelerated convergence rate for the original problem (5.150):

$$f(x_k) - f^\star \quad = \quad O\Big(\tfrac{1}{k^3}\Big).$$

Literature

The first interior-point method with polynomial time complexity based on potential reduction was developed in [22], and the first polynomial-time path-following methods were developed in [39] and [14]. They were extended to general non-linear convex optimization problems using the machinery of self-concordant barriers in [34]. We recommend [31], [40], and [28] for an in-depth study of interior-point theory.

The global convergence rates for the cubic regularization of Newton's method were developed in [35], and the first appearance of the cubic regularization dates back to [17]. Adaptive and inexact cubically regularized second-order methods were developed in [6, 7]. Universal Newton methods adjusting to the Hölder constant of the Hessian were developed in [16].

Quasi-self-concordant functions were introduced in [1] and subsequently studied in [45, 21, 8]. The gradient regularization of Newton's method was considered in [38, 46, 26, 11, 9]. In these notes, we followed the presentation from [8].

Bibliography

[1] Francis Bach. Self-concordant analysis for logistic regression. 2010.

[2] Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

[3] Jonathan Borwein and Adrian Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer, 2006.

[4] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and trends in Machine Learning*, 8(3-4):231–357, 2015.

[5] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184(1):71–120, 2020.

[6] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Adaptive cubic regularisation methods for unconstrained optimization. part i: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011.

[7] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Adaptive cubic regularisation methods for unconstrained optimization. part ii: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011.

[8] Nikita Doikov. Minimizing quasi-self-concordant functions by gradient regularization of Newton method. *Mathematical Programming*, pages 1–39, 2025.

[9] Nikita Doikov, Konstantin Mishchenko, and Yurii Nesterov. Super-universal regularized Newton method. *SIAM Journal on Optimization*, 34(1):27–56, 2024.

[10] Nikita Doikov and Yurii Nesterov. Contracting proximal methods for smooth convex optimization. *SIAM Journal on Optimization*, 30(4):3146–3169, 2020.

[11] Nikita Doikov and Yurii Nesterov. Gradient regularization of Newton method with Bregman distances. *Mathematical programming*, 204(1):1–25, 2024.

[12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[13] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

[14] Clovis C Gonzaga. Path-following methods for linear programming. *SIAM review*, 34(2):167–224, 1992.

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

[16] Geovani Nunes Grapiglia and Yu Nesterov. Regularized Newton methods for minimizing functions with Hölder continuous hessians. *SIAM Journal on Optimization*, 27(1):478–506, 2017.

[17] Andreas Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical report, Technical report NA/12, 1981.

[18] Cristóbal Guzmán and Arkadi Nemirovski. On lower complexity bounds for large-scale smooth convex optimization. *Journal of Complexity*, 31(1):1–14, 2015.

[19] Elad Hazan and Sham Kakade. Revisiting the polyak step size. *arXiv preprint arXiv:1905.00313*, 2019.

[20] Lars Hörmander. *Notions of convexity*. Springer, 1994.

[21] Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients. *arXiv preprint arXiv:1806.00413*, 2018.

[22] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[24] Guanghui Lan. *First-order and stochastic optimization methods for machine learning*, volume 1. Springer, 2020.

[25] Naoki Marumo and Akiko Takeda. Universal heavy-ball method for nonconvex optimization under Hölder continuous Hessians. *Mathematical Programming*, 212(1):147–175, 2025.

[26] Konstantin Mishchenko. Regularized Newton method with global $\mathcal{O}(1/k^2)$ convergence. *SIAM Journal on Optimization*, 33(3):1440–1462, 2023.

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

[28] Arkadi Nemirovski. Interior point polynomial time methods in convex programming. *Lecture notes*, 42(16):3215–3224, 2004.

[29] Arkadi Nemirovski and David Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

[30] Yurii Nesterov. A method for solving the convex programming problem with convergence rate o (1/k2). In *Dokl akad nauk Sssr*, volume 269, page 543, 1983.

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

[32] Yurii Nesterov. Primal subgradient methods with predefined step sizes. *Journal of Optimization Theory and Applications*, 203(3):2083–2115, 2024.

[33] Yurii Nesterov. Set-limited functions and polynomial-time interior-point methods. *Journal of Optimization Theory and Applications*, 202(1):11–26, 2024.

[34] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[35] Yurii Nesterov and Boris T Polyak. Cubic regularization of Newton method and its global performance. *Mathematical programming*, 108(1):177–205, 2006.

[36] Constantin Niculescu and Lars-Erik Persson. *Convex functions and their applications*, volume 23. Springer, 2006.

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

[38] Roman A Polyak. Regularized Newton method for unconstrained convex optimization. *Mathematical programming*, 120:125–145, 2009.

[39] James Renegar. A polynomial-time algorithm, based on newton's method, for linear programming. *Mathematical programming*, 40(1):59–93, 1988.

[40] James Renegar. *A mathematical view of interior-point methods in convex optimization*. SIAM, 2001.

[41] Ralph Tyrell Rockafellar. Convex analysis. 2015.

[42] Anton Rodomanov. *Quasi-Newton methods with provable efficiency guarantees.* PhD thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2022.

[43] Anton Rodomanov, Xiaowen Jiang, and Sebastian Stich. Universality of adagrad stepsizes for stochastic optimization: Inexact oracle, acceleration and variance reduction. *Advances in Neural Information Processing Systems*, 37:26770–26813, 2024.

[44] Naum Zuselevich Shor. *Minimization methods for non-differentiable functions*. Springer Science & Business Media, 2012.

[45] Tianxiao Sun and Quoc Tran-Dinh. Generalized self-concordant functions: a recipe for Newton-type methods. *Mathematical Programming*, 178(1-2):145–213, 2019.

[46] Kenji Ueda and Nobuo Yamashita. A regularized Newton method without line search for unconstrained optimization. *Technical Report*, 2009.

[47] Moslem Zamani and François Glineur. Exact convergence rate of the last iterate in subgradient methods. *SIAM Journal on Optimization*, 35(3):2182–2201, 2025.