Lecture Notes on Nonlinear Optimization

Shixuan Zhang

ISEN 623, Spring 2024

Contents

1 Introduction

1.1 Smooth Functions and Optimization

In this course, we study problems of the following form:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) = 0, \qquad i = 1, \dots, m', \\ & g_i(x) \le 0, \qquad i = m'+1, \dots, m. \end{aligned} \tag{1.1}$$

Here, $x \in \mathbb{R}^n$ is a real vector, and $f, g_1, \dots, g_m$ are deterministically given "sufficiently smooth" functions. To be more precise, recall that a vector-valued function (or map) $h : \mathbb{R}^n \to \mathbb{R}^d$ is *continuous* at $x \in \mathbb{R}^n$ if for every $\epsilon > 0$, there exists $\delta > 0$ such that

$$\forall\, y \in \mathbb{R}^n, \|y - x\| < \delta \implies \|h(y) - h(x)\| < \epsilon,$$

where $\|\cdot\|$ is any norm on $\mathbb{R}^n$ (or $\mathbb{R}^d$). Using the limit notation, the above condition can be written more compactly as $\lim_{y \to x} f(y) = f(x)$, or simply $f(y) \to f(x)$ as $y \to x$. The function h is further *differentiable* at a point $x \in \mathbb{R}^n$ if there exists a linear function $h'(x) : \mathbb{R}^n \to \mathbb{R}^d$ such that

$$\lim_{y \to 0} \frac{\|h(x+y) - h(x) - h'(x)y\|}{\|y\|} = 0.$$

The linear function $h'(x)$ depends on the point x, so h' can be viewed as a map from $\mathbb{R}^n$ to all linear maps $\mathbb{R}^n \to \mathbb{R}^d$, called the *differential* of h. When the differential function h' itself is differentiable, we can again take its differential h'', called the *second-order* differential, and the same can be repeated for $h^{(k)}$, *k-th* differential whenever it exists. We assume that our functions $f, g_1, \dots, g_m$ are all *k-th continuously differentiable* on $\mathbb{R}^n$ for some $k \ge 1$, meaning that all the k-th differentials exist and are continuous on $\mathbb{R}^n$, which we denote as $f, g_1, \dots, g_m \in C^k(\mathbb{R}^n)$. If f is a univariate function, we simply use f' and f'' to denote its first- and second-order differential. We then use $\nabla f, \nabla g_1, \dots, \nabla g_m : \mathbb{R}^n \to \mathbb{R}^n$ to denote their first-order differential functions, which are often called *gradient vectors*, and $\nabla^2 f, \nabla^2 g_1, \dots, \nabla^2 g_m : \mathbb{R}^n \to \mathbb{R}^{n(n+1)/2}$ to denote their second-order differential functions, or *Hessian matrices* when $k \ge 2$. Occasionally, we view $g_I : \mathbb{R}^n \to \mathbb{R}^{|I|}$ as a differentiable map for some $I \subseteq \{1, \dots, m\}$, the differential of which can be written as $[\nabla g_i{}^{\mathsf{T}}]_{i \in I}^{\mathsf{T}}$ and is called a *Jacobian matrix*.

When $m = 0$, the problem (1.1) is known as an *unconstrained optimization*, and otherwise a *constrained optimization*. A natural question is to ask whether the minimum in (1.1) exists. The answer is no in general, as illustrated by the following examples.

Example 1.1. • *Consider $f(x) = x$ for $x \in \mathbb{R}$, and $m = 0$. Then for any $c \in \mathbb{R}$, we have*

$f(c-1) = c-1 < c$, *so there is no minimum.*

- *Consider* $f(x) = \exp(-x)$ *for* $x \in \mathbb{R}$, *and* $m = 0$. *Clearly* $f(x) > 0$, *but for any* $c > 0$, *we have* $f(1 + \log c) < c$, *so there is no minimum.*

Strictly speaking, we should use infimum (which is the greatest lower bound on (1.1)) in the place of minimum as it may not be attained. To ease our mind, we want to make a simplifying assumption that will appear repeatedly in the course. Recall that a set $X \subseteq \mathbb{R}^n$ is *open* if for any point $x \in X$, there exists a neighborhood of x, e.g., $U := \{y \in \mathbb{R}^n : \|y - x\| < r\}$ for some $r > 0$, such that $U \subset X$; a set is *closed* if its complement is open; and a set $X \subseteq \mathbb{R}^n$ is *bounded* if there exists $r \geq 0$ such that for all $y \in X$, $\|y\| \leq r$. It is straightforward to check a set $X \subseteq \mathbb{R}^n$ is closed if and only if for any sequence $\{x^i\}_{i=1}^\infty \subseteq X$ such that $\lim_{i\to\infty} x^i = x \in \mathbb{R}^n$, then $x \in X$.

Let $X \subseteq \mathbb{R}^n$ denote a closed *feasible set*, e.g., $\mathbb{R}^n$ in the unconstrained case or $\{x \in \mathbb{R}^n : g_i(x) = 0,\ i = 1, \ldots, m', g_j(x) \leq 0,\ j = m'+1, \ldots, m\}$ in the constrained case, and $X(a) := X \cap \{x \in \mathbb{R}^n : f(x) \leq a\}$ for any $a \in \mathbb{R}$, which is called a *(sub)level set* (with level a). One can easily check by the definition that for a continuous map, the preimage of an open (resp. a closed) set is open (resp. closed). Consequently our feasible set X, together with all the level sets $X(a)$, $a \in \mathbb{R}$, is automatically closed.

Assumption 1.2. *There exists* $a \in \mathbb{R}$ *such that* $X(a)$ *is nonempty and bounded.*

Proposition 1.3. *Under Assumption 1.2, the minimum in* (1.1) *exists, i.e., there exists* $x^* \in X$ *such that* $f(x^*) \leq f(x)$ *for any* $x \in X$.

We will need the following topological fact on $\mathbb{R}^n$ to prove this claim: a closed, bounded subset is (sequentially) *compact* (the proof of which can be found in standard textbooks, e.g., [Rud76, Theorem 2.41]).

Lemma 1.4. *Suppose* $Y \subset \mathbb{R}^n$ *is closed and bounded. Then for any sequence* $\{x^i\}_{i=1}^\infty \subset Y$, *there exists a subsequence* $\{x^{i_j}\}_{j=1}^\infty$ *and* $y \in Y$ *such that* $\lim_{j\to\infty} x^{i_j} = y$.

Proof for Proposition 1.3. Take $a \in \mathbb{R}$ such that $X(a)$ is closed and bounded. We first show that f has a lower bound on $X(a)$. Assume for contradiction that for each $i \in \mathbb{N}$, there exists $x^i \in X$ with $f(x^i) < -i$. By Lemma 1.4, there exists a subsequence $x^{i_j} \to y \in X(a)$ as $j \to \infty$. The continuity of f asserts that $f(y) = \lim_{j\to\infty} f(x^{i_j})$, but the limit does not exist by construction, hence a contradiction.

Now we take a sequence $\{y^i\}_{i=1}^\infty \subset X(a)$ such that $\lim_{i\to\infty} f(y^i) = \inf_{x\in X(a)} f(x) > -\infty$. Apply Lemma 1.4 again, there exists a subsequence $y^{i_j} \to x^* \in X(a)$ as $j \to \infty$. Thus by continuity $f(x^*) = \inf_{x\in X(a)} f(x) = \inf_{x\in X} f(x)$, where the second inequality is due to $f(x) > a$ for any $x \in X \setminus X(a)$. □

While this course is named very generally "nonlinear optimization," it is limited to a specific class of problems as detailed below.

(i) We only consider "continuous-type" variables, i.e., $x \in \mathbb{R}^n$. When some of the variables are restricted to discrete sets, e.g., $\{0,1\}$ or $\mathbb{Z}$, the problem is often known as *discrete* or *integer optimization*, and could also involve nonlinear functions. While these problems are very interesting to study, they will not appear in this course as the methodologies are typically different from the continuous case.

(ii) We assume that all of the functions $f, g_1, \ldots, g_m$ are deterministically given as part of the problem description. This is to say that we should be able to evaluate these functions, together with their differentials (gradient, and Hessian when $f, g_1, \ldots, g_m \in C^2(\mathbb{R}^n)$) at arbitrary points without any error. Besides, there are many practically important situations where the functions are given through *samples* or *simulations*, e.g., $f(x) = \mathbb{E}_\xi F(x, \xi)$ is an expectation where we only have access to F and samples of ξ. We will not discuss these formulations in this course.

1.2 Convex Sets and Functions

Arguably, the most important concept in optimization is *convexity*. We begin with the definition of a *convex set*.

Definition 1.5. *A subset $C \subseteq \mathbb{R}^n$ is* convex *if for any pair of points $x, y \in C$ and any $0 \le t \le 1$, $tx + (1-t)y \in C$.*

Intuitively, this means the *line segment* connecting any two points in the set stays in the set. Note that both the empty set and the whole space $\mathbb{R}^n$ are convex by this convention.

Exercise 1.6. *Show that for any index set I (which is possibly infinite), and $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ for all $i \in I$, the following set is closed and convex:*

$$X := \{x \in \mathbb{R}^n : a_i^\mathsf{T} x \le b_i,\ \forall\, i \in I\}.$$

It follows from Exercise 1.6 that for any matrix $A \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^m$, the system of linear inequalities $Ax \le b$ defines a closed convex set, which is called a *polyhedron*. Geometrically, it is cut out by *halfspaces* $a_i^\mathsf{T} x \le b_i$ for each row $i = 1, \ldots, m$. On a convex set, we can define a convex function.

Definition 1.7. *Let $X \subseteq \mathbb{R}^n$ be a convex set. A function $f : X \to \mathbb{R}$ is* convex *if for any points $x, y \in X$ and $0 \le t \le 1$, the inequality holds:*

$$f(tx + (1-t)y) \le tf(x) + (1-t)f(y).$$

Clearly by definition, if $f : X \to \mathbb{R}$ is convex and $Y \subseteq X$ is a convex subset, the restriction $f|_Y$ is also convex. To see a concrete example of convex functions, recall

that any norm $\|\cdot\|$ on $\mathbb{R}^n$ is subadditive: $\|x+y\| \le \|x\| + \|y\|$ for any $x, y \in \mathbb{R}^n$, and absolute homogeneous: $\|tx\| = |t|\|x\|$ for any $t \in \mathbb{R}$ and $x \in \mathbb{R}^n$, which implies the convexity: $\|tx + (1-t)y\| \le \|tx\| + \|(1-t)y\| = t\|x\| + (1-t)\|y\|$ for any $x, y \in \mathbb{R}^n$ and $0 \le t \le 1$. In particular, the standard Euclidean norm $\|(x_1, \dots, x_n)\|_2 := (x_1^2 + \cdots + x_n^2)^{1/2} = \sqrt{x^\mathsf{T} x}$ for $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ is a convex function.

Exercise 1.8. *Let $X \subseteq \mathbb{R}^n$ be a convex set and $f : X \to \mathbb{R}$ a convex function. Then the* epigraph $\operatorname{epi} f := \{(x,t) \in X \times \mathbb{R} : t \ge f(x)\}$ *and the level sets $X(a)$ are again convex for any $a \in \mathbb{R}$.*

Example 1.9. *The closed Euclidean ball $B_r := \{x \in \mathbb{R}^n : \|x\|_2 \le r\}$ is convex by Exercise 1.8, for any radius $r \ge 0$. Note that for any $x \notin B_r$, we can set $a := x/\|x\|_2$ such that $a^\mathsf{T} x = x^\mathsf{T} x/\|x\|_2 = \|x\|_2 > r$. Moreover, if $\|x\|_2 \le r$, then for any $\|a\|_2 = 1$, $a^\mathsf{T} x \le \|a\|_2\|x\|_2 \le r$. Thus we have shown that $B_r = \{x \in \mathbb{R}^n : a^\mathsf{T} x \le r,\ \forall a \in \mathbb{R}^n, \|a\|_2 = 1\}$, that is a form appeared in Exercise 1.6.*

We show that a closed convex set always allows *separation* by a hyperplane from any exterior point, as in the previous example.

Theorem 1.10. *Let $C \subseteq \mathbb{R}^n$ be a closed convex set and $x \notin C$. There exists $d \in \mathbb{R}^n$ such that $d^\mathsf{T} x < \inf_{y \in C} d^\mathsf{T} y$.*

Proof. The claim is trivial if $C = \varnothing$ because any vector d satisfies the condition. Thus we suppose $x' \in C$, with $r := \|x - x'\|_2$ and consider the convex function $f(y) := \|y - x\|_2^2$ on $y \in C(r^2) := C \cap \{y \in C : f(y) \le r^2\}$. Clearly $C(r^2)$ is closed, bounded, and nonempty, so by Proposition 1.3 there is $y^* \in C(r^2)$ such that $f(y^*) \le f(y) \le r^2$ for any $y \in C(r^2)$. Then by assumption $c := f(y^*) > 0$ because otherwise $x = y^* \in C$ by the definition of norms. Now let $d := y^* - x$ and note that

$$f(ty + (1-t)y^*) = \|y^* - x + t(y - y^*)\|_2^2 \ge \|y^* - x\|_2^2 = f(y^*),$$

for any $y \in C$ and $0 \le t \le 1$, which expands into

$$2t(y^* - x)^\mathsf{T}(y - y^*) + t^2\|y - y^*\|_2^2 \ge 0.$$

By taking $t \to 0$, we see that $d^\mathsf{T}(y - y^*) \ge 0$, which in turn gives

$$d^\mathsf{T} y \ge d^\mathsf{T} y^* = d^\mathsf{T} x + d^\mathsf{T}(y^* - x) = d^\mathsf{T} x + c > d^\mathsf{T} x. \qquad \square$$

A point $x \in \mathbb{R}^n$ is called a *boundary* point of $X \subseteq \mathbb{R}^n$ if for any $r > 0$, there exists $y \in X$ and $y' \notin X$ such that both $\|x - y\| < r$ and $\|x - y'\| < r$. A point $x \in X$ that is not a boundary point is then called an *interior* point, or we say x is in the interior $\operatorname{int} X$ of X. A set $X \subseteq \mathbb{R}^n$ together with all of its boundary point is called its *closure*, denoted

as cl X, which is the smallest closed subset of $\mathbb{R}^n$ containing X. The boundary points of X are exactly cl $X \setminus$ int X. For boundary points on a convex set, instead of finding a separating hyperplane, we can find a *supporting* hyperplane, as follows.

Theorem 1.11. *Let $C \subseteq \mathbb{R}^n$ be a convex set and $x \in \text{cl } X \setminus \text{int } X$. There exists $d \in \mathbb{R}^n$ such that $d^\mathsf{T} x = \inf_{y \in C} d^\mathsf{T} y$.*

Proof. Take a sequence $\{x^i\}_{i=1}^\infty \subset \mathbb{R}^n \setminus \text{cl } X$ with $\lim_{i\to\infty} x^i = x$. For each i, by Theorem 1.10, there exists $d^i \in \mathbb{R}^n$ such that $(d^i)^\mathsf{T} x^k < \inf_{y\in C}(d^i)^\mathsf{T} y$. By replacing d^i with $d^i/\|d^i\|$, we may assume that $\|d^i\| = 1$ for all i. By Lemma 1.4, there exists a subsequence i_j such that $\lim_{j\to\infty} d^{i_j} = d$ for some $d \in \mathbb{R}^n$. We thus have for any $y \in C$, $d^\mathsf{T} x = \lim_{j\to\infty}(d^{i_j})^\mathsf{T} x^{i_j} \le \lim_{j\to\infty}(d^{i_j})^\mathsf{T} y = d^\mathsf{T} y$. □

When looking at the epigraph of a convex function, the supporting hyperplane is closely related to the differential at a point, as used in the following proof.

Theorem 1.12. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function that is differentiable at $x \in \mathbb{R}^n$. Then for any $y \in \mathbb{R}^n$,*

$$f(y) - f(x) \ge \nabla f(x)^\mathsf{T}(y - x).$$

Proof. Consider the epigraph epi $f \subseteq \mathbb{R}^{n+1}$, which is convex by Exercise 1.8. The point $(x, f(x))$ is a boundary point of epi f due to the continuity of f at x. By Theorem 1.11, there exists $g \in \mathbb{R}^n$ and $c \in \mathbb{R}$ such that $g^\mathsf{T} x + cf(x) \le g^\mathsf{T} y + cz$ for any $y \in \mathbb{R}^n$, $z \ge f(y)$. By letting $z \to +\infty$, we know that $c > 0$ so we can set $c = 1$ by replacing g with g/c. Take $z = f(y)$, we get $f(y) - f(x) \ge -g^\mathsf{T}(y - x)$. Assume $-g \ne \nabla f(x)$. There exists $u \in \mathbb{R}^n$ with $\|u\| = 1$ such that $\nabla f(x)^\mathsf{T} u < 0 < -g^\mathsf{T} u$. This is a contradiction because by the definition of $\nabla f(x)$, there exists $\delta > 0$ such that for any $0 < t < \delta$,

$$f(x + tu) < f(x) + t\nabla f(x)^\mathsf{T} u + \frac{t}{2}|\nabla f(x)^\mathsf{T} u| < f(x) \le f(x + tu) + tg^\mathsf{T} u.$$

Therefore, $-g = \nabla f(x)$, which completes the proof. □

Convexity simplifies analysis and boosts algorithmic performance in many ways. Although this course targets at general (smooth) nonlinear optimization problems, we will highlight some nice results in the convex cases.

2 Essentials of Unconstrained Optimization

2.1 Solutions and Optimality Conditions

For an unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} \quad f(x), \tag{2.1}$$

it is important to define the "solutions" of interest.

Definition 2.1. *For (2.1), a point $x^* \in \mathbb{R}^n$ is called*

- a local minimum *if there exists $\epsilon > 0$ such that $f(x^*) \le f(x)$ for any $x \ne x^*$ with $\|x - x^*\| < \epsilon$; and further* a strict local minimum *if $f(x^*) < f(x)$ holds in this definition;*
- a global minimum *if $f(x^*) \le f(x)$ for all $x \ne x^*$; and further* a strict global minimum *if $f(x^*) < f(x)$ holds in this definition.*

Example 2.2.

- *Let $f(x) = x^2$ for $x \ge 0$ and $f(x) = 0$ otherwise. Then $x^* = 0$ is a local minimum and a global minimum, but not a strict one.*
- *Let*
$$f(x) = \begin{cases} (x+1)^2 - 1, & x \le 0, \\ -(x-1)^2 + 1, & 0 < x < 2, \\ \frac{1}{2}(x-4)^2 - 2, & x \ge 2. \end{cases}$$
Then $x^* = -1$ is a strict local minimum, but not a global minimum because $f(4) = -2 < -1 = f(x^*)$.

The situation is much simplified when the function is linear or quadratic.

Exercise 2.3.

- *If $f(x) = g^\mathsf{T} x + c$ is a linear function for some $g \in \mathbb{R}^n$ and $c \in \mathbb{R}$, then $x^* \in \mathbb{R}^n$ is a local minimum of f if and only if $g = 0$. In this case, any $x \in \mathbb{R}^n$ is indeed a global minimum of f.*
- *If $f(x) = \frac{1}{2} x^\mathsf{T} H x + g^\mathsf{T} x + c$ is a quadratic function for some real symmetric matrix $H \in \mathcal{S}^n$, vector $g \in \mathbb{R}^n$, and number $c \in \mathbb{R}$, then $x^* \in \mathbb{R}^n$ is a local minimum of f if and only if $Hx^* = -g$ and $H \succeq 0$ (i.e., H is positive semidefinite). In this case, x^* is also a global minimum; it is a strict (local or global) minimum if and only if $H \succ 0$ (i.e., H is positive definite).*

While the mathematical definition suggests that one should seek a global minimum for (2.1), it is often satisfactory to find a local minimum from the practical point of view. Besides, as we will see later in this course, many famously challenging problems can be reduced to finding a global minimum of certain nonlinear function f, which leaves little hope that we can always do this efficiently. Nevertheless, the concept of local minimality still does not directly lead to computational tractability, because it involves finding a open neighborhood and checking the function values at all other points in it. Instead, we use the following necessary or sufficient conditions as surrogates of local minimality.

Theorem 2.4 (First-order Necessary Condition)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable in an open neighborhood of $x^* \in \mathbb{R}^n$. If x^* is a local minimum of f, then $\nabla f(x^*) = 0$.*

Theorem 2.5 (Second-order Optimality Conditions). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable in an open neighborhood of $x^* \in \mathbb{R}^n$.*

- *(Necessary) If x^* is a local minimum of f, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq 0$.*
- *(Sufficient) If $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$, then x^* is a (strict) local minimum of f.*

An intuitive way to think about the necessary conditions is that we are locally approximating the function f using a linear or a quadratic function (see Lemma 2.7 below), so by Exercise 2.3 we must have its linear coefficient being 0 and quadratic coefficient matrix being positive semidefinite. The points satisfying the necessary conditions in Theorems 2.4 and 2.5 are often called *first-* and *second-order stationary points*, respectively. However, a second-order stationary point may not be a local minimum, as shown in the following example.

Example 2.6. *Consider $f(x) = x^3$ for $x \in \mathbb{R}$. Then $f'(0) = f''(0) = 0$, which makes $x^* = 0$ a second-order stationary point of f. It is not a local minimum because $f(x) < 0 = f(0)$ for any $x < 0$.*

To prove these conditions, we need a form of Taylor's formula from calculus, the proof of which can be found for example in [Zor15, Section 8.4.4].

Lemma 2.7. *If $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable in a neighborhood of $x \in \mathbb{R}^n$, then for any $y \in \mathbb{R}^n$, there exists $0 < t < 1$ such that*

$$f(x+y) = f(x) + \nabla f(x+ty)^\mathsf{T} y.$$

Moreover, if f is twice continuously differentiable in a neighborhood of x, then there exists $0 < s < 1$ such that

$$f(x+y) = f(x) + \nabla f(x)^\mathsf{T} y + \frac{1}{2} y^\mathsf{T} \nabla^2 f(x+sy) y.$$

Proof for Theorem 2.4. Assume for contradiction that $\nabla f(x^*) \neq 0$ and take $y = -\nabla f(x^*)$. Since $\nabla f(x^*)^\mathsf{T} y < 0$, by the continuity of ∇f in the neighborhood of x^*, there exists $s > 0$ such that $\nabla f(x^* + ry)^\mathsf{T} y < 0$ for all $0 < r < s$. For any such r, by Lemma 2.7, there exists $0 < t < 1$ such that $f(x^* + ry) = f(x^*) + \nabla f(x^* + try)^\mathsf{T}(ry) < f(x^*)$, which contradicts with that $f(x^*)$ is a local minimum. □

Proof for Theorem 2.5. For the necessary condition, the equality $\nabla f(x^*) = 0$ is shown in Theorem 2.4 and the positive semidefiniteness is proved similarly given Lemma 2.7. We show the sufficient condition as follows. By the continuity of $\nabla^2 f$ in a neighborhood of x^*, there exists $\epsilon > 0$ such that $\nabla^2 f(x) \succ 0$ for any $x \in \mathbb{R}^n$, $\|x - x^*\| < \epsilon$. Take any $y \in \mathbb{R}^n$ with $\|y - x^*\| < \epsilon$. By Lemma 2.7, there exists $0 < s < 1$ such that $f(x^* + y) = f(x^*) + \frac{1}{2} y^\mathsf{T} \nabla^2 f(x^* + sy) y$. This implies that $f(x^* + y) > f(x^*)$ for any $\|y\| < \epsilon$, so x^* is a strict local minimum. □

Theoretically, one can define higher-order optimality conditions and stationary points, but computing the higher-order differentials and checking their "positive definiteness" are often challenging and thus of less interest. One may also notice a "missing" first-order sufficient condition for local optimality, which is due to the lack of strictness of optimal solutions for a linear function (see Exercise 2.3). However, for convex function, the first-order necessary condition is indeed sufficient for (global) minimality.

Theorem 2.8. *If $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function, then any of its local minimum point is also a global minimum point. Moreover, if f is differentiable at $x^* \in \mathbb{R}^n$, then x^* is a global minimum of f if and only if $\nabla f(x^*) = 0$.*

Proof. For the first assertion, assume for contradiction that there exists a local minimum $x^* \in \mathbb{R}^n$ and another point $y \in \mathbb{R}^n$ such that $f(y) < f(x)$. Then by convexity of f, we have $f(x^* + t(y - x)) \leq tf(y) + (1 - t)f(x^*) < f(x^*)$ for any $0 < t < 1$. Take $t \to 0$, this contradicts the local minimality of x^*.
For the second assertion, note that by Theorem 1.12, we have $f(y) - f(x^*) \geq \nabla f(x^*)(y - x^*)$ for any $y \in \mathbb{R}^n$. Thus $\nabla f(x^*) = 0$ implies the global minimality of x^*. □

2.2 Iterative Algorithms and Newton's Method

In this course, we focus on *iterative algorithms* for nonlinear optimization. In plain words, such methods produce a sequence $\{x^i\}_{i=1}^{\infty}$ by iteratively updating our incumbent solution x^i to x^{i+1}. In particular,

- the method is called *zeroth-order* if it only uses information $f(x^j)$ for $j \leq i$ to produce x^{i+1};
- it is called *first-order* if it uses additionally the information $\nabla f(x^j)$ for $j \leq i$;
- and *second-order* if $\nabla^2 f(x^j)$ for $j \leq i$ are also used.

Theoretically one can imagine more higher-order methods, but they are rarely used in practice due to the cost of computing these differentials. For comparison, we outline an *enumerative algorithm* that is impractical for large dimensions n.

Example 2.9. *Suppose we only want to search for a solution for $\min_x f(x)$ within the box $\{x \in \mathbb{R}^n : -b \leq x_i \leq b, i = 1, \ldots, n\}$ for some $b > 0$. Then we can check all the function values $f(-b + 2a_1 b/N, \ldots, -b + 2a_n b/N)$ for all $a \in \{0, 1, \ldots, N\}^n$ and find a "minimum" among them. The benefit of such grid method is that we can approximately find a global minimum, but with a huge drawback: roughly speaking the approximation error could be proportional to b/N. Therefore, to reduce the approximation error by half, we may need to double N and consequently enumerate 2^n as many points (e.g., $2^{100} > 10^{69}$ for $n = 100$)!*

The iterative algorithms, limited to local minima or stationary points as they may be, often perform much more efficiently in terms of the *accuracy* of the solutions, as discussed below.

Suppose we have a sequence of points $\{x^i\}_{i=1}^{\infty}$ with $x^i \to x^*$ as $k \to \infty$. We say that the points have a *(geometric) convergence of order k* if

$$\limsup_{i\to\infty} \frac{\|x^{i+1} - x^*\|}{\|x^i - x^*\|^k} < \infty,$$

that is, there exist $j \in \mathbb{N}$ and $c > 0$ such that for all $i > j$, $\|x^{i+1} - x^*\| < c\|x^i - x^*\|^k$. It is clear that an order-$(k+1)$ convergence is also an order-k convergence. The order-2 geometric convergence is also called *quadratic convergence*, and the order-1 geometric convergence is called *linear convergence* if the above limit superior is strictly less than 1. Between linear and quadratic convergences, the term *superlinear convergence* rate is also used, if

$$\lim_{i\to\infty} \frac{\|x^{i+1} - x^*\|}{\|x^i - x^*\|} = 0.$$

The term *sublinear convergence* is also occasionally used when the above limit is 1, but it is not very informative. Alternatively, one can say the sequence has an *arithmetic convergence of order k* when there exists $c > 0$ such that $\|x^i - x^*\| \le c \cdot i^{-k}$.

Example 2.10.

- *The sequence $x^i = 1$ does not converge to 0, but satisfies*

$$\limsup_{i\to\infty} \frac{|x^{i+1}|}{|x^i|} = 1 < \infty.$$

  Thus it is important to check convergence before discussing the rate of convergence!
- *The sequences of real numbers $x^i = 1/i$ and $y^i = 1/i^2$ both converge to 0 sublinearly because $\lim_{i\to\infty} i/(i+1) = \lim_{i\to\infty} i^2/(i+1)^2 = 1$. However, the sequence $\{x^i\}$ converges arithmetically of order 1, while $\{y^i\}$ converges arithmetically of order 2.*

A celebrated algorithmic idea in optimization and numerical methods is called *Newton's method*, which can be traced back to the Babylonian or Heron's method for finding square roots. For our problem (2.1), to find a stationary point, by Lemma 2.7, when f is sufficiently smooth, we can simply approximate it with a quadratic function near a given point x^i

$$f(x) \approx f(x^i) + \nabla f(x^i)^\mathsf{T}(x - x^i) + \frac{1}{2}(x - x^i)^\mathsf{T}\nabla^2 f(x^i)(x - x^i).$$

Then by Exercise 2.3, an approximate minimizer should be

$$x^{i+1} \leftarrow x^i - [\nabla^2 f(x^i)]^{-1}\nabla f(x^i), \tag{2.2}$$

assuming that $[\nabla^2 f(x^i)]^{-1}$ exists, which is the update formula for Newton's method. As the second-order differential is evaluated in (2.2), it is a second-order method. We next show its convergence rate as our first example of convergence analysis.

Theorem 2.11. *Suppose $x^* \in \mathbb{R}^n$ is a local minimum of a function $f \in C^3(U)$ for some neighborhood U of x^*, and $\nabla^2 f(x^*) \succ 0$. Then the Newton's method (2.2) converges quadratically to x^* provided that the starting point x^0 is sufficiently close to x^*.*

We use the notation $|||M||| := \sup_{\|x\|=1} \|Mx\|$ to denote the matrix (operator) norm for any matrix $M \in \mathbb{R}^{m\times n}$. One can check that it is indeed a norm and continuous in M.

Proof. Because $f \in C^3(U)$, we can apply Lemma 2.7 to $\nabla f(x)$ at x^* such that there exists $r, \alpha > 0$ and $\|\nabla f(x^*) - \nabla f(x) - \nabla^2 f(x^*)(x^* - x)\| \le \alpha \|x - x^*\|^2$ for any $x \in \mathbb{R}^n$, $\|x - x^*\| < r$. As $\nabla^2 f(x^*)$ is positive definite, reducing r if needed, by continuity of $\nabla^2 f(x)^{-1}$, there exists $\beta > 0$ such that $|||\nabla^2 f(x)^{-1}||| < \beta$ for any $\|x - x^*\| < r$. Now suppose $\|x^0 - x^*\| < \min\{r, \frac{1}{2\alpha\beta}\}$. Then inductively for $i = 0, 1, 2, \ldots$,

$$\begin{aligned}
\|x^{i+1} - x^*\| &= \|x^i - x^* - \nabla^2 f(x^i)^{-1} \nabla f(x^i)\| \\
&= \|\nabla^2 f(x^i)^{-1}(\nabla^2 f(x^i)(x^i - x^*) - \nabla f(x^i))\| \\
&\le |||\nabla^2 f(x^i)^{-1}|||\|\nabla^2 f(x^i)(x^i - x^*) - \nabla f(x^i)\| \\
&\le \beta\alpha \|x^i - x^*\|^2 < \frac{1}{2}\|x^i - x^*\| < \min\{r, \frac{1}{2\alpha\beta}\}.
\end{aligned}$$

Moreover, this shows that $\|x^i - x^*\| \le (\frac{1}{2})^{i-1}\|x^0 - x^*\|$, so $x^i \to x^*$ as $i \to \infty$ and the convergence is quadratic by the inequality $\|x^{i+1} - x^*\| \le \beta\alpha\|x^i - x^*\|^2$. □

The assumption that x^0 is sufficiently close to x^* is necessary, even when the function f is convex, as can be seen from the following example.

Exercise 2.12. *Suppose $f \in C^2(\mathbb{R}^n)$. Then f is convex if and only if $\nabla^2 f(x) \succeq 0$ for any $x \in \mathbb{R}^n$. In this case, we say that f is α-strongly convex if the minimum eigenvalue of $\nabla^2 f(x)$ is at least $\alpha > 0$ for any $x \in \mathbb{R}^n$.*

Example 2.13. *Consider a univariate polynomial function $f(x) = 4x^6 - 15x^4 + 42x^2$, $x \in \mathbb{R}$, with its first-order derivative $\nabla f(x) = 24x^5 - 60x^3 + 84x$ and second-order derivative $\nabla^2 f(x) = 120x^4 - 180x^2 + 84 \ge \frac{33}{2} > 0$. By Exercise 2.12, we know that f is $\frac{33}{2}$-strongly convex, and has an obvious minimum $x^* = 0$, which is strict and thus unique by Theorem 2.5. Nevertheless, if we start with $x^0 = 1$, we see that $\nabla f(x^0) = 48$ and $\nabla^2 f(x^0) = 24$, so $x^1 = -1$ by (2.2). Then $\nabla f(x^1) = -48$ and $\nabla^2 f(x^1) = 24$ so we would get $x^2 = 1$ again. This leads to a* cycle *between $x^{2i} = 1$ and $x^{2i+1} = -1$ for $i = 0, 1, 2, \ldots$, so the Newton's method (2.2) does not converge in this case.*

3 Basic Descent Methods

3.1 Global Convergence

A *descent* method generates a monotone sequence $\{x^i\}_{i=0}^{\infty}$ with $f(x^0) \geq f(x^1) \geq \cdots \geq f(x^i) \geq \cdots$. An obvious benefit of descent methods is that we may restrict our attention to the initial level set $X(f(x^0))$ for convergence analysis, which is particularly useful for unconstrained optimization problems as $\mathbb{R}^n$ itself is not bounded. Another important fact is the following general framework for global convergence analysis.

Theorem 3.1. *Let $X \subseteq \mathbb{R}^n$ be an open set, $\{x^i\}_{i=0}^{\infty} \subseteq X$, and $f \in C^1(X)$ such that $X(f(x^0))$ is closed and bounded.*

- *If for any $\epsilon > 0$, there exists $\delta > 0$ such that whenever $\|\nabla f(x^i)\| \geq \epsilon$, $f(x^i) - f(x^{i+1}) \geq \delta$, then any limit point x^* of $\{x^i\}$ satisfies $\|\nabla f(x^*)\| = 0$.*
- *Suppose $f \in C^2(X)$. If for any $\epsilon > 0$, there exists $\delta > 0$ such that whenever $\|\nabla f(x^i)\| \geq \epsilon$ or $\lambda_{\min}(\nabla^2 f(x^i)) \leq -\epsilon$, $f(x^i) - f(x^{i+1}) \geq \delta$, then any limit point x^* of $\{x^i\}$ satisfies $\|\nabla f(x^*)\| = 0$ and $\nabla^2 f(x^*) \succeq 0$.*

Proof. For notational simplicity, we only prove the first assertion, while the second follows from an identical argument. The existence of limit points follows from the assumption on $X(f(x^0))$ and Lemma 1.4. Assume for contradiction that there is a limit point $x^* \in X$ of $\{x^i\}$, i.e., there is a subsequence $\{x^{i_j}\}$ with $x^{i_j} \to x^*$ as $j \to \infty$, such that $\nabla f(x^*) > 0$. By the continuity of ∇f, there exists $\epsilon > 0$ and $r > 0$ such that $\|\nabla f(x)\| \geq \epsilon$ for any $x \in X$, $\|x - x^*\| \leq r$. This means that there is an integer $N > 0$ such that $\|\nabla f(x^{i_j})\| \geq \epsilon$ for all $j > N$. By assumption, $f(x^{i_j}) - f(x^{i_j+1}) \geq \delta$ for all $j > N$, which implies that $f(x^i) \to -\infty$ as $i \to \infty$. This is a contradiction as f attains its minimum on X by Proposition 1.3. □

Note that the values ϵ, δ in Theorem 3.1 can be arbitrary, but they have to be *independent* of the point x^i. The surprisingly simple yet powerful idea is sometimes referred to as Lyapunov-type argument, and can be extended to a more quantitative bound: for example, if there exists $c > 0$ such that $\delta \geq c\epsilon$, then to get an iterate x^i with $\|\nabla f(x^i)\| \leq \epsilon$, we need at most $\lfloor \frac{f(x^0) - f^*}{c\epsilon} \rfloor$ iterations, where $f^* := \min_{x \in X} f(x)$. This is an order-1 arithmetic convergence for any subsequence $\{x^{i_j}\}$ that converges.

A special case is that f has a unique stationary point x^* in $X(f(x^0))$, which would ensure $x^i \to x^*$ as $i \to \infty$ by Theorem 3.1. Generally speaking, we do not have control over which limit point it converges to, unless we impose further restrictions on our descent method.

3.2 Trust Region Methods

A reasonable explanation on the divergence of Newton's method is that it relies on local quadratic approximation but sometime goes too far from its region of validness. Thus a natural idea is to only look for a new point x^{i+1} within a given radius of the current point x^i. To be precise, we consider the so-called *trust-region* subproblem

$$\begin{aligned} \min_{y\in\mathbb{R}^n} \quad & \frac{1}{2}y^\mathsf{T} H^i y + (g^i)^\mathsf{T} y \\ \text{s. t.} \quad & \|y\|_2^2 \le \delta_i^2, \end{aligned} \tag{3.1}$$

where $\delta_i > 0$ is a preset radius, the symmetric matrix H^i and the vector g^i are local approximation for our function f, which are usually set to be $\nabla^2 f(x^i)$ and $\nabla f(x^i)$, respectively. After getting an optimal update y^i from eq. (3.1), we set $x^{i+1} \leftarrow x^i + y^i$. For simplicity, we always use the standard Euclidean norm $\|\cdot\| = \|\cdot\|_2$ in this section. We want to show that with the additional trust region constraint helps us satisfy the descent condition in Theorem 3.1.

Note that generally H^i may not be positive semidefinite, especially when the iterate x^i is not close to a second-order stationary point x^* with $\nabla f^2(x^*) \succ 0$. Thus the problem (3.1) may have local minima that are not global minima. A very useful observation below shows that even without convexity, we can solve the trust region subproblem for a (high accuracy) global solution. The proof uses the following simple fact, which we will discuss in a more general form later in the lectures for constrained optimization.

Exercise 3.2. *Let $f \in C^1(\mathbb{R}^n)$. If x is a minimum of f on the sphere $\{y \in \mathbb{R}^n : y^\mathsf{T} y = 1\}$, then there exists $\lambda \in \mathbb{R}$ such that the $\nabla f(x) + \lambda x = 0$.*

Theorem 3.3. *For the trust region subproblem (3.1), y^i is an optimal solution if and only if there exists $\mu \ge 0$ such that*

$$(H^i + \mu I)y^i = -g^i, \quad H^i + \mu I \succeq 0, \quad \mu(\|y^i\|_2^2 - \delta_i^2) = 0.$$

Proof. Assume first the existence of $\mu \ge 0$ satisfying the three conditions. By Exercise 2.3, we know that y^i is a global optimal solution to the problem

$$\min_{y\in\mathbb{R}^n} \quad \frac{1}{2}y^\mathsf{T}(H^i + \mu I)y + (g^i)^\mathsf{T} y.$$

Thus for any $\|y\| \le \delta_i$, we have

$$\frac{1}{2}y^\mathsf{T} H^i y + (g^i)^\mathsf{T} y \ge \frac{1}{2}(y^i)^\mathsf{T} H^i y^i + (g^i)^\mathsf{T} y^i + \frac{\mu}{2}((y^i)^\mathsf{T} y^i - y^\mathsf{T} y) \ge \frac{1}{2}(y^i)^\mathsf{T} H^i y^i + (g^i)^\mathsf{T} y^i,$$

where the last inequality is due to that $\mu((y^i)^\mathsf{T} y^i - \delta_i^2) = 0$. This shows that y^i is indeed

a global minimum to the trust region problem (3.1).

Conversely, assume that y^i is a global minimum of (3.1). If $\|y^i\| < \delta_i$, then it is actually an unconstrained minimization, so by Exercise 2.3, it holds that $H^i y^i = -g^i$ and $H^i \succeq 0$. In this case we can simply choose $\mu = 0$. Thus we assume that $\|y^i\| = \delta_i$. By Exercise 3.2, there exists $\mu \in \mathbb{R}$ such that $H^i y^i + g^i + \mu y^i = 0$. Then for any $y^\mathsf{T} y = \delta_i^2$,

$$\begin{aligned}(y - y^i)^\mathsf{T}(H^i + \mu I)(y - y^i) &= y^\mathsf{T}(H^i + \mu I)y - 2y^\mathsf{T}(H^i + \mu I)y^i + (y^i)^\mathsf{T}(H^i + \mu I)y^i \\ &= y^\mathsf{T}(H^i + \mu I)y + 2(g^i)^\mathsf{T} y + (y^i)^\mathsf{T}(H^i + \mu I)y^i \\ &\geq 2(y^i)^\mathsf{T}(H^i + \mu I)y^i + 2(g^i)^\mathsf{T} y^i = 0.\end{aligned}$$

This shows that $H^i + \mu I \succeq 0$ and any such μ satisfies the desired conditions.

It remains to show that we can always choose a nonnegative $\mu \geq 0$. Assume for contradiction that only $\mu < 0$ is possible. In this case, any $y \in \mathbb{R}^n$ with $\|y\|_2 \geq \delta_i$ satisfies

$$\frac{1}{2} y^\mathsf{T} H^i y + (g^i)^\mathsf{T} y \geq \frac{1}{2}(y^i)^\mathsf{T} H^i y^i + (g^i)^\mathsf{T} y^i + \mu((y^i)^\mathsf{T} y^i - y^\mathsf{T} y) \geq \frac{1}{2}(y^i)^\mathsf{T} H^i y^i + (g^i)^\mathsf{T} y^i,$$

because $\mu((y^i)^\mathsf{T} y^i - y^\mathsf{T} y) = \mu(\delta_i^2 - y^\mathsf{T} y) \geq 0$. From the assumption that y^i is a global minimum of (3.1), we know that y^i is a global minimum of $\frac{1}{2} y^\mathsf{T} H^i y + (g^i)^\mathsf{T} y$ on $\mathbb{R}^n$. Then by Exercise 2.3, we can simply set $\mu = 0$, and this contradiction completes the proof. □

Theorem 3.3 leads to the following procedure to find an optimal solution to the subproblem (3.1). We first find an eigenvalue decomposition $H^i = Q\Lambda Q^\mathsf{T}$ for some orthogonal matrix $Q = (q^1, \ldots, q^n)$ and a diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Then we sequentially check the following cases to find y^i.

(i) If $\lambda_1 \geq 0$, set $\mu = 0$ and let $y^i := -Q\Lambda^\dagger Q^\mathsf{T} g^i$, where $\Lambda^\dagger = \operatorname{diag}(0, \ldots, 0, \lambda_k^{-1}, \ldots, \lambda_n^{-1})$ with $k := \min\{1 \leq i \leq n : \lambda_i > 0\}$.

(ii) If $\lambda_1 < 0$ and $(q^1)^\mathsf{T} g^i = \cdots = (q^{k-1})^\mathsf{T} g^i = 0$, set $\mu = -\lambda_1 > 0$ and $y^i := -Q(\Lambda + \mu I)^\dagger Q^\mathsf{T} g^i + \nu q^1$, where $k := \min\{1 \leq i \leq n : \lambda_i + \mu > 0\}$ and

$$\nu := \sqrt{\delta_i^2 - \sum_{j=k}^{n} \frac{((q^j)^\mathsf{T} g^i)^2}{(\lambda_j + \mu)^2}}.$$

(iii) Otherwise, $\lambda_1 < 0$ and $(q^j)^\mathsf{T} g^i \neq 0$ for some $j < k$. Consider

$$y(\mu) := -Q(\Lambda + \mu I)^{-1} Q^\mathsf{T} g^i = -\sum_{j=1}^{n} \frac{(q^j)^\mathsf{T} g^i}{\lambda_j + \mu} q^j \text{ for } \mu > -\lambda_1 > 0.$$

The root of the function

$$r(\mu) := \|y(\mu)\|^2 - \delta_i^2 = \sum_{j=1}^{n} \frac{((q^i)^\mathsf{T} g^i)^2}{(\lambda_j + \mu)^2} - \delta_i^2$$

gives the desired μ^* and thus $y^i := y(\mu^*)$. Notice that $\mu^* \leq -\lambda_1 + \|g^i\|/\delta_i$ and $r(\mu)$ is a monotone decreasing (rational) function, which allows efficient high-accuracy solutions (through bisections and Newton's method as discussed in section 3.3).

To ensure that eq. (3.1) gives a descent step $x^{i+1} \leftarrow x^i + y^i$, one may need to decrease (or increase) δ_i based on the value of $f(x^i + y^i)$. To be more precise, let $m_i(y) := \frac{1}{2} y^\mathsf{T} H^i y + (g^i)^\mathsf{T} y$ and set

$$\rho_i := \frac{f(x^i) - f(x^i + y^i)}{m_i(0) - m_i(y^i)} \tag{3.2}$$

to be the ratio of *actual reduction* and *predicted reduction*. With preselected constants $0 < a, b < 1$ and $\delta > 0$, a simple version of the trust region update is described in Algorithm 3.1, in which we scale down the radius δ_i if ρ_i is too small to ensure a sufficient descent. It is easy to see that the new iterate x^{i+1} produced by Algorithm 3.1

Algorithm 3.1 A Trust Region Descent Method

Require: $0 < a, b < 1$, $\delta > 0$, and $x^i \in \mathbb{R}^n$
1: set $\delta_i \leftarrow \delta$
2: solve eq. (3.1) for y^i and calculate ρ_i.
3: **if** $\rho_i < a$ **then**
4: set $\delta_i \leftarrow b\delta_i$ and go back to step 2
5: **end if**
6: **return** $x^{i+1} \leftarrow x^i + y^i$

satisfies $f(x^{i+1}) \leq f(x^i) + am_i(y^i)$. We next show that such descent is sufficient for global convergence (in the sense of Theorem 3.1) for a nice class of functions.

Definition 3.4. *For any subset $X \subseteq \mathbb{R}^n$, a map $h : X \to \mathbb{R}^d$ is* β-Lipschitz continuous *if for any $x, y \in \mathbb{R}^n$, $\|h(x) - h(y)\| \leq \beta\|x - y\|$. A function $f \in C^k(X)$ is called k-th-order β-Lipschitz continuous if its k-th order differential is β-Lipschitz continuous (in the corresponding vector or matrix norms).*

Theorem 3.5. *Suppose $f \in C^2(\mathbb{R}^n)$ is second-order β-Lipschitz continuous on $X(f(x^0))$ for a given $x^0 \in \mathbb{R}^n$. Then any limit point x^* of the sequence $\{x^i\}_{i=0}^\infty$ generated by Algorithm 3.1 with $g^i = \nabla f(x^i)$ and $H^i = \nabla^2 f(x^i)$ satisfies the second-order necessary condition.*

Proof. From the second-order β-Lipschitz continuity on $X_0 := X(f(x^0))$, using Lemma 2.7, we know for any $x, x + y \in X_0$ there exists some $0 < s < 1$ and

$$\|f(x+y) - f(x) - \nabla f(x)^\mathsf{T} y - \frac{1}{2} y^\mathsf{T} \nabla^2 f(x) y\| = \frac{1}{2}\|y^\mathsf{T}(\nabla^2 f(x+sy) - \nabla^2 f(x))y\| \leq \frac{\beta}{2}\|y\|^3.$$

Moreover, there exists $\eta > 0$ such that $|||H^i||| \le \eta$ for all i. Now suppose at a point x^i that does not satisfy the second-order necessary condition, i.e., either $\|g^i\| > 0$ or $\lambda_{\min}(H^i) < 0$.

- If $\|g^i\| > 0$, then by setting $y(t) := tg^i$, it is straightforward to verify that $m_i(y^i) \le \min_{0 \le t \le \delta_i/\|g^i\|} m_i(y(t)) \le -\frac{1}{2}\|g^i\| \min\{\delta_i, \frac{\|g^i\|}{|||H^i|||}\}$. Thus whenever

$$\delta_i \le \min\left\{\left(\frac{(1-a)\|g^i\|}{\beta}\right)^{1/2}, \left(\frac{(1-a)\|g^i\|^2}{\beta\eta}\right)^{1/3}\right\} =: \bar{\delta}(\|g^i\|),$$

  we would have $\rho_i \ge a$ and y^i will be used to update x^{i+1}. Consequently, $\delta_i \ge \min\{\delta, b\bar{\delta}(\|g^i\|)\}$ and thus $f(x^i) - f(x^{i+1}) \ge \frac{1-a}{2}\|g^i\| \min\{\delta, b\bar{\delta}(\|g^i\|), \frac{\|g^i\|}{\eta}\}$ (which only depends on $\|g^i\|$ and other constants).
- If $\lambda_{\min}(H^i) < 0$, then $m_i(y^i) = -\frac{1}{2}(y^i)^\mathsf{T}(H^i + \mu I)y^i - \frac{\mu}{2}\|y^i\|^2 \le \frac{\lambda_{\min}(H^i)\delta_i^2}{2}$ so $f(x^i) - f(x^{i+1}) \ge -\frac{\lambda_{\min}(H^i)\delta_i^2}{2} - \frac{\beta\delta_i^3}{2}$. Thus as long as $\delta_i \le (1-a)\frac{\mu}{\beta}$, we would have $\rho_i \ge a$ and y^i will be used. Consequently, $\delta_i \ge \min\{\delta, b(a-1)\lambda_{\min}(H^i)/\beta\} =: \tilde{\delta}(\lambda_{\min}(H^i))$ and $f(x^i) - f(x^{i+1}) \ge -\frac{a}{2}\lambda_{\min}(H^i)\tilde{\delta}(\lambda_{\min}(H^i))^2$.

Therefore, by Theorem 3.1, we know that any limit point x^* of $\{x^i\}_{i=0}^\infty$ satisfies the second-order necessary condition. □

In Theorem 2.11, we saw that a basic Newton's method has quadratic convergence when the sequence $\{x^i\}$ converges to x^* satisfying the second-order sufficient condition. A natural question is whether this is still true for the trust region method. An observation is that in this case, $H^i \succ 0$ for sufficiently large i and thus $|||(H^i)^{-1}||| \le \gamma$ for some $\gamma > 0$, which implies that $\|y^i\| = \|(H^i)^{-1}g^i\| \le \gamma\|g^i\|$. Meanwhile, the accepted radius $\bar{\delta}(\|g^i\|)$ has an order strictly less than 1 (in terms of $\|g^i\|$), so y^i will be in the interior of the trust region whenever $\|g^i\|$ is sufficiently small. This leads to a proof of the following claim.

Exercise 3.6. *Let $f \in C^2(\mathbb{R}^n)$ be a second-order β-Lipschitz continuous function on $X(f(x^0))$ for some given $x^0 \in \mathbb{R}^n$. Suppose $\{x^i\}_{i=1}^\infty$ is a sequence generated by the trust region method (Algorithm 3.1) and $x^i \to x^*$ for some $x^* \in U$ satisfying the second-order sufficient condition for f. Then x^i converges to x^* quadratically.*

We remark that Algorithm 3.1 with the update procedure following Theorem 3.3 is an idealized version and may be less practical, as the numerical root-finding of $r(\mu)$ and the eigenvalue decomposition $H^i = Q\Lambda Q^\mathsf{T}$ can be both challenging. Instead, one may want to find an approximate minimum with "good" descent of the function value, such as searching along the direction $y(t) := tg^i$, as in the proof of Theorem 3.5. This leads to the discussion of *Cauchy points* with more specialized procedures and convergence analyses. For more details, please refer to [NW06, Chapter 4] or [CGT00].

3.3 Line Search Methods

As we have seen in the previous discussion, one can search for an optimal solution as the new iterate along a chosen direction, instead of optimizing over a region. This is exactly the main idea of another popular class of algorithms, namely *line search* methods. A typical line search method consists of the following two steps in its i-th update.

(i) Select a direction $d^i \in \mathbb{R}^n$.

(ii) Determine the step length $\tau_i \in \arg\min_{\tau \in \mathbb{R}_{\geq 0}} f(x^i + \tau d^i)$.

Then one can set $x^{i+1} \leftarrow x^i + \tau_i d^i$ and continue to the next iteration. The problem of finding the step length is a 1-dimensional optimization problem, it is sometimes much easier than the general optimization in $\mathbb{R}^n$. For simplicity, we denote $\phi_i(\tau) := f(x^i + \tau d^i)$ in each iteration i, and note that $\phi_i'(\tau) = \nabla f(x^i + \tau d^i)^\mathsf{T} d^i$ can be computed by the gradient of f. We illustrate this in two special cases and suppress the subscript i in ϕ_i if no confusion is caused.

When f is a polynomial function with a degree $\deg f \in \mathbb{Z}_{\geq 2}$. Then the line search problem $\min_{\tau \in \mathbb{R}} \phi(\tau)$ can be solved via finding (at most $\deg f + 1$) critical points of ϕ and comparing the function value at these points, as suggested by the first-order necessary condition (Theorem 2.4). Finding the critical points is equivalent finding the roots of the derivative of ϕ, which is again a polynomial function, so this procedure can be done either analytically or numerically to a high precision.

Another important case is that f is a convex function and we below describe a method called *bisection* for the line search problem. Assume that we can estimate an interval $[a_0, b_0] \subset \mathbb{R}$ where the function $\phi(\tau)$ has its differential f' change sign on it, i.e., $\phi'(a) < 0 < \phi'(b)$. It is easy to analyze the performance of the bisection method, as an optimal solution $\tau_* \in \arg\min_{\tau \in [a_0, b_0]} \phi(\tau)$ always lies within the current interval $[a_j, b_j]$ for each iteration j.

Algorithm 3.2 Bisection

Require: $\epsilon > 0$, $[a_0, b_0] \subset \mathbb{R}$ such that $\phi'(a_0) < 0 < \phi'(b_0)$

let $j \leftarrow 0$
repeat
  set $c_j \leftarrow (a_j + b_j)/2$
  if $f'(c_j) > 0$ **then**
    set $b_{j+1} \leftarrow c_j$ and $a_{j+1} \leftarrow a_j$
  else if $f'(c_j) < 0$ **then**
    set $a_{j+1} \leftarrow c_j$ and $b_{j+1} \leftarrow b_j$
  else
    break
  end if
  update $j \leftarrow j + 1$
until $b_j - a_j < \epsilon$
return $(a_j + b_j)/2$

Exercise 3.7. *If $\phi \in C^1([a_0, b_0])$ is convex and $\phi'(a_0) < 0 < \phi'(b_0)$, then Algorithm 3.2 gives $|c_j - \tau_*| \le (\frac{1}{2})^j (b_0 - a_0)$ for some $\tau_* \in \arg\min_{\tau \in [a_0, b_0]} \phi(\tau)$ after j-th iteration. This implies its linear convergence.*

Note that Algorithm 3.2 is a first-order method as it uses the derivative ϕ' information. When ϕ'' is also available, bisection can be combined with the basic Newton's method for higher accuracy in practice, i.e., use bisection until we get sufficiently close to an τ_* such that the basic Newton's method would converge. If it is computationally prohibitive to evaluate ϕ', a similar method called *golden section search* can be applied with only zeroth order information (i.e., the function values of ϕ), with a linear convergence constant $\frac{\sqrt{5}-1}{2} \approx 0.618$. More discussion can be found in [LY21, Section 8.1].

Next we try to qualify the Newton's method for global convergence. While the step length selection makes it easier to ensure descent, in general we cannot guarantee that the system of equations for finding the Newton's direction

$$H^i d = -g^i \tag{3.3}$$

has a solution $d = d^i \in \mathbb{R}^n$, where H^i is typically chosen to be $\nabla^2 f(x^i)$ (or its approximation) and g^i to be $\nabla f(x^i)$, e.g., any function f that is locally linear at x^i would have $\nabla^2 f(x^i) = 0$. When it does, it is not necessarily true that d^i leads to a descent in the function value, i.e.,

$$f(x^i) > \min_{\tau \ge 0} f(x^i + \tau d^i). \tag{3.4}$$

Examples can be constructed in one dimension, where f is locally concave but increasing at x^i. To avoid these issues, we consider a special class of functions $f \in C^2(\mathbb{R}^n)$ that are α-strongly convex for some $\alpha > 0$, i.e., $\nabla^2 f(x) \succeq \alpha I$ for all $x \in \mathbb{R}^n$. These functions clearly have nonsingular Hessian matrices $\nabla^2 f(x)$ and give a descent direction in the Newton's step because $\nabla f(x)^\top [\nabla^2 f(x)]^{-1} \nabla f(x) \ge 0$ for any x. We use the nice properties to show a global convergence result for these functions.

Theorem 3.8. *Let $f \in C^2(\mathbb{R}^n)$ be an α-strongly convex function and first-order β-Lipschitz continuous on $X(f(x^0))$ for some $x^0 \in \mathbb{R}^n$. Then the line search Newton's method generates a sequence $\{x^i\}_{i=0}^\infty$ (via the Newton's step (3.3) with $H^i = \nabla^2 f(x^i)$ and $g^i = \nabla f(x^i)$) that converges quadratically to the unique minimum $x^* = \arg\min_{x \in \mathbb{R}^n} f(x)$.*

Proof. By assumption, we have $\alpha I \preceq \nabla^2 f(x^i) \preceq \beta I$ for any iteration i. Thus us-

ing Lemma 2.7, we have

$$\begin{aligned} f(x^{i+1}) &\le f(x^i) + \nabla f(x^i)^\top (x^{i+1} - x^i) + \frac{\beta}{2}\|x^{i+1} - x^i\|^2 \\ &= f(x^i) - \tau_i \nabla f(x^i)^\top [\nabla^2 f(x^i)]^{-1} \nabla f(x^i) + \frac{\beta \tau_i^2}{2} \|[\nabla^2 f(x^i)]^{-1} \nabla f(x^i)\| \\ &\le f(x^i) - \tau_i \nabla f(x^i)^\top [\nabla^2 f(x^i)]^{-1} \nabla f(x^i) + \frac{\beta \tau_i^2}{2\alpha} \nabla f(x^i)^\top [\nabla^2 f(x^i)]^{-1} \nabla f(x^i) \\ &= f(x^i) - \Big(\tau_i - \frac{\beta \tau_i^2}{2\alpha}\Big) \nabla f(x^i)^\top [\nabla^2 f(x^i)]^{-1} \nabla f(x^i). \end{aligned}$$

Here, the first inequality is due to $\nabla^2 f(x^i) \preceq \beta I$; the next equality is due to $x^{i+1} = x^i + \tau_i [\nabla^2 f(x^i)]^{-1} \nabla f(x^i)$; and the second inequality is due to $[\nabla^2 f(x^i)]^{-1} \preceq \frac{1}{\alpha} I$. The step length τ_i should give a descent greater than or equal to the one given by $\frac{\alpha}{\beta}$, which implies that

$$f(x^{i+1}) \le f(x^i) - \frac{\alpha}{2\beta} \nabla f(x^i)^\top [\nabla^2 f(x^i)]^{-1} \nabla f(x^i) \le f(x^i) - \frac{\alpha}{2\beta^2} \|\nabla f(x^i)\|^2,$$

because $[\nabla^2 f(x^i)]^{-1} \succeq \frac{1}{\beta} I$. Note that this descent only depends on $\|\nabla f(x^i)\|$, so by Theorem 3.1 we know that $x^i \to x^*$ as $i \to \infty$.

Now let $x^{i,\diamond} := x^i - [\nabla^2 f(x^i)]^{-1} \nabla f(x^i)$ denote the basic Newton's update. By definition of line search methods, we have $f(x^{i+1}) \le f(x^{i,\diamond})$, which implies that

$$\|x^{i+1} - x^*\|^2 \le \frac{2}{\alpha}(f(x^{i+1}) - f(x^*)) \le \frac{2}{\alpha}(f(x^{i,\diamond}) - f(x^*)) \le \frac{\beta}{\alpha}\|x^{i,\diamond} - x^*\|^2.$$

When i is sufficiently large, by Theorem 2.11, we have $\|x^{i,\diamond} - x^*\| \le C\|x^i - x^*\|^2$, for some constant $C > 0$, from which we conclude the quadratic convergence

$$\|x^{i+1} - x^*\| \le C\sqrt{\frac{\beta}{\alpha}}\|x^i - x^*\|^2. \qquad \square$$

While Theorem 3.8 illustrates the power of line search methods in the strongly convex setting, it is not easy to generalize it to cases without convexity. In particular, we have the following two challenges:

(i) when the step length function ϕ is not convex, there could be multiple local minima and finding an exact minimum along the line can still be challenging;

(ii) when $\nabla^2 f(x)$ has full rank but is indefinite, the Newton's direction $-[\nabla^2 f(x)]^{-1} \nabla f(x)$ may not give a descent for any step length $\tau \ge 0$.

For Challenge (i), a common practice is to use an *inexact line search* that finds a "reasonably good" step length $\tau_i > 0$. Despite the possible suboptimality, convergence can still be established provided that the following condition is satisfied.

Definition 3.9. *Fix constants $0 < a < 1$ and $b > 1$. We say that the step length $\tau_i > 0$ satisfies*

the (a,b)-Armijo condition *if* $\phi_i(\tau_i) \le \phi_i(0) + a\tau_i\phi_i'(0)$ *and* $\phi_i(b\tau_i) \ge \phi_i(0) + ab\tau_i\phi_i'(0)$.

Note that the directional derivative of the line search problem $\phi'(0) = \nabla f(x^i)^\top d^i < 0$ whenever d^i is a descent direction. To satisfy the Armijo condition, we can use the following *backtracking* method: given x^i and d^i, consider an initial step length $\tau_i = \tau > 0$ and repeat $\tau_i \leftarrow \tau_i/b$ until $\phi_i(\tau_i) \le \phi_i(0) + a\tau_i\phi'(0)$.

Exercise 3.10. *Suppose* $f \in C^1(\mathbb{R}^n)$. *Given any* $0 < a < 1$ *and* $b > 1$, *if* $\phi_i'(0) < 0$, *then the backtracking method terminates in finitely many steps with* $\tau_i > 0$ *satisfying the Armijo condition.*

Other conditions for inexact line search termination include *Wolfe (curvature) condition* or *Goldstein conditions*, which aim to ensure that the step length τ_i leads to larger derivative $\phi_i'(\tau_i)$ (or smaller $|\phi_i'(\tau_i)|$), or to ensure that τ_i is sufficiently large. However, the conditions cannot be satisfied by using backtracking alone and in practice often requires more sophisticated methods. Please refer to [NW06, Chapter 3] for more discussion.

For Challenge (ii), one can use modified Hessian matrix H^i instead of the indefinite $\nabla^2 f(x^i)$. Perhaps the simplest strategy is to set $H^i \leftarrow \nabla^2 f(x^i) + \mu I$ for a sufficiently large $\mu > 0$ such that $H^i \succ 0$. Such modification would guarantee that $d^i := -[H^i]^{-1}\nabla f(x^i)$ is a descent direction. However, as one would expect, the convergence rate would also be compromised unless $\mu \to 0$ sufficiently fast. For an illustrative purpose, let us consider an extreme case where μ stays large, and in fact, much larger than needed. Note that the line search direction

$$d^i(\mu) := -\frac{[H^i]^{-1}\nabla f(x^i)}{\|[H^i]^{-1}\nabla f(x^i)\|} = -\frac{(I + \frac{1}{\mu}\nabla^2 f(x^i))^{-1}\nabla f(x^i)}{\|(I + \frac{1}{\mu}\nabla^2 f(x^i))^{-1}\nabla f(x^i)\|} \to -\nabla f(x^i) \tag{3.5}$$

as $\mu \to \infty$. In other words, using an (overly) large μ pulls the search direction towards the opposite direction of the gradient, which is clearly a descent direction. This limiting behavior gives the famous *gradient descent method*, which simply sets $d^i = -\nabla f(x^i)$ in each iteration i. Its global convergence is established below.

Theorem 3.11. *Suppose* $f \in C^1(\mathbb{R}^n)$ *that is first-order* β*-Lipschitz continuous on* $X(f(x^0))$ *for some* $x^0 \in \mathbb{R}^n$. *Consider any sequence* $\{x^i\}_{i=0}^\infty$ *generated by the gradient descent method with line search steps satisfying the* (a,b)*-Armijo condition for some* $0 < a < 1$ *and* $b > 1$. *Then any limit point of* $\{x^i\}$ *must satisfy the first-order necessary condition. In particular,*

$$\min_{0 \le j \le i} \|\nabla f(x^j)\|^2 \le \frac{b\beta(f(x^0) - f^*)}{a(1-a)i}$$

for any $i \in \mathbb{N}$, *where* $f^* := \min_{x \in \mathbb{R}^n} f(x)$.

Proof. Following Theorem 3.1, we want to show that for any $\epsilon > 0$, there exists $\delta > 0$ such that whenever $\|\nabla f(x)\| \geq \epsilon$, we have $f(x) - f(x - \tau\nabla f(x)) \geq \delta$ for any τ satisfying the Armijo condition. By definition, $f(x) - f(x - \tau\nabla f(x)) \geq a\tau\|\nabla f(x)\|^2$, so it suffices to show that such τ is bounded from below (independent of the choice of x). Using Taylor's approximation (Lemma 2.7) and the first-order β-Lipschitz continuity, we have

$$f(x - b\tau\nabla f(x)) \leq f(x) - b\tau\|\nabla f(x)\|^2 + \beta b^2\tau^2\|\nabla f(x)\|^2.$$

Use the definition of the Armijo condition again, we also have

$$f(x - b\tau\nabla f(x)) \geq f(x) - ab\tau\|\nabla f(x)\|^2.$$

These two inequalities imply that any step length satisfying the Armijo condition must satisfy

$$(1-a)b\tau \leq \beta b^2\tau^2 \implies \tau \geq \frac{(1-a)}{\beta b}.$$

Consequently, we have

$$f(x) - f(x - \tau\nabla f(x)) \geq a\tau\|\nabla f(x)\|^2 \geq \frac{a(1-a)\|\nabla f(x)\|^2}{\beta b} =: \delta > 0.$$

Moreover, if $\min_{0\leq j\leq i}\|\nabla f(x^j)\|^2 > \epsilon$ then $f(x^i) \leq f(x^0) - \dfrac{a(1-a)i\epsilon}{\beta b}$. Setting $\epsilon = \frac{b\beta(f(x^0)-f^*)}{a(1-a)i}$ gives a contradiction with $f(x^i) \geq f^*$, which completes the proof. □

Theorem 3.11 shows global convergence of gradient descent methods. The rate of the convergence remains a question. As we know that Newton's method finds an minimum of a quadratic function $f(x) := \frac{1}{2}x^\mathsf{T}Hx + g^\mathsf{T}x$ in one iteration, assuming $H \succ 0$, it is thus of interest to see how the gradient descent method (with exact line search steps) performs in this case. Recall that for a positive definite matrix H, its *condition number* $\kappa(H) := \lambda_{\max}(H)/\lambda_{\min}(H)$ is the ratio of the largest and smallest eigenvalues. A useful inequality regarding the condition number, due to Kantorovich, is the following.

Theorem 3.12. *Let $H \succ 0$ and $\kappa(H)$ be its condition number. Then for any $y \in \mathbb{R}^n$,*

$$\frac{\|y\|_2^2}{(y^\mathsf{T}Hy)(y^\mathsf{T}H^{-1}y)} \geq \frac{4\kappa(H)}{(1+\kappa(H))^2}.$$

Proof. Let $H = Q\Lambda Q^\mathsf{T}$ denote an eigenvalue decomposition of H, where Q is an orthogonal matrix and $\Lambda := \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ is the diagonal matrix of positive eigenvalues $0 < \lambda_1 \leq \cdots \leq \lambda_n$. Take $u := Q^\mathsf{T}y$ and $v_i := u_i^2/\|u\|_2^2$, $i = 1, \ldots, n$, and the left-hand

side becomes

$$\frac{\|u\|_2^2}{(u^\mathsf{T}\Lambda u)(u^\mathsf{T}\Lambda^{-1}u)} = \frac{1}{(\sum_{i=1}^n \lambda_i v_i)(\sum_{i=1}^n v_i/\lambda_i)} = \frac{\rho(\sum_{i=1}^n \lambda_i v_i)}{\sum_{i=1}^n v_i \rho(\lambda_i)},$$

where $\rho(t) := 1/t$ is the reciprocal function. By the convexity of ρ on $\mathbb{R}_{>0}$ and that $\sum_{i=1}^n v_i = 1$, the minimum of the above ratio can be attained at $v_2 = \cdots = v_{n-1} = 0$, so

$$\begin{aligned}\frac{\|y\|_2^2}{(y^\mathsf{T}Hy)(y^\mathsf{T}H^{-1}y)} &\geq \min_{v_1+v_n=1} \frac{1}{(v_1\lambda_1 + v_n\lambda_n)(v_1/\lambda_1 + v_n/\lambda_n)} \\ &= \min_{0\leq v_n\leq 1} \frac{\kappa(H)}{(1-v_n+v_n\kappa(H))((1-v_n)\kappa(H)+v_n)} = \frac{\kappa(H)}{(1+\kappa(H))^2}.\end{aligned}$$

The last step is follows from the inequality of arithmetic and geometric means, where the equality holds for $v_n = \frac{1}{2}$. □

For simplicity, we will use a weighted norm $\|x\|_H := (x^\mathsf{T}Hx)^{1/2}$ to quantify the error instead of the usual Euclidean norm, but it is easy to see that they are equivalent, i.e., there exists $c > 0$ such that $\frac{1}{c}\|x\| \leq \|x\|_H \leq c\|x\|$ for any $x \in \mathbb{R}^n$. In this way we have $\|x - x^*\|_H^2 = 2f(x) + (x^*)^\mathsf{T}Hx^*$.

Theorem 3.13. *Consider a quadratic function $f(x) := \frac{1}{2}x^\mathsf{T}Hx + g^\mathsf{T}x$ for some $H \succ 0$ and $g \in \mathbb{R}^n$. Then for any $x^0 \in \mathbb{R}^n$, the sequence $\{x^i\}_{i=0}^\infty$ produced by the gradient descent method with an exact line search converges linearly to $x^* = -H^{-1}g$. More precisely, for each $i \in \mathbb{Z}_{\geq 0}$,*

$$\|x^{i+1} - x^*\|_H \leq \left(\frac{\kappa(H)-1}{\kappa(H)+1}\right)^2 \|x^i - x^*\|_H.$$

Proof. Let $g^i := \nabla f(x^i) = Hx^i + g = H(x^i - x^*)$ denote the gradient in each iteration i, and the corresponding line search function $\phi_i(\tau) = f(x^i - \tau g^i)$ with $\phi_i'(\tau) = -(g^i)^\mathsf{T}(H(x^i - \tau g^i) + g) = -\|g^i\|^2 + \tau(g^i)^\mathsf{T}Hg^i$. This shows that step length is

$$\tau_i = \frac{\|g^i\|^2}{(g^i)^\mathsf{T}Hg^i},$$

and thus by Theorem 3.12,

$$\begin{aligned}\frac{\|x^i - x^*\|_H^2 - \|x^{i+1} - x^*\|_H^2}{\|x^i - x^*\|_H^2} &= \frac{2\tau_i(g^i)^\mathsf{T}H(x^i - x^*) - \tau_i^2(g^i)^\mathsf{T}Hg^i}{(x^i - x^*)^\mathsf{T}H(x^i - x^*)} \\ &= \frac{\|g^i\|^2}{((g^i)^\mathsf{T}Hg^i)((g^i)^\mathsf{T}H^{-1}g^i)} \geq \frac{4\kappa(H)}{(1+\kappa(H))^2}.\end{aligned}$$

The proof is then completed by subtracting 1 on both sides. □

Theorem 3.13 shows the (global) linear convergence of the gradient descent method

with exact line search on strongly convex quadratic functions. This rate can also be shown (in terms of the objective function values) in the general case, assuming that the second-order sufficient condition holds for the limit point, in an asymptotic sense.

Exercise 3.14. *Given any $f \in C^1(\mathbb{R}^n)$ and $x^0 \in \mathbb{R}^n$, assume that a sequence $\{x^i\}_{i=0}^{\infty}$ generated by the gradient descent method with exact line search steps converges to a point $x^* \in \mathbb{R}^n$ such that $f \in C^2(U)$ for some neighborhood U of x^* and $\nabla^2 f(x^*) \succ 0$. Then $f(x^i)$ converges linearly to $f(x^*)$.*

While the actual rate of convergence may depend on the starting point, there are case studies that empirically corroborates the linear rate described in Theorem 3.13. This is somewhat unfavorable as the rate adversely depends on the condition number of H, even without any consideration of numerical errors. To help visualization, a typical trajectory of the gradient descent method in a plane is shown in Figure 3.1. Note that two consecutive updates are perpendicular to each other, known as the "zigzag" behavior, which reflects the obstacle of condition number for convergence. One can show that this is true in general.

Exercise 3.15. *Let $f \in C^1(\mathbb{R}^n)$ and $\{x^i\}_{i=0}^{\infty}$ be a sequence generated by the gradient descent method with exact line search steps. Then $x^{i+2} - x^{i+1}$ is orthogonal to $x^{i+1} - x^i$, for each $i \in \mathbb{N}$.*

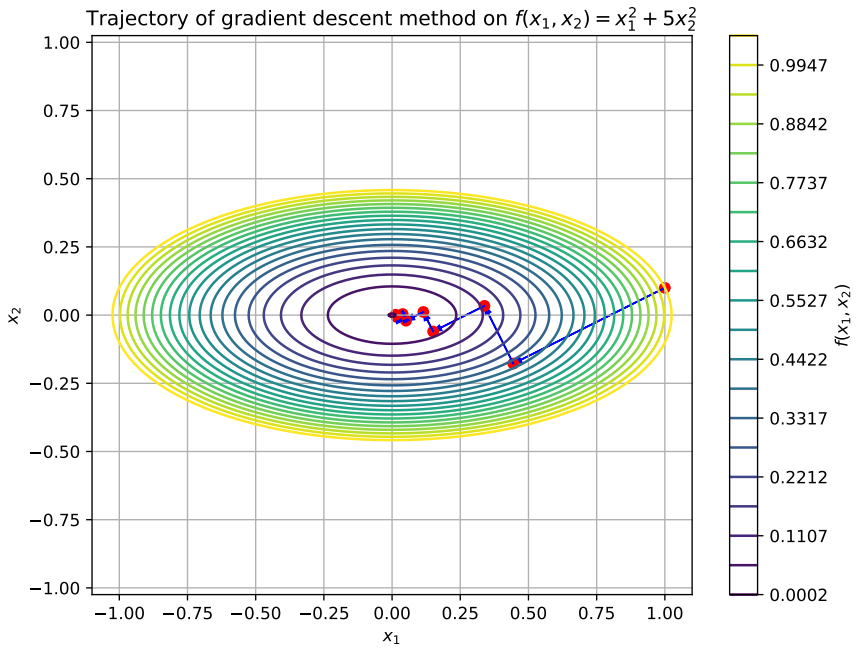

Figure 3.1: A typical trajectory of the gradient descent method

4 First-Order Descent Methods

In this section, we consider some more "advanced" first-order methods (i.e., only first-order differential information of the function is used) that imitate Newton's method in some way, for unconstrained optimization problems. This is out of practical considerations that finding the inverse of H^i or $\nabla^2 f(x^i)$ can be computationally formidable for large n, which is roughly speaking on the order of n^3 arithmetic operations. In contrast, updating points with gradients, for example, only takes n arithmetic operations.

4.1 Conjugate Gradient Methods

In Section 3.3, we have seen that the usual gradient descent method can be slowed down by a large condition number, when applied to a strongly convex quadratic function $f(x) = \frac{1}{2}x^\mathsf{T} Hx + g^\mathsf{T} x$. An alternative method, called *conjugate gradient method*, solves this issue to some extent by utilizing *conjugate directions*. We say that a set of vectors $\{d^0, \dots, d^m\}$ are H-conjugate to each other if $(d^i)^\mathsf{T} H d^j = 0$ for any $i \neq j$, $0 \leq i, j \leq m$. Conjugate directions are linearly independent, and can be used to find the minimum of f in n steps with the exact line search method.

Theorem 4.1. *Let $f(x) = \frac{1}{2}x^\mathsf{T} Hx + g^\mathsf{T} x$ and $\{d^i\}_{i=0}^{n-1}$ be a set of H-conjugate vectors. Then*

- *$\{d^i\}$ are linearly independent;*
- *for any $x^0 \in \mathbb{R}^n$, the sequence generated by $x^{i+1} = x^i + \tau_i d^i$ with $\tau_i \in \arg\min_{\tau \geq 0} f(x^i + \tau d^i)$ for each $i = 0, 1 \dots, n-1$ satisfies $x^n = x^* := \arg\min_{x \in \mathbb{R}^n} f(x)$.*

Proof. For the first assertion, suppose there exist $\sigma_0, \sigma_1, \dots, \sigma_{n-1} \in \mathbb{R}$ such that $\sum_{i=0}^{n-1} \sigma_i d^i = 0$. Then for any $j = 0, \dots, n-1$, if we multiply both sides by $(d^j)^\mathsf{T} H$, the H-conjugacy implies that $\sigma_j (d^j)^\mathsf{T} H d^j = 0$, so $\sigma_j = 0$ because $H \succ 0$.

For the second assertion, note that the step lengths satisfy the first-order necessary condition of the line search functions

$$(d^i)^\mathsf{T}(H(x^i + \tau_i d^i) + g) = 0 \implies \tau_i = -\frac{(g^i)^\mathsf{T} d^i}{(d^i)^\mathsf{T} H d^i},$$

where $g^i := Hx^i + g = \nabla f(x^i)$, for each $i = 0, \dots, n-1$. Thus $x^j = x^0 + \sum_{i=1}^{j-1} \sigma_i d^i$, which implies that $(d^j)^\mathsf{T} H(x^j - x^0) = 0$ for each $j = 1, \dots, n$. Now by the first assertion, there exist $\sigma_0, \dots, \sigma_{n-1} \in \mathbb{R}$ such that $x^* - x^0 = \sum_{i=0}^{n-1} \sigma_i d^i$. Again multiplying both sides by $(d^j)^\mathsf{T} H$, we get

$$\sigma_i = \frac{(d^i)^\mathsf{T} H(x^* - x^0)}{(d^i)^\mathsf{T} H d^i} = \frac{(d^i)^\mathsf{T} H(x^* - x^i)}{(d^i)^\mathsf{T} H d^i} = \frac{(d^i)^\mathsf{T}(-g^i)}{(d^i)^\mathsf{T} H d^i} = \tau_i.$$

This shows that $x^* - x^0 = \sum_{i=0}^{n-1} \tau_i d^i = x^n - x^0$ so $x^n = x^*$. □

To obtain the H-conjugacy directions, we can use the previous line search directions to modify the new gradient direction. To be precise, we summarize the conjugate gradient method in Algorithm 4.1.

Algorithm 4.1 Conjugate Gradient Method

Require: $x^0 \in \mathbb{R}^n$, $H \succ 0$, $g \in \mathbb{R}^n$
1: set $d^0 = -g^0 \leftarrow Hx^0 + g$
2: **for** $i = 0, \dots, n-1$ **do**
3: update $x^{i+1} \leftarrow x^i + \tau_i d^i$ with $\tau_i = -\dfrac{(g^i)^\mathsf{T} d^i}{(d^i)^\mathsf{T} H d^i}$
4: **if** $g^{i+1} = Hx^{i+1} + g = 0$ **then**
5: **return** x^{i+1}
6: **end if**
7: set $d^{i+1} \leftarrow -g^{i+1} + \sigma_i d^i$ with $\sigma_i = \dfrac{(g^{i+1})^\mathsf{T} H d^i}{(d^i)^\mathsf{T} H d^i}$
8: **end for**

We now verify that the directions $d^0, \dots, d^{n-1}$ in Algorithm 4.1 are indeed H-conjugate, so it must return the optimal solution x^* within n iterations by Theorem 4.1. The subspace $K_i(g^0; H) := \text{span}\{g^0, Hg^0, H^2 g^0, \dots, H^i g^0\}$ is often referred to as the *Krylov subspace of degree i*.

Theorem 4.2. *Suppose $g^i \neq 0$ for any $i = 0, 1, \dots, n-1$ in Algorithm 4.1. Then the generated sequences $\{d^i\}_{i=0}^{n-1}$ and $\{g^i\}_{i=0}^{n-1}$ satisfy*

- $\text{span}\{d^0, \dots, d^i\} = \text{span}\{g^0, \dots, g^i\} = K_i(g^0; H)$, *and*
- $(g^i)^\mathsf{T} d^j = (d^i)^\mathsf{T} H d^j = 0$ *for any* $j = 0, 1, \dots, i-1$, *for any* $i = 1, \dots, n-1$.

Proof. We prove both assertions by induction on i. For $i = 0$, they are trivially true. Now suppose that they are true for some i. Note that $g^{i+1} = Hx^{i+1} + g = g^i + \tau_i H d^i$. By the induction hypothesis, $(g^{i+1})^\mathsf{T} d^j = (g^i)^\mathsf{T} d_j + \tau_i (d^i)^\mathsf{T} H d^j = 0$ for any $j < i$. Thus by the definition of τ_i, $(g^{i+1})^\mathsf{T} d^i = (g^i)^\mathsf{T} d^i + \tau_i (d^i)^\mathsf{T} H d^i = 0$ shows the equality $(g^{i+1})^\mathsf{T} d_j = 0$ for all $j < i+1$.

Now the induction hypothesis also implies $g^{i+1} \in K_{i+1}(g^0; H)$ as $g^i, d^i \in K_i(g^0; H) \subseteq K_{i+1}(g^0; H)$; in fact $g^{i+1} \notin K_i(g^0; H)$ by the argument above, so we must have $K_{i+1}(g^0; H) = \text{span}\{g^0, \dots, g^{i+1}\}$. It follows that $\text{span}\{d^0, \dots, d^{i+1}\} = K_{i+1}(g^0; H)$ from the relation $d^{i+1} = -g^{i+1} + \sigma_i d^i$ and the induction hypothesis.

Finally we show the H-conjugacy of the directions d^{i+1} with $d^0, \dots, d^i$. Note that $(d^{i+1})^\mathsf{T} H d^j = -(g^{i+1})^\mathsf{T} H d^j + \sigma_i (d^i)^\mathsf{T} H d^j$ for any $j \leq i$. When $j = i$, this is clearly zero by the definition of σ_i. When $j < i$, $Hd^j \in K_i(g^0; H)$ implies that $(g^{i+1})^\mathsf{T} H d^j = 0$. The induction hypothesis says $(d^i)^\mathsf{T} H d^j = 0$, so $(d^{i+1})^\mathsf{T} H d^j = 0$, which completes the induction step. □

Theorem 4.2 shows the n-step convergence of the conjugate gradient method. It reduces the dependence on the condition number as in the case of the gradient descent

methods, but such convergence guarantee still may not be satisfactory when n is large. Practically speaking, the hope is that it returns an optimal or near-optimal solution much earlier than the n-th iteration. This can be established, by looking at the number of *distinct* eigenvalues of H. Informally, the conjugate gradient method would terminate in $k \ll n$ steps if H only has k distinct eigenvalues. When the eigenvalues of H can be clustered into $k \ll n$ groups, then the method would also return a near-optimal solution after k iterations. For more discussion, please refer to [NW06, Section 5].

Theorem 4.2 also provides an alternative way of writing the step length τ_i and direction update coefficient σ_i. As $(g^i)^\mathsf{T} d^i = (g^i)^\mathsf{T}(-g^i + \sigma_{i-1} d^{i-1}) = -\|g^i\|_2^2$, we have

$$\tau_i = \frac{\|g^i\|_2^2}{(d^i)^\mathsf{T} H d^i}.$$

Similarly, because $g^i \in K_i(g^0; H)$, $(g^{i+1})^\mathsf{T} g^i = 0$, and thus

$$(g^{i+1})^\mathsf{T} H d^i = \frac{1}{\tau_i}(g^{i+1})^\mathsf{T} H(x^{i+1} - x^i) = \frac{1}{\tau_i}(g^{i+1})^\mathsf{T}(g^{i+1} - g^i) = \frac{1}{\tau_i}\|g^{i+1}\|_2^2.$$

This implies that

$$\sigma_i = \frac{(g^{i+1})^\mathsf{T} H d^i}{(d^i)^\mathsf{T} H d^i} = \frac{\|g^{i+1}\|_2^2}{\|g^i\|_2^2}, \tag{4.1}$$

or

$$\sigma_i = \frac{(g^{i+1})^\mathsf{T}(g^{i+1} - g^i)}{\|g^i\|_2^2}. \tag{4.2}$$

The formulas eq. (4.1) and eq. (4.2) help us extend the conjugate gradient method to general nonlinear functions $f \in C^1(\mathbb{R}^n)$, known as the *Fletcher-Reeves* and the *Polak-Ribiere* methods, respectively, As usual, we use $g^i := \nabla f(x^i)$ and then use σ_i to update the line search direction. We describe a possible implementation of general conjugate direction methods in Algorithm 4.2.

The global convergence of such general conjugate gradient method is directly implied by the convergence of gradient descent method (as in each loop g^0 is the gradient direction).

Exercise 4.3. *Given any $f \in C^1(\mathbb{R}^n)$ and $x^0 \in \mathbb{R}^n$, any limit point of the sequence $\{x^j\}_{j=1}^\infty$ generated by Algorithm 4.2 satisfies the first-order necessary condition.*

The rate of convergence of Algorithm 4.2 can also be characterized as follows. Suppose the generated sequence $x^j \to x^*$ for some x^* satisfying the second-order sufficient condition. Then there exists some $C > 0$ such that $\|x^{j+1} - x^*\| \le C\|x^j - x^*\|^2$ for all $j \in \mathbb{Z}_{\ge 0}$. A proof for this convergence claim can be found in [Coh72] and a more complete survey of such methods can be found in [HZ06]. Note that this is not the same as the usual quadratic convergence because we need to execute n steps (in updating $x^{j,i}$) for each of such reduction in the distance $x^j - x^*$, which can be undesired for large n.

Algorithm 4.2 General Conjugate Gradient Method

Require: $x^0 \in \mathbb{R}^n$, $\epsilon \geq 0$
set $j \leftarrow 0$
while $\|\nabla f(x^j)\| > \epsilon$ **do**
  let $g^0 \leftarrow \nabla f(x^j)$ and $d^0 \leftarrow -g^0$
  for $i = 0, \ldots, n-1$ **do**
    if $g^i = 0$ **then**
      break
    end if
    update $x^j \leftarrow x^{j,i+1} = x^{j,i} + \tau_i d^i$ with $\tau_i \in \arg\min_{\tau \geq 0} f(x^{j,i} + \tau d^i)$
    compute $g^{i+1} \leftarrow \nabla f(x^{j,i+1})$
    set $d^{i+1} \leftarrow -g^{i+1} + \sigma_i d^i$ with σ_i calculated by eq. (4.1) or eq. (4.2)
  end for
  update $j \leftarrow j + 1$
end while

4.2 Quasi-Newton Methods

The main idea behind quasi-Newton methods is to approximate the inverse of the Hessian matrix using information from the iterations. Ideally as the sequence $\{x^i\}_{i=0}^{\infty}$ converges to a desired solution x^* (with a positive definite Hessian), the approximation also converges to the Hessian at x^*, imitating the convergence of the Newton's method. More precisely, suppose $f \in C^2(\mathbb{R}^n)$. For any two points x^i and x^{i+1}, let $g^i := \nabla f(x^i)$ and $g^{i+1} := \nabla f(x^{i+1})$, with $p^i := x^{i+1} - x^i$. Then a *secant* approximation of the Hessian matrix in iteration i is

$$\nabla^2 f(x^i) p^i \approx g^{i+1} - g^i =: q^i,$$

which is exact when f is a quadratic function (so $\nabla^2 f(x)$ does not vary with x). In particular, if we use a matrix R_i to approximate the inverse of $\nabla^2 f(x^i)$ based on the data from first i steps of the descent process, it is natural to expect the (inverse) secant conditions to hold

$$R_i q^j = p^j, \quad j = 0, 1, \ldots, i-1. \tag{4.3}$$

Below we discuss update schemes for R_i that preserves the secant conditions (4.3).

Since the Hessian matrix and its inverse are symmetric matrices, we should keep our approximation R_i symmetric as well. Perhaps the simplest way is to use *rank-one corrections*, i.e.,

$$R_{i+1} = R_i + \rho_i s^i (s^i)^\mathsf{T}, \tag{4.4}$$

for some $\rho_i \in \mathbb{R}$ and $s^i \in \mathbb{R}^n$. This obviously preserves symmetry of the approximation matrix R_i. To find a good choice of the vector s^i, consider the secant condition for $j = i$

$$p^i = R_{i+1} q^i = R_i q^i + \rho_i s^i (s^i)^\mathsf{T} q^i,$$

which implies by left multiplication with $(q^i)^\mathsf{T}$ that

$$(q^i)^\mathsf{T} p^i = (q^i)^\mathsf{T} R_i q^i + \rho_i ((s^i)^\mathsf{T} q^i)^2.$$

Thus eq. (4.4) becomes

$$R_{i+1} = R_i + \frac{(p^i - R_i q^i)(p^i - R_i q^i)^\mathsf{T}}{\rho_i ((s^i)^\mathsf{T} q^i)^2} = R_i + \frac{(p^i - R_i q^i)(p^i - R_i q^i)^\mathsf{T}}{(q^i)^\mathsf{T}(p^i - R_i q^i)}. \tag{4.5}$$

In other words, the secant condition for $j = i$ determines the rank-one correction. In fact, when f is a quadratic function, all secant conditions for $j \le i-1$ are also satisfied.

Theorem 4.4. *Suppose $f(x) = \frac{1}{2}x^\mathsf{T} H x + g^\mathsf{T} x$ for $x \in \mathbb{R}^n$. Given any initial symmetric $n \times n$ matrix R_0, any points $\{x^j\}_{i=0}^i$ with $p^j := x^{j+1} - x^j$ and $q^j := H(x^{j+1} - x^j)$, $j = 0, \dots, i-1$, the approximation matrices $\{R_j\}_{j=0}^i$ generated by eq. (4.5) satisfy*

$$p^j = R_{i+1} q^j, \quad \forall\, j = 0, 1, \dots, i.$$

Proof. We prove the assertion by induction. Suppose the secant condition is satisfied by some R_i and all p^j, q^j for $j \le i-1$, which is trivially true for $i = 0$. Then for any $j \le i$,

$$R_{i+1} q^j = R_i q^j + \frac{(q^j)^\mathsf{T}(p^i - R_i q^i)}{(q^i)^\mathsf{T}(p^i - R_i q^i)}(p^i - R_i q^i) = p^j + \frac{(q^j)^\mathsf{T} p^i - (p^j)^\mathsf{T} q^i}{(q^i)^\mathsf{T}(p^i - R_i q^i)}(p^i - R_i q^i),$$

by the induction hypothesis. Now from the relation that $(q^j)^\mathsf{T} p^i = (p^j)^\mathsf{T} H p^i = (p^j)^\mathsf{T} q^i$, we see that the numerator vanishes in the second term, completing the proof for the induction step. □

Theorem 4.4 would imply an n-step convergence, i.e., $R_n = H^{-1}$, for quadratic functions $f(x) = \frac{1}{2}x^\mathsf{T} H x + g^\mathsf{T} x$ for some $H \succ 0$, assuming that the updates $p^0, \dots, p^{n-1}$ are linearly independent. Consequently, the quasi-Newton method would terminate within $n+1$ steps for the quadratic function. However, there is a series drawback: the denominator $(q^i)^\mathsf{T}(p^i - R_i q^i)$ may not be positive, and consequently the approximation matrix R_{i+1} may not be positive definite (even when the true Hessian is). Thus even with exact line search steps, it is not always guaranteed that the quasi-Newton method with rank-one corrections is a descent method. Nevertheless, the line search quasi-Newton method (Algorithm 4.3) sometimes leads to good numerical results.

To ensure that $R_{i+1} \succ 0$, we can consider *rank-two corrections* to get R_{i+1} from R_i. The earliest such method is known known as the *Davidon-Fletcher-Powell* (DFP) method:

$$R_{i+1} = R_{i+1}^{\mathrm{DFP}} := R_i + \frac{p^i (p^i)^\mathsf{T}}{(p^i)^\mathsf{T} q^i} - \frac{R_i q^i (q^i)^\mathsf{T} R_i}{(q^i)^\mathsf{T} R_i q^i}. \tag{4.6}$$

Algorithm 4.3 Line Search Quasi-Newton Method

Require: $x^0 \in \mathbb{R}^n$, $R_0 \succ 0$, $\epsilon > 0$
1: **while** $\|g^i\| > \epsilon$ **do**
2: set $d^i \leftarrow -R_i g^i$
3: update $x^{i+1} \leftarrow x^i + \tau_i d^i$ where $\tau_i \in \arg\min_{\tau \geq 0} f(x^i + \tau d^i)$, and set $p^i = \tau_i d^i$
4: evaluate $g^{i+1} = \nabla f(x^{i+1})$ and set $q^i = g^{i+1} - g^i$
5: calculate R_{i+1} from R_i, p^i, and q^i using eq. (4.5), (4.6) or (4.11)
6: set $i \leftarrow i + 1$
7: **end while**

This obviously satisfies the secant condition $R_{i+1}^{\text{DFP}} q^i = p^i$. To show the positive definiteness of R_{i+1}^{DFP} assuming that of R_i, note that by the exactness of the line search steps, we have $\phi_i'(\tau_i) = (p^i)^\mathsf{T} g^{i+1} = 0$. This implies by the definition of q^i that

$$(p^i)^\mathsf{T} q^i = (p^i)^\mathsf{T}(g^{i+1} - g^i) = -(p^i)^\mathsf{T} g^i = \tau_i (g^i)^\mathsf{T} R_i g^i. \tag{4.7}$$

Take any $x \in \mathbb{R}^n$, the DFP update formula (4.6) shows that

$$x^\mathsf{T} R_{i+1}^{\text{DFP}} x = x^\mathsf{T} R_i x + \frac{(x^\mathsf{T} p^i)^2}{(p^i)^\mathsf{T} q^i} - \frac{(x^\mathsf{T} R_i q^i)^2}{(q^i)^\mathsf{T} R_i q^i}.$$

Rewriting $a := R_i^{1/2} x$ and $b := R_i^{1/2} q^i$, this becomes

$$x^\mathsf{T} R_{i+1}^{\text{DFP}} x = \frac{(a^\mathsf{T} a)(b^\mathsf{T} b) - (a^\mathsf{T} b)^2}{b^\mathsf{T} b} + \frac{(x^\mathsf{T} p^i)^2}{\tau_i (g^i)^\mathsf{T} R_i g^i} \geq 0.$$

In fact, the first term vanishes only if there exists $\lambda \in \mathbb{R}$ such that $a = \lambda b$, which says $x = \lambda q^i$ and $\lambda \neq 0$, and thus in this case

$$x^\mathsf{T} p^i = \lambda (q^i)^\mathsf{T} R_i p^i = \lambda \tau_i (g^i)^\mathsf{T} R_i g^i \neq 0 \implies x^\mathsf{T} R_{i+1} x = \lambda^2 \tau_i (g^i)^\mathsf{T} R_i g^i > 0.$$

Therefore, the DFP method preserves positive definiteness of R_i. We next show that it has the same finite convergence property when applied to strongly convex quadratic functions.

Theorem 4.5. *Let $f(x) = \frac{1}{2} x^\mathsf{T} H x + g^\mathsf{T} x$ be a quadratic function with $H \succ 0$. Then for any $x^0 \in \mathbb{R}^n$, the sequences generated by the DFP method (Algorithm 4.3 with (4.6)) satisfy*

$$\begin{aligned} (p^i)^\mathsf{T} H p^j &= 0, \qquad j = 0, 1, \ldots, i-1, \\ R_{i+1} H p^j &= p^j, \qquad j = 0, 1, \ldots, i. \end{aligned}$$

Proof. We first note that

$$R_{i+1}Hp^i = R_{i+1}q^i = \left(R_i + \frac{p^i(p^i)^\mathsf{T}}{(p^i)^\mathsf{T}q^i} - \frac{R_iq^i(q^i)^\mathsf{T}R_i}{(q^i)^\mathsf{T}R_iq^i}\right)q^i = p^i,$$

for any $i \geq 0$. We prove the assertions by induction on i: the base case $i = 0$ is shown by the above calculation. Suppose the assertions are true for $i-1$. For any $j < i$, we have $g^i = g^{j+1} + H(\sum_{k=j+1}^{i} p^k)$, which by the induction hypothesis $(p^j)^\mathsf{T}Hp^k = 0$ shows that $(p^j)^\mathsf{T}g^i = (p^j)^\mathsf{T}g^{j+1} = \phi_j'(\tau_j) = 0$. Hence using the induction hypothesis $R_iHp^j = p^j$, we see that $(p^j)^\mathsf{T}HR_ig^i = 0$. Thus since $p^i = -\tau_iR_ig^i$ and $\tau_i \neq 0$, we get $(p^j)^\mathsf{T}Hp^i = 0$, for any $j < i$, which completes the first part of the induction step.

Now since $R_iHp^j = p^j$ for any $j \leq i-1$ by the induction hypothesis, we have $(q^i)^\mathsf{T}R_iHp^j = (q^i)^\mathsf{T}p^j = (p^i)^\mathsf{T}Hp^j = 0$. Consequently, for $j \leq i-1$,

$$R_{i+1}Hp^j = R_iHp^j + \frac{p^i(p^i)^\mathsf{T}Hp^j}{(p^i)^\mathsf{T}q^i} - \frac{R_iq^i(q^i)^\mathsf{T}R_iHp^j}{(q^i)^\mathsf{T}R_iq^i} = R_iHp^j = p^j.$$

This together with the relation $R_{i+1}Hp^i = p^i$ shown in the beginning completes the second part of the induction step. □

Theorem 4.5 shows that the DFP method is actually a conjugate direction method! Thus by Theorem 4.1, we know that it finds an optimal solution to the strongly convex quadratic function in no more than n iterations. Another way to interpret Theorem 4.5 is to note that $p^0, \ldots, p^{n-1}$ are eigenvectors of the matrix R_nH, corresponding to the eigenvalue 1. As they are linearly independent by Theorem 4.1, we conclude that $R_nH = I$ and thus $R_n = H^{-1}$.

So far we have used the secant condition $R_{i+1}q^i = p^i$ for updating R_i, an approximation of the inverse of the Hessian $\nabla^2 f(x^i)$. If we want to directly approximate the Hessian using a matrix H_i, we can impose the secant condition $H_{i+1}p^i = q^i$. The symmetry between p^i and q^i in fact leads to the complementary *rank-one* and *rank-two* updates

$$H_{i+1} = H_i + \frac{(q^i - H_ip^i)(q^i - H_ip^i)^\mathsf{T}}{(p^i)^\mathsf{T}(q^i - H_ip^i)}, \tag{4.8}$$

and

$$H_{i+1} = H_i + \frac{q^i(q^i)^\mathsf{T}}{(q^i)^\mathsf{T}p^i} - \frac{H_ip^i(p^i)^\mathsf{T}H_i}{(p^i)H_ip^i}. \tag{4.9}$$

These formulas eqs. (4.8) and (4.9) can be used directly in a trust-region method (see Section 3.2). Alternatively, the updates of H_i translate into updates of R_i through the Sherman-Morrison formula:

$$(A + bc^\mathsf{T})^{-1} = A^{-1} - \frac{A^{-1}bc^\mathsf{T}A^{-1}}{1 + c^\mathsf{T}A^{-1}b}, \tag{4.10}$$

where A is an invertible $n \times n$ matrix and $b, c \in \mathbb{R}^n$, assuming that $1 + c^\mathsf{T} A^{-1} b \neq 0$. It can be checked that the rank-one correction for R_i we get from eq. (4.8) is the same as as eq. (4.4), and by using eq. (4.10) twice, we get the following formula for R_i:

$$\begin{aligned} R_{i+1} = R_{i+1}^{\text{BFGS}} :=& \left(I - \frac{p^i (q^i)^\mathsf{T}}{(p^i)^\mathsf{T} q^i}\right) R_i \left(I - \frac{q^i (p^i)^\mathsf{T}}{(p^i)^\mathsf{T} q^i}\right) + \frac{p^i (p^i)^\mathsf{T}}{(p^i)^\mathsf{T} q^i} \\ =& R_i + \left(1 + \frac{(q^i)^\mathsf{T} R_i q^i}{(p^i)^\mathsf{T} q^i}\right) \frac{p^i (p^i)^\mathsf{T}}{(p^i)^\mathsf{T} q^i} - \frac{p^i (q^i)^\mathsf{T} R_i + R_i q^i (p^i)^\mathsf{T}}{(p^i)^\mathsf{T} q^i}, \end{aligned} \tag{4.11}$$

which is known as the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) method. It is again easy to check that $R_{i+1}^{\text{BFGS}} q^i = p^i$, and assuming $(p^i)^\mathsf{T} q^i > 0$, $R_{i+1}^{\text{BFGS}} \succ 0$ whenever $R_i \succ 0$. Empirically the quasi-Newton method (Algorithm 4.3) with BFGS updates often performs better than the DFP updates. As both the DFP and the BFGS methods are rank-two corrections, they can be unified into a general framework, called *Broyden family*, defined as

$$R_{i+1}^{\phi} := (1 - \phi) R_{i+1}^{\text{DFP}} + \phi R_{i+1}^{\text{BFGS}}, \tag{4.12}$$

where $\phi \in \mathbb{R}$ is a preselected parameter, R_{i+1}^{DFP} and R_{i+1}^{BFGS} are calculated using eqs. (4.6) and (4.11) with some given R_i, p^i, and q^i.

Typically we would consider $0 \leq \phi \leq 1$ because the exactness of line search steps would ensure that $R_{i+1}^{\text{BFGS}} \succ 0$ (and hence R_{i+1}^{ϕ}) by eq. (4.7). In practice, we can let ϕ vary with the iteration index i. Discussions on how the choice of ϕ affects the quasi-Newton method, together with convergence analysis can be found in [NW06, Chapter 6]. Informally speaking, under the assumption of strong convexity and first-order Lipschitz continuity, the Broyden family quasi-Newton method would converge superlinearly to the minimum. In a simplest case of a strongly convex quadratic function, the convergence of Broyden family methods (including the BFGS method) can be shown using the same argument as in Theorem 4.5.

Exercise 4.6. *Let $f(x) = \frac{1}{2} x^\mathsf{T} H x + g^\mathsf{T} x$ be a quadratic function with $H \succ 0$. Then for any $x^0 \in \mathbb{R}^n$, the sequences generated by a Broyden family method (Algorithm 4.3 with (4.12) for some $\phi \in \mathbb{R}$) satisfy*

$$\begin{aligned} (p^i)^\mathsf{T} H p^j &= 0, \qquad j = 0, 1, \ldots, i-1, \\ R_{i+1} H p^j &= p^j, \qquad j = 0, 1, \ldots, i. \end{aligned}$$

Consequently, the method terminates within n iterations by Theorem 4.1.

For very large n, the storage of the approximation matrix R_i can be overly memory-consuming. The update formulas (4.6) or (4.11) can be applied on demand, given the past history of the vectors p^j and q^j, $j = i-1, \ldots, i-l$ for some $l \in \mathbb{Z}_{\geq 1}$ (Algorithm 4.4).

Here, l is the number of iterations we want to store the vectors p^j and q^j, which is typically chosen to be $l \ll n$, hence the name *limited-memory*. In the extreme case where

Algorithm 4.4 Limited-Memory Quasi-Newton Update

Require: p^j and q^j for $j = i-1, \ldots, i-l$
1: set $R_{i-l} = I$
2: **for** $j = i-l, i-l+1, \ldots, i-1$ **do**
3: calculate R_{j+1} from R_j, p^j, and q^j using eq. (4.5), (4.6) or (4.11)
4: set $j \leftarrow j+1$
5: **end for**
6: **return** R_i

$l = 1$, it is also called *memoryless*, and the BFGS update formula in this case becomes

$$R_{i+1}^{\text{1BFGS}} := I - \frac{q^i (p^i)^\mathsf{T} + p^i (q^i)^\mathsf{T}}{(p^i)^\mathsf{T} q^i} + \left(1 + \frac{(q^i)^\mathsf{T} q^i}{(p^i)^\mathsf{T} p^i}\right) \frac{p^i (p^i)^\mathsf{T}}{(p^i)^\mathsf{T} q^i}. \tag{4.13}$$

As a result, the line search direction in the BFGS quasi-Newton method (Algorithm 4.3) becomes

$$d^{i+1} = -g^{i+1} + \frac{q^i (p^i)^\mathsf{T} g^{i+1} + p^i (q^i)^\mathsf{T} g^{i+1}}{(p^i)^\mathsf{T} q^i} - + \left(1 + \frac{(q^i)^\mathsf{T} q^i}{(p^i)^\mathsf{T} p^i}\right) \frac{p^i (p^i)^\mathsf{T} g^{i+1}}{(p^i)^\mathsf{T} q^i}. \tag{4.14}$$

Again by the exactness of the line search steps, we have $(p^i)^\mathsf{T} g^{i+1} = 0$, so eq. (4.14) is simplified to

$$d^{i+1} = -g^{i+1} + \frac{(q^i)^\mathsf{T} g^{i+1}}{(p^i)^\mathsf{T} q^i} p^i = -g^{i+1} + \sigma_i d^i, \tag{4.15}$$

where we use the fact $(p^i)^\mathsf{T} q^i = \tau_i (g^i)^\mathsf{T} R_i g^i$ by eq. (4.7) and $\sigma_i = \dfrac{(g^{i+1})^\mathsf{T} (g^{i+1} - g^i)}{(g^i)^\mathsf{T} g^i}$ is the Polak-Ribiere conjugate gradient coefficient eq. (4.2). Thus the limited-memory BFGS quasi-Newton method can be viewed as an intermediate method between the general conjugate gradient method (Algorithm 4.2) and the full-memory quasi-Newton method (Algorithm 4.3). More discussion can be found in [NW06, Chapter 7].

5 Essentials of Constrained Optimization

5.1 Optimality Conditions and Constraint Qualification

We turn our attention to constrained optimization problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s. t.} \quad & g_i(x) = 0, \qquad i = 1, \ldots, m', \\ & g_i(x) \le 0, \qquad i = m'+1, \ldots, m. \end{aligned} \tag{5.1}$$

Analogous to the unconstrained case, we say that a point $x^* \in \mathbb{R}^n$ is *a local minimum* if there exists $\epsilon > 0$ such that $f(x^*) \le f(x)$ for any $x \ne x^*$, $x \in X := \{x \in \mathbb{R}^n : g_i(x) = 0,\ i = 1, \dots, m',\ g_i(x) \le 0,\ i = m'+1, \dots, m\}$ with $\|x - x^*\| < \epsilon$; *a global minimum* if $f(x^*) \le f(x)$ for all $x \in X \setminus \{x^*\}$; and is said to be *strict* if $f(x^*) < f(x)$ holds in either definition. The goal of this section is to characterize the optimality conditions for eq. (5.1) parallel to those studied in Section 2.1.

As the feasibility set X is defined by nonlinear constraint functions g_i, we are interested in the *tangent directions* at a given point $x \in X$, i.e., the directions we can locally reach from the x and stay within X. Geometrically it can be defined as follows.

Definition 5.1. *Let $X \subseteq \mathbb{R}^n$ and $x \in X$. The* tangent cone *of X at x, denoted as $T_x(X)$, consists of all directions $d \in \mathbb{R}^n$ such that there exist a sequence $\{y^i\}_{i=1}^\infty \subset X$ and $\{\eta_i\}_{i=1}^\infty \subset \mathbb{R}_{>0}$ with $\lim_{i\to\infty} \eta_i = 0$ and $\lim_{i\to\infty} \dfrac{1}{\eta_i}(y^i - x) = d$.*

In general, a *cone* $C \subseteq \mathbb{R}^n$ often refers to a subset that is invariant under positive scaling, i.e., for any $d \in C$ and $\rho > 0$, $\rho d \in C$. The tangent cone $T_x(X)$ is indeed a cone because for any $d \in T_x(X)$, we can replace each η_i with η_i/ρ in Definition 5.1 and get $\rho d \in T_x(X)$. Moreover, it is closed because for any $d^i \to d$, $\{d^i\} \subset T_x(X)$, we can pick an index $j = j(i)$ for $\{y^{ij}\}$ and $\{\eta_{ij}\}$ in the definition such that $\lim_{i\to\infty} \frac{1}{\eta_{ij}}(y^{ij} - d) = 0$. However, the definition of tangent cones is not very convenient to use as it is based on limits of sequences, which motivates us to consider the following alternative characterization. To check in which direction we can move away from our point $x \in X$, a heuristic strategy is to linearize all the functions using first-order Taylor approximation (assuming that they are continuously differentiable) at the point x:

$$\begin{aligned} \min \quad & \nabla f(x)^\mathsf{T} d \\ \text{s.t.} \quad & \nabla g_i(x)^\mathsf{T} d = 0, \qquad i \in E := \{1, \dots, m'\}, \\ & \nabla g_i(x)^\mathsf{T} d \le 0, \qquad i \in A(x), \end{aligned} \tag{5.2}$$

where $A(x) := \{m'+1 \le i \le m : g_i(x) = 0\}$ is the *active set of inequality constraints* at the point x. It is easy to check that any direction $d \in T_x(X)$ must satisfy the linearized constraints in eq. (5.2). For example, if $\nabla g_i(x)^\mathsf{T} d > 0$ for some $i \in A(x)$, then there exists $\{y^j\}_j \subset X$ such that $\nabla g_i(x)^\mathsf{T}(y^i - x) > 0$ for all sufficiently large j. Thus from the continuity of ∇g_i and Lemma 2.7, we know that for large j, $g_i(y^j) = g_i(x) + \nabla g_i(x + s(y^j - x))^\mathsf{T}(y^j - x) > g_i(x) = 0$, for some $0 \le s \le 1$, contradicting $y^j \in X$.

The observation can be stated more concisely using the notion of *dual cones* in convex geometry. Given a cone $C \subseteq \mathbb{R}^n$, its dual cone is defined as $C^* := \{d \in \mathbb{R}^n : d^\mathsf{T} c \ge 0,\ \forall\, c \in C\}$, which is closed and convex by Exercise 1.6.

Exercise 5.2. *Given cones $C_1 \subseteq C_2 \subseteq \mathbb{R}^n$, the dual cones satisfy the reverse containment $C_2^* \subseteq C_1^*$.*

For the problem (5.1), let $G := \{g_1, \ldots, g_m\}$ and

$$N_x(G) := \left\{ \sum_{i \in E \cup A(x)} \lambda_i \nabla g_i(x) : \lambda_i \geq 0,\ i \in A(x),\ \text{and } \lambda_i \in \mathbb{R},\ i \in E \right\}$$

denote the convex cone generated by the gradients of active constraint functions, which is often called the *normal cone* at x. Similarly, let

$$L_x(G) := \left\{ d \in \mathbb{R}^n : \nabla g_i^\mathsf{T} d = 0,\ i \in E,\ \nabla g_i^\mathsf{T} d \leq 0,\ i \in A(x) \right\}$$

denote the *linearized tangent cone* at x as constructed in the problem (5.2). Then $L_x(G) = -N_x(G)^*$ and $N_x(G) \subseteq -T_x(X)^*$ by the above discussion.

Definition 5.3. *We say* (Guignard) constraint qualification *holds at a point $x \in X$ if $N_x(G) = -T_x(X)^*$. A constrained optimization problem* (5.1) *has constraint qualification if the constraint qualification holds at all of its local minima.*

Now suppose $x^* \in X$ is a local minimum. Using the same argument, we must have $\nabla f(x^*)^\mathsf{T} d \geq 0$ for any $d \in T_x(X)$, or equivalently $\nabla f(x^*) \in T_x(X)^*$. Thus given constraint qualification, we are ready to write down a first-order necessary condition for local minima, known as the *Karush-Kuhn-Tucker* (KKT) condition.

Theorem 5.4. *Suppose $f, g_1, \ldots, g_m \in C^1(\mathbb{R}^n)$ and the problem* (5.1) *has constraint qualification. If $x^* \in X$ is a local minimum, then there exist $\lambda \in \mathbb{R}^m$ such that*

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) &= 0, \\ \lambda_i g_i(x^*) &= 0, \qquad i = 1, \ldots, m, \\ \lambda_i &\geq 0, \qquad i = m' + 1, \ldots, m. \end{aligned} \tag{5.3}$$

Proof. This follows directly from the way we define $N_x(G)$ and the constraint qualification $N_x(G) = -T_x(X)^*$. □

The vector λ in Theorem 5.4 is often called *Lagrange multipliers* and when there is only equality constraints (i.e., $m = m'$ in eq. (5.1)), the equations $\lambda_i g_i(x^*) = 0$ are automatically satisfied, and the KKT condition is the same as the *Lagrange condition* for constrained extrema in calculus.

Example 5.5. *Consider the problem*

$$\begin{aligned} \min_{x_1,x_2\in\mathbb{R}} \quad & f(x_1,x_2) := (x_1+1)^2+(x_2+1)^2 \\ \text{s.t.} \quad & g_1(x_1,x_2) := x_1x_2 = 0, \\ & g_2(x_1,x_2) := -x_1 \le 0, \\ & g_3(x_1,x_2) := -x_2 \le 0. \end{aligned}$$

We see that at the point $x^* = (0,0)$, the tangent cone $T_{x^*}(X)$ is generated by the vectors $(1,0)$ and $(0,1)$, and thus $-T_{x^*}(X)^* = \{(x_1,x_2) \in \mathbb{R}^2 : x_1 \le 0, x_2 \le 0\}$. The normal cone at x^* is generated by the vectors $(0,0)$, $(-1,0)$, and $(0,-1)$. Thus constraint qualification holds at x^* because $N_{x^*}(G) = -T_{x^*}(X)$. In this case, $\nabla f(x^*) = (2,2)$, so we can choose the Lagrangian multiplier to be $\lambda = (0,2,2)$ in Theorem 5.4.

An important observation is that the normal cone $N_x(G)$ is always generated by finitely many vectors through nonnegative scalar multiplication. Geometrically such cones are called polyhedral. To be more precise, for a given set of vectors $\{y^i\}_{i\in I} \subset \mathbb{R}^n$, we use $\text{cone}(\{y^i\}_{i\in I}) := \{\sum_{i\in I} a_i y^i : \text{ there exists a finite set } J \subseteq I \text{ such that } a_i \ge 0,\ i \in J,\ a_i = 0,\ i \notin J\}$ to denote the *cone generated by* $\{y^i\}_{i\in I}$. A cone is *polyhedral* if it can be generated by a finite set of vectors $\text{cone}(\{y^1,\dots,y^k\})$. Below is a useful fact about the dual cone of polyhedral cones.

Theorem 5.6 (Farkas' lemma)**.** *Let P be a polyhedral cone. Then P is closed. Consequently, for any $y \notin P$, there exists $x \in P^*$ such that $x^\mathsf{T} y < 0$.*

Proof. Let $\{y^1,\dots,y^k\}$ denote a generating set for the polyhedral cone P. Take any linearly independent subset $\{y^i\}_{i\in I} \subseteq \{y^1,\dots,y^k\}$ and we claim that the cone $P_I := \text{cone}(\{y^i\}_{i\in I})$ is closed. To see this, consider the $(|I|-1)$-simplex $\Delta_I := \{\sum_{i\in I} a_i y^i : \sum_{i\in I} a_i = 1,\ a_i \ge 0,\ i \in I\}$. It is easy to check that Δ_I is compact, i.e., closed and bounded. Moreover, $0 \notin \Delta_I$ because of the linear independence of $\{y^i\}_{i\in I}$. Now take any sequence $\{u^j\}_{j=1}^\infty \subset P_I$ such that $u^j \to u \in \mathbb{R}^n$. For each j, we can write $u^j = b_j w^j$ for some $b_j \ge 0$ and $w^j \in \Delta_I$ by the definition of P_I. By the compactness of Δ_I, there exist a subsequence $\{w^{j_l}\}_{l=1}^\infty$ and $w \in \Delta_I$ such that $w^{j_l} \to w$ by Lemma 1.4. Since $\|w^{j_l}\| > \frac{1}{2}\|w\| > 0$ for all sufficiently large l, and $\|u^{j_l}\| < 2\|u\|$, we know that $b_{j_l} < 4\|u\|/\|w\|$ is bounded. Thus by again taking a subsequence of j_l if necessary, we may assume that $b_{j_l} \to b$ for some $b \in \mathbb{R}_{\ge 0}$. Therefore $u = \lim_{l\to\infty} u^{j_l} = \lim_{l\to\infty} b_{j_l} w^{j_l} = bw \in P_I$, which shows that P_I is closed.

Note that P is a finite union of all such cones P_I with linearly independent set of generators I, we know that P is also closed. A direct argument for this is for any sequence $u^j \to u \in \mathbb{R}^n$, there exist a subsequence $\{u^{j_l}\}$ of $\{u^j\}$ and a subset $I \subseteq \{1,\dots,k\}$ such that $u^{j_l} \in P_I$, and hence $u^{j_l} \to u$ implies $u \in P_I \subseteq P$. To show the

existence of $x \in P^*$, by Theorem 1.10, there exists such x such that $x^\mathsf{T} y < \inf_{y' \in P} x^\mathsf{T} y'$. As P is a cone, the right-hand side must be 0, which completes the proof. □

Exercise 5.7. *For any polyhedral cone $P \subseteq \mathbb{R}^n$, the dual of the dual cone $P^{**} = P$.*

Next we discuss important cases where constraint qualification (and thus the KKT condition) holds. Perhaps the simplest case is that all constraints g_i are linear functions. In this case, it is easy to describe the tangent cone and using Theorem 5.6 one can show that the constraint qualification holds automatically.

Exercise 5.8. *Suppose $g_1, \ldots, g_m$ are linear functions. Then for any $x \in X$, the tangent cone $T_x(X) = L_x(G)$. Thus the constraint qualification $N_x(G) = -T_x(X)^*$ holds at any $x \in X$.*

For nonlinear constraints, while the constraint qualification needs to be assumed, it holds in many practical cases. To be more precise, let us consider the following specific constraint qualifications that are widely used in the literature.

Definition 5.9. *Let $x \in X$. We say that*

- *the* linear independence constraint qualification *(LICQ) holds at x if the gradients $\nabla g_i(x)$ are linearly independent for all $i \in E \cup A(x)$.*
- *the* Mangasarian-Fromovitz constraint qualification *(MFCQ) holds at x if the gradients $\nabla g_i(x)$ are linearly independent for $i \in E$, and there exists $d \in \mathbb{R}^n$ such that $\nabla g_i(x)^\mathsf{T} d < 0$ for all $i \in A(x)$ and $\nabla g_j(x)^\mathsf{T} d = 0$ for all $j \in E$.*

To show that LICQ and MFCQ are indeed constraint qualifications, we need the following *implicit function theorem* from calculus (the proof of which can be found in [Zor15, Section 8.5] for example). To simply our notation, we use $B_1^m(x; r) := \{y \in \mathbb{R}^m : |x_i - y_i| < r,\ i = 1, \ldots, m\}$ to denote the m-dimensional width-$(2r)$ box centered at the given point x.

Lemma 5.10. *Let $G : U \to \mathbb{R}^m$ be a k-times continuously differentiable map on an open subset $U \subset \mathbb{R}^{m+l}$ such that for some $(\bar{u}, \bar{v}) \in U$, we have $G(\bar{u}, \bar{v}) = 0$, and the differential $\nabla_u G(\bar{u}, \bar{v})$ is an invertible matrix, then there exist $\epsilon > 0$ and a k-times continuously differentiable map $h : B_1(\bar{u}; \epsilon) \to B_1(\bar{v}; \epsilon)$ such that*

$$G(u, v) = 0 \iff v = h(u), \quad \forall\, (u, v) \in B_1^m(\bar{u}; \epsilon) \times B_1^l(\bar{v}; \epsilon),$$

and $\nabla h(u) = -[\nabla_u G(u, h(u))]^{-1} \nabla_v G(u, h(u))$.

Theorem 5.11. *For any $x \in X$, MFCQ holds at x if LICQ holds at x. Moreover, if MFCQ holds at x, then $T_x(X) = L_x(G)$ and the constraint qualification $N_x(G) = -T_x(X)^*$ holds.*

Proof. We first show that LICQ implies MFCQ at x. It is clear that ∇g_i, $i \in E$ are linearly independent. To construct a desired $d \in \mathbb{R}^n$, consider the matrix $J(x) :=$

$(\nabla g_i^\top(x))_{i\in E\cup A(x)}$, which has full row rank $|E|+|A(x)| \le n$. Hence we can augment it (by adding rows) into a nonsingular $n \times n$ matrix $\bar{J}(x)$, which gives a unique solution $d \in \mathbb{R}^n$ to the system $\bar{J}(x)d = e$, where $e \in \mathbb{R}^m$ is the vector with $e_i = -1$ for any $i \in A(x)$ and 0 otherwise. This gives the desired d in MFCQ.

We next show that MFCQ is indeed a constraint qualification at x, i.e., $-N_x(G) = L_x(G)^* = T_x(X)^*$. By Exercise 5.2 and Exercise 5.7, it suffices to show that $L_x(G) \subseteq T_x(X)$ under MFCQ. Let $d \in L_x(G)$ and $d^\circ \in \mathbb{R}^n$ such that $\nabla g_i^\top d^\circ = 0$ for any $i \in E$, and $\nabla g_i^\top d^\circ < 0$ for all $i \in A(x)$, the existence of which is guaranteed by MFCQ. For any $0 \le t \le 1$, consider $e = e_t := d + td^\circ \in \mathbb{R}^n$ and we claim that $e \in T_x(X)$ for all sufficiently small $t > 0$. This ensures $d \in T_x(X)$ by taking $t \to 0$ and using the fact that $T_x(X)$ is closed.

It remains to show the claim. Consider a map $F : \mathbb{R}^{m'+1} \to \mathbb{R}^{m'}$ by $F_i(y,s) := g_i(x + se + J(x)^\top y)$ for each $i \in E$, where $J(x) := (\nabla g_i(x)^\top)_{i\in E}$ denotes the Jacobian matrix at x. The nonlinear equation $F(y,s) = 0$ has a solution $(0,0)$ with $\nabla_y F(0,0) = J(x)J(x)^\top$, which has full rank by the linear independence assumption of $\nabla g_i(x)$ for $i \in E$. Lemma 5.10 then shows that there exists a continuously differentiable map $y : (-\epsilon, \epsilon) \to \mathbb{R}^{m'}$ such that $y(0) = 0$, $F(y(s), s) = 0$, and $y'(s) = -\nabla_y F(y(s), s)^{-1} \nabla_s F(y(s), s)$ for all $-\epsilon < s < \epsilon$. Hence we have $y'(0) = 0$. Now put $x(s) := x + se + J(x)^\top y(s)$ for $-\epsilon < s < \epsilon$, which gives $x(0) = x$, $x'(0) = e$ and $g_i(x(s)) = 0$ for all $i \in E$. Moreover, $g_i(x(s)) < 0$ for all $i \notin E \cup A(x)$, and $g_i'(x(0)) = \nabla g_i^\top e < 0$ for all $i \in A(x)$. This ensures that $x(s) \in X$ for all sufficiently small $s > 0$ and thus $e = x'(0) \in T_x(X)$. □

One of the reasons that LICQ (or MFCQ) is favored is that it holds in *almost all* cases. To be more precise, Sard's theorem from differential geometry (see e.g., [Lee12, Chapter 6]) tells us that given a sufficiently smooth map $G : \mathbb{R}^n \to \mathbb{R}^m$ (assuming $m \le n$), the set $C := \{x \in \mathbb{R}^n : \operatorname{rank}(\nabla g_i(x)^\top)_{i=1}^m < m\}$ of rank-deficient Jacobian matrices has its image $G(C)$ of *Lebesgue measure zero* in $\mathbb{R}^m$. Informally speaking, this means that the perturbed problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) - \epsilon_i = 0, \qquad i = 1, \dots, m', \\ & g_i(x) - \epsilon_i \le 0, \qquad i = m'+1, \dots, m, \end{aligned} \tag{5.4}$$

for some random vector $\epsilon = (\epsilon_1, \dots, \epsilon_m)$ following a continuous distribution (e.g., multivariate normal distribution) on $\mathbb{R}^m$ will satisfy LICQ (and thus MFCQ) with probability one. In this sense, perturbation is often mentioned to assume that the optimization problem satisfies the constraint qualification. It is worth mentioning that, this does not automatically mean we will easily find the KKT multipliers $\lambda_1, \dots, \lambda_m$. In fact, such multipliers may only be found with very large norm (e.g., $|\lambda_i| = 10^{100}$),

leading to numerical difficulties for optimization algorithms in practice.

Another reason for using LICQ is that one can define second-order optimality conditions, analogous to the unconstrained problems. Let us define the *Lagrange function* associated with the constrained optimization (5.1)

$$L(x, \lambda) := f(x) + \sum_{i=1}^{m} \lambda_i g_i(x), \tag{5.5}$$

where $x \in \mathbb{R}^n$ and $\lambda \in \Lambda := \mathbb{R}^{m'} \times \mathbb{R}^{m-m'}_{\geq 0}$. Then the KKT condition at $x^* \in X$ translates to the existence of $\lambda^* \in \Lambda$ such that $\lambda_i^* g_i(x^*) = 0$ for $i = 1, \ldots, m$, and the first-order optimality condition $\nabla_x L(x^*, \lambda^*) = 0$ in the variable x. In particular, LICQ ensures that the multiplier λ^* must be unique whenever it exists. The second-order optimality conditions can be formulated as follows. Let $(x^*, \lambda^*) \in X \times \Lambda$ denote a KKT pair.

- (Necessary) If x^* is a local minimum for (5.1), then

$$d^\mathsf{T} \nabla^2_{xx} L(x^*, \lambda^*) d \geq 0$$

  for any $d \in \mathbb{R}^n$ such that $\nabla g_i(x^*)^\mathsf{T} d = 0$, $i \in E \cup A(x^*)$.
- (Sufficient) If for any $d \neq 0$ such that $\nabla g_i(x^*)^\mathsf{T} d = 0$, $i \in E \cup A_+(x^*)$,

$$d^\mathsf{T} \nabla^2_{xx} L(x^*, \lambda^*) d > 0,$$

  then x^* is a local minimum for (5.1). Here, we use $A_+(x^*) := \{m' + 1 \leq i \leq m : \lambda_i^* > 0\}$ to denote the indices of inequality constraints, associated with which the multiplier is positive.

A proof of this second-order optimality condition is based on a variant of the implicit function theorem and can be found in [BN23, Section 4.2].

When the constraints in (5.1) are convex (i.e., $g_1, \ldots, g_{m'}$ are linear functions, and $g_{m'+1}, \ldots, g_m$ are convex functions), there is a simple constraint qualification.

Definition 5.12. *We say the* Slater constraint qualification *or* Slater condition *holds for problem* (5.1) *if there exists $x^\circ \in X$ such that $g_i(x^\circ) = 0$ for $i = 1, \ldots, m'$ and $g_i(x^\circ) < 0$ for $i = m' + 1, \ldots, m$.*

Theorem 5.13. *Suppose $g_1, \ldots, g_{m'} : \mathbb{R}^n \to \mathbb{R}$ are linear functions, and $g_{m'+1}, \ldots, g_m : \mathbb{R}^n \to \mathbb{R}$ are convex functions. If the Slater condition holds for* (5.1)*, then constraint qualification holds at any $x \in X$.*

Proof. Without loss of generality, we may assume that $\{\nabla g_i\}_{i=1}^{m'}$ are linearly independent by taking a maximal such subset of them. Thus by Theorem 5.11, it suffices to show that for any $x \in X$, MFCQ holds at x. Take $d := x^\circ - x \in \mathbb{R}^n$. Clearly $\nabla g_i(x)^\mathsf{T} d =$

$g_i(x^\circ) - g_i(x) = 0$ for each $i \in E$. For any $i = m' + 1, \dots, m$, the convexity implies

$$g_i(x) + \nabla g_i(x)^\mathsf{T} d \le g(x^\circ) < 0,$$

by Theorem 1.12. As $g_i(x) = 0$ for $i \in A(x)$, we conclude that d is the desired direction in MFCQ. □

Theorem 5.13 shows a more verifiable constraint qualification for convex constraints. In fact, if the objective function is further assumed to be convex, then the KKT condition is also sufficient for (global) optimality, analogous to the unconstrained optimization problems.

Theorem 5.14. *Suppose $f, g_{m'+1}, \dots, g_m$ are convex and $g_1, \dots, g_{m'}$ are linear in (5.1). If the pair $(x^*, \lambda^*) \in X \times \Lambda$ satisfies the KKT condition, then x^* is a global minimum of (5.1).*

Proof. By assumption, the Lagrange function $L(x, \lambda)$ is a convex function in x for any $\lambda \in \Lambda$. Therefore by Theorem 1.12, we have

$$L(x, \lambda^*) \ge L(x^*, \lambda^*) + \nabla_x L(x^*, \lambda^*)^\mathsf{T}(x - x^*) = L(x^*, \lambda^*) = f(x^*),$$

because of the first-order condition $\nabla_x L(x^*, \lambda^*) = 0$ and the complementarity condition $\lambda_i^* g_i(x^*) = 0$. Now from $\lambda^* \in \Lambda$, we conclude that for any $x \in X$

$$f(x) \ge f(x) + \sum_{i=1}^m \lambda^* g_i(x) = L(x, \lambda^*) \ge f(x^*).$$ □

Theorem 5.14 can also be interpreted as a strong duality result (cf. linear optimization duality in [LY21, Chapter 3]). For any problem (5.1) (that is not necessarily convex), consider the *Lagrangian dual* function

$$\phi(\lambda) := \inf_{x \in \mathbb{R}^n} L(x, \lambda). \tag{5.6}$$

Exercise 5.15. *The dual function $\phi(\lambda)$ is a concave function on the set $\{\lambda \in \Lambda : \phi(\lambda) > -\infty\}$. Moreover, we have*

$$\sup_{\lambda \in \Lambda} \phi(\lambda) \le \inf_{x \in X} f(x).$$

Using this notation, Theorem 5.14 says that for a convex optimization problem (5.1), any KKT pair (x^*, λ^*) satisfies the equality in Exercise 5.15, because

$$\phi(\lambda^*) \le \sup_{\lambda \in \Lambda} \phi(\lambda) \le \inf_{x \in X} f(x) = f(x^*),$$

where the first equality is due to the first-order optimality condition $\nabla_x L(x^*, \lambda^*) = 0$, which shows $x^* \in \arg\min_{x \in \mathbb{R}^n} L(x, \lambda^*)$ by Theorem 2.8.

5.2 Introduction to Complexity Theory

It is natural to ask the question: how "difficult" is it to solve a (constrained) nonlinear optimization problem? The answer clearly depends on our definition of difficulty. Imagine there is an "omniscient" machine, which is often called an *oracle*, that answers a certain type of questions immediately (e.g., our nonlinear optimization problem), then what we ordinary people view as difficult can be easy for the machine. Thus for our discussion, we need to first define the notion of complexity.

By *problem class*, we refer to a family of problems sharing some basic properties. For example, if all of the constraints in (5.1) are linear and the objective function is quadratic, then all of such problems form a problem class, which is often referred to as *(linearly constrained) quadratic optimization*, and can be written as

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & x^\top Hx + g^\top x \\ \text{s.t.} \quad & a_i^\top x - b_i \le 0,\ i = 1,\dots,m, \end{aligned} \tag{5.7}$$

for some symmetric matrix $H \in \mathbb{R}^{n\times n}$, vectors $g, a_1, \dots, a_m \in \mathbb{R}^n$, and numbers $b_1, \dots, b_m \in \mathbb{R}$. An *instance* in a problem class is a problem with some specific data set. For example, an instance of the linearly constrained quadratic optimization can be

$$\begin{aligned} \min_{x\in\mathbb{R}^2} \quad & -x_1^2 - x_2^2 + x_1 + x_2 \\ \text{s.t.} \quad & -x_1 \le 0, \\ & x_1 - 1 \le 0, \\ & -x_2 \le 0, \\ & x_2 - 1 \le 0. \end{aligned}$$

To measure the size of an instance, we assume that all the data $(H, g, a_1, \dots, a_m, b_1, \dots, b_m)$ consists of rational entries. For a rational number $p/q \in \mathbb{Q}$ where $p \in \mathbb{Z}$ and $q \in \mathbb{Z}_{>0}$ are integers, we define the *encoding size* of p/q as $1 + \lceil \log_2(|p|+1) \rceil + \lceil \log_2(|q|+1) \rceil$. The size of an instance is then the sum of all encoding sizes of the entries of its data.

To compare different instance sizes (and later different algorithms), we say that a function $\phi : S \to \mathbb{R}_{\ge 0}$ is *polynomially bounded* by another function $\psi : S \to \mathbb{R}_{\ge 0}$ if there exists a polynomial $\pi : \mathbb{R} \to \mathbb{R}$ such that $\phi(s) \le \pi(\psi(s))$ for any $s \in S$. When such polynomial π is linear, then people simply write $\phi(s) = O(\psi(s))$ for $s \in S$. For example, given a rational vector $v := (p_1/q_1, \dots, p_n/q_n) \in \mathbb{Q}^n$ with encoding size s, the encoding size of Lv is bounded $O(ns)$, where L is the least common multiple of $q_1, \dots, q_n$. This suggests that we may assume the data of (5.7) are all integers without any significant impact on the instance size.

Exercise 5.16. *Given a nonsingular rational matrix $A \in \mathbb{Q}^{n\times n}$ and a rational vector $b \in \mathbb{Q}^n$,*

the encoding size of the solution to $Ax = b$ is polynomially bounded by the encoding sizes of A and b.

An *algorithm* for solving a problem is a procedure that produces a correct answer in a finite amount of time on any given instance. Assuming an *oracle*, the *oracle-based time complexity* of an algorithm is the number of oracle calls it requires to solve any instance in the problem class, which is usually expressed through parameters defining the problem class (e.g., degrees of the polynomial functions in the objective or constraints), and the dimensions (e.g., n and m). Note that for instances with rational problem data that have bounded encoding sizes, the dimensions usually imply the encoding size for bounded-degree polynomial objective and constraints. We simply say *time complexity* if we only assume the oracle of basic arithmetics, meaning that we can precisely calculate the sum, difference, products, and quotients of any two rational numbers immediately, unless stated otherwise. An algorithm is said to have *polynomial time* if both the time complexity is polynomially bounded by the instance size. Similarly a notion of *space complexity* can be defined for an algorithm as the encoding size of the intermediate algorithmic data before an it terminates.

A *decision problem* is a problem whose answer is either "yes" or "no." Such problem is said to be in *complexity class* P if there exists a polynomial time algorithm to find the answer, whose space complexity is also polynomially bounded. We are also very interested in an important complexity class, named NP. A heuristic way to describe NP problem classes is that they admit a *certificate* to the "yes" answer that can be checked in polynomial time. Sometimes people also talk about co-NP class, in which there are certificates to the "no" answer that can be checked in polynomial time. We say a decision problem D in NP is *NP-complete* if all other problems D' in NP can be *reduced* to D in polynomial time. That is to say, there exists a polynomial time algorithm such that for every instance I of D', produces an instance of D whose answer is "yes" if and only if the answer to I is "yes." We can define a problem to be *co-NP-complete* in a similar way. Any decision problem D is called *NP-hard* if any problem D' in NP can be reduced to D in polynomial time. Intuitively speaking, if we can find a polynomial time algorithm to solve an NP-hard problem, then we can answer any decision problem in NP in polynomial time. From the definitions, a decision problem is NP-complete if and only if it is both in NP and is NP-hard.

In the context of optimization problems, such as $\min\{f(x) : x \in X\}$, its associated decision problem is usually the following:

$$\text{Given } v \in \mathbb{R}, \text{ does there exist } x \in X \subseteq \mathbb{R}^n \text{ such that } f(x) \leq v?$$

Such decision problem is in NP if for every $v \in \mathbb{Q}$, we can find a solution $x \in X \cap \mathbb{Q}^n$ with $f(x) \leq v$ and its encoding size polynomially bounded by the description of X and f (typically given by polynomial functions with rational coefficients), whenever the

answer is "yes." An example would the quadratic optimization problem (5.7). From the KKT condition (as the constraint qualification holds by Exercise 5.8) and Exercise 5.16, one can show the following.

Exercise 5.17. *The decision problem associated with any bounded quadratic optimization* (5.7) *is in NP.*

To show NP-hardness of a decision problem, we need to reduce a known NP-complete problem to it. A popular choice is the *3-satisfiability* (or 3-SAT) problem:

> Given a set of Boolean variables $x_1, \ldots, x_n \in \{0, 1\}$, and a Boolean expression
>
> $$z = \prod_{i=1}^{m} \max\{z_{i1}(x), z_{i2}(x), z_{i3}(x)\},$$
>
> where each $z_{ij}(x)$ is either x_k or $1 - x_k$ for some $k = 1, \ldots, n$, for each $i = 1, \ldots, m$, $j = 1, 2, 3$, does there exist a solution for $x_1, \ldots, x_n$ such that $z = 1$?

It is easy to see that the 3-SAT problem has an "yes" answer if and only if the linear constraints $z_{i1}(x) + z_{i2}(x) + z_{i3}(x) \geq 1$ hold for any $i = 1, \ldots, m$, and the Boolean restrictions $x_i \in \{0, 1\}$ can be enforced by $x_i^2 - x_i = 0$. Thus deciding whether general nonlinear constrained optimization problem is feasible is NP-hard. With a slightly more effort, we can show that this is also the case for linearly constrained quadratic optimization problem.

Theorem 5.18. *The decision problem associated with the quadratic optimization* (5.7) *is NP-hard.*

Proof. Given a 3-SAT instance (with Boolean variables $x_1, \ldots, x_n$), we construct a quadratic optimization instance with variables $y_1, \ldots, y_n$ as follows

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} y_i(1 - y_i) \\ \text{s.t.} \quad & 0 \leq y_i \leq 1, \\ & z_{i1}(y) + z_{i2}(y) + z_{i3}(y) \geq 1, \qquad i = 1, \ldots, m. \end{aligned}$$

It is straightforward to check that this quadratic optimization problem size is polynomially bounded by n and m, and thus polynomially by the encoding size of the 3-SAT problem. We claim that the answer to the 3-SAT problem is "yes" if and only if the minimum value of the quadratic optimization is less than or equal to 0. To see the if direction, note that $y_i(1 - y_i) \geq 0$ for each i, so the minimum value being 0 implies that $y_i \in \{0, 1\}$. Thus we can simply set x_i to be the associated optimal solution of y_i. To see the only-if direction, we simply set $y_i = x_i$ for each $i = 1, \ldots, n$, which clearly satisfies the constraints by definition of the 3-SAT problem. □

Theorem 5.18 suggests that finding a global minimum of a linearly constrained quadratic optimization problem can be very challenging in general. It is slightly surprising that, even checking the local minimality of a feasible point is also NP-hard. The proof can be found in the classical paper [MK85] and is closely related to the NP-hardness of the following problem:

Given a symmetric matrix $H \in \mathbb{Q}^{n\times n}$, is $x^\top Hx \geq 0$ for all $x \in \mathbb{R}^n_{\geq 0}$?

Such matrices are called *copositive* and form a convex cone in $\mathbb{R}^{n\times n}$. They have been used for reformulation of many NP-hard problems.

Our discussion shows that, in general, we should not expect to solve nonlinear optimization problems to a global or local minimum certifiably in polynomial time (unless one expects to do so for any problems in the NP class). In practice, people aim at finding a point that satisfies the KKT condition as a tractable surrogate for local minimum points. There is, however, a nice exception where all of the objective and constraint functions are convex. In the rest of this section, we outline a polynomial-time algorithm (in terms of the dimensions), named *ellipsoid method*, for handling such convex nonlinear optimization problems.

From now on we only assume that $f, g_1, \ldots, g_m$ in (5.1) are continuously differentiable, and the gradient of each can be evaluated in $O(n)$ time. As we may be dealing with general nonlinear functions, it is no longer sufficient to assume exact arithmetics only for rational numbers. Instead, we assume the oracle for *real number arithmetics*, i.e., we can represent and take basic arithmetics of real numbers with fixed computational cost. This is of course an idealized assumption, but it helps us focus on the difficulty from the optimization problem, not from doing the arithmetics.

The decision problem we face is the following: given $v \in \mathbb{R}$,

does there exist $x \in X(v) := \{x \in X : f(x) \leq v\}$?

We make the following assumption on the problem class under consideration.

Assumption 5.19. *Suppose $f, g_1, \ldots, g_m$ are convex functions with $m' = 0$ in (5.1). For the level set $X(v) \subseteq \mathbb{R}^n$, we assume that*

(i) *we can find $y \in \mathbb{R}^n$ and $R > 0$ such that $X(v) \subseteq B_R(y) := \{x \in \mathbb{R}^n : \|x - y\|_2 \leq R\}$; and*

(ii) *there exist $y' \in \mathbb{R}^n$ and $r > 0$ such that $B_r(y') \subseteq X(v)$.*

The first condition in Assumption 5.19 is not hard to satisfy, as in many practical problems we would have a bounded feasibility set X, so $X(v)$ would also be bounded by the same radius. The second condition in Assumption 5.19 is for later convergence analysis and basically says that the set should not be too small or too "narrow" which can be quantified through the radius of the ball contained in its interior. This radius r is

related to the optimality gap $v - f^*$ via the Lipschitz or first-order Lipschitz constant of f on X. The assumption $m' = 0$ is because otherwise our variable would be inevitably contained in an affine subspace defined by the linear equality constraints, so $X(v)$ does not have an interior and fails the second condition. However, we may restrict our attention to a lower dimensional space $\mathbb{R}^{n'}$ with $n' \leq n$ through an affine change of coordinates so the assumption is not restrictive for our discussion on the complexity.

Before we present a polynomial-time algorithm, recall that an *ellipsoid* is the set $E(y, Q) := \{x \in \mathbb{R}^n : (x-y)^\mathsf{T} Q(x-y) \leq 1\}$ parametrized by the center $y \in \mathbb{R}^n$ and the $n \times n$ positive definite matrix $Q \succ 0$. Clearly $B_r(y) = E(y, \frac{1}{r^2} I)$. An ellipsoid is an affine linear image of a n-dimensional ball, which tells us its volume by $\operatorname{vol} E(y, Q) = \operatorname{vol} B_1(0) \det(Q^{-1/2})$. When the defining matrix $Q \succeq 0$ is only positive semidefinite, we call such $E(y, Q)$ a *degenerate* ellipsoid.

We are now ready to present the *ellipsoid algorithm* for solving the decision problem. The main idea is that for the intersection $E(y^i, Q_i) \cap \{x \in \mathbb{R}^n : (h^i)^\mathsf{T} x \leq (h^i)^\mathsf{T} y^i\}$ in the iteration i, we can find a new ellipsoid $E(y^{i+1}, Q_{i+1})$ containing the intersection where

$$\begin{aligned} y^{i+1} &:= y^i - \frac{1}{n+1} \frac{Q_i^{-1} h^i}{((h^i)^\mathsf{T} Q_i^{-1} h^i)^{1/2}}, \\ Q_{i+1} &:= \frac{n^2-1}{n^2} \left(Q_i^{-1} - \frac{2}{n+1} \frac{Q_i^{-1} h^i (h^i)^\mathsf{T} Q_i^{-1}}{(h^i)^\mathsf{T} Q_i^{-1} h^i} \right)^{-1}. \end{aligned} \tag{5.8}$$

Algorithm 5.1 Ellipsoid Algorithm

Require: $x^0 \in \mathbb{R}^n$, $Q_0 := \frac{1}{R^2} I \succ 0$ such that $X(v) \subseteq E(x^0, Q_0)$

while $\max\{f(x^i) - v, g_1(x^i), \ldots, g_m(x^i)\} > 0$ **do**
  if $f(x^i) > v$ **then**
    let $h^i := \nabla f(x^i)$
  else
    for $j = 1, \ldots, m$ **do**
      if $g_j(x^i) > 0$ **then**
        let $h^i := \nabla g_j(x^i)$
      end if
    end for
  end if
  set Q_{i+1} and y^{i+1} using eq. (5.8) and let $i \leftarrow i + 1$
end while

The convergence of Algorithm 5.1 is based on the following observation, the proof for which is mainly calculation and can be found in [LY21, Chapter 5.3] for example.

Lemma 5.20. *The ellipsoid* $E(y^{i+1}, Q_{i+1}) \supseteq E(y^i, Q_i) \cap \{x \in \mathbb{R}^n : (h^i)^\mathsf{T} x \leq (h^i)^\mathsf{T} y^i\}$ *for*

each $i = 0, 1, \ldots$. *Moreover,*

$$\frac{\operatorname{vol} E(y^{i+1}, Q_{i+1})}{\operatorname{vol} E(y^i, Q_i)} = \frac{n}{n+1}\left(\frac{n^2}{n^2-1}\right)^{(n-1)/2} < \exp\left(-\frac{1}{2(n+1)}\right) < 1.$$

Theorem 5.21. *Under Assumption 5.19, Algorithm 5.1 returns* $y^i \in X(v)$ *with* $i \leq 2n(n+1)\log(\frac{R}{r})$. *Thus the total number of arithmetic operations is bounded by* $O(n^3 m \log(\frac{R}{r}))$.

Proof. If $y^i \notin X(v)$, then there either $f(x) \leq f(y^i)$ or $g_j(x) \leq g_j(y^i)$ holds for some $j = 1, \ldots, m$, for all $x \in X(v)$. Thus by convexity and Theorem 1.12, we must have $(h^i)^\top x \leq (h^i)^\top y^i$ for any $x \in X(v)$. Lemma 5.20 then ensures that $E(y^{i+1}, Q_{i+1}) \supseteq X(v) \supseteq B_r(y')$. Moreover, $\operatorname{vol} E(y^{i+1}, Q_{i+1}) \geq \operatorname{vol} B_r(y')$, which implies by the same lemma that

$$\operatorname{vol} B_1(0) \cdot R^n \exp\left(-\frac{i}{2(n+1)}\right) \geq \operatorname{vol} B_1(0) \cdot r^n.$$

Our assertion then follows by taking the logarithm on both sides. □

Theorem 5.21 certifies the polynomial time of the ellipsoid algorithm because the number of real number arithmetics is polynomially bounded by n, m, and $\log(1/r)$. Here, we do not talk about the encoding size anymore, but $\log(1/r)$ behaves in a similar way as the *accuracy* of the optimal value. We further remark that the key step in Algorithm 5.1 is finding the separating vector h^i, which is independent from the construction of the ellipsoids and hence also independent from the convergence. Thus the algorithm has been generalized to much broader problem classes without continuous differentiability or even without convexity, assuming the *separation oracle*, which returns the desired h^i in each iteration. For this reason, people nowadays are very interested in the complexity bounds for nonlinear optimization algorithms, beyond the traditional notion of convergence rates defined in Section 2.2.

6 Overview of Constrained Optimization Algorithms

Informally speaking, algorithms for constrained optimization (5.1) can be put into three categories: primal methods, dual methods, and primal-dual methods. Primal methods directly update *feasible* solution $x^i \in X$ in each iteration i. Dual methods update the multiplier $\lambda^i \in \Lambda$, in the hope of recovering the primal solution using the KKT condition and getting a lower bound in the convex case (see Theorem 5.14). Primal-dual methods aim at solving the KKT conditions directly. In between the primal and the dual methods, we also have *barrier* and *penalty* methods that turn the constrained optimization into a sequence of unconstrained optimization problems as surrogates.

6.1 Primal Methods

The challenge of applying the descent methods in Section 3 in a constrained setting is how to ensure feasibility of the iterates. For illustrative purpose, we will mostly restrict our attention to linear constraints $g_j(x) = (a^j)^\top x + b_j$ for some $a^j \in \mathbb{R}^n$ and $b_j \in \mathbb{R}$. Perhaps the simplest idea is to use a linear approximation of the nonlinear objective function, i.e., let d^i denote an optimal solution to the following linear optimization

$$\begin{aligned} \min_d \quad & \nabla f(x^i)^\top d \\ \text{s.t.} \quad & (a^j)^\top d = 0, && j = 1, \dots, m', \\ & (a^j)^\top (x^i + d) + b^j \le 0, && j = m'+1, \dots, m, \end{aligned} \tag{6.1}$$

and then set $x^{i+1} = x^i + \tau_i d^i$ where $\tau_i \in \arg\min_{0 \le \tau \le 1} f(x^i + \tau d^i)$ can be determined by the line search subproblem. This is a very simple form of the *Frank-Wolfe* method, or *reduced gradient* method. The latter name comes from the fact that all equality constraints in (6.1) are simply restricting d to an affine subspace. Assuming linear independence of $a^1, \dots, a^{m'}$, even though $d \in \mathbb{R}^n$, it is determined by only $n - m'$ of its components (which can be utilized by linear optimization methods, e.g., the simplex method). One can further add the restrictions $|d_i| \le \delta_i$ for some $\delta_i > 0$ in (6.1) to improve the approximation accuracy, similar to the trust region subproblem we discussed. Such method is often known as the *Zoutendijk method* of *feasible direction* method. Using linear optimization duality, it is easy to show that if the feasible direction method terminates with $d^i = 0$, then x^i is a KKT point.

In practice, the feasible direction method may not have global convergence and can be very numerically inefficient even for linear constraints. A more popular alternative is called the *active set* method. For any working subset $W_i \subseteq \{m'+1, \dots, m\}$, let $L_i := \{x \in \mathbb{R}^n : (a^j)^\top x + b_j = 0,\ j \in E \cup W_i\}$ denote an subspace of $\mathbb{R}^n$ defined by the equality constraints and the working inequality constraints in the iteration i. Then we solve an unconstrained optimization

$$y^i \in \arg\min_{y \in L_i} f(y), \tag{6.2}$$

and set $d^i := y^i - x^i$. The subproblem (6.2) is unconstrained in the sense that by choosing a basis Q of L_i and a vector $p \in \mathbb{R}^n$, any $y \in L_i$ can be written as $y = p + Qu$ for $u \in \mathbb{R}^{n - \dim(L_i)}$. To ensure feasibility of the next iterate x^{i+1}, we add a restriction $\tau_i \le \bar{\tau}_i$ to the line search step, where $\bar{\tau}_i := \sup\{\tau \ge 0 : (a^j)^\top (x^i + \tau d^i) + b^j \le 0,\ j \notin W_i\}$. Thus the iteration will be determined once we pick the subspace L_i.

When $\tau_i = \bar{\tau}_i$, it means that some new inequality constraint becomes active, and it is then natural to put it in the working set W_{i+1}. However, if we keep adding constraints

into the working set, it is possible that we may end up with a nondegenerate set of n equations, from which we cannot update our point any more. A good way to drop constraints from the working set W_{i+1} is to look at the KKT multipliers λ^i at the point x^i

$$\nabla f(x^i) + \sum_{j \in W_i} \lambda_j^i a^j = 0. \tag{6.3}$$

If $\lambda_j^i \geq 0$ for all $j \in W_i$, then one can extend it to a multiplier $\lambda^* \in \Lambda$ by setting $\lambda_j^* = 0$ for all $j \notin W_i$ and $\lambda_j^* = \lambda_j^i$ for all other j. Thus x^i is a KKT point so we may terminate the algorithm. Otherwise, we have $\lambda_j^i < 0$ for some $j \in W_i$. This means that we can locally improve our function value f by dropping the j-th constraint in the index set W_i. More precisely, we claim the following.

Exercise 6.1. *Suppose $\lambda_k^i < 0$ for some $k \in W_i$. Let d^i be the orthogonal projection of $\nabla f(x^i)$ on $L_i' := \{d \in \mathbb{R}^n : \nabla g_j(x^i)^\mathsf{T} d = 0,\ j \in W_i \setminus \{k\}\}$. Then d^i is a feasible descent direction, i.e., $\nabla g_k(x^i)^\mathsf{T} d^i < 0$ and $\nabla f(x^i)^\mathsf{T} d^i < 0$.*

Thus we update our set by

$$W_{i+1} \leftarrow W_i \cup A(x^{i+1}) \setminus \{j : \lambda_j^{i+1} < 0\}. \tag{6.4}$$

Convergence can be established after changing the working set for a finite number of times.

Theorem 6.2. *Suppose that for every subset W of $\{m' + 1, \ldots, m\}$, the unconstrained optimization problem (6.2) is well-defined and has a unique solution. Then the points generated by the active set method with working set updates (6.4) converges to a KKT point of (5.1) after a finite number of changes in the working set.*

Proof. The proof directly follows from Exercise 6.1: after each update of the working set W_i to W_{i+1}, we either find a KKT point when $\lambda_j^i \geq 0$ for all $j \in W_i$, or have a strict decrease in the objective when $\lambda_k^i < 0$ for some $k \in W_i$. In the latter case the working set W_i cannot be used again by our assumption of the uniqueness, so our method terminates in finitely many working set updates because there are only finitely many subsets W of $\{m' + 1, \ldots, m\}$. □

In some sense, the active set method can be viewed as a generalization of the simplex method for linear optimization, and it can be efficiently applied to linearly constrained convex quadratic optimization problems, where the solution to (6.2) can be found through matrix inversion. The active set method can also be extended to nonlinear constraints, assuming LICQ or some other regularity condition on the constraints $E \cup W_i$ in each iteration i. We refer to [BN23, Section 9.1] for more detailed discussion.

6.2 Barrier and Penalty Methods

The idea of transforming constrained optimization into unconstrained ones can be realized more directly through a *barrier* or *penalty* method. Consider an inequality-only constrained optimization problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_j(x) \le 0, \quad j = 1, \dots, m, \end{aligned} \tag{6.5}$$

We assume that Slater condition holds, i.e., $X := \{x \in \mathbb{R}^n : g_j(x) \le 0,\ j = 1, \dots, m\}$ has a nonempty interior and tackles the following problem as a surrogate for eq. (6.5)

$$\begin{aligned} \min \quad & f(x) + \frac{1}{c} b(x) \\ \text{s.t.} \quad & x \in \operatorname{int} X, \end{aligned} \tag{6.6}$$

where $c > 0$ is a chosen parameter and $b(x)$ is a *barrier function* satisfying

- $b(x)$ is continuous on $\operatorname{int} X$,
- $b(x)$ is bounded from below, i.e., there exists $a \in \mathbb{R}$ such that $b(x) \ge a$ for any $x \in X$, and
- $b(x) \to +\infty$ when x approaches the boundary of X.

If we start with some $x^0 \in \operatorname{int} X$ in (6.6), then the last condition will ensure we stay within the interior of X using any descent method. Thus the barrier method is often called *interior-point* method. Common choices for the barrier function include

- the reciprocal barrier function

$$b(x) = -\sum_{j=1}^{m} \frac{1}{-g_j(x)},$$

- and the logarithmic barrier function

$$b(x) = -\sum_{j=1}^{m} \log(-g_j(x)).$$

Exercise 6.3. *The reciprocal and logarithmic barrier functions are well-defined and satisfy all three conditions for barrier functions.*

In practice, the logarithmic barrier function is more popularly used. Note that while we ensure the feasibility of any solution returned by (6.6), we compromise its optimality in terms of the original constrained optimization problem (6.5). A larger value of the parameter $c > 0$ can mitigate this issue because $h(x, c) := f(x) + b(x)/c$ decreases for every x as c increases. Very often people take an adaptive strategy to increase c. Let $\{c_k\}_{k=0}^{\infty}$ be a monotone increasing sequence $c_{k+1} > c_k$ for each k, and let x^k denote the

(global) solution to (6.6) for $c = c_k$.

Theorem 6.4. *Any limit point of the sequence $\{x^k\}_{k=0}^{\infty}$ generated by the barrier method is a solution to the constrained optimization problem (6.5).*

Proof. The assertion is trivial if the sequence does not have any limit point. Thus by restricting our attention to a subsequence, we may assume $x^k \to x^*$ for some $x^* \in X$, as X is closed. Assume for contradiction that $x^* \notin \arg\min_{x \in X} f(x)$. Then we can find $x' \in \operatorname{int} X$ such that $f(x^*) > f(x') + \delta$ for some $\delta > 0$. Thus for any c_k, we have

$$f(x^k) + \frac{1}{c_k} b(x^k) = h(x^k, c_k) \leq h(x', c_k) = f(x') + \frac{1}{c_k} b(x').$$

Let $a := \inf_{x \in X} b(x) > -\infty$. It follows that

$$f(x^k) \leq f(x') + \frac{1}{c_k}(b(x') - a).$$

For all sufficiently large k, this implies that $f(x^k) \leq f(x') + \frac{\delta}{2} < f(x^*)$, which contradicts with the assumption $x^k \to x^*$ because of the continuity of f. □

When there are equality constraints, the interior of X may be empty, so it is not easy to apply the barrier method. Instead, we can consider a penalized problem

$$\min_{x \in \mathbb{R}^n} \quad f(x) + \rho p(x), \tag{6.7}$$

where $\rho > 0$ is a chosen parameter and $p(x)$ is a *penalty function* satisfying

- $p(x)$ is continuous and nonnegative for all $x \in \mathbb{R}^n$,
- $p(x) = 0$ if and only if $x \in X$.

Popular choices of penalty functions include

- the quadratic penalty function

$$p(x) = \sum_{j=1}^{m'} g_j^2(x) + \sum_{j=m'+1}^{m} \max\{g_j(x), 0\}^2,$$

- and the absolute value penalty function

$$p(x) = \sum_{j=1}^{m'} |g_j(x)| + \sum_{j=m'+1}^{m} \max\{g_j(x), 0\}.$$

Exercise 6.5. *The quadratic and absolute value penalty functions satisfy the two conditions for penalty functions. Moreover, the quadratic penalty function is continuously differentiable assuming that $g_1, \ldots, g_m$ are.*

As in the case of barrier methods, we would have optimal solutions asymptotically by increasing our parameter ρ. More precisely, let $\{\rho_k\}_{k=0}^{\infty}$ be a monotone increasing sequence with $\rho_{k+1} > \rho_k$ for each k. Suppose x^k is the global minimum to the penalty problem (6.7) with $\rho = \rho_k$. Then with a similar argument as in Theorem 6.4, one can prove the following.

Exercise 6.6. *Any limit point of the sequence $\{x^k\}_{k=0}^{\infty}$ generated by the penalty method is a solution to the constrained optimization problem (5.1).*

Unlike the barrier method, the penalty method does not always maintain the feasibility of the iterates x^k. It is then natural to ask whether we can ensure $x^k \in X$ if we set ρ_k to be large. The answer is negative for the quadratic penalty function, as illustrated by the following example.

Example 6.7. *Consider the constrained optimization*

$$\begin{aligned} \min \quad & x \\ \text{s.t.} \quad & x = 0, \end{aligned}$$

which obviously has the optimal value 0 and the feasibility set $X = \{0\}$. For any $\rho > 0$, the penalty surrogate is

$$\min \quad x + \rho x^2,$$

which has the optimal value $-\frac{1}{4\rho}$ with the unique solution $x^* = -\frac{1}{2\rho} \notin X$ for any $\rho > 0$.

Example 6.7 shows that the quadratic penalty function may not guarantee feasibility of the solution x^k no matter how large the parameter ρ_k is set. One way to work around this is to use the absolute value penalty function instead, which leads to *exactness* much more often than the quadratic penalty function. However, if we want to keep our objective function in the penalty subproblem (6.7) continuously differentiable, an alternative approach is to use the Lagrange multipliers as described in the next section.

6.3 Dual Methods

The name of dual methods comes from Lagrangian duality, as discussed in Section 5.1 for convex problems. To illustrate, let us assume that we only have linear equality constraints $g_j(x) = 0$, $j = 1, \ldots, m$. One way to see the effect of violating these constraints is through the *perturbation function*

$$p(u) := \min\{f(x) : g_j(x) = u_j,\ j = 1, \ldots, m\} \tag{6.8}$$

for some vector $u \in \mathbb{R}^m$. Then our constrained optimization problem is equivalent to the evaluation of $p(0)$, or

$$\begin{aligned} \min \quad & p(u) \\ \text{s.t.} \quad & u = 0. \end{aligned} \tag{6.9}$$

Exercise 6.8. *Suppose f is convex and $g_1, \ldots, g_m$ are linear functions, then the perturbation function p is also a convex function. Moreover, p is strongly convex if so is f.*

Assuming strong duality holds for (6.9), then we can solve the Lagrangian dual problem

$$\max_{\lambda \in \mathbb{R}^m} \min_{u \in \mathbb{R}^m} \quad p(u) + \lambda^\mathsf{T} u = \max_{\lambda \in \mathbb{R}^m} \min_{x \in \mathbb{R}^n} \quad f(x) + \sum_{j=1}^m \lambda_j g_j(x), \tag{6.10}$$

to get an optimal multiplier $\lambda^* \in \mathbb{R}^m$ such that $p(0) = \min_{u \in \mathbb{R}^n} \{p(u) + (\lambda^*)^\mathsf{T} u\}$. When the function f is further strongly convex, then by Exercise 6.8 we know there is a unique solution $u^* = 0$ and a unique solution

$$x^* \in \arg\min_{x \in \mathbb{R}^n} \left\{ f(x) + \sum_{j=1}^m \lambda_j^* g_j(x) \right\}, \tag{6.11}$$

so $g_j(x^*) = u_j^* = 0$ for $j = 1, \ldots, m$. In other words, the solution from the unconstrained problem (6.11) must be feasible to the constrained problem, which can be found efficiently using methods discussed in Sections 3 and 4. In general, instead of assuming f is strongly convex, we may assume that the Hessian $\nabla^2_{xx} L(x^*, \lambda^*) \succ 0$ is positive definite at a local minimum x^*, so that the solution is still well-defined and numerically attainable if we restrict to a neighborhood $B_\epsilon(x^*)$ in (6.11). In plain words, we may recover feasibility if we find an optimal multiplier and start our local unconstrained optimization iteration sufficiently close to the true local minimum.

Challenges arise when f is not strongly convex. Even in the context of linear optimization, it is not guaranteed that we will recover feasibility by solving the unconstrained problem (6.11). It is even worse if f is not convex, so that strong duality is quite unlikely to hold for (6.9). One way to tackle this is to "convexify" the problem by considering

$$\begin{aligned} \min \quad & p(u) + \rho \|u\|_2^2 \\ \text{s.t.} \quad & u = 0, \end{aligned} \tag{6.12}$$

for some chosen parameter $\rho > 0$. The reformulation (6.12) is equivalent to (6.9), but the sum $p(u) + \rho\|u\|_2^2$ can often become strongly convex for large ρ, e.g., when $p \in C^2(U)$ for some neighborhood U containing 0. In this case, we can solve the problem

$$\max_{\lambda \in \mathbb{R}^m} \min_{u \in \mathbb{R}^m} \quad p(u) + \lambda^\mathsf{T} u + \rho \|u\|_2^2, \tag{6.13}$$

or equivalently

$$\max_{\lambda\in\mathbb{R}^m}\min_{x\in\mathbb{R}^n} \quad L_\rho(x,\lambda) := f(x) + \lambda^\mathsf{T} g(x) + \rho\|g(x)\|_2^2, \tag{6.14}$$

where $g(x) := (g_1(x),\dots,g_m(x)) \in \mathbb{R}^m$. Problem (6.14) is known as the *augmented Lagrangian dual* to the original constrained problem. Once we obtain an optimal multiplier λ^* from (6.13) or (6.14), then we can recover a feasible and optimal solution from the unconstrained problem

$$x^* \in \arg\min_{x\in\mathbb{R}^n} L(x,\lambda^*). \tag{6.15}$$

So far we have been focusing on recovering feasible solutions after obtaining an optimal multiplier λ^*. Algorithms for finding such a multiplier can be developed based on the following observation. Let

$$\phi_\rho(\lambda) := \min\{p(u) + \lambda^\mathsf{T} u + \rho\|u\|_2^2 : u \in \mathbb{R}^m\} \tag{6.16}$$

denote the augmented Lagrangian dual function associated with the problem (6.12). Then the gradient of ϕ_ρ exists if the minimizer is unique in (6.16).

Exercise 6.9. *Suppose* $\{u^*\} = \arg\min\{p(u) + \lambda^\mathsf{T} u + \rho\|u\|_2^2 : u \in \mathbb{R}^m\}$ *for some* $\lambda \in \mathbb{R}^m$. *Then* $\nabla\phi(\lambda) = u^*$.

A simple example of the *dual ascent method* is the following. In each iteration i, we

- find $x^{i+1} \in \arg\min_{x\in\mathbb{R}^n} L_\rho(x,\lambda^i)$,
- and update $\lambda^{i+1} = \lambda^i + \frac{1}{\rho} g(x^{i+1})$.

The step length $\frac{1}{\rho}$ is set to be constant, which is a simplified (yet often very practical) version compared to the line search methods in Section 3.3 as it avoids the repeated evaluation of the inner minimization. It is noteworthy that the resulting sequence $\{\phi_\rho(\lambda^i)\}$ may not be monotone, so caution may be needed on when to terminate the algorithm.

Most of the discussion here can be generalized to inequality constraints. The augmented Lagrangian dual method also relates to the penalty method, as the term $\lambda^\mathsf{T} g(x)$ can be viewed as a special penalty function. We illustrate this by the following example (cf. Example 6.7).

Example 6.10. *Consider the constrained optimization in Example 6.7. For any* $\rho > 0$, *the augmented Lagrangian dual is*

$$\max_{\lambda}\min_{x} \quad x + \lambda x + \rho x^2.$$

By setting $\lambda^* = -1$, *we see that the unique optimal solution is* $x^* = 0$, *which is feasible to the original problem.*

References

[BN23] Aharon Ben-Tal and Arkadi Nemirovski. *Lecture Notes Optimization III: Convex Analysis, Nonlinear Programming Theory, Nonlinear Programming Algorithms*. 2023.

[CGT00] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000.

[Coh72] Arthur I Cohen. "Rate of convergence of several conjugate gradient algorithms". In: *SIAM Journal on Numerical Analysis* 9.2 (1972), pp. 248–259.

[HZ06] William W Hager and Hongchao Zhang. "A survey of nonlinear conjugate gradient methods". In: *Pacific journal of Optimization* 2.1 (2006), pp. 35–58.

[Lee12] John M Lee. *Introduction to Smooth Manifolds*. Spinger, 2012.

[LY21] David G Luenberger and Yinyu Ye. *Linear and nonlinear programming*. Vol. 2. Springer, 2021.

[MK85] Katta G Murty and Santosh N Kabadi. *Some NP-complete problems in quadratic and nonlinear programming*. Tech. rep. 1985.

[NW06] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

[Rud76] Walter Rudin. *Principles of mathematical analysis*. Vol. 3. McGraw-Hill New York, 1976.

[Zor15] Vladimir A. Zorich. *Mathematical Analysis I*. Springer-Verlag, Berlin, 2015.