CHAPTER 1

NONLINEAR PROGRAMMING

"Since the fabric of the universe is most perfect, and is the work of a most wise Creator, nothing whatsoever takes place in the universe in which some form of maximum and minimum does not appear."

—Leonhard Euler

1.1 INTRODUCTION

In this chapter, we introduce the nonlinear programming (NLP) problem. Our purpose is to provide some background on nonlinear problems; indeed, an exhaustive discussion of both theoretical and practical aspects of nonlinear programming can be the subject matter of an entire book.

There are several reasons for studying nonlinear programming in an optimal control class. First and foremost, anyone interested in optimal control should know about a number of fundamental results in nonlinear programming. As optimal control problems are optimization problems in (infinite-dimensional) functional spaces, while nonlinear programming are optimization problems in Euclidean spaces, optimal control can indeed be seen as a generalization of nonlinear programming.

Second and as we shall see in Chapter 3, NLP techniques are used routinely and are particularly efficient in solving optimal control problems. In the case of a *discrete* control problem, i.e., when the controls are exerted at discrete points, the problem can be directly stated as a NLP problem. In a *continuous* control problem, on the other hand, i.e., when the controls are *functions* to be exerted over a prescribed planning horizon, an *approximate* solution can be found by solving a NLP problem.

Throughout this section, we shall consider the following NLP problem:

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{x} \in X \end{array} \tag{NLP}$$

where X is a subset of $\mathrm{I\!R}^{n_x}$, $\mathbf{x}$ is a vector of n_x components $x_1, \ldots, x_{n_x}$, and $f : X \to \mathrm{I\!R}$, $\mathbf{g} : X \to \mathrm{I\!R}^{n_g}$ and $\mathbf{h} : X \to \mathrm{I\!R}^{n_h}$ are defined on X.

The function f is usually called the *objective function* or *criterion function*. Each of the constraints $g_i(\mathbf{x}) \leq 0$, $i = 1, \ldots, n_g$, is called an *inequality constraint*, and each of the constraints $h_i(\mathbf{x}) = 0$, $i = 1, \ldots, n_h$, is called an *equality constraint*. Note also that the set X typically includes lower and upper bounds on the variables; the reason for separating variable bounds from the other inequality constraints is that they can play a useful role in some algorithms, i.e., they are handled in a specific way. A vector $\mathbf{x} \in X$ satisfying all the constraints is called a *feasible solution* to the problem; the collection of all such points forms the *feasible region*. The NLP problem, then, is to find a feasible point $\mathbf{x}^\star$ such that $f(\mathbf{x}) \geq f(\mathbf{x}^\star)$ for each feasible point $\mathbf{x}$. Needless to say, a NLP problem can be stated as a maximization problem, and the inequality constraints can be written in the form $\mathbf{g}(\mathbf{x}) \geq \mathbf{0}$.

Example 1.1. Consider the following problem

$$\begin{array}{lrcl} \min\limits_{\mathbf{x}} & \multicolumn{3}{l}{(x_1-3)^2 + (x_2-2)^2} \\ \text{s.t.} & x_1^2 - x_2 - 3 & \leq & 0 \\ & x_2 - 1 & \leq & 0 \\ & -x_1 & \leq & 0. \end{array}$$

The objective function and the three inequality constraints are:

$$\begin{aligned} f(x_1,x_2) &= (x_1-3)^2 + (x_2-2)^2 \\ g_1(x_1,x_2) &= x_1^2 - x_2 - 3 \\ g_2(x_1,x_2) &= x_2 - 1 \\ g_3(x_1,x_2) &= -x_1. \end{aligned}$$

Fig. 1.1. illustrates the feasible region. The problem, then, is to find a point in the feasible region with the smallest possible value of $(x_1-3)^2 + (x_2-2)^2$. Note that points (x_1, x_2) with $(x_1-3)^2 + (x_2-2)^2 = c$ are circles with radius $\sqrt{c}$ and center $(3, 2)$. This circle is called the *contour* of the objective function having the value c. In order to minimize c, we must find the circle with the smallest radius that intersects the feasible region. As shown in Fig. 1.1., the smallest circle corresponds to $c = 2$ and intersects the feasible region at the point $(2, 1)$. Hence, the optimal solution occurs at the point $(2, 1)$ and has an objective value equal to 2.

The graphical approach used in Example 1.1 above, i.e., find an optimal solution by determining the objective contour with the smallest objective value that intersects the feasible region, is only suitable for small problems; it becomes intractable for problems containing

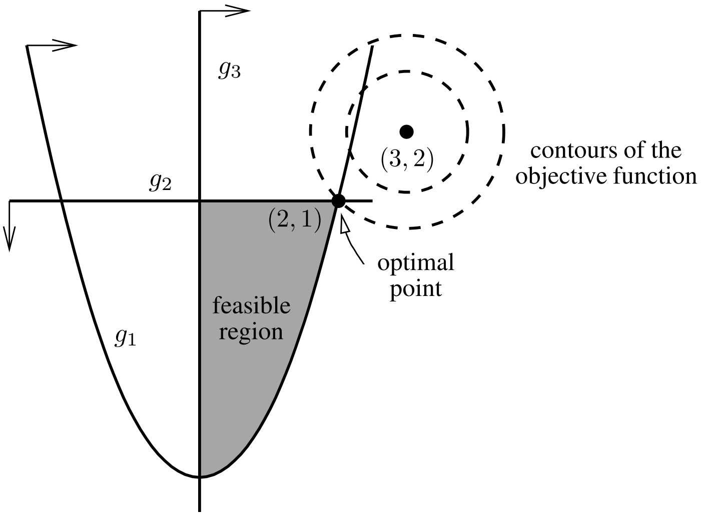


Figure 1.1. Geometric solution of a nonlinear problem.

more than three variables, as well as for problems having complicated objective and/or constraint functions.

This chapter is organized as follows. We start by defining what is meant by optimality, and give conditions under which a minimum (or a maximum) exists for a nonlinear program in §1.2. The special properties of convex programs are then discussed in §1.3. Then, both necessary and sufficient conditions of optimality are presented for NLP problems. We successively consider unconstrained problems (§1.4), problems with inequality constraints (§1.5), and problems with both equality and inequality constraints (§1.7). Finally, several numerical optimization techniques will be presented in §1.8, which are instrumental to solve a great variety of NLP problems.

1.2 DEFINITIONS OF OPTIMALITY

A variety of different definitions of optimality are used in different contexts. It is important to understand fully each definition and the context within which it is appropriately used.

1.2.1 Infimum and Supremum

Let $S \subset \mathbb{R}$ be a nonempty set.

Definition 1.2 (Infimum, Supremum). *The* infimum *of* S*, denoted as* $\inf S$*, provided it exists, is the* greatest lower bound *for* S*, i.e., a number* α *satisfying:*

(i) $z \geq \alpha \quad \forall z \in S$,

(ii) $\forall \bar{\alpha} > \alpha, \exists z \in S$ *such that* $z < \bar{\alpha}$.

Similarly, the supremum *of* S*, denoted as* $\sup S$*, provided it exists, is the* least upper bound *for* S*, i.e., a number* α *satisfying:*

(i) $z \leq \alpha \quad \forall z \in S$,

(ii) $\forall \bar{\alpha} < \alpha, \exists z \in S$ *such that* $z > \bar{\alpha}$.

The first question one may ask concerns the existence of infima and suprema in $\mathbb{R}$. In particular, one **cannot** prove that in $\mathbb{R}$, every set bounded from above has a supremum, and every set bounded from below has an infimum. This is an **axiom**, known as the *axiom of completeness*:

Axiom 1.3 (Axiom of Completeness). *If a nonempty subset of real numbers has an upper bound, then it has a least upper bound. If a nonempty subset of real numbers has a lower bound, it has a greatest lower bound.*

It is important to note that the real number $\inf S$ (resp. $\sup S$), with S a nonempty set in $\mathbb{R}$ bounded from below (resp. from above), although it exist, need not be an element of S.

Example 1.4. Let $S = (0, +\infty) = \{z \in \mathbb{R} : z > 0\}$. Clearly, $\inf S = 0$ and $0 \notin S$.

Notation 1.5. Let $S := \{f(\mathbf{x}) : \mathbf{x} \in D\}$ be the image of the feasible set $D \subset \mathbb{R}^n$ of an optimization problem under the objective function f. Then, the notation

$$\inf_{\mathbf{x} \in D} f(\mathbf{x}) \quad \text{or} \quad \inf\{f(\mathbf{x}) : \mathbf{x} \in D\}$$

refers to the number $\inf S$. Likewise, the notation

$$\sup_{\mathbf{x} \in D} f(\mathbf{x}) \quad \text{or} \quad \sup\{f(\mathbf{x}) : \mathbf{x} \in D\}$$

refers to $\sup S$.

Clearly, the numbers $\inf S$ and $\sup S$ may not be attained by the value $f(\mathbf{x})$ at any $\mathbf{x} \in D$. This is illustrated in an example below.

Example 1.6. Clearly, $\inf\{\exp(x) : x \in (0, +\infty)\} = 1$, but $\exp(x) > 1$ for all $x \in (0, +\infty)$.

By convention, the infimum of an empty set is $+\infty$, while the supremum of an empty set is $-\infty$. That is, if the values $\pm\infty$ are allowed, then infima and suprema always exist.

1.2.2 Minimum and Maximum

Consider the standard problem formulation

$$\min_{\mathbf{x} \in D} f(\mathbf{x})$$

where $D \subset \mathbb{R}^n$ denotes the *feasible set*. Any $\mathbf{x} \in D$ is said to be a *feasible point*; conversely, any $\mathbf{x} \in \mathbb{R}^n \setminus D := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \notin D\}$ is said to be *infeasible*.

Definition 1.7 ((Global) Minimum, Strict (Global) Minimum). *A point* $\mathbf{x}^\star \in D$ *is said to be a (global)*[1] minimum *of* f *on* D *if*

$$f(\mathbf{x}) \geq f(\mathbf{x}^\star) \quad \forall \mathbf{x} \in D, \tag{1.1}$$

i.e., a minimum is a feasible point whose objective function value is less than or equal to the objective function value of all other feasible points. It is said to be a strict *(global)* minimum *of* f *on* D *if*

$$f(\mathbf{x}) > f(\mathbf{x}^\star) \quad \forall \mathbf{x} \in D,\ \mathbf{x} \neq \mathbf{x}^\star.$$

A (global) maximum is defined by reversing the inequality in Definition 1.7:

Definition 1.8 ((Global) Maximum, Strict (Global) Maximum). *A point* $\mathbf{x}^\star \in D$ *is said to be a (global)* maximum *of* f *on* D *if*

$$f(\mathbf{x}) \leq f(\mathbf{x}^\star) \quad \forall \mathbf{x} \in D. \tag{1.2}$$

It is said to be a strict *(global)* maximum *of* f *on* D *if*

$$f(\mathbf{x}) < f(\mathbf{x}^\star) \quad \forall \mathbf{x} \in D,\ \mathbf{x} \neq \mathbf{x}^\star.$$

The important distinction between minimum/maximum and infimum/supremum is that the value $\min\{f(\mathbf{x}) : \mathbf{x} \in D\}$ **must** be attained at **one or more** points $\mathbf{x} \in D$, whereas the value $\inf\{f(\mathbf{x}) : \mathbf{x} \in D\}$ does not necessarily have to be attained at any points $\mathbf{x} \in D$. Yet, if a minimum (resp. maximum) exists, then its optimal value will equal the infimum (resp. supremum).

Note also that if a minimum exists, it is **not** necessarily **unique**. That is, there may be multiple or even an infinite number of feasible points that satisfy the inequality (1.1) and are thus minima. Since there is in general a set of points that are minima, the notation

$$\arg\min\{f(\mathbf{x}) : \mathbf{x} \in D\} \ := \ \{\mathbf{x} \in D : f(\mathbf{x}) = \inf\{f(\mathbf{x}) : \mathbf{x} \in D\}\}$$

is introduced to denote the set of minima; this is a (possibly empty) **set** in $\mathbb{R}^n$.[2]

A minimum $\mathbf{x}^\star$ is often referred to as an *optimal solution*, a *global optimal solution*, or simply a *solution* of the optimization problem. The real number $f(\mathbf{x}^\star)$ is known as the (global) *optimal value* or *optimal solution value*. Regardless of the number of minima, there is always a **unique** real number that is the optimal value (if it exists). (The notation $\min\{f(\mathbf{x}) : \mathbf{x} \in D\}$ is used to refer to this real value.)

Unless the objective function f and the feasible set D possess special properties (e.g., convexity), it is usually very hard to devise algorithms that are capable of locating or estimating a global minimum or a global maximum with certainty. This motivates the definition of local minima and maxima, which, by the nature of their definition in terms of local information, are much more convenient to locate with an algorithm.

Definition 1.9 (Local Minimum, Strict Local Minimum). *A point* $\mathbf{x}^\star \in D$ *is said to be a* local minimum *of* f *on* D *if*

$$\exists \varepsilon > 0 \text{ such that } f(\mathbf{x}) \geq f(\mathbf{x}^\star) \quad \forall \mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^\star) \cap D.$$

[1] Strictly, it is not necessary to qualify minimum with ‘global’ because minimum means a feasible point at which the smallest objective function value is attained. Yet, the qualification global minimum is often made to emphasize that a local minimum is not adequate.

[2] The notation $\bar{\mathbf{x}} = \arg\min\{f(\mathbf{x}) : \mathbf{x} \in D\}$ is also used by some authors. In this case, $\arg\min\{f(\mathbf{x}) : \mathbf{x} \in D\}$ should be understood as a function returning a point $\bar{\mathbf{x}}$ that minimizes f on D. (See, e.g., `http://planetmath.org/encyclopedia/ArgMin.html`.)

$\mathbf{x}^\star$ *is said to be a* strict local minimum *if*

$$\exists\varepsilon > 0 \textit{ such that } f(\mathbf{x}) > f(\mathbf{x}^\star) \quad \forall \mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^\star) \setminus \{\mathbf{x}^\star\} \cap D.$$

The qualifier 'local' originates from the requirement that $\mathbf{x}^\star$ be a minimum only for those feasible points in a neighborhood around the local minimum.

Remark 1.10. Trivially, the property of $\mathbf{x}^\star$ being a global minimum implies that $\mathbf{x}^\star$ is also a local minimum because a global minimum is local minimum with ε set arbitrarily large.

A local maximum is defined by reversing the inequalities in Definition 1.9:

Definition 1.11 (Local Maximum, Strict Local Maximum). *A point* $\mathbf{x}^\star \in D$ *is said to be a* local maximum *of* f *on* D *if*

$$\exists\varepsilon > 0 \textit{ such that } f(\mathbf{x}) \leq f(\mathbf{x}^\star) \quad \forall \mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^\star) \cap D.$$

$\mathbf{x}^\star$ *is said to be a* strict local maximum *if*

$$\exists\varepsilon > 0 \textit{ such that } f(\mathbf{x}) < f(\mathbf{x}^\star) \quad \forall \mathbf{x} \in \mathcal{B}_\varepsilon(\mathbf{x}^\star) \setminus \{\mathbf{x}^\star\} \cap D.$$

Remark 1.12. It is important to note that the concept of a global minimum or a global maximum of a function on a set is defined without the notion of a *distance* (or a norm in the case of a vector space). In contrast, the definition of a local minimum or a local maximum requires that a distance be specified on the set of interest. In $\mathbb{R}^{n_x}$, norms are equivalent, and it is readily shown that local minima (resp. maxima) in $(\mathbb{R}^{n_x}, \|\cdot\|_\alpha)$ match local minima (resp. maxima) in $(\mathbb{R}^{n_x}, \|\cdot\|_\beta)$, for any two arbitrary norms $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ in $\mathbb{R}^{n_x}$ (e.g., the Euclidean norm $\|\cdot\|_2$ and the infinite norm $\|\cdot\|_\infty$). Yet, this nice property does not hold in linear functional spaces, as those encountered in problems of the calculus of variations (§2) and optimal control (§3).

Fig. 1.2. illustrates the various definitions of minima and maxima. Point x^1 is the unique global maximum; the objective value at this point is also the supremum. Points a, x^2, and b are strict local minima because there exists a neighborhood around each of these point for which a, x^2, or b is the unique minimum (on the intersection of this neighborhood with the feasible set D). Likewise, point x^3 is a strict local maximum. Point x^4 is the unique global minimum; the objective value at this point is also the infimum. Finally, point x^5 is simultaneously a local minimum and a local maximum because there are neighborhoods for which the objective function remains constant over the entire neighborhood; it is neither a strict local minimum, nor a strict local maximum.

Example 1.13. Consider the function

$$f(x) = \begin{cases} +1 & \text{if } x < 0 \\ -1 & \text{otherwise.} \end{cases} \tag{1.3}$$

The point $x^\star = -1$ is a local minimum for

$$\min_{x \in [-2,2]} f(x)$$

with value $f(x^\star) = +1$. The optimal value of (1.3) is -1, and $\arg\min\{f(x) : x \in [-2,2]\} = [0,2]$.

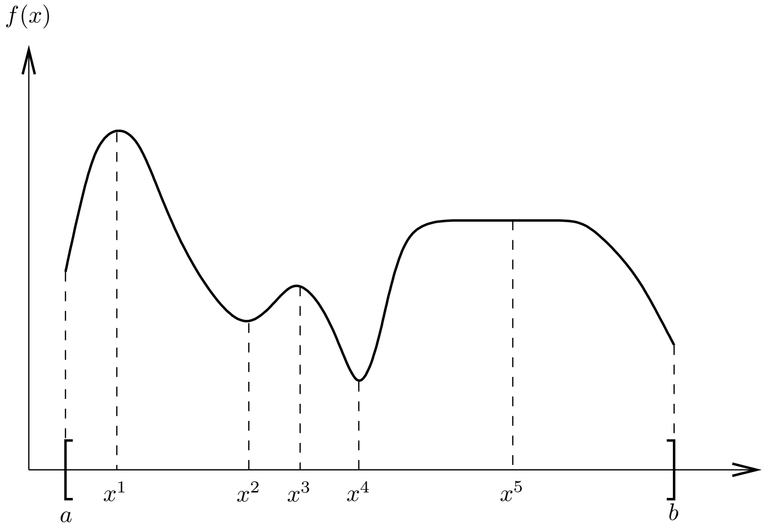


Figure 1.2. The various types of minima and maxima.

1.2.3 Existence of Minima and Maxima

A crucial question when it comes to optimizing a function on a given set, is whether a minimizer or a maximizer exist for that function in that set. Strictly, a minimum or maximum should only be referred to when it is known to exist.

Fig 1.3. illustrates three instances where a minimum does not exist. In Fig 1.3.(a), the infimum of f over $S := (a, b)$ is given by $f(b)$, but since S is not closed and, in particular, $b \notin S$, a minimum does not exist. In Fig 1.3.(b), the infimum of f over $S := [a, b]$ is given by the limit of $f(x)$ as x approaches c from the left, i.e., $\inf\{f(x) : x \in S\} = \lim_{x \to c^-} f(x)$. However, because f is discontinuous at c, a minimizing solution does not exist. Finally, Fig 1.3.(c) illustrates a situation within which f is unbounded over the unbounded set $S := \{x \in \mathbb{R} : x \geq a\}$.

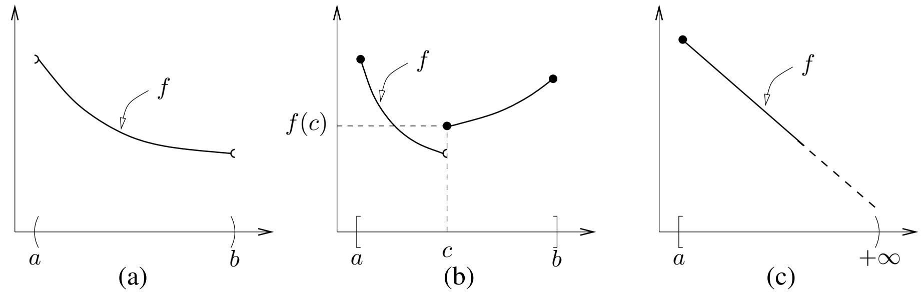


Figure 1.3. The nonexistence of a minimizing solution.

We now formally state and prove the result that if S is nonempty, closed, and bounded, and if f is continuous on S, then, unlike the various situations of Fig. 1.3., a minimum exists.

Theorem 1.14 (Weierstrass' Theorem). *Let S be a nonempty, compact set, and let $f : S \to \mathbb{R}$ be continuous on S. Then, the problem $\min\{f(\mathbf{x}) : \mathbf{x} \in S\}$ attains its minimum, that is, there exists a minimizing solution to this problem.*

Proof. Since f is continuous on S and S is both closed and bounded, f is bounded below on S. Consequently, since $S \neq \emptyset$, there exists a greatest lower bound $\alpha := \inf\{f(\mathbf{x} : \mathbf{x} \in S\}$ (see Axiom 1.3). Now, let $0 < \varepsilon < 1$, and consider the set $S_k := \{\mathbf{x} \in S : \alpha \leq f(\mathbf{x}) \leq \alpha + \varepsilon^k\}$ for $k = 1, 2, \ldots$. By the definition of an infimum, $S_k \neq \emptyset$ for each k, and so we may construct a sequence of points $\{\mathbf{x}_k\} \subset S$ by selecting a point $\mathbf{x}_k$ for each $k = 1, 2, \ldots$. Since S is bounded, there exists a convergent subsequence $\{\mathbf{x}_k\}_{\mathcal{K}} \subset S$ indexed by the set $\mathcal{K} \subset \mathbb{N}$; let $\bar{\mathbf{x}}$ denote its limit. By the closedness of S, we have $\bar{\mathbf{x}} \in S$; and by the continuity of f on S, since $\alpha \leq f(\mathbf{x}_k) \leq \alpha + \varepsilon^k$, we have $\alpha = \lim_{k\to\infty, k\in\mathcal{K}} f(\mathbf{x}_k) = f(\bar{\mathbf{x}})$. Hence, we have shown that there exist a solution $\bar{\mathbf{x}} \in S$ such that $f(\bar{\mathbf{x}}) = \alpha = \inf\{f(\mathbf{x} : \mathbf{x} \in S\}$, i.e., $\bar{\mathbf{x}}$ is a minimizing solution. □

The hypotheses of Theorem 1.14 can be justified as follows: (i) the feasible set must be **nonempty**, otherwise there are no feasible points at which to attain the minimum; (ii) the feasible set must contain its boundary points, which is ensured by assuming that the feasible set is **closed**; (iii) the objective function must be **continuous** on the feasible set, otherwise the limit at a point may not exist or be different from the value of the function at that point; and (iv) the feasible set must be **bounded** because otherwise even a continuous function can be unbounded on the feasible set.

Example 1.15. Theorem 1.14 establishes that a minimum (and a maximum) of

$$\min_{x\in[-1,1]} x^2$$

exists, since $[-1, 1]$ is a nonempty, compact set and $x \mapsto x^2$ is a continuous function on $[-1, 1]$. On the other hand, minima can still exist even though the set is not compact or the function is not continuous, for Theorem 1.14 only provides a sufficient condition. This is the case for the problem

$$\min_{x\in(-1,1)} x^2,$$

which has a minimum at $x = 0$. (See also Example 1.13.)

Example 1.16. Consider the NLP problem of Example 1.1 (p. 2),

$$\begin{array}{rrcl} \displaystyle\min_{\mathbf{x}} & (x_1-3)^2 + (x_2-2)^2 & & \\ \text{s.t.} & x_1^2 - x_2 - 3 & \leq & 0 \\ & x_2 - 1 & \leq & 0 \\ & -x_1 & \leq & 0. \end{array}$$

The objective function being continuous and the feasible region being nonempty, closed and bounded, the existence of a minimum to this problem directly follows from Theorem 1.14.

1.3 CONVEX PROGRAMMING

A particular class of nonlinear programs is that of convex programs (see Appendix A.3 for a general overview on convex sets and convex functions):

Definition 1.17 (Convex Program). *Let C be a nonempty convex set in $\mathrm{I\!R}^n$, and let $f : C \to \mathrm{I\!R}$ be convex on C. Then,*

$$\min_{\mathbf{x}\in C} f(\mathbf{x})$$

is said to be a convex program *(or a* convex optimization problem*).*

Convex programs possess nicer theoretical properties than general nonconvex problems. The following theorem is a fundamental result in convex programming:

Theorem 1.18. *Let $\mathbf{x}^\star$ be a local minimum of a convex program. Then, $\mathbf{x}^\star$ is also a global minimum.*

Proof. $\mathbf{x}^\star$ being a local minimum,

$$\exists \varepsilon > 0 \text{ such that } f(\mathbf{x}) \geq f(\mathbf{x}^\star), \forall \mathbf{x} \in \mathcal{B}_\varepsilon\left(\mathbf{x}^\star\right).$$

By contradiction, suppose that $\mathbf{x}^\star$ is not a global minimum. Then,

$$\exists \bar{\mathbf{x}} \in C \text{ such that } f(\bar{\mathbf{x}}) < f(\mathbf{x}^\star). \tag{1.4}$$

Let $\lambda \in (0,1)$ be chosen such that $\mathbf{y} := \lambda\bar{\mathbf{x}} + (1-\lambda)\mathbf{x}^\star \in \mathcal{B}_\varepsilon\left(\mathbf{x}^\star\right)$. By convexity of C, $\mathbf{y}$ is in C. Next, by convexity of f on C and (1.4),

$$f(\mathbf{y}) \leq \lambda f(\bar{\mathbf{x}}) + (1-\lambda) f(\mathbf{x}^\star) < \lambda f(\mathbf{x}^\star) + (1-\lambda) f(\mathbf{x}^\star) = f(\mathbf{x}^\star),$$

hence contradicting the assumption that $\mathbf{x}^\star$ is a local minimum. □

Example 1.19. Consider once again the NLP problem of Example 1.1 (p. 2),

$$\begin{array}{rrcl} \min\limits_{\mathbf{x}} & (x_1-3)^2 + (x_2-2)^2 & & \\ \text{s.t.} & x_1^2 - x_2 - 3 & \leq & 0 \\ & x_2 - 1 & \leq & 0 \\ & -x_1 & \leq & 0. \end{array}$$

The objective function f and the inequality constraints g_1, g_2 and g_3 being convex, every local solution to this problem is also a global solution by Theorem 1.18; henceforth, $(1,2)$ is a global solution and the global solution value is 4.

In convex programming, any local minimum is therefore a local optimum. This is a powerful result that makes any local optimization algorithm a global optimization algorithm when applied to a convex optimization problem. Yet, Theorem 1.18 only gives a sufficient condition for that property to hold. That is, a nonlinear program with nonconvex participating functions may not necessarily have local minima that are not global minima.

1.4 UNCONSTRAINED PROBLEMS

An unconstrained problem is a problem of the form to minimize (or maximize) $f(\mathbf{x})$ without any constraints on the variables $\mathbf{x}$:

$$\min\{f(\mathbf{x}) : \mathbf{x} \in \mathrm{IR}^{n_x}\}.$$

Note that the feasible domain of $\mathbf{x}$ being unbounded, Weierstrass' Theorem 1.14 does not apply, and one does not know with certainty, whether a minimum actually exists for that problem.[3] Moreover, even if the objective function is convex, one such minimum may not exist (think of $f : x \mapsto \exp x$!). Hence, we shall proceed with the theoretically unattractive task of seeking minima and maxima of functions which need not have them!

Given a point $\mathbf{x}$ in IR^{n_x}, necessary conditions help determine whether or not a point is a local or a global minimum of a function f. For this purpose, we are mostly interested in obtaining conditions that can be checked algebraically.

Definition 1.20 (Descent Direction). *Suppose that $f : \mathrm{IR}^{n_x} \to \mathrm{IR}$ is continuous at $\bar{\mathbf{x}}$. A vector $\mathbf{d} \in \mathrm{IR}^{n_x}$ is said to be a* descent direction*, or an* improving direction*, for f at $\bar{\mathbf{x}}$ if*

$$\exists \delta > 0 : f(\bar{\mathbf{x}} + \lambda \mathbf{d}) < f(\bar{\mathbf{x}}) \quad \forall \lambda \in (0, \delta).$$

Moreover, the cone of descent directions *at $\bar{\mathbf{x}}$, denoted by $\mathcal{F}(\bar{\mathbf{x}})$, is given by*

$$\mathcal{F}(\bar{\mathbf{x}}) := \{\mathbf{d} : \exists \delta > 0 \textit{ such that } f(\bar{\mathbf{x}} + \lambda \mathbf{d}) < f(\bar{\mathbf{x}}) \quad \forall \lambda \in (0, \delta)\}.$$

The foregoing definition provides a **geometrical** characterization for a descent direction. yet, an **algebraic** characterization for a descent direction would be more useful from a practical point of view. In response to this, let us assume that f is differentiable and define the following set at $\bar{\mathbf{x}}$:

$$\mathcal{F}_0(\bar{\mathbf{x}}) := \{\mathbf{d} : \boldsymbol{\nabla} f(\bar{\mathbf{x}})^{\mathsf{T}} \mathbf{d} < 0\}.$$

This is illustrated in Fig. 1.4., where the half-space $\mathcal{F}_0(\bar{\mathbf{x}})$ and the gradient $\boldsymbol{\nabla} f(\bar{\mathbf{x}})$ are translated from the origin to $\bar{\mathbf{x}}$ for convenience.

The following lemma proves that every element $\mathbf{d} \in \mathcal{F}_0(\bar{\mathbf{x}})$ is a descent direction at $\bar{\mathbf{x}}$.

Lemma 1.21 (Algebraic Characterization of a Descent Direction). *Suppose that $f : \mathrm{IR}^{n_x} \to \mathrm{IR}$ is differentiable at $\bar{\mathbf{x}}$. If there exists a vector $\mathbf{d}$ such that $\boldsymbol{\nabla} f(\bar{\mathbf{x}})^{\mathsf{T}} \mathbf{d} < 0$, then $\mathbf{d}$ is a descent direction for f at $\bar{\mathbf{x}}$. That is,*

$$\mathcal{F}_0(\bar{\mathbf{x}}) \subseteq \mathcal{F}(\bar{\mathbf{x}}).$$

Proof. f being differentiable at $\bar{\mathbf{x}}$,

$$f(\bar{\mathbf{x}} + \lambda \mathbf{d}) = f(\bar{\mathbf{x}}) + \lambda \boldsymbol{\nabla} f(\bar{\mathbf{x}})^{\mathsf{T}} \mathbf{d} + \lambda \|\mathbf{d}\| \alpha(\lambda \mathbf{d})$$

where $\lim_{\lambda \to 0} \alpha(\lambda \mathbf{d}) = 0$. Rearranging the terms and dividing by $\lambda \neq 0$, we get

$$\frac{f(\bar{\mathbf{x}} + \lambda \mathbf{d}) - f(\bar{\mathbf{x}})}{\lambda} = \boldsymbol{\nabla} f(\bar{\mathbf{x}})^{\mathsf{T}} \mathbf{d} + \|\mathbf{d}\| \alpha(\lambda \mathbf{d}).$$

[3] For unconstrained optimization problems, the existence of a minimum can actually be guaranteed if the objective objective function is such that $\lim_{\|\mathbf{x}\| \to +\infty} f(\mathbf{x}) = +\infty$ (O-coercive function).

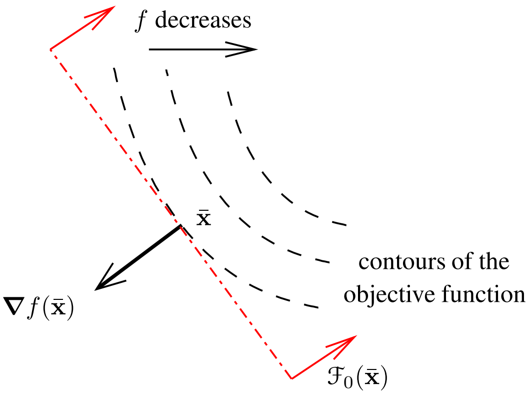

f decreases
$\bar{\mathbf{x}}$
$\boldsymbol{\nabla} f(\bar{\mathbf{x}})$
contours of the
objective function
$\mathcal{F}_0(\bar{\mathbf{x}})$

Figure 1.4. Illustration of the set $\mathcal{F}_0(\bar{\mathbf{x}})$.

Since $\boldsymbol{\nabla} f(\bar{\mathbf{x}})^{\mathsf{T}}\mathbf{d} < 0$ and $\lim_{\lambda\to 0} \alpha(\lambda\mathbf{d}) = 0$, there exists a $\delta > 0$ such that $\boldsymbol{\nabla} f(\bar{\mathbf{x}})^{\mathsf{T}}\mathbf{d} + \|\mathbf{d}\|\alpha(\lambda\mathbf{d}) < 0$ for all $\lambda \in (0, \delta)$. □

We are now ready to derive a number of necessary conditions for a point to be a local minimum of an unconstrained optimization problem.

Theorem 1.22 (First-Order Necessary Condition for a Local Minimum). *Suppose that* $f : \mathbb{R}^{n_x} \to \mathbb{R}$ *is differentiable at* $\mathbf{x}^\star$. *If* $\mathbf{x}^\star$ *is a local minimum, then* $\boldsymbol{\nabla} f(\mathbf{x}^\star) = \mathbf{0}$.

Proof. The proof proceeds by contraposition. Suppose that $\boldsymbol{\nabla} f(\mathbf{x}^\star) \neq \mathbf{0}$. Then, letting $\mathbf{d} = -\boldsymbol{\nabla} f(\mathbf{x}^\star)$, we get $\boldsymbol{\nabla} f(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{d} = -\|\boldsymbol{\nabla} f(\mathbf{x}^\star)\|^2 < 0$. By Lemma 1.21,

$$\exists \delta > 0 : f(\mathbf{x}^\star + \lambda\mathbf{d}) < f(\mathbf{x}^\star) \quad \forall \lambda \in (0, \delta),$$

hence contradicting the assumption that $\mathbf{x}^\star$ is a local minimum for f. □

Remark 1.23 (Obtaining Candidate Solution Points). The above condition is called a *first-order necessary condition* because it uses the first-order derivatives of f. This condition indicates that the candidate solutions to an unconstrained optimization problem can be found by solving a system of n_x algebraic (nonlinear) equations. Points $\bar{\mathbf{x}}$ such that $\boldsymbol{\nabla} f(\bar{\mathbf{x}}) = \mathbf{0}$ are known as *stationary points*. Yet, a stationary point need *not* be a local minimum as illustrated by the following example; it could very well be a local maximum, or even a *saddle point*.

Example 1.24. Consider the problem

$$\min_{x\in\mathbb{R}} \quad x^2 - x^4.$$

The gradient vector of the objective function is given by

$$\boldsymbol{\nabla} f(x) = 2x - 4x^3,$$

which has three distinct roots $x_1^\star = 0$, $x_2^\star = \frac{1}{\sqrt{2}}$ and $x_2^\star = -\frac{1}{\sqrt{2}}$. Out of these values, $x_1^\star$ gives the smallest cost value, $f(x_1^\star) = 0$. Yet, we cannot declare $x_1^\star$ to be the global minimum, because we do not know whether a (global) minimum exists for this problem. Indeed, as shown in Fig. 1.5., none of the stationary points is a global minimum, because f decreases to $-\infty$ as $|x| \to \infty$.

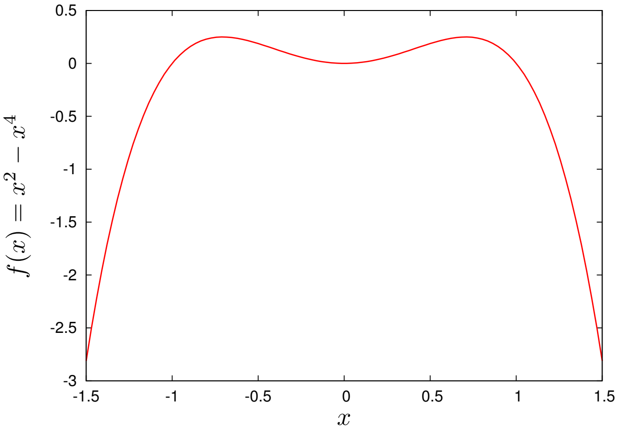


Figure 1.5. Illustration of the objective function in Example 1.24.

More restrictive necessary conditions can also be derived in terms of the Hessian matrix H whose elements are the second-order derivatives of f. One such second-order condition is given below.

Theorem 1.25 (Second-Order Necessary Conditions for a Local Minimum). *Suppose that $f : \mathbb{R}^{n_x} \to \mathbb{R}$ is twice differentiable at $\mathbf{x}^\star$. If $\mathbf{x}^\star$ is a local minimum, then $\boldsymbol{\nabla} f(\mathbf{x}^\star) = \mathbf{0}$ and $\mathsf{H}(\mathbf{x}^\star)$ is positive semidefinite.*

Proof. Consider an arbitrary direction $\mathbf{d}$. Then, from the differentiability of f at $\mathbf{x}^\star$, we have

$$f(\mathbf{x}^\star + \lambda \mathbf{d}) = f(\mathbf{x}^\star) + \lambda \boldsymbol{\nabla} f(\mathbf{x}^\star)^{\mathsf{T}} \mathbf{d} + \frac{\lambda^2}{2} \mathbf{d}^{\mathsf{T}} \mathsf{H}(\mathbf{x}^\star) \mathbf{d} + \lambda^2 \|\mathbf{d}\|^2 \alpha(\lambda \mathbf{d}), \tag{1.5}$$

where $\lim_{\lambda \to 0} \alpha(\lambda \mathbf{d}) = 0$. Since $\mathbf{x}^\star$ is a local minimum, from Theorem 1.22, $\boldsymbol{\nabla} f(\mathbf{x}^\star) = 0$. Rearranging the terms in (1.5) and dividing by λ^2, we get

$$\frac{f(\mathbf{x}^\star + \lambda \mathbf{d}) - f(\mathbf{x}^\star)}{\lambda^2} = \frac{1}{2} \mathbf{d}^{\mathsf{T}} \mathsf{H}(\mathbf{x}^\star) \mathbf{d} + \|\mathbf{d}\|^2 \alpha(\lambda \mathbf{d}).$$

Since $\mathbf{x}^\star$ is a local minimum, $f(\mathbf{x}^\star + \lambda \mathbf{d}) \geq f(\mathbf{x}^\star)$ for λ sufficiently small. By taking the limit as $\lambda \to 0$, it follows that $\mathbf{d}^{\mathsf{T}} \mathsf{H}(\mathbf{x}^\star) \mathbf{d} \geq 0$. Since $\mathbf{d}$ is arbitrary, $\mathsf{H}(\mathbf{x}^\star)$ is therefore positive semidefinite. □

Example 1.26. Consider the problem

$$\min_{\mathbf{x} \in \mathbb{R}^2} \quad x_1 x_2.$$

The gradient vector of the objective function is given by

$$\nabla f(\mathbf{x}) = \left[\begin{array}{cc} x_2 & x_1 \end{array}\right]^\mathsf{T}$$

so that the only stationary point in $\mathrm{I\!R}^2$ is $\bar{\mathbf{x}} = (0,0)$. Now, consider the Hessian matrix of the objective function at $\bar{\mathbf{x}}$:

$$\mathsf{H}(\bar{\mathbf{x}}) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \forall \mathbf{x} \in \mathrm{I\!R}^2.$$

It is easily checked that $\mathsf{H}(\bar{\mathbf{x}})$ is indefinite, therefore, by Theorem 1.25, the stationary point $\bar{\mathbf{x}}$ is not a (local) minimum (nor is it a local maximum). Such stationary points are called *saddle points* (see Fig. 1.6. below).

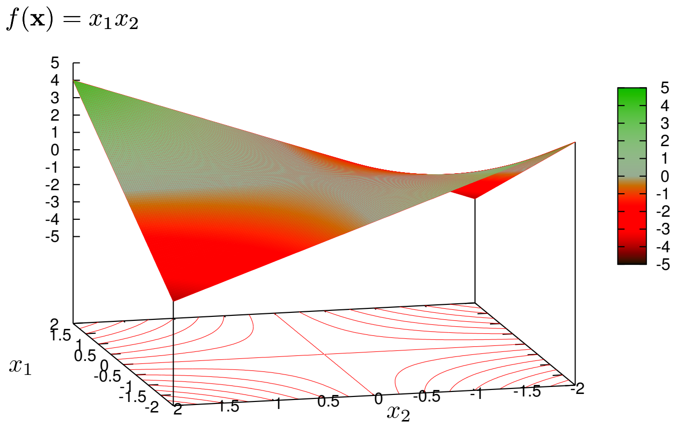


Figure 1.6. Illustration of the objective function in Example 1.26.

The conditions presented in Theorems 1.22 and 1.25 are necessary conditions. That is, they must hold true at every local optimal solution. Yet, a point satisfying these conditions need not be a local minimum. The following theorem gives sufficient conditions for a stationary point to be a *global* minimum point, provided the objective function is convex on $\mathrm{I\!R}^{n_x}$.

Theorem 1.27 (First-Order Sufficient Conditions for a Strict Local Minimum). *Suppose that $f : \mathrm{I\!R}^{n_x} \to \mathrm{I\!R}$ is differentiable at $\mathbf{x}^\star$ and convex on $\mathrm{I\!R}^{n_x}$. If $\nabla f(\mathbf{x}^\star) = \mathbf{0}$, then $\mathbf{x}^\star$ is a global minimum of f on $\mathrm{I\!R}^{n_x}$.*

Proof. f being convex on $\mathrm{I\!R}^{n_x}$ and differentiable at $\mathbf{x}^\star$, by Theorem A.17, we have

$$f(\mathbf{x}) \geq f(\mathbf{x}^\star) + \nabla f(\mathbf{x}^\star)^\mathsf{T}[\mathbf{x} - \mathbf{x}^\star] \quad \forall \mathbf{x} \in \mathrm{I\!R}^{n_x}.$$

But since $\mathbf{x}^\star$ is a stationary point,

$$f(\mathbf{x}) \geq f(\mathbf{x}^\star) \quad \forall \mathbf{x} \in \mathrm{I\!R}^{n_x}.$$

□

The convexity condition required by the foregoing theorem is actually very restrictive, in the sense that many practical problems are nonconvex. In the following theorem, we give sufficient conditions for characterizing a local minimum point, provided the objective function is strictly convex in a neighborhood of that point.

Theorem 1.28 (Second-Order Sufficient Conditions for a Strict Local Minimum). *Suppose that $f : \mathbb{R}^{n_x} \to \mathbb{R}$ is twice differentiable at $\mathbf{x}^\star$. If $\nabla f(\mathbf{x}^\star) = \mathbf{0}$ and $\mathsf{H}(\mathbf{x}^\star)$ is positive definite, then $\mathbf{x}^\star$ is a local minimum of f.*

Proof. f being twice differentiable at $\mathbf{x}^\star$, we have

$$f(\mathbf{x}^\star + \mathbf{d}) \ = \ f(\mathbf{x}^\star) + \nabla f(\mathbf{x}^\star)^\mathsf{T}\mathbf{d} + \frac{1}{2}\mathbf{d}^\mathsf{T}\mathsf{H}(\mathbf{x}^\star)\mathbf{d} + \|\mathbf{d}\|^2\alpha(\mathbf{d}),$$

for each $\mathbf{d} \in \mathbb{R}^{n_x}$, where $\lim_{\mathbf{d}\to\mathbf{0}} \alpha(\mathbf{d}) = 0$. Let λ^L denote the smallest eigenvalue of $\mathsf{H}(\mathbf{x}^\star)$. Then, $\mathsf{H}(\mathbf{x}^\star)$ being positive definite we have $\lambda^L > 0$, and $\mathbf{d}^\mathsf{T}\mathsf{H}(\mathbf{x}^\star)\mathbf{d} \ \geq \ \lambda^L\|\mathbf{d}\|^2$. Moreover, from $\nabla f(\mathbf{x}^\star) = \mathbf{0}$, we get

$$f(\mathbf{x}^\star + \mathbf{d}) - f(\mathbf{x}^\star) \ \geq \ \left[\frac{\lambda}{2} + \alpha(\mathbf{d})\right]\|\mathbf{d}\|^2.$$

Since $\lim_{\mathbf{d}\to\mathbf{0}} \alpha(\mathbf{d}) = 0$,

$$\exists\eta > 0 \text{ such that } |\alpha(\mathbf{d})| < \frac{\lambda}{4} \quad \forall\mathbf{d} \in \mathcal{B}_\eta\left(\mathbf{0}\right),$$

and finally,

$$f(\mathbf{x}^\star + \mathbf{d}) - f(\mathbf{x}^\star) \ \geq \ \frac{\lambda}{4}\|\mathbf{d}\|^2 \ > \ 0 \quad \forall\mathbf{d} \in \mathcal{B}_\eta\left(\mathbf{0}\right) \setminus \{\mathbf{0}\},$$

i.e., $\mathbf{x}^\star$ is a strict local minimum of f. □

Example 1.29. Consider the problem

$$\min_{\mathbf{x}\in\mathbb{R}^2} \quad (x_1 - 1)^2 - x_1x_2 + (x_2 - 1)^2.$$

The gradient vector and Hessian matrix at $\bar{\mathbf{x}} = (2, 2)$ are given by

$$\begin{aligned}\nabla f(\bar{\mathbf{x}}) \ &= \begin{bmatrix} 2(\bar{x}_1 - 1) - \bar{x}_2 & 2(\bar{x}_2 - 1) - \bar{x}_1 \end{bmatrix}^\mathsf{T} \ = \ \mathbf{0} \\ \mathsf{H}(\bar{\mathbf{x}}) \ &= \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \succ \mathbf{0}\end{aligned}$$

Hence, by Theorem 1.25, $\bar{\mathbf{x}}$ is a local minimum of f. ($\bar{\mathbf{x}}$ is also a global minimum of f on $\mathbb{R}^2$ since f is convex.) The objective function is pictured in Fig. 1.7. below.

We close this subsection by reemphasizing the fact that every local minimum of an unconstrained optimization problem $\min\{f(\mathbf{x} : \mathbf{x} \in \mathbb{R}^{n_x}\}$ is a global minimum if f is a convex function on $\mathbb{R}^{n_x}$ (see Theorem 1.18). Yet, convexity of f is *not* a necessary condition for each local minimum to be a global minimum. As just an example, consider the function $x \mapsto \exp\left(-\frac{1}{x^2}\right)$ (see Fig 1.8.). In fact, such functions are said to be *pseudoconvex*.

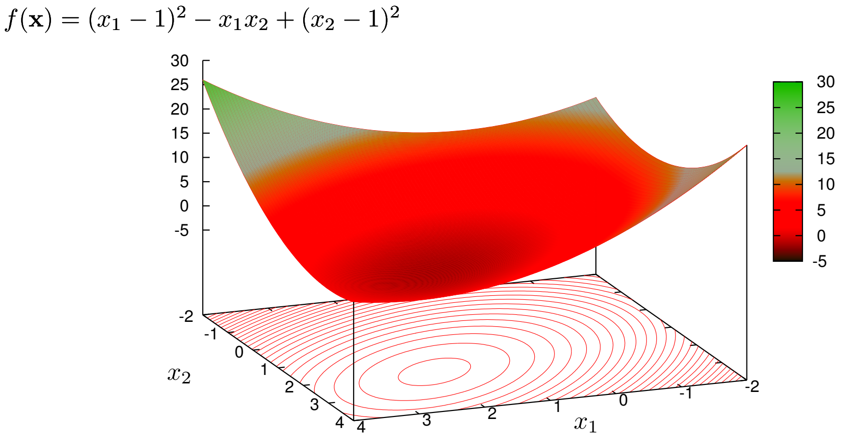


Figure 1.7. Illustration of the objective function in Example 1.29.

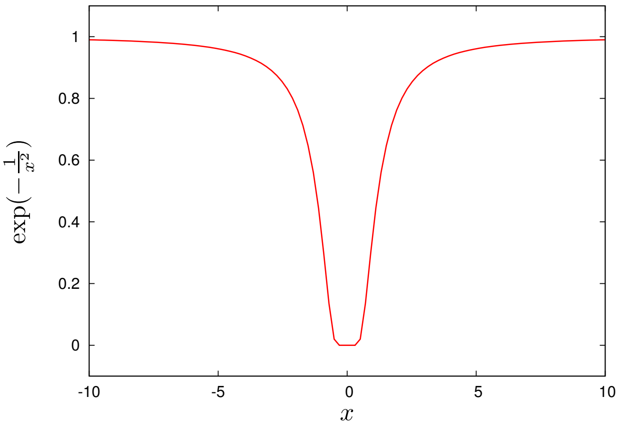


Figure 1.8. Plot of the pseudoconvex function $x \mapsto \exp\left(-\frac{1}{x^2}\right)$.

1.5 PROBLEMS WITH INEQUALITY CONSTRAINTS

In practice, few problems can be formulated as unconstrained programs. This is because the feasible region is generally restricted by imposing constraints on the optimization variables.

In this section, we first present theoretical results for the problem to:

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in S, \end{array}$$

for a general set S (geometric optimality conditions). Then, we let S be more specifically defined as the feasible region of a NLP of the form to minimize $f(\mathbf{x})$, subject to $\mathbf{g}(\mathbf{x}) \leq 0$ and $\mathbf{x} \in X$, and derive the Karush-Kuhn-Tucker (KKT) conditions of optimality.

1.5.1 Geometric Optimality Conditions

Definition 1.30 (Feasible Direction). *Let S be a nonempty set in $\mathbb{R}^{n_x}$. A vector $\mathbf{d} \in \mathbb{R}^{n_x}$, $\mathbf{d} \neq \mathbf{0}$, is said to be a* feasible direction *at $\bar{\mathbf{x}} \in \mathrm{cl}\,(S)$ if*

$$\exists \delta > 0 \text{ such that } \bar{\mathbf{x}} + \eta \mathbf{d} \in S \quad \forall \eta \in (0, \delta).$$

Moreover, the cone of feasible directions *at $\bar{\mathbf{x}}$, denoted by $\mathcal{D}(\bar{\mathbf{x}})$, is given by*

$$\mathcal{D}(\bar{\mathbf{x}}) := \{\mathbf{d} \neq \mathbf{0} : \exists \delta > 0 \text{ such that } \bar{\mathbf{x}} + \eta \mathbf{d} \in S \quad \forall \eta \in (0, \delta)\}.$$

From the above definition and Lemma 1.21, it is clear that a small movement from $\bar{\mathbf{x}}$ along a direction $\mathbf{d} \in \mathcal{D}(\bar{\mathbf{x}})$ leads to feasible points, whereas a similar movement along a direction $\mathbf{d} \in \mathcal{F}_0(\bar{\mathbf{x}})$ leads to solutions of improving objective value (see Definition 1.20). As shown in Theorem 1.31 below, a (geometric) necessary condition for local optimality is that: *"Every improving direction is not a feasible direction."* This fact is illustrated in Fig. 1.9., where both the half-space $\mathcal{F}_0(\bar{\mathbf{x}})$ and the cone $\mathcal{D}(\bar{\mathbf{x}})$ (see Definition A.10) are translated from the origin to $\bar{\mathbf{x}}$ for clarity.

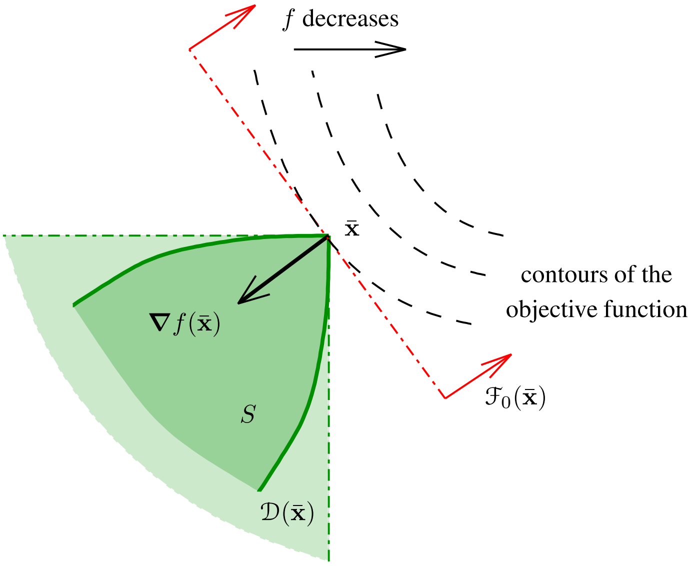

f decreases
$\bar{\mathbf{x}}$
contours of the objective function
$\nabla f(\bar{\mathbf{x}})$
S
$\mathcal{F}_0(\bar{\mathbf{x}})$
$\mathcal{D}(\bar{\mathbf{x}})$

Figure 1.9. Illustration of the (geometric) necessary condition $\mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}(\bar{\mathbf{x}}) = \emptyset$.

Theorem 1.31 (Geometric Necessary Condition for a Local Minimum). *Let S be a nonempty set in $\mathbb{R}^{n_x}$, and let $f : \mathbb{R}^{n_x} \to \mathbb{R}$ be a differentiable function. Suppose that $\bar{\mathbf{x}}$ is a local minimizer of the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in S$. Then, $\mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}(\bar{\mathbf{x}}) = \emptyset$.*

Proof. By contradiction, suppose that there exists a vector $\mathbf{d} \in \mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}(\bar{\mathbf{x}})$, $\mathbf{d} \neq \mathbf{0}$. Then, by Lemma 1.21,

$$\exists \delta_1 > 0 \text{ such that } f(\bar{\mathbf{x}} + \eta \mathbf{d}) < f(\bar{\mathbf{x}}) \quad \forall \eta \in (0, \delta_1).$$

Moreover, by Definition 1.30,

$$\exists \delta_2 > 0 \text{ such that } \bar{\mathbf{x}} + \eta \mathbf{d} \in S \quad \forall \eta \in (0, \delta_2).$$

Hence,

$$\exists \mathbf{x} \in \mathcal{B}_\eta\left(\bar{\mathbf{x}}\right) \cap S \text{ such that } f(\bar{\mathbf{x}} + \eta \mathbf{d}) < f(\bar{\mathbf{x}}),$$

for every $\eta \in (0, \min\{\delta_1, \delta_2\})$, which contradicts the assumption that $\bar{\mathbf{x}}$ is a local minimum of f on S (see Definition 1.9). □

1.5.2 KKT Conditions

We now specify the feasible region as

$$S := \{\mathbf{x} : g_i(\mathbf{x}) \leq 0 \quad \forall i = 1, \ldots, n_g\},$$

where $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_g$, are continuous functions. In the geometric optimality condition given by Theorem 1.31, $\mathcal{D}(\bar{\mathbf{x}})$ is the cone of feasible directions. From a practical viewpoint, it is desirable to convert this geometric condition into a more usable condition involving algebraic equations. As Lemma 1.33 below indicates, we can define a cone $\mathcal{D}_0(\bar{\mathbf{x}})$ in terms of the gradients of the *active constraints* at $\bar{\mathbf{x}}$, such that $\mathcal{D}_0(\bar{\mathbf{x}}) \subseteq \mathcal{D}(\bar{\mathbf{x}})$. For this, we need the following:

Definition 1.32 (Active Constraint, Active Set). *Let $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_g$, and consider the set $S := \{\mathbf{x} : g_i(\mathbf{x}) \leq 0, i = 1, \ldots, n_g\}$. Let $\bar{\mathbf{x}} \in S$ be a feasible point. For each $i = 1, \ldots, n_g$, the constraint g_i is said to* active *or* binding *at $\bar{\mathbf{x}}$ if $g_i(\bar{\mathbf{x}}) = 0$; it is said to be* inactive *at $\bar{\mathbf{x}}$ if $g_i(\bar{\mathbf{x}}) < 0$. Moreover,*

$$\mathcal{A}(\bar{\mathbf{x}}) \ := \ \{i : g_i(\bar{\mathbf{x}}) = 0\},$$

denotes the set of active constraints *at $\bar{\mathbf{x}}$.*

Lemma 1.33 (Algebraic Characterization of a Feasible Direction). *Let $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_g$ be differentiable functions, and consider the set $S := \{\mathbf{x} : g_i(\mathbf{x}) \leq 0, i = 1, \ldots, n_g\}$. For any feasible point $\bar{\mathbf{x}} \in S$, we have*

$$\mathcal{D}_0(\bar{\mathbf{x}}) \ := \ \{\mathbf{d} : \nabla g_i(\bar{\mathbf{x}})^{\mathsf{T}} \mathbf{d} < 0 \quad \forall i \in \mathcal{A}(\bar{\mathbf{x}})\} \ \subseteq \ \mathcal{D}(\bar{\mathbf{x}}).$$

Proof. Suppose $\mathcal{D}_0(\bar{\mathbf{x}})$ is nonempty, and let $\mathbf{d} \in \mathcal{D}_0(\bar{\mathbf{x}})$. Since $\nabla g_i(\bar{\mathbf{x}}) \mathbf{d} < 0$ for each $i \in \mathcal{A}(\bar{\mathbf{x}})$, then by Lemma 1.21, $\mathbf{d}$ is a descent direction for g_i at $\bar{\mathbf{x}}$, i.e.,

$$\exists \delta_2 > 0 \text{ such that } g_i(\bar{\mathbf{x}} + \eta \mathbf{d}) < g_i(\bar{\mathbf{x}}) = 0 \quad \forall \eta \in (0, \delta_2), \ \forall i \in \mathcal{A}(\bar{\mathbf{x}}).$$

Furthermore, since $g_i(\bar{\mathbf{x}}) < 0$ and g_i is continuous at $\bar{\mathbf{x}}$ (for it is differentiable) for each $i \notin \mathcal{A}(\bar{\mathbf{x}})$,

$$\exists \delta_1 > 0 \text{ such that } g_i(\bar{\mathbf{x}} + \eta \mathbf{d}) < 0 \quad \forall \eta \in (0, \delta_1), \ \forall i \notin \mathcal{A}(\bar{\mathbf{x}}).$$

Furthermore, Overall, it is clear that the points $\bar{\mathbf{x}} + \eta \mathbf{d}$ are in S for all $\eta \in (0, \min\{\delta_1, \delta_2\})$. Hence, by Definition 1.30, $\mathbf{d} \in \mathcal{D}(\bar{\mathbf{x}})$. □

Remark 1.34. This lemma together with Theorem 1.31 directly leads to the result that $\mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}_0(\bar{\mathbf{x}}) = \emptyset$ for any local solution point $\bar{\mathbf{x}}$, i.e.,

$$\arg\min\{f(\mathbf{x}) : \mathbf{x} \in S\} \subset \{\mathbf{x} \in \mathbb{R}^{n_x} : \mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}_0(\bar{\mathbf{x}}) = \emptyset\}.$$

The foregoing geometric characterization of local solution points applies equally well to either interior points $\operatorname{int}(S) := \{\mathbf{x} \in \mathbb{R}^{n_x} : g_i(\mathbf{x}) < 0, \forall i = 1, \ldots, n_g\}$, or boundary points being at the boundary of the feasible domain. At an interior point, in particular, any direction is feasible, and the necessary condition $\mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}_0(\bar{\mathbf{x}}) = \emptyset$ reduces to $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$, which gives the same condition as in unconstrained optimization (see Theorem 1.22).

Note also that there are several cases where the condition $\mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}_0(\bar{\mathbf{x}}) = \emptyset$ is satisfied by non-optimal points. In other words, this condition is necessary but *not sufficient* for a point $\bar{\mathbf{x}}$ to be a local minimum of f on S. For instance, any point $\bar{\mathbf{x}}$ with $\nabla g_i(\bar{\mathbf{x}}) = 0$ for some $i \in \mathcal{A}(\bar{\mathbf{x}})$ trivially satisfies the condition $\mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}_0(\bar{\mathbf{x}}) = \emptyset$. Another example is given below.

Example 1.35. Consider the problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \; & f(\mathbf{x}) := x_1^2 + x_2^2 \qquad (1.6) \\ \text{s.t.} \; & g_1(\mathbf{x}) := x_1 \leq 0 \\ & g_2(\mathbf{x}) := -x_1 \leq 0. \end{aligned}$$

Clearly, this problem is convex and $\mathbf{x}^\star = (0,0)^\mathsf{T}$ is the unique global minimum.

Now, let $\bar{\mathbf{x}}$ be any point on the line $\mathcal{C} := \{\mathbf{x} : x_1 = 0\}$. Both constraints g_1 and g_2 are active at $\bar{\mathbf{x}}$, and we have $\nabla g_1(\bar{\mathbf{x}}) = -\nabla g_2(\bar{\mathbf{x}}) = (1,0)^\mathsf{T}$. Therefore, no direction $\mathbf{d} \neq \mathbf{0}$ can be found such that $\nabla g_1(\bar{\mathbf{x}})^\mathsf{T}\mathbf{d} < 0$ and $\nabla g_2(\bar{\mathbf{x}})^\mathsf{T}\mathbf{d} < 0$ simultaneously, i.e., $\mathcal{D}_0(\bar{\mathbf{x}}) = \emptyset$. In turn, this implies that $\mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}_0(\bar{\mathbf{x}}) = \emptyset$ is trivially satisfied for any point on $\mathcal{C}$.

On the other hand, observe that the condition $\mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}(\bar{\mathbf{x}}) = \emptyset$ in Theorem 1.31 excludes all the points on $\mathcal{C}$, but the origin, since a feasible direction at $\bar{\mathbf{x}}$ is given, e.g., by $\mathbf{d} = (0,1)^\mathsf{T}$.

Next, we reduce the geometric necessary optimality condition $\mathcal{F}_0(\bar{\mathbf{x}}) \cap \mathcal{D}_0(\bar{\mathbf{x}}) = \emptyset$ to a statement in terms of the gradients of the objective function and of the active constraints. The resulting first-order optimality conditions are known as the *Karush-Kuhn-Tucker (KKT) necessary conditions*. Beforehand, we introduce the important concepts of a *regular point* and of a *KKT point*.

Definition 1.36 (Regular Point (for a Set of Inequality Constraints)). *Let $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_g$, be differentiable functions on $\mathbb{R}^{n_x}$ and consider the set $S := \{\mathbf{x} \in \mathbb{R}^{n_x} : g_i(\mathbf{x}) \leq 0, i = 1, \ldots, n_g\}$. A point $\bar{\mathbf{x}} \in S$ is said to be a* regular point *if the gradient vectors $\nabla g_i(\bar{\mathbf{x}})$, $i \in \mathcal{A}(\bar{\mathbf{x}})$, are linearly independent,*

$$\operatorname{rank}\left(\nabla g_i(\bar{\mathbf{x}}), i \in \mathcal{A}(\bar{\mathbf{x}})\right) = |\mathcal{A}(\bar{\mathbf{x}})|.$$

Definition 1.37 (KKT Point). *Let $f : \mathbb{R}^{n_x} \to \mathbb{R}$ and $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_g$ be differentiable functions. Consider the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. If a point $(\bar{\mathbf{x}}, \bar{\boldsymbol{\nu}}) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_g}$ satisfies the conditions:*

$$\nabla f(\bar{\mathbf{x}}) + \bar{\boldsymbol{\nu}}^\mathsf{T} \nabla \mathbf{g}(\bar{\mathbf{x}}) = \mathbf{0} \qquad (1.7)$$

$$\bar{\boldsymbol{\nu}} \geq \mathbf{0} \qquad (1.8)$$

$$\mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0} \qquad (1.9)$$

$$\bar{\boldsymbol{\nu}}^\mathsf{T} \mathbf{g}(\bar{\mathbf{x}}) = 0, \qquad (1.10)$$

then $(\bar{\mathbf{x}}, \bar{\boldsymbol{\nu}})$ *is said to be a* KKT point.

Remark 1.38. The scalars ν_i, $i = 1, \ldots, n_g$, are called the *Lagrange multipliers*. The condition (1.7), i.e., the requirement that $\bar{\mathbf{x}}$ be feasible, is called the *primal feasibility* (PF) condition; the conditions (1.8) and (1.9) are referred to as the *dual feasibility* (DF) conditions; finally, the condition (1.10) is called the *complementarity slackness*[4] (CS) condition.

Theorem 1.39 (KKT Necessary Conditions). *Let* $f : \mathbb{R}^{n_x} \to \mathbb{R}$ *and* $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_g$ *be differentiable functions. Consider the problem to minimize* $f(\mathbf{x})$ *subject to* $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. *If* $\mathbf{x}^\star$ *is a local minimum and a regular point of the constraints, then there exists a* unique *vector* $\boldsymbol{\nu}^\star$ *such that* $(\mathbf{x}^\star, \boldsymbol{\nu}^\star)$ *is a KKT point.*

Proof. Since $\mathbf{x}^\star$ solves the problem, then there exists no direction $\mathbf{d} \in \mathbb{R}^{n_x}$ such that $\nabla f(\bar{\mathbf{x}})^\mathsf{T}\mathbf{d} < 0$ and $\nabla g_i(\bar{\mathbf{x}})^\mathsf{T}\mathbf{d} < 0$, $\forall i \in \mathcal{A}(\mathbf{x}^\star)$ simultaneously (see Remark 1.34). Let $\mathsf{A} \in \mathbb{R}^{(|\mathcal{A}(\mathbf{x}^\star)|+1) \times n_x}$ be the matrix whose rows are $\nabla f(\bar{\mathbf{x}})^\mathsf{T}$ and $\nabla g_i(\bar{\mathbf{x}})^\mathsf{T}$, $i \in \mathcal{A}(\mathbf{x}^\star)$. Clearly, the statement $\{\exists \mathbf{d} \in \mathbb{R}^{n_x} : \mathsf{A}\mathbf{d} < \mathbf{0}\}$ is false, and by Gordan's Theorem 1.A.78, there exists a nonzero vector $\mathbf{p} \geq \mathbf{0}$ in $\mathbb{R}^{|\mathcal{A}(\mathbf{x}^\star)|+1}$ such that $\mathsf{A}^\mathsf{T}\mathbf{p} = \mathbf{0}$. Denoting the components of $\mathbf{p}$ by u_0 and u_i for $i \in \mathcal{A}(\mathbf{x}^\star)$, we get:

$$u_0 \nabla f(\mathbf{x}^\star) + \sum_{i \in \mathcal{A}(\mathbf{x}^\star)} u_i \nabla g_i(\mathbf{x}^\star) = \mathbf{0}$$

where $u_0 \geq 0$ and $u_i \geq 0$ for $i \in \mathcal{A}(\mathbf{x}^\star)$, and $(u_0, \mathbf{u}_{\mathcal{A}(\mathbf{x}^\star)}) \neq (0, \mathbf{0})$ (here $\mathbf{u}_{\mathcal{A}(\mathbf{x}^\star)}$ is the vector whose components are the u_i's for $i \in \mathcal{A}(\mathbf{x}^\star)$). Letting $u_i = 0$ for $i \notin \mathcal{A}(\mathbf{x}^\star)$, we then get the conditions:

$$\begin{aligned}
u_0 \nabla f(\mathbf{x}^\star) + \mathbf{u}^\mathsf{T} \nabla \mathbf{g}(\mathbf{x}^\star) &= \mathbf{0} \\
\mathbf{u}^\mathsf{T}\mathbf{g}(\mathbf{x}^\star) &= \mathbf{0} \\
u_0, \mathbf{u} &\geq 0 \\
(u_0, \mathbf{u}) &\neq (0, \mathbf{0}),
\end{aligned}$$

where $\mathbf{u}$ is the vector whose components are u_i for $i = 1, \ldots, n_g$. Note that $u_0 \neq 0$, for otherwise the assumption of linear independence of the active constraints at $\mathbf{x}^\star$ would be violated. Then, letting $\boldsymbol{\nu}^\star = \frac{1}{u_0}\mathbf{u}$, we obtain that $(\mathbf{x}^\star, \boldsymbol{\nu}^\star)$ is a KKT point. □

Remark 1.40. One of the major difficulties in applying the foregoing result is that we do not know *a priori* which constraints are active and which constraints are inactive, i.e., the active set is *unknown*. Therefore, it is necessary to investigate *all* possible active sets for finding candidate points satisfying the KKT conditions. This is illustrated in Example 1.41 below.

Example 1.41 (Regular Case). Consider the problem

$$\begin{aligned}
\min_{\mathbf{x} \in \mathbb{R}^3} \;\; & f(\mathbf{x}) := \frac{1}{2}(x_1^2 + x_2^2 + x_3^2) \qquad (1.11) \\
\text{s.t.} \;\; & g_1(\mathbf{x}) := x_1 + x_2 + x_3 + 3 \leq 0 \\
& g_2(\mathbf{x}) := x_1 \leq 0.
\end{aligned}$$

[4] Often, the condition (1.10) is replaced by the equivalent conditions:

$$\bar{\nu}_i g_i(\bar{\mathbf{x}}) = 0 \quad \text{for } i = 1, \ldots, n_g.$$

Note that every feasible point is regular (the point (0,0,0) being infeasible), so $\mathbf{x}^\star$ must satisfy the dual feasibility conditions:

$$\begin{aligned} x_1^\star + \nu_1^\star + \nu_2^\star &= 0 \\ x_2^\star + \nu_1^\star &= 0 \\ x_3^\star + \nu_1^\star &= 0. \end{aligned}$$

Four cases can be distinguished:

(i) The constraints g_1 and g_2 are both *inactive*, i.e., $x_1^\star + x_2^\star + x_3^\star < -3$, $x_1^\star < 0$, and $\nu_1^\star = \nu_2^\star = 0$. From the latter together with the dual feasibility conditions, we get $x_1^\star = x_2^\star = x_3^\star = 0$, hence contradicting the former.

(ii) The constraint g_1 is *inactive*, while g_2 is *active*, i.e., $x_1^\star + x_2^\star + x_3^\star < -3$, $x_1^\star = 0$, $\nu_2^\star \geq 0$, and $\nu_1^\star = 0$. From the latter together with the dual feasibility conditions, we get $x_2^\star = x_3^\star = 0$, hence contradicting the former once again.

(iii) The constraint g_1 is *active*, while g_2 is *inactive*, i.e., $x_1^\star + x_2^\star + x_3^\star = -3$, $x_1^\star < 0$, and $\nu_1^\star \geq 0$, and $\nu_2^\star = 0$. Then, the point $(\mathbf{x}^\star, \boldsymbol{\nu}^\star)$ such that $x_1{}^\star = x_2{}^\star = x_3{}^\star = -1$, $\nu_1^\star = 1$ and $\nu_2^\star = 0$ is a KKT point.

(iv) The constraints g_1 and g_2 are both *active*, i.e., $x_1^\star + x_2^\star + x_3^\star = -3$, $x_1^\star = 0$, and $\nu_1^\star, \nu_2^\star > 0$. Then, we obtain $x_2{}^\star = x_3{}^\star = -\frac{3}{2}$, $\nu_1^\star = \frac{3}{2}$, and $\nu_2^\star = -\frac{3}{2}$, hence contradicting the dual feasibility condition $\nu_2^\star \geq 0$.

Overall, there is a *unique* candidate for a local minimum. Yet, it cannot be concluded as to whether this point is actually a global minimum, or even a local minimum, of (1.11). This question will be addressed later on in Example 1.45.

Remark 1.42 (Constraint Qualification). It is *very* important to note that for a local minimum $\mathbf{x}^\star$ to be a KKT point, an additional condition must be placed on the behavior of the constraint, i.e., **not every local minimum is a KKT point**; such a condition is known as a *constraint qualification*. In Theorem 1.39, it is shown that one possible constraint qualification is that $\mathbf{x}^\star$ be a regular point, which is the well known *linear independence constraint qualification* (LICQ). A weaker constraint qualification (i.e., implied by LICQ) known as the *Mangasarian-Fromovitz constraint qualification* (MFCQ) requires that there exits (at least) one direction $\mathbf{d} \in \mathcal{D}_0(\mathbf{x}^\star)$, i.e., such that $\nabla g_i(\mathbf{x}^\star)^\mathsf{T}\mathbf{d} < 0$, for each $i \in \mathcal{A}(\mathbf{x}^\star)$. Note, however, that the Lagrange multipliers are guaranteed to be unique if LICQ holds (as stated in Theorem 1.39), while this uniqueness property may be lost under MFCQ.

The following example illustrates the necessity of having a constraint qualification for a KKT point to be a local minimum point of an NLP.

Example 1.43 (Non-Regular Case). Consider the problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & f(\mathbf{x}) := -x_1 \\ \text{s.t.} \quad & g_1(\mathbf{x}) := x_2 - (1 - x_1)^3 \leq 0 \\ & g_2(\mathbf{x}) := -x_2 \leq 0. \end{aligned} \tag{1.12}$$

The feasible region is shown in Fig. 1.10. below. Note that a minimum point of (1.12) is $\mathbf{x}^\star = (1, 0)^\mathsf{T}$. The dual feasibility condition relative to variable x_1 reads:

$$-1 + 3\nu_1(1 - x_1)^2 \;=\; 0.$$

It is readily seen that this condition cannot be met at any point on the straight line $\mathcal{C} := \{\mathbf{x} : x_1 = 1\}$, including the minimum point $\mathbf{x}^\star$. In other words, the KKT conditions are not necessary in this example. This is because no constraint qualification can hold at $\mathbf{x}^\star$. In particular, $\mathbf{x}^\star$ not being a regular point, LICQ does not hold; moreover, the set $\mathcal{D}_0(\mathbf{x}^\star)$ being empty (the direction $\mathbf{d} = (-1, 0)^\mathsf{T}$ gives $\boldsymbol{\nabla} g_1(\mathbf{x}^\star)^\mathsf{T}\mathbf{d} = \boldsymbol{\nabla} g_2(\mathbf{x}^\star)^\mathsf{T}\mathbf{d} = 0$, while any other direction induces a violation of either one of the constraints), MFCQ does not hold at $\mathbf{x}^\star$ either.

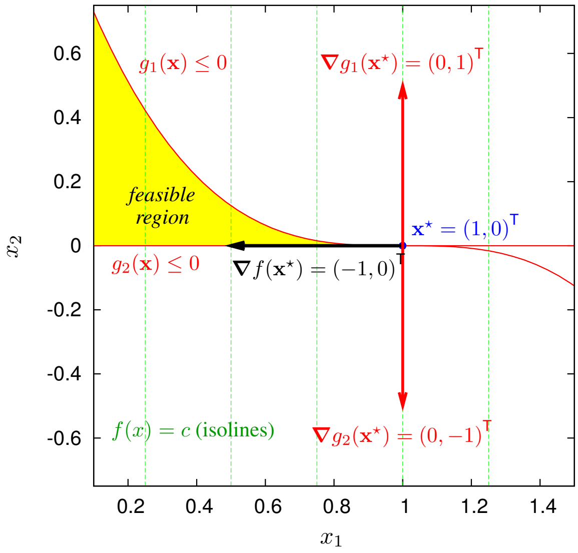


Figure 1.10. Solution of Example 1.43.

The following theorem provides a sufficient condition under which any KKT point of an inequality constrained NLP problem is guaranteed to be a global minimum of that problem.

Theorem 1.44 (KKT sufficient Conditions). *Let $f : \mathbb{R}^{n_x} \to \mathbb{R}$ and $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_g$, be convex and differentiable functions. Consider the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. If $(\mathbf{x}^\star, \boldsymbol{\nu}^\star)$ is a KKT point, then $\mathbf{x}^\star$ is a global minimum of that problem.*

Proof. Consider the function $\mathcal{L}(\mathbf{x}) := f(\mathbf{x}) + \sum_{i=1}^{n_g} \nu_i^\star g_i(\mathbf{x})$. Since f and g_i, $i = 1, \ldots, n_g$, are convex functions, and $\nu_i \geq 0$, $\mathcal{L}$ is also convex. Moreover, the dual feasibility conditions impose that we have $\boldsymbol{\nabla}\mathcal{L}(\mathbf{x}^\star) = \mathbf{0}$. Hence, by Theorem 1.27, $\mathbf{x}^\star$ is a global minimizer for $\mathcal{L}$ on $\mathbb{R}^{n_x}$, i.e.,

$$\mathcal{L}(\mathbf{x}) \;\geq\; \mathcal{L}(\mathbf{x}^\star) \quad \forall \mathbf{x} \in \mathbb{R}^{n_x}.$$

In particular, for each $\mathbf{x}$ such that $g_i(\mathbf{x}) \leq g_i(\mathbf{x}^\star) = 0$, $i \in \mathcal{A}(\mathbf{x}^\star)$, we have

$$f(\mathbf{x}) - f(\mathbf{x}^\star) \;\geq\; -\sum_{i \in \mathcal{A}(\mathbf{x}^\star)} \mu_i^\star \left[g_i(\mathbf{x}) - g_i(\mathbf{x}^\star)\right] \;\geq\; 0.$$

Noting that $\{\mathbf{x} \in \mathbb{R}^{n_x} : g_i(\mathbf{x}) \leq 0, i \in \mathcal{A}(\mathbf{x}^\star)\}$ contains the feasible domain $\{\mathbf{x} \in \mathbb{R}^{n_x} : g_i(\mathbf{x}) \leq 0, i = 1, \ldots, n_g\}$, we therefore showed that $\mathbf{x}^\star$ is a global minimizer for the problem. □

Example 1.45. Consider the same Problem (1.11) as in Example 1.41 above. The point $(\mathbf{x}^\star, \boldsymbol{\nu}^\star)$ with $x_1{}^\star = x_2{}^\star = x_3{}^\star = -1$, $\nu_1^\star = 1$ and $\nu_2^\star = 0$, being a KKT point, and both the objective function and the feasible set being convex, by Theorem 1.44, $\mathbf{x}^\star$ is a global minimum for the Problem (1.11).

Both second-order necessary and sufficient conditions for inequality constrained NLP problems will be presented later on in §1.7.

1.6 PROBLEMS WITH EQUALITY CONSTRAINTS

In this section, we shall consider nonlinear programming problems with equality constraints of the form:

$$\begin{array}{rl} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & h_i(\mathbf{x}) \;=\; 0 \quad i = 1, \ldots, n_h. \end{array}$$

Based on the material presented in §1.5, it is tempting to convert this problem into an inequality constrained problem, by replacing each equality constraints $h_i(\mathbf{x}) = 0$ by two inequality constraints $h_i^+(\mathbf{x}) = h_i(\mathbf{x}) \leq 0$ and $h_i^-(\mathbf{x}) = -h(\mathbf{x}) \leq 0$. Given a feasible point $\bar{\mathbf{x}} \in \mathbb{R}^{n_x}$, we would have $h_i^+(\bar{\mathbf{x}}) = h_i^-(\bar{\mathbf{x}}) = 0$ and $\nabla h_i^+(\bar{\mathbf{x}}) = -\nabla h_i^-(\bar{\mathbf{x}})$. Therefore, there could exist no vector $\mathbf{d}$ such that $\nabla h_i^+(\bar{\mathbf{x}}) < 0$ and $\nabla h_i^-(\bar{\mathbf{x}}) < 0$ simultaneously, i.e., $\mathcal{D}_0(\bar{\mathbf{x}}) = \emptyset$. In other words, the geometric conditions developed in the previous section for inequality constrained problems are satisfied by all feasible solutions and, hence, are not informative (see Example 1.35 for an illustration). A different approach must therefore be used to deal with equality constrained problems. After a number of preliminary results in §1.6.1, we shall describe the method of Lagrange multipliers for equality constrained problems in §1.6.2.

1.6.1 Preliminaries

An equality constraint $h(\mathbf{x}) = 0$ defines a set on $\mathbb{R}^{n_x}$, which is best view as a hypersurface. When considering $n_h \geq 1$ equality constraints $h_1(\mathbf{x}), \ldots, h_{n_h}(\mathbf{x})$, their intersection forms a (possibly empty) set $S := \{\mathbf{x} \in \mathbb{R}^{n_x} : h_i(\mathbf{x}) = 0, i = 1, \ldots, n_h\}$.

Throughout this section, we shall assume that the equality constraints are differentiable; that is, the set $S := \{\mathbf{x} \in \mathbb{R}^{n_x} : h_i(\mathbf{x}) = 0, i = 1, \ldots, n_h\}$ is said to be *differentiable manifold* (or *smooth manifold*). Associated with a point on a differentiable manifold is the *tangent set* at that point. To formalize this notion, we start by defining *curves* on a manifold. A curve $\boldsymbol{\xi}$ on a manifold S is a continuous application $\boldsymbol{\xi} : \mathcal{I} \subset \mathbb{R} \to S$, i.e., a family of points $\boldsymbol{\xi}(t) \in S$ continuously parameterized by t in an interval $\mathcal{I}$ of $\mathrm{I\!R}$. A curve is said to pass through the point $\bar{\mathbf{x}}$ if $\bar{\mathbf{x}} = \boldsymbol{\xi}(\bar{t})$ for some $\bar{t} \in \mathcal{I}$; the *derivative* of a curve at $\bar{t}$, provided it exists, is defined as $\dot{\boldsymbol{\xi}}(\bar{t}) := \lim_{h \to 0} \frac{\boldsymbol{\xi}(\bar{t}+h) - \boldsymbol{\xi}(\bar{t})}{h}$. A curve is said to be *differentiable* (or *smooth*) if a derivative exists for each $t \in \mathcal{I}$.

Definition 1.46 (Tangent Set). *Let S be a (differentiable) manifold in $\mathrm{I\!R}^{n_x}$, and let $\bar{\mathbf{x}} \in S$. Consider the collection of* all *the continuously differentiable curves on S passing through $\bar{\mathbf{x}}$. Then, the collection of* all *the vectors tangent to these curves at $\bar{\mathbf{x}}$ is said to be the* tangent set *to S at $\bar{\mathbf{x}}$, denoted by $\mathscr{T}(\bar{\mathbf{x}})$.*

If the constraints are *regular*, in the sense of Definition 1.47 below, then S is (locally) of dimension $n_x - n_h$, and $\mathscr{T}(\bar{\mathbf{x}})$ constitutes a subspace of dimension $n_x - n_h$, called the *tangent space*.

Definition 1.47 (Regular Point (for a Set of Equality Constraints)). *Let $h_i : \mathrm{I\!R}^{n_x} \to \mathrm{I\!R}$, $i = 1, \ldots, n_h$, be differentiable functions on $\mathrm{I\!R}^{n_x}$ and consider the set $S := \{\mathbf{x} \in \mathrm{I\!R}^{n_x} : h_i(\mathbf{x}) = 0, i = 1, \ldots, n_h\}$. A point $\bar{\mathbf{x}} \in S$ is said to be a* regular point *if the gradient vectors $\boldsymbol{\nabla} h_i(\bar{\mathbf{x}})$, $i = 1, \ldots, n_h$, are linearly independent, i.e.,*

$$\operatorname{rank} \begin{pmatrix} \boldsymbol{\nabla} h_1(\bar{\mathbf{x}}) & \boldsymbol{\nabla} h_2(\bar{\mathbf{x}}) & \cdots & \boldsymbol{\nabla} h_{n_h}(\bar{\mathbf{x}}) \end{pmatrix} = n_h.$$

Lemma 1.48 (Algebraic Characterization of a Tangent Space). *Let $h_i : \mathrm{I\!R}^{n_x} \to \mathrm{I\!R}$, $i = 1, \ldots, n_h$, be differentiable functions on $\mathrm{I\!R}^{n_x}$ and consider the set $S := \{\mathbf{x} \in \mathrm{I\!R}^{n_x} : h_i(\mathbf{x}) = 0, i = 1, \ldots, n_h\}$. At a regular point $\bar{\mathbf{x}} \in S$, the tangent space is such that*

$$\mathscr{T}(\bar{\mathbf{x}}) = \{\mathbf{d} : \boldsymbol{\nabla}\mathbf{h}(\bar{\mathbf{x}})^{\mathsf{T}}\mathbf{d} = \mathbf{0}\}.$$

Proof. The proof is technical and is omitted here (see, e.g., [36, §10.2]). □

1.6.2 The Method of Lagrange Multipliers

The idea behind the method of Lagrange multipliers for solving equality constrained NLP problems of the form

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & h_i(\mathbf{x}) \;=\; 0 \quad i = 1, \ldots, n_h. \end{array}$$

is to restrict the search of a minimum on the manifold $S := \{\mathbf{x} \in \mathrm{I\!R}^{n_x} : h_i(\mathbf{x}) = 0, \forall i = 1, \ldots, n_h\}$. In other words, we derive optimality conditions by considering the value of the objective function along curves on the manifold S passing through the optimal point.

The following Theorem shows that the tangent space $\mathscr{T}(\bar{\mathbf{x}})$ at a regular (local) minimum point $\bar{\mathbf{x}}$ is orthogonal to the gradient of the objective function at $\bar{\mathbf{x}}$. This important fact is illustrated in Fig. 1.11. in the case of a single equality constraint.

Theorem 1.49 (Geometric Necessary Condition for a Local Minimum). *Let $f : \mathrm{I\!R}^{n_x} \to \mathrm{I\!R}$ and $h_i : \mathrm{I\!R}^{n_x} \to \mathrm{I\!R}$, $i = 1, \ldots, n_h$, be continuously differentiable functions on $\mathrm{I\!R}^{n_x}$. Suppose that $\mathbf{x}^\star$ is a local minimum point of the problem to minimize $f(\mathbf{x})$ subject to the constraints $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. Then, $\boldsymbol{\nabla} f(\mathbf{x}^\star)$ is orthogonal to the tangent space $\mathscr{T}(\mathbf{x}^\star)$,*

$$\mathcal{F}_0(\mathbf{x}^\star) \cap \mathscr{T}(\mathbf{x}^\star) \;=\; \emptyset.$$

Proof. By contradiction, assume that there exists a $\mathbf{d} \in \mathscr{T}(\mathbf{x}^\star)$ such that $\boldsymbol{\nabla} f(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{d} \neq 0$. Let $\boldsymbol{\xi} : \mathcal{I} = [-a, a] \to S$, $a > 0$, be any smooth curve passing through $\mathbf{x}^\star$ with $\boldsymbol{\xi}(0) = \mathbf{x}^\star$

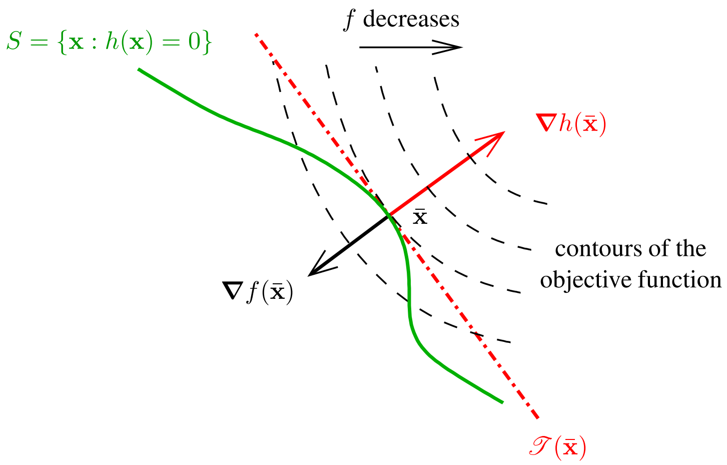

$S = \{\mathbf{x} : h(\mathbf{x}) = 0\}$
f decreases
$\boldsymbol{\nabla} h(\bar{\mathbf{x}})$
$\bar{\mathbf{x}}$
contours of the objective function
$\boldsymbol{\nabla} f(\bar{\mathbf{x}})$
$\mathscr{T}(\bar{\mathbf{x}})$

Figure 1.11. Illustration of the necessary conditions of optimality with equality constraints.

and $\dot{\boldsymbol{\xi}}(0) = \mathbf{d}$. Let also φ be the function defined as $\varphi(t) := f(\boldsymbol{\xi}(t)), \forall t \in \mathcal{I}$. Since $\mathbf{x}^\star$ is a local minimum of f on $S := \{\mathbf{x} \in \mathbb{R}^{n_x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$, by Definition 1.9, we have

$$\exists \eta > 0 \text{ such that } \varphi(t) = f(\boldsymbol{\xi}(t)) \geq f(\mathbf{x}^\star) = \varphi(0) \quad \forall t \in \mathcal{B}_\eta(0) \cap \mathcal{I}.$$

It follows that $t^\star = 0$ is an unconstrained (local) minimum point for φ, and

$$0 \;=\; \nabla\varphi(0) \;=\; \boldsymbol{\nabla} f(\mathbf{x}^\star)^{\mathsf{T}}\dot{\boldsymbol{\xi}}(0) \;=\; \boldsymbol{\nabla} f(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{d}.$$

We thus get a contradiction with the fact that $\boldsymbol{\nabla} f(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{d} \neq 0$. □

Next, we take advantage of the forgoing geometric characterization, and derive first-order necessary conditions for equality constrained NLP problems.

Theorem 1.50 (First-Order Necessary Conditions). *Let $f : \mathbb{R}^{n_x} \to \mathbb{R}$ and $h_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_h$, be continuously differentiable functions on $\mathbb{R}^{n_x}$. Consider the problem to minimize $f(\mathbf{x})$ subject to the constraints $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. If $\mathbf{x}^\star$ is a local minimum and is a regular point of the constraints, then there exists a unique vector $\boldsymbol{\lambda}^\star \in \mathbb{R}^{n_h}$ such that*

$$\boldsymbol{\nabla} f(\mathbf{x}^\star) + \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^\star)^{\mathsf{T}}\boldsymbol{\lambda}^\star = \mathbf{0}.$$

Proof. [5] Since $\mathbf{x}^\star$ is a local minimum of f on $S := \{\mathbf{x} \in \mathbb{R}^{n_x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$, by Theorem 1.49, we have $\mathcal{F}_0(\mathbf{x}^\star) \cap \mathscr{T}(\mathbf{x}^\star) = \emptyset$, i.e., the system

$$\boldsymbol{\nabla} f(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{d} < 0 \qquad \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{d} = 0,$$

is inconsistent. Consider the following two sets:

$$\begin{aligned} C_1 &:= \{(z_1, \mathbf{z}_2) \in \mathbb{R}^{n_h+1} : z_1 = \boldsymbol{\nabla} f(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{d}, \quad \mathbf{z}_2 = \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{d}\} \\ C_2 &:= \{(z_1, \mathbf{z}_2) \in \mathbb{R}^{n_h+1} : z_1 < 0, \quad \mathbf{z}_2 = \mathbf{0}.\} \end{aligned}$$

[5] See also in Appendix of §1 for an alternative proof of Theorem 1.50 that does not use the concept of tangent sets.

Clearly, C_1 and C_2 are convex, and $C_1 \cap C_2 = \emptyset$. Then, by the separation Theorem A.9, there exists a nonzero vector $(\mu, \boldsymbol{\lambda}) \in \mathrm{IR}^{n_h+1}$ such that

$$\mu \nabla f(\mathbf{x}^\star)^\mathsf{T}\mathbf{d} + \boldsymbol{\lambda}^\mathsf{T}[\nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T}\mathbf{d}] \ \geq \ \mu z_1 + \boldsymbol{\lambda}^\mathsf{T}\mathbf{z}_2 \qquad \forall \mathbf{d} \in \mathrm{IR}^{n_x},\ \forall (z_1, \mathbf{z}_2) \in C_2.$$

Letting $\mathbf{z}_2 = \mathbf{0}$ and since z_1 can be made an arbitrarily large negative number, it follows that $\mu \geq 0$. Also, letting $(z_1, \mathbf{z}_2) = (0, \mathbf{0})$, we must have $[\mu \nabla f(\mathbf{x}^\star) + \boldsymbol{\lambda}^\mathsf{T}\nabla \mathbf{h}(\mathbf{x}^\star)]^\mathsf{T}\mathbf{d} \geq \mathbf{0}$, for each $\mathbf{d} \in \mathrm{IR}^{n_x}$. In particular, letting $\mathbf{d} = -[\mu \nabla f(\mathbf{x}^\star) + \boldsymbol{\lambda}^\mathsf{T}\nabla \mathbf{h}(\mathbf{x}^\star)]$, it follows that $-\|\mu \nabla f(\mathbf{x}^\star) + \boldsymbol{\lambda}^\mathsf{T}\nabla \mathbf{h}(\mathbf{x}^\star)\|^2 \geq 0$, and thus,

$$\mu \nabla f(\mathbf{x}^\star) + \boldsymbol{\lambda}^\mathsf{T}\nabla \mathbf{h}(\mathbf{x}^\star) = \mathbf{0} \qquad \text{with } (\mu, \boldsymbol{\lambda}) \neq (0, \mathbf{0}). \tag{1.13}$$

Finally, note that $\mu > 0$, for otherwise (1.13) would contradict the assumption of linear independence of $\nabla h_i(\mathbf{x}^\star)$, $i = 1, \ldots, n_h$. The result follows by letting $\boldsymbol{\lambda}^\star := \frac{1}{\mu}\boldsymbol{\lambda}$, and noting that the linear independence assumption implies the uniqueness of these Lagrangian multipliers. □

Remark 1.51 (Obtaining Candidate Solution Points). The first-order necessary conditions

$$\nabla f(\mathbf{x}^\star) + \nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\lambda}^\star = \mathbf{0},$$

together with the constraints

$$\mathbf{h}(\mathbf{x}^\star) = \mathbf{0},$$

give a total of $n_x + n_h$ (typically nonlinear) equations in the variables $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$. Hence, these conditions are complete in the sense that they determine, at least locally, a unique solution. However, as in the unconstrained case, a solution to the first-order necessary conditions need not be a (local) minimum of the original problem; it could very well correspond to a (local) maximum point, or some kind of saddle point. These consideration are illustrated in Example 1.54 below.

Remark 1.52 (Regularity-Type Assumption). It is important to note that for a local minimum to satisfy the foregoing first-order conditions and, in particular, for a unique Lagrange multiplier vector to exist, it is necessary that the equality constraint satisfy a regularity condition. In other word, the first-order conditions may not hold at a local minimum point that is non-regular. An illustration of these considerations is provided in Example 1.55.

There exists a number of similarities with the constraint qualification needed for a local minimizer of an inequality constrained NLP problem to be KKT point; in particular, the condition that the minimum point be a regular point for the constraints corresponds to LICQ (see Remark 1.42).

Remark 1.53 (Lagrangian). It is convenient to introduce the *Lagrangian* $\mathcal{L} : \mathrm{IR}^{n_x} \times \mathrm{IR}^{n_h} \to \mathrm{IR}$ associated with the constrained problem, by adjoining the cost and constraint functions as:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) \ := \ f(\mathbf{x}) + \boldsymbol{\lambda}^\mathsf{T}\mathbf{h}(\mathbf{x}).$$

Thus, if $\mathbf{x}^\star$ is a local minimum which is regular, the first-order necessary conditions are written as

$$\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = \mathbf{0} \tag{1.14}$$

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = \mathbf{0}, \tag{1.15}$$

the latter equations being simply a restatement of the constraints. Note that the solution of the original problem typically corresponds to a saddle point of the Lagrangian function.

Example 1.54 (Regular Case). Consider the problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^2} \; & f(\mathbf{x}) := x_1 + x_2 \\ \text{s.t.} \; & h(\mathbf{x}) := x_1^2 + x_2^2 - 2 = 0. \end{aligned} \tag{1.16}$$

Observe first that every feasible point is a regular point for the equality constraint (the point (0,0) being infeasible). Therefore, every local minimum is a stationary point of the Lagrangian function by Theorem 1.50.

The gradient vectors $\nabla f(\mathbf{x})$ and $\nabla h(\mathbf{x})$ are given by

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 1 & 1 \end{pmatrix}^{\mathsf{T}} \quad \text{and} \quad \nabla h(\mathbf{x}) = \begin{pmatrix} 2x_1 & 2x_2 \end{pmatrix}^{\mathsf{T}},$$

so that the first-order necessary conditions read

$$\begin{aligned} 2\lambda x_1 &= -1 \\ 2\lambda x_2 &= -1 \\ x_1^2 + x_2^2 &= 2. \end{aligned}$$

These three equations can be solved for the three unknowns x_1, x_2 and λ. Two candidate local minimum points are obtained: (i) ${x_1}^\star = {x_2}^\star = -1$, $\lambda^\star = \frac{1}{2}$, and (ii) ${x_1}^\star = {x_2}^\star = 1$, $\lambda^\star = -\frac{1}{2}$. These results are illustrated on Fig. 1.12.. It can be seen that only the former actually corresponds to a local minimum point, while the latter gives a local maximum point.

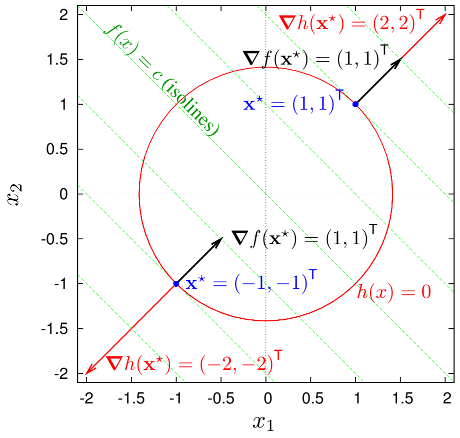


Figure 1.12. Solution of Example 1.54.

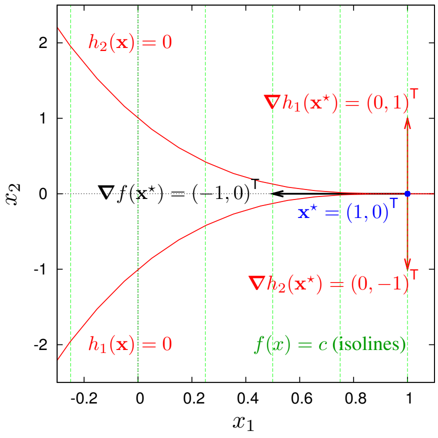


Figure 1.13. Solution of Example 1.55.

Example 1.55 (Non-Regular Case). Consider the problem

$$\begin{array}{cl}\min\limits_{\mathbf{x}\in\mathbb{R}^2} & f(\mathbf{x}) := -x_1 \\ \text{s.t.} & h_1(\mathbf{x}) := (1-x_1)^3 + x_2 = 0 \\ & h_2(\mathbf{x}) := (1-x_1)^3 - x_2 = 0. \end{array} \tag{1.17}$$

As shown by Fig. 1.13., this problem has only one feasible point, namely, $\mathbf{x}^\star = (1,0)^\mathsf{T}$; that is, $\mathbf{x}^\star$ is also the unique global minimum of (1.17). However, at this point, we have

$$\nabla f(\mathbf{x}^\star) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \nabla h_1(\mathbf{x}^\star) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \nabla h_2(\mathbf{x}^\star) = \begin{pmatrix} 0 \\ -1 \end{pmatrix},$$

hence the first-order conditions

$$\lambda_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \lambda_2 \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

cannot be satisfied. This illustrates the fact that a minimum point may not be a stationary point for the Lagrangian if that point is non-regular.

The following theorem provides second-order necessary conditions for a point to be a local minimum of a NLP problem with equality constraints.

Theorem 1.56 (Second-Order Necessary Conditions). *Let $f : \mathbb{R}^{n_x} \to \mathbb{R}$ and $h_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_h$, be twice continuously differentiable functions on $\mathbb{R}^{n_x}$. Consider the problem to minimize $f(\mathbf{x})$ subject to the constraints $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. If $\mathbf{x}^\star$ is a local minimum and is a regular point of the constraints, then there exists a unique vector $\boldsymbol{\lambda}^\star \in \mathbb{R}^{n_h}$ such that*

$$\nabla f(\mathbf{x}^\star) + \nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T} \boldsymbol{\lambda}^\star = \mathbf{0},$$

and

$$\mathbf{d}^\mathsf{T}\left(\nabla^2 f(\mathbf{x}^\star) + \nabla^2 \mathbf{h}(\mathbf{x}^\star)^\mathsf{T} \boldsymbol{\lambda}^\star\right)\mathbf{d} \geq 0 \qquad \forall \mathbf{d} \textit{ such that } \nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T}\mathbf{d} = 0.$$

Proof. Note first that $\nabla f(\mathbf{x}^\star) + \nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T} \boldsymbol{\lambda}^\star = \mathbf{0}$ directly follows from Theorem 1.50.

Let $\mathbf{d}$ be an arbitrary direction in $\mathscr{T}(\mathbf{x}^\star)$; that is, $\nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T}\mathbf{d} = 0$ since $\mathbf{x}^\star$ is a regular point (see Lemma 1.48). Consider any twice-differentiable curve $\boldsymbol{\xi} : \mathcal{I} = [-a, a] \to S$, $a > 0$, passing through $\mathbf{x}^\star$ with $\boldsymbol{\xi}(0) = \mathbf{x}^\star$ and $\dot{\boldsymbol{\xi}}(0) = \mathbf{d}$. Let φ be the function defined as $\varphi(t) := f(\boldsymbol{\xi}(t))$, $\forall t \in \mathcal{I}$. Since $\mathbf{x}^\star$ is a local minimum of f on $S := \{\mathbf{x} \in \mathbb{R}^{n_x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$, $t^\star = 0$ is an unconstrained (local) minimum point for φ. By Theorem 1.25, it follows that

$$0 \ \leq \ \nabla^2 \varphi(0) \ = \ \dot{\boldsymbol{\xi}}(0)^\mathsf{T} \nabla^2 f(\mathbf{x}^\star) \dot{\boldsymbol{\xi}}(0) + \nabla f(\mathbf{x}^\star)^\mathsf{T} \ddot{\boldsymbol{\xi}}(0).$$

Furthermore, differentiating the relation $\mathbf{h}(\boldsymbol{\xi}(\mathbf{t}))^\mathsf{T}\boldsymbol{\lambda} = 0$ twice, we obtain

$$\dot{\boldsymbol{\xi}}(0)^\mathsf{T}\left(\nabla^2 \mathbf{h}(\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\lambda}\right)\dot{\boldsymbol{\xi}}(0) + \left(\nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\lambda}\right)^\mathsf{T}\ddot{\boldsymbol{\xi}}(0) \ = \ 0.$$

Adding the last two equations yields

$$\mathbf{d}^\mathsf{T}\left(\nabla^2 f(\mathbf{x}^\star) + \nabla^2 \mathbf{h}(\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\lambda}^\star\right)\mathbf{d} \geq 0,$$

and this condition must hold for every $\mathbf{d}$ such that $\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{d} = 0$. □

Remark 1.57 (Eigenvalues in Tangent Space). In the foregoing theorem, it is shown that the matrix $\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$ restricted to the subspace $\mathscr{T}(\mathbf{x}^\star)$ plays a key role. Geometrically, the restriction of $\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$ to $\mathscr{T}(\mathbf{x}^\star)$ corresponds to the projection $\mathcal{P}_{\mathscr{T}(\mathbf{x}^\star)}[\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)]$.

A vector $\mathbf{y} \in \mathscr{T}(\mathbf{x}^\star)$ is said to be an *eigenvector* of $\mathcal{P}_{\mathscr{T}(\mathbf{x}^\star)}[\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)]$ if there is a real number μ such that

$$\mathcal{P}_{\mathscr{T}(\mathbf{x}^\star)}[\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)]\mathbf{y} \;=\; \mu\mathbf{y};$$

the corresponding μ is said to be an *eigenvalue* of $\mathcal{P}_{\mathscr{T}(\mathbf{x}^\star)}[\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)]$. (These definitions coincide with the usual definitions of eigenvector and eigenvalue for real matrices.)

Now, to obtain a matrix representation for $\mathcal{P}_{\mathscr{T}(\mathbf{x}^\star)}[\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)]$, it is necessary to introduce a basis of the subspace $\mathscr{T}(\mathbf{x}^\star)$, say $\mathsf{E} = (\mathbf{e}_1, \ldots, \mathbf{e}_{n_x - n_h})$. Then, the eigenvalues of $\mathcal{P}_{\mathscr{T}(\mathbf{x}^\star)}[\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)]$ are the same as those of the $(n_x - n_h) \times (n_x - n_h)$ matrix $\mathsf{E}^{\mathsf{T}}\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)\mathsf{E}$; in particular, they are independent of the particular choice of basis E.

Example 1.58 (Regular Case Continued). Consider the problem (1.16) addressed earlier in Example 1.54. Two candidate local minimum points, (i) $x_1{}^\star = x_2{}^\star = -1$, $\lambda^\star = \frac{1}{2}$, and (ii) $x_1{}^\star = x_2{}^\star = 1$, $\lambda^\star = -\frac{1}{2}$, were obtained on application of the first-order necessary conditions. The Hessian matrix of the Lagrangian function is given by

$$\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}, \lambda) \;=\; \boldsymbol{\nabla}^2 f(\mathbf{x}) + \lambda\boldsymbol{\nabla}^2 h(\mathbf{x}) \;=\; \lambda\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix},$$

and a basis of the tangent subspace at a point $\mathbf{x} \in \mathscr{T}(\mathbf{x})$, $\mathbf{x} \neq (0, 0)$, is

$$\mathsf{E}(\mathbf{x}) \;:=\; \begin{pmatrix} -x_2 \\ x_1 \end{pmatrix}.$$

Therefore,

$$\mathsf{E}^{\mathsf{T}}\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})\mathsf{E} \;=\; 2\lambda(x_1^2 + x_2^2).$$

In particular, for the candidate solution point (i), we have

$$\mathsf{E}^{\mathsf{T}}\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)\mathsf{E} \;=\; 2 \;>\; 0,$$

hence satisfying the second-order necessary conditions (in fact, this point also satisfies the second-order sufficient conditions of optimality discussed hereafter). On the other hand, for the candidate solution point (ii), we get

$$\mathsf{E}^{\mathsf{T}}\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)\mathsf{E} \;=\; -2 \;<\; 0$$

which does not satisfy the second-order requirement, so this point cannot be a local minimum.

The conditions given in Theorems 1.50 and 1.56 are necessary conditions that must hold at each local minimum point. Yet, a point satisfying these conditions may not be a local minimum. The following theorem provides sufficient conditions for a stationary point of the Lagrangian function to be a (local) minimum, provided that the Hessian matrix of the Lagrangian function is locally convex along directions in the tangent space of the constraints.

Theorem 1.59 (Second-Order Sufficient Conditions). *Let* $f : \mathbb{R}^{n_x} \to \mathbb{R}$ *and* $h_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_h$, *be twice continuously differentiable functions on* $\mathbb{R}^{n_x}$. *Consider the problem to minimize* $f(\mathbf{x})$ *subject to the constraints* $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. *If* $\mathbf{x}^\star$ *and* $\boldsymbol{\lambda}^\star$ *satisfy*

$$\begin{aligned}\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = 0\\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = 0,\end{aligned}$$

and

$$\mathbf{y}^{\mathsf{T}}\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)\mathbf{y} > 0 \quad \forall \mathbf{y} \neq \mathbf{0} \text{ such that } \nabla\mathbf{h}(\mathbf{x}^\star)^{\mathsf{T}}\mathbf{y} = 0,$$

where $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^{\mathsf{T}}\mathbf{h}(\mathbf{x})$, *then* $\mathbf{x}^\star$ *is a strict local minimum.*

Proof. Consider the augmented Lagrangian function

$$\bar{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^{\mathsf{T}}\mathbf{h}(\mathbf{x}) + \frac{c}{2}\|\mathbf{h}(\mathbf{x})\|^2,$$

where c is a scalar. We have

$$\begin{aligned}\nabla_{\mathbf{x}}\bar{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda}) &= \nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \bar{\boldsymbol{\lambda}})\\ \nabla^2_{\mathbf{xx}}\bar{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda}) &= \nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}, \bar{\boldsymbol{\lambda}}) + c\nabla\mathbf{h}(\mathbf{x})^{\mathsf{T}}\nabla\mathbf{h}(\mathbf{x}),\end{aligned}$$

where $\bar{\boldsymbol{\lambda}} = \boldsymbol{\lambda} + c\mathbf{h}(\mathbf{x})$. Since $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$ satisfy the sufficient conditions and by Lemma 1.A.79, we obtain

$$\nabla_{\mathbf{x}}\bar{\mathcal{L}}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = 0 \quad \text{and} \quad \nabla^2_{\mathbf{xx}}\bar{\mathcal{L}}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) \succ 0,$$

for sufficiently large c. $\bar{\mathcal{L}}$ being definite positive at $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$,

$$\exists \varrho > 0, \delta > 0 \text{ such that } \bar{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda}^\star) \geq \bar{\mathcal{L}}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) + \frac{\varrho}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2 \quad \text{for } \|\mathbf{x} - \mathbf{x}^\star\| < \delta.$$

Finally, since $\bar{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda}^\star) = f(\mathbf{x})$ when $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, we get

$$f(\mathbf{x}) \geq f(\mathbf{x}^\star) + \frac{\varrho}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2 \quad \text{if } \mathbf{h}(\mathbf{x}) = \mathbf{0}, \|\mathbf{x} - \mathbf{x}^\star\| < \delta,$$

i.e., $\mathbf{x}^\star$ is a strict local minimum. □

Example 1.60. Consider the problem

$$\begin{aligned}\min_{\mathbf{x} \in \mathbb{R}^3} \; & f(\mathbf{x}) := -x_1x_2 - x_1x_3 - x_2x_3\\ \text{s.t.} \; & h(\mathbf{x}) := x_1 + x_2 + x_3 - 3 = 0.\end{aligned} \tag{1.18}$$

The first-order conditions for this problem are

$$\begin{aligned}-(x_2 + x_3) + \lambda &= 0\\ -(x_1 + x_3) + \lambda &= 0\\ -(x_1 + x_2) + \lambda &= 0,\end{aligned}$$

together with the equality constraint. It is easily checked that the point $x_1^\star = x_2^\star = x_3^\star = 1$, $\lambda^\star = 2$ satisfies these conditions. Moreover,

$$\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = \boldsymbol{\nabla}^2 f(\mathbf{x}^\star) = \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix},$$

and a basis of the tangent space to the constraint $h(\mathbf{x}) = 0$ at $\mathbf{x}^\star$ is

$$\mathsf{E} := \begin{pmatrix} 0 & 2 \\ 1 & -1 \\ -1 & -1 \end{pmatrix}.$$

We thus obtain

$$\mathsf{E}^{\mathsf{T}}\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)\mathsf{E} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix},$$

which is clearly a definite positive matrix. Hence, $\mathbf{x}^\star$ is a strict local minimum of (1.18). (Interestingly enough, the Hessian matrix of the objective function itself is indefinite at $\mathbf{x}^\star$ in this case.)

We close this section by providing insight into the Lagrange multipliers.

Remark 1.61 (Interpretation of the Lagrange Multipliers). The concept of Lagrange multipliers allows to adjoin the constraints to the objective function. That is, one can view constrained optimization as a search for a vector $\mathbf{x}^\star$ at which the gradient of the objective function is a linear combination of the gradients of constraints.

Another insightful interpretation of the Lagrange multipliers is as follows. Consider the set of perturbed problems $v^\star(y) := \min\{f(\mathbf{x}) : h(\mathbf{x}) = y\}$. Suppose that there is a unique regular solution point for each y, and let $\{\boldsymbol{\xi}^\star(y)\} := \arg\min\{f(\mathbf{x}) : h(\mathbf{x}) = y\}$ denote the evolution of the optimal solution point as a function of the perturbation parameter y. Clearly,

$$v(0) = f(\mathbf{x}^\star) \quad \text{and} \quad \boldsymbol{\xi}(0) = \mathbf{x}^\star.$$

Moreover, since $h(\boldsymbol{\xi}(y)) = y$ for each y, we have

$$\nabla_y h(\boldsymbol{\xi}(y)) = 1 = \boldsymbol{\nabla}_{\mathbf{x}} h(\boldsymbol{\xi}(y))^{\mathsf{T}} \boldsymbol{\nabla}_y \boldsymbol{\xi}(y).$$

Denoting by $\lambda^\star$ the Lagrange multiplier associated to the constraint $h(\mathbf{x}) = 0$ in the original problem, we have

$$\nabla_y v(0) = \boldsymbol{\nabla}_{\mathbf{x}} f(\mathbf{x}^\star)^{\mathsf{T}} \boldsymbol{\nabla}_y \boldsymbol{\xi}(0) = -\lambda^\star \boldsymbol{\nabla}_{\mathbf{x}} h(\mathbf{x}^\star)^{\mathsf{T}} \boldsymbol{\nabla}_y \boldsymbol{\xi}(0) = -\lambda^\star.$$

Therefore, the Lagrange multipliers $\lambda^\star$ can be interpreted as the sensitivity of the objective function f with respect to the constraint h. Said differently, $\lambda^\star$ indicates how much the optimal cost would change, if the constraints were perturbed.

This interpretation extends straightforwardly to NLP problems having inequality constraints. The Lagrange multipliers of an active constraints $g(\mathbf{x}) \leq 0$, say $\nu^\star > 0$, can be interpreted as the sensitivity of $f(\mathbf{x}^\star)$ with respect to a change in that constraints, as $g(\mathbf{x}) \leq y$; in this case, the positivity of the Lagrange multipliers follows from the fact that by increasing y, the feasible region is relaxed, hence the optimal cost cannot increase. Regarding inactive constraints, the sensitivity interpretation also explains why the Lagrange multipliers are zero, as any infinitesimal change in the value of these constraints leaves the optimal cost value unchanged.

1.7 GENERAL NLP PROBLEMS

In this section, we shall consider general, nonlinear programming problems with both equality and inequality constraints,

$$\begin{array}{rrcl} \text{minimize} & f(\mathbf{x}) & & \\ \text{subject to} & g_i(\mathbf{x}) & \leq & 0, \quad i = 1, \ldots, n_g \\ & h_i(\mathbf{x}) & = & 0, \quad i = 1, \ldots, n_h. \end{array}$$

Before stating necessary and sufficient conditions for such problems, we shall start by revisiting the KKT conditions for inequality constrained problems, based on the method of Lagrange multipliers described in §1.6.

1.7.1 KKT Conditions for Inequality Constrained NLP Problems Revisited

Consider the problem to minimize a function $f(\mathbf{x})$ for $\mathbf{x} \in S := \{\mathbf{x} \in \mathbb{R}^{n_x} : g_i(\mathbf{x}) \leq 0, i = 1, \ldots, n_g\}$, and suppose that $\mathbf{x}^\star$ is a local minimum point. Clearly, $\mathbf{x}^\star$ is also a local minimum of the inequality constrained problem where the inactive constraints $g_i(\mathbf{x}) \leq 0$, $i \notin \mathcal{A}(\mathbf{x}^\star)$, have been discarded. Thus, in effect, *inactive constraints at* $\mathbf{x}^\star$ *don't matter*; they can be ignored in the statement of optimality conditions.

On the other hand, *active inequality constraints can be treated to a large extend as equality constraints* at a local minimum point. In particular, it should be clear that $\mathbf{x}^\star$ is also a local minimum to the equality constrained problem

$$\begin{array}{rrcl} \text{minimize} & f(\mathbf{x}) & & \\ \text{subject to} & g_i(\mathbf{x}) & = & 0, \quad i \in \mathcal{A}(\mathbf{x}^\star). \end{array}$$

That is, it follows from Theorem 1.50 that, if $\mathbf{x}^\star$ is a regular point, there exists a unique Lagrange multiplier vector $\boldsymbol{\nu}^\star \in \mathbb{R}^{n_g}$ such that

$$\boldsymbol{\nabla} f(\mathbf{x}^\star) + \sum_{i \in \mathcal{A}(\mathbf{x}^\star)} \nu_i^\star \boldsymbol{\nabla} g_i(\mathbf{x}^\star) \;=\; \mathbf{0}.$$

Assigning zero Lagrange multipliers to the inactive constraints, we obtain

$$\boldsymbol{\nabla} f(\mathbf{x}^\star) + \boldsymbol{\nabla}\mathbf{g}(\mathbf{x}^\star)^{\mathsf{T}} \boldsymbol{\nu}^\star \;= \mathbf{0} \tag{1.19}$$

$$\nu_i^\star \;= 0, \quad \forall i \notin \mathcal{A}(\mathbf{x}^\star). \tag{1.20}$$

This latter condition can be rewritten by means of the following inequalities

$$\nu_i^\star g_i(\mathbf{x}^\star) = 0 \qquad \forall i = 1, \ldots, n_g.$$

The argument showing that $\boldsymbol{\nu} \geq \mathbf{0}$ is a little more elaborate. By contradiction, assume that $\nu_\ell < 0$ for some $\ell \in \mathcal{A}(\mathbf{x}^\star)$. Let $\mathsf{A} \in \mathbb{R}^{(n_g+1) \times n_x}$ be the matrix whose rows are $\boldsymbol{\nabla} f(\mathbf{x}^\star)$ and $\boldsymbol{\nabla} g_i(\mathbf{x}^\star)$, $i = 1, \ldots, n_g$. Since $\mathbf{x}^\star$ is a regular point, the Lagrange multiplier vector $\boldsymbol{\nu}^\star$ is unique. Therefore, the condition

$$\mathsf{A}^{\mathsf{T}} \mathbf{y} \;=\; \mathbf{0},$$

can only be satisfied by $\mathbf{y}^\star := \eta \begin{pmatrix} 1 & \boldsymbol{\nu}^\star \end{pmatrix}^{\mathsf{T}}$ with $\eta \in \mathbb{R}$. Because $\nu_\ell < 0$, we know by Gordan's Theorem 1.A.78 that there exists a direction $\bar{\mathbf{d}} \in \mathbb{R}^{n_x}$ such that $\mathsf{A}\bar{\mathbf{d}} < \mathbf{0}$. In other words,

$$\bar{\mathbf{d}} \in \mathcal{F}_0(\mathbf{x}^\star) \cap \mathcal{D}_0(\mathbf{x}^\star) \neq \emptyset,$$

which contradicts the hypothesis that $\mathbf{x}^\star$ is a local minimizer of f on S (see Remark 1.34).

Overall, these results thus constitute the KKT optimality conditions as stated in Theorem 1.39. But although the foregoing development is straightforward, it is somewhat limited by the regularity-type assumption at the optimal solution. Obtaining more general constraint qualifications (see Remark 1.42) requires that the KKT conditions be derived using an alternative approach, e.g., the approach described earlier in §1.5. Still, the conversion to equality constrained problem proves useful in many situations, e.g., for deriving second-order sufficiency conditions for inequality constrained NLP problems.

1.7.2 Optimality Conditions for General NLP Problems

We are now ready to generalize the necessary and sufficient conditions given in Theorems 1.39, 1.50, 1.56 and 1.59 to general NLP problems.

Theorem 1.62 (First- and Second-Order Necessary Conditions). *Let* $f : \mathbb{R}^{n_x} \to \mathbb{R}$, $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_g$, *and* $h_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_h$, *be twice continuously differentiable functions on* $\mathbb{R}^{n_x}$. *Consider the problem P to minimize* $f(\mathbf{x})$ *subject to the constraints* $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ *and* $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. *If* $\mathbf{x}^\star$ *is a local minimum of P and is a regular point of the constraints, then there exist unique vectors* $\boldsymbol{\nu}^\star \in \mathbb{R}^{n_g}$ *and* $\boldsymbol{\lambda}^\star \in \mathbb{R}^{n_h}$ *such that*

$$\begin{aligned}
\nabla f(\mathbf{x}^\star) + \nabla \mathbf{g}(\mathbf{x}^\star)^\mathsf{T} \boldsymbol{\nu}^\star + \nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T} \boldsymbol{\lambda}^\star &= \mathbf{0} && (1.21)\\
\boldsymbol{\nu}^\star &\geq \mathbf{0} && (1.22)\\
\mathbf{g}(\mathbf{x}^\star) &\leq \mathbf{0} && (1.23)\\
\mathbf{h}(\mathbf{x}^\star) &= \mathbf{0} && (1.24)\\
\boldsymbol{\nu}^{\star\mathsf{T}} \mathbf{g}(\mathbf{x}^\star) &= 0, && (1.25)
\end{aligned}$$

and

$$\mathbf{y}^\mathsf{T} \left(\nabla^2 f(\mathbf{x}^\star) + \nabla^2 \mathbf{g}(\mathbf{x}^\star)^\mathsf{T} \boldsymbol{\nu}^\star + \nabla^2 \mathbf{h}(\mathbf{x}^\star)^\mathsf{T} \boldsymbol{\lambda}^\star \right) \mathbf{y} \geq 0,$$

for all $\mathbf{y}$ *such that* $\nabla g_i(\mathbf{x}^\star)^\mathsf{T} \mathbf{y} = 0$, $i \in \mathcal{A}(\mathbf{x}^\star)$, *and* $\nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T} \mathbf{y} = \mathbf{0}$.

Theorem 1.63 (Second-Order Sufficient Conditions). *Let* $f : \mathbb{R}^{n_x} \to \mathbb{R}$, $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_g$, *and* $h_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, n_h$, *be twice continuously differentiable functions on* $\mathbb{R}^{n_x}$. *Consider the problem P to minimize* $f(\mathbf{x})$ *subject to the constraints* $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ *and* $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. *If there exists* $\mathbf{x}^\star$, $\boldsymbol{\nu}^\star$ *and* $\boldsymbol{\lambda}^\star$ *satisfying the KKT conditions (1.21–1.25), and*

$$\mathbf{y}^\mathsf{T} \nabla^2_{\mathbf{xx}} \mathcal{L}(\mathbf{x}^\star, \boldsymbol{\nu}^\star, \boldsymbol{\lambda}^\star) \mathbf{y} > 0$$

for all $\mathbf{y} \neq \mathbf{0}$ *such that*

$$\begin{aligned}
\nabla g_i(\mathbf{x}^\star)^\mathsf{T} \mathbf{y} &= 0 \qquad i \in \mathcal{A}(\mathbf{x}^\star) \text{ with } \nu_i^\star > 0\\
\nabla g_i(\mathbf{x}^\star)^\mathsf{T} \mathbf{y} &\leq 0 \qquad i \in \mathcal{A}(\mathbf{x}^\star) \text{ with } \nu_i^\star = 0\\
\nabla \mathbf{h}(\mathbf{x}^\star)^\mathsf{T} \mathbf{y} = 0&,
\end{aligned}$$

where $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\nu}^\mathsf{T} \mathbf{g}(\mathbf{x}) + \boldsymbol{\lambda}^\mathsf{T} \mathbf{h}(\mathbf{x})$, *then* $\mathbf{x}^\star$ *is a strict local minimum of P.*

Likewise, the KKT sufficient conditions given in Theorem 1.44 for convex, inequality constrained problems can be generalized to general convex problems as follows:

Theorem 1.64 (KKT sufficient Conditions). *Let* $f : \mathbb{R}^{n_x} \to \mathbb{R}$ *and* $g_i : \mathbb{R}^{n_x} \to \mathbb{R}$*,* $i = 1, \ldots, n_g$*, be convex and differentiable functions. Let also* $h_i : \mathbb{R}^{n_x} \to \mathbb{R}$*,* $i = 1, \ldots, n_h$*, be affine functions. Consider the problem to minimize* $f(\mathbf{x})$ *subject to* $\mathbf{x} \in S := \{\mathbf{x} \in \mathbb{R}^{n_x} : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$*. If* $(\mathbf{x}^\star, \boldsymbol{\nu}^\star, \boldsymbol{\lambda}^\star)$ *satisfies the KKT conditions (1.21–1.25), then* $\mathbf{x}^\star$ *is a global minimizer for* f *on* S*.*

1.8 NUMERICAL METHODS FOR NONLINEAR PROGRAMMING PROBLEMS

Nowadays, strong and efficient mathematical programming techniques are available for solving a great variety of nonlinear problems, which are based on solid theoretical results and extensive numerical studies. Approximated functions, derivatives and optimal solutions can be employed together with optimization algorithms to reduce the computational time.

The aim of this section is *not* to describe state-of-the-art algorithms in nonlinear programming, but to explain, in a simple way, a number of modern algorithms for solving nonlinear problems. These techniques are typically *iterative* in the sense that, given an initial point $\mathbf{x}^0$, a sequence of points, $\{\mathbf{x}^k\}$, is obtained by repeated application of an algorithmic rule. The objective is to make this sequence converge to a point $\bar{\mathbf{x}}$ in a pre-specified *solution set*. Typically, the solution set is specified in terms of optimality conditions, such as those presented in §1.4 through §1.7.

We start by recalling a number of concepts in §1.8.1. Then, we discuss the principles of Newton-like algorithms for nonlinear systems in §1.8.2, and use these concepts for the solution of unconstrained optimization problems in §1.8.3. Finally, algorithms for solving general, nonlinear problems are presented in §1.8.4, with emphasizes on sequential unconstrained minimization (SUM) and sequential quadratic programming (SQP) techniques.

1.8.1 Preliminaries

Two essential questions must be addressed concerning iterative algorithms. The first question, which is qualitative in nature, is whether a given algorithm in some sense yields, at least in the limit, a solution to the original problem; the second question, the more quantitative one, is concerned with how fast the algorithm converges to a solution. We elaborate on these concepts in this subsection.

The convergence of an algorithm is said to *asymptotic* when the solution is not achieved after a finite number of iterations; except for particular problems such as linear and quadratic programming, this is generally the case in nonlinear programming. That is, a very desirable property of an optimization algorithm is global convergence:

Definition 1.65 (Global Convergence, Local Convergence). *An algorithm is said to be* globally convergent *if, for any initial point* $\mathbf{x}^0$*, it generates a sequence of points that converges to a point* $\bar{\mathbf{x}}$ *in the solution set. It is said to be* locally convergent *if there exists a* $\varrho > 0$ *such that for any initial point* $\mathbf{x}^0$ *such that* $\|\bar{\mathbf{x}} - \mathbf{x}^0\| < \varrho$*, it generates a sequence of points converging to* $\bar{\mathbf{x}}$ *in the solution set.*

Most modern mathematical programming algorithms are globally convergent. Locally convergent algorithms are not useful in practice because the neighborhood of convergence is not known in advance and can be arbitrarily small.

Next, what distinguishes optimization algorithms with the global convergence property is the order of convergence:

Definition 1.66 (Order of Convergence). *The* order of convergence *of a sequence* $\{\mathbf{x}^k\} \rightarrow \bar{\mathbf{x}}$ *is the largest nonnegative integer* p *such that*

$$\lim_{k\to\infty} \frac{\|\mathbf{x}^{k+1} - \bar{\mathbf{x}}\|}{\|\mathbf{x}^k - \bar{\mathbf{x}}\|^p} = \beta < \infty,$$

When $p = 1$ *and the convergence ratio* $\beta < 1$*, the convergence is said to be* linear*; if* $\beta = 0$*, the convergence is said to be* superlinear. *When* $p = 2$*, the convergence is said to be* quadratic.

Since they involve the limit when $k \rightarrow \infty$, p and β are a measure of the *asymptotic* rate of convergence, i.e., as the iterates gets close to the solution; yet, a sequence with a good order of convergence may be very "slow" far from the solution. Clearly, the convergence is faster when p is larger and β is smaller. Near the solution, if the convergence rate is linear, then the error is multiplied by β at each iteration. The error reduction is squared for quadratic convergence, i.e., each iteration roughly doubles the number of significant digits. The methods that will be studied hereafter have convergence rates varying between linear and quadratic.

Example 1.67. Consider the problem to minimize $f(x) = x^2$, subject to $x \geq 1$.

Let the (point-to-point) algorithmic map $\mathcal{M}_1$ be defined defined as $\mathcal{M}_1(x) = \frac{1}{2}(x+1)$. It is easily verified that the sequence obtained by applying the map $\mathcal{M}_1$, with any starting point, converges to the optimal solution $x^\star = 1$, i.e., $\mathcal{M}_1$ is globally convergent. For examples, with $x^0 = 4$, the algorithm generates the sequence $\{4, 2.5, 1.75, 1.375, 1.1875, \ldots\}$. We have $(x^{k+1} - 1) = \frac{1}{2}(x^k - 1)$, so that the limit in Definition 1.66 is $\beta = \frac{1}{2}$ with $p = 1$; moreover, for $p > 1$, this limit is infinity. Consequently, $x^k \rightarrow 1$ linearly with convergence ratio $\frac{1}{2}$.

On the other hand, consider the (point-to-point) algorithmic map $\mathcal{M}_2$ be defined defined as $\mathcal{M}_2(x; k) = 1 + \frac{1}{2^{k+1}}(x-1)$. Again, the sequence obtained by applying $\mathcal{M}_2$ converges to $x^\star = 1$, from any starting point. However, we now have $\frac{|x^{k+1}-1|}{|x^k-1|} = \frac{1}{2^k}$, which approaches 0 as $k \rightarrow \infty$. Hence, $x^k \rightarrow 1$ superlinearly in this case. With $x^0 = 4$, the algorithm generates the sequence $\{4, 2.5, 1.375, 1.046875, \ldots\}$.

The algorithmic maps $\mathcal{M}_1$ and $\mathcal{M}_2$ are illustrated on the left and right plots in Fig 1.14., respectively.

It should also be noted that for most algorithms, the user must set initial values for certain parameters, such as the starting point and the initial step size, as well as parameters for terminating the algorithm. Optimization procedures are often quite sensitive to these parameters, and may produce different results, or even stop prematurely, depending on their values. Therefore, it is crucial for the user to understand the principles of the algorithms used, so that he or she can select adequate values for the parameters and diagnose the reasons of a premature termination (failure).

1.8.2 Newton-like Algorithms for nonlinear Systems

The fundamental approach to most iterative schemes was suggested over 300 years ago by Newton. In fact, Newton's method is the basis for nearly all the algorithms that are described herein.

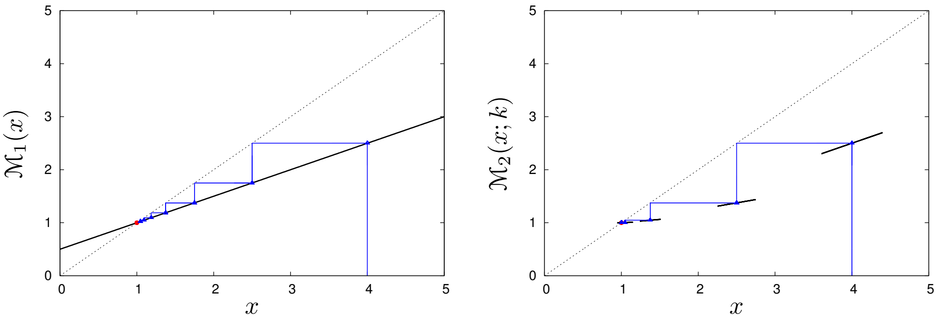


Figure 1.14. Illustration of algorithmic maps $\mathcal{M}_1$ and $\mathcal{M}_2$ in Example 1.67, with $x^0 = 4$.

Suppose one wants to find the value of the variable $\mathbf{x} \in \mathrm{IR}^{n_x}$ such that

$$\boldsymbol{\phi}(\mathbf{x}) \ = \ \mathbf{0},$$

where $\boldsymbol{\phi} : \mathrm{IR}^{n_x} \to \mathrm{IR}^{n_x}$ is continuously differentiable. Let us assume that one such solution exists, and denote it by $\mathbf{x}^\star$. Let also $\mathbf{x}$ be a guess for the solution. The basic idea of Newton's method is to approximate the nonlinear function $\boldsymbol{\phi}$ by the first two terms in its Taylor series expansion about the current point $\mathbf{x}$. This yields a linear approximation for the vector function $\boldsymbol{\phi}$ at the new point $\bar{\mathbf{x}}$,

$$\boldsymbol{\phi}(\bar{\mathbf{x}}) \ = \ \boldsymbol{\phi}(\mathbf{x}) + \boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x}) \left[\bar{\mathbf{x}} - \mathbf{x}\right]. \tag{1.26}$$

Using this linear approximation, and provided that the Jacobian matrix $\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x})$ is nonsingular, a new estimate for the solution $\mathbf{x}^\star$ can be computed by solving (1.26) such that $\boldsymbol{\phi}(\bar{\mathbf{x}}) = \mathbf{0}$, i.e.,

$$\bar{\mathbf{x}} \ = \ \mathbf{x} - \left[\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x})\right]^{-1} \boldsymbol{\phi}(\mathbf{x}).$$

Letting $\mathbf{d} := -\left[\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x})\right]^{-1} \boldsymbol{\phi}(\mathbf{x})$, we get the update $\bar{\mathbf{x}} = \mathbf{x} + \mathbf{d}$.

Of course, $\boldsymbol{\phi}$ being a nonlinear function, one cannot expect that $\boldsymbol{\phi}(\bar{\mathbf{x}}) = \mathbf{0}$, but there is much hope that $\bar{\mathbf{x}}$ be a better estimate for the root $\mathbf{x}^\star$ than the original guess $\mathbf{x}$. In other words, we might expect that

$$|\bar{\mathbf{x}} - \mathbf{x}^\star| \leq |\mathbf{x} - \mathbf{x}^\star| \quad \text{and} \quad |\boldsymbol{\phi}(\bar{\mathbf{x}})| \leq |\boldsymbol{\phi}(\mathbf{x})|.$$

If the new point is an improvement, then it makes sense to repeat the process, thereby defining a sequence of points $\mathbf{x}^0, \mathbf{x}^1, \ldots$. An algorithm implementing Newton's method is as follows:

Initialization Step

Let $\varepsilon > 0$ be a termination scalar, and choose an initial point $\mathbf{x}^0$. Let $k = 0$ and go to the main step.

Main Step

1. Solve the linear system $\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x}^k)\mathbf{d}^k = -\boldsymbol{\phi}(\mathbf{x}^k)$ for $\mathbf{d}^k$.
2. Compute the new estimate $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}^k$.
3. If $\|\boldsymbol{\phi}(\mathbf{x}^{k+1})\| < \varepsilon$, stop; otherwise, replace $k \leftarrow k+1$, and go to step 1.

It can be shown that the rate of convergence for Newton's method is quadratic (see Definition 1.66). Loosely speaking, it implies that each successive estimate of the solution *doubles* the number significant digits, which is a very desirable property for an algorithm to possess.

Theorem 1.68. *Let $\boldsymbol{\phi} : \mathrm{IR}^{n_x} \to \mathrm{IR}^{n_x}$ be continuously differentiable, and consider Newton's algorithm defined by the map $\mathcal{M}(\mathbf{x}) := \mathbf{x} - \boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x})^{-1}\boldsymbol{\phi}(\mathbf{x})$. Let $\mathbf{x}^\star$ be such that $\boldsymbol{\phi}(\mathbf{x}^\star) = \mathbf{0}$, and suppose that $\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x}^\star)$ is nonsingular. Let the starting point $\mathbf{x}^0$ be sufficiently close to $\mathbf{x}^\star$, so that there exist $c_1, c_2 > 0$ with $c_1 c_2 \|\mathbf{x}^0 - \mathbf{x}^\star\| < 1$, and*

$$\begin{aligned} \|\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x})^{-1}\| \; &\leq c_1 \\ \|\boldsymbol{\phi}(\mathbf{x}^\star) - \boldsymbol{\phi}(\mathbf{x}) - \boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x})\,[\mathbf{x}^\star - \mathbf{x}]\,\| \; &\leq c_2\|\mathbf{x}^\star - \mathbf{x}\|^2, \end{aligned} \tag{1.27}$$

for each $\mathbf{x}$ satisfying $\|\mathbf{x}^\star - \mathbf{x}\| \leq \|\mathbf{x}^\star - \mathbf{x}^0\|$. Then, Newton's algorithm converges with a quadratic rate of convergence.

Proof. See [6, Theorem 8.6.5] for a proof. □

But can anything go wrong with Newton's method?

Lack of Global Convergence First and foremost, if the initial guess is not sufficiently close to the solution, i.e., within the *region of convergence*, Newton's method may diverge. Said differently, Newton's method as presented above does *not* have the global convergence property (see Definition 1.65 and Example 1.69 hereafter). This is because $\mathbf{d}^k := \boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x}^k)^{-1}\boldsymbol{\phi}(\mathbf{x}^k)$ may not be a valid descent direction far from the solution, and even if $\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x}^k)\mathbf{d}^k < 0$, a unit step size might not give a descent in $\boldsymbol{\phi}$. Globalization strategies, which aim at correcting this latter deficiency, will be presented in §1.8.3.1 in the context of unconstrained optimization.

Singular Jacobian Matrix A second difficulty occurs when the Jacobian matrix $\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x}^k)$ becomes singular during the iteration process, since the correction defined by (1.8.2) is not defined in this case. Note that the assumption (1.27) in Theorem 1.68 guarantees that $\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x}^k)$ cannot be singular. But when the Jacobian matrix is singular at the solution point $\mathbf{x}^\star$, then Newton's method looses its quadratic convergence property.

Computational Efficiency Finally, at each iteration, Newton's method requires (i) that the Jacobian matrix $\boldsymbol{\nabla}\boldsymbol{\phi}(\mathbf{x}^k)$ be computed, which may be difficult and/or costly especially when the expression of $\boldsymbol{\phi}(\mathbf{x})$ is complicated, and (ii) that a linear system be solved. The analytic Jacobian can be replaced by a finite-difference approximation, yet this is costly as n_x additional evaluations of $\boldsymbol{\phi}$ are required at each iterations. With the objective of reducing the computational effort, *quasi-Newton* methods generate an approximation of the Jacobian matrix, based on the information gathered from the iteration progress. To avoid solving a linear system for the search direction, variants of quasi-Newton methods also exist that generate an approximation of the inverse of the Jacobian matrix. Such methods will be described in §1.8.3.2 in the context of unconstrained optimization.

Example 1.69. Consider the problem to find a solution to the nonlinear equation

$$f(x) \;=\; \arctan(x) \;=\; 0.$$

The Newton iteration sequence obtained by starting from $x^0 = 1$ is as follows:

k	x^k	$\lvert f(x^k)\rvert$
0	1	0.785398
1	-0.570796	0.518669
2	0.116860	0.116332
3	-1.061022×10^{-3}	1.061022×10^{-3}
4	7.963096×10^{-10}	7.963096×10^{-10}

Notice the very fast convergence to the solution $x^\star = 0$, as could be expected from Theorem 1.68. The first three iterations are represented in Fig. 1.15., on the left plot.

However, convergence is not guaranteed for any initial guess. More precisely, it can be shown that Newton's method actually diverges when the initial guess is chosen such that $|x^0| > \alpha$, with $\alpha \approx 1.3917452002707$ being a solution of $\arctan(z) = \frac{2z}{1+z^2}$; further, the method cycles indefinitely for $|x^0| = \alpha$. Both these situations are illustrated in the right plot and the bottom plot of Fig. 1.15., respectively.

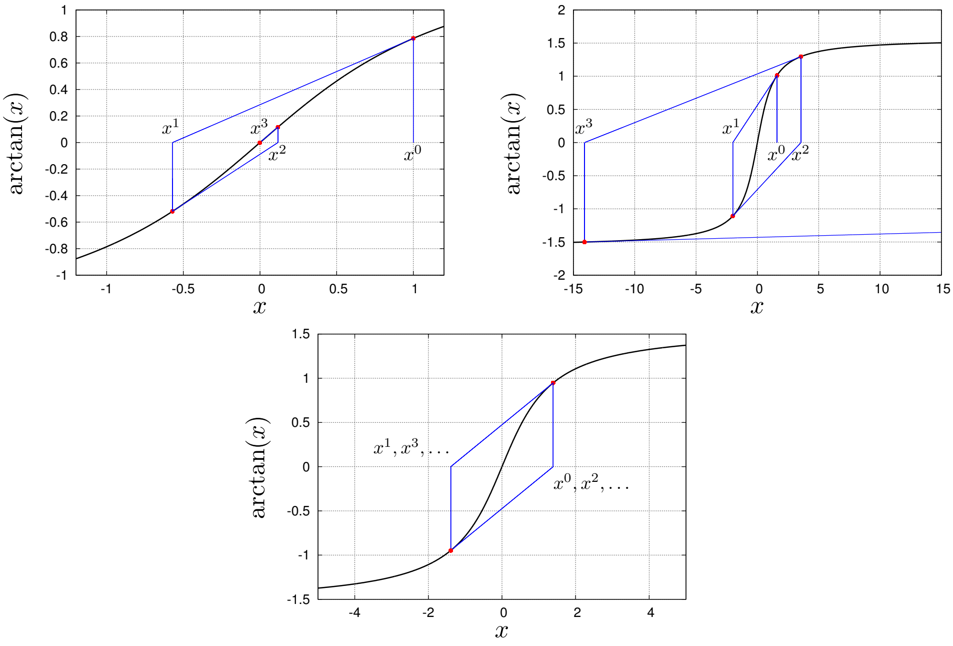


Figure 1.15. Illustration of Newton's algorithm in Example 1.69. Left plot: convergent iterates; right plot: divergent iterates; bottom plot: iterates cycle indefinitely.

1.8.3 Unconstrained Optimization

We now turn to a description of basic techniques used for iteratively solving unconstrained problems of the form:

$$\text{minimize: } f(\mathbf{x}); \quad \mathbf{x} \in \mathbb{R}^{n_x}.$$

Many unconstrained optimization algorithms work along the same lines. Starting from an initial point, a direction of movement is determined according to a fixed rule, and then a move is made in that direction so that the objective function value is reduced; at the new point, a new direction is determined and the process is repeated. The main difference between these algorithms rest with the rule by which successive directions of movement are selected. A distinction is usually made between those algorithms which determine the search direction without using gradient information (*gradient-free* methods), and those using gradient (and higher-order derivatives) information (*gradient-based* methods). Here, we shall focus our attention on the latter class of methods, and more specifically on Newton-like algorithms.

Throughout this subsection, we shall assume that the objective function f is twice continuously differentiable. By Theorem 1.22, a necessary condition for $\mathbf{x}^\star$ to be a local minimum of f is $\nabla f(\mathbf{x}^\star) = \mathbf{0}$. Hence, the idea is to devise an iterative scheme that finds a point satisfying the foregoing condition. Following the techniques discussed earlier in §1.8.2, this can be done by using a Newton-like algorithm, with ϕ corresponding to the gradient ∇f of f, and $\nabla\phi$ to its Hessian matrix H.

At each iteration, a new iterate $\mathbf{x}^{k+1}$ is obtained such that the linear approximation to the gradient at that point is zero,

$$\nabla \mathbf{f}(\mathbf{x}^{k+1}) \;=\; \nabla \mathbf{f}(\mathbf{x}^k) + \mathsf{H}(\mathbf{x}^k)\left[\mathbf{x}^{k+1} - \mathbf{x}^k\right] \;=\; \mathbf{0}.$$

The linear approximation yields the Newton search direction

$$\mathbf{d}^k \;:=\; \mathbf{x}^{k+1} - \mathbf{x}^k \;=\; -[\mathsf{H}(\mathbf{x}^k)]^{-1}\nabla f(\mathbf{x}^k). \tag{1.28}$$

As discussed in §1.8.2, if it converges, Newton's method exhibits a quadratic rate of convergence when the Hessian matrix H is nonsingular at the solution point. However, since the Newton iteration is based on finding a zero of the gradient vector, there is no guarantee that the step will move towards a local minimum, rather than a saddle point or even a maximum. To preclude this, the Newton steps should be taken *downhill*, i.e., the following descent condition should be satisfied at each iteration,

$$\nabla f(\mathbf{x}^k)^{\mathsf{T}}\mathbf{d}^k \;<\; 0.$$

Interestingly enough, with the Newton direction (1.28), the descent condition becomes

$$\nabla f(\mathbf{x}^k)^{\mathsf{T}}\mathsf{H}(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^k) \;>\; 0.$$

That is, a sufficient condition to obtain a descent direction at $\mathbf{x}^k$ is that the Hessian matrix $\mathsf{H}(\mathbf{x}^k)$ be positive definite. Moreover, if $\mathsf{H}(\mathbf{x}^\star)$ is positive definite at a local minimizer $\mathbf{x}^\star$ of f, then the Newton iteration converges to $\mathbf{x}^\star$ when started sufficiently close to $\mathbf{x}^\star$. (Recall that, by Theorem 1.28, positive definiteness of $\mathsf{H}(\mathbf{x}^\star)$ is a sufficient condition for a local minimum of f to be a strict local minimum.)

We now discuss two important improvements to Newton's method, which are directly related to the issues discussed in §1.8.2, namely (i) the lack of global convergence, and (ii) computational efficiency.

1.8.3.1 Globalization Strategies Up to this point, the development has focused on the application of Newton's method. However, even in the simplest one-dimensional applications, Newton's method has deficiencies (see, e.g., Example 1.69). Methods for correcting global convergence deficiencies are referred to as *globalization* strategies. It should be stressed than an efficient globalization strategy should only alter the iterates when a problem is encountered, but it should *not* impede the ultimate behavior of the method, i.e., the quadratic convergence of a Newton's method should be retained.

In unconstrained optimization, one can detect problems in a very simple fashion, by monitoring whether the next iterate $\mathbf{x}^{k+1}$ satisfies a descent condition with respect to the actual iterate $\mathbf{x}^k$, e.g., $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$. Then, either one of two globalization strategies can be used to correct the Newton step. The first strategy, known as *line search* method, is to alter the *magnitude* of the step; the second one, known as *trust region* method, is to modify both the step magnitude and *direction*. We shall only concentrate on the former class of globalization strategies subsequently.

A line search method proceeds by replacing the *full* Newton step $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}^k$ with

$$\mathbf{x}^{k+1} \ = \ \mathbf{x}^k + \alpha \mathbf{d}^k,$$

where the step-length $\alpha \geq 0$ is chosen such that the objective function is reduced,

$$f(\mathbf{x}^k + \alpha \mathbf{d}^k) \ < \ f(\mathbf{x}^k).$$

Clearly, the resulting minimization problem can be solved by any one-dimensional *exact* minimization technique (e.g., Newton's method itself). However, such techniques are costly in the sense that they often require many iterations to converge and, therefore, many function (or even gradient) evaluations.

In response to this, most modern algorithms implement so-called *inexact line search criteria*, which aim to find a step-length α giving an "acceptable" decrease in the objective function. Note that sacrificing accuracy, we might impair the convergence of the overall algorithm that iteratively employs such a line search. However, by adopting a line search that guarantees a sufficient degree of descent in the objective function, the convergence of the overall algorithm can still be established.

We now describe one popular definition of an acceptable step-length know as *Armijo's rule*; other popular approaches are the *quadratic* and *cubic fit* techniques, as well as Wolfe's and Glodstein's tests. Armijo's rule is driven by two parameters $0 < \kappa_1 < 1$ and $\kappa_2 > 1$, which respectively manage the acceptable step-length from being too large or too small. (Typical vales are $\kappa_1 = 0.2$ and $\kappa_2 = 2$.) Define the line search function $\ell(\alpha) := f(\mathbf{x}^k + \alpha \mathbf{d}^k)$, for $\alpha \geq 0$, and consider the modified first-order approximation $\hat{\ell}(\alpha) := \ell(0) + \kappa_1 \alpha \ell'(0)$. A step-length $\bar{\alpha} \in (0, 1)$ is deemed acceptable if the following conditions hold:

$$\ell(\bar{\alpha}) \ \leq \hat{\ell}(\bar{\alpha}) \tag{1.29}$$

$$\ell(\kappa_2 \bar{\alpha}) \ \geq \hat{\ell}(\kappa_2 \bar{\alpha}). \tag{1.30}$$

The condition (1.29) prevents the step-length $\bar{\alpha}$ from being too large, whereas the condition (1.30) prevents $\bar{\alpha}$ from being too small. The acceptable region defined by the Armijo's rule is shown in Fig. 1.16. below.

1.8.3.2 Recursive Updates Another limitation of Newton's method when applied to unconstrained optimization problems is that the Hessian matrix of the objective function is needed at each iteration, then a linear system must be solved for obtaining the search direction. For many applications, this can be a costly computational burden. In response to this, *quasi-Newton* methods attempt to construct this information recursively. However, by so doing, the quadratic rate of convergence is lost.

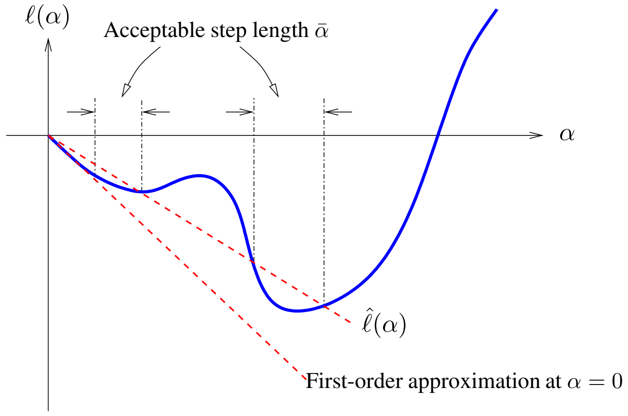

$\ell(\alpha)$
Acceptable step length $\bar{\alpha}$
α
$\hat{\ell}(\alpha)$
First-order approximation at $\alpha = 0$

Figure 1.16. Illustration of Armijo's rule

The basic idea for many quasi-Newton methods is that two successive iterates $\mathbf{x}^k$, $\mathbf{x}^{k+1}$, together with the corresponding gradients $\nabla f(\mathbf{x}^k)$, $\nabla f(\mathbf{x}^{k+1})$, yield curvature information by means of the first-order approximation relation

$$\nabla f(\mathbf{x}^{k+1}) \;=\; \nabla f(\mathbf{x}^k) + \mathsf{H}(\mathbf{x}^k)\boldsymbol{\delta}^{\mathbf{k}} + \text{h.o.t.},$$

with $\boldsymbol{\delta}^k := \mathbf{x}^{k+1} - \mathbf{x}^k$. In particular, given n_x linearly independent iteration increments $\boldsymbol{\delta}^0, \ldots, \boldsymbol{\delta}^{n_x-1}$, an approximation of the Hessian matrix can be obtained as

$$\mathsf{H}(\mathbf{x}^{n_x}) \;\approx\; \left[\begin{array}{ccc} \boldsymbol{\gamma}^0 & \cdots & \boldsymbol{\gamma}^{n_x-1} \end{array}\right]\left[\begin{array}{ccc} \boldsymbol{\delta}^{\mathbf{0}} & \cdots & \boldsymbol{\delta}^{n_x-1} \end{array}\right]^{-1},$$

or for the inverse Hessian matrix as

$$\mathsf{H}(\mathbf{x}^{n_x})^{-1} \;\approx\; \left[\begin{array}{ccc} \boldsymbol{\delta}^{\mathbf{0}} & \cdots & \boldsymbol{\delta}^{n_x-1} \end{array}\right]\left[\begin{array}{ccc} \boldsymbol{\gamma}^0 & \cdots & \boldsymbol{\gamma}^{n_x-1} \end{array}\right]^{-1},$$

where $\boldsymbol{\gamma}^k := \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)$.

Note that when the objective function is quadratic, the previous relations are exact. Many interesting quasi-Newton methods use similar ways, although more sophisticated, to construct an approximate Hessian matrix B^k that progressively approaches the inverse Hessian. One of the most popular class of quasi-Newton methods (known as the *Broyden family*) proceeds as follows:

$$\begin{aligned}\mathsf{B}^{k+1} \;&:= \mathsf{B}^k + \frac{\boldsymbol{\delta}^k {\boldsymbol{\delta}^k}^{\mathsf{T}}}{{\boldsymbol{\delta}^k}^{\mathsf{T}} \boldsymbol{\gamma}^k} - \frac{\mathsf{B}^k \boldsymbol{\gamma}^k {\boldsymbol{\gamma}^k}^{\mathsf{T}} \mathsf{B}^k}{{\boldsymbol{\gamma}^k}^{\mathsf{T}} \mathsf{B}^k \boldsymbol{\gamma}^k} \\ &\quad + \xi {\boldsymbol{\gamma}^k}^{\mathsf{T}} \mathsf{B}^k \boldsymbol{\gamma}^k \left(\frac{\boldsymbol{\delta}^k}{{\boldsymbol{\delta}^k}^{\mathsf{T}} \boldsymbol{\gamma}^k} - \frac{\mathsf{B}^k \boldsymbol{\gamma}^k}{{\boldsymbol{\gamma}^k}^{\mathsf{T}} \mathsf{B}^k \boldsymbol{\gamma}^k}\right)\left(\frac{\boldsymbol{\delta}^k}{{\boldsymbol{\delta}^k}^{\mathsf{T}} \boldsymbol{\gamma}^k} - \frac{\mathsf{B}^k \boldsymbol{\gamma}^k}{{\boldsymbol{\gamma}^k}^{\mathsf{T}} \mathsf{B}^k \boldsymbol{\gamma}^k}\right)^{\mathsf{T}},\end{aligned} \tag{1.31}$$

where $0 \leq \xi \leq 1$. It is easily seen that when supplemented with a line search strategy, ${\mathbf{d}^k}^{\mathsf{T}} \boldsymbol{\gamma}^k < 0$ at each k, and hence the Hessian matrix approximations are guaranteed to exist. Moreover, it can be shown that the successive approximates remain are positive-definite provided that B^0 is itself positive-definite.

By setting $\xi = 0$, (1.31) yields the *Davidon-Fletcher-Powell (DFP) method*, which is historically the first quasi-Newton method, while setting $\xi = 1$ gives the *Broyden-Fletcher-Goldfard-Shanno (BFGS) method*, for which there is substantial evidence that it is the best general purpose quasi-Newton method currently known.

1.8.3.3 Summary A Newton-like algorithm including both a line search method (Armijo's rule) and Hessian recursive update (DFP update) is as follows:

Initialization Step

Let $\varepsilon > 0$ be a termination scalar, and choose an initial point $\mathbf{x}^0 \in \mathrm{I\!R}^{n_x}$ and a symmetric, definite positive matrix $\mathsf{B}^0 \in \mathrm{I\!R}^{n_x \times n_x}$. Let $k = 0$, and go to the main step.

Main Step

1. *Search Direction* – Obtain the search direction from $\mathbf{d}^k = -\mathsf{B}^k \boldsymbol{\nabla} f(\mathbf{x}^k)$.
2. *Line Search* – Find a step α^k satisfying Armijo's conditions (1.29,1.30).
3. *Update* – Compute the new estimates

$$\begin{aligned} \mathbf{x}^{k+1} &:= \mathbf{x}^k + \alpha^k \mathbf{d}^k \\ \mathsf{B}^{k+1} &:= \mathsf{B}^k + \frac{\boldsymbol{\delta}^k {\boldsymbol{\delta}^k}^{\mathsf{T}}}{{\boldsymbol{\delta}^k}^{\mathsf{T}} \boldsymbol{\gamma}^k} - \frac{\mathsf{B}^k \boldsymbol{\gamma}^k {\boldsymbol{\gamma}^k}^{\mathsf{T}} \mathsf{B}^k}{{\boldsymbol{\gamma}^k}^{\mathsf{T}} \mathsf{B}^k \boldsymbol{\gamma}^k}, \end{aligned}$$

   with $\boldsymbol{\delta}^k := \mathbf{x}^{k+1} - \mathbf{x}^k$ and $\boldsymbol{\gamma}^k := \boldsymbol{\nabla} f(\mathbf{x}^{k+1}) - \boldsymbol{\nabla} f(\mathbf{x}^k)$.
4. If $\|\boldsymbol{\nabla} f(\mathbf{x}^{k+1})\| < \varepsilon$, stop; otherwise, replace $k \leftarrow k + 1$, and go to step 1.

The standard unconstrained optimization algorithm in the version 3.0.2 of the Optimization Toolbox in MATLAB® is an implementation of quasi-Newton's method, with DFP or BFGS update, and a line search strategy. (See MATLAB® help pages for more information about the algorithm and the function `fminunc`.)

Example 1.70. Consider the problem to find a minimum to Rosenbrock's function

$$f(x) \;=\; (1 - x_1)^2 + c\left(x_2 - x_1^2\right)^2,$$

for $\mathbf{x} \in \mathrm{I\!R}^2$, with $c := 105$. We solved this problem using the function `fminunc` of the Optimization Toolbox in MATLAB®. The M-files are as follows:

```
clear all;
x0 = [ 5; 5 ];
options = optimset('GradObj', 'on', 'Display', 'iter', ...
                   'DerivativeCheck', 'on', 'LargeScale', 'off', ...
                   'HessUpdate', 'bfgs', 'Diagnostics', 'on', ...
                   'LineSearchType', 'cubicpoly', 'tolX', 1e-10, ...
                   'tolFun', 1e-10)
c = 105;
[xopt, fopt, iout] = fminunc( @(x) exm1(x,c), x0, options );
```

```
%%%%%%%%%%%%%%% FUNCTION TO BE MINIMIZED %%%%%%%%%%%%%%%
% ROSENBROCK FUNCTION: f(x,y) := (1-x)^2+c*(y-x^2)^2
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [f,g] = exm1(x,c)
  f = (1-x(1))^2 + c*(x(2)-x(1)^2)^2; % function
  if nargout > 1
    g = [ -2*(1-x(1)) + 2*c*(x(2)-x(1)^2)*(-2*x(1)) % gradient
          2*c*(x(2)-x(1)^2) ];
  end
end
```

The results are shown in Fig. 1.17. Observe the slow convergence of the iterates far from the optimal solution $\mathbf{x}^\star = (1, 1)$, but the very fast convergence in the vicinity of $\mathbf{x}^\star$.

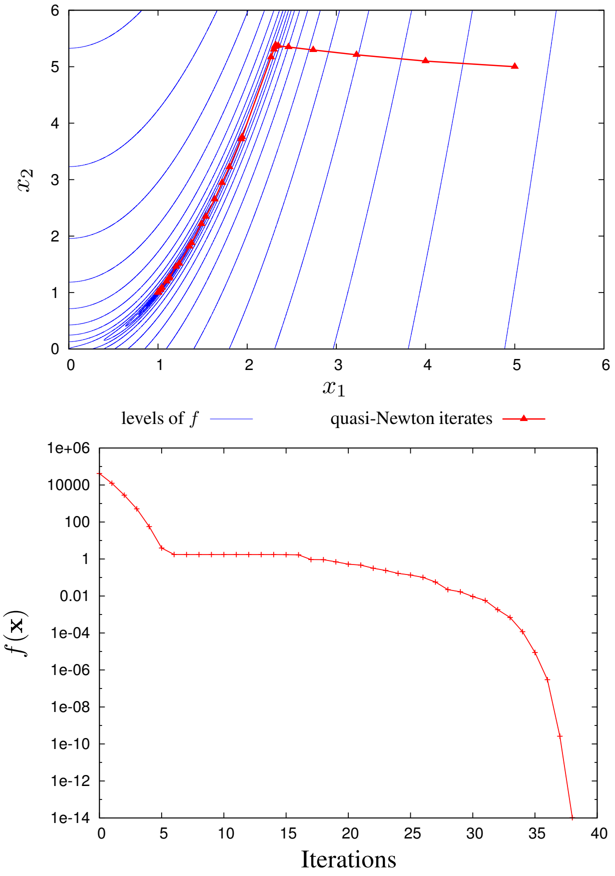


Figure 1.17. Illustration of quasi-Newton's algorithm for Rosenbrock's function in Example 1.70.

1.8.4 Constrained Nonlinear Optimization

In this subsection, we turn our attention to algorithms for iteratively solving constrained problems of the form:

$$\text{minimize: } f(\mathbf{x}); \quad \mathbf{x} \in X \subset \mathbb{R}^{n_x}.$$

Many modern deterministic algorithms for constrained NLP problems are based on the (rather natural) principle that, instead of solving a difficult problem directly, one had better solve a sequence of simpler, but related, subproblems, which converges to a solution of the original problem either in a finite number of steps or in the limit. Working along these lines, two classes of algorithms can be distinguished for solution of NLP problems with equality and/or inequality constraints. On the one hand, *penalty function* and *interior-point methods* consist of solving the problem as a sequence of unconstrained problems (or problems with simple constraints), so that algorithms for unconstrained optimization can be used. These methods, which do not rely on the KKT theory described earlier in §1.5 through §1.7, shall be briefly presented in §1.8.4.1 and §1.8.4.2. On the other hand, *Newton-like methods* solve NLP problems by attempting to find a point satisfying the necessary conditions of optimality (KKT conditions in general). Successive quadratic programming (SQP), which shall be presented in §1.8.4.3, represents one such class of methods.

1.8.4.1 Penalty Function Methods Methods using penalty functions transform a constrained problem into a single unconstrained problem or a sequence of unconstrained problems. This is done by placing the constraints into the objective function via a penalty parameter in a way that penalizes any violation of the constraints. To illustrate it, consider the NLP problem

$$\begin{array}{rl} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{x} \in X \end{array} \tag{1.32}$$

where X is a subset of $\mathbb{R}^{n_x}$, $\mathbf{x}$ is a vector of n_x components $x_1, \ldots, x_{n_x}$, and $f : X \to \mathbb{R}$, $\mathbf{g} : X \to \mathbb{R}^{n_g}$ and $\mathbf{h} : X \to \mathbb{R}^{n_h}$ are defined on X.

In general, a suitable penalty function $\alpha(\mathbf{x})$ for problem (1.32) is defined by

$$\alpha(\mathbf{x}) \;=\; \sum_{k=1}^{n_g} \phi[g_k(\mathbf{x})] + \sum_{k=1}^{n_h} \psi[h_k(\mathbf{x})], \tag{1.33}$$

where ϕ and ψ are continuous functions satisfying the conditions

$$\left\{ \begin{array}{ll} \phi(z) = 0 & \text{if } z \leq 0 \\ \phi(z) > 0 & \text{otherwise} \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{ll} \psi(z) = 0 & \text{if } z = 0 \\ \psi(z) > 0 & \text{otherwise} \end{array} \right. \tag{1.34}$$

Typically, ϕ and ψ are of the forms

$$\phi(z) \;=\; (\max\{0, z\})^p \quad \text{and} \quad \psi(z) \;=\; |z|^p,$$

with p a positive integer (taking $p \geq 2$ provides continuously differentiable penalty functions). The function $f(\mathbf{x}) + \mu\alpha(\mathbf{x})$ is referred to as the *auxiliary function*.

Example 1.71. Consider the problem to minimize $f(x) = x$, subject to $g(x) = -x+2 \leq 0$. It is immediately evident that the optimal solution lies at the point $x^\star = 2$, and has objective value $f(x^\star) = 2$.

Now, consider the penalty problem to minimize $f(x) + \mu\alpha(x) = x + \mu \max\{0, 2-x\}^2$ in $\mathbb{R}$, where μ is a large number. Note first that for any μ, the auxiliary function is convex. Thus, a necessary and sufficient condition for optimality is that the gradient of $f(x)+\mu\alpha(x)$

be equal to zero, yielding $x^\mu = 2 - \frac{1}{2\mu}$. Thus, the solution of the penalty problem can be made arbitrarily close to the solution of the original problem by choosing μ sufficiently large. Moreover, $f(x^\mu) + \mu\alpha(x^\mu) = 2 - \frac{1}{4\mu}$, which can also be made arbitrarily close to $f(x^\star)$ by taking μ sufficiently large. These considerations are illustrated in Fig. 1.18. below.

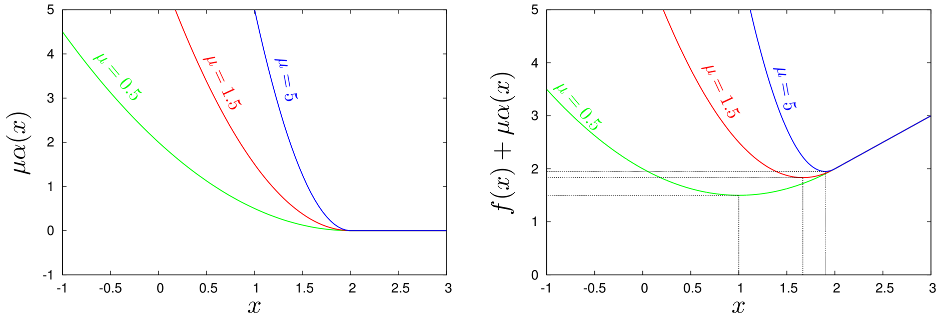


Figure 1.18. Illustration of the penalty (left plot) and auxiliary (right plot) functions in Example 1.71.

The conclusions of Example 1.71 that the solution of the penalty problem can be made arbitrarily close to the solution of the original problem, and the optimal auxiliary function value arbitrarily close to the optimal objective value, by choosing μ sufficiently large, is formalized in the following:

Theorem 1.72. *Consider the NLP problem* (1.32)*, where f, $\mathbf{g}$ and $\mathbf{h}$ are continuous functions on $\mathbb{R}^{n_x}$ and X is a nonempty convex set in $\mathbb{R}^{n_x}$. Suppose that* (1.32) *has a feasible solution, and let α be a continuous function given by (1.33,1.34). Suppose further that for each μ, there exists a solution $\mathbf{x}^\mu \in X$ to the problem* $\min\{f(\mathbf{x}) + \mu\alpha(\mathbf{x}) : \mathbf{x} \in X\}$*, and that $\{\mathbf{x}^\mu\}$ is contained in a compact subset of X. Then,*

$$\min\{f(\mathbf{x}) : \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in X\} \;=\; \sup_{\mu \geq 0}\, \theta(\mu) \;=\; \lim_{k \to \infty}\, \theta(\mu),$$

with $\theta(\mu) := f(\mathbf{x}^\mu) + \mu\alpha(\mathbf{x}^\mu)$. Furthermore, the limit $\bar{\mathbf{x}}$ of any convergent subsequence of $\{\mathbf{x}^\mu\}$ is an optimal solution to the original problem and $\mu\alpha(\mathbf{x}^\mu) \to 0$ as $\mu \to \infty$.

Proof. See [6, Theorem 9.2.2] for a proof. □

Note that the assumption that X is compact is necessary, for it possible that the optimal objective values of the original and penalty problems are not equal otherwise. Yet, this assumption is not very restrictive in most practical cases as the variables usually lie between finite upper and lower bounds. Note also that no restriction is imposed on f, $\mathbf{g}$ and $\mathbf{h}$ other than continuity. However, the application of an efficient solution procedure for the (unconstrained) auxiliary problems may impose additional restriction on these functions (see §1.8.3).

Under the conditions that (i) f, $\mathbf{g}$, $\mathbf{h}$ in (1.32) and ϕ, ψ in (1.33,1.34) are continuously differentiable, and (ii) $\bar{x}$ is a regular point (see Definitions 1.36 and 1.47), the solution to the penalty problem can be used to recover the Lagrange multipliers associated with the constraints at optimality. In the particular case where $X = \mathbb{R}^{n_x}$, we get

$$\begin{aligned} \nu_i^\mu &= \mu\phi'\left[g_i(\mathbf{x}^\mu)\right] \quad \forall i \in \mathcal{A}(\bar{\mathbf{x}}) && (1.35)\\ \lambda_i^\mu &= \mu\psi'\left[h_i(\mathbf{x}^\mu)\right] \quad \forall i = 1,\ldots,n_h. && (1.36)\end{aligned}$$

The larger μ, the better the approximation of the Lagrange multipliers,

$$\boldsymbol{\nu}^\mu \to \boldsymbol{\nu}^\star \text{ and } \boldsymbol{\lambda}^\mu \to \boldsymbol{\lambda}^\star \text{ as } \mu \to \infty.$$

Example 1.73. Consider the same problem as in Example 1.71. The auxiliary function $f(x) + \mu\alpha(x) = x + \mu\max\{0, 2-x\}^2$ being continuously differentiable, the Lagrange multiplier associated to the inequality constraint $g(x) = -x + 2 \leq 0$ can be recovered as $\nu^\mu = 2\mu\max\{0, 2-x^\mu\} = 1$ (assuming $\mu > 0$). Note that the exact value of the Lagrange multiplier is obtained for each $\mu > 0$ here, because g is a linear constraint.

From a computational viewpoint, superlinear convergence rates might be achievable, in principle, by applying Newton's method (or its variants such as quasi-Newton methods). Yet, one can expect ill-conditioning problems when μ is taken very large in the penalty problem. With a large μ, more emphasis is placed on feasibility, and most procedures for unconstrained optimization will move quickly towards a feasible point. Even though this point may be far from the optimum, both slow convergence and premature termination can occur due to very small step size and finite precision computations (round-off errors).

As a result of the above mentioned difficulties associated with large penalty parameters, most algorithms using penalty functions employ a sequence of increasing penalty parameters. With each new value of the penalty parameter, an optimization technique is employed, starting with the optimal solution corresponding to the previously chosen parameters value. such an approach is often referred to as *sequential unconstrained minimization* (SUM) technique. This way, a sequence of infeasible points is typically generated, whose limit is an optimal solution to the original problem (hence the term *exterior penalty function approach*).

To conclude our discussion on the penalty function approach, we give an algorithm to solve problem (1.32), where the penalty function used is of the form specified in (1.33,1.34).

Initialization Step

Let $\varepsilon > 0$ be a termination scalar, and choose an initial point $\mathbf{x}^0$, a penalty parameter $\mu^0 > 0$, and a scalar $\beta > 1$. Let $k = 0$ and go to the main step.

Main Step

1. Starting with $\mathbf{x}^k$, get a solution to the problem

$$\mathbf{x}^{k+1} \in \arg\min\{f(\mathbf{x}) + \mu^k\alpha(\mathbf{x}) : \mathbf{x} \in X\}$$

2. If $\mu^k\alpha(\mathbf{x}^{k+1}) < \varepsilon$, stop; otherwise, let $\mu^{k+1} = \beta\mu^k$, replace $k \leftarrow k+1$, and go to step 1.

1.8.4.2 Interior-Point Methods Similar to penalty functions, *barrier functions* can also be used to transform a constrained problem into an unconstrained problem (or into a sequence of unconstrained problems). These functions act as a barrier and prevent the iterates from leaving the feasible region. If the optimal solution occurs at the boundary of the feasible domain, the procedure moves from the interior to the boundary of the domain, hence the name *interior-point methods*. To illustrate these methods, consider the NLP problem

$$\begin{array}{rl} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{x} \in X \end{array} \tag{1.37}$$

where X is a subset of $\mathbb{R}^{n_x}$, and $f : X \to \mathbb{R}$, $\mathbf{g} : X \to \mathbb{R}^{n_g}$ are continuous on $\mathbb{R}^{n_x}$. Note that equality constraints, if any, should be accommodated within the set X. (In the case of affine equality constraints, one can possibly eliminate them after solving for some variables in terms of the others, thereby reducing the dimension of the problem.) The reason why this treatment is necessary is because barrier function methods require the set $\{\mathbf{x} \in \mathbb{R}^{n_x} : \mathbf{g}(\mathbf{x}) < 0\}$ to be **nonempty**; this would obviously be not possible if the equality constraints $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ were accommodated within the set of inequalities as $\mathbf{h}(\mathbf{x}) \leq \mathbf{0}$ and $\mathbf{h}(\mathbf{x}) \geq \mathbf{0}$.

A barrier problem formulates as

$$\begin{array}{rl} \min\limits_{\mathbf{x}} & \theta(\mu) \\ \text{s.t.} & \mu \geq 0, \end{array} \tag{1.38}$$

where $\theta(\mu) := \inf\{f(\mathbf{x}) + \mu b(\mathbf{x}) : \mathbf{g}(\mathbf{x}) < \mathbf{0}, \mathbf{x} \in X\}$. Ideally, the barrier function b should take value zero on the region $\{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq 0\}$, and value ∞ on its boundary. This would guarantee that the iterates do not leave the domain $\{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq 0\}$ provided the minimization problem started at an interior point. However, this discontinuity poses serious difficulties for any computational procedure. Therefore, this ideal construction of b is replaced by the more realistic requirement that b be nonnegative and continuous over the region $\{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq 0\}$ and approach infinity as the boundary is approached from the interior:

$$b(\mathbf{x}) = \sum_{k=1}^{n_g} \phi[g_k(\mathbf{x})], \tag{1.39}$$

where ϕ is a continuous function over $\{z : z < 0\}$ that satisfies the conditions

$$\begin{cases} \phi(z) \geq 0 \quad \text{if } z < 0 \\ \lim_{z \to 0^-} \phi(z) = +\infty \ . \end{cases} \tag{1.40}$$

In particular, μb approaches the ideal barrier function described above as μ approaches zero.

Typically barrier functions are

$$b(\mathbf{x}) \ = \ -\sum_{k=1}^{n_g} \frac{1}{g_k(\mathbf{x})} \quad \text{or} \quad b(\mathbf{x}) \ = \ -\sum_{k=1}^{n_g} \ln[\min\{1, -g_k(\mathbf{x})\}].$$

The following barrier function, known as *Frisch's logarithmic barrier function*, is also widely used

$$b(\mathbf{x}) \ = \ -\sum_{k=1}^{n_g} \ln[-g_k(\mathbf{x})].$$

The function $f(\mathbf{x}) + \mu b(\mathbf{x})$ is referred to as the *auxiliary function*.

Given $\mu > 0$, evaluating $\theta(\mu) = \inf\{f(\mathbf{x}) + \mu b(\mathbf{x}) : \mathbf{g}(\mathbf{x}) < 0, \mathbf{x} \in X\}$ seems no simpler than solving the original problem because of the constraint $\mathbf{g}(\mathbf{x}) < 0$. However, starting the optimization from a point in the region $S := \{\mathbf{x} : \mathbf{g}(\mathbf{x}) < 0\} \cap X$ yields an optimal point in S, even when the constraint $\mathbf{g}(\mathbf{x}) < 0$ is ignored. This is because b approaches infinity as the iterates approach the boundary of $\{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq 0\}$ from within S, hence preventing them from leaving the set S. This is formalized in the following:

Theorem 1.74. *Consider the NLP problem* (1.37)*, where f and $\mathbf{g}$ are continuous functions on $\mathbb{R}^{n_x}$ and X is a nonempty convex set in $\mathbb{R}^{n_x}$. Suppose that* (1.37) *has an optimal solution $\mathbf{x}^\star$ with the property that, given any neighborhood $\mathcal{B}_\eta(\mathbf{x}^\star)$ around $\mathbf{x}^\star$, there exists an $\mathbf{x} \in X \cap \mathcal{B}_\eta(\mathbf{x}^\star)$ such that $\mathbf{g}(\mathbf{x}) < \mathbf{0}$. Suppose further that for each μ, there exists a solution $\mathbf{x}^\mu \in X$ to the problem $\min\{f(\mathbf{x}) + \mu b(\mathbf{x}) : \mathbf{x} \in X\}$. Then,*

$$\min\{f(\mathbf{x}) : \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{x} \in X\} \;=\; \lim_{\mu \downarrow 0} \theta(\mu) \;=\; \inf_{\mu > 0} \theta(\mu),$$

with $\theta(\mu) := f(\mathbf{x}^\mu) + \mu b(\mathbf{x}^\mu)$. Furthermore, the limit of any convergent subsequence of $\{\mathbf{x}^\mu\}$ is an optimal solution to the original problem, and $\mu b(\mathbf{x}^\mu) \to 0$ as $\mu \to 0$.

Proof. See [6, Theorem 9.4.3] for a proof. □

Under the conditions that (i) f, $\mathbf{g}$ in (1.32) and ϕ in (1.33,1.34) are continuously differentiable, and (ii) $\bar{x}$ is a regular point (see Definitions 1.36 and 1.47), the solution to the barrier function problem can be used to recover the Lagrange multipliers associated with the constraints at optimality. In the particular case where $X = \mathbb{R}^{n_x}$, we get

$$\nu_i^\mu \;= \mu\phi'\left[g_i(\mathbf{x}^\mu)\right] \quad \forall i \in \mathcal{A}(\bar{\mathbf{x}}). \tag{1.41}$$

The approximation of the Lagrange multipliers, gets better as μ gets closer to 0,

$$\boldsymbol{\nu}^\mu \to \boldsymbol{\nu}^\star \text{ as } \mu \to 0^+.$$

Example 1.75. Consider the problem to minimize $f(x) = x$, subject to $g(x) = -x+2 \leq 0$, the solution of which lies at the point $x^\star = 2$ with objective value $f(x^\star) = 2$.

Now, consider the barrier function problem to minimize $f(x) + \mu b(x) = x - \frac{\mu}{2-x}$ in $\mathbb{R}$, where μ is a large number. Note first that for any μ, the auxiliary function is convex. Thus, a necessary and sufficient condition for optimality is that the gradient of $f(x) + \mu b(x)$ be equal to zero, yielding $x^\mu = 2 + \sqrt{\mu}$ (assuming $\mu > 0$). Thus, the solution of the penalty problem can be made arbitrarily close to the solution of the original problem by choosing μ sufficiently close to zero. Moreover, $f(x^\mu) + \mu b(x^\mu) = 2 - 2\sqrt{\mu}$, which can also be made arbitrarily close to $f(x^\star)$ by taking μ sufficiently close to zero. These considerations are illustrated in Fig. 1.19. below.

Regarding the Lagrange multiplier associated to the inequality constraint $g(x) = -x + 2 \leq 0$, the objective and constraint functions being continuously differentiable, an approximate value can be obtained as $\nu^\mu = \frac{\mu}{2-x^\mu}^2 = 1$. Here again, the exact value of the Lagrange multiplier is obtained for each $\mu > 0$ because g is a linear constraint.

The use of barrier functions for solving constrained NLP problems also faces several computational difficulties. First, the search must start with a point $\mathbf{x} \in X$ such that

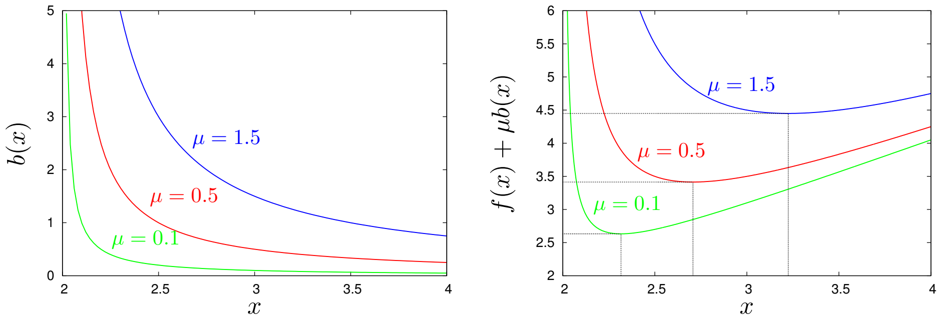


Figure 1.19. Illustration of the barrier (left plot) and auxiliary (right plot) functions in Example 1.75.

$\mathbf{g}(\mathbf{x}) < \mathbf{0}$, and finding such a point may not be an easy task for some problems. Also, because of the structure of the barrier function b, and for small values of the parameter μ, most search techniques may face serious ill-conditioning and difficulties with round-off errors while solving the problem to minimize $f(\mathbf{x}) + \mu b(\mathbf{x})$ over $\mathbf{x} \in X$, especially as the boundary of the region $\{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq 0\}$ is approached. Accordingly, interior-point algorithms employ a sequence of decreasing penalty parameters $\{\mu^k\} \rightarrow 0$ as $k \rightarrow \infty$; with each new value μ^k, an optimal solution to the barrier problem is sought by starting from the previous optimal solution. As in the exterior penalty function approach, it is highly recommended to use suitable second-order Newton or quasi-Newton methods for solving the successive barrier problems.

We describe below a scheme using barrier functions of the form (1.39,1.40) for optimizing a nonlinear programming problem such as (1.37).

Initialization Step

Let $\varepsilon > 0$ be a termination scalar, and choose an initial point $\mathbf{x}^0$ with $\mathbf{g}(\mathbf{x}^0) < 0$. Let $\mu^0 > 0$, $\beta \in (0, 1)$, $k = 0$, and go to the main step.

Main Step

1. Starting with $\mathbf{x}^k$, get a solution to the problem

$$\mathbf{x}^{k+1} \in \arg\min\{f(\mathbf{x}) + \mu^k b(\mathbf{x}) : \mathbf{x} \in X\}$$

2. If $\mu^k b(\mathbf{x}^{k+1}) < \varepsilon$, stop; otherwise, let $\mu^{k+1} = \beta\mu^k$, replace $k \leftarrow k + 1$, and go to step 1.

Note that although the constraint $\mathbf{g}(\mathbf{x}) < 0$ may be ignored, it is considered in the problem formulation as most line search methods use discrete steps, and a step could lead to a point outside the feasible region (where the value of the barrier function is a large negative number), when close to the boundary. Therefore, the problem can effectively be treated as an unconstrained optimization problem only if an explicit check for feasibility is made.

In recent years, there has been much excitement because some variants of the interior-point algorithm can be shown to be polynomial in time for many classes of convex programs. Moreover, interior-point codes are now proving to be highly competitive with codes based on other algorithms, such as SQP algorithms presented subsequently. A number of free and commercial interior-point solvers are given in Tab. 1.1. below.

Table 1.1. A number of open-source and commercial codes implementing interior-point techniques for NLP problems.

Solver	Website	Licensing	Characteristics
IPOPT	`projects.coin-or.org/Ipopt`	free, open source	line search, filter, preconditioned CG
LOQO	`www.princeton.edu/~rvdb/`	commercial	primal-dual, direct factorization
KNITRO	`www.ziena.com/knitro.htm`	commercial	trust-region, primal barrier/SQP with primal-dual scaling, direct factorization/preconditioned CG

Note first that there is currently no function implementing interior-point methods for NLP problems in the version 3.0.2 of MATLAB®'s Optimization Toolbox. The solvers listed in Tab. 1.2. are stand-alone (either in C/C++ or fortran77 programming language). However, all can be used through the modeling language AMPL (`http://www.ampl.com/`); KNITRO can also be used in MATLAB® through both the TOMLAB optimization environment (`http://tomopt.com/tomlab/`) and the modeling language GAMS (`http://www.gams.com/`).

1.8.4.3 Successive Quadratic Programming *Successive quadratic programming* (SQP) methods, also known as *sequential*, or *recursive*, quadratic programming, employ Newton's method (or quasi-Newton methods) to directly solve the KKT conditions for the original problem. As a result, the accompanying subproblem turns out to be the minimization of a quadratic approximation to the Lagrangian function subject to a linear approximation to the constraints. Hence, this type of process is also known as a *projected Lagrangian*, or the *Newton-Lagrange*, approach. By its nature, this method produces both primal and dual (Lagrange multiplier) solutions.

Equality Constrained Case. To present the concept of SQP, consider first the nonlinear problem P to

$$\begin{aligned}&\text{minimize: } f(\mathbf{x})\\&\text{subject to: } h_i(\mathbf{x})=0, \quad i=1,\ldots,n_h,\end{aligned}$$

where $\mathbf{x}\in\mathbb{R}^{n_x}$, and f, $\mathbf{h}$ are twice continuously differentiable. We shall also assume throughout that the equality constraints are linearly independent at a solution of P. (The extension for including inequality constraints is considered subsequently.)

By Theorem 1.50, the first-order necessary conditions of optimality for Problem P require a primal solution $\mathbf{x}^\star\in\mathbb{R}^{n_x}$ and a Lagrange multiplier vector $\boldsymbol{\lambda}^\star\in\mathbb{R}^{n_h}$ such that

$$\mathbf{0} = \nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^\star,\boldsymbol{\lambda}^\star) = \nabla f(\mathbf{x}^\star)+\nabla\mathbf{h}(\mathbf{x}^\star)^{\mathsf{T}}\boldsymbol{\lambda}^\star \tag{1.42}$$

$$\mathbf{0} = \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\mathbf{x}^\star,\boldsymbol{\lambda}^\star) = \mathbf{h}(\mathbf{x}^\star), \tag{1.43}$$

where $\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}) := f(\mathbf{x})+\boldsymbol{\lambda}^{\mathsf{T}}\mathbf{h}(\mathbf{x})$. Now, consider a Newton-like method to solve (1.42,1.43). Given an iterate $(\mathbf{x}^k,\boldsymbol{\lambda}^k)$, a new iterate $(\mathbf{x}^{k+1},\boldsymbol{\lambda}^{k+1})$ is obtained by solving the first-order approximation

$$\mathbf{0} = \begin{pmatrix}\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^k,\boldsymbol{\lambda}^k)\\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\mathbf{x}^k,\boldsymbol{\lambda}^k)\end{pmatrix} + \begin{pmatrix}\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^k,\boldsymbol{\lambda}^k) & \nabla\mathbf{h}(\mathbf{x}^k)^{\mathsf{T}}\\ \nabla\mathbf{h}(\mathbf{x}^k) & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{x}^{k+1}-\mathbf{x}^k\\ \boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\end{pmatrix}$$

to (1.42,1.43). Denoting $\mathbf{d}^k := \mathbf{x}^{k+1} - \mathbf{x}^k$, the above linear system can be rewritten as

$$\begin{pmatrix} \nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k) & \nabla\mathbf{h}(\mathbf{x}^k)^{\mathsf{T}} \\ \nabla\mathbf{h}(\mathbf{x}^k) & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{d}^k \\ \boldsymbol{\lambda}^{k+1} \end{pmatrix} = - \begin{pmatrix} \nabla f(\mathbf{x}^k) \\ \mathbf{h}(\mathbf{x}^k) \end{pmatrix}, \tag{1.44}$$

which can be solved for $(\mathbf{d}^k, \boldsymbol{\lambda}^{k+1})$, if a solution exists. Setting $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}^k$, and incrementing k by 1, we can then repeat the process until $\mathbf{d}^k = \mathbf{0}$ happens to solve (1.44). When this occurs, if at all, noting (1.42,1.43), we shall have found a stationary point to Problem P.

Interestingly enough, a *quadratic programming* (QP) minimization subproblem can be employed in lieu of the foregoing linear system to find *any* optimal solution for P,

$$\begin{aligned} &\min_{\mathbf{d}^k} f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^{\mathsf{T}}\mathbf{d}^k + \frac{1}{2}\mathbf{d}^{k\mathsf{T}}\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k)\mathbf{d}^k \qquad (\text{QP}(\mathbf{x}^k, \boldsymbol{\lambda}^k)) \\ &\text{s.t. } h_i(\mathbf{x}) + \nabla h_i(\mathbf{x}^k)^{\mathsf{T}}\mathbf{d}^k = 0 \quad i = 1, \ldots, n_h. \end{aligned}$$

Note in particular that an optimum $\mathbf{d}^k$ to QP$(\mathbf{x}^k, \boldsymbol{\lambda}^k)$, if it exists, together with the set of Lagrange multipliers $\boldsymbol{\lambda}^{k+1}$ associated with the linearized constraints, is a stationary point for QP$(\mathbf{x}^k, \boldsymbol{\lambda}^k)$ and satisfies equations (1.42,1.43). That is, solving QP$(\mathbf{x}^k, \boldsymbol{\lambda}^k)$ is attractive because it tends to drive the solution towards a desirable stationary point satisfying (1.42,1.43) whenever alternatives exist. Assuming a well-behaved QP, a rudimentary SQP algorithm is as follows:

Initialization Step

Choose an initial primal/dual point $(\mathbf{x}^0, \boldsymbol{\lambda}^0)$, let $k = 0$, and go to the main step.

Main Step

1. Solve the quadratic subproblem QP$(\mathbf{x}^k, \boldsymbol{\lambda}^k)$ to obtain a solution $\mathbf{d}^k$ along with a set of Lagrange multipliers $\boldsymbol{\lambda}^{k+1}$.
2. If $\mathbf{d}^k = \mathbf{0}$, then $(\mathbf{d}^k, \boldsymbol{\lambda}^{k+1})$ satisfies the stationarity conditions (1.42,1.43) for problem P; stop. Otherwise, let $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}^k$, replace $k \leftarrow k + 1$, and go to step 1.

In the case $\mathbf{x}^\star$ is a regular stationary solution for Problem P which, together with a set of Lagrange multipliers $\boldsymbol{\lambda}^\star$, satisfies the second-order sufficiency conditions of Theorem 1.59, then the matrix

$$\mathsf{W} := \begin{pmatrix} \nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k) & \nabla\mathbf{h}(\mathbf{x}^k)^{\mathsf{T}} \\ \nabla\mathbf{h}(\mathbf{x}^k) & \mathbf{0} \end{pmatrix},$$

can be shown to be nonsingular. Hence, the above rudimentary SQP algorithm exhibits a quadratic rate of convergence by Theorem 1.68.

Extension to Inequality Constrained Case. We now consider the inclusion of inequality constraints $g_i(\mathbf{x}) \le 0$, $i = 1, \ldots, n_g$, in Problem P,

$$\begin{aligned} \text{minimize: } & f(\mathbf{x}) \\ \text{subject to: } & g_i(\mathbf{x}) \le 0, \quad i = 1, \ldots, n_g \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, n_h, \end{aligned}$$

where $\mathbf{g}$ is twice continuously differentiable.

Given an iterate $(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$, where $\boldsymbol{\lambda}^k$ and $\boldsymbol{\nu}^k \geq \mathbf{0}$ are the Lagrange multiplier estimates for the equality and inequality constraints, respectively, consider the following QP subproblem as a direct extension of QP$(\mathbf{x}^k, \boldsymbol{\lambda}^k)$:

$$\begin{aligned}
\min_{\mathbf{d}^k} \ & f(\mathbf{x}^k) + \boldsymbol{\nabla} f(\mathbf{x}^k)^{\mathsf{T}} \mathbf{d}^k + \frac{1}{2} \mathbf{d}^{k\mathsf{T}} \boldsymbol{\nabla}^2_{\mathbf{xx}} \mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k) \mathbf{d}^k \qquad (\text{QP}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)) \\
\text{s.t. } & g_i(\mathbf{x}) + \boldsymbol{\nabla} g_i(\mathbf{x}^k)^{\mathsf{T}} \mathbf{d}^k = 0 \quad i = 1, \ldots, n_g \\
& h_i(\mathbf{x}) + \boldsymbol{\nabla} h_i(\mathbf{x}^k)^{\mathsf{T}} \mathbf{d}^k = 0 \quad i = 1, \ldots, n_h,
\end{aligned}$$

where $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := f(\mathbf{x}) + \boldsymbol{\nu}^{\mathsf{T}} \mathbf{g}(\mathbf{x}) + \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{h}(\mathbf{x})$. Note that the KKT conditions for QP$(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$ require that, in addition to primal feasibility, Lagrange multipliers $\boldsymbol{\lambda}^{k+1}$, $\boldsymbol{\nu}^{k+1}$ be found such that

$$\begin{aligned}
\boldsymbol{\nabla} f(\mathbf{x}^k) + \boldsymbol{\nabla}^2_{\mathbf{xx}} \mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k) \mathbf{d}^k + \boldsymbol{\nabla} \mathbf{g}(\mathbf{x}^k)^{\mathsf{T}} \boldsymbol{\nu}^{k+1} + \boldsymbol{\nabla} \mathbf{h}(\mathbf{x}^k)^{\mathsf{T}} \boldsymbol{\lambda}^{k+1} &= \mathbf{0} \\
\left[\mathbf{g}(\mathbf{x}^k) + \boldsymbol{\nabla} \mathbf{g}(\mathbf{x}^k)^{\mathsf{T}} \mathbf{d}^k \right]^{\mathsf{T}} \boldsymbol{\nu}^{k+1} &= \mathbf{0},
\end{aligned}$$

with $\boldsymbol{\nu}^{k+1} \geq \mathbf{0}$ and $\boldsymbol{\lambda}^{k+1}$ unrestricted in sign. Clearly, if $\mathbf{d}^k = \mathbf{0}$, then $\mathbf{x}^k$ together with $\boldsymbol{\lambda}^{k+1}$, $\boldsymbol{\nu}^{k+1}$ yields a KKT solution to the original problem P. Otherwise, we set $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}^k$ as before, increment k by 1, and repeat the process. Regarding convergence rate, it can be shown that if $\mathbf{x}^\star$ is a regular KKT solution which, together with $\boldsymbol{\lambda}^\star$, $\boldsymbol{\nu}^\star$ satisfies the second-order sufficient conditions of Theorem 1.63, and if $(\mathbf{x}^0, \boldsymbol{\lambda}^0, \boldsymbol{\nu}^0)$ is initialized sufficiently close to $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$, then the foregoing iterative procedure shall exhibit a *quadratic* convergence rate.

An Improved SQP Algorithm. The SQP method, as presented thus far, obviously shares the disadvantages of Newton's method: (i) it requires second-order derivatives $\boldsymbol{\nabla}^2_{\mathbf{xx}} \mathcal{L}$ to be calculated, which in addition might not be positive definite, and (ii) it lacks the global convergence property.

(i) Regarding second-order derivatives, a quasi-Newton positive definite approximation can be used for $\boldsymbol{\nabla}^2_{\mathbf{xx}} \mathcal{L}$. For example, given a positive definite approximation B^k for $\boldsymbol{\nabla}^2_{\mathbf{xx}} \mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$, the quadratic problem QP$(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$ can be solved with $\boldsymbol{\nabla}^2_{\mathbf{xx}} \mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$ replaced by B^k. For example, an approximation of the inverse Hessian matrix can be obtained via a Broyden-like procedure (1.31), with γ^k given by

$$\gamma^k := \boldsymbol{\nabla} \mathcal{L}(\mathbf{x}^{k+1}, \boldsymbol{\lambda}^{k+1}, \boldsymbol{\nu}^{k+1}) - \boldsymbol{\nabla} \mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^{k+1}, \boldsymbol{\nu}^{k+1}),$$

as explained earlier in §1.8.3.2. It can be shown that this modification to the rudimentary SQP algorithm, similar to the quasi-Newton modification of Newton' algorithm, looses the quadratic convergence rate property. Instead, it can be shown that the convergence is superlinear when initialized sufficiently close to a solution $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$ that satisfies both regularity and second-order sufficiency conditions. However, this superlinear convergence rate is strongly based on the use of unit step sizes (see point (ii) below).

(ii) In order to remedy the global convergence deficiency, a globalization strategy can be used, e.g., a line search procedure (see §1.8.3.1). Unlike unconstrained optimization problems, however, the choice of a suitable line search (or *merit*) function providing a measure of progress is not obvious in the presence of constraints. Two such popular choices of a line search function are

- The ℓ_1 *Merit Function:*

$$\ell_1(\mathbf{x};\mu) \ := f(\mathbf{x}) + \mu \left[\sum_{i=1}^{n_h} |h_i(\mathbf{x})| + \sum_{i=1}^{n_g} \max\{0, g_i(\mathbf{x})\}\right], \qquad (1.45)$$

which satisfies the important property that $\mathbf{x}^\star$ is a local minimizer of ℓ_1, provided $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$ satisfies the second-order sufficient conditions (see Theorem 1.63) and the penalty parameter μ is so chosen that $\mu > |\lambda_i^\star|$, $i = 1, \ldots, n_h$, and $\mu > \nu_i^\star$, $i = 1, \ldots, n_g$. Yet, the ℓ_1 merit function is not differentiable at those $\mathbf{x}$ with either $g_i(\mathbf{x}) = 0$ or $h_i(\mathbf{x}) = 0$, and it can be unbounded below even though $\mathbf{x}^\star$ is a local minimizer.

- The *Augmented Lagrangian (ALAG) Merit Function:*

$$\ell_2(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}; \mu) \ := f(\mathbf{x}) + \sum_{i=1}^{n_h} \lambda_i h_i(\mathbf{x}) + \frac{\mu}{2} \sum_{i=1}^{n_h} [h_i(\mathbf{x})]^2 + \frac{1}{2} \sum_{i=1}^{n_g} \psi_i(\mathbf{x}, \boldsymbol{\nu}; \mu) \qquad (1.46)$$

with $\psi_i(\mathbf{x}, \boldsymbol{\nu}; \mu) := \frac{1}{\mu}\left(\max\{0, \nu_i + \mu g_i(\mathbf{x})\}^2 - \nu_i^2\right)$, has similar properties to the ℓ_1 merit function, provided μ is chosen large enough, and is continuously differentiable (although its Hessian matrix is discontinuous). Yet, for $\mathbf{x}^\star$ to be a (local) minimizer of $\ell_2(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}; \mu)$, it is necessary that $\boldsymbol{\lambda} = \boldsymbol{\lambda}^\star$ and $\boldsymbol{\nu} = \boldsymbol{\nu}^\star$.

An SQP algorithm including the modifications discussed in (i) and (ii) is as follows:

Initialization Step

Choose an initial primal/dual point $(\mathbf{x}^0, \boldsymbol{\lambda}^0, \boldsymbol{\nu}^0)$, with $\boldsymbol{\nu}^0 \geq \mathbf{0}$, and a positive definite matrix B^0. Let $k = 0$, and go to the main step.

Main Step

1. Solve the quadratic subproblem QP$(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$, with $\boldsymbol{\nabla}^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$ replaced by B^k, to obtain a direction $\mathbf{d}^k$ along with a set of Lagrange multipliers $(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\nu}^{k+1})$.
2. If $\mathbf{d}^k = \mathbf{0}$, then $(\mathbf{x}^k, \boldsymbol{\lambda}^{k+1}, \boldsymbol{\nu}^{k+1})$ satisfies the KKT conditions for problem P; stop.
3. Find $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha^k \mathbf{d}^k$, where α^k improves $\ell_1(\mathbf{x}^k + \alpha\mathbf{d}^k)$ over $\{\alpha \in \mathrm{I\!R} : \alpha > 0\}$ [or any other suitable merit functions]. Update B^k to a positive definite matrix B^{k+1} [e.g., according to the quasi-Newton update scheme (1.31)]. Replace $k \leftarrow k + 1$, and go to step 1.

A number of free and commercial interior-point solvers is given in Tab 1.2. below.

Note first that `fmincon` is the function implementing SQP in the version 3.0.2 of MATLAB®'s Optimization Toolbox. The other solvers listed in Tab. 1.2. are stand-alone (either in C/C++ or fortran77 programming language). However, NLPQL, SNOPT and filterSQP can be used in MATLAB® through the TOMLAB optimization environment

Table 1.2. A number of open-source and commercial codes implementing SQP techniques.

Solver	Website	Licensing	Characteristics
fmincon	`http://www.mathworks.com/access/helpdesk/help/toolbox/optim/`	commercial	line search, active set, dense problems
NLPQL	`http://www.uni-bayreuth.de/departments/math/~kschittkowski/nlpqlp22.htm`	commercial	line search, active set, dense problems
RFSQP	`http://www.aemdesign.com/RFSQPwhatis.htm`	free for acad.	line search, active set, feasible SQP, dense problem
SNOPT	`http://www.sbsi-sol-optimize.com/asp/sol_products_snopt_desc.htm`	commercial	line search, active set, reduced Hessian, sparse/large-scale problems
filterSQP	`http://www-unix.mcs.anl.gov/~leyffer/solvers.html`	commercial	trust region, exact Hessian, dense/sparse problems

(`http://tomopt.com/tomlab/`); RFSQP, SNOPT and filterSQP can be used through the AMPL modeling language (`http://www.ampl.com/`); and, finally, SNOPT can be used through the modeling language GAMS (`http://www.gams.com/`).

Example 1.76. Consider the problem to find a solution to the problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^2} \ & f(\mathbf{x}) := x_1^2 + x_2^2 + \log(x_1 x_2) \qquad (1.47)\\ \text{s.t.} \ & g(\mathbf{x}) := 1 - x_1 x_2 \leq 0 \\ & 0 \leq x_1, x_2 \leq 10. \end{aligned}$$

We solved this problem using the function `fmincon` of the Optimization Toolbox in MATLAB®. The M-files are as follows:

```
clear all
x0 = [ 2; 1 ];
xL = [ 0.1; 0.1 ];
xU = [ 10; 10 ];
options = optimset('Display', 'iter', 'GradObj', 'on', ...
                   'GradConstr', 'on', 'DerivativeCheck', 'on', ...
                   'LargeScale', 'off', 'HessUpdate', 'bfgs', ...
                   'Diagnostics', 'on',  'TolX', 1e-7, ...
                   'TolFun', 1e-7, 'TolCon', 1e-7, ...
                   'MaxFunEval', 100, 'MaxIter', 100 )
[xopt, fopt, iout] = fmincon( @SQP_fun, x0, [], [], [], [], xL, xU, ...
                              @SQP_ctr, options );
```

```
%%%%%%%%%%%%%%% FUNCTION TO BE MINIMIZED %%%%%%%%%%%%%%%
% Objective: f(x,y) := x^2+y^2+log(x*y)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [f,df] = SQP_fun(x)
  f = x(1)^2+x(2)^2+log(x(1)*x(2)); % function
  if nargout > 1
```

```
    df = [ 2*x(1)+1/x(1)      % gradient
           2*x(2)+1/x(2) ];
  end
end
```

```
%%%%%%%%%%%%%%%%%%%%%% CONSTRAINTS %%%%%%%%%%%%%%%%%%%%%
% inequality constraint: g(x,y) := x*y
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [g,h,dg,dh] = SQP_ctr(x)
  g = [ 1-x(1)*x(2) ]; % inequality constraints
  h = [];              % equality constraints
  if nargout > 2
    dg = [ -x(2); -x(1) ]; % gradient of inequality constraints
    dh = [];               % gradient of equality constraints
  end
end
```

The results are shown in Fig. 1.20. Notice, the rather fast convergence to the optimal solution $\mathbf{x}^\star = (1,1)$. Note also that the SQP algorithm does not necessarily take a feasible path to reach an optimal solution.

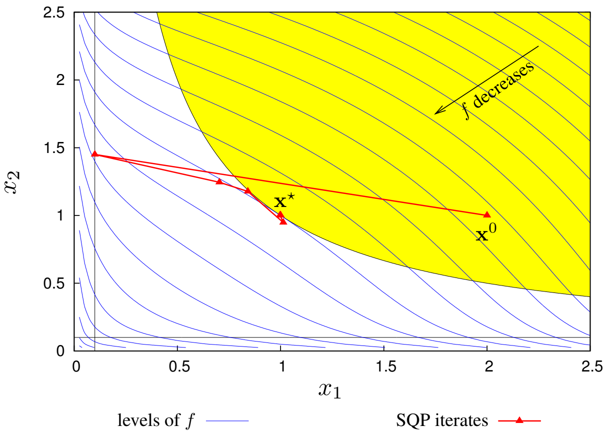


Figure 1.20. SQP iterates for Problem (1.47).

What Can Go Wrong?. The material presented up to this point was intended to give the reader an understanding of how SQP methods should work. Things do not go so smoothly in practice though. We now discuss a number of common difficulties that can be encountered, and suggest remedial actions to correct the deficiencies. Because real applications may involve more than a single difficulty, the user must be prepared to correct *all* problems before obtaining satisfactory performance from an optimization software.

Infeasible Constraints One of the most common difficulties occurs when the NLP problem has infeasible constraints, i.e., the constraints taken all together have *no* solution. Applying general-purpose SQP software to such problems typically produces one or more of the following symptoms:

- one of the QP subproblem happen to be infeasible, which occurs when the linearized constraints have no solution;
- many NLP iterations produce very little progress;
- the penalty parameters μ in (1.45) or (1.46) grows very large; or
- the Lagrange multipliers become very large.

Although robust SQP software attempt to diagnose this situation, ultimately the only remedy is to reformulate the NLP.

Rank-Deficient Constraints In contrast to the previous situation, it is possible that the constraints be consistent, but the Jacobian matrix of the active constraints, at the solution point, be either ill-conditioned or rank deficient. This situation was illustrated in Examples 1.43 and 1.55. The application of general-purpose SQP software is likely to produce the following symptoms:

- many NLP iterations produce very little progress;
- the penalty parameters μ in (1.45) or (1.46) grows very large;
- the Lagrange multipliers become very large; or
- the rank deficiency in the Jacobian of the active constraints is detected.

Note that many symptoms of rank-deficient constraints are the same as those of inconsistent constraints. It is therefore quite common to confuse this deficiency with inconsistent constraints. Again, the remedy is to reformulate the problem.

Redundant Constraints A third type of difficulty occurs when the NLP problem contains redundant constraints. Two types of redundancy may be distinguished. In the first type, some of the constraints are unnecessary to the problem formulation, which typically results in the following symptoms:

- the Lagrange multipliers are close to zero; and
- the solver has difficulty in detecting the active set.

In the second type, the redundant constraints give rise to rank deficiency, and the problem then exhibits symptoms similar to the rank-deficient case discussed previously. Obviously, the remedy is to reformulate the problem by eliminating the redundant constraints.

Discontinuities Perhaps the biggest obstacle encountered in the practical application of SQP methods (as well as many other NLP methods including SUM techniques) is the presence of discontinuous behavior. All of the numerical methods described herein assume continuous and differentiable objective function and constraints. Yet, there are many common examples of discontinuous functions in practice, including: `IF` tests in codes; absolute value, $\min$, and $\max$ functions; linear interpolation of data; internal iterations such as root finding; etc.

Regarding SQP methods, the standard QP subproblems are no longer appropriate when discontinuities are present. In fact, the KKT necessary conditions simply do not apply! The most common symptoms of discontinuous functions are:

- the iterates converge slowly or, even, diverge;
- the line search takes very small steps ($\alpha \approx 0$); and
- the Hessian matrix becomes badly ill-conditioned.

The remedy consists in reformulating discontinuous problems into smooth problems: for absolute value, $\min$, and $\max$ functions, tricks can be used that introduce slack variables and additional constraints; linear data interpolation can be replaced by higher order interpolation schemes that are continuous through second derivatives; internal iterations can also be handled via additional NLP constraints; etc.

Inaccurate Gradient Estimation Any SQP code requires that the user supply the objective function and constraint values, as well as their gradient (and possibly their Hessian too). In general, the user is proposed the option to calculate the gradients via finite differences, e.g.,

$$\boldsymbol{\nabla} f(\mathbf{x}) \;\approx\; \frac{f(\mathbf{x}+\boldsymbol{\delta}\mathbf{x}) - f(\mathbf{x})}{\boldsymbol{\delta}\mathbf{x}}.$$

However, this may cause the problem to stop prematurely. First of all, the choice of the perturbation vector $\boldsymbol{\delta}\mathbf{x}$ is highly non trivial. If too large a value clearly provides inaccurate estimates, too small a value may also result in very bad estimates due to finite arithmetic precision computations. Therefore, one must try to find a trade-off between these two extreme situations. The difficulty stems from the fact that a trade-off may not necessarily exist if the requested accuracy for the gradient is too high. In other word, the error made in the finite-difference approximation of a gradient cannot be made as small as desired. Further, the maximum accuracy that can be achieved with finite difference is both problem dependent (e.g., badly-scaled functions are more problematic than well-scaled functions) *and* machine dependent (e.g., `double precision` computations provides more accurate estimates than `single precision` computations). Typical symptoms of inaccurate gradient estimates in an SQP code are:

- the iterates converge slowly, and the solver may stop prematurely at a suboptimal point (jamming); and
- the line search takes very small steps ($\alpha \approx 0$).

The situation can be understood as follows. Assume that the gradient estimate is contaminated with noise. Then, instead of computing the true value $\nabla\mathcal{L}(\mathbf{x})$, we get $\nabla\mathcal{L}(\mathbf{x}) + \varepsilon$. But since the iteration seeks a point such that $\nabla\mathcal{L}(\mathbf{x}) = \mathbf{0}$, we can expect either a degraded rate of convergence or, worse, no convergence at all, because ultimately the gradient will be dominated by noise.

To avoid these problems, the user should always consider providing the gradients explicitely to the SQP solver, instead of relying on finite-difference estimates. For large-scale problems, this is obviously a time-consuming and error-prone task. In response to this, efficient *algorithmic differentiation* tools (also called *automatic differentiation*) have been developed within the last fifteen years. The idea behind it is that, given a piece of program calculating a number of function values (e.g., in fortran77 or C language), an auxiliary program is generated that calculates the derivatives of these functions. (See, e.g., the book by A. Griewank [24] for a general introduction on the topic.)

Scaling Scaling affects everything! Poor scaling can make a good algorithm behave badly. Scaling changes the convergence rate, termination tests, and numerical conditioning.

The most common way of scaling a problem is by introducing scaled variables of the form

$$\tilde{x}_k \ := \ u_k x_k + r_k,$$

for $k = 1, \ldots, n_x$, with u_k and r_k being scale weights and shifts, respectively. Likewise, the objective function and constraints are commonly scaled using

$$\begin{aligned}\tilde{f} &:= \omega_0 f\\ \tilde{g}_k &:= \omega_k g_k,\end{aligned}$$

for $k = 1, \ldots, n_g$. The idea is to let the optimization algorithm work with the well-scaled quantities in order to improve performance. However, what *well-scaled quantities* mean is hard to define, although conventional wisdom suggests the following hints

- normalize the independent variables to have the same range, e.g., $0 \leq \tilde{x}_k \leq 1$;
- normalize the dependent functions to have the same magnitude, e.g., $\tilde{f} \approx \tilde{g}_1 \approx \ldots \approx \tilde{g}_{n_g} \approx 1$;
- normalize the rows and columns of the Jacobian to be of the same magnitude;
- scale the dependent functions so that the Lagrange multipliers are close to one, e.g., $|\lambda_1| \approx \ldots \approx |\lambda_{n_g}| \approx 1$; etc.

1.9 NOTES AND REFERENCES

The material on convex optimization (§1.3) is taken from the book by Boyd and Vandenberghe [10, Chapters 2 and 3]. Many more details and properties of convex programs can be found in this book, together with algorithms specifically tailored to such problems.

Regarding conditions of optimality, the material presented in §1.4 and §1.5 is mostly a summary of the material in Bazaraa, Sherali and Shetty's book [6, Chap. 4 and 5]. The derivation of necessary and sufficient conditions of optimality for equality constrained problems in §1.6 is inspired from the material in Luenberger's book [36, Chap. 10].

Additional information on the concept of algorithm (§1.8.1) and, more particularly, on the convergence aspects can be found ,e.g., in [6, Chap. 7] and [36, Chap. 6]. Regarding, unconstrained minimization techniques (§1.8.3), many additional details are given in Bertsekas' book [7, Chap. 1], as well as in [6, Chap. 8] and [36, Chap. 7 to 9]. More information on sequential unconstrained minimization algorithms (§1.8.4.1 and 1.8.4.2) can be found in [6, Chap. 9] and [36, Chap. 12]; and on sequential quadratic programming algorithms (§1.8.4.3), in [6, Chap. 10]. Many practical details on the numerical algorithms were also found in Betts' book [8, Chap. 1] and Herskovits' overview article [26].

Finally, material on Lagrangian duality theory and saddle point optimality conditions has been omitted from this chapter. The interested reader in referred to [6, Chap. 6].

Appendix: Technical Lemmas and Alternative Proofs

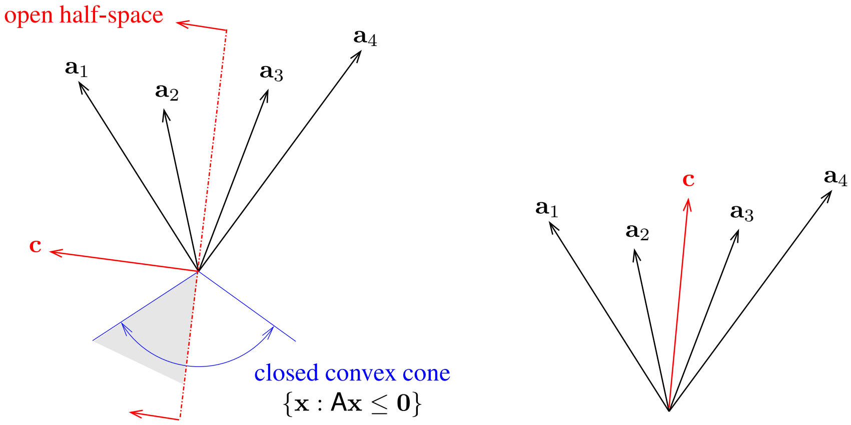
Theorem 1.A.77 (Farkas' Theorem). *Let* A *be an* $m \times n$ *matrix and* $\mathbf{c}$ *be an* n *vector. Then, exactly one of the following two statements holds:*

System 1. $\exists \mathbf{x} \in \mathrm{I\!R}^n$ *such that* $\mathsf{A}\mathbf{x} \leq \mathbf{0}$ *and* $\mathbf{c}^\mathsf{T}\mathbf{x} > 0$*,*

System 2. $\exists \mathbf{y} \in \mathrm{I\!R}^m$ *such that* $\mathsf{A}^\mathsf{T}\mathbf{y} = \mathbf{c}$ *and* $\mathbf{y} \geq \mathbf{0}$.

Proof. See, e.g., [6, Theorem 2.4.5] for a proof. □

Farkas' Theorem is used extensively in the derivation of optimality conditions of (linear and) nonlinear programming problems. A geometrical interpretation of Farkas' Theorem is shown in Fig. 1.A.1.. If $\mathbf{a}_1, \ldots, \mathbf{a}_m$ denote the rows of A, then system 2 has a solution if $\mathbf{c}$ lies in the convex cone generated by $\mathbf{a}_1, \ldots, \mathbf{a}_m$; On the other hand, system 1 has a solution if the closed convex cone $\{\mathbf{x} : \mathsf{A}\mathbf{x} \leq \mathbf{0}\}$ and the open half-space $\{\mathbf{x} : \mathbf{c}^\mathsf{T}\mathbf{x} > \mathbf{0}\}$ have a nonempty intersection.



System 1 has a solution

System 2 has a solution

Figure 1.A.1. Illustration of Farkas' Theorem (with $n = 2$ and $m = 4$).

Corollary 1.A.78 (Gordan's Theorem). *Let* A *be an* $m \times n$ *matrix. Then, exactly one of the following two statements holds:*

System 1. $\exists \mathbf{x} \in \mathrm{I\!R}^n$ *such that* $\mathsf{A}\mathbf{x} < \mathbf{0}$,

System 2. $\exists \mathbf{y} \in \mathrm{I\!R}^m, \mathbf{y} \neq \mathbf{0}$ *such that* $\mathsf{A}^\mathsf{T}\mathbf{y} = \mathbf{0}$ *and* $\mathbf{y} \geq \mathbf{0}$.

Proof. System 1 can be equivalently written as $\mathsf{A}\mathbf{x} + \varrho\mathbf{e}$ where $\varrho > 0$ is a scalar and $\mathbf{e}$ is a vector of m ones. Rewriting this in the form of System 1 in Farkas' Theorem 1.A.77, we get $(\mathsf{A}\ \ \mathbf{e})\mathbf{p}$ and $(0, \ldots, 0, 1)\mathbf{p} > 0$ where $\mathbf{p} := (\mathbf{x}\ \ \varrho)$. The associated System 2 by Farkas' Theorem 1.A.77 states that $(\mathsf{A}\ \ \mathbf{e})^\mathsf{T}(0, \ldots, 0, 1)^\mathsf{T}$ and $\mathbf{y} \geq \mathbf{0}$ for some $\mathbf{y} \in \mathrm{I\!R}^m$, i.e., $\mathsf{A}^\mathsf{T}\mathbf{y} = \mathbf{0}$, $\mathbf{e}^\mathsf{T}\mathbf{y} = 1$ and $\mathbf{y} \geq \mathbf{0}$, which is equivalent to the System 2 of the corollary. □

Below is an alternative proof for Theorem 1.50 on p. 24 that does not use the concept of tangent sets.

Alternative Proof for Theorem 1.50. For $k = 1, 2, \ldots$, let $\varphi^k : \mathrm{I\!R}^{n_x} \to \mathrm{I\!R}$ be the (continuously differentiable) functions defined as:

$$\varphi^k(\mathbf{x}) = f(\mathbf{x}) + \frac{k}{2}\|\mathbf{h}(\mathbf{x})\|^2 + \frac{\alpha}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2,$$

where $\alpha > 0$. Let also $\varepsilon > 0$ be chosen such that:

$$f(\mathbf{x}^\star) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in \bar{\mathcal{B}}_\varepsilon(\mathbf{x}^\star),$$

with $\bar{\mathcal{B}}_\varepsilon(\mathbf{x}^\star) := \{\mathbf{x} \in D : \|\mathbf{x} - \mathbf{x}^\star\| \leq \varepsilon\}$, and denote $\mathbf{x}^k \in \arg\min\{\varphi^k(\mathbf{x}) : \mathbf{x} \in \bar{\mathcal{B}}_\varepsilon(\mathbf{x}^\star)\}$. [6]

Since $\mathbf{x}^\star \in \bar{\mathcal{B}}_\varepsilon(\mathbf{x}^\star)$, we have

$$f(\mathbf{x}^k) + \frac{k}{2}\|\mathbf{h}(\mathbf{x}^k)\|^2 + \frac{\alpha}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2 = \varphi^k(\mathbf{x}^k) \leq \varphi^k(\mathbf{x}^\star) = f(\mathbf{x}^\star) \quad \forall k \geq 1.$$

Hence, $\lim_{k\to\infty}\|\mathbf{h}(\mathbf{x}^k)\| = 0$, so for every limit point $\bar{\mathbf{x}} \in S$ of $\{\mathbf{x}^k\}$, we have $\mathbf{h}(\bar{\mathbf{x}}) = 0$. Moreover, $\mathbf{x}^\star$ being a local solution and $\bar{\mathbf{x}} \in \bar{\mathcal{B}}_\varepsilon(\mathbf{x}^\star)$ being a feasible point,

$$f(\mathbf{x}^\star) \leq f(\bar{\mathbf{x}}). \tag{1.A.1}$$

On the other hand, noting that $f(\mathbf{x}^k) + \frac{\alpha}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2 \leq f(\mathbf{x}^\star)$ for each $k \geq 1$, and taking the limit as $k \to \infty$, we have

$$f(\bar{\mathbf{x}}) + \frac{\alpha}{2}\|\bar{\mathbf{x}} - \mathbf{x}^\star\|^2 \leq f(\mathbf{x}^\star). \tag{1.A.2}$$

Combining (1.A.1) and (1.A.2), we obtain $\|\bar{\mathbf{x}} - \mathbf{x}^\star\| = 0$, so that $\bar{\mathbf{x}} = \mathbf{x}^\star$ and $\lim_{k\to\infty}\|\bar{\mathbf{x}}\| = \mathbf{x}^\star$.

Since $\lim_{k\to\infty}\|\bar{\mathbf{x}}\| = \mathbf{x}^\star$ and $\mathbf{x}^\star \in \text{int}\left(\bar{\mathcal{B}}_\varepsilon(\mathbf{x}^\star)\right)$,

$$\exists K_1 \text{ such that } \mathbf{x}^k \in \text{int}\left(\bar{\mathcal{B}}_\varepsilon(\mathbf{x}^\star)\right) \quad \forall k > K_1,$$

i.e., $\mathbf{x}^k$ is an *unconstrained* minimum of φ^k. By Theorem 1.22 (page 11), we get

$$\mathbf{0} = \boldsymbol{\nabla}\varphi^k(\mathbf{x}^k) = \boldsymbol{\nabla} f(\mathbf{x}^k) + k\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^k)^{\mathsf{T}}\mathbf{h}(\mathbf{x}^k) + \alpha(\mathbf{x}^k - \mathbf{x}^\star) \quad \forall k > K_1. \tag{1.A.3}$$

$\mathbf{x}^\star$ being a regular point for the equality constraints, $\text{rank}\,(\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^\star)) = n_h$. By continuity of $\mathbf{h}$,

$$\exists K_2 \text{ such that } \text{rank}\left(\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^k)\right) = n_h \quad \forall k > K_2.$$

Therefore, $\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^k)\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^k)^{\mathsf{T}}$ is invertible for $k > K_2$, and we have

$$k\mathbf{h}(\mathbf{x}^k) = -\left[\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^k)\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^k)^{\mathsf{T}}\right]^{-1}\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^k)\left[\boldsymbol{\nabla} f(\mathbf{x}^k) + \alpha(\mathbf{x}^k - \mathbf{x}^\star)\right] \quad \forall k > K,$$

where $K = \max\{K_1, K_2\}$. Taking the limit as $k \to \infty$,

$$\lim_{k\to\infty} k\mathbf{h}(\mathbf{x}^k) = \boldsymbol{\lambda}^\star,$$

with $\boldsymbol{\lambda}^\star := -[\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^\star)\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^\star)^{\mathsf{T}}]^{-1}\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^\star)\boldsymbol{\nabla} f(\mathbf{x}^\star)$. Finally, taking the limit as $k \to \infty$ in (1.A.3), we obtain

$$\boldsymbol{\nabla} f(\mathbf{x}^\star) + \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^\star)^{\mathsf{T}}\boldsymbol{\lambda}^\star = \mathbf{0}.$$

□

Lemma 1.A.79. *Let* P *and* Q *be two symmetric matrices, such that* $\mathsf{P} \succeq \mathbf{0}$ *and* $\mathsf{P} \succ \mathbf{0}$ *on the null space of* Q *(i.e.,* $\mathbf{y}^{\mathsf{T}}\mathsf{P}\mathbf{y} > 0, \forall \mathbf{y} \neq \mathbf{0}$ *with* $\mathsf{Q}\mathbf{y} = 0$*). Then,*

$$\exists \bar{c} > 0 \textit{ such that } \mathsf{P} + c\mathsf{Q} \succ \mathbf{0} \quad \forall c > \bar{c}.$$

[6] The minimum $\mathbf{x}^k$ exists because $\bar{\mathcal{B}}_\varepsilon(\mathbf{x}^\star)$ is nonempty, closed and bounded, and φ^k is continuous on $\bar{\mathcal{B}}_\varepsilon(\mathbf{x}^\star)$ — see Theorem 1.14 (page 7).

Proof. Assume the contrary. Then,

$$\forall k > 0, \exists \mathbf{x}^k, \|\mathbf{x}^k\| = 1 \text{ such that } {\mathbf{x}^k}^{\mathsf{T}}\mathsf{P}\mathbf{x}^k + k{\mathbf{x}^k}^{\mathsf{T}}\mathsf{Q}\mathbf{x}^k \preceq \mathbf{0}. \tag{1.A.4}$$

Consider a subsequence $\{\mathbf{x}^k\}_{\mathcal{K}}$ converging to some $\bar{\mathbf{x}}$ with $\|\bar{\mathbf{x}}\| = 1$. Dividing (1.A.4) by k, and taking the limit as $k \in \mathcal{K} \to \infty$, we obtain

$$\bar{\mathbf{x}}^{\mathsf{T}}\mathsf{Q}\bar{\mathbf{x}} \preceq \mathbf{0}.$$

On the other hand, Q being semidefinite positive, we must have

$$\bar{\mathbf{x}}^{\mathsf{T}}\mathsf{Q}\bar{\mathbf{x}} \succeq \mathbf{0},$$

hence $\bar{\mathbf{x}}^{\mathsf{T}}\mathsf{Q}\bar{\mathbf{x}} = \mathbf{0}$. That is, using the hypothesis, $\bar{\mathbf{x}}^{\mathsf{T}}\mathsf{P}\bar{\mathbf{x}} \succ \mathbf{0}$. This contradicts the fact that

$$\bar{\mathbf{x}}^{\mathsf{T}}\mathsf{P}\bar{\mathbf{x}} + \limsup_{k\to\infty, k\in\mathcal{K}} k{\mathbf{x}^k}^{\mathsf{T}}\mathsf{Q}\mathbf{x}^k \preceq \mathbf{0}.$$

□