John Bagterp Jørgensen

# Numerical Methods for Constrained Optimization

Lecture Notes for
02612 Constrained Optimization

February 2, 2021

Springer

# Contents

# Part I
# Introduction

# Chapter 1
# Introduction

Constrained optimization problems are ubiquitous in science, engineering, management and economics. Physical laws are often based on an optimization principle. The equilibrium of a closed chemical system appears when the entropy is maximal. The equilibrium position of a mechanical system appears when the potential energy is minimal. A construction must be build with minimal amount of concrete while satisfying some strength requirements. A factory must be operated such that the operation costs are minimal. The government spending and taxes must be selected to maximize the total utility of society while respecting economic laws. All these problems are examples of constrained optimization problems.

In this book we discuss numerical methods for solution of smooth constrained optimization problems. That means that the functions involved should be sufficiently smooth, i.e. twice continuously differentiable, and the variables involved must be real. Hence, we do not consider integer optimization in this book.

## 1.1 Constrained Optimization Problems

Constrained optimization problem consists of variables, $x \in \mathbb{R}^n$, an objective function, $f : \mathbb{R}^n \mapsto \mathbb{R}$, equality constraint functions $c_i : \mathbb{R}^n \mapsto \mathbb{R}$ for $i \in \mathcal{E}$, and inequality constraint functions $c_i : \mathbb{R}^n \mapsto \mathbb{R}$ for $i \in \mathcal{I}$. For smooth problems, the objective function, $f$, is twice continuously differentiable, i.e. $f \in \mathcal{C}^2$, and all constraint functions, $c_i$ for $i \in \mathcal{E} \cup \mathcal{I}$, are twice continuously differentiable, i.e. $c_i \in \mathcal{C}^2(\mathbb{R}^n)$ for $i \in \mathcal{E} \cup \mathcal{I}$.

The variables $x \in \mathbb{R}^n$ are computed such that $f(x)$ is minimal (or maximal) while respecting the equality constraints, $c_i(x) = 0$ for $i \in \mathcal{E}$, and the inequality constraints, $c_i(x) \geq 0$ for $i \in \mathcal{I}$. A constrained minimization problem is denoted

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) && && (1.1a) \\ s.t. \quad & c_i(x) = 0 && i \in \mathcal{E} && (1.1b) \\ & c_i(x) \geq 0 && i \in \mathcal{I} && (1.1c) \end{aligned}$$

Sometimes it is more convenient to state an optimization problem using the vector function $g : \mathbb{R}^n \mapsto \mathbb{R}^{m_E}$ to represent the equality constraints, and the vector function $h : \mathbb{R}^n \mapsto \mathbb{R}^{m_I}$ to represent the inequality constraints. In this case we state the constrained optimization problem as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) && (1.2a) \\ s.t. \quad & g(x) = 0 && (1.2b) \\ & h(x) \geq 0 && (1.2c) \end{aligned}$$

The best numerical method for solution of a particular constrained optimization problem depends critically on the type of functions representing the objective, the equality constraints, and the inequality constraints. In the following we discuss a number of important optimization problems that we will study.

### 1.1.1 Nonlinear Program (NLP)

Constrained nonlinear optimization problems are also called Nonlinear Programs (NLP). A nonlinear program is denoted

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) && && (1.3a) \\ s.t. \quad & c_i(x) = 0 && i \in \mathcal{E} && (1.3b) \\ & c_i(x) \geq 0 && i \in \mathcal{I} && (1.3c) \end{aligned}$$

in which $f : \mathbb{R}^n \mapsto \mathbb{R}$, $f \in \mathcal{C}^2(\mathbb{R}^n)$, $c_i : \mathbb{R}^n \mapsto \mathbb{R}$ and $c_i \in \mathcal{C}^2(\mathbb{R}^n)$ for $i \in \mathcal{E} \cup \mathcal{I}$.

The feasible set of the nonlinear program (1.3) is the set of points that satisfies the constraints. The feasible set is also called the feasible region and is denoted

$$\Omega = \{x \in \mathbb{R}^n : \quad c_i(x) = 0, \, i \in \mathcal{E}, \quad c_i(x) \geq 0, \, i \in \mathcal{I}\} \tag{1.4}$$

With this notation we can state the nonlinear program (1.3) compactly as

$$\min_{x \in \Omega} f(x) \tag{1.5}$$

A solution to (1.5) and (1.3) is called a minimizer. The value of the objective function of a minimizer is called the minimum value. A point $x^* \in \mathbb{R}^n$ is a global minimizer if it is feasible, $x^* \in \Omega$, and its value is less than the value of all other feasible points, i.e. $f(x) \geq f(x^*)$ for all $x \in \Omega$.

Consequently, we may define a global minimizer, $x^*$, as

$$f(x) \geq f(x^*) \quad \forall x \in \Omega,\, x^* \in \Omega \tag{1.6}$$

For convenience, we always state the constrained optimization problem (1.3) and (1.5) as a minimization problem. A maximization problem, $\max_{x\in\Omega} F(x)$, may be converted to a minimization problem using

$$\max_{x\in\Omega} F(x) \quad = \quad -\min_{x\in\Omega}(-F(x)) \quad = \quad -\min_{x\in\Omega} f(x) \tag{1.7}$$

with $f(x) = -F(x)$.

Also for notational convenience and without loss of generality, the inequality constraints are stated as larger than constraints, i.e. $c_i(x) \geq 0$ for $i \in \mathcal{I}$. Less than inequality constrains, $d_i(x) \leq 0$ for $i \in \mathcal{I}_l$, can be converted to larger than inequality constraints upon multiplication by $-1$

$$d_i(x) \leq 0 \quad \Leftrightarrow \quad -d_i(x) \geq 0 \quad \Leftrightarrow \quad c_i(x) \geq 0,\, c_i(x) = -d_i(x) \quad i \in \mathcal{I}_l \tag{1.8}$$

**Example 1 (Nonlinear Program).** Consider the two-dimensional nonlinear program

$$\min_{(x_1,x_2)\in\mathbb{R}^2} \quad f(x_1,x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{1.9a}$$

$$s.t. \quad c_1(x_1,x_2) = (x_1+2)^2 - x_2 \geq 0 \tag{1.9b}$$

$$c_2(x_1,x_2) = -4x_1 + 10x_2 \geq 0 \tag{1.9c}$$

This NLP has no equality constraints and two inequality constraints. As it is a two-dimensional problem, i.e. $x \in \mathbb{R}^2$, we can easily visualize this NLP using contour plots. Two dimensional problems are useful to illustrate algorithms as they can visualized.

The contour plot of the objective function, $f(x)$, is shown in Figure 1.1. The lines in this figure are points $(x_1, x_2)$ that have the same objective function value. The non-shaded region is the set of feasible points

$$\Omega = \left\{(x_1,x_2) \in \mathbb{R}^2 :\; c_1(x) = (x_1+2)^2 - x_2 \geq 0,\; c_2(x) = -4x_1 + 10x_2 \geq 0\right\}$$

The minimizer is the feasible point, $x = (x_1, x_2) \in \Omega$, i.e. a point in the non-shaded region for which the objective function $f(x)$ attain the smallest values. The blue dots are all the local minima for the NLP (1.9). The global minimizer is located in $x^* = (3, 2)$.

The red points mark saddle points and local maximizers. Most optimization algorithms compute a point that satisfied the so-called first order optimality conditions. These conditions are satisfied by local and global minimizers as well as saddle points, local maximizers, and global maximizers.

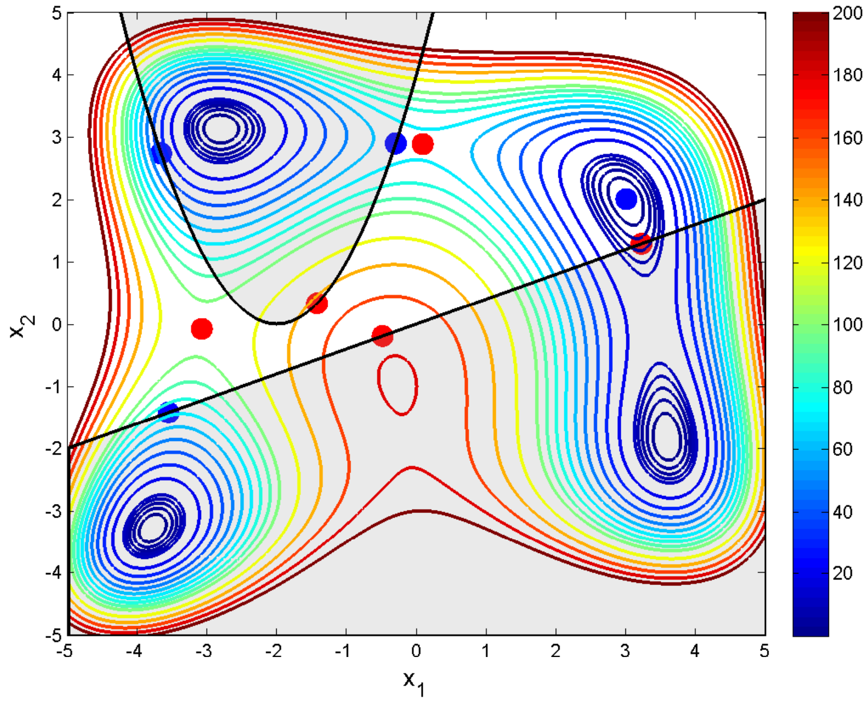



**Fig. 1.1** Contour plot of (1.9). The blue marks are local minima, while the red marks are either local maxima or saddle points.

As is obvious in this example with several local minima, maxima and saddle points, optimization algorithms based on first order optimality conditions cannot guarantee location of a minimum. They might as well find a local maximum or a saddle point. Even if a minimum is found, these algorithms can only guarantee that the minimum is a local minimum. This is one of the major limitations of algorithms for solution of general (non-convex) constrained optimization problems.

■

### *1.1.2 Convex Program*

Consider the constrained optimization problem

$$\min_{x \in \Omega} f(x) \tag{1.10}$$

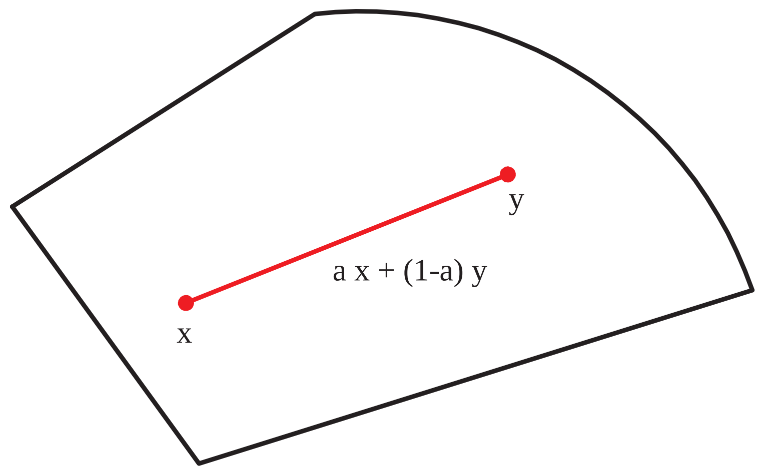



**Fig. 1.2** Convex set. All points in a line segment connecting two points in the convex set are also in the convex set.

in which the feasible region $\Omega$ is a convex set and the objective function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a convex function that is twice continuously differentiable. This is called a convex program. A convex program is different from a general (non-convex) NLP (1.3) in the sense that the first order optimality conditions are both necessary and sufficient for a minimizer. Consequently, if an optimization algorithm compute a point satisfying the first order conditions for a convex program (1.10), then the point is a global minimizer. It cannot be a maximizer, a saddle point, or just a local minimizer. However, the minimizer satisfying the first order optimality conditions is not necessarily unique. There may be several points satisfying the first order optimality conditions and attaining the same function value. If the objective function is *strictly convex*, the point satisfying the first order optimality conditions of convex program is a *unique global minimizer*.

A convex set, $\mathcal{C} \subset \mathbb{R}^n$, is defined as a set that satisfies

$$\forall x, y \in \mathcal{C} : \quad \alpha x + (1-\alpha)y \in \mathcal{C} \quad \forall \alpha \in [0,\, 1] \tag{1.11}$$

This means that every convex combination, $\alpha x + (1-\alpha)y$ for $\alpha \in [0,\, 1]$, of points, $x$ and $y$, in the convex set $\mathcal{C}$ belongs to the convex set itself. A convex set is illustrated in Figure 1.2. a convex set is a set in which every point of the line connecting two points in the set is also in the convex set

A convex function $f : \mathcal{C} \mapsto \mathbb{R}$ is defined as a function that satisfies

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) \quad \forall x, y \in \mathbb{R}^n,\ \forall \alpha \in [0,\, 1] \tag{1.12}$$

Figure 1.3 illustrates a convex function. The *epigraph* of a function is the area above the function value for every $x \in \mathcal{C}$. Hence the epigraph of a

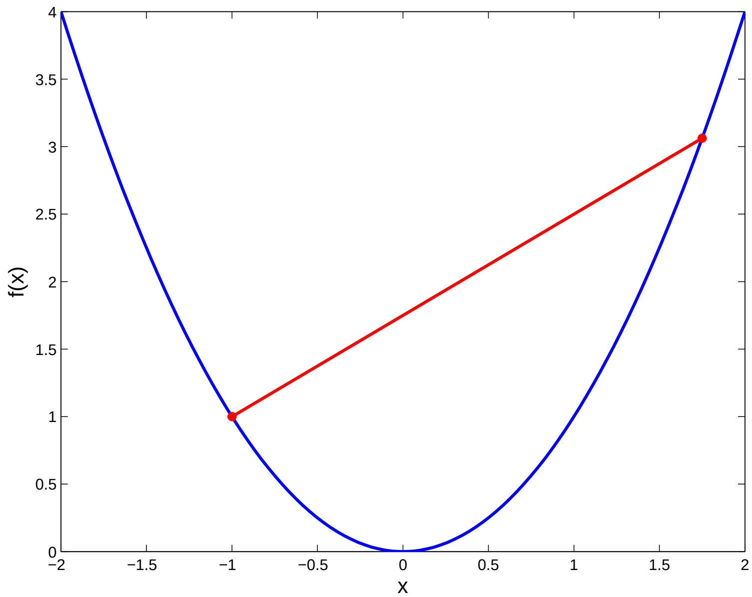



**Fig. 1.3** A convex function $f(x)$ (blue line). The red line is $\alpha f(x) + (1-\alpha)f(y)$ for all $\alpha \in [0,1]$. Note that the blue line is below the red line for all points $\alpha x + (1-\alpha)y$ for $\alpha \in [0,1]$, i.e. $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)\ \forall \alpha \in [0,\,1]$.

convex function is a convex set. A function $f : \mathcal{C} \mapsto \mathbb{R}$ is *strictly convex* if

$$f(\alpha x+(1-\alpha)y) < \alpha f(x)+(1-\alpha)f(y) \quad \forall x, y \in \mathbb{R}^n, x \neq y\, \forall \alpha \in (0,\,1) \tag{1.13}$$

The feasible set

$$\Omega = \{x \in \mathbb{R}^n : \quad c_i(x) = 0,\, i \in \mathcal{E}, \quad c_i(x) \geq 0,\, i \in \mathcal{I}\} \tag{1.14}$$

will only be a convex set if the equality constraints are affine functions, $c_i(x) = a_i'x + b_i$ for $i \in \mathcal{E}$, and the inequality constraints, $c_i(x) \geq 0$ for $i \in \mathcal{I}$, are concave functions. A function $c_i(x)$ is a *concave* function if $-c_i(x)$ is a convex function. Therefore, the feasible set of a convex program may be denoted

$$\Omega = \{x \in \mathbb{R}^n : \quad c_i(x) = a_i'x + b_i = 0,\, i \in \mathcal{E}, \quad c_i(x) \geq 0,\, i \in \mathcal{I}\} \tag{1.15}$$

with $c_i(x)$ being concave functions for $i \in \mathcal{I}$.

Consequently, the convex program (1.10) may be stated as

$$\min_{x \in \mathbb{R}^n} \quad f(x) \tag{1.16a}$$
$$s.t. \quad c_i(x) = a_i' x + b_i = 0 \qquad i \in \mathcal{E} \tag{1.16b}$$
$$c_i(x) \geq 0 \qquad i \in \mathcal{I} \tag{1.16c}$$

in which $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a convex twice continuously differentiable function, and $c_i(x)$ for $i \in \mathcal{I}$ are concave twice continuously differentiable functions.

**Example 2 (Univariate Convex Program).** Consider the univariate convex program

$$\min_{x \in \mathbb{R}_{++}} \quad f(x) = (x-1)^2 - \sqrt{x} - \ln(x) \tag{1.17a}$$
$$s.t. \quad c_1(x) = x - 2 \geq 0 \tag{1.17b}$$

Note that this objective functions is only defined for the set $\mathbb{R}_{++} = \{x \in \mathbb{R} : x > 0\}$. Also note that this is a constrained convex optimization problem as the objective function $f(x)$ is a convex function (see Figure 1.4), and the feasible set $\Omega = \{x \in \mathbb{R}_{++} : c_1(x) = x - 2 \geq 0\}$ is a convex set. In fact, the objective functions is strictly convex as is evident from Figure 1.4.

Consequently, as (1.17) is a strictly convex program, it has a unique minimizer, $x^* = 2$. This is illustrated in Figure 1.4.

■

**Example 3 (Multivariate Convex Program).** Consider the two-dimensional nonlinear program

$$\min_{x \in \mathbb{R}^2_{++}} \quad f(x) = (x_1-1)^2 + (x_2-1)^2 - \sqrt{x_1+x_2} - \ln(x_1) - \ln(x_2) \tag{1.18a}$$
$$s.t. \quad c_1(x) = x_1 + x_2 - 4 \geq 0 \tag{1.18b}$$
$$c_2(x) = -x_1^2 - x_2^2 + 16 \geq 0 \tag{1.18c}$$

The contours of this nonlinear program are illustrated in Figure 1.5. The objective function $f(x)$ is a strictly convex function on $\mathbb{R}^2_{++}$. The feasible region

$$\Omega = \left\{ x \in \mathbb{R}^2_{++} : c_1(x) = x_1 + x_2 - 3 \geq 0, \, c_2(x) = -x_1^2 - x_2^2 + 16 \geq 0 \right\}$$

is the non-shaded are. This area is a convex set and therefore the feasible set, $\Omega$, is a convex set. There is one point, $x^* = (2, 2)$, that satisfies the first order optimality conditions. This point is a global minimizer. This is illustrated in Figure 1.5 by the blue mark. It should be noted that (1.18) is a nonlinear program. However, the main assessment of the hardness of a problem is not whether it is linear or nonlinear but whether it is convex or non-convex. For non-convex optimization problems, the points found by typical algorithms can be a local minimizer, a saddle point, or a maximizer.

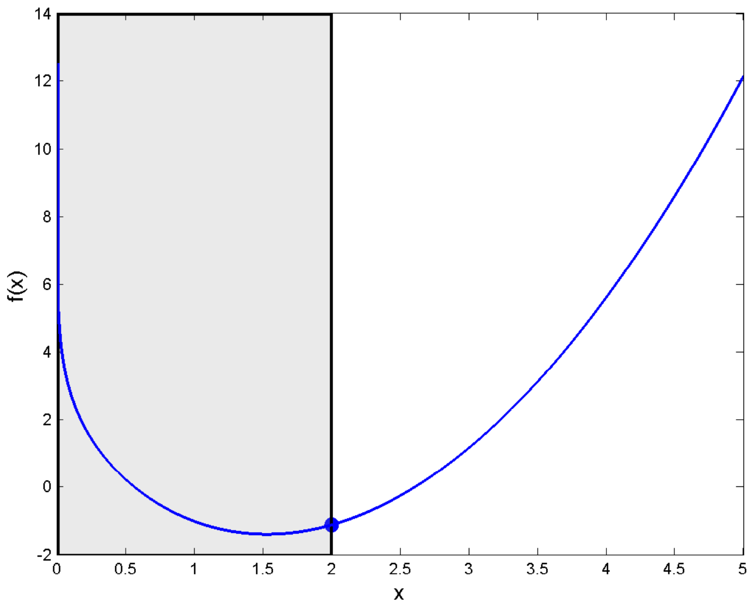



**Fig. 1.4** Univariate Convex Program (1.17). The feasible region $\Omega = \{x \in \mathbb{R}_{++} : c_1(x) = x - 2 \geq 0\}$ is the non-shaded area. The minimizer is the blue mark.

For convex programs the point found is guaranteed to be a global minimizer. For strictly convex programs the global minimizer found is unique.

■

### *1.1.3 Convex Quadratic Program (QP)*

Convex quadratic programs (QP) are an important class of convex optimization problems. A convex quadratic program may be stated as

$$\min_{x \in \mathbb{R}^n} \quad f(x) = \frac{1}{2}x'Hx + g'x + \rho \qquad H \succeq 0 \tag{1.19a}$$

$$s.t. \quad c_i(x) = a_i'x + b_i = 0 \qquad i \in \mathcal{E} \tag{1.19b}$$

$$c_i(x) = a_i'x + b_i \geq 0 \qquad i \in \mathcal{I} \tag{1.19c}$$

The objective function is a quadratic function with a positive semi-definite Hessian matrix, $H \succeq 0$. A positive semi-definite matrix satisfies $x'Hx \geq 0$ for all $x \in \mathbb{R}^n$. All eigenvalues of a positive semi-definite matrix are non-

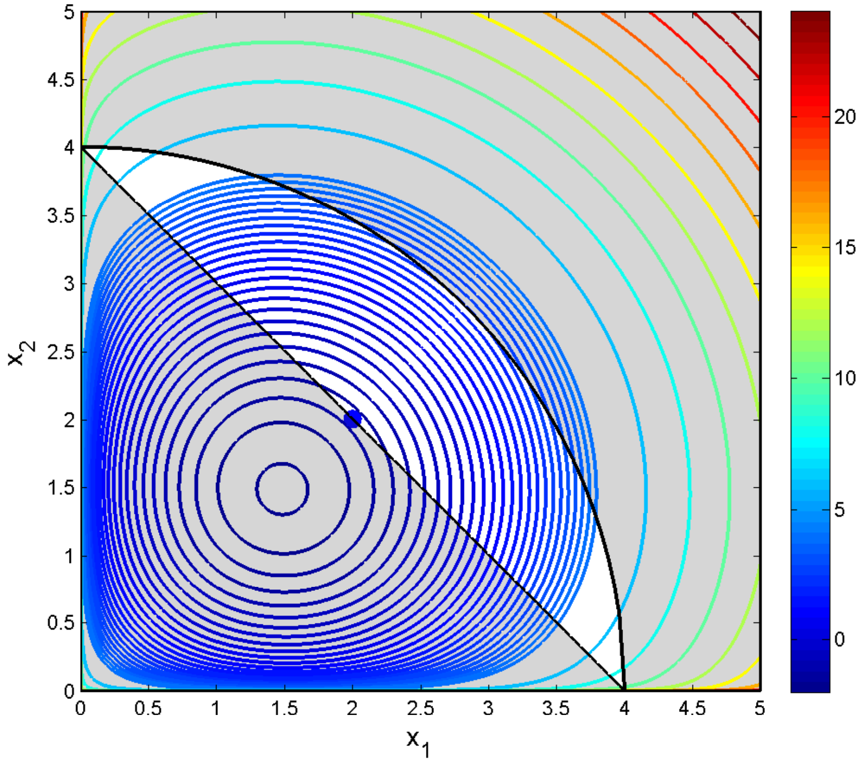



**Fig. 1.5** Multivariate convex program (1.18).

negative. The Hessian matrix, $H \in \mathbb{R}^{n \times n}$, in the objective function of convex quadratic programs is also symmetric.

**Example 4 (Convex Quadratic Functions).** The univariate scalar function

$$f(x) = ax^2 + bx + c \qquad a \geq 0 \quad x \in \mathbb{R} \tag{1.20}$$

is a quadratic semi-definite function for $a \geq 0$. The case $a = 0$ is pathological as in this case the function is not quadratic function but an affine function. (1.20) is a strictly convex function for $a > 0$. The strictly convex univariate quadratic function $f(x) = x^2 + x + 1$ $(a = 1 > 0)$ is plotted in Figure 1.6.

Consider the multivariate scalar quadratic function, $f : \mathbb{R}^2 \mapsto \mathbb{R}$, defined as

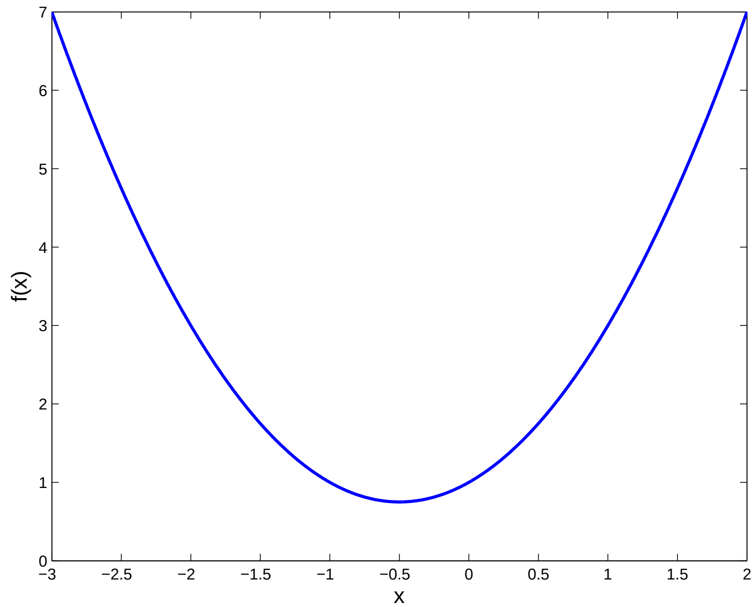



**Fig. 1.6** Univariate strictly convex quadratic function. $f(x) = x^2 + x + 1$. Note that this function has a unique minimum. Strictly convex scalar univariate quadratic functions are functions of the type $f(x) = ax^2 + bx + c$ with $a > 0$ and $x \in \mathbb{R}$.

$$\begin{aligned} f(x) &= \frac{1}{2}x'Hx + g'x + \rho \\ &= \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}' \begin{bmatrix} 6 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 3 \\ 2 \end{bmatrix}' \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 4 \\ &= 3x_1^2 + 2x_2^2 + x_1x_2 + 3x_1 + 2x_2 + 4 \end{aligned} \tag{1.21}$$

with

$$H = \begin{bmatrix} 6 & 1 \\ 1 & 4 \end{bmatrix} \succ 0 \quad g = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad \rho = 4$$

This is a strictly convex program as the matrix $H$ is positive definite as both eigenvalues (3.5858 and 6.4142) of $H$ are positive. (1.21) is plotted in Figure 1.21. By the contour plot in Figure 1.7, it is evident that the strictly convex quadratic function (1.21) has a single minimum value at a unique point. A strictly convex quadratic function has exactly one minimizer.

■

The feasible set

$$\Omega = \{x \in \mathbb{R}^n : c_i(x) = a_i'x + b_i = 0, \; i \in \mathcal{E}, \; c_i(x) = a_i'x + b_i \geq 0, \; i \in \mathcal{I}\} \tag{1.22}$$

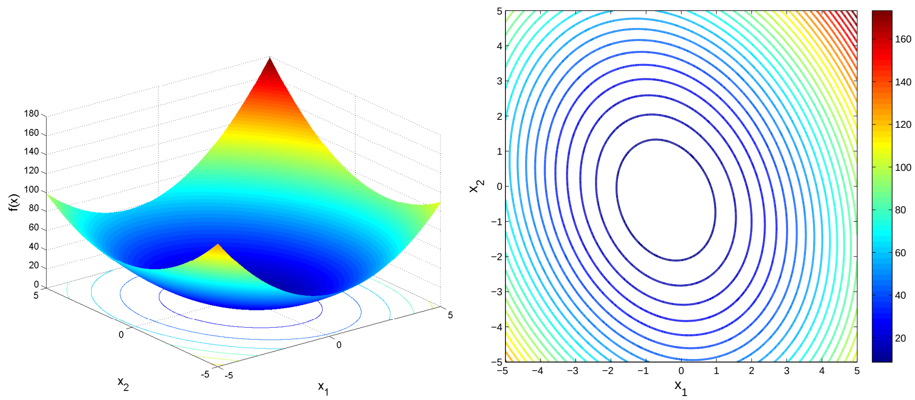



**Fig. 1.7** Multivariate convex quadratic function (1.21). Left: A surface plot of (1.21) with the contours indicated. Right: Contour plot of (1.21).

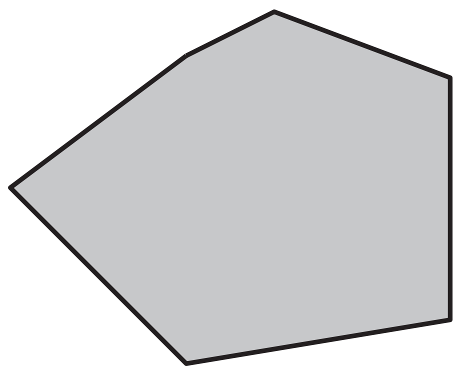


**Fig. 1.8** A polytope.

of a quadratic program (1.19) is a convex set as the equality constraints are affine functions, i.e. $c_i(x) = a_i'x + b_i$ for $i \in \mathcal{E}$, and the inequality constraints are concave functions, i.e. $c_i(x) = a_i'x + b_i$ for $i \in \mathcal{I}$ are concave as an affine function is concave (and convex). The convex set (1.22) is an example of a *polytope*. We have illustrated a polytope in Figure 1.8.

**Example 5 (Convex Quadratic Program).** Consider the two-dimensional convex quadratic program

$$
\begin{aligned}
\min_{x \in \mathbb{R}^2} \quad & f(x) = 3x_1^2 + 2x_2^2 + x_1x_2 + 3x_1 + 2x_2 + 4 && (1.23a) \\
s.t. \quad & c_1(x) = x_1 \geq 0 && (1.23b) \\
& c_2(x) = x_2 \geq 0 && (1.23c) \\
& c_3(x) = x_1 + x_2 - 3 \geq 0 && (1.23d)
\end{aligned}
$$

The objective function (1.23a) is a strictly convex function as it is identical with (1.21). The constraints are all affine functions. Consequently, the constrained optimization problem (1.23) is a strictly convex quadratic program.

The feasible region

$$\Omega = \left\{x \in \mathbb{R}^2 : c_1(x) = x_1 \geq 0,\, c_2(x) = x_2 \geq 0,\, c_3(x) = x_1 + x_2 - 3 \geq 0\right\}$$

of (1.23) is illustrated as the non-shaded region in Figure 1.9. Note that this region is unbounded. It is only the visualization that is restricted to $x = (x_1, x_2) \in [-5,\, 5] \times [-5,\, 5]$. The contour lines of (1.23a) are also illustrated in Figure 1.9. The minimizer of (1.23) exists and is unique. It is $x^* = (x_1^*, x_2^*) =$

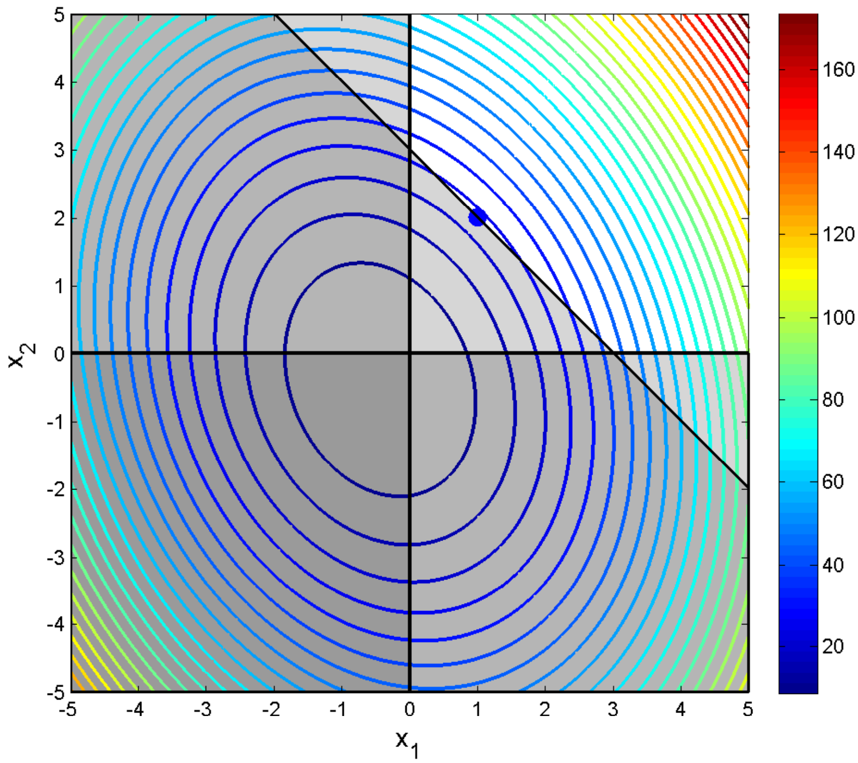



**Fig. 1.9** Convex quadratic program.

$(1, 2)$ and denoted by a blue mark in Figure 1.9.

■

The unconstrained strictly convex quadratic program and the equality constrained convex quadratic program are special types of convex quadratic programs that have important applications in their own right. They are also important as subproblems in the solution of convex quadratic programs (1.19).

The unconstrained strictly convex quadratic program may be stated as

$$\min_{x \in \mathbb{R}^n} \; f(x) = \frac{1}{2}x'Hx + g'x + \rho \qquad H \succ 0 \tag{1.24}$$

with $H \in \mathbb{R}^{n \times n}$ being a symmetric positive definite matrix. The solution to (1.24) exists and is unique. The minimizer may be computed using the optimality condition

$$\nabla f(x) = Hx + g = 0 \tag{1.25}$$

The optimality condition (1.25) may be solved by solution of the system of linear equations

$$Hx = -g \tag{1.26}$$

using a Cholesky factorization as $H$ is symmetric positive definite. Consequently, the solution of (1.24) is computed by solution of (1.26) using a Cholesky factorization.

The equality constrained convex quadratic program appear in applications and is also used as a subproblem for solution of the inequality constrained convex quadratic program (1.19). The equality constrained convex quadratic program is

$$\min_{x \in \mathbb{R}^n} \quad f(x) = \frac{1}{2}x'Hx + g'x + \rho \qquad H \succeq 0 \tag{1.27a}$$

$$s.t. \quad c_i(x) = a_i'x + b_i = 0 \qquad i \in \mathcal{E} = \{1, 2, \ldots, m\} \tag{1.27b}$$

The equality constrained convex QP (1.27) does not have any inequality constraints. It only have equality constraints. The matrix $H \in \mathbb{R}^{n \times n}$ is at least symmetric positive semi-definite. The constraints may be expressed as

$$c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \\ \vdots \\ c_m(x) \end{bmatrix} = \begin{bmatrix} a_1'x + b_1 \\ a_2'x + b_2 \\ \vdots \\ a_m'x + b_m \end{bmatrix} = \begin{bmatrix} a_1' \\ a_2' \\ \vdots \\ a_m' \end{bmatrix} x + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = A'x + b = 0 \tag{1.28}$$

with

$$A = \begin{bmatrix} a_1 \; a_2 \; \ldots \; a_m \end{bmatrix} \tag{1.29a}$$

$$b = \begin{bmatrix} b_1 \; b_2 \; \ldots \; b_m \end{bmatrix}' \tag{1.29b}$$

Define

$$d = -b \tag{1.29c}$$

such that (1.28) may be expressed as the linear system of equations

$$A'x = d \qquad A \in \mathbb{R}^{n \times m},\ x \in \mathbb{R}^n,\ d \in \mathbb{R}^m \tag{1.29d}$$

Using this notation we may express the equality constrained convex QP (1.27) as

$$\min_{x \in \mathbb{R}^n} \quad f(x) = \frac{1}{2}x'Hx + g'x + \rho \qquad H \succeq 0 \tag{1.30a}$$

$$s.t. \qquad A'x = d \tag{1.30b}$$

The minimizer of (1.30) is obtained as the solution of the following linear system of equations

$$\begin{bmatrix} H & -A \\ -A' & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = - \begin{bmatrix} g \\ d \end{bmatrix} \tag{1.31}$$

The minimizer exists and is unique if the matrix $K = \begin{bmatrix} H & -A \\ -A' & 0 \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$ is non-singular. $K$ is called the Karush-Kuhn-Tucker matrix or the KKT matrix. Similarly, the linear system of equations (1.31) is called the KKT system. They are also called the first order optimality conditions for (1.30). As $H$ is symmetric, the KKT matrix, $K$, is also symmetric. The matrix $K$ always has both positive and negative eigenvalues for $H \succeq 0$. Such matrices are called indefinite matrices. Therefore, (1.31) is symmetric indefinite system of linear equations. Such systems may be solved by the block symmetric indefinite $LDL^T$ factorization. Also a number of factorizations for the KKT-system (1.31) with specialized structure exists. However, the main point is that the solution of a general convex quadratic program (1.19) may be solved by solution of a sequence of equality constrained quadratic programs (1.30) which in turn may be solved by solution of the KKT system (1.31).

### *1.1.4 Linear Program (LP)*

The linear program (LP) is another example of a convex program. Mathematically, a linear program may be expressed as

$$\min_{x \in \mathbb{R}^n} \quad f(x) = g'x + \rho \tag{1.32a}$$

$$s.t. \quad c_i(x) = a_i'x + b_i = 0 \quad i \in \mathcal{E} \tag{1.32b}$$

$$c_i(x) = a_i'x + b_i \geq 0 \quad i \in \mathcal{I} \tag{1.32c}$$

The objective function, $f : \mathbb{R}^n \mapsto \mathbb{R}$, and the constraint functions, $c_i : \mathbb{R}^n \mapsto \mathbb{R}$ for $i \in \mathcal{E} \cup \mathcal{I}$, are affine functions. Affine functions are convex as well as concave. Therefore, the objective function is a convex function and the inequality constraint functions are concave functions. Trivially, the equality constraint functions are affine. Therefore, a linear program (1.32) is an example of a convex program.

The feasible region

$$\Omega = \{x \in \mathbb{R}^n : c_i(x) = a_i' x + b_i = 0,\, i \in \mathcal{E}, \quad c_i(x) = a_i' x + b_i \geq 0,\, i \in \mathcal{I}\} \tag{1.33}$$

of (1.32) is a polytope (see Figure 1.8).

Sometimes, a linear program is stated as

$$\min_{x \in \mathbb{R}^n} \quad \phi(x) = g' x \tag{1.34a}$$
$$s.t. \quad a_i' x = d_i \qquad i \in \mathcal{E} \tag{1.34b}$$
$$a_i' x \geq d_i \qquad i \in \mathcal{I} \tag{1.34c}$$

In this case, it is obvious why it is called a linear program. The objective function $\phi(x) = g'x$ as well as the constraint functions $a_i'x$ for $i \in \mathcal{E} \cup \mathcal{I}$ are linear functions. Let $d_i = -b_i$ for $i \in \mathcal{E} \cup \mathcal{I}$. Then the solution, $x^* \in \mathbb{R}^n$, of (1.34) is identical to the solution of (1.32). The value of the programs are related as $f(x^*) = \phi(x^*) + \rho$.

**Example 6 (Linear Program).** Consider the two dimensional linear program

$$\min_{x \in \mathbb{R}^2} \quad f(x) = -2x_1 - x_2 \tag{1.35a}$$
$$s.t. \quad c_1(x) = x_1 \geq 0 \tag{1.35b}$$
$$c_2(x) = x_2 \geq 0 \tag{1.35c}$$
$$c_3(x) = -x_1 - x_2 + 4 \geq 0 \tag{1.35d}$$

This linear program is visualized in Figure 1.10 by its contour plot. The feasible region

$$\Omega = \left\{x \in \mathbb{R}^2 : x_1 \geq 0,\, x_2 \geq 0,\, -x_1 - x_2 + 4 \geq 0\right\}$$

is the non-shaded (white) region in Figure 1.10. This is a polytope. For the two dimension LP, i.e. $x \in \mathbb{R}^2$, the minimizer, $x^* = (4, 0)$, is located at the intersection of two constraints.

This observation is a general property of linear programs. If the minimizer of an n-dimensional linear program exists, it is located at the intersection of (at least) $n$ constraints. The solution is not necessarily unique. Furthermore, there is no guarantee that the solution of a linear program exists. The constraints may be inconsistent such that the feasible region is empty, or the program may be unbounded.

■

Linear programs (1.32) are solved using either an interior-point algorithm or the simplex algorithm. The major computational operation in the interior-

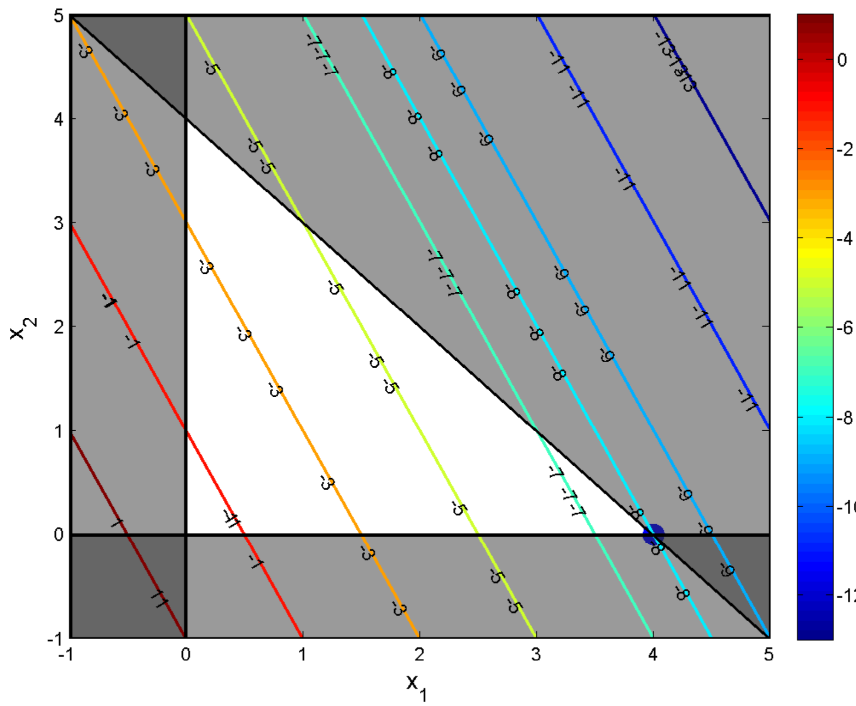



**Fig. 1.10** Contour plot of the linear program (1.35). The feasible region is the non-shaded region. The minimizer is $x^* = (4, 0)$ and the minimum value if $f(x^*) = -8$. Note that the two-dimensional minimizer is located at the intersection of two constraints.

point algorithm is solution of a KKT-system

$$\begin{bmatrix} D & -A \\ -A' & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = - \begin{bmatrix} r \\ d \end{bmatrix} \tag{1.36}$$

in which $D$ is a diagonal matrix with positive entries and $A = \begin{bmatrix} a_i \end{bmatrix}_{i \in \mathcal{E}}$. In the simplex algorithm (revised simplex algorithm), the major computation effort concern solution of the KKT system

$$\begin{bmatrix} 0 & -B \\ -B' & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = - \begin{bmatrix} g \\ d \end{bmatrix} \tag{1.37}$$

in which $B \in \mathbb{R}^{n \times n}$ is a non-singular matrix. (1.37) may be solved by solution of

$$By = g \tag{1.38a}$$

$$B'x = d \tag{1.38b}$$

using an LU factorization of $B$.

# Part II
# Theory

# Chapter 2
# Optimality Conditions

The main result stated in this chapter is the first order optimality conditions for constrained optimization problems. These conditions are also called the Karush-Kuhn-Tucker conditions or the KKT conditions. The KKT conditions constitute the foundation for numerical solution of constrained optimization problems and can be used for locating a constrained local minimizer. Constrained optimization problems are solved numerically by algorithms that find solutions to the KKT conditions. Typically, Newton type algorithms are used to generate sequences that converge to a KKT point, and this point is taken as a solution to the constrained optimization problem. Sufficient second order optimality conditions may be used to confirm that the KKT point is a local minimum, while the necessary second order optimality conditions may be used to prove that a KKT is not a local minimum. For convex programs, the KKT conditions are both necessary and sufficient for global optimality.

We first review the optimality conditions for univariate and multivariate unconstrained optimization. Then we present the first order optimality conditions for constrained optimization problems. The first order optimality conditions are presented in two different forms; one form that is suitable for active set algorithm and another form that is suitable for interior point algorithms.

## 2.1 Unconstrained Optimization

The optimality conditions for constrained optimization is similar to the optimality conditions for unconstrained optimization. In this section we review the optimality conditions for unconstrained optimization. We show a number of univariate unconstrained optimization problems, state the optimality conditions for univariate unconstrained optimization problems, and generalize these conditions to the multivariate unconstrained optimization problem.

### *2.1.1 Univariate*

Consider the univariate unconstrained optimization problem

$$\min_{x \in \mathcal{D}} f(x) \tag{2.1}$$

in which $\mathcal{D} \subseteq \mathbb{R}$ is some open set. We assume that $f$ is twice continuously differentiable in the domain $\mathcal{D}$, i.e. $f \in \mathcal{C}^2(\mathcal{D})$.

A global minimizer, $x^* \in \mathcal{D}$, is defined by

$$f(x^*) \leq f(x) \qquad \forall x \in \mathcal{D} \tag{2.2}$$

and the value $f(x^*)$ is called the global minimum. Define a neighborhood, $N(x^*)$, as

$$\mathcal{N}(x^*) = \{x \in \mathcal{D} : \, |x - x^*| \leq \epsilon\} \tag{2.3}$$

for some $\epsilon > 0$. Then we say that $x^* \in \mathcal{D}$ is a local minimizer of (2.1) if

$$f(x^*) \leq f(x) \qquad \forall x \in \mathcal{N}(x^*) \tag{2.4}$$

Two type of conditions are used to determine local minimizers. The *necessary* optimality conditions, assume that we know the local minimizer and specify conditions that are satisfied in this case. They may be used to prove that a point is not a local minimizer by assuming that a point is a local minimizer and then showing that the necessary optimality conditions are not satisfied. The second kind of optimality conditions are the *sufficient* optimality conditions. The sufficient optimality conditions are used to compute a local minimizer. If the second order optimality conditions are satisfied at a point, then the point is a local minimizer. Optimization algorithms are constructed such that they compute a point satisfying the sufficient optimality conditions. The necessary and sufficient optimality conditions for the univariate unconstrained optimization problem (2.1) are stated in the next two propositions.

**Proposition 2.1 (Necessary Optimality Conditions).** *Let $x \in \mathcal{D}$ be a local minimizer of (2.1). Then*

$$\frac{df}{dx}(x) = 0 \tag{2.5a}$$

$$\frac{d^2 f}{dx^2}(x) \geq 0 \tag{2.5b}$$

□

**Proposition 2.2 (Sufficient Optimality Conditions).** *Let $x \in \mathcal{D}$ satisfy*

$$\frac{df}{dx}(x) = 0 \tag{2.6a}$$

$$\frac{d^2 f}{dx^2}(x) > 0 \tag{2.6b}$$

*Then $x$ is a local minimizer of (2.1).*

□

**Remark 1 (First and Second Order Optimality Conditions).** The condition $\frac{df}{dx}(x) = 0$ is called the first order optimality condition, while $\frac{d^2 f}{dx^2}(x) \geq 0$ and $\frac{d^2 f}{dx^2}(x) > 0$ are called second order optimality conditions.

■

**Remark 2 (Numerical Computation of a Minimizer).** Numerical algorithms compute a local minimizer using some variant of Newton's method to the equation

$$F(x) = \frac{df}{dx}(x) = 0 \tag{2.7}$$

Hence, using a first order Taylor approximation around $x_k$

$$F(x) \approx F(x_k) + \frac{dF}{dx}(x_k)(x - x_k) = 0 \tag{2.8}$$

the iterates $\{x_k\}$ are computed as

$$x_{k+1} = x_k - \alpha_k \left[\frac{dF}{dx}(x_k)\right]^{-1} F(x_k) \tag{2.9}$$

with $\alpha_k$ being the step length. As $F(x) = \frac{df}{dx}(x)$, this is equivalent with

$$x_{k+1} = x_k - \alpha_k \left[\frac{d^2 f}{dx^2}(x_k)\right]^{-1} \frac{df}{dx}(x_k) \tag{2.10}$$

Most numerical optimization algorithms determine minimizers by application of some variant of Newton's method to the first order optimality conditions.

■

**Proposition 2.3 (Convex Optimization - Optimality Conditions).** *Let $f : \mathcal{D} \mapsto \mathbb{R}$ be a convex function. Then $x \in \mathcal{D}$ is global minimizer of (2.1) if and only if*

$$\frac{df}{dx}(x) = 0 \tag{2.11}$$

□

**Remark 3 (Univariate Convex Optimization).**

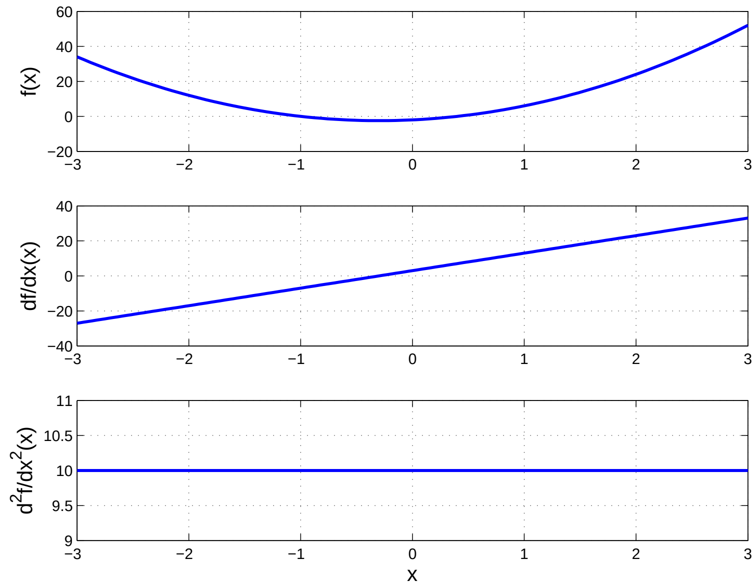



**Fig. 2.1** Plots of a univariate convex function, $f(x) = 5x^2 + 3x - 2$, and its derivatives.

1. If $f$ is convex the minimum does not need to exist or be unique.
2. If $f$ is *strictly convex*, the minimum exists and is unique.
3. Proposition 2.3 is a necessary and sufficient condition. It concerns a global minimizer and not just a local minimizer.

■

In the next examples we illustrate the meaning of the necessary and sufficient optimality conditions for univariate unconstrained optimization problems.

**Example 7 (Convex and Concave Univariate Functions).** Consider the univariate unconstrained optimization problem

$$\min_{x \in \mathbb{R}} f(x) = 5x^2 + 3x - 2 \tag{2.12}$$

The stationary point is

$$\frac{df}{dx}(x) = 10x + 3 = 0 \qquad \Leftrightarrow \qquad x = -3/10 \tag{2.13}$$

and we note that the second derivative is positive at this point, i.e. $\frac{d^2 f}{dx^2}(x) = \frac{d^2 f}{dx^2}(\frac{-3}{10}) = 10 > 0$. Consequently $x = -3/10$ is a minimizer. This is illustrated in Figure 2.1.

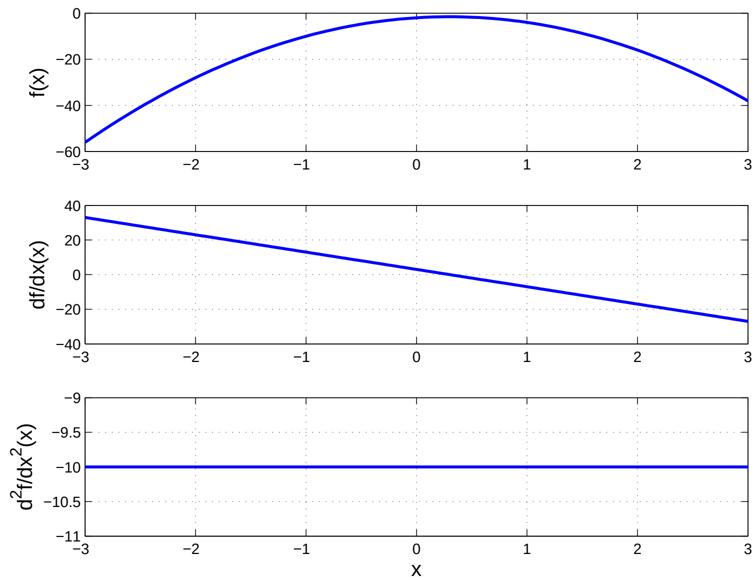



**Fig. 2.2** Plots of a univariate concave function, $f(x) = -5x^2 + 3x - 2$, and its derivatives.

Next consider the univariate unconstrained optimization problem

$$\min_{x \in \mathbb{R}} f(x) = -5x^2 + 3x - 2 \tag{2.14}$$

The stationary point of this problem is

$$\frac{df}{dx}(x) = -10x + 3 = 0 \quad \Leftrightarrow \quad x = 3/10 \tag{2.15}$$

and the second derivative is negative, i.e. $\frac{d^2f}{dx^2}(x) = \frac{d^2f}{dx^2}(0) = -10 < 0$. Figure 2.2 illustrates the plots of $f(x)$, $\frac{df}{dx}(x)$, and $\frac{d^2}{dx^2}(x)$.

This example illustrates that stationarity of a point, $x$ ($\frac{df}{dx}(x) = 0$), is not enough to determine whether the point is a minimizer. If the second derivative is positive, the point $x$ is a local minimizer. If the second derivative is negative, the point is a local maximizer. When the second derivative is zero, we cannot say anything about the nature of the point.

When the functions are strictly convex as in this example, the local minimizer is also the unique global minimizer.

■

**Example 8 (The Hessian is Zero in the Stationary Point).** Consider the functions

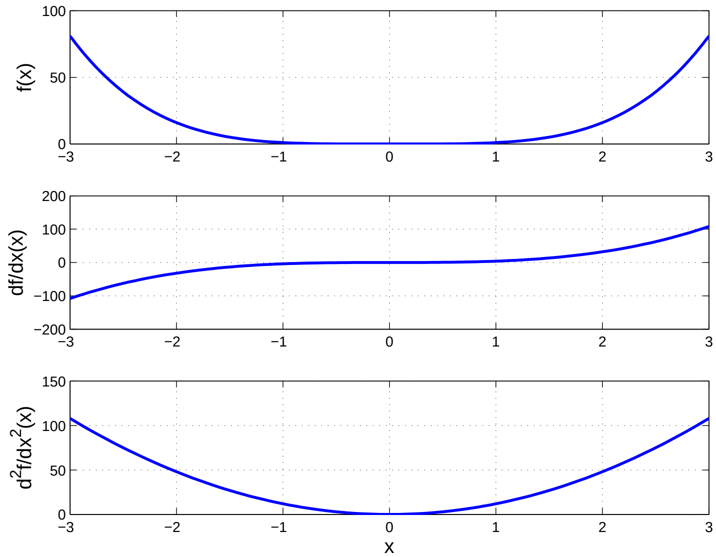



**Fig. 2.3** Plot of $f(x) = x^4$ and its derivatives.

$$f_1(x) = x^4$$
$$f_2(x) = -x^4$$
$$f_3(x) = x^3$$

which have the first and second derivatives

$$\begin{array}{ccc} f_1(x) = x^4 & f_2(x) = -x^4 & f_3(x) = x^3 \\ \dfrac{df_1}{dx}(x) = 4x^3 & \dfrac{df_2}{dx}(x) = -4x^3 & \dfrac{df_3}{dx}(x) = 3x^2 \\ \dfrac{d^2 f_1}{dx^2}(x) = 12x^2 & \dfrac{d^2 f_2}{dx^2}(x) = -12x^2 & \dfrac{d^2 f_3}{dx^2}(x) = 6x \end{array}$$

and are plotted in Figures 2.3-2.5.

Notice that $x^* = 0$ is a stationary point for all three functions, i.e. $\frac{df_1}{dx}(0) = \frac{df_2}{dx}(0) = \frac{df_3}{dx}(0) = 0$. Furthermore, the second derivative at $x^* = 0$ is zero for all three cases, i.e. $\frac{d^2 f_1}{dx^2}(0) = \frac{d^2 f_2}{dx^2}(0) = \frac{d^2 f_3}{dx^2}(0) = 0$.

However 0 is a minimizer for $f_1$, a maximizer for $f_2$, and a saddle point (inflection point) for $f_3$. Consequently, having found a point $x^* \in \mathcal{D}$ that is stationary, $\frac{df}{dx}(x^*) = 0$, and have zero second derivative, $\frac{d^2 f}{dx^2}(x^*) = 0$, is not *sufficient* to characterize the point as a minimizer. The point can be a minimizer, but it can also be a saddle point or a maximizer.

■

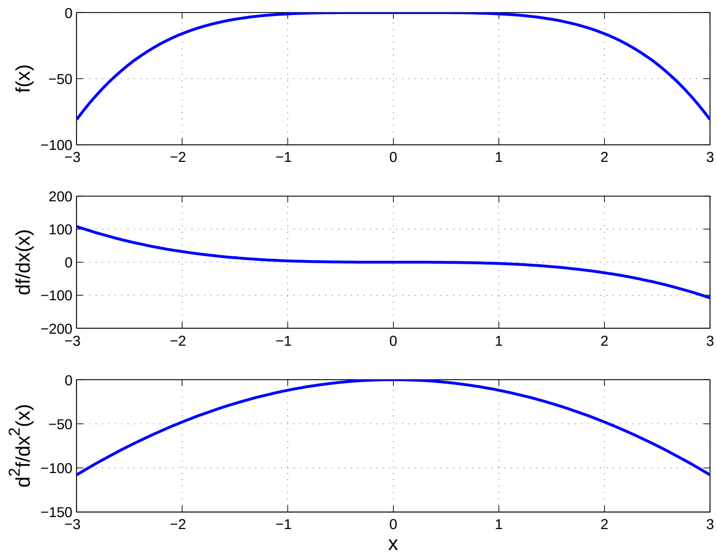



**Fig. 2.4** Plot of $f(x) = -x^4$ and its derivatives.

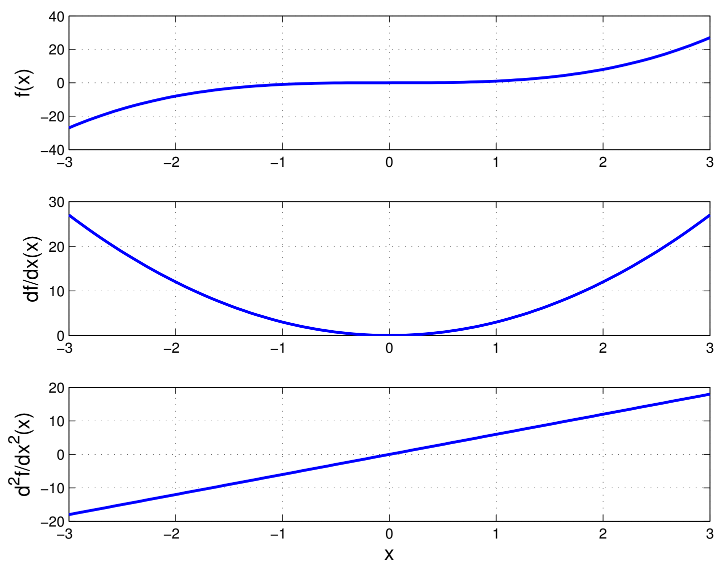



**Fig. 2.5** Plot of $f(x) = x^3$ and its derivatives.

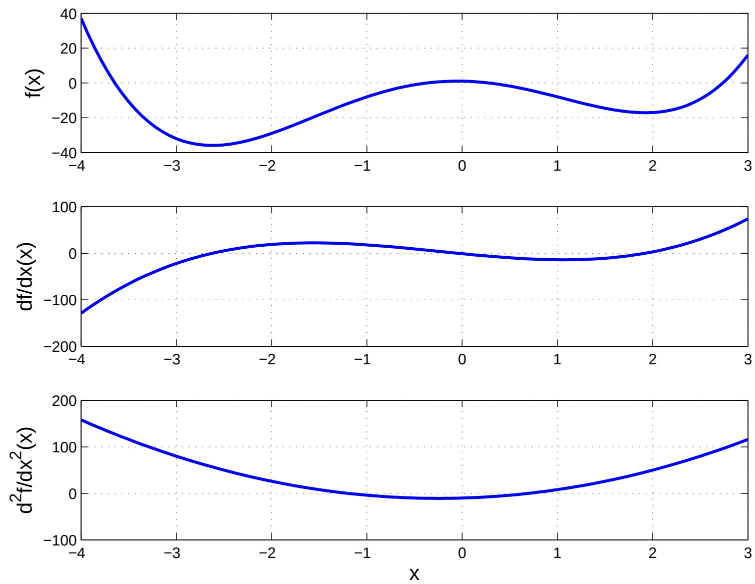



**Fig. 2.6** Plot of $f(x) = x^4 + x^3 - 10x^2 - x + 1$ and its derivatives.

**Example 9 (Univariate Non-Convex Function).** Consider the univariate function

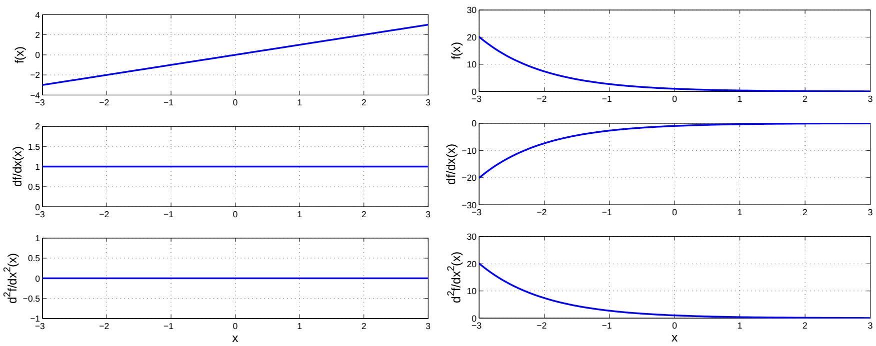
$$f(x) = x^4 + x^3 - 10x^2 - x + 1 \tag{2.16}$$

This function and its first and second derivatives are plotted in Figure 2.6. By inspection, it is obvious that the function has two minima (a global and a local minima) and a local maximum. At the minimizers and the maximizer $\frac{df}{dx}(x) = 0$. The second derivative is positive at the minimizers, $\frac{d^2f}{dx^2}(x) > 0$, while it is positive at the maximizer.

Consequently, a point $x$ with $\frac{df}{dx}(x) = 0$ is not sufficient to guarantee that the point is a minimizer of a general smooth non-convex function. The point can be a minimizer, but it can also be a maximizer (or a saddle point). Even if the second derivative is positive, $\frac{df^2}{dx^2}(x) > 0$, in the stationary point ($\frac{df}{dx}(x) = 0$), we are not guaranteed that the point is the global minimizer. However, we know that it is a local minimizer.

This illustrates why most numerical optimization algorithms at best are able to find local minimizers and not global minimizer of non-convex functions. If they just solve $\frac{df}{dx}(x) = 0$ and have no additional checks we do not even know if the point found is a minimizer.

■




(a) $f_1(x) = x$ (b) $f_2(x) = e^{-x}$

**Fig. 2.7** Univariate unconstrained optimization problems without a minimizer. Plots of the functions and their derivatives.

**Example 10 (Optimization Problems Without a Minimizer).** Unconstrained optimization problems do need to have a solution.

Consider

$$\min_{x \in \mathbb{R}} f_1(x) = x \tag{2.17}$$

for which $f_1(x) = x \to -\infty$ as $x \to -\infty$. The objective function is unbounded from below. Consequently the minimizer does not exist. This is consistent with the fact that the first order optimality conditions has no solution, i.e. $\frac{df_1}{dx}(x) = 1 \neq 0$.

As another example, consider

$$\min_{x \in \mathbb{R}} f_2(x) = e^{-x} \tag{2.18}$$

for which $f_2(x) \to 0$ as $x \to \infty$. The function $f_2(x) = e^{-x}$ is bounded from below but the bound is never attained. Consequently, the minimizer does not exist. This is consistent with the fact that $\frac{df_2}{dx}(x) = -e^{-x} = 0$ does not have a solution as $e^{-x} > 0$.

■

**Example 11 (Non-Smooth Optimization Problem).** Consider the minimization problem

$$\min_{x \in \mathbb{R}} f(x) = |x| \tag{2.19}$$

The non-smooth $f(x) = |x|$ and its derivatives are illustrated in Figure 2.8. The minimizer is $x^* = 0$ and is unique. However, the derivative $\frac{df}{dx}(x)$ is not a continuous function and never zero. The optimality conditions stated in Proposition 2.1 and Proposition 2.2 cannot be used to determine the minimizer of the problem (2.19) as the objective function is not twice continuously differentiable; the first derivative is not continuous.

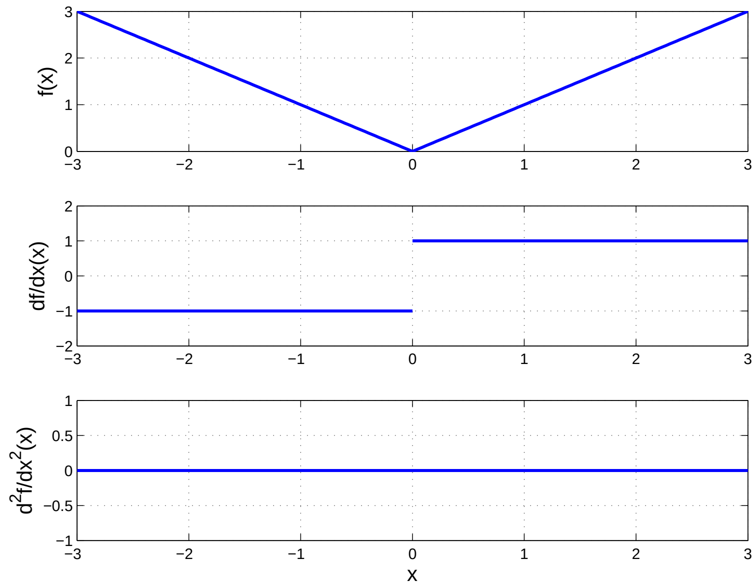



**Fig. 2.8** Plot of the non-smooth function $f(x) = |x|$ and its derivatives.

■

**Example 12 (Optimization in Closed Set).** Consider the optimization problem

$$\min_{x\in\mathbb{R}_+} f(x) = \sqrt{x} \tag{2.20}$$

Obviously, the minimizer is $x = 0$. The objective function and its derivatives are

$$f(x) = x^{1/2} = \sqrt{x} \tag{2.21a}$$

$$\frac{df}{dx}(x) = \frac{1}{2}x^{-1/2} = \frac{1}{2\sqrt{x}} \tag{2.21b}$$

$$\frac{d^2 f}{dx^2}(x) = -\frac{1}{4}x^{-3/2} = \frac{-1}{4x\sqrt{x}} \tag{2.21c}$$

At the minimizer, $x = 0$, the first derivative, $\frac{df}{dx}(0)$, is not defined and $\frac{df}{dx}(x) \to \infty$ for $x \to 0^+$. Therefore, this examples seems to contradict the optimality conditions saying that $\frac{df}{dx}(x) = 0$ at a minimizer. The problem is that the non-negative real-numbers (0 and the positive real numbers), $\mathbb{R}_+$, is a closed set with 0 on its boundary. The optimality conditions are valid only for optimization problems specified in open domains, i.e. $x \in \mathcal{D} \subseteq \mathbb{R}$ with $\mathcal{D}$ being an open set.

■

### 2.1.2 Multivariate

Consider the multivariate unconstrained optimization problem

$$\min_{x \in \mathcal{D}} \quad f(x) \qquad \mathcal{D} \subseteq \mathbb{R}^n \tag{2.22}$$

in which $\mathcal{D} \subset \mathbb{R}^n$ is some open set. We assume that $f$ is twice continuously differentiable in the domain $\mathcal{D}$, i.e. $f \in \mathcal{C}^2(\mathcal{D})$.

A global minimizer, $x^* \in \mathcal{D}$, of (2.22) is defined as a point $x^*$ satisfying

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{D} \tag{2.23}$$

Define a neighborhood of $x^* \in \mathcal{D}$ as

$$\mathcal{N}(x^*) = \{x \in \mathcal{D} : \|x - x^*\| \leq \epsilon\} \tag{2.24}$$

for some $\epsilon > 0$. Then a local minimizer, $x^* \in \mathcal{D}$, of (2.22) is defined by

$$f(x^*) \leq f(x) \qquad \forall x \in \mathcal{N}(x^*) \tag{2.25}$$

The optimality conditions for smooth unconstrained multivariate optimization are generalizations of the optimality conditions smooth unconstrained univariate optimization. The necessary optimality conditions may be used to prove that a point is not a local minimizer, while the sufficient optimality conditions may be used to compute a local minimizer.

**Proposition 2.4 (Necessary Optimality Conditions).** *Let $x \in \mathcal{D}$ be a local minimizer of (2.22). Then*

$$\nabla f(x) = 0 \tag{2.26a}$$

$$\nabla^2 f(x) \succeq 0 \tag{2.26b}$$

□

**Proposition 2.5 (Sufficient Optimality Conditions).** *Let $x \in \mathcal{D}$ satisfy*

$$\nabla f(x) = 0 \tag{2.27a}$$

$$\nabla^2 f(x) \succ 0 \tag{2.27b}$$

*Then $x$ is a local minimizer of (2.22).*

□

**Remark 4 (Positive Definite and Positive Semi-Definite Matrices).** A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called positive definite if $x'Ax > 0$ for all $x \neq 0$ and positive semi-definite if $x'Ax \geq 0$ for all $x$. The matrix, $A$, is positive definite if all its eigenvalues are positive. The matrix, $A$, is positive semi-definite if all its eigenvalues are non-negative (zero or positive). We denote a positive definite matrix as $A \succ 0$ and a positive semi-definite matrix as $A \succeq 0$.

■

**Proposition 2.6 (Convex Unconstrained Optimization - Optimality Condition).** *Let $f : \mathcal{D} \mapsto \mathbb{R}$ be a convex function. Then $x \in \mathcal{D}$ is a global minimizer of (2.22) if and only if*

$$\nabla f(x) = 0 \tag{2.28}$$

□

**Remark 5 (The Optimality Conditions and Numerical Optimization).** Numerical algorithms for unconstrained optimization, locate local minimizers by solution of the first order part of the optimality conditions. Let $F : \mathcal{D} \mapsto \mathbb{R}^n$ with $\mathcal{D} \subseteq \mathbb{R}^n$ defined by

$$F(x) = \nabla f(x) = 0 \tag{2.29}$$

This vector function may be approximated by a first order Taylor approximation around an iterate $x_k$

$$F(x) \approx F(x_k) + \nabla F(x_k)'(x - x_k) = 0 \tag{2.30}$$

This is the same as

$$\nabla f(x) \approx \nabla f(x_k) + \nabla^2 f(x_k)'(x - x_k) = 0 \tag{2.31}$$

Consequently, we may compute the minimizer $x$ as the limit point of the sequence $\{x_k\}$ with the iterates $x_k$ determined by

$$\nabla^2 f(x_k) d_k = -\nabla f(x_k) \qquad \text{(Solve for } d_k\text{)} \tag{2.32a}$$

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.32b}$$

$\alpha_k$ is a step length parameter. This algorithm for solution of (2.29) is some variation of Newton's algorithm for solution of systems of nonlinear equations.

■

**Remark 6 (Numerical Optimization Algorithms).** The numerical optimization algorithms for solution of (2.22) has as its core solution of the linear system

$$Hd_k = -\nabla f(x_k) \tag{2.33}$$

The algorithms differ by their approximation of the Hessian matrix ($H \approx \nabla^2 f(x_k)$)

| | | |
|---|---|---|
| $H = I$ | Steepest Descent | (2.34a) |
| $H = \nabla^2 f(x_k)$ | Newton | (2.34b) |
| $H := H + v_k w_k'$ | Quasi-Newton | (2.34c) |
| $H = \nabla^2 f(x_k) + \mu_k I$ | Trust Region / Levenberg Marquardt | (2.34d) |

and the linear algebra used for solving the linear system.

■

**Example 13 (Unconstrained Strictly Convex Optimization).** Consider the strictly convex quadratic optimization problem

$$\min_{x\in\mathbb{R}^n} \quad f(x) = \frac{1}{2}x'Hx + g'x + \rho \qquad H \succ 0 \tag{2.35}$$

$H \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. The optimization problem is strictly convex as the Hessian matrix is positive definite, $H \succ 0$. The strictly convex objective function and its derivatives are

$$f(x) = \frac{1}{2}x'Hx + g'x + \rho \tag{2.36a}$$

$$\nabla f(x) = Hx + g \tag{2.36b}$$

$$\nabla^2 f(x) = H \succ 0 \tag{2.36c}$$

The necessary and sufficient optimality conditions for a global minimizer of a strictly convex quadratic program is

$$\nabla f(x) = Hx + g = 0 \tag{2.37}$$

This may be rearranged such that the global minimizer is obtained by solution of

$$Hx = -g \tag{2.38}$$

As $H$ is symmetric positive definite, the solution is obtained using a Cholesky factorization of $H$, i.e. $H = LL'$. We may denote the solution $x = -H^{-1}g$ but with the understanding that it is solved using the Cholesky factorization and not by direct inversion of the Hessian matrix $H$.

■

**Example 14 (Unconstrained 2-Dimensional Quadratic Optimization).** Consider the unconstrain two-dimensional quadratic optimization problem

$$\min_{x\in\mathbb{R}^2} f(x) = 3x_1^2 + 2x_2^2 + x_1x_2 + 3x_1 + 2x_2 + 4 \tag{2.39}$$

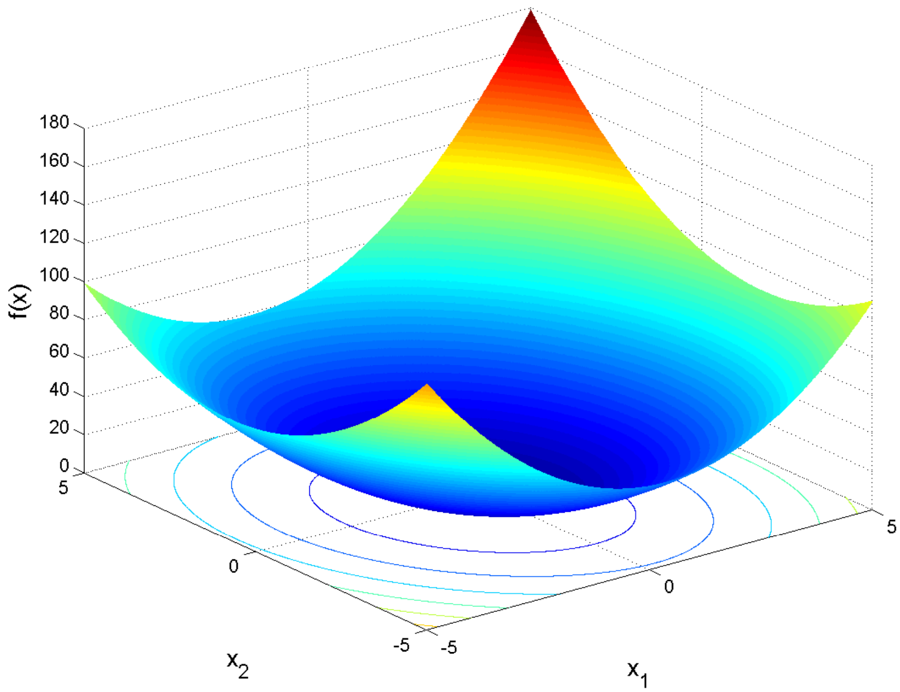



**Fig. 2.9** Plot of the objective function in (2.39).

The objective function is plotted in Figure 2.9. It is flat in the bottom at the minimizer. This corresponds to all derivatives being zero, i.e. $\frac{\partial f}{\partial x_1}(x) = \frac{\partial f}{\partial x_2}(x) = 0$ at the minimizer $x$. The objective function is a quadratic function that may be expressed as

$$f(x) = \frac{1}{2}x'Hx + g'x + \rho \tag{2.40}$$

with

$$H = \begin{bmatrix} 6 & 1 \\ 1 & 4 \end{bmatrix} \succ 0 \qquad g = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \qquad \rho = 4 \tag{2.41}$$

The Hessian matrix $H$ is positive definite as both its eigenvalues (3.58 and 6.41) are positive. The optimal solution is computed by solving $\nabla f(x) = Hx + g = 0$, i.e.

$$\begin{bmatrix} 6 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = - \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -0.4348 \\ -0.3913 \end{bmatrix} \tag{2.42}$$

The contour plot of the quadratic function and the minimizer is illustrated in Figure 2.10.

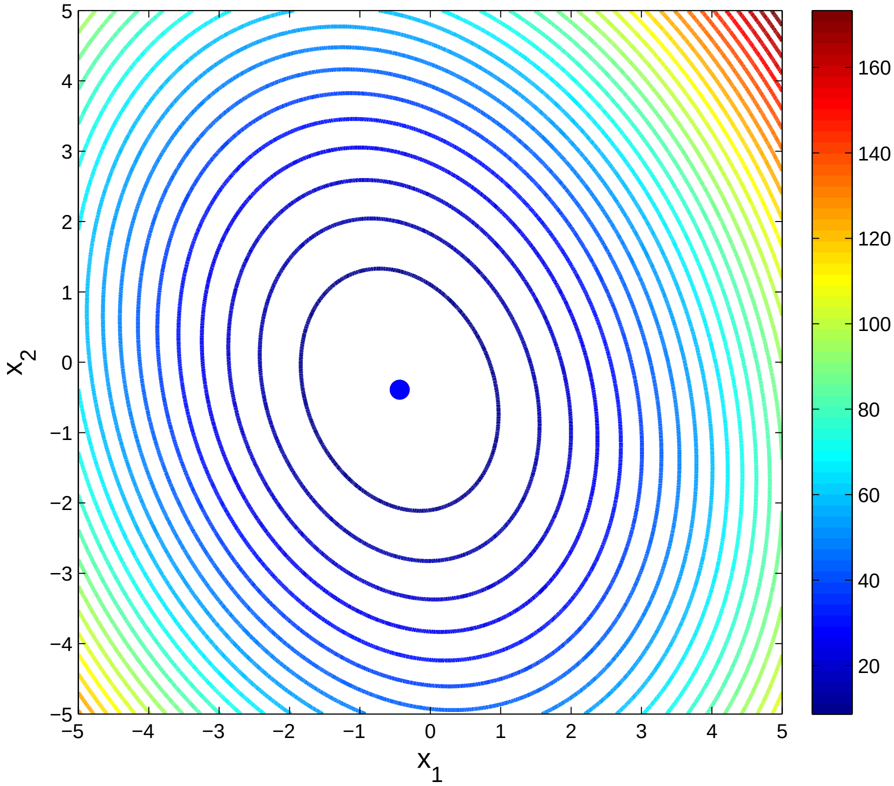



**Fig. 2.10** Contour plot and the minimizer for (2.39).

■

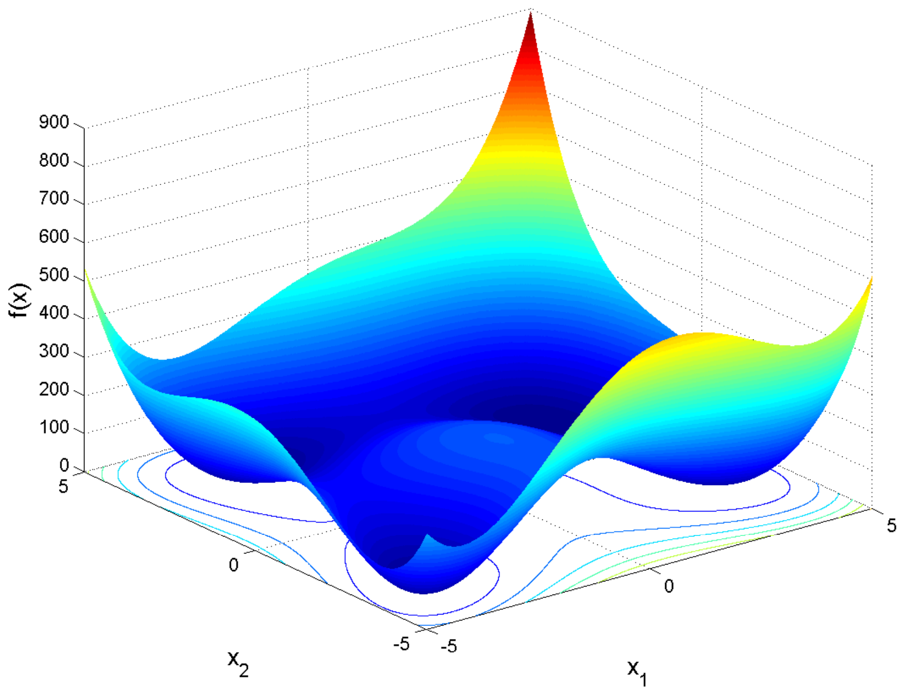



**Fig. 2.11** Plot of the objective function in (2.39).

**Example 15 (Nonlinear Unconstrained Optimization).** Consider the unconstrained nonlinear optimization problem

$$\min_{(x_1,x_2)\in\mathbb{R}^2} f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{2.43}$$

The objective function $f(x)$ is illustrated in Figure 2.11. The contour plot of $f(x)$ is illustrated in Figure 2.12. The local minimizers (blue markers), the saddle points (green markers), and the local maximizer (red marker) are also illustrated in Figure 2.12.

The first derivative, $\nabla f(x)$, of the objective function, $f(x)$, is

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \end{bmatrix} = \begin{bmatrix} 4x_1(x_1^2 + x_2 - 11) + 2(x_1 + x_2^2 - 7) \\ 2(x_1^2 + x_2 - 11) + 4x_2(x_1 + x_2^2 - 7) \end{bmatrix} \tag{2.44}$$

and the second derivative (Hessian), $\nabla^2 f(x)$, of the objective function, $f(x)$, is

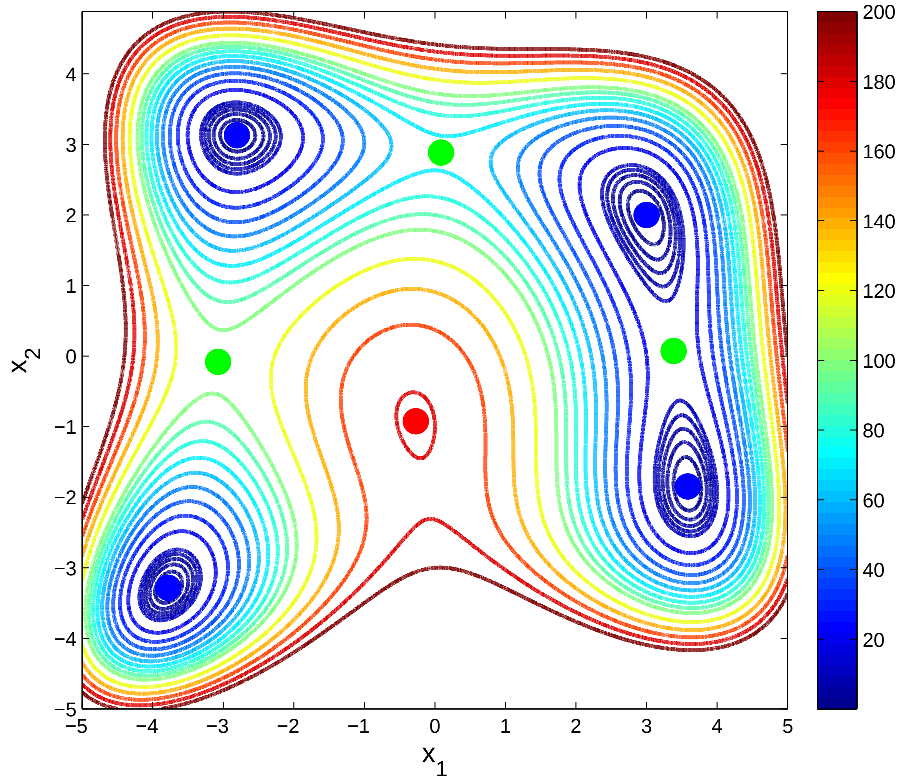



**Fig. 2.12** Contour plot of the objective function in (2.43). The blue markers locate local minimizers, the green markers locate saddle points, and the red marker locate the local maximizer.

$$\begin{aligned}\nabla^2 f(x) &= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) \end{bmatrix} \\ &= \begin{bmatrix} 4(x_1^2 + x_2 - 11) + 8x_1^2 + 2 & 4x_1 + 4x_2 \\ 4x_1 + 4x_2 & 4(x_1 + x_2^2 - 7) + 8x_2^2 + 2 \end{bmatrix}\end{aligned} \tag{2.45}$$

The stationary points, may be found using a nonlinear equation solver such as `fsolve` to solve $\nabla f(x) = 0$. Below we have listed the stationary points, i.e. the points computed such that $\nabla f(x) = 0$. We have computed the Hessian $\nabla^2 f(x)$ in the stationary points and the corresponding eigenvalues, $\lambda$, of the Hessian matrix.

The local minimizers (blue marks in Figure 2.12) are listed below. The Hessian is positive definite, $\nabla^2 f(x) \succ 0$, in these points as all eigenvalues are positive.

$$x = \begin{bmatrix} -2.8051 \\ 3.1313 \end{bmatrix} \quad \nabla^2 f(x) = \begin{bmatrix} 64.9495 & 1.3048 \\ 1.3048 & 80.4409 \end{bmatrix} \quad \lambda = \begin{bmatrix} 64.8404 \\ 80.5501 \end{bmatrix}$$
$$x = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad \nabla^2 f(x) = \begin{bmatrix} 74 & 20 \\ 20 & 34 \end{bmatrix} \quad \lambda = \begin{bmatrix} 25.7157 \\ 82.2843 \end{bmatrix}$$
$$x = \begin{bmatrix} -3.7793 \\ -3.2832 \end{bmatrix} \quad \nabla^2 f(x) = \begin{bmatrix} 116.2655 & -28.2500 \\ -28.2500 & 88.2345 \end{bmatrix} \quad \lambda = \begin{bmatrix} 70.7144 \\ 133.7856 \end{bmatrix}$$
$$x = \begin{bmatrix} 3.5844 \\ -1.8481 \end{bmatrix} \quad \nabla^2 f(x) = \begin{bmatrix} 104.7850 & 6.9452 \\ 6.9452 & 29.3246 \end{bmatrix} \quad \lambda = \begin{bmatrix} 28.6907 \\ 105.4189 \end{bmatrix}$$

The saddle points are the green markers in Figure 2.12. The saddle points are characterized by $\nabla f(x) = 0$ and the Hessian matrix, $\nabla^2 f(x)$, being indefinite, i.e. having both positive and negative eigenvalues. The saddle points are listed below

$$x = \begin{bmatrix} -0.1280 \\ -1.9537 \end{bmatrix} \quad \nabla^2 f(x) = \begin{bmatrix} -49.6184 & -8.3267 \\ -8.3267 & 19.2922 \end{bmatrix} \quad \lambda = \begin{bmatrix} -50.6102 \\ 20.2840 \end{bmatrix}$$
$$x = \begin{bmatrix} 0.0867 \\ 2.8843 \end{bmatrix} \quad \nabla^2 f(x) = \begin{bmatrix} -30.3728 & 11.8837 \\ 11.8837 & 74.1738 \end{bmatrix} \quad \lambda = \begin{bmatrix} -31.7066 \\ 75.5076 \end{bmatrix}$$
$$x = \begin{bmatrix} 3.3852 \\ 0.0739 \end{bmatrix} \quad \nabla^2 f(x) = \begin{bmatrix} 95.8066 & 13.8360 \\ 13.8360 & -12.3939 \end{bmatrix} \quad \lambda = \begin{bmatrix} -14.1352 \\ 97.5479 \end{bmatrix}$$

The local maximizer is

$$x = \begin{bmatrix} -0.2708 \\ -0.9230 \end{bmatrix} \quad \nabla^2 f(x) = \begin{bmatrix} -44.8119 & -4.7755 \\ -4.7755 & -16.8594 \end{bmatrix} \quad \lambda = \begin{bmatrix} -45.6052 \\ -16.0660 \end{bmatrix}$$

and is located at the red marker in Figure 2.12. The gradient is zero at the local maximizer, $\nabla f(x) = 0$, and the Hessian matrix is negative definite, $\nabla^2 f(x) \prec 0$ as all its eigenvalues are negative.

■

**Example 16 (Nonlinear Unconstrained Optimization).** Consider the unconstrained optimization problem

$$\min_{(x_1,x_2)\in\mathbb{R}^2} f(x_1,x_2) = (x_1^2 + x_2 - 1)^2 \tag{2.46}$$

Obviously, the minimizers are located on the curve $\mathcal{S}$

$$\begin{aligned} \mathcal{S} &= \left\{(x_1,x_2) \in \mathbb{R}^2 : \, x_1^2 + x_2 - 1 = 0\right\} \\ &= \left\{(x_1,x_2) \in \mathbb{R}^2 : \, x_2 = 1 - x_1^2\right\} \end{aligned} \tag{2.47}$$

The objective function is plotted in Figure 2.13 and the contour plot of the objective function is illustrated in Figure 2.14. The curve of minimizers, $\mathcal{S}$, is illustrated in Figure 2.14 by the black broken line.

The derivative of the objective function is

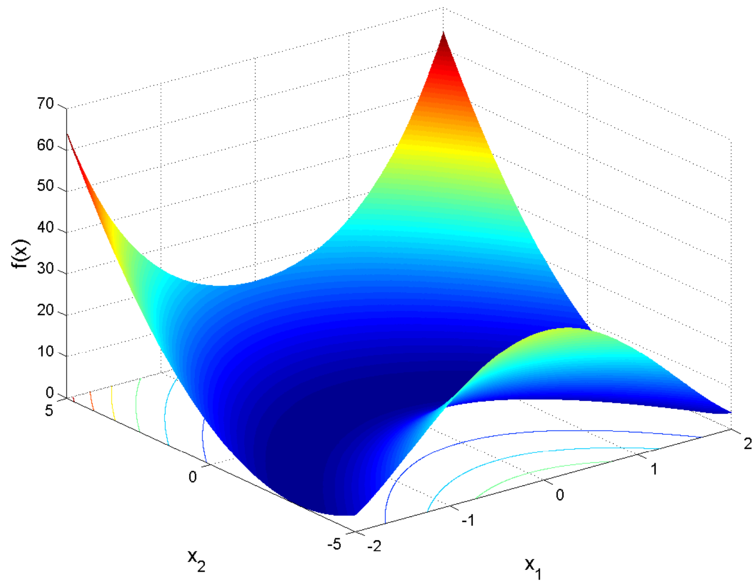



**Fig. 2.13** Plot of the objective function in (2.46).

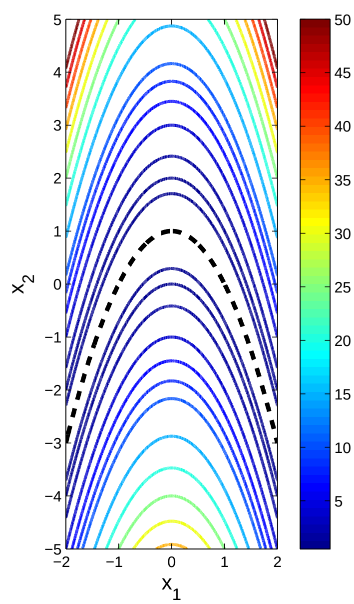



**Fig. 2.14** Contour plot of the objective function in (2.46). The minimizers are located at the broken line.

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \end{bmatrix} = \begin{bmatrix} 4x_1(x_1^2 + x_2 - 1) \\ 2(x_1^2 + x_2 - 1) \end{bmatrix} \tag{2.48}$$

and the second derivative is

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) \end{bmatrix} = \begin{bmatrix} 4(x_1^2 + x_2 - 1) + 8x_1^2 & 4x_1 \\ 4x_1 & 2 \end{bmatrix} \tag{2.49}$$

Points $x$ on the curve of minimizers $\mathcal{S}$ satisfy

$$\nabla f(x) = \begin{bmatrix} 4x_1(x_1^2 + x_2 - 1) \\ 2(x_1^2 + x_2 - 1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \qquad \forall x \in \mathcal{S} \tag{2.50a}$$

$$\nabla^2 f(x) = \begin{bmatrix} 8x_1^2 & 4x_1 \\ 4x_1 & 2 \end{bmatrix} \qquad \forall x \in \mathcal{S} \tag{2.50b}$$

For any value $x \in \mathcal{S}$, the Hessian matrix, $\nabla^2 f(x)$, has a positive eigenvalue and an eigenvalue that is zero, e.g.

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ 1 - x_1^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \nabla^2 f(x) = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} \qquad \lambda = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

Consequently, the Hessian matrix $\nabla^2 f(x)$ is positive semi-definite at the minimizers $x \in \mathcal{S}$, i.e. $\nabla^2 f(x) \succeq 0$ for all $x \in \mathcal{S}$.

■

## 2.2 Constrained Optimization

Consider the constrained optimization problem

$$\min_{x \in \mathcal{D}} \quad f(x) \tag{2.51a}$$

$$s.t. \quad c_i(x) = 0 \qquad i \in \mathcal{E} \tag{2.51b}$$

$$c_i(x) \geq 0 \qquad i \in \mathcal{I} \tag{2.51c}$$

with $\mathcal{D} \subseteq \mathbb{R}^n$ being an open set. The objective function and the constraint functions are assumed to be twice continuously differentiable, i.e. $f \in \mathcal{C}^2(\mathcal{D})$ and $c_i \in \mathcal{C}^2(\mathcal{D})$ for $i \in \mathcal{E} \cup \mathcal{I}$.

The domain $\mathcal{D} \subseteq \mathbb{R}^n$ is an open set defined as the largest open set in $\mathbb{R}^n$ for which both the objective function and all the constraint functions are defined, i.e.

$$\mathcal{D} = \mathcal{D}_f \bigcap_{i \in \mathcal{E} \cup \mathcal{I}} \mathcal{D}_{c_i} \tag{2.52}$$

with $\mathcal{D}_f \subseteq \mathbb{R}^n$ being the domain of $f$ and $\mathcal{D}_{c_i} \subseteq \mathbb{R}^n$ being the domain of $c_i$ for $i \in \mathcal{E} \cup \mathcal{I}$.

The feasible region is

$$\Omega = \{x \in \mathcal{D} : \quad c_i(x) = 0, i \in \mathcal{E}; \quad c_i(x) \geq 0, \, i \in \mathcal{I}\} \tag{2.53}$$

Therefore, the feasible region is the set of points satisfying the constraints. Using the feasible region, $\Omega$, we may express (2.51) in the short form

$$\min_{x \in \Omega} f(x) \tag{2.54}$$

The constrained *global* and *local* minimizer is defined in the following definitions.

**Definition 2.7 (Constrained Global Minimizer).** Let $x^* \in \Omega$. Then $x^*$ is a global minimizer of (2.51) if

$$f(x^*) \leq f(x) \qquad \forall x \in \Omega \tag{2.55}$$

□

**Definition 2.8 (Constrained Local Minimizer).** Let $x^* \in \Omega$. Let a local neighborhood around $x^*$ be defined as

$$\mathcal{N}_\epsilon(x^*) = \{x \in \mathcal{D} : \, \|x - x^*\| < \epsilon\} \tag{2.56}$$

Then $x^*$ is a local minimizer of (2.51) if

$$f(x^*) \leq f(x) \qquad \forall x \in \Omega \cap \mathcal{N}_\epsilon(x^*) \tag{2.57}$$

□

The set of active constraints is an index set that is useful for characterizing minimizers and is also used in the active set algorithms. The set of active constraints is defined in the following definition.

**Definition 2.9 (Set of Active Constraints).** Consider the constrained optimization problem (2.51) and let $\Omega$ be the feasible region. Let $x \in \Omega$. Then the set of active constraints at $x \in \Omega$ is

$$\begin{aligned} \mathcal{A}(x) &= \{i \in \mathcal{E} \cup \mathcal{I} : \, c_i(x) = 0\} \\ &= \mathcal{E} \cup \{i \in \mathcal{I} : \, c_i(x) = 0\} \end{aligned} \tag{2.58}$$

□

## 2.3 The Lagrange Function

The Lagrange function, $\mathcal{L}(x,\lambda)$, is very useful in characterizing minimizers of constrained optimization problems. Consider the constrained optimization problem

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{2.59a}$$

$$s.t. \quad c_i(x) = 0 \qquad i \in \mathcal{E} \tag{2.59b}$$

$$c_i(x) \geq 0 \qquad i \in \mathcal{I} \tag{2.59c}$$

The associated Lagrange function is

$$\begin{aligned}\mathcal{L}(x,\lambda) &= f(x) - \sum_{i\in\mathcal{E}\cup\mathcal{I}} \lambda_i c_i(x) \\ &= f(x) - \lambda' c(x)\end{aligned} \tag{2.60}$$

Each constraint, $c_i(x)$, $i \in \mathcal{E}\cup\mathcal{I}$, has an associated Lagrange multiplier, $\lambda_i$, $i \in \mathcal{E}\cup\mathcal{I}$. This implies that $\lambda$ is a vector of the same dimension as the vector $c(x)$.

## 2.4 First Order Optimality Conditions

The Karush-Kuhn-Tucker (KKT) conditions for constrained optimization problems are the equivalent of the first order optimality conditions for unconstrained optimization problems. They are necessary conditions for a constrained local and global minimizer, but they are not sufficient conditions.

**Proposition 2.10 (Necessary First Order Optimality Conditions).** *Consider the constrained optimization problem* (2.59) *and the associated Lagrange function* (2.60)*. Let $x$ be a constrained local minimizer of* (2.59)*. Then there exists a vector $\lambda$ such that*

$$\nabla_x \mathcal{L}(x,\lambda) = \nabla f(x) - \sum_{i\in\mathcal{E}\cup\mathcal{I}} \lambda_i \nabla c_i(x) = 0 \tag{2.61a}$$

$$c_i(x) = 0 \qquad i \in \mathcal{E} \tag{2.61b}$$

$$c_i(x) \geq 0 \qquad i \in \mathcal{I} \tag{2.61c}$$

$$\lambda_i \geq 0 \qquad i \in \mathcal{I} \tag{2.61d}$$

$$c_i(x) = 0 \quad \vee \quad \lambda_i = 0 \qquad i \in \mathcal{I} \tag{2.61e}$$

$\square$

The necessary first order optimality conditions are also called the KKT conditions and (2.61) is called the stationarity condition, (2.61b)-(2.61c) are called the primal feasibility conditions, (2.61d) is called the dual feasibiliy condition, and (2.61e) is called the complementarity condition.

Let $\mathcal{A}$ denote the set of active constraints, $\mathcal{A} = \mathcal{A}(x)$. Then Proposition 2.10 states

$$\nabla_x \mathcal{L}(x, \lambda) = \nabla f(x) - \sum_{i \in \mathcal{A}} \lambda_i \nabla c_i(x) = 0 \tag{2.62a}$$

$$c_i(x) = 0 \qquad i \in \mathcal{A} \tag{2.62b}$$

and

$$c_i(x) \geq 0 \qquad i \in \mathcal{I} \setminus (\mathcal{I} \cap \mathcal{A}) \tag{2.63a}$$

$$\lambda_i = 0 \qquad i \in \mathcal{I} \setminus (\mathcal{I} \cap \mathcal{A}) \tag{2.63b}$$

$$\lambda_i \geq 0 \qquad i \in \mathcal{I} \cap \mathcal{A} \tag{2.63c}$$

The active-set algorithms use this formulation for solving the constrained optimization problem (2.59). Active set algorithms are combinatorial algorithms in which the algorithm guess the active set, $\mathcal{A}$, solves (2.62) for $x$ and $\lambda_i$ with $i \in \mathcal{A}$. Optimality is checked by checking for primal feasibility, $c_i(x) \geq 0$, $i \in \mathcal{I} \setminus (\mathcal{I} \cap \mathcal{A})$, and dual feasibility, $\lambda_i \geq 0$, $i \in \mathcal{I} \cap \mathcal{A}$. If these conditions are satisfied, a KKT point (a point that satisfies Proposition 2.10) has been found. Otherwise another active set is guessed and the procedure is repeated until a KKT point has been found.

$$F(x, \lambda_{\mathcal{A}}) = \begin{bmatrix} \nabla f(x) - \nabla c_{\mathcal{A}}(x)\lambda_{\mathcal{A}} \\ c_{\mathcal{A}}(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \tag{2.64}$$

In practical computations, the working set, $\mathcal{W} \subseteq \mathcal{A}$, is used instead of the set of active constraints is used to ensure regularity. In this case, the algorithm iterates on the working set, $\mathcal{W}$, and solves

$$F(x, \lambda_{\mathcal{W}}) = \begin{bmatrix} \nabla f(x) - \nabla c_{\mathcal{W}}(x)\lambda_{\mathcal{W}} \\ c_{\mathcal{W}}(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \tag{2.65}$$

with the Lagrange multipliers of the inequality constraints not in the working set fixed to

$$\lambda_i = 0 \quad i \in \mathcal{I} \setminus (\mathcal{I} \cap \mathcal{W}) \tag{2.66}$$

Optimality is checked by verifying

$$c_i(x) = 0 \qquad i \in \mathcal{E} \setminus (\mathcal{E} \cap \mathcal{W}) \tag{2.67a}$$

$$c_i(x) \geq 0 \qquad i \in \mathcal{I} \setminus (\mathcal{I} \cap \mathcal{W}) \tag{2.67b}$$

$$\lambda_i \geq 0 \qquad i \in \mathcal{I} \cap \mathcal{W} \tag{2.67c}$$

If these conditions are not satisfied, the working set, $\mathcal{W}$, is modified and the procedure is repeated.

The conditions

$$c_i(x) \geq 0 \qquad i \in \mathcal{I} \tag{2.68a}$$
$$\lambda_i \geq 0 \qquad i \in \mathcal{I} \tag{2.68b}$$
$$c_i(x) = 0 \quad \vee \quad \lambda_i = 0 \qquad i \in \mathcal{I} \tag{2.68c}$$

can be expressed compactly as

$$c_i(x) \geq 0 \quad \perp \quad \lambda_i \geq 0 \qquad i \in \mathcal{I} \tag{2.69}$$

This condition implies that for each inequality, $i \in \mathcal{I}$, $c_i(x) \geq 0$, $\lambda_i \geq 0$, and either $c_i(x) = 0$ or $\lambda_i = 0$, i.e. $c_i(x)$ and $\lambda_i$ are orthogonal. Consequently, Proposition 2.10 can be expressed as in following corrolary.

**Corollary 2.11 (Necessary First Order Optimality Conditions).** *Consider the constrained optimization problem* (2.59) *and the associated Lagrange function* (2.60). *Let $x$ be a constrained local minimizer of* (2.59). *Then there exists a vector $\lambda$ such that*

$$\nabla_x \mathcal{L}(x, \lambda) = 0 \tag{2.70a}$$
$$c_i(x) = 0 \qquad i \in \mathcal{E} \tag{2.70b}$$
$$c_i(x) \geq 0 \quad \perp \quad \lambda_i \geq 0 \qquad i \in \mathcal{I} \tag{2.70c}$$

*where*

$$\nabla_x \mathcal{L}(x, \lambda) = \nabla f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i \nabla c_i(x) \tag{2.71}$$

□

It is also clear that

$$c_i(x) = 0 \qquad \vee \qquad \lambda_i = 0 \qquad i \in \mathcal{I} \tag{2.72}$$

is equivalent to

$$c_i(x)\lambda_i = 0 \qquad i \in \mathcal{I} \tag{2.73}$$

Consequently, we can reformulate Proposition 2.10 in the following equivalent proposition.

**Proposition 2.12 (Necessary First Order Optimality Conditions).** *Consider the constrained optimization problem* (2.59) *and the associated Lagrange function* (2.60). *Let $x$ be a constrained local minimizer of* (2.59). *Then there exists a vector $\lambda$ such that*

$$\begin{aligned}
&\nabla_x \mathcal{L}(x,\lambda) = \nabla f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i \nabla c_i(x) = 0 && && (2.74a) \\
&c_i(x) = 0 && i \in \mathcal{E} && (2.74b) \\
&c_i(x) \geq 0 && i \in \mathcal{I} && (2.74c) \\
&\lambda_i \geq 0 && i \in \mathcal{I} && (2.74d) \\
&c_i(x)\lambda_i = 0 && i \in \mathcal{I} && (2.74e)
\end{aligned}$$

□

Define slack variables, $s_i$, $i \in \mathcal{I}$, as

$$s_i = c_i(x) \geq 0 \qquad i \in \mathcal{I} \tag{2.75}$$

such that

$$\begin{aligned}
&c_i(x) - s_i = 0 && i \in \mathcal{I} && (2.76a) \\
&s_i \geq 0 && i \in \mathcal{I} && (2.76b)
\end{aligned}$$

Therefore, we may state the optimality conditions in Proposition 2.12 as

$$F(x,s,\lambda) = \begin{bmatrix} \nabla_x \mathcal{L}(x,\lambda) \\ c_{\mathcal{E}}(x) \\ c_{\mathcal{I}}(x) - s \\ S\Lambda_{\mathcal{I}} e \end{bmatrix} = \begin{bmatrix} \nabla f(x) - \nabla c(x)\lambda \\ c_{\mathcal{E}}(x) \\ c_{\mathcal{I}}(x) - s \\ S\Lambda_{\mathcal{I}} e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0 \tag{2.77a}$$

$$s_i, \lambda_i \geq 0 \qquad i \in \mathcal{I} \tag{2.77b}$$

where

$$S = \mathrm{diag}(s) \quad \Lambda_{\mathcal{I}} = \mathrm{diag}(\left[\lambda_i\right]_{i \in \mathcal{I}}) \quad e = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{2.78}$$

such that $S\Lambda_{\mathcal{I}} e = 0$ represents the complementarity conditions $s_i \lambda_i = 0$ for $i \in \mathcal{I}$. Equation (2.77) is used in interior-point algorithms for the solution of (2.59).

## 2.5 Convex Programming and Optimality Conditions

For convex programs, the necessary first order optimality conditions stated in Proposition 2.10 and Proposition 2.12 are also sufficient. Furthermore, such a point is a *global* minimizer.

## 2.6 Second Order Optimality Conditions

For the general nonconvex program, we need second order sufficient conditions to ensure local optimality. This is equivalent to the situation in the unconstrained optimization case. Let $\mathcal{F}(x)$ denote the set of feasible directions in a given feasible point, $x \in \Omega$. This is given as the directions, $h$, satisfying

$$\nabla c_i(x)'h = 0 \qquad \forall i \in \mathcal{E} \tag{2.79a}$$
$$\nabla c_i(x)'h \geq 0 \qquad \forall i \in \mathcal{I} \cap \mathcal{A}(x) \tag{2.79b}$$

The second order necessary conditions for a local minimizer and the sufficient conditions for a local minimizer are stated in the next two propositions.

**Proposition 2.13 (Necessary Second Order Optimality Condition).** *Let $x$ be a local minimizer of* (2.59) *and the pair $(x, \lambda)$ satisfies Proposition 2.10. Assume that the active constraints are linearly independent. Then*

$$h'\nabla^2_{xx}\mathcal{L}(x,\lambda)h \geq 0 \qquad \forall h \in \mathcal{F}(x) \tag{2.80}$$

□

**Proposition 2.14 (Sufficient Second Order Optimality Condition).** *Assume that $x \in \Omega$ and that the pair $(x, \lambda)$ satisfies Proposotion 2.10. Suppose also that*

$$h'\nabla^2_{xx}\mathcal{L}(x,\lambda)h > 0 \qquad \forall h \in \mathcal{F}(x) \tag{2.81}$$

*Then $x$ is a strict local minimizer for* (2.59).

□

## 2.7 Duality

The dual program to the primal constrained optimization program (2.59) is

$$\max_{x,\lambda} \quad \mathcal{L}(x,\lambda) \tag{2.82a}$$
$$s.t. \quad \nabla_x \mathcal{L}(x,\lambda) = 0 \tag{2.82b}$$
$$\lambda_i \geq 0 \qquad i \in \mathcal{I} \tag{2.82c}$$

in which

$$\mathcal{L}(x,\lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x) \tag{2.83a}$$
$$\nabla_x \mathcal{L}(x,\lambda) = \nabla f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i \nabla c_i(x) \tag{2.83b}$$

The first order optimality conditions to the dual program (2.82) are equivalent to the optimality conditions of the primal constrained optimization program (2.59).

## 2.8 Sensitivity of the Optimal Solution

Consider the constrained optimization problem

$$\min_{x \in \mathbb{R}^n} \quad f(x) \tag{2.84a}$$
$$s.t. \quad c_i(x) = p_i \qquad i \in \mathcal{E} \tag{2.84b}$$
$$c_i(x) \geq p_i \qquad i \in \mathcal{I} \tag{2.84c}$$

parameterized by the parameter vector, $p$. The feasible space for this constrained optimization problem is

$$\Omega(p) = \{x \in \mathcal{D} : \quad c_i(x) = p_i,\, i \in \mathcal{E}; \quad c_i(x) \geq p_i,\, i \in \mathcal{I}\} \tag{2.85}$$

and the value function, $V(p)$, as function of the parameter, $p$, is

$$V(p) = \min_{x \in \Omega(p)} f(x) \tag{2.86}$$

The minimizer, $x(p)$, as function of the parameter, $p$, is

$$x(p) = \arg \min_{x \in \Omega(p)} f(x) \tag{2.87}$$

It turns out that the sensitivity, $\partial_p V$, of the value function with respect to the parameter, $p$, is equal to the Lagrange multipliers

$$\frac{\partial V}{\partial p_i}(p) = \lambda_i \qquad i \in \mathcal{E} \cup \mathcal{I} \tag{2.88}$$

This is equivalent to

$$\nabla V(p) = \lambda \tag{2.89}$$

### 2.8.1 The Implicit Function Theorem

Let $x \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$ be related by the function, $F : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$:

$$F(x, p) = 0 \tag{2.90}$$

This defines $x = x(p)$ as an implicit function of $p$ and under regularity conditions expressed in the implicit function theorem, the parameter sensitivity is

$$\nabla_p x(p) = -\nabla_p F(x,p) \left[\nabla_x F(x,p)\right]^{-1} \tag{2.91}$$

This is compact notation for

$$\begin{bmatrix} \frac{\partial x_1}{\partial p_1} & \frac{\partial x_2}{\partial p_1} & \cdots & \frac{\partial x_n}{\partial p_1} \\ \frac{\partial x_1}{\partial p_2} & \frac{\partial x_2}{\partial p_2} & \cdots & \frac{\partial x_n}{\partial p_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial x_1}{\partial p_m} & \frac{\partial x_2}{\partial p_m} & \cdots & \frac{\partial x_n}{\partial p_m} \end{bmatrix} = - \begin{bmatrix} \frac{\partial F_1}{\partial p_1} & \frac{\partial F_2}{\partial p_1} & \cdots & \frac{\partial F_n}{\partial p_1} \\ \frac{\partial F_1}{\partial p_2} & \frac{\partial F_2}{\partial p_2} & \cdots & \frac{\partial F_n}{\partial p_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial F_1}{\partial p_m} & \frac{\partial F_2}{\partial p_m} & \cdots & \frac{\partial F_n}{\partial p_m} \end{bmatrix} \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_2}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_1} \\ \frac{\partial F_1}{\partial x_2} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_n}{\partial x_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial F_1}{\partial x_n} & \frac{\partial F_2}{\partial x_n} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix}^{-1} \tag{2.92}$$

and equivalent to

$$\frac{\partial}{\partial p} x(p) = \left[\frac{\partial}{\partial x} F(x,p)\right]^{-1} \frac{\partial}{\partial p} F(x,p) \tag{2.93}$$

which is compact notation for

$$\begin{bmatrix} \frac{\partial x_1}{\partial p_1} & \frac{\partial x_1}{\partial p_2} & \cdots & \frac{\partial x_1}{\partial p_m} \\ \frac{\partial x_2}{\partial p_1} & \frac{\partial x_2}{\partial p_2} & \cdots & \frac{\partial x_2}{\partial p_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial x_n}{\partial p_1} & \frac{\partial x_n}{\partial p_2} & \cdots & \frac{\partial x_n}{\partial p_m} \end{bmatrix} = - \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial F_1}{\partial p_1} & \frac{\partial F_1}{\partial p_2} & \cdots & \frac{\partial F_1}{\partial p_m} \\ \frac{\partial F_2}{\partial p_1} & \frac{\partial F_2}{\partial p_2} & \cdots & \frac{\partial F_2}{\partial p_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial F_n}{\partial p_1} & \frac{\partial F_n}{\partial p_2} & \cdots & \frac{\partial F_n}{\partial p_m} \end{bmatrix} \tag{2.94}$$

### 2.8.2 Sensitivity of Equality Constrained Optimization

Consider the smooth equality constrained optimization problem

$$\min_{x \in \mathcal{D}} \quad f(x,p) \tag{2.95a}$$

$$s.t. \quad c_i(x,p) = 0 \qquad i \in \mathcal{E} \tag{2.95b}$$

$p \in \mathbb{R}^m$ is a parameter vector. The Lagrangian of (2.95) is

$$\mathcal{L}(x,\lambda;p) = f(x,p) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x,p) = f(x,p) - \lambda' c(x,p) \tag{2.96}$$

and the corresponding KKT conditions can be expressed as

$$\begin{aligned}\nabla_x \mathcal{L}(x,\lambda;p) &= \nabla_x f(x,p) - \sum_{i\in\mathcal{E}} \lambda_i \nabla_x c_i(x,p) \\ &= \nabla_x f(x,p) - \nabla_x c(x,p)\lambda = 0\end{aligned} \tag{2.97a}$$

$$\nabla_\lambda \mathcal{L}(x,\lambda;p) = -c(x,p) = 0 \tag{2.97b}$$

Let $z = [x;\ \lambda]$ such that $z(p) = [x(p);\ \lambda(p)]$. Using this notation, the KKT conditions can be expressed as

$$F(z,p) = \begin{bmatrix} \nabla_x \mathcal{L}(x,\lambda;p) \\ \nabla_\lambda \mathcal{L}(x,\lambda;p) \end{bmatrix} = \begin{bmatrix} \nabla_x f(x,p) - \nabla_x c(x,p)\lambda \\ -c(x,p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \tag{2.98}$$

The implicit function theorem yields

$$\nabla_p z(p) = -\nabla_p F(z,p) \left[\nabla_z F(z,p)\right]^{-1} \tag{2.99}$$

$$\nabla_p F(z,p) = \begin{bmatrix} \nabla^2_{xp} \mathcal{L}(x,\lambda;p) & \nabla^2_{\lambda p} \mathcal{L}(x,\lambda;p) \end{bmatrix} \tag{2.100}$$

$$\begin{aligned}\nabla_z F(z,p) &= \begin{bmatrix} \nabla^2_{xx} \mathcal{L}(x,\lambda;p) & \nabla^2_{\lambda x} \mathcal{L}(x,\lambda;p) \\ \nabla^2_{x\lambda} \mathcal{L}(x,\lambda;p) & \nabla^2_{\lambda\lambda} \mathcal{L}(x,\lambda;p) \end{bmatrix} \\ &= \begin{bmatrix} \nabla^2_{xx} \mathcal{L}(x,\lambda;p) & \nabla^2_{\lambda x} \mathcal{L}(x,\lambda;p) \\ \nabla^2_{x\lambda} \mathcal{L}(x,\lambda;p) & 0 \end{bmatrix}\end{aligned} \tag{2.101}$$

with

$$\nabla^2_{xx} \mathcal{L}(x,\lambda;p) = \nabla^2_{xx} f(x,p) - \sum_{i\in\mathcal{E}} \lambda_i \nabla^2_{xx} c_i(x,p) \tag{2.102a}$$

$$\nabla^2_{x\lambda} \mathcal{L}(x,\lambda;p) = -\nabla_x c(x,p)' \tag{2.102b}$$

$$\nabla^2_{xp} \mathcal{L}(x,\lambda;p) = \nabla^2_{xp} f(x,p) - \sum_{i\in\mathcal{E}} \lambda_i \nabla^2_{xp} c_i(x,p) \tag{2.102c}$$

$$\nabla^2_{\lambda x} \mathcal{L}(x,\lambda;p) = -\nabla_x c(x,p) \tag{2.102d}$$

$$\nabla^2_{\lambda\lambda} \mathcal{L}(x,\lambda;p) = 0 \tag{2.102e}$$

$$\nabla^2_{\lambda p} \mathcal{L}(x,\lambda;p) = -\nabla_p c(x,p) \tag{2.102f}$$

Consequently, the parameter sensitivity of the optimal solution is

$$\begin{bmatrix} \nabla_p x(p) & \nabla_p \lambda(p) \end{bmatrix} = -\begin{bmatrix} \nabla^2_{xp} \mathcal{L}(x,\lambda;p) & -\nabla_p c(x,p) \end{bmatrix} \begin{bmatrix} \nabla^2_{xx} \mathcal{L}(x,\lambda;p) & -\nabla_x c(x,p) \\ -\nabla_x c(x,p)' & 0 \end{bmatrix}^{-1} \tag{2.103}$$

Note that typically the KKT-matrix has already been factorized in obtained the solution. Consequently, the linear system (2.103) can be solved efficiently by back-substitutions without factorizing the KKT matrix.

The solution $x(p)$ and $\lambda(p)$ may be approximated by

$$x(p) \approx x(p_0) + \nabla_p x(p_0)'(p - p_0) \tag{2.104}$$

$$\lambda(p) \approx \lambda(p_0) + \nabla_p \lambda(p_0)'(p - p_0) \tag{2.105}$$

and the corresponding value function is approximately

$$\begin{aligned} V(p) &= f(x(p), p) \\ &\approx f(x(p_0), p_0) + [\nabla_p x(p_0) \nabla_x f(x(p_0), p_0) + \nabla_p f(x(p_0), p_0)]' (p - p_0) \end{aligned} \tag{2.106}$$

# Part III
# Appendices

# Appendix A
# Derivatives

In this chapter we present the notation used for derivatives and some results related to their applications.

## A.1 Objective Function

We consider the function $f : \mathbb{R}^n \mapsto \mathbb{R}$ and denote it $f(x)$ for $x \in \mathbb{R}^n$. We assume that $f$ is continuous and twice differentiable, i.e. $f \in \mathcal{C}^2(\mathbb{R}^n)$.

The gradient, $\nabla f(x)$, of $f(x)$ is

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix} \tag{A.1}$$

Let $f_x(x) = \frac{\partial f}{\partial x}(x)$ be the transpose of the gradient, $\nabla f(x)$. That is

$$f_x(x) = \frac{\partial f}{\partial x}(x) = \nabla f(x)' = \left[ \frac{\partial f}{\partial x_1}(x) \; \frac{\partial f}{\partial x_1}(x) \; \dots \; \frac{\partial f}{\partial x_n}(x) \right] \tag{A.2}$$

The Hessian matrix, $\nabla^2 f(x)$, is

$$
\begin{aligned}
\nabla^2 f(x) &= \left[\nabla \tfrac{\partial f}{\partial x_1}(x)\ \nabla \tfrac{\partial f}{\partial x_2}(x)\ \ldots\ \nabla \tfrac{\partial f}{\partial x_n}(x)\ \right] \\
&= \begin{bmatrix}
\frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_2}\right)(x) & \ldots & \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_n}\right)(x) \\
\frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_2}\right)(x) & \ldots & \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_n}\right)(x) \\
\vdots & \vdots & & \vdots \\
\frac{\partial}{\partial x_n}\left(\frac{\partial f}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_n}\left(\frac{\partial f}{\partial x_2}\right)(x) & \ldots & \frac{\partial}{\partial x_n}\left(\frac{\partial f}{\partial x_n}\right)(x)
\end{bmatrix} \\
&= \begin{bmatrix}
\frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \ldots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\
\frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_2}(x) & \ldots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\
\vdots & \vdots & & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \ldots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x)
\end{bmatrix}
\end{aligned} \tag{A.3}
$$

The Hessian matrix, $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$, is symmetric.

## A.2 Constraint Function

Let $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ be a vector function

$$
c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \\ \vdots \\ c_m(x) \end{bmatrix} \tag{A.4}
$$

The gradient of $c(x)$ is

$$
\begin{aligned}
\nabla c(x) &= \left[\nabla c_1(x)\ \nabla c_2(x)\ \ldots\ \nabla c_m(x)\right] \\
&= \begin{bmatrix}
\frac{\partial c_1}{\partial x_1}(x) & \frac{\partial c_2}{\partial x_1}(x) & \ldots & \frac{\partial c_m}{\partial x_1}(x) \\
\frac{\partial c_1}{\partial x_2}(x) & \frac{\partial c_2}{\partial x_2}(x) & \ldots & \frac{\partial c_m}{\partial x_2}(x) \\
\vdots & \vdots & & \vdots \\
\frac{\partial c_1}{\partial x_n}(x) & \frac{\partial c_2}{\partial x_n}(x) & \ldots & \frac{\partial c_m}{\partial x_n}(x)
\end{bmatrix}
\end{aligned} \tag{A.5}
$$

The Jacobian of $c(x)$ is denoted $J(x) = c_x(x) = \frac{\partial c}{\partial x}(x)$, and is the transpose of the gradient $\nabla c(x)$. Consequently

$$
J(x) = c_x(x) = \frac{\partial c}{\partial x}(x) = \nabla c(x)' = \begin{bmatrix}
\frac{\partial c_1}{\partial x_1}(x) & \frac{\partial c_1}{\partial x_2}(x) & \ldots & \frac{\partial c_1}{\partial x_n}(x) \\
\frac{\partial c_2}{\partial x_1}(x) & \frac{\partial c_2}{\partial x_2}(x) & \ldots & \frac{\partial c_2}{\partial x_n}(x) \\
\vdots & \vdots & & \vdots \\
\frac{\partial c_m}{\partial x_1}(x) & \frac{\partial c_m}{\partial x_2}(x) & \ldots & \frac{\partial c_m}{\partial x_n}(x)
\end{bmatrix} \tag{A.6}
$$

$$\nabla^2 c_i(x) = \begin{bmatrix} \frac{\partial^2 c_i}{\partial x_1 \partial x_1}(x) & \frac{\partial^2 c_i}{\partial x_1 \partial x_2}(x) & \dots & \frac{\partial^2 c_i}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 c_i}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 c_i}{\partial x_2 \partial x_2}(x) & \dots & \frac{\partial^2 c_i}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 c_i}{\partial x_n \partial x_1}(x) & \frac{\partial^2 c_i}{\partial x_n \partial x_2}(x) & \dots & \frac{\partial^2 c_i}{\partial x_n \partial x_n}(x) \end{bmatrix} \quad i \in \{1, 2, \dots, m\} \tag{A.7}$$

## A.3 Taylor Approximation

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a continuous function that is twice differentiable. Then the first order Taylor approximation of $f(x)$ in the point $x_0 \in \mathbb{R}^n$ is

$$f(x) \approx f(x_0) + \nabla f(x_0)'(x - x_0) \tag{A.8}$$

Similarly, the second order Taylor approximation is

$$f(x) \approx f(x_0) + \nabla f(x_0)'(x - x_0) + \frac{1}{2}(x - x_0)'\nabla^2 f(x_0)(x - x_0) \tag{A.9}$$

It is important to notice that with the definition of $\nabla f(x)$, we transponse $\nabla f(x_0)$ in the first order term of the Taylor approximations.

Let $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ be a continuous differentiable function. Then the first order Taylor approximation around $x_0 \in \mathbb{R}^n$ is

$$\begin{aligned} c(x) &\approx c(x_0) + \nabla c(x_0)'(x - x_0) \\ &= c(x_0) + c_x(x_0)(x - x_0) \\ &= c(x_0) + J(x_0)(x - x_0) \end{aligned} \tag{A.10}$$

## A.4 Examples

In this section we provide some examples that are useful in the study of numerical optimization algorithms. Firstly, we provide the derivatives of linear function. Secondly, we provide the derivatives of an example used repeatedly in the study of numerical optimization algorithms.

### *A.4.1 Linear Scalar Function*

Consider the linear scalar function, $f : \mathbb{R}^3 \mapsto \mathbb{R}$, defined as

$$
\begin{aligned}
f(x) = g_1 x_1 + g_2 x_2 + g_3 x_3 &= \begin{bmatrix} g_1 \ g_2 \ g_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\
&= \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}' \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = g'x
\end{aligned}
\tag{A.11}
$$

with

$$
x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad g = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}
$$

The gradient, $\nabla f(x)$, is

$$
\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \frac{\partial f}{\partial x_3}(x) \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = g \tag{A.12}
$$

Trivially, the Hessian matrix, $\nabla^2 f(x)$ is

$$
\begin{aligned}
\nabla^2 f(x) &= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_3}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_2}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_3}(x) \\ \frac{\partial^2 f}{\partial x_3 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_3 \partial x_2}(x) & \frac{\partial^2 f}{\partial x_3 \partial x_3}(x) \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_2}\right)(x) & \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_3}\right)(x) \\ \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_2}\right)(x) & \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_3}\right)(x) \\ \frac{\partial}{\partial x_3}\left(\frac{\partial f}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_3}\left(\frac{\partial f}{\partial x_2}\right)(x) & \frac{\partial}{\partial x_3}\left(\frac{\partial f}{\partial x_3}\right)(x) \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 0
\end{aligned}
\tag{A.13}
$$

## *A.4.2 Linear Vector Function*

Consider the linear vector function, $c : \mathbb{R}^3 \mapsto \mathbb{R}^3$, defined by

$$
\begin{aligned}
c(x) = \begin{bmatrix} c_1(x_1, x_2, x_3) \\ c_2(x_1, x_2, x_3) \\ c_3(x_1, x_2, x_3) \end{bmatrix} &= \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \end{bmatrix} \\
&= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = Ax
\end{aligned}
\tag{A.14}
$$

with

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Then

$$\nabla c(x) = \begin{bmatrix} \frac{\partial c_1}{\partial x_1}(x) & \frac{\partial c_2}{\partial x_1}(x) & \frac{\partial c_3}{\partial x_1}(x) \\ \frac{\partial c_1}{\partial x_2}(x) & \frac{\partial c_2}{\partial x_2}(x) & \frac{\partial c_3}{\partial x_2}(x) \\ \frac{\partial c_1}{\partial x_3}(x) & \frac{\partial c_2}{\partial x_3}(x) & \frac{\partial c_3}{\partial x_3}(x) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} = A' \tag{A.15}$$

### *A.4.3 Quadratic Scalar Function*

Consider the quadratic scalar function, $f : \mathbb{R}^3 \mapsto \mathbb{R}$, defined by

$$\begin{aligned} f(x) = \frac{1}{2}x'Hx &= \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}' \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= \frac{1}{2}h_{11}x_1^2 + \frac{1}{2}(h_{12} + h_{21})x_1x_2 + \frac{1}{2}(h_{13} + h_{31})x_1x_3 \\ &\quad + \frac{1}{2}h_{22}x_2^2 + \frac{1}{2}(h_{23} + h_{32})x_2x_3 + \frac{1}{2}h_{33}x_3^2 \end{aligned} \tag{A.16}$$

in which

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \qquad H' = H$$

Note that $H$ is a symmetric matrix, i.e. $H = H'$.

The gradient, $\nabla f(x)$, of $f(x)$ is

$$\begin{aligned} \nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \frac{\partial f}{\partial x_3}(x) \end{bmatrix} &= \begin{bmatrix} h_{11}x_1 + \frac{1}{2}(h_{12} + h_{21})x_2 + \frac{1}{2}(h_{13} + h_{31})x_3 \\ \frac{1}{2}(h_{12} + h_{21})x_1 + h_{22}x_2 + \frac{1}{2}(h_{23} + h_{32})x_3 \\ \frac{1}{2}(h_{13} + h_{31})x_1 + \frac{1}{2}(h_{23} + h_{32})x_2 + h_{33}x_3 \end{bmatrix} \\ &= \begin{bmatrix} h_{11}x_1 + h_{12}x_2 + h_{13}x_3 \\ h_{21}x_1 + h_{22}x_2 + h_{23}x_3 \\ h_{31}x_1 + h_{32}x_2 + h_{33}x_3 \end{bmatrix} \\ &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = Hx \end{aligned} \tag{A.17}$$

The symmetry of $H$ has been used in this derivation, e.g. $\frac{1}{2}(h_{12} + h_{21}) = \frac{1}{2}(h_{12} + h_{12}) = h_{12} = h_{21}$.

Remember that the gradient of a linear vector function $c(x) = Ax$ is $\nabla c(x) = A'$. The Hessian $\nabla^2 f(x)$ is the gradient of $\nabla f(x) = Hx$. Consequently, the Hessian, $\nabla^2 f(x)$, is

$$\nabla^2 f(x) = \nabla\left(\nabla f(x)\right) = \nabla_x\left(Hx\right) = H' = H \tag{A.18}$$

### *A.4.4 Quadratic Function II*

Consider the quadratic function

$$f(x) = \frac{1}{2}x'Hx + g'x + \rho \tag{A.19}$$

The gradient and the Hessian of this function are

$$\nabla f(x) = Hx + g \tag{A.20}$$

$$\nabla^2 f(x) = H \tag{A.21}$$

### *A.4.5 Nonlinear Function*

Consider the nonlinear function

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{A.22}$$

with $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$. The gradient $\nabla f(x)$ is

$$\begin{aligned}\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \end{bmatrix} &= \begin{bmatrix} 2(x_1^2 + x_2 - 11)2x_1 + 2(x_1 + x_2^2 - 7) \\ 2(x_1^2 + x_2 - 11) + 2(x_1 + x_2^2 - 7)2x_2 \end{bmatrix} \\ &= \begin{bmatrix} 4x_1(x_1^2 + x_2 - 11) + 2(x_1 + x_2^2 - 7) \\ 2(x_1^2 + x_2 - 11) + 4x_2(x_1 + x_2^2 - 7) \end{bmatrix}\end{aligned} \tag{A.23}$$

and the Hessian matrix, $\nabla^2 f(x)$, is

$$\begin{aligned}\nabla^2 f(x) &= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_2}(x) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_2}\right)(x) \\ \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_2}\right)(x) \end{bmatrix} \\ &= \begin{bmatrix} 4(x_1^2 + x_2 - 11) + 8x_1^2 + 2 & 4x_1 + 4x_2 \\ 4x_1 + 4x_2 & 4(x_1 + x_2^2 - 7) + 8x_2^2 + 2 \end{bmatrix}\end{aligned} \tag{A.24}$$

$\nabla^2 f(x)$ is symmetric as required.

### A.4.6 Nonlinear Vector Functions

Consider the vector function

$$c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \end{bmatrix} = \begin{bmatrix} (x_1+2)^2 - x_2 \\ -4x_1 + 10x_2 \end{bmatrix} \tag{A.25}$$

The gradient, $\nabla c(x)$, is

$$\nabla c(x) = \begin{bmatrix} \frac{\partial c_1}{\partial x_1}(x) & \frac{\partial c_2}{\partial x_1}(x) \\ \frac{\partial c_1}{\partial x_2}(x) & \frac{\partial c_2}{\partial x_2}(x) \end{bmatrix} = \begin{bmatrix} 2(x_1+2) & -4 \\ -1 & 10 \end{bmatrix} \tag{A.26}$$

The Hessian matrix, $\nabla c_1(x)$, of $c_1(x)$ is

$$\begin{aligned} \nabla^2 c_1(x) &= \begin{bmatrix} \frac{\partial^2 c_1}{\partial x_1 \partial x_1}(x) & \frac{\partial^2 c_1}{\partial x_1 \partial x_2}(x) \\ \frac{\partial^2 c_1}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 c_1}{\partial x_2 \partial x_2}(x) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial c_1}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_1}\left(\frac{\partial c_1}{\partial x_2}\right)(x) \\ \frac{\partial}{\partial x_2}\left(\frac{\partial c_1}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_2}\left(\frac{\partial c_1}{\partial x_2}\right)(x) \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{A.27a}$$

and the Hessian matrix, $\nabla c_2(x)$, of $c_2(x)$ is

$$\begin{aligned} \nabla^2 c_2(x) &= \begin{bmatrix} \frac{\partial^2 c_2}{\partial x_1 \partial x_1}(x) & \frac{\partial^2 c_2}{\partial x_1 \partial x_2}(x) \\ \frac{\partial^2 c_2}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 c_2}{\partial x_2 \partial x_2}(x) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial c_2}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_1}\left(\frac{\partial c_2}{\partial x_2}\right)(x) \\ \frac{\partial}{\partial x_2}\left(\frac{\partial c_2}{\partial x_1}\right)(x) & \frac{\partial}{\partial x_2}\left(\frac{\partial c_2}{\partial x_2}\right)(x) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{A.27b}$$

## A.5 Composite Function and Chain Rule

We provide the chain rule for the composite functions $F(x) = g(f(x))$ and $F(x) = g(x, f(x))$.

### A.5.1 Result I: $F(x) = g(f(x))$

We consider the composite function $F(x) = g(f(x))$. This function may be viewed as the composition

$$f : \mathbb{R}^n \mapsto \mathbb{R}^m \qquad y = f(x) \tag{A.28a}$$
$$g : \mathbb{R}^m \mapsto \mathbb{R}^k \qquad z = g(y) \tag{A.28b}$$
$$F : \mathbb{R}^n \mapsto \mathbb{R}^k \qquad z = F(x) = g(f(x)) \tag{A.28c}$$

The chain rule provides a formula for computation of the Jacobian and the gradient of $F(x)$. The Jacobian is

$$F_x(x) = g_y(f(x))f_x(x) \tag{A.29}$$

and the gradient is

$$\nabla F(x) = \nabla f(x)\nabla g(f(x)) \tag{A.30}$$

### *A.5.2 Result II: $F(x) = g(x, f(x))$*

We consider the composite function $F(x) = g(x, f(x))$. This function may be viewed as the composition

$$f : \mathbb{R}^n \mapsto \mathbb{R}^m \qquad y = f(x) \tag{A.31a}$$
$$g : \mathbb{R}^m \mapsto \mathbb{R}^k \qquad z = g(x, y) \tag{A.31b}$$
$$F : \mathbb{R}^n \mapsto \mathbb{R}^k \qquad z = F(x) = g(x, f(x)) \tag{A.31c}$$

The chain rule provides a formula for computation of the Jacobian and the gradient of $F(x)$. The Jacobian is

$$F_x(x) = g_x(x, f(x)) + g_y(x, f(x))f_x(x) \tag{A.32}$$

and the gradient is

$$\nabla F(x) = \nabla_x g(x, f(x)) + \nabla f(x)\nabla_y g(x, f(x)) \tag{A.33}$$

## A.6 Finite Difference Numerical Approximation

### *A.6.1 Univariate Scalar Function*

Consider the univariate scalar function $f : \mathbb{R} \mapsto \mathbb{R}$. Use $f(x)$ to denote this function. The gradient of this function may be approximated by the difference approximations

$$\text{forward:} \quad \frac{df}{dx}(x) \approx \frac{f(x+h) - f(x)}{(x+h) - x} = \frac{f(x+h) - f(x)}{h} \tag{A.34a}$$

$$\text{backward:} \quad \frac{df}{dx}(x) \approx \frac{f(x) - f(x-h)}{x - (x-h)} = \frac{f(x) - f(x-h)}{h} \tag{A.34b}$$

$$\text{central:} \quad \frac{df}{dx}(x) \approx \frac{f(x+h) - f(x-h)}{(x+h) - (x-h)} = \frac{f(x+h) - f(x-h)}{2h} \tag{A.34c}$$

Note that the central difference approximation can be regarded as an average of the forward and backward difference approximation

$$\begin{aligned} \frac{df}{dx}(x) &\approx \frac{1}{2}\left(\frac{f(x+h) - f(x)}{h} + \frac{f(x) - f(x-h)}{h}\right) \\ &= \frac{f(x+h) - f(x-h)}{2h} \end{aligned} \tag{A.35}$$

An approximation of the 2nd derivative is

$$\frac{d^2 f}{dx^2}(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{A.36}$$

We may arrive at this approximation in several ways. One way to arrive at it is

$$\begin{aligned} \frac{d^2 f}{dx^2}(x) &\approx \frac{\frac{df}{dx}(x+h) - \frac{df}{dx}(x)}{h} \approx \frac{\frac{f(x+h)-f(x)}{h} - \frac{f(x)-f(x-h)}{h}}{h} \\ &= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \end{aligned} \tag{A.37}$$

## *A.6.2 Multivariate Scalar Function*

Consider the multivariate scalar function $f : \mathbb{R}^n \mapsto \mathbb{R}$ denoted $f(x)$.

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix} \tag{A.38}$$

Let

$$e_i = \begin{bmatrix} 0 \\ \vdots \end{bmatrix} \tag{A.39}$$

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x+he_i)-f(x)}{h} \tag{A.40}$$

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x)-f(x-he_i)}{h} \tag{A.41}$$

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x+he_i)-f(x-he_i)}{2h} \tag{A.42}$$

### A.6.3 Multivariate Vector Function

Consider the multivariate vector function $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ denoted

$$c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \\ \vdots \\ c_m(x) \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n \tag{A.43}$$

### A.6.4 Numerical Jacobian - Matlab Implementations

**Algorithm 1** Jacobian by Forward Finite Difference Approximation

```
function J = JacobianFDforward(fun,c,x,varargin)

n = length(x);
m = length(c);
J = zeros(m,n);

for i=1:n
    e = sqrt(eps*max(abs(x(i)),1.0));
    xtmp = x;
    xtmp(i) = x(i)+e;
    h = xtmp(i)-x(i);
    ctmp = feval(fun,xtmp,varargin{:});
    J(:,i) = (ctmp-c)/h;
end
```

**Algorithm 2** Jacobian by Backward Finite Difference Approximation

```
function J = JacobianFDbackward(fun,c,x,varargin)

n = length(x);
m = length(c);
J = zeros(m,n);

for i=1:n
    e = sqrt(eps*max(abs(x(i)),1.0));
    xtmp = x;
    xtmp(i) = x(i)-e;
    h = x(i)-xtmp(i);
    ctmp = feval(fun,xtmp,varargin{:});
    J(:,i) = (c-ctmp)/h;
end
```

### *A.6.5 Numerical Hessian - Matlab Implementations*

## A.7 Exercises

### *Problem 1: Gradient and Hessian of Multivariate Scalar Function*

Consider the function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ defined by

$$f(x) = x_1^2 - 2x_1 + 3x_1x_2 + 4x_2^3 \tag{A.44}$$

1. Make a contour plot and approximately locate the minima.
2. Derive an analytical expression for the gradient, $\nabla f(x)$.
3. Derive an analytical expression for the Hessian, $\nabla^2 f(x)$
4. Implement a Matlab function for computation of $f(x)$, $\nabla f(x)$, and $\nabla^2 f(x)$. The function must have the interface `function [f,df,d2f] = FunEx1(x)`.
5. Use finite difference approximations to verify your expressions in the point $x_0 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$.

### *Problem 2: Rosenbrock function*

$$f(x) = p_1(x_2 - x_1^2)^2 + p_2(1 - x_1)^2 \quad p = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 100 \\ 1 \end{bmatrix} \tag{A.45}$$

1. Make a contour plot and approximately locate the minima.
2. Derive an analytical expression for the gradient $\nabla f(x)$.

3. Derive an analytical expression for the Hessian $\nabla^2 f(x)$.
4. Implement a Matlab function for computation of $f(x)$, $\nabla f(x)$ and $\nabla^2 f(x)$. It must have the interface `function [f,df,d2f] = Rosenbrock(x,p)`

## *Problem 3: Derivatives of a Multivariate Vector Function*

Consider the function $c : \mathbb{R}^2 \mapsto \mathbb{R}^2$ defined by

$$c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \end{bmatrix} = \begin{bmatrix} c_1(x_1, x_2) \\ c_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} e^{x_1} - x_2 \\ x_1^2 - 2x_2 \end{bmatrix} \tag{A.46}$$

1. Derive an analytical expression for the gradient $\nabla c(x)$.
2. Derive an analytical expression for the Jacobian $J(x) = c_x(x) = \frac{\partial c}{\partial x}(x)$.
3. Implement a Matlab function for computation of $c(x)$, $\nabla c(x)$, and $\nabla^2 c_i(x)$ for $i = \{1, 2\}$. The function must have the interface `function [c,dc,d2c] = FunEx2(x)`. Hint: Use a 3-dimensional array to store $\nabla^2 c(x)$, i.e. $\nabla^2 c_1(x) =$ `d2c(:,:,1)` and $\nabla^2 c_2(x) =$ `d2c(:,:,2)`.
4. Implement a Matlab function for computation of $c(x)$ and $J(x)$. The function must have the interface `function [c,J] = FunJacEx2(x)`.
5. Use finite difference approximations to verify your expressions in the point $x_0 = \begin{bmatrix} 1.7 \\ 2.1 \end{bmatrix}$.