OPTIMIZATION

An introduction

A. Astolfi

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Optimization is the act of achieving the best possible result under given circumstances. In design, construction, maintenance, ..., engineers have to take decisions. The goal of all such decisions is either to minimize effort or to maximize benefit.
The effort or the benefit can be usually expressed as a function of certain design variables. Hence, optimization is the process of finding the conditions that give the maximum or the minimum value of a function.
It is obvious that if a point $x^\star$ corresponds to the minimum value of a function $f(x)$, the same point corresponds to the maximum value of the function $-f(x)$. Thus, optimization can be taken to be minimization.
There is no single method available for solving all optimization problems efficiently. Hence, a number of methods have been developed for solving different types of problems.
Optimum seeking methods are also known as mathematical programming techniques, which are a branch of operations research. Operations research is *coarsely* composed of the following areas.

- Mathematical programming methods. These are useful in finding the minimum of a function of several variables under a prescribed set of constraints.
- Stochastic process techniques. These are used to analyze problems which are described by a set of random variables of known distribution.
- Statistical methods. These are used in the analysis of experimental data and in the construction of empirical models.

These lecture notes deal mainly with the theory and applications of mathematical programming methods. Mathematical programming is a vast area of mathematics and engineering. It includes

- calculus of variations and optimal control;
- linear, quadratic and non-linear programming;
- geometric programming;
- integer programming;
- network methods (PERT);
- game theory.

The existence of optimization can be traced back to Newton, Lagrange and Cauchy. The development of differential methods for optimization was possible because of the contribution of Newton and Leibnitz. The foundations of the calculus of variations were laid by Bernoulli, Euler, Lagrange and Weierstrasse. Constrained optimization was first studied by Lagrange and the notion of descent was introduced by Cauchy.

Despite these early contributions, very little progress was made till the 20th century, when computer power made the implementation of optimization procedures possible and this in turn stimulated further research methods.
The major developments in the area of numerical methods for unconstrained optimization have been made in the UK. These include the development of the simplex method (Dantzig, 1947), the principle of optimality (Bellman, 1957), necessary and sufficient conditions of optimality (Kuhn and Tucker, 1951).
Optimization in its broadest sense can be applied to solve any engineering problem, *e.g.*

- design of aircraft for minimum weight;
- optimal (minimum time) trajectories for space missions;
- minimum weight design of structures for earthquake;
- optimal design of electric networks;
- optimal production planning, resources allocation, scheduling;
- shortest route;
- design of optimum pipeline networks;
- minimum processing time in production systems;
- optimal control.

1.2 Statement of an optimization problem

An optimization, or a mathematical programming problem can be stated as follows.
Find

$$x = (x^1, x^2, ..., x^n)$$

which minimizes

$$f(x)$$

subject to the constraints

$$g_j(x) \leq 0 \tag{1.1}$$

for $j = 1, \ldots, m$, and

$$l_j(x) = 0 \tag{1.2}$$

for $j = 1, \ldots, p$.
The variable x is called the design vector, $f(x)$ is the objective function, $g_j(x)$ are the inequality constraints and $l_j(x)$ are the equality constraints. The number of variables n and the number of constraints $p + m$ need not be related. If $p + m = 0$ the problem is called an unconstrained optimization problem.

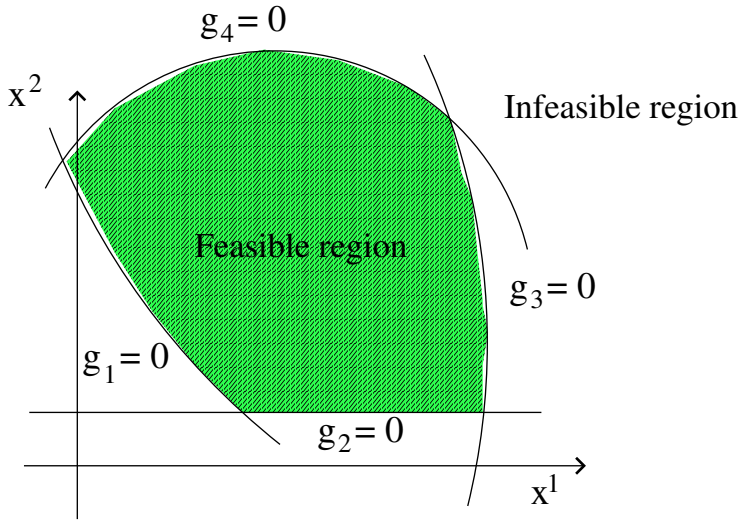

Figure 1.1: Feasible region in a two-dimensional design space. Only inequality constraints are present.

1.2.1 Design vector

Any system is described by a set of quantities, some of which are viewed as variables during the design process, and some of which are preassigned parameters or are imposed by the *environment*. All the quantities that can be treated as variables are called design or decision variables, and are collected in the design vector x.

1.2.2 Design constraints

In practice, the design variables cannot be selected arbitrarily, but have to satisfy certain requirements. These restrictions are called design constraints. Design constraints may represent limitation on the performance or behaviour of the system or physical limitations. Consider, for example, an optimization problem with only inequality constraints, *i.e.* $g_j(x) \leq 0$. The set of values of x that satisfy the equations $g_j(x) = 0$ forms a hypersurface in the design space, which is called constraint surface. In general, if n is the number of design variables, the constraint surface is an $n-1$ dimensional surface. The constraint surface divides the design space into two regions: one in which $g_j(x) < 0$ and one in which $g_j(x) > 0$. The points x on the constraint surface satisfy the constraint critically, whereas the points x such that $g_j(x) > 0$, for some j, are infeasible, *i.e.* are unacceptable, see Figure 1.1.

1.2.3 Objective function

The classical design procedure aims at finding an acceptable design, *i.e.* a design which satisfies the constraints. In general there are several acceptable designs, and the purpose

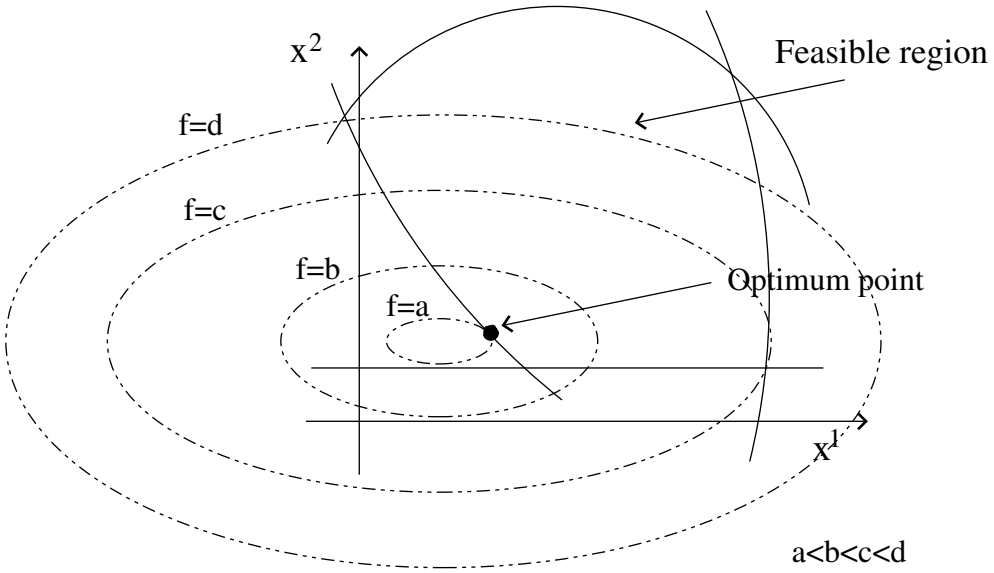

Figure 1.2: Design space, objective functions surfaces, and optimum point.

of the optimization is to single out the best possible design. Thus, a criterion has to be selected for comparing different designs. This criterion, when expressed as a function of the design variables, is known as objective function. The objective function is in general specified by physical or economical considerations. However, the selection of an objective function is not trivial, because what is the optimal design with respect to a certain criterion may be unacceptable with respect to another criterion. Typically there is a trade off performance–cost, or performance–reliability, hence the selection of the objective function is one of the most important decisions in the whole design process. If more than one criterion has to be satisfied we have a multiobjective optimization problem, that may be approximately solved considering a cost function which is a weighted sum of several objective functions.

Given an objective function $f(x)$, the locus of all points x such that $f(x) = c$ forms a hypersurface. For each value of c there is a different hypersurface. The set of all these surfaces are called objective function surfaces.

Once the objective function surfaces are drawn, together with the constraint surfaces, the optimization problem can be easily solved, at least in the case of a two dimensional decision space, as shown in Figure 1.2. If the number of decision variables exceeds two or three, this graphical approach is not viable and the problem has to be solved as a mathematical problem. Note however that more general problems have similar geometrical properties of two or three dimensional problems.

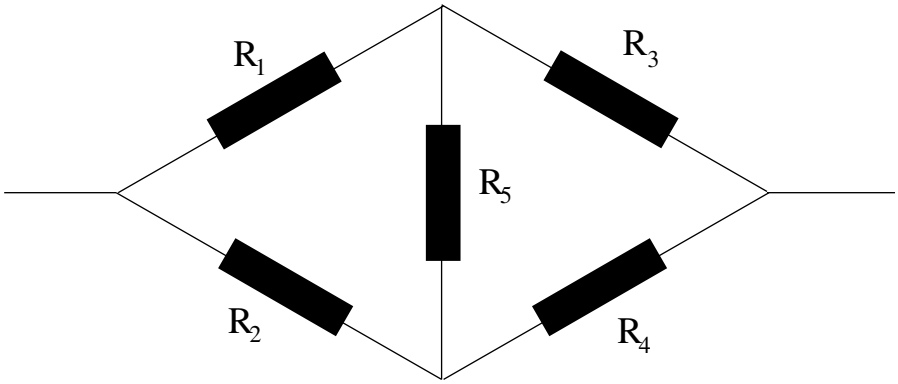

Figure 1.3: Electrical bridge network.

1.3 Classification of optimization problems

Optimization problem can be classified in several ways.

- Existence of constraints. An optimization problem can be classified as a constrained or an unconstrained one, depending upon the presence or not of constraints.
- Nature of the equations. Optimization problems can be classified as linear, quadratic, polynomial, non-linear depending upon the nature of the objective functions and the constraints. This classification is important, because computational methods are usually selected on the basis of such a classification, *i.e.* the nature of the involved functions dictates the type of solution procedure.
- Admissible values of the design variables. Depending upon the values permitted for the design variables, optimization problems can be classified as integer or real valued, and deterministic or stochastic.

1.4 Examples

Example 1 A travelling salesman has to cover n towns. He plans to start from a particular town numbered 1, visit each one of the other $n-1$ towns, and return to the town 1. The distance between town i and j is given by d_{ij}. How should he select the sequence in which the towns are visited to minimize the total distance travelled?

Example 2 The bridge network in Figure 1.3 consists of five resistors R_i, $i = 1, \ldots, 5$. Let I_i be the current through the resistance R_i, find the values of R_i so that the total dissipated power is minimum. The current I_i can vary between the lower limit $\underline{I}_i$ and the upper limit $\bar{I}_i$ and the voltage drop $V_i = R_i I_i$ must be equal to a constant c_i.

Example 3 A manufacturing firm produces two products, A and B, using two limited resources, 1 and 2. The maximum amount of resource 1 available per week is 1000 and the

Article type	w_i	v_i	c_i
1	4	9	5
2	8	7	6
3	2	4	3

Table 1.1: Properties of the articles to load.

maximum amount of resource 2 is 250. The production of one unit of A requires 1 unit of resource 1 and 1/5 unit of resource 2. The production of one unit of B requires 1/2 unit of resource 1 and 1/2 unit of resource 2. The unit cost of resource 1 is $1-0.0005u_1$, where u_1 is the number of units of resource 1 used. The unit cost of resource 2 is $3/4-0.0001u_2$, where u_2 is the number of units of resource 2 used. The selling price of one unit of A is

$$2 - 0.005x_A - 0.0001x_B$$

and the selling price of one unit of B is

$$4 - 0.002x_A - 0.01x_B,$$

where x_A and x_B are the number of units of A and B sold. Assuming that the firm is able to sell all manufactured units, maximize the weekly profit.

Example 4 A cargo load is to be prepared for three types of articles. The weight, w_i, volume, v_i, and value, c_i, of each article is given in Table 1.1.
Find the number of articles x_i selected from type i so that the total value of the cargo is maximized. The total weight and volume of the cargo cannot exceed 2000 and 2500 units respectively.

Example 5 There are two types of gas molecules in a gaseous mixture at equilibrium. It is known that the Gibbs free energy

$$G(x) = c_1x^1 + c_2x^2 + x^1log(x^1/x_T) + x^2log(x^2/x_T),$$

with $x_T = x^1 + x^2$ and c_1, c_2 known parameters depending upon the temperature and pressure of the mixture, has to be minimum in these conditions. The minimization of $G(x)$ is also subject to the mass balance equations:

$$x^1a_{i1} + x^2a_{i2} = b_i,$$

for $i = 1, \ldots, m$, where m is the number of atomic species in the mixture, b_i is the total weight of atoms of type i, and a_{ij} is the number of atoms of type i in the molecule of type j. Show that the problem of determining the equilibrium of the mixture can be posed as an optimization problem.

Chapter 2

Unconstrained optimization

2.1 Introduction

Several engineering, economic and planning problems can be posed as optimization problems, *i.e.* as the problem of determining the points of minimum of a function (possibly in the presence of conditions on the decision variables). Moreover, also numerical problems, such as the problem of solving systems of equations or inequalities, can be posed as an optimization problem.
We start with the study of optimization problems in which the decision variables are defined in $\mathbb{R}^n$: unconstrained optimization problems. More precisely we study the problem of determining local minima for differentiable functions. Although these methods are seldom used in applications, as in real problems the decision variables are subject to constraints, the techniques of unconstrained optimization are instrumental to solve more general problems: the knowledge of good methods for local unconstrained minimization is a necessary pre-requisite for the solution of constrained and global minimization problems.
The methods that will be studied can be classified from various points of view. The most interesting classification is based on the information available on the function to be optimized, namely

- methods without derivatives (direct search, finite differences);
- methods based on the knowledge of the first derivatives (gradient, conjugate directions, quasi-Newton);
- methods based on the knowledge of the first and second derivatives (Newton).

2.2 Definitions and existence conditions

Consider the optimization problem:

Problem 1 *Minimize*

$$f(x) \qquad \text{subject to } x \in \mathcal{F}$$

in which $f : \mathbb{R}^n \to \mathbb{R}$ *and*[1] $\mathcal{F} \subset \mathbb{R}^n$.

With respect to this problem we introduce the following definitions.

Definition 1 *A point* $x \in \mathcal{F}$ *is a global minimum*[2] *for the Problem 1 if*

$$f(x) \leq f(y)$$

for all $y \in \mathcal{F}$.
A point $x \in \mathcal{F}$ *is a strict (or isolated) global minimum (or minimiser) for the Problem 1 if*

$$f(x) < f(y)$$

[1] The set $\mathcal{F}$ may be specified by equations of the form (1.1) and/or (1.2).

[2] Alternatively, the term *global minimiser* can be used to denote a point at which the function f attains its global minimum.

for all $y \in \mathcal{F}$ *and* $y \neq x$.
A point $x \in \mathcal{F}$ *is a local minimum (or minimiser) for the Problem 1 if there exists* $\rho > 0$ *such that*

$$f(x) \leq f(y)$$

for all $y \in \mathcal{F}$ *such that* $\|y - x\| < \rho$.
A point $x \in \mathcal{F}$ *is a strict (or isolated) local minimum (or minimiser) for the Problem 1 if there exists* $\rho > 0$ *such that*

$$f(x) < f(y)$$

for all $y \in \mathcal{F}$ *such that* $\|y - x\| < \rho$ *and* $y \neq x$.

Definition 2 *If* $x \in \mathcal{F}$ *is a local minimum for the Problem 1 and if* x *is in the interior of* $\mathcal{F}$ *then* x *is an unconstrained local minimum of* f *in* $\mathcal{F}$.

The following result provides a sufficient, but not necessary, condition for the existence of a global minimum for Problem 1.

Proposition 1 *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a continuous function and let* $\mathcal{F} \subset \mathbb{R}^n$ *be a compact set*[3]. *Then there exists a global minimum of* f *in* $\mathcal{F}$.

In unconstrained optimization problems the set $\mathcal{F}$ coincides with $\mathbb{R}^n$, hence the above statement cannot be used to establish the existence of global minima. To address the existence problem it is necessary to consider the structure of the level sets of the function f. See also Section 1.2.3.

Definition 3 *Let* $f : \mathbb{R}^n \to \mathbb{R}$. *A level set of* f *is any non-empty set described by*

$$\mathcal{L}(\alpha) = \{x \in \mathbb{R}^n \ : \ f(x) \leq \alpha\},$$

with $\alpha \in \mathbb{R}$.

For convenience, if $x_0 \in \mathbb{R}^n$ we denote with $\mathcal{L}_0$ the level set $\mathcal{L}(f(x_0))$. Using the concept of level sets it is possible to establish a simple sufficient condition for the existence of global solutions for an unconstrained optimization problem.

Proposition 2 *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a continuous function. Assume there exists* $x_0 \in \mathbb{R}^n$ *such that the level set* $\mathcal{L}_0$ *is compact. Then there exists a point of global minimum of* f *in* $\mathbb{R}^n$.

Proof. By Proposition 1 there exists a global minimum $x_\star$ of f in $\mathcal{L}_0$, *i.e.* $f(x_\star) \leq f(x)$ for all $x \in \mathcal{L}_0$. However, if $x \notin \mathcal{L}_0$ then $f(x) > f(x_0) \geq f(x_\star)$, hence $x_\star$ is a global minimum of f in $\mathbb{R}^n$. ◁

It is obvious that the structure of the level sets of the function f plays a fundamental role in the solution of Problem 1. The following result provides a necessary and sufficient condition for the compactness of all level sets of f.

[3] A compact set is a bounded and closed set.

Proposition 3 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function. All level sets of f are compact if and only if for any sequence $\{x_k\}$ one has*

$$\lim_{k\to\infty} \|x_k\| = \infty \quad \Rightarrow \quad \lim_{k\to\infty} f(x_k) = \infty.$$

Remark. In general $x_k \in \mathbb{R}^n$, namely

$$x_k = \begin{bmatrix} x_k^1 \\ x_k^2 \\ \vdots \\ x_k^n \end{bmatrix},$$

i.e. we use superscripts to denote components of a vector. ◇

A function that satisfies the condition of the above proposition is said to be radially unbounded.

Proof. We only prove the necessity. Suppose all level sets of f are compact. Then, proceeding by contradiction, suppose there exist a sequence $\{x_k\}$ such that $\lim_{k\to\infty} \|x_k\| = \infty$ and a number $\gamma > 0$ such that $f(x_k) \leq \gamma < \infty$ for all k. As a result

$$\{x_k\} \subset \mathcal{L}(\gamma).$$

However, by compactness of $\mathcal{L}(\gamma)$ it is not possible that $\lim_{k\to\infty} \|x_k\| = \infty$. ◁

Definition 4 *Let $f : \mathbb{R}^n \to \mathbb{R}$. A vector $d \in \mathbb{R}^n$ is said to be a descent direction for f in $x_\star$ if there exists $\delta > 0$ such that*

$$f(x_\star + \lambda d) < f(x_\star),$$

for all $\lambda \in (0, \delta)$.

If the function f is differentiable it is possible to give a simple condition guaranteeing that a certain direction is a descent direction.

Proposition 4 *Let $f : \mathbb{R}^n \to \mathbb{R}$ and assume[4] ∇f exists and is continuous. Let $x_\star$ and d be given. Then, if $\nabla f(x_\star)' d < 0$ the direction d is a descent direction for f at $x_\star$.*

Proof. Note that $\nabla f(x_\star)' d$ is the directional derivative of f (which is differentiable by hypothesis) at $x_\star$ along d, *i.e.*

$$\nabla f(x_\star)' d = \lim_{\lambda\to 0^+} \frac{f(x_\star + \lambda d) - f(x_\star)}{\lambda},$$

[4]We denote with ∇f the gradient of the function f, *i.e.* $\nabla f = [\frac{\partial f}{\partial x^1}, \cdots, \frac{\partial f}{\partial x^n}]'$. Note that ∇f is a column vector.

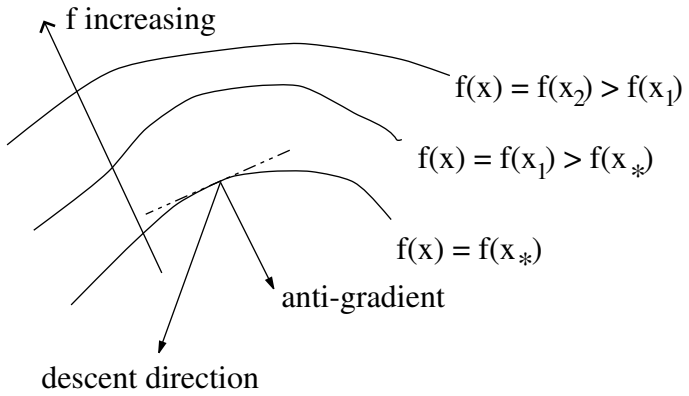

Figure 2.1: Geometrical interpretation of the anti-gradient.

and this is negative by hypothesis. As a result, for $\lambda > 0$ and sufficiently small

$$f(x_\star + \lambda d) - f(x_\star) < 0,$$

hence the claim. ◁

The proposition establishes that if $\nabla f(x_\star)'d < 0$ then for sufficiently small positive displacements along d and starting at $x_\star$ the function f is decreasing. It is also obvious that if $\nabla f(x_\star)'d > 0$, d is a direction of *ascent*, *i.e.* the function f is increasing for sufficiently small positive displacements from $x_\star$ along d. If $\nabla f(x_\star)'d = 0$, d is orthogonal to $\nabla f(x_\star)$ and it is not possible to establish, without further knowledge on the function f, what is the nature of the direction d.
From a geometrical point of view (see also Figure 2.1), the sign of the *directional derivative* $\nabla f(x_\star)'d$ gives information on the angle between d and the direction of the gradient at $x_\star$. If $\nabla f(x_\star)'d > 0$ the angle between $\nabla f(x_\star)$ and d is acute. If $\nabla f(x_\star)'d < 0$ the angle between $\nabla f(x_\star)$ and d is obtuse. Finally, if $\nabla f(x_\star)'d = 0$, $\nabla f(x_\star)$ and d are orthogonal. Note that the gradient $\nabla f(x_\star)$ is a direction orthogonal to the level surface $\{x : f(x) = f(x_\star)\}$ and it is a direction of ascent, hence the anti-gradient $-\nabla f(x_\star)$ is a descent direction.

Remark. The scalar product $x'y$ between the two vectors x and y can be used to define the angle between x and y. For, define the angle between x and y as the number $\theta \in [0, \pi]$ such that[5]

$$\cos\theta = \frac{x'y}{\|x\|_E \|y\|_E}.$$

If $x'y = 0$ one has $\cos\theta = 0$ and the vectors are orthogonal, whereas if x and y has the same direction, *i.e.* $x = \lambda y$ with $\lambda > 0$, $\cos\theta = 1$. ◇

[5] $\|x\|_E$ denotes the Euclidean norm of the the vector x, *i.e.* $\|x\|_E = \sqrt{x'x}$.

We are now ready to state and prove some necessary conditions and some sufficient conditions for a local minimum.

Theorem 1 *[First order necessary condition] Let $f : \mathbb{R}^n \to \mathbb{R}$ and assume ∇f exists and is continuous. The point $x_\star$ is a local minimum of f only if*

$$\nabla f(x_\star) = 0.$$

Remark. A point $x_\star$ such that $\nabla f(x_\star) = 0$ is called a stationary point of f. ◇

Proof. If $\nabla f(x_\star) \neq 0$ the direction $d = -\nabla f(x_\star)$ is a descent direction. Therefore, in a neighborhood of $x_\star$ there is a point $x_\star + \lambda d = x_\star - \lambda \nabla f(x_\star)$ such that

$$f(x_\star - \lambda \nabla f(x_\star)) < f(x_\star),$$

and this contradicts the hypothesis that $x_\star$ is a local minimum. ◁

Theorem 2 *[Second order necessary condition] Let $f : \mathbb{R}^n \to \mathbb{R}$ and assume[6] $\nabla^2 f$ exists and is continuous. The point $x_\star$ is a local minimum of f only if*

$$\nabla f(x_\star) = 0$$

and

$$x' \nabla^2 f(x_\star) x \geq 0$$

for all $x \in \mathbb{R}^n$.

Proof. The first condition is a consequence of Theorem 1. Note now that, as f is two times differentiable, for any $x \neq x_\star$ one has

$$f(x_\star + \lambda x) = f(x_\star) + \lambda \nabla f(x_\star)' x + \frac{1}{2} \lambda^2 x' \nabla^2 f(x_\star) x + \beta(x_\star, \lambda x),$$

where

$$\lim_{\lambda \to 0} \frac{\beta(x_\star, \lambda x)}{\lambda^2 \|x\|^2} = 0,$$

or what is the same (note that x is fixed)

$$\lim_{\lambda \to 0} \frac{\beta(x_\star, \lambda x)}{\lambda^2} = 0.$$

[6] We denote with $\nabla^2 f$ the Hessian matrix of the function f, *i.e.*

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x^1 \partial x^1} & \cdots & \frac{\partial^2 f}{\partial x^1 \partial x^n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x^n \partial x^1} & \cdots & \frac{\partial^2 f}{\partial x^n \partial x^n} \end{bmatrix}.$$

Note that $\nabla^2 f$ is a square matrix and that, under suitable regularity conditions, the Hessian matrix is symmetric.

Moreover, the condition $\nabla f(x_\star) = 0$ yields

$$\frac{f(x_\star + \lambda x) - f(x_\star)}{\lambda^2} = \frac{1}{2} x' \nabla^2 f(x_\star) x + \frac{\beta(x_\star, \lambda x)}{\lambda^2}. \tag{2.1}$$

However, as $x_\star$ is a local minimum, the left hand side of equation (2.1) must be non-negative for all λ sufficiently small, hence

$$\frac{1}{2} x' \nabla^2 f(x_\star) x + \frac{\beta(x_\star, \lambda x)}{\lambda^2} \geq 0,$$

and

$$\lim_{\lambda \to 0} \left(\frac{1}{2} x' \nabla^2 f(x_\star) x + \frac{\beta(x_\star, \lambda x)}{\lambda^2} \right) = \frac{1}{2} x' \nabla^2 f(x_\star) x \geq 0,$$

which proves the second condition. ◁

Theorem 3 (Second order sufficient condition) *Let $f : \mathbb{R}^n \to \mathbb{R}$ and assume $\nabla^2 f$ exists and is continuous. The point $x_\star$ is a strict local minimum of f if*

$$\nabla f(x_\star) = 0$$

and

$$x' \nabla^2 f(x_\star) x > 0$$

for all non-zero $x \in \mathbb{R}^n$.

Proof. To begin with, note that as $\nabla^2 f(x_\star) > 0$ and $\nabla^2 f$ is continuous, then there is a neighborhood Ω of $x_\star$ such that for all $y \in \Omega$

$$\nabla^2 f(y) > 0.$$

Consider now the Taylor series expansion of f around the point $x_\star$, *i.e.*

$$f(y) = f(x_\star) + \nabla f(x_\star)'(y - x_\star) + \frac{1}{2}(y - x_\star)' \nabla^2 f(\xi)(y - x_\star),$$

where $\xi = x_\star + \theta(y - x_\star)$, for some $\theta \in [0, 1]$. By the first condition one has

$$f(y) = f(x_\star) + \frac{1}{2}(y - x_\star)' \nabla^2 f(\xi)(y - x_\star),$$

and, for any $y \in \Omega$ such that $y \neq x_\star$,

$$f(y) > f(x_\star),$$

which proves the claim. ◁

The above results can be easily modified to derive necessary conditions and sufficient conditions for a local maximum. Moreover, if $x_\star$ is a stationary point and the Hessian matrix

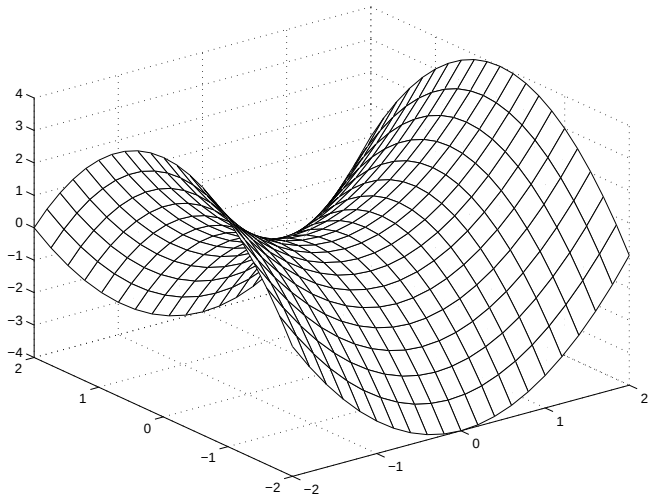

Figure 2.2: A saddle point in $\mathbb{R}^2$.

$\nabla^2 f(x_\star)$ is indefinite, the point $x_\star$ is neither a local minimum neither a local maximum. Such a point is called a saddle point (see Figure 2.2 for a geometrical illustration).
If $x_\star$ is a stationary point and $\nabla^2 f(x_\star)$ is semi-definite it is not possible to draw any conclusion on the point $x_\star$ without further knowledge on the function f. Nevertheless, if $n = 1$ and the function f is infinitely times differentiable it is possible to establish the following necessary and sufficient condition.

Proposition 5 *Let $f : \mathbb{R} \to \mathbb{R}$ and assume f is infinitely times differentiable. The point $x_\star$ is a local minimum if and only if there exists an even integer $r > 1$ such that*

$$\frac{d^k f(x_\star)}{dx^k} = 0$$

for $k = 1, 2, \ldots, r-1$ and

$$\frac{d^r f(x_\star)}{dx^r} > 0.$$

Necessary and sufficient conditions for $n > 1$ can be only derived if further hypotheses on the function f are added, as shown for example in the following fact.

Proposition 6 (Necessary and sufficient condition for convex functions) *Let $f : \mathbb{R}^n \to \mathbb{R}$ and assume ∇f exists and it is continuous. Suppose f is convex,* i.e.

$$f(y) - f(x) \geq \nabla f(x)'(y - x) \tag{2.2}$$

for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$. The point $x_\star$ is a global minimum if and only if $\nabla f(x_\star) = 0$.

Proof. The necessity is a consequence of Theorem 1. For the sufficiency note that, by equation (2.2), if $\nabla f(x_\star) = 0$ then

$$f(y) \geq f(x_\star),$$

for all $y \in I\!R^n$. ◁

From the above discussion it is clear that to establish the property that $x_\star$, satisfying $\nabla f(x_\star) = 0$, is a global minimum it is enough to assume that the function f has the following property: for all x and y such that

$$\nabla f(x)'(x - y) \geq 0$$

one has

$$f(y) \geq f(x).$$

A function f satisfying the above property is said pseudo-convex. Note that a differentiable convex function is also pseudo-convex, but the opposite is not true. For example, the function $x + x^3$ is pseudo-convex but it is not convex. Finally, if f is strictly convex or strictly pseudo-convex the global minimum (if it exists) is also unique.

2.3 General properties of minimization algorithms

Consider the problem of minimizing the function $f : I\!R^n \to I\!R$ and suppose that ∇f and $\nabla^2 f$ exist and are continuous. Suppose that such a problem has a solution, and moreover that there exists x_0 such that the level set

$$\mathcal{L}(f(x_0)) = \{x \in I\!R^n \ : \ f(x) \leq f(x_0)\}$$

is compact.
General unconstrained minimization algorithms allow only to determine stationary points of f, *i.e.* to determine points in the set

$$\Omega = \{x \in I\!R^n \ : \ \nabla f(x) = 0\}.$$

Moreover, for almost all algorithms, it is possible to exclude that the points of Ω yielded by the algorithm are local maxima. Finally, some algorithms yield points of Ω that satisfy also the second order necessary conditions.

2.3.1 General unconstrained minimization algorithm

An *algorithm* for the solution of the considered minimization problem is a sequence $\{x_k\}$, obtained starting from an initial point x_0, having some convergence properties in relation with the set Ω. Most of the algorithms that will be studied in this notes can be described in the following general way.

1. Fix a point $x_0 \in I\!R^n$ and set $k = 0$.

2. If $x_k \in \Omega$ STOP.

3. Compute a direction of research $d_k \in I\!R^n$.

4. Compute a step $\alpha_k \in I\!R$ along d_k.

5. Let $x_{k+1} = x_k + \alpha_k d_k$. Set $k = k + 1$ and go back to 2.

The existing algorithms differ in the way the direction of research d_k is computed and on the criteria used to compute the step α_k. However, independently from the particular selection, it is important to study the following issues:

- the existence of accumulation points for the sequence $\{x_k\}$;
- the behavior of such accumulation points in relation with the set Ω;
- the speed of convergence of the sequence $\{x_k\}$ to the points of Ω.

2.3.2 Existence of accumulation points

To make sure that any subsequence of $\{x_k\}$ has an accumulation point it is necessary to assume that the sequence $\{x_k\}$ remains bounded, *i.e.* that there exists $M > 0$ such that $\|x_k\| < M$ for any k. If the level set $\mathcal{L}(f(x_0))$ is compact, the above condition holds if $\{x_k\} \in \mathcal{L}(f(x_0))$. This property, in turn, is guaranteed if

$$f(x_{k+1}) < f(x_k),$$

for any k such that $x_k \notin \Omega$. The algorithms that satisfy this property are denominated descent methods. For such methods , if $\mathcal{L}(f(x_0))$ is compact and if ∇f is continuous one has

- $\{x_k\} \in \mathcal{L}(f(x_0))$ and any subsequence of $\{x_k\}$ admits a subsequence converging to a point of $\mathcal{L}(f(x_0))$;
- the sequence $\{f(x_k)\}$ has a limit, *i.e.* there exists $\bar{f} \in I\!R$ such that

$$\lim_{k\to\infty} f(x_k) = \bar{f};$$

- there always exists an element of Ω in $\mathcal{L}(f(x_0))$. In fact, as f has a minimum in $\mathcal{L}(f(x_0))$, this minimum is also a minimum of f in $I\!R^n$. Hence, by the assumptions of ∇f, such a minimum must be a point of Ω.

Remark. To guarantee the descent property it is necessary that the research directions d_k be directions of descent. This is true if

$$\nabla f(x_k)' d_k < 0,$$

for all k. Under this condition there exists an interval $(0, \alpha_\star]$ such that

$$f(x_k + \alpha d_k) < f(x_k),$$

for any $\alpha \in (0, \alpha_\star]$. ◇

Remark. The existence of accumulation points for the sequence $\{x_k\}$ and the convergence of the sequence $\{f(x_k)\}$ do not guarantee that the accumulation points of $\{x_k\}$ are local minima of f or stationary points. To obtain this property it is necessary to impose further restrictions on the research directions d_k and on the steps α_k. ◇

2.3.3 Condition of angle

The condition which is in general imposed on the research directions d_k is the so-called condition of angle, that can be stated as follows.

Condition 1 *There exists $\epsilon > 0$, independent from k, such that*

$$\nabla f(x_k)' d_k \leq -\epsilon \|\nabla f(x_k)\| \|d_k\|,$$

for any k.

From a geometric point of view the above condition implies that the cosine of the angle between d_k and $-\nabla f(x_k)$ is larger than a certain quantity. This condition is imposed to avoid that, for some k, the research direction is orthogonal to the direction of the gradient. Note moreover that, if the angle condition holds, and if $\nabla f(x_k) \neq 0$ then d_k is a descent direction. Finally, if $\nabla f(x_k) \neq 0$, it is always possible to find a direction d_k such that the angle condition holds. For example, the direction $d_k = -\nabla f(x_k)$ is such that the angle condition is satisfied with $\epsilon = 1$.

Remark. Let $\{B_k\}$ be a sequence of matrices such that

$$mI \leq B_k \leq MI,$$

for some $0 < m < M$, and for any k, and consider the directions

$$d_k = -B_k \nabla f(x_k).$$

Then a simple computation shows that the angle condition holds with $\epsilon = m/M$. ◇

The angle condition imposes a constraint only on the research directions d_k. To make sure that the sequence $\{x_k\}$ converges to a point in Ω it is necessary to impose further conditions on the step α_k, as expressed in the following statements.

Theorem 4 *Let $\{x_k\}$ be the sequence obtained by the algorithm*

$$x_{k+1} = x_k + \alpha_k d_k,$$

for $k \geq 0$. Assume that

(H1) ∇f *is continuous and the level set* $\mathcal{L}(f(x_0))$ *is compact.*

(H2) There exists $\epsilon > 0$ *such that*

$$\nabla f(x_k)' d_k \leq -\epsilon \|\nabla f(x_k)\| \|d_k\|,$$

for any $k \geq 0$.

(H3) $f(x_{k+1}) < f(x_k)$ *for any* $k \geq 0$.

(H4) The property

$$\lim_{k \to \infty} \frac{\nabla f(x_k)' d_k}{\|d_k\|} = 0$$

holds.

Then

(C1) $\{x_k\} \in \mathcal{L}(f(x_0))$ *and any subsequence of* $\{x_k\}$ *has an accumulation point.*

(C2) $\{f(x_k)\}$ *is monotonically decreasing and there exists* $\bar{f}$ *such that*

$$\lim_{k \to \infty} f(x_k) = \bar{f}.$$

(C3) $\{\nabla f(x_k)\}$ *is such that*

$$\lim_{k \to \infty} \|\nabla f(x_k)\| = 0.$$

(C4) Any accumulation point $\bar{x}$ *of* $\{x_k\}$ *is such that* $\nabla f(\bar{x}) = 0$.

Proof. Conditions (C1) and (C2) are a simple consequence of (H1) and (H3). Note now that (H2) implies

$$\epsilon \|\nabla f(x_k)\| \leq \frac{|\nabla f(x_k)' d_k|}{\|d_k\|},$$

for all k. As a result, and by (H4),

$$\lim_{k \to \infty} \epsilon \|\nabla f(x_k)\| \leq \lim_{k \to \infty} \frac{|\nabla f(x_k)' d_k|}{\|d_k\|} = 0$$

hence (C3) holds. Finally, let $\bar{x}$ be an accumulation point of the sequence $\{x_k\}$, *i.e.* there is a subsequence that converges to $\bar{x}$. For such a subsequence, and by continuity of f, one has

$$\lim_{k \to \infty} \nabla f(x_k) = \nabla f(\bar{x}),$$

and, by (C3),

$$\nabla f(\bar{x}) = 0,$$

which proves (C4). ◁

Remark. Theorem 4 does not guarantee the convergence of the sequence $\{x_k\}$ to a unique accumulation point. Obviously $\{x_k\}$ has a unique accumulation point if either $\Omega \cap \mathcal{L}(f(x_0))$ contains only one point or $x, y \in \Omega \cap \mathcal{L}(f(x_0))$, with $x \neq y$ implies $f(x) \neq f(y)$. Finally, if the set $\Omega \cap \mathcal{L}(f(x_0))$ contains a finite number of points, a sufficient condition for the existence of a unique accumulation point is

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0.$$

$\diamond$

Remark. The angle condition can be replaced by the following one. There exists $\eta > 0$ and $q > 0$, both independent from k, such that

$$\nabla f(x_k)' d_k \leq -\eta \|\nabla f(x_k)\|^q \|d_k\|.$$

$\diamond$

The result illustrated in Theorem 4 requires the fulfillment of the angle condition or of a similar one, *i.e.* of a condition involving ∇f. In many algorithms that do not make use of the gradient it may be difficult to check the validity of the angle condition, hence it is necessary to use different conditions on the research directions. For example, it is possible to replace the angle condition with a property of linear independence of the research directions.

Theorem 5 *Let $\{x_k\}$ be the sequence obtained by the algorithm*

$$x_{k+1} = x_k + \alpha_k d_k,$$

for $k \geq 0$. Assume that

- *$\nabla^2 f$ is continuous and the level set $\mathcal{L}(f(x_0))$ is compact.*
- *There exist $\sigma > 0$, independent from k, and $k_0 > 0$ such that, for any $k \geq k_0$ the matrix P_k composed of the columns*

  $$\frac{d_k}{\|d_k\|}, \frac{d_{k+1}}{\|d_{k+1}\|}, \ldots, \frac{d_{k+n-1}}{\|d_{k+n-1}\|},$$

  is such that

  $$|\det P_k| \geq \sigma.$$

- $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$.
- *$f(x_{k+1}) < f(x_k)$ for any $k \geq 0$.*
- *The property*

  $$\lim_{k\to\infty} \frac{\nabla f(x_k)' d_k}{\|d_k\|} = 0$$

  holds.

Then

- $\{x_k\} \in \mathcal{L}(f(x_0))$ *and any subsequence of* $\{x_k\}$ *has an accumulation point.*
- $\{f(x_k)\}$ *is monotonically decreasing and there exists* $\bar{f}$ *such that*

$$\lim_{k\to\infty} f(x_k) = \bar{f}.$$

- *Any accumulation point* $\bar{x}$ *of* $\{x_k\}$ *is such that* $\nabla f(\bar{x}) = 0$.

Moreover, if the set $\Omega \cap \mathcal{L}(f(x_0))$ *is composed of a finite number of points, the sequence* $\{x_k\}$ *has a unique accumulation point.*

2.3.4 Speed of convergence

Together with the property of convergence of the sequence $\{x_k\}$ it is important to study also the speed of convergence. To study such a notion it is convenient to assume that $\{x_k\}$ converges to a point $x_\star$.
If there exists a finite k such that $x_k = x_\star$ then we say that the sequence $\{x_k\}$ has finite convergence. Note that if $\{x_k\}$ is generated by an algorithm, there is a stopping condition that has to be satisfied at step k.
If $x_k \neq x_\star$ for any finite k, it is possible (and convenient) to study the asymptotic properties of $\{x_k\}$. One criterion to estimate the speed of convergence is based on the behavior of the error $\mathcal{E}_k = \|x_k - x_\star\|$, and in particular on the relation between $\mathcal{E}_{k+1}$ and $\mathcal{E}_k$.
We say that $\{x_k\}$ has speed of convergence of order p if

$$\lim_{k\to\infty} \left(\frac{\mathcal{E}_{k+1}}{\mathcal{E}_k^p} \right) = C_p$$

with $p \geq 1$ and $0 < C_p < \infty$. Note that if $\{x_k\}$ has speed of convergence of order p then

$$\lim_{k\to\infty} \left(\frac{\mathcal{E}_{k+1}}{\mathcal{E}_k^q} \right) = 0,$$

if $1 \leq q < p$, and

$$\lim_{k\to\infty} \left(\frac{\mathcal{E}_{k+1}}{\mathcal{E}_k^q} \right) = \infty,$$

if $q > p$. Moreover, from the definition of speed of convergence, it is easy to see that if $\{x_k\}$ has speed of convergence of order p then, for any $\epsilon > 0$ there exists k_0 such that

$$\mathcal{E}_{k+1} \leq (C_p + \epsilon)\mathcal{E}_k^p,$$

for any $k > k_0$.
In the cases $p = 1$ or $p = 2$ the following terminology is often used. If $p = 1$ and $0 < C_1 \leq 1$ the speed of convergence is linear; if $p = 1$ and $C_1 > 1$ the speed of convergence is sublinear; if

$$\lim_{k\to\infty} \left(\frac{\mathcal{E}_{k+1}}{\mathcal{E}_k} \right) = 0$$

the speed of convergence is superlinear, and finally if $p = 2$ the speed of convergence is quadratic.
Of special interest in optimization is the case of superlinear convergence, as this is the kind of convergence that can be established for the *efficient* minimization algorithms. Note that if x_k has superlinear convergence to $x_\star$ then

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x_k\|}{\|x_k - x_\star\|} = 1.$$

Remark. In some cases it is not possible to establish the existence of the limit

$$\lim_{k\to\infty} \left(\frac{\mathcal{E}_{k+1}}{\mathcal{E}_k^q} \right).$$

In these cases an estimate of the speed of convergence is given by

$$Q_p = \limsup_{k\to\infty} \left(\frac{\mathcal{E}_{k+1}}{\mathcal{E}_k^q} \right).$$

$\diamond$

2.4 Line search

A line search is a method to compute the step α_k along a given direction d_k. The choice of α_k affects both the convergence and the speed of convergence of the algorithm. In any line search one considers the function of one variable $\phi : \mathbb{R} \to \mathbb{R}$ defined as

$$\phi(\alpha) = f(x_k + \alpha d_k) - f(x_k).$$

The derivative of $\phi(\alpha)$ with respect to α is given by

$$\dot{\phi}(\alpha) = \nabla f(x_k + \alpha d_k)' d_k$$

provided that ∇f is continuous. Note that $\nabla f(x_k + \alpha d_k)' d_k$ describes the slope of the tangent to the function $\phi(\alpha)$, and in particular

$$\dot{\phi}(0) = \nabla f(x_k)' d_k$$

coincides with the directional derivative of f at x_k along d_k.
From the general convergence results described, we conclude that the line search has to enforce the following conditions

$$f(x_{k+1}) < f(x_k)$$

$$\lim_{k\to\infty} \frac{\nabla f(x_k)' d_k}{\|d_k\|} = 0$$

and, whenever possible, also the condition

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0.$$

To begin with, we assume that the directions d_k are such that

$$\nabla f(x_k)' d_k < 0$$

for all k, *i.e.* d_k is a descent direction, and that it is possible to compute, for any fixed x, both f and ∇f. Finally, we assume that the level set $\mathcal{L}(f(x_0))$ is compact.

2.4.1 Exact line search

The exact line search consists in finding α_k such that

$$\phi(\alpha_k) = f(x_k + \alpha_k d_k) - f(x_k) \leq f(x_k + \alpha d_k) - f(x_k) = \phi(\alpha)$$

for any $\alpha \geq 0$. Note that, as d_k is a descent direction and the set

$$\{\alpha \in I\!R^+ \ : \ \phi(\alpha) \leq \phi(0)\}$$

is compact, because of compactness of $\mathcal{L}(f(x_0))$, there exists an α_k that minimizes $\phi(\alpha)$. Moreover, for such α_k one has

$$\dot{\phi}(\alpha_k) = \nabla f(x_k + \alpha_k d_k)' d_k = 0,$$

i.e. if α_k minimizes $\phi(\alpha)$ the gradient of f at $x_k + \alpha_k d_k$ is orthogonal to the direction d_k. From a geometrical point of view, if α_k minimizes $\phi(\alpha)$ then the level surface of f through the point $x_k + \alpha_k d_k$ is tangent to the direction d_k at such a point. (If there are several points of tangency, α_k is the one for which f has the smallest value).
The search of α_k that minimizes $\phi(\alpha)$ is very *expensive*, especially if f is not convex. Moreover, in general, the whole minimization algorithm does not gain any special advantage from the knowledge of such *optimal* α_k. It is therefore more convenient to use approximate methods, *i.e.* methods which are computationally simple and which guarantee particular convergence properties. Such methods are aimed at finding an interval of acceptable values for α_k subject to the following two conditions

- α_k has to guarantee a sufficient reduction of f;
- α_k has to be sufficiently distant from 0, *i.e.* $x_k + \alpha_k d_k$ has to be sufficiently away from x_k.

2.4.2 Armijo method

Armijo method was the first non-exact linear search method.
Let $a > 0$, $\sigma \in (0, 1)$ and $\gamma \in (0, 1/2)$ be given and define the set of points

$$A = \{\alpha \in R \ : \ \alpha = a\sigma^j, \ j = 0, 1, \ldots\}.$$

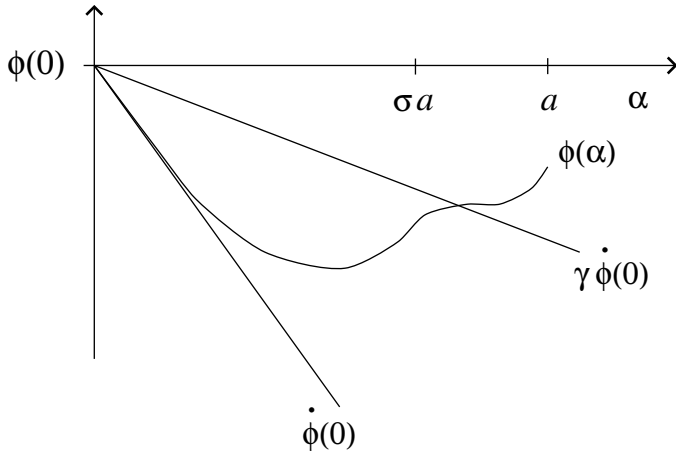

Figure 2.3: Geometrical interpretation of Armijo method.

Armijo method consists in finding the largest $\alpha \in A$ such that

$$\phi(\alpha) = f(x_k + \alpha d_k) - f(x_k) \leq \gamma\alpha\nabla f(x_k)'d_k = \gamma\alpha\dot{\phi}(0).$$

Armijo method can be implemented using the following (conceptual) algorithm.

Step 1. Set $\alpha = a$.

Step 2. If

$$f(x_k + \alpha d_k) - f(x_k) \leq \gamma\alpha\nabla f(x_k)'d_k$$

set $\alpha_k = \alpha$ and STOP. Else go to **Step 3.**

Step 3. Set $\alpha = \sigma\alpha$, and go to **Step 2.**

From a geometric point of view (see Figure 2.3) the condition in **Step 2** requires that α_k is such that $\phi(\alpha_k)$ is below the straight line passing through the point $(0, \phi(0))$ and with slope $\gamma\dot{\phi}(0)$. Note that, as $\gamma \in (0, 1/2)$ and $\dot{\phi}(0) < 0$, such a straight line has a slope smaller than the slope of the tangent at the curve $\phi(\alpha)$ at the point $(0, \phi(0))$.
For Armijo method it is possible to prove the following convergence result.

Theorem 6 *Let $f : \mathbb{R}^n \to \mathbb{R}$ and assume ∇f is continuous and $\mathcal{L}(f(x_0))$ is compact. Assume $\nabla f(x_k)'d_k < 0$ for all k and there exist $C_1 > 0$ and $C_2 > 0$ such that*

$$C_1 \geq \|d_k\| \geq C_2\|\nabla f(x_k)\|^q,$$

for some $q > 0$ and for all k.
Then Armijo method yields in a finite number of iterations a value of $\alpha_k > 0$ satisfying the condition in **Step 2**. *Moreover, the sequence obtained setting $x_{k+1} = x_k + \alpha_k d_k$ is such that*

$$f(x_{k+1}) < f(x_k),$$

for all k, and

$$\lim_{k\to\infty} \frac{\nabla f(x_k)' d_k}{\|d_k\|} = 0.$$

Proof. We only prove that the method cannot loop indefinitely between **Step 2** and **Step 3**. In fact, if this is the case, then the condition in **Step 2** will never be satisfied, hence

$$\frac{f(x_k + a\sigma^j d_k) - f(x_k)}{a\sigma^j} > \gamma \nabla f(x_k)' d_k.$$

Note now that $\sigma^j \to 0$ as $j \to \infty$, and the above inequality for $j \to \infty$ is

$$\nabla f(x_k)' d_k > \gamma \nabla f(x_k)' d_k,$$

which is not possible since $\gamma \in (0, 1/2)$ and $\nabla f(x_k)' d_k \neq 0$. ◁

Remark. It is interesting to observe that in Theorem 6 it is not necessary to assume that $x_{k+1} = x_k + \alpha_k d_k$. It is enough that x_{k+1} is such that

$$f(x_{k+1}) \leq f(x_k + \alpha_k d_k),$$

where α_k is generated using Armijo method. This implies that all acceptable values of α are those such that

$$f(x_k + \alpha d_k) \leq f(x_k + \alpha_k d_k).$$

As a result, Theorem 6 can be used to prove also the convergence of an algorithm based on the exact line search. ◇

2.4.3 Goldstein conditions

The main disadvantage of Armijo method is in the fact that, to find α_k, all points in the set A, starting from the point $\alpha = a$, have to be tested till the condition in **Step 2** is fulfilled. There are variations of the method that do not suffer from this disadvantage. A criterion similar to Armijo's, but that allows to find an acceptable α_k in one step, is based on the so-called Goldstein conditions.
Goldstein conditions state that given $\gamma_1 \in (0, 1)$ and $\gamma_2 \in (0, 1)$ such that $\gamma_1 < \gamma_2$, α_k is any positive number such that

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \alpha_k \gamma_1 \nabla f(x_k)' d_k$$

i.e. there is a sufficient reduction in f, and

$$f(x_k + \alpha_k d_k) - f(x_k) \geq \alpha_k \gamma_2 \nabla f(x_k)' d_k$$

i.e. there is a sufficient distance between x_k and x_{k+1}.
From a geometric point of view (see Figure 2.4) this is equivalent to select α_k as any point such that the corresponding value of f is included between two straight lines, of slope

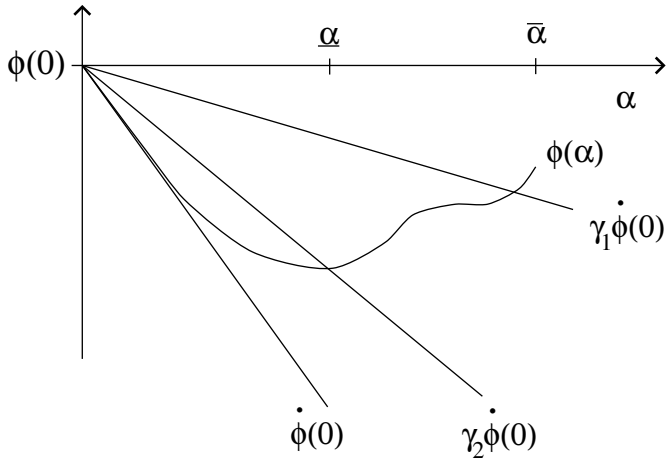

Figure 2.4: Geometrical interpretation of Goldstein method.

$\gamma_1 \nabla f(x_k)' d_k$ and $\gamma_2 \nabla f(x_k)' d_k$, respectively, and passing through the point $(0, \phi(0))$. As $0 < \gamma_1 < \gamma_2 < 1$ it is obvious that there exists always an interval $I = [\underline{\alpha}, \overline{\alpha}]$ such that Goldstein conditions hold for any $\alpha \in I$.
Note that, a result similar to Theorem 6, can be also established if the sequence $\{x_k\}$ is generated using Goldstein conditions.
The main disadvantage of Armijo and Goldstein methods is in the fact that none of them impose conditions on the derivative of the function $\phi(\alpha)$ in the point α_k, or what is the same on the value of $\nabla f(x_{k+1})' d_k$. Such extra conditions are sometimes useful in establishing convergence results for particular algorithms. However, for simplicity, we omit the discussion of these more general conditions.

2.4.4 Line search without derivatives

It is possible to construct methods similar to Armijo's or Goldstein's also in the case that no information on the derivatives of the function f is available.
Suppose, for simplicity, that $\|d_k\| = 1$, for all k, and that the sequence $\{x_k\}$ is generated by

$$x_{k+1} = x_k + \alpha_k d_k.$$

If ∇f is not available it is not possible to decide *a priori* if the direction d_k is a descent direction, hence it is necessary to consider also negative values of α.
We now describe the simplest line search method that can be constructed with the considered hypothesis. This method is a modification of Armijo method and it is known as parabolic search.
Given $\lambda_0 > 0$, $\sigma \in (0, 1/2)$, $\gamma > 0$ and $\rho \in (0, 1)$. Compute α_k and λ_k such that one of the following conditions hold.

Condition (i)

- $\lambda_k = \lambda_{k-1}$;
- α_k is the largest value in the set

$$A = \{\alpha \in I\!R \ : \ \alpha = \pm\sigma^j, \ j = 0, 1, \ldots\}$$

such that

$$f(x_k + \alpha_k d_k) \leq f(x_k) - \gamma\alpha_k^2,$$

or, equivalently, $\phi(\alpha_k) \leq -\gamma\alpha_k^2$.

Condition (ii)

- $\alpha_k = 0$, $\lambda_k \leq \rho\lambda_{k-1}$;
- $\min\left(f(x_k + \lambda_k d_k), f(x_k - \lambda_k d_k)\right) \geq f(x_k) - \gamma\lambda_k^2$.

At each step it is necessary to satisfy either Condition (i) or Condition (ii). Note that this is always possible for any $d_k \neq 0$. Condition (i) requires that α_k is the largest number in the set A such that $f(x_k + \alpha_k d_k)$ is below the parabola $f(x_k) - \gamma\alpha^2$. If the function $\phi(\alpha)$ has a stationary point for $\alpha = 0$ then there is no $\alpha \in A$ such that Condition (i) holds. However, in this case it is possible to find λ_k such that Condition (ii) holds. If Condition (ii) holds then $\alpha_k = 0$, *i.e.* the point x_k remains unchanged and the algorithms continues with a new direction $d_{k+1} \neq d_k$.
For the parabolic search algorithm it is possible to prove the following convergence result.

Theorem 7 *Let $f : I\!R^n \to I\!R$ and assume ∇f is continuous and $\mathcal{L}(f(x_0))$ is compact. If α_k is selected following the conditions of the parabolic search and if $x_{k+1} = x_k + \alpha_k d_k$, with $\|d_k\| = 1$ then the sequence $\{x_k\}$ is such that*

$$f(x_{k+1}) \leq f(x_k)$$

for all k,

$$\lim_{k\to\infty} \nabla f(x_k)' d_k = 0$$

and

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0.$$

Proof. (Sketch) Note that Condition (i) implies $f(x_{k+1}) < f(x_k)$, whereas Condition (ii) implies $f(x_{k+1}) = f(x_k)$. Note now that if Condition (ii) holds for all $k \geq \bar{k}$, then $\alpha_k = 0$ for all $k \geq \bar{k}$, *i.e.* $\|x_{k+1} - x_k\| = 0$. Moreover, as λ_k is reduced at each step, necessarily $\nabla f(x_{\bar{k}})' \bar{d} = 0$, where $\bar{d}$ is a limit of the sequence $\{d_k\}$. ◁

2.4.5 Implementation of a line search algorithm

On the basis of the conditions described so far it is possible to construct algorithms that yield α_k in a finite number of steps. One such an algorithm can be described as follows. (For simplicity we assume that ∇f is known.)

- Initial data. x_k, $f(x_k)$, $\nabla f(x_k)$, $\underline{\alpha}$ and $\overline{\alpha}$.
- Initial guess for α. A possibility is to select α as the point in which a parabola through $(0, \phi(0))$ with derivative $\dot{\phi}(0)$ for $\alpha = 0$ takes a pre-specified minimum value $f_\star$. Initially, *i.e.* for $k = 0$, $f_\star$ has to be selected by the designer. For $k > 0$ it is possible to select $f_\star$ such that

  $$f(x_k) - f_\star = f(x_{k-1}) - f(x_k).$$

  The resulting α is

  $$\alpha_\star = -2\,\frac{f(x_k) - f_\star}{\nabla f(x_k)' d_k}.$$

  In some algorithms it is convenient to select $\alpha \leq 1$, hence the initial guess for α will be $\min(1, \alpha_\star)$.
- Computation of α_k. A value for α_k is computed using a line search method. If $\alpha_k \leq \underline{\alpha}$ the direction d_k may not be a descent direction. If $\alpha_k \geq \overline{\alpha}$ the level set $\mathcal{L}(f(x_k))$ may not be compact. If $\alpha_k \notin [\underline{\alpha}, \overline{\alpha}]$ the line search fails, and it is necessary to select a new research direction d_k. Otherwise the line search terminates and $x_{k+1} = x_k + \alpha_k d_k$.

2.5 The gradient method

The gradient method consists in selecting, as research direction, the direction of the anti-gradient at x_k, *i.e.*

$$d_k = -\nabla f(x_k),$$

for all k. This selection is justified noting that the direction[7]

$$-\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|_E}$$

is the direction that minimizes the directional derivative, among all direction with unitary Euclidean norm. In fact, by Schwartz inequality, one has

$$|\nabla f(x_k)' d| \leq \|d\|_E \|\nabla f(x_k)\|_E,$$

and the equality sign holds if and only if $d = \lambda \nabla f(x_k)$, with $\lambda \in \mathbb{R}$. As a consequence, the problem

$$\min_{\|d\|_E = 1} \nabla f(x_k)' d$$

[7] We denote with $\|v\|_E$ the Euclidean norm of the vector v, *i.e.* $\|v\|_E = \sqrt{v'v}$.

has the solution $d_\star = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|_E}$. For this reason, the gradient method is sometimes called the method of the steepest descent. Note however that the (local) optimality of the direction $-\nabla f(x_k)$ depends upon the selection of the norm, and that with a proper selection of the norm, any descent direction can be regarded as the steepest descent.
The real interest in the direction $-\nabla f(x_k)$ rests on the fact that, if ∇f is continuous, then the former is a continuous descent direction, which is zero only if the gradient is zero, *i.e.* at a stationary point.
The gradient algorithm can be schematized has follows.

Step 0. Given $x_0 \in I\!R^n$.

Step 1. Set $k = 0$.

Step 2. Compute $\nabla f(x_k)$. If $\nabla f(x_k) = 0$ STOP. Else set $d_k = -\nabla f(x_k)$.

Step 3. Compute a step α_k along the direction d_k with any line search method such that

$$f(x_k + \alpha_k d_k) \leq f(x_k)$$

and

$$\lim_{k\to\infty} \frac{\nabla f(x_k)' d_k}{\|d_k\|} = 0.$$

Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$. Go to **Step 2**.

By the general results established in Theorem 4, we have the following fact regarding the convergence properties of the gradient method.

Theorem 8 *Consider $f : I\!R^n \to I\!R$. Assume ∇f is continuous and the level set $\mathcal{L}(f(x_0))$ is compact. Then any accumulation point of the sequence $\{x_k\}$ generated by the gradient algorithm is a stationary point of f.*

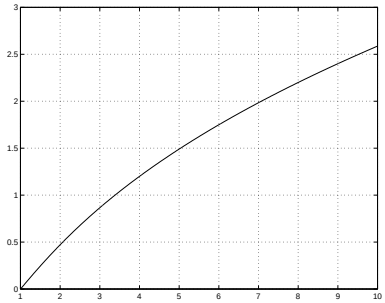

Figure 2.5: The function $\sqrt{\xi}\,\dfrac{\xi - 1}{\xi + 1}$.

To estimate the speed of convergence of the method we can consider the behavior of the method in the minimization of a quadratic function, *i.e.* in the case

$$f(x) = \frac{1}{2}x'Qx + c'x + d,$$

with $Q = Q' > 0$. In such a case it is possible to obtain the following estimate

$$\|x_{k+1} - x_\star\| \leq \sqrt{\frac{\lambda_M}{\lambda_m}}\ \frac{\sqrt{\frac{\lambda_M}{\lambda_m} - 1}}{\sqrt{\frac{\lambda_M}{\lambda_m} + 1}}\|x_k - x_\star\|,$$

where $\lambda_M \geq \lambda_m > 0$ are the maximum and minimum eigenvalue of Q, respectively. Note that the above estimate is exact for some initial points x_0. As a result, if $\lambda_M \neq \lambda_m$ the gradient algorithm has linear convergence, however, if λ_M/λ_m is large the convergence can be very slow (see Figure 2.5).
Finally, if $\lambda_M/\lambda_m = 1$ the gradient algorithm converges in one step. From a geometric point of view the ratio λ_M/λ_m expresses the ratio between the lengths of the maximum and the minimum axes of the ellipsoids, that constitute the level surfaces of f. If this ratio is big there are points from which the gradient algorithm converges very slowly, see *e.g.* Figure 2.6.

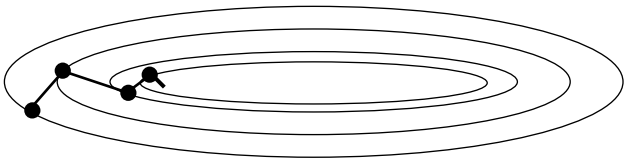

Figure 2.6: Behavior of the gradient algorithm.

In the non-quadratic case, the performance of the gradient method are unacceptable, especially if the level surfaces of f have high curvature.

2.6 Newton's method

Newton's method, with all its variations, is the most important method in unconstrained optimization. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a given function and assume that $\nabla^2 f$ is continuous. Newton's method for the minimization of f can be derived assuming that, given x_k, the point x_{k+1} is obtained minimizing a quadratic approximation of f. As f is two times differentiable, it is possible to write

$$f(x_k + s) = f(x_k) + \nabla f(x_k)'s + \frac{1}{2}s'\nabla^2 f(x_k)s + \beta(x_k, s),$$

in which

$$\lim_{\|s\|\to 0} \frac{\beta(x_k, s)}{\|s\|^2} = 0.$$

For $\|s\|$ sufficiently small, it is possible to approximate $f(x_k + s)$ with its quadratic approximation

$$q(s) = f(x_k) + \nabla f(x_k)'s + \frac{1}{2}s'\nabla^2 f(x_k)s.$$

If $\nabla^2 f(x_k) > 0$, the value of s minimizing $q(s)$ can be obtained setting to zero the gradient of $q(s)$, *i.e.*

$$\nabla q(s) = \nabla f(x_k) + \nabla^2 f(x_k)s = 0,$$

yielding

$$s = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k).$$

The point x_{k+1} is thus given by

$$x_{k+1} = x_k - \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k).$$

Finally, Newton's method can be described by the simple scheme.

Step 0. Given $x_0 \in I\!R^n$.

Step 1. Set $k = 0$.

Step 2. Compute

$$s = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k).$$

Step 3. Set $x_{k+1} = x_k + s$, $k = k + 1$. Go to **Step 2**.

Remark. An equivalent way to introduce Newton's method for unconstrained optimization is to regard the method as an algorithm for the solution of the system of n non-linear equations in n unknowns given by

$$\nabla f(x) = 0.$$

For, consider, in general, a system of n equations in n unknown

$$F(x) = 0,$$

with $x \in I\!R^n$ and $F : I\!R^n \to I\!R^n$. If the Jacobian matrix of F exists and is continuous, then one can write

$$F(x + s) = F(x) + \frac{\partial F}{\partial x}(x_k)s + \gamma(x, s),$$

with

$$\lim_{\|s\| \to 0} \frac{\gamma(x, s)}{\|s\|} = 0.$$

Hence, given a point x_k we can determine $x_{k+1} = x_k + s$ setting s such that

$$F(x_k) + \frac{\partial F}{\partial x}(x_k)s = 0.$$

If $\frac{\partial F}{\partial x}(x_k)$ is invertible we have

$$s = -\left[\frac{\partial F}{\partial x}(x_k)\right]^{-1} F(x_k),$$

hence Newton's method for the solution of the system of equation $F(x) = 0$ is

$$x_{k+1} = x_k - \left[\frac{\partial F}{\partial x}(x_k)\right]^{-1} F(x_k), \tag{2.3}$$

with $k = 0, 1, \ldots$. Note that, if $F(x) = \nabla f$, then the above iteration coincides with Newton's method for the minimization of f. ◇

To study the convergence properties of Newton's method we can consider the algorithm for the solution of a set of non-linear equations, summarized in equation (2.3). The following local convergence result, providing also an estimate of the speed of convergence, can be proved.

Theorem 9 *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ and assume that F is continuously differentiable in an open set $\mathcal{D} \subset \mathbb{R}^n$. Suppose moreover that*

- *there exists $x_\star \in \mathcal{D}$ such that $F(x_\star) = 0$;*
- *the Jacobian matrix $\frac{\partial F}{\partial x}(x_\star)$ is non-singular;*
- *there exists $L > 0$ such that*[8]

$$\left\| \frac{\partial F}{\partial x}(z) - \frac{\partial F}{\partial x}(y) \right\| \leq L\|z - y\|,$$

for all $z \in \mathcal{D}$ and $y \in \mathcal{D}$.

Then there exists and open set $\mathcal{B} \subset \mathcal{D}$ such that for any $x_0 \in \mathcal{B}$ the sequence $\{x_k\}$ generated by equation (2.3) remains in $\mathcal{B}$ and converges to $x_\star$ with quadratic speed of convergence.

The result in Theorem 9 can be easily recast as a result for the convergence of Newton's method for unconstrained optimization. For, it is enough to note that all hypotheses on F and $\frac{\partial F}{\partial x}$ translate into hypotheses on ∇f and $\nabla^2 f$. Note however that the result is only local and does not allow to distinguish between local minima and local maxima. To construct an algorithm for which the sequence $\{x_k\}$ does not converge to maxima, and for which global convergence, *i.e.* convergence from points outside the set $\mathcal{B}$, holds, it is possible to modify Newton's method considering a line search along the direction $d_k = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$. As a result, the modified Newton's algorithm

$$x_{k+1} = x_k - \alpha_k \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k), \tag{2.4}$$

[8] This is equivalent to say that $\frac{\partial F}{\partial x}(x)$ is Lipschitz continuous in $\mathcal{D}$.

in which α_k is computed using any line search algorithm, is obtained. If $\nabla^2 f$ is uniformly positive definite, and this implies that the function f is convex, the direction $d_k = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$ is a descent direction satisfying the condition of angle. Hence, by Theorem 4, we can conclude the (global) convergence of the algorithm (2.4). Moreover, it is possible to prove that, for k sufficiently large, the step $\alpha_k = 1$ satisfies the conditions of Armijo method, hence the sequence $\{x_k\}$ has quadratic speed of convergence.

Remark. If the function to be minimized is quadratic, *i.e.*

$$f(x) = \frac{1}{2} x' Q x + c' x + d,$$

and if $Q > 0$, Newton's method yields the (global) minimum of f in one step. $\diamond$

In general, *i.e.* if $\nabla^2 f(x)$ is not positive definite for all x, Newton's method may be inapplicable because either $\nabla^2 f(x_k)$ is not invertible, or $d_k = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$ is not a descent direction. In these cases it is necessary to further modify Newton's method. Diverse criteria have been proposed, most of which rely on the substitution of the matrix $\nabla^2 f(x_k)$ with a matrix $M_k > 0$ which is *close in some sense* to $\nabla^2 f(x_k)$. A simpler modification can be obtained using the direction $d_k = -\nabla f(x_k)$ whenever the direction $d_k = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$ is not a descent direction. This modification yields the following algorithm.

Step 0. Given $x_0 \in I\!R^n$ and $\epsilon > 0$.

Step 1. Set $k = 0$.

Step 2. Compute $\nabla f(x_k)$. If $\nabla f(x_k) = 0$ STOP. Else compute $\nabla^2 f(x_k)$. If $\nabla^2 f(x_k)$ is singular set $d_k = -\nabla f(x_k)$ and go to **Step 6**.

Step 3. Compute Newton direction s solving the (linear) system

$$\nabla^2 f(x_k) s = -\nabla f(x_k).$$

Step 4. If

$$|\nabla f(x_k)' s| < \epsilon \|\nabla f(x_k)\| \|s\|$$

set $d_k = -\nabla f(x_k)$ and go to **Step 6**.

Step 5. If

$$\nabla f(x_k)' s < 0$$

set $d_k = s$; if

$$\nabla f(x_k)' s > 0$$

set $d_k = -s$.

Step 6. Make a line search along d_k assuming as initial estimate $\alpha = 1$. Compute $x_{k+1} = x_k + \alpha_k d_k$, set $k = k + 1$ and go to **Step 2**.

The above algorithm is such that the direction d_k satisfies the condition of angle, *i.e.*

$$\nabla f(x_k)' d_k \leq \epsilon \|\nabla f(x_k)\| \|d_k\|,$$

for all k. Hence, the convergence is guaranteed by the general result in Theorem 4. Moreover, if ϵ is sufficiently small, if the hypotheses of Theorem 9 hold, and if the line search is performed with Armijo method and with the initial guess $\alpha = 1$, then the above algorithm has quadratic speed of convergence.
Finally, note that it is possible to modify Newton's method, whenever it is not applicable, without making use of the direction of the anti-gradient. We now briefly discuss two such modifications.

2.6.1 Method of the trust region

A possible approach to modify Newton's method to yield global convergence is to set the direction d_k and the step α_k in such a way to minimize the quadratic approximation of f on a sphere centered at x_k and of radius a_k. Such a sphere is called *trust region*. This name refers to the fact that, in a small region around x_k we are confident (we trust) that the quadratic approximation of f is a *good* approximation.
The method of the trust region consists in selecting $x_{k+1} = x_k + s_k$, where s_k is the solution of the problem

$$\min_{\|s\| \leq a_k} q(s), \tag{2.5}$$

with

$$q(s) = f(x_k) + \nabla f(x_k)' s + \frac{1}{2} s' \nabla^2 f(x_k) s,$$

and $a_k > 0$ the estimate at step k of the trust region. As the above (constrained) optimization problem has always a solution, the direction s_k is always defined. The computation of the estimate a_k is done, iteratively, in such a way to enforce the condition $f(x_{k+1}) < f(x_k)$ and to make sure that $f(x_k + s_k) \approx q(s_k)$, *i.e.* that the change of f and the estimated change of f are *close*.
Using these simple ingredients it is possible to construct the following algorithm.

Step 0. Given $x_0 \in \mathbb{R}^n$ and $a_0 > 0$.

Step 1. Set $k = 0$.

Step 2. Compute $\nabla f(x_k)$. If $\nabla f(x_k) = 0$ STOP. Else compute $\nabla^2 f(x_k)$.

Step 3. Compute s_k solving problem (2.5).

Step 4. Compute[9]

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{q(s_k) - f(x_k)}. \tag{2.6}$$

[9] If f is quadratic then $\rho_k = 1$ for all k.

Step 5. If $\rho_k < 1/4$ set $a_{k+1} = \|s_k\|/4$. If $\rho_k > 3/4$ and $\|s_k\| = a_k$ set $a_{k+1} = 2a_k$. Else set $a_{k+1} = a_k$.

Step 6. If $\rho_k \leq 0$ set $x_{k+1} = x_k$. Else set $x_{k+1} = x_k + s_k$.

Step 7. Set $k = k + 1$ and go to **Step 2**.

Remark. Equation (2.6) expresses the ratio between the actual change of f and the estimated change of f. ◇

It is possible to prove that, if $\mathcal{L}(f(x_0))$ is compact and $\nabla^2 f$ is continuous, any accumulation point resulting from the above algorithm is a stationary point of f, in which the second order necessary conditions hold.
The update of a_k is devised to enlarge or shrink the region of confidence on the basis of the number ρ_k. It is possible to show that if $\{x_k\}$ converges to a local minimum in which $\nabla^2 f$ is positive definite, then ρ_k converges to one and the direction s_k coincides, for k sufficiently large, with the Newton direction. As a result, the method has a quadratic speed of convergence.
In practice, the solution of the problem (2.5) cannot be obtained analytically, hence approximate problems have to be solved. For, consider s_k as the solution of the equation

$$\left(\nabla^2 f(x_k) + \nu_k I\right) s = -\nabla f(x_k), \tag{2.7}$$

in which $\nu_k > 0$ has to be determined with proper considerations. Under certain hypotheses, the s_k determined solving equation (2.7) coincides with the s_k computed using the method of the trust region.

Remark. A potential disadvantage of the method of the trust region is to reduce the step along Newton direction even if the selection $\alpha_k = 1$ would be feasible. ◇

2.6.2 Non-monotonic line search

Experimental evidence shows that Newton's method gives the best result if the step $\alpha_k = 1$ is used. Therefore, the use of $\alpha_k < 1$ along Newton direction, resulting *e.g.* from the application of Armijo method, results in a degradation of the performance of the algorithm. To avoid this phenomenon it has been suggested to relax the condition $f(x_{k+1}) < f(x_k)$ imposed on Newton algorithm, thus allowing the function f to increase for a certain number of steps. For example, it is possible to substitute the *reduction* condition of Armijo method with the condition

$$f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq M} [f(x_{k-j})] + \gamma \alpha_k \nabla f(x_k)' d_k$$

for all $k \geq M$, where $M > 0$ is a fixed integer independent from k.

	Gradient method	Newton's method
Information required at each step	f and ∇f	f, ∇f and $\nabla^2 f$
Computation to find the research direction	$\nabla f(x_k)$	$\nabla f(x_k)$, $\nabla^2 f(x_k)$, $-[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$
Convergence	Global if $\mathcal{L}(f(x_0))$ compact and ∇f continuous	Local, but may be rendered global
Behavior for quadratic functions	Asymptotic convergence	Convergence in one step
Speed of convergence	Linear for quadratic functions	Quadratic (under proper hypotheses)

Table 2.1: Comparison between the gradient method and Newton's method.

2.6.3 Comparison between Newton's method and the gradient method

The gradient method and Newton's method can be compared from different point of views, as described in Table 2.1. From the table, it is obvious that Newton's method has better convergence properties but it is computationally more expensive. There exist methods which preserve some of the advantages of Newton's method, namely speed of convergence faster than the speed of the gradient method and finite convergence for quadratic functions, without requiring the knowledge of $\nabla^2 f$. Such methods are

- the conjugate directions methods;
- quasi-Newton methods.

2.7 Conjugate directions methods

Conjugate directions methods have been motivated by the need of improving the convergence speed of the gradient method, without requiring the computation of $\nabla^2 f$, as required in Newton's method.
A basic characteristic of conjugate directions methods is to find the minimum of a quadratic function in a finite number of steps. These methods have been introduced for the solution of systems of linear equations and have later been extended to the solution of unconstrained optimization problems for non-quadratic functions.

Definition 5 *Given a matrix $Q = Q'$, the vectors d_1 and d_2 are said to be Q-conjugate if*

$$d_1' Q d_2 = 0.$$

Remark. If $Q = I$ then two vectors are Q-conjugate if they are orthogonal. ◇

Theorem 10 *Let $Q \in \mathbb{R}^{n \times n}$ and $Q = Q' > 0$. Let $d_i \in \mathbb{R}^n$, for $i = 0, \cdots, k$, be non-zero vectors. If d_i are mutually Q-conjugate,* i.e.

$$d_i' Q d_j = 0,$$

for all $i \neq j$, then the vectors d_i are linearly independent.

Proof. Suppose there exists constants α_i, with $\alpha_i \neq 0$ for some i, such that

$$\alpha_0 d_0 + \cdots \alpha_k d_k = 0.$$

Then, left multiplying with Q and d_j' yields

$$\alpha_j d_j' Q d_j = 0,$$

which implies, as $Q > 0$, $\alpha_j = 0$. Repeating the same considerations for all $j \in [0, k]$ yields the claim. ◁

Consider now a quadratic function

$$f(x) = \frac{1}{2} x' Q x + c' x + d,$$

with $x \in \mathbb{R}^n$ and $Q = Q' > 0$. The (global) minimum of f is given by

$$x_\star = -Q^{-1} c,$$

and this can be computed using the procedure given in the next statement.

Theorem 11 *Let $Q = Q' > 0$ and let d_0, d_1, $\cdots$, d_{n-1} be n non-zero vectors mutually Q-conjugate. Consider the algorithm*

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

with

$$\alpha_k = -\frac{\nabla f(x_k)' d_k}{d_k' Q d_k} = -\frac{(x_k' Q + c') d_k}{d_k' Q d_k}.$$

Then, for any x_0, the sequence $\{x_k\}$ converges, in at most n steps, to $x_\star = -Q^{-1} c$, i.e. *it converges to the minimum of the quadratic function f.*

Remark. Note that α_k is selected at each step to minimize the function $f(x_k + \alpha d_k)$ with respect to α, *i.e.* at each step an exact line search in the direction d_k is performed. $\diamond$

In the above statement we have assumed that the directions d_k have been preliminarily assigned. However, it is possible to construct a procedure in which the directions are computed iteratively. For, consider the quadratic function $f(x) = \frac{1}{2}x'Qx + c'x + d$, with $Q > 0$, and the following algorithm, known as conjugate gradient method.

Step 0. Given $x_0 \in I\!R^n$ and the direction

$$d_0 = -\nabla f(x_0) = -(Qx_0 + c).$$

Step 1. Set $k = 0$.

Step 2. Let

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

with

$$\alpha_k = -\frac{\nabla f(x_k)' d_k}{d_k' Q d_k} - \frac{(x_k' Q + c') d_k}{d_k' Q d_k}.$$

Step 3. Compute d_{k+1} as follows

$$d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k,$$

with

$$\beta_k = \frac{\nabla f(x_{k+1})' Q d_k}{d_k' Q d_k}.$$

Step 4. Set $k = k + 1$ and go to **Step 2**.

Remark. As already observed, α_k is selected to minimize the function $f(x_k + \alpha d_k)$. Moreover, this selection of α_k is also such that

$$\nabla f(x_{k+1})' d_k = 0. \tag{2.8}$$

In fact,

$$Qx_{k+1} = Qx_k + \alpha_k Q d_k$$

hence

$$\nabla f(x_{k+1}) = \nabla f(x_k) + \alpha_k Q d_k. \tag{2.9}$$

Left multiplying with d_k' yields

$$d_k' \nabla f(x_{k+1}) = d_k' \nabla f(x_k) + d_k' Q d_k \alpha_k = d_k' \nabla f(x_k) - d_k' Q d_k \frac{\nabla f(x_k)' d_k}{d_k' Q d_k} = 0.$$

$\diamond$

Remark. β_k is such that d_{k+1} is Q-conjugate with respect to d_k. In fact,

$$d_k' Q d_{k+1} = d_k' Q \left(-\nabla f(x_{k+1}) + \frac{\nabla f(x_{k+1})' Q d_k}{d_k' Q d_k} d_k \right) = d_k' Q \left(-\nabla f(x_{k+1}) + \nabla f(x_{k+1}) \right) = 0.$$

Moreover, this selection of β_k yields also

$$\nabla f(x_k)' d_k = -\nabla f(x_k)' \nabla f(x_k). \tag{2.10}$$

$\diamond$

For the conjugate gradient method it is possible to prove the following fact.

Theorem 12 *The conjugate gradient method yields the minimum of the quadratic function*

$$f(x) = \frac{1}{2} x' Q x + c' x + d,$$

with $Q = Q' > 0$, in at most n iterations, i.e. *there exists $m \leq n-1$ such that*

$$\nabla f(x_{m+1}) = 0.$$

Moreover

$$\nabla f(x_j)' \nabla f(x_i) = 0 \tag{2.11}$$

and

$$d_j' Q d_i = 0, \tag{2.12}$$

for all $[0, m+1] \ni i \neq j \in [0, m+1]$.

Proof. To prove the (finite) convergence of the sequence $\{x_k\}$ it is enough to show that the directions d_k are Q-conjugate, *i.e.* that equation (2.12) holds. In fact, if equation (2.12) holds the claim is a consequence of Theorem 11. $\triangleleft$

The conjugate gradient algorithm, in the form described above, cannot be used for the minimization of non-quadratic functions, as it requires the knowledge of the matrix Q, which is the Hessian of the function f. Note that the matrix Q appears at two levels in the algorithm: in the computation of the scalar β_k required to compute the new direction of research, and in the computation of the step α_k. It is therefore necessary to modify the algorithm to avoid the computation of $\nabla^2 f$, but at the same time it is reasonable to make sure that the modified algorithm coincides with the above one in the quadratic case.

2.7.1 Modification of β_k

To begin with note that, by equation (2.9), β_k can be written as

$$\beta_k = \frac{\nabla f(x_{k+1})' \dfrac{\nabla f(x_{k+1}) - \nabla f(x_k)}{\alpha_k}}{d_k' \dfrac{\nabla f(x_{k+1}) - \nabla f(x_k)}{\alpha_k}} = \frac{\nabla f(x_{k+1})' \left[\nabla f(x_{k+1}) - \nabla f(x_k) \right]}{d_k' \left[\nabla f(x_{k+1}) - \nabla f(x_k) \right]},$$

and, by equation (2.8),

$$\beta_k = -\frac{\nabla f(x_{k+1})' \left[\nabla f(x_{k+1}) - \nabla f(x_k)\right]}{d_k' \nabla f(x_k)}. \tag{2.13}$$

Using equation (2.13), it is possible to construct several expressions for β_k, all equivalent in the quadratic case, but yielding different algorithms in the general (non-quadratic) case. A first possibility is to consider equations (2.10) and (2.11) and to define

$$\beta_k = \frac{\nabla f(x_{k+1})' \nabla f(x_{k+1})}{\nabla f(x_k)' \nabla f(x_k)} = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}, \tag{2.14}$$

which is known as Fletcher-Reeves formula.
A second possibility is to write the denominator as in equation (2.14) and the numerator as in equation (2.13), yielding

$$\beta_k = \frac{\nabla f(x_{k+1})' \left[\nabla f(x_{k+1}) - \nabla f(x_k)\right]}{\|\nabla f(x_k)\|^2}, \tag{2.15}$$

which is known as Polak-Ribiere formula. Finally, it is possible to have the denominator as in (2.13) and the numerator as in (2.14), *i.e.*

$$\beta_k = -\frac{\|\nabla f(x_{k+1})\|^2}{d_k' \nabla f(x_k)}. \tag{2.16}$$

2.7.2 Modification of α_k

As already observed, in the quadratic version of the conjugate gradient method also the step α_k depends upon Q. However, instead of using the α_k given in **Step 2** of the algorithm, it is possible to use a line search along the direction α_k. In this way, an algorithm for non-quadratic functions can be constructed. Note that α_k, in the algorithm for quadratic functions, is also such that $d_k \nabla f(x_{k+1}) = 0$. Therefore, in the line search, it is reasonable to select α_k such that, not only $f(x_{k+1}) < f(x_k)$, but also d_k is approximately orthogonal to $\nabla f(x_{k+1})$.

Remark. The condition of approximate orthogonality between d_k and $\nabla f(x_{k+1})$ cannot be enforced using Armijo method or Goldstein conditions. However, there are more sophisticated line search algorithms, known as Wolfe conditions, which allow to enforce the above constraint. ◇

2.7.3 Polak-Ribiere algorithm

As a result of the modifications discussed in the last sections, it is possible to construct an algorithm for the minimization of general functions. For example, using equation (2.15) we obtain the following algorithm, due to Polak-Ribiere, which has proved to be one of the most efficient among the class of conjugate directions methods.

Step 0. Given $x_0 \in I\!R^n$.

Step 1. Set $k = 0$.

Step 2. Compute $\nabla f(x_k)$. If $\nabla f(x_k) = 0$ STOP. Else let

$$d_k = \begin{cases} -\nabla f(x_0), & \text{if } k = 0 \\ -\nabla f(x_k) + \dfrac{\nabla f(x_k)' \left[\nabla f(x_k) - \nabla f(x_{k-1})\right]}{\|\nabla f(x_{k-1})\|^2} d_{k-1}, & \text{if } k \geq 1. \end{cases}$$

Step 3. Compute α_k performing a line search along d_k.

Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$ and go to **Step 2**.

Remark. The line search has to be sufficiently accurate, to make sure that all directions generated by the algorithm are descent directions. A suitable line search algorithm is the so-called Wolfe method, which is a modification of Goldstein method. ◇

Remark. To guarantee global convergence of a subsequence it is possible to use, every n steps, the direction $-\nabla f$. In this case, it is said that the algorithm uses a *restart* procedure. For the algorithm with restart it is possible to have quadratic speed of convergence in n steps, *i.e*

$$\|x_{k+n} - x_\star\| \leq \gamma \|x_k - x_\star\|^2,$$

for some $\gamma > 0$. ◇

Remark. It is possible to modify Polak-Ribiere algorithm to make sure that at each step the angle condition holds. In this case, whenever the direction d_k does not satisfy the angle condition, it is sufficient to use the direction $-\nabla f$. Note that, enforcing the angle condition, yields a globally convergent algorithm. ◇

Remark. Even if the use of the direction $-\nabla f$ every n steps, or whenever the angle condition is not satisfied, allows to prove global convergence of Polak-Ribiere algorithm, it has been observed in numerical experiments that such modified algorithms do not perform as well as the original one. ◇

2.8 Quasi-Newton methods

Conjugate gradient methods have proved to be more efficient than the gradient method. However, in general, it is not possible to guarantee superlinear convergence. The main advantage of conjugate gradient methods is in the fact that they do not require to construct and store any matrix, hence can be used in large scale problems.

In small and medium scale problems, *i.e.* problems with less then a few hundreds decision variables, in which $\nabla^2 f$ is not available, it is convenient to use the so-called quasi-Newton methods.
Quasi Newton methods, as conjugate directions methods, have been introduced for quadratic functions. They are described by an algorithm of the form

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k),$$

with H_0 given. The matrix H_k is an approximation of $[\nabla^2 f(x_k)]^{-1}$ and it is computed iteratively at each step.
If f is a quadratic function, the gradient of f is given by

$$\nabla f(x) = Qx + c,$$

for some Q and c, hence for any $x \in I\!R^n$ and $y \in I\!R^n$ one has

$$\nabla f(y) - \nabla f(x) = Q(y - x),$$

or, equivalently,

$$Q^{-1}[\nabla f(y) - \nabla f(x)] = y - x.$$

It is then natural, in general, to construct the sequence $\{H_k\}$ such that

$$H_{k+1}[\nabla f(x_{k+1}) - \nabla f(x_k)] = x_{k+1} - x_k. \tag{2.17}$$

Equation (2.17) is known as quasi-Newton equation.
There exist several update methods satisfying the quasi-Newton equation. For simplicity, set

$$\gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k),$$

and

$$\delta_k = x_{k+1} - x_k.$$

As a result, equation (2.17) can be rewritten as

$$H_{k+1}\gamma_k = \delta_k.$$

One of the first quasi-Newton methods has been proposed by Davidon, Fletcher and Powell, and can be summarized by the equations

$$\text{DFP} \begin{cases} H_0 &= I \\ H_{k+1} &= H_k + \dfrac{\delta_k \delta_k'}{\delta_k' \gamma_k} - \dfrac{H_k \gamma_k \gamma_k' H_k}{\gamma_k' H_k \gamma_k}. \end{cases} \tag{2.18}$$

It is easy to show that the matrix H_{k+1} satisfies the quasi-Newton equation (2.17), *i.e.*

$$\begin{aligned} H_{k+1}\gamma_k &= H_k\gamma_k + \frac{\delta_k \delta_k'}{\delta_k' \gamma_k}\gamma_k - \frac{H_k \gamma_k \gamma_k' H_k}{\gamma_k' H_k \gamma_k}\gamma_k \\ &= H_k\gamma_k + \frac{\delta_k' \gamma_k}{\delta_k' \gamma_k}\delta_k - \frac{\gamma_k' H_k \gamma_k}{\gamma_k' H_k \gamma_k} H_k \gamma_k \\ &= \delta_k. \end{aligned}$$

Moreover, it is possible to prove the following fact, which gives conditions such that the matrices generated by DFP method are positive definite for all k.

Theorem 13 *Let $H_k = H_k' > 0$ and assume $\delta_k'\gamma_k > 0$. Then the matrix*

$$H_k + \frac{\delta_k\delta_k'}{\delta_k'\gamma_k} - \frac{H_k\gamma_k\gamma_k'H_k}{\gamma_k'H_k\gamma_k}$$

is positive definite.

DFP method has the following properties. In the quadratic case, if α_k is selected to minimize

$$f(x_k - \alpha H_k \nabla f(x_k)),$$

then

- the directions $d_k = -H_k\nabla f(x_k)$ are mutually conjugate;
- the minimum of the (quadratic) function is found in at most n steps, moreover $H_n = Q^{-1}$;
- the matrices H_k are always positive definite.

In the non-quadratic case

- the matrices H_k are positive definite (hence $d_k = -H_k\nabla f(x_k)$ is a descent direction) if $\delta_k'\gamma_k > 0$;
- it is globally convergent if f is strictly convex and if the line search is exact;
- it has superlinear speed of convergence (under proper hypotheses).

A second, and more general, class of update formulae, including as a particular case DFP formula, is the so-called Broyden class, defined by the equations

$$\text{Broyden} \begin{cases} H_0 &= I \\ H_{k+1} &= H_k + \dfrac{\delta_k\delta_k'}{\delta_k'\gamma_k} - \dfrac{H_k\gamma_k\gamma_k'H_k}{\gamma_k'H_k\gamma_k} + \phi v_k v_k', \end{cases} \tag{2.19}$$

with $\phi \geq 0$ and

$$v_k = (\gamma_k'H_k\gamma_k)^{1/2}\left(\frac{\delta_k}{\delta_k'\gamma_k} - \frac{H_k\gamma_k}{\gamma_k'H_k\gamma_k}\right).$$

If $\phi = 0$ then we obtain DFP formula, whereas for $\phi = 1$ we have the so-called Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula, which is one of the preferred algorithms in applications. From Theorem 13 it is easy to infer that, if $H_0 > 0$, $\gamma_k'\delta_k > 0$ and $\phi \geq 0$, then all formulae in the class of Broyden generate matrices $H_k > 0$.

Remark. Note that the condition $\delta_k'\gamma_k > 0$ is equivalent to

$$(\nabla f(x_{k+1}) - \nabla f(x_k))' d_k > 0,$$

and this can be enforced with a sufficiently precise line search. ◇

For the method based on BFGS formula, a global convergence result, for convex functions and in the case of non-exact (but sufficiently accurate) line search, has been proved. Moreover, it has been shown that the algorithm has superlinear speed of convergence. This algorithm can be summarized as follows.

Step 0. Given $x_0 \in I\!R^n$.

Step 1. Set $k = 0$.

Step 2. Compute $\nabla f(x_k)$. If $\nabla f(x_k) = 0$ STOP. Else compute H_k with BFGS equation and set

$$d_k = -H_k \nabla f(x_k).$$

Step 3. Compute α_k performing a line search along d_k.

Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$ and go to **Step 2**.

In the general case it is not possible to prove global convergence of the algorithm. However, this can be enforced verifying (at the end of **Step 2**), if the direction d_k satisfies an angle condition, and if not use the direction $d_k = -\nabla f(x_k)$. However, as already observed, this modification improves the convergence properties, but reduces (sometimes drastically) the speed of convergence.

2.9 Methods without derivatives

All the algorithms that have been discussed presuppose the knowledge of the derivatives (first and/or second) of the function f. There are, however, also methods which do not require such a knowledge. These methods can be divided in two classes: direct research methods and methods using finite difference approximations.
Direct search methods are based upon the direct comparison of the values of the function f in the points generated by the algorithm, without making use of the necessary condition of optimality $\nabla f = 0$. In this class, the most interesting methods, *i.e.* the methods for which it is possible to give theoretical results, are those that make use cyclically of n linearly independent directions. The simplest possible method, known as the method of the coordinate directions, can be described by the following algorithm.

Step 0. Given $x_0 \in I\!R^n$.

Step 1. Set $k = 0$.

Step 2. Set $j = 0$.

Step 3. Set $d_k = e_j$, where e_j is the j-th coordinate direction.

Step 4. Compute α_k performing a line search without derivatives along d_k.

Step 5. Set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$.

Step 6. If $j < n$ set $j = j + 1$ and go to **Step 3**. If $j = n$ go to **Step 2**.

It is easy to verify that the matrix

$$P_k = \left[\begin{array}{cccc} d_k & d_{k+1} & \cdots & d_{k+n-1} \end{array} \right]$$

is such that

$$|\det P_k| = 1,$$

hence, if the line search is such that

$$\lim_{k\to\infty} \frac{\nabla f(x_k)' d_k}{\|d_k\|} = 0$$

and

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0,$$

convergence to stationary points is ensured by the general result in Theorem 5. Note that, the line search can be performed using the parabolic line search method described in Section 2.4.4.

The method of the coordinate directions is not very efficient, in terms of speed of convergence. Therefore, a series of heuristics have been proposed to improve its performance. One such heuristics is the so-called method of Jeeves and Hooke, in which not only the search along the coordinate directions is performed, but also a search along directions joining pairs of points generated by the algorithm. In this way, the search is performed along what may be considered to be the most promising directions.

An alternative direct search method is the so-called simplex method (which should not be confused with the simplex method of linear programming). The method starts with $n + 1$ (equally spaced) points $x_{(i)} \in I\!R^n$ (these points give a simplex in $I\!R^n$). In each of these points the function f is computed and the vertex where the function f attains the maximum value is determined. Suppose this is the vertex $x_{(n+1)}$. This vertex is reflected with respect to the center of the simplex, *i.e.* the point

$$x_c = \frac{1}{n+1} \sum_{i=1}^{n+1} x_{(i)}.$$

As a result, the new vertex

$$x_{(n+2)} = x_c + \alpha(x_c - x_{(n+1)})$$

where $\alpha > 0$, is constructed, see Figure 2.7. The procedure is then repeated.

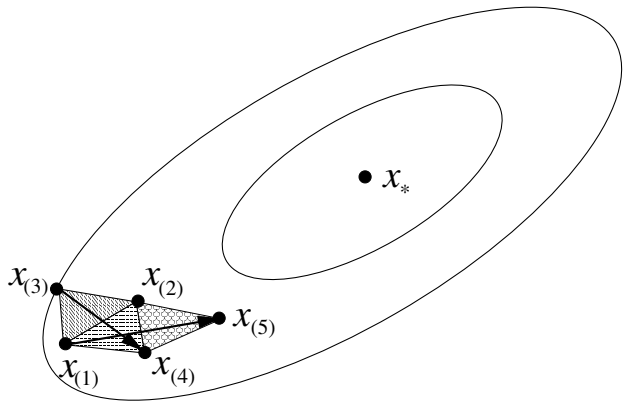

Figure 2.7: The simplex method. The points $x_{(1)}$, $x_{(2)}$ and $x_{(3)}$ yields the starting simplex. The second simplex is given by the points $x_{(1)}$, $x_{(2)}$ and $x_{(4)}$. The third simplex is given by the points $x_{(2)}$, $x_{(4)}$ and $x_{(5)}$.

It is possible that the vertex that is generated by one step of the algorithm is (again) the one where the function f has its maximum. In this case, the algorithm cycles, hence the next vertex has to be determined using a different strategy. For example, it is possible to construct the next vertex by reflecting another of the remaining n vertex, or to shrink the simplex.
As a stopping criterion it is possible to consider the condition

$$\frac{1}{n+1}\sum_{i=1}^{n+1}\left(f(x_{(i)})-\bar{f}\right)^2<\epsilon \tag{2.20}$$

where $\epsilon>0$ is assigned by the designer, and

$$\bar{f}=\frac{1}{n+1}\sum_{i=1}^{n+1}f(x_{(i)}),$$

i.e. $\bar{f}$ is the mean value of the $f(x_{(i)})$. Condition (2.20) implies that the points $x_{(i)}$ are all in a region where the function f is *flat*.
As already observed, direct search methods are not very efficient, and can be used only for problems with a few decision variables and when approximate solutions are acceptable. As an alternative, if the derivatives of the function f are not available, it is possible to resort to numeric differentiation, *e.g* the entries of the gradient of f can be computed using the so-called forward difference approximation, *i.e.*

$$\frac{\partial f(x)}{\partial x_i}\approx\frac{f(x+te_i)-f(x)}{t},$$

where e_i is the i-th column of the identity matrix of dimension n, and $t > 0$ has to be fixed by the user. Note that there are methods for the computation of the optimal value of t, *i.e.* the value of t which minimizes the approximation error.

Chapter 3

Nonlinear programming

3.1 Introduction

In this chapter we discuss the basic tools for the solution of optimization problems of the form

$$P_0 \begin{cases} \min\limits_x f(x) \\ g(x) = 0 \\ h(x) \leq 0. \end{cases} \tag{3.1}$$

In the problem P_0 there are both equality and inequality constraints[1]. However, sometimes for simplicity, or because a method has been developed for problems with special structure, we will refer to problems with only equality constraints, *i.e.* to problems of the form

$$P_1 \begin{cases} \min\limits_x f(x) \\ g(x) = 0, \end{cases} \tag{3.2}$$

or to problems with only inequality constraints, *i.e.* to problems of the form

$$P_2 \begin{cases} \min\limits_x f(x) \\ h(x) \leq 0. \end{cases} \tag{3.3}$$

In all the above problems we have $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, and $h : \mathbb{R}^n \to \mathbb{R}^p$. From a formal point of view it is always possible to transform the equality constraint $g_i(x) = 0$ into a pair of inequality constraints, *i.e.* $g_i(x) \leq 0$ and $-g_i(x) \leq 0$. Hence, problem P_1 can be (equivalently) described by

$$\tilde{P}_1 \begin{cases} \min\limits_x f(x) \\ g(x) \leq 0 \\ -g(x) \leq 0, \end{cases}$$

which is a special case of problem P_2. In the same way, it is possible to transform the inequality constraint $h_i(x) \leq 0$ into the equality constraint $h_i(x) + y_i^2 = 0$, where y_i is an auxiliary variable (also called *slack* variable). Therefore, defining the extended vector $z = [x', y']'$, problem P_2 can be rewritten as

$$\tilde{P}_2 \begin{cases} \min\limits_z f(x) \\ h(x) + Y = 0, \end{cases}$$

with

$$Y = \begin{bmatrix} y_1^2 \\ y_2^2 \\ \vdots \\ y_p^2 \end{bmatrix},$$

which is a special case of problem P_1.

[1]The condition $h(x) \leq 0$ has to be understood element-wise, *i.e.* $h_i(x) \leq 0$ for all i.

Note however, that the transformation of equality constraints into inequality constraints yields an increase in the number of constraints, whereas the transformation of inequality constraints into equality constraints results in an increased number of variables.
Given problem P_0 (or P_1, or P_2), a point x satisfying the constraints is said to be an admissible point, and the set of all admissible points is called the admissible set and it is denoted with $\mathcal{X}$. Note that the problem makes sense only if $\mathcal{X} \neq \emptyset$.
In what follows it is assumed that the functions f, g and h are two times differentiable, however we do not make any special hypothesis on the form of such functions. Note however, that if g and h are linear there are special algorithms, and linear/quadratic programming algorithms are used if f is linear/quadratic and g and h are linear. We do not discuss these special algorithms, and concentrate mainly on algorithms suitable for general problems.

3.2 Definitions and existence conditions

Consider the problem P_0 (or P_1, or P_2). The following definitions are instrumental to provide a necessary condition and a sufficient condition for the existence of local minima.

Definition 6 *An open ball with center $x^\star$ and radius $\theta > 0$ is the set*

$$B(x^\star, \theta) = \{x \in I\!R^n \mid \|x - x^\star\| < \theta\}.$$

Definition 7 *A point $x^\star \in \mathcal{X}$ is a constrained local minimum if there exists $\theta > 0$ such that*

$$f(y) \geq f(x^\star), \tag{3.4}$$

for all $y \in \mathcal{X} \cap B(x^\star, \theta)$.
A point $x^\star \in \mathcal{X}$ is a constrained global minimum if

$$f(y) \geq f(x^\star), \tag{3.5}$$

for all $y \in \mathcal{X}$.
If the inequality (3.4) (or (3.5)) holds with a strict inequality sign for all $y \neq x^\star$ then the minimum is said to be strict.

Definition 8 *The i-th inequality constraints $h_i(x)$ is said to be active at $\tilde{x}$ if $h_i(\tilde{x}) = 0$. The set $I_a(\tilde{x})$ is the set of all indexes i such that $h_i(\tilde{x}) = 0$,* i.e.

$$I_a(\tilde{x}) = \{i \in \{1, 2, \cdots, p\} \mid h_i(\tilde{x}) = 0\}.$$

The vector $h_a(\tilde{x})$ is the subvector of $h(x)$ corresponding to the active constraints, i.e.

$$h_a(\tilde{x}) = \{h_i(\tilde{x}) \mid i \in I_a(\tilde{x}).$$

Definition 9 *A point $\tilde{x}$ is a regular point for the constraints if at $\tilde{x}$ the gradients of the active constraints,* i.e. *the vectors $\nabla g_i(\tilde{x})$, for $i = 1, \cdots, m$ and $\nabla h_i(\tilde{x})$, for $i \in I_a(\tilde{x})$, are linearly independent.*

The definition of regular point is given because, the necessary and the sufficient conditions for optimality, in the case of regular points are relatively simple. To state these conditions, and with reference to problem P_0, consider the Lagrangian function

$$L(x, \lambda, \rho) = f(x) + \lambda' g(x) + \rho' h(x) \tag{3.6}$$

with $\lambda \in I\!R^m$ and $\rho \in I\!R^p$. The vectors λ and ρ are called multipliers.
With the above ingredients and definitions it is now possible to provide a necessary condition and a sufficient condition for local optimality.

Theorem 14 *[First order necessary condition] Consider problem P_0. Suppose $x^\star$ is a local solution of the problem P_0, and $x^\star$ is a regular point for the constraints.*
Then there exist (unique) multipliers $\lambda^\star$ and $\rho^\star$ such that[2]

$$\begin{aligned} &\nabla_x L(x^\star, \lambda^\star, \rho^\star) = 0 \\ &g(x^\star) = 0 \\ &h(x^\star) \leq 0 \\ &\rho^\star \geq 0 \\ &(\rho^\star)' h(x^\star) = 0. \end{aligned} \tag{3.7}$$

Conditions (3.7) are known as Kuhn-Tucker conditions.

Definition 10 *Let $x^\star$ be a local solution of problem P_0 and let $\rho^\star$ be the corresponding (optimal) multiplier. At $x^\star$ the condition of strict complementarity holds if $\rho_i^\star > 0$ for all $i \in I_a(x^\star)$.*

Theorem 15 *[Second order sufficient condition] Consider the problem P_0. Assume that there exist $x^\star$, $\lambda^\star$ and $\rho^\star$ satisfying conditions (3.7). Suppose moreover that $\rho^\star$ is such that the condition of strict complementarity holds at $x^\star$. Suppose finally that*

$$s' \nabla^2_{xx} L(x^\star, \lambda^\star, \rho^\star) s > 0 \tag{3.8}$$

for all $s \neq 0$ such that

$$\begin{bmatrix} \dfrac{\partial g(x^\star)}{\partial x} \\ \dfrac{\partial h_a(x^\star)}{\partial x} \end{bmatrix} s = 0.$$

Then $x^\star$ is a strict constrained local minimum of problem P_0.

Remark. Necessary and sufficient conditions for a global minimum can be given under proper convexity hypotheses, *i.e.* if the function f is convex in $\mathcal{X}$, and if $\mathcal{X}$ is a convex set. This is the case, for example if there are no inequality constraints and if the equality constraints are linear. ◇

[2] We denote with $\nabla_x f$ the vector of the partial derivatives of f with respect to x.

Remark. If all points in $\mathcal{X}$ are regular points for the constraints then conditions (3.7) yield a set of points $\mathcal{P}$, *i.e.* the points satisfying conditions (3.7), and among these points there are all constrained local minima (and also the constrained global minimum, if it exists). However, if there are points in $\mathcal{X}$ which are not regular points for the constraints, then the set $\mathcal{P}$ may not contain all constrained local minima. These have to be searched in the set $\mathcal{P}$ and in the set of non-regular points. $\diamond$

Remark. In what follows, we will always tacitly assume that the conditions of regularity and of strict complementarity hold. $\diamond$

3.2.1 A simple proof of Kuhn-Tucker conditions for equality constraints

Consider problem P_1, *i.e.* a minimization problem with only equality constraints, and a point $x^\star$ such that $g(x^\star) = 0$, *i.e.* $x^\star \in \mathcal{X}$. Suppose that[3]

$$\text{rank}\frac{\partial g}{\partial x}(x^\star) = m$$

i.e. $x^\star$ is a regular point for the constraints, and that $x^\star$ is a constrained local minimum. By the implicit function theorem, there exist a neighborhood of $x^\star$, a partition of the vector x, *i.e.*

$$x = \begin{bmatrix} u \\ v \end{bmatrix},$$

with $u \in I\!R^m$ and $v \in I\!R^{n-m}$, and a function ϕ such that the constrains $g(x) = 0$ can be (locally) rewritten as

$$u = \phi(v).$$

As a result (locally)

$$\begin{cases} \min\limits_x f(x) \\ g(x) = 0 \end{cases} \quad \Leftrightarrow \quad \begin{cases} \min\limits_{u,v} f(u,v) \\ u = \phi(v) \end{cases} \quad \Leftrightarrow \quad \min_v f(\phi(v), v),$$

i.e. problem P_1 is (locally) equivalent to a unconstrained minimization problem. Therefore

$$0 = \nabla f(\phi(v^\star), v^\star) = \left(\frac{\partial f}{\partial u}\frac{\partial \phi}{\partial v} + \frac{\partial f}{\partial v}\right)_{x^\star} = \left(-\frac{\partial f}{\partial u}\left(\frac{\partial g}{\partial u}\right)^{-1}\frac{\partial g}{\partial v} + \frac{\partial f}{\partial v}\right)_{x^\star}.$$

Setting

$$\lambda^\star = \left(-\frac{\partial f}{\partial u}\left(\frac{\partial g}{\partial u}\right)^{-1}\right)'_{x^\star}$$

yields

$$\left(\frac{\partial f}{\partial v} + (\lambda^\star)'\frac{\partial g}{\partial v}\right)_{x^\star} = 0 \tag{3.9}$$

[3]Note that m is the number of the equality constraints, and that, to avoid trivial cases, $m < n$.

and

$$\left(\frac{\partial f}{\partial u} + (\lambda^\star)' \frac{\partial g}{\partial u}\right)_{x^\star} = 0. \tag{3.10}$$

Finally, let

$$L = f + \lambda' g,$$

note that equations (3.9) and (3.10) can be rewritten as

$$\nabla_x L(x^\star, \lambda^\star) = 0,$$

and this, together with $g(x^\star) = 0$, is equivalent to equations (3.7).

3.2.2 Quadratic cost function with linear equality constraints

Consider the function

$$f(x) = \frac{1}{2} x' Q x,$$

with $x \in I\!R^n$ and $Q = Q' > 0$, the equality constraints

$$g(x) = Ax - b = 0,$$

with $b \in I\!R^m$ and $m < n$, and the Lagrangian function

$$L(x, \lambda) = \frac{1}{2} x' Q x + \lambda'(Ax - b).$$

A simple application of Theorem 14 yields the necessary conditions of optimality

$$\begin{array}{rcl} \nabla_x L(x^\star, \lambda^\star) & = & Qx^\star + A'\lambda^\star = 0 \\ g(x^\star) & = & Ax^\star - b = 0. \end{array} \tag{3.11}$$

Suppose now that the matrix A is such that $AQ^{-1}A'$ is invertible[4]. As a result, the only solution of equations (3.11) is

$$x^\star = Q^{-1}A'(AQ^{-1}A')^{-1}b \qquad \lambda^\star = -(AQ^{-1}A')^{-1}b.$$

Finally, by Theorem 15, it follows that $x^\star$ is a strict constrained (global) minimum.

3.3 Nonlinear programming methods: introduction

The methods of non-linear programming that have been mostly studied in recent years belong to two categories. The former includes all methods based on the transformation of a constrained problem into one of more unconstrained problems, in particular the so-called (exact or sequential) penalty function methods and (exact or sequential) augmented Lagrangian methods. Sequential methods are based on the solution of a sequence of problems, with the property that the sequence of the solutions of the subproblems converge

[4]This is the case if rank$A = m$.

to the solution of the original problem. Exact methods are based on the fact that, under suitable assumptions, the optimal solutions of an unconstrained problem coincides with the optimal solution of the original problem.
The latter includes the methods based on the transformation of the original problem into a sequence of constrained quadratic problems.
From the above discussion it is obvious that, to construct algorithms for the solution of non-linear programming problems, it is necessary to use efficient unconstrained optimization routines.
Finally, in any practical implementation, it is also important to quantify the complexity of the algorithms in terms of complexity (and number) of operations (inversion of matrices, differentiation, ...) and the speed of convergence. These issues are still largely open, and will not be addressed in this notes.

3.4 Sequential and exact methods

3.4.1 Sequential penalty functions

In this section we study the so-called external sequential penalty functions. This name is based on the fact that the solutions of the resulting unconstrained problems are in general not admissible. There are also internal penalty functions (known as barrier functions) but this can be used only for problems in which the admissible set has a non-empty interior. As a result, such functions cannot be used in the presence of equality constraints.
The basic idea of external sequential penalty functions is very simple. Consider problem P_0, the function

$$q(x) = \begin{cases} 0, & \text{if } x \in \mathcal{X} \\ \\ +\infty, & \text{if } x \notin \mathcal{X} \end{cases} \tag{3.12}$$

and the function

$$F = f + q. \tag{3.13}$$

It is obvious that the unconstrained minimization of F yields a solution of problem P_0. However, because of its discontinuous nature, the minimization of F cannot be performed. Nevertheless, it is possible to construct a sequence of continuously differentiable functions, converging to F, and it is possible to study the convergence of the minima of such a sequence of functions to the solutions of problem P_0.
For, consider a continuously differentiable function p such that

$$p(x) = \begin{cases} 0, & \text{if } x \in \mathcal{X} \\ \\ > 0, & \text{if } x \notin \mathcal{X}, \end{cases} \tag{3.14}$$

and the function

$$F_\epsilon = f + \frac{1}{\epsilon} p,$$

with $\epsilon > 0$. It is obvious that[5]

$$\lim_{\epsilon \to 0} F_\epsilon = F.$$

The function F_ϵ is known as external penalty function. The attribute external is due to the fact that, if $\bar{x}$ is a minimum of F_ϵ in general $p(\bar{x}) \neq 0$, *i.e.* $\bar{x} \notin \mathcal{X}$. The term $\frac{1}{\epsilon}p$ is called penalty term, as it penalizes the violation of the constraints. In general, the function p has the following form

$$p = \sum_{i=1}^{m} (g_i)^2 + \sum_{i=1}^{p} (\max(0, h_i))^2. \tag{3.15}$$

Consider now a strictly decreasing sequence $\{\epsilon_k\}$ such that $\lim_{k\to\infty} \epsilon_k = 0$. The sequential penalty function method consists in solving the sequence of unconstrained problems

$$\min_x F_{\epsilon_k}(x),$$

with $x \in I\!R^n$. The most important convergence results for this methods are summarized in the following statements.

Theorem 16 *Consider the problem P_0. Suppose that for all $\sigma > 0$ the set*[6]

$$\mathcal{X}^\sigma = \{x \in I\!R^n \mid |g_i(x)| \leq \sigma, i = 1, \cdots, m\} \cap \{x \in I\!R^n \mid h_i(x) \leq \sigma, i = 1, \cdots, p\}$$

is compact. Suppose moreover that for all k the function $F_{\epsilon_k}(x)$ has a global minimum x_k. Then the sequence $\{x_k\}$ has (at least) one converging subsequence, and the limit of any converging subsequence is a global minimum for problem P_0.

Theorem 17 *Let $x^\star$ be a strict constrained local minimum for problem P_0. Then there exist a sequence $\{x_k\}$ and an integer $\bar{k} > 0$ such that $\{x_k\}$ converges to $x^\star$ and, for all $k \geq \bar{k}$, x_k is a local minimum of $F_{\epsilon_k}(x)$.*

The construction of the function F_ϵ is apparently very simple, and this is the main advantage of the method. However, the minimization of the function F_ϵ may be difficult, especially for small values of ϵ. In fact, it is possible to show, even via simple examples, that as ϵ tends to zero the Hessian matrix of the function F_ϵ become ill conditioned. As a result, any unconstrained minimization algorithm used to minimize F_ϵ has a very slow convergence rate. To alleviate this problem, it is possible to use, in the minimization of $F_{\epsilon_{k+1}}$, as initial point the point x_k. However, this is close to the minimum of $F_{\epsilon_{k+1}}$ only if ϵ_{k+1} is close to ϵ_k, *i.e.* only if the sequence $\{\epsilon_k\}$ converges slowly to zero.
We conclude that, to avoid the ill conditioning of the Hessian matrix of F_ϵ, hence the slow convergence of each unconstrained optimization problem, it is necessary to slow down the convergence of the sequence $\{x_k\}$, *i.e.* slow convergence is an intrinsic property of the method. This fact has motivated the search for alternatives methods, as described in the next sections.

[5] Because of the discontinuity of F, the limit has to be considered with proper *care*.
[6] The set $\mathcal{X}^\sigma$ is sometimes called the relaxed admissible set.

Remark. It is possible to show that the local minima of F_ϵ describe (continuous) trajectories that can be extrapolated. This observation is exploited in some sophisticated methods for the selection of initial estimate for the point x_k. However, even with the addition of this extrapolation procedure, the convergence of the method remains slow. ◇

Remark. Note that, if the function p is defined as in equation (3.15), then the function F_ϵ is not two times differentiable everywhere, *i.e.* it is not differentiable in all points in which an inequality constraints is active. This property restricts the class of minimization algorithms that can be used to minimize F_ϵ. ◇

3.4.2 Sequential augmented Lagrangian functions

Consider problem P_1, *i.e.* an optimization problem with only equality constraints. For such a problem the Lagrangian function is

$$L = f + \lambda' g,$$

and the first order necessary conditions require the existence of a multiplier $\lambda^\star$ such that, for any local solution $x^\star$ of problem P_1 one has

$$\begin{aligned} &\nabla_x L(x^\star, \lambda^\star) = 0 \\ &\nabla_\lambda L(x^\star, \lambda^\star) = g(x^\star) = 0. \end{aligned} \tag{3.16}$$

The first of equations (3.16) is suggestive of the fact that the function $L(x, \lambda^\star)$ has a unconstrained minimum in $x^\star$. This is actually not the case, as $L(x, \lambda^\star)$ is not convex in a neighborhood of $x^\star$. However it is possible to render the function $L(x, \lambda^\star)$ convex with the addition of a penalty term, yielding the new function, known as augmented Lagrangian function[7],

$$L_a(x, \lambda^\star) = L(x, \lambda^\star) + \frac{1}{\epsilon}\|g(x)\|^2, \tag{3.17}$$

which, for ϵ sufficiently small, but such that $1/\epsilon$ is finite, has a unconstrained minimum in $x^\star$. This intuitive discussion can be given a formal justification, as shown in the next statement.

Theorem 18 *Suppose that at $x^\star$ and $\lambda^\star$ the sufficient conditions for a strict constrained local minimum for problem P_1 hold. Then there exists $\bar\epsilon > 0$ such that for any $\epsilon \in (0, \bar\epsilon)$ the point $x^\star$ is a unconstrained local minimum for the function $L_a(x, \lambda^\star)$.*
Vice-versa, if for some $\bar\epsilon$ and $\lambda^\star$, at $x^\star$ the sufficient conditions for a unconstrained local minimum for the function $L_a(x, \lambda^\star)$ hold, and $g(x^\star) = 0$, then $x^\star$ is a strict constrained local minimum for problem P_1.

The above theorem highlights the fact that, under the stated assumptions, the function $L_a(x, \lambda^\star)$ is an (external) penalty function, with the property that, to give local minima

[7] To be precise we should write $L_a(x, \lambda^\star, \epsilon)$, however we omit the argument ϵ.

for problem P_1 it is not necessary that $\epsilon \to 0$. Unfortunately, this result is not of practical interest, because it requires the knowledge of $\lambda^\star$. To obtain a useful algorithm, it is possible to make use of the following considerations.
By the implicit function theorem, applied to the first of equation (3.16), we infer that there exist a neighborhood of $\lambda^\star$, a neighborhood of $x^\star$, and a continuously differentiable function $x(\lambda)$ such that (locally)

$$\nabla_x L_a(x(\lambda), \lambda) = 0.$$

Moreover, for any $\epsilon \in (0, \bar{\epsilon})$, as $\nabla^2_{xx} L_a(x^\star, \lambda^\star)$ is positive definite also $\nabla^2_{xx} L_a(x, \lambda)$ is locally positive definite. As a result, $x(\lambda)$ is the only value of x that, for any fixed λ, minimizes the function $L_a(x, \lambda)$. It is therefore reasonable to assume that if λ_k is a good estimate of $\lambda^\star$, then the minimization of $L_a(x, \lambda_k)$ for a sufficiently small value of ϵ, yields a point x_k which is a good approximation of $x^\star$.
On the basis of the above discussion it is possible to construct the following minimization algorithm for problem P_1.

Step 0. Given $x_0 \in I\!R^n$, $\lambda_1 \in I\!R^m$ and $\epsilon_1 > 0$.

Step 1. Set $k = 1$.

Step 2. Find a local minimum x_k of $L_a(x, \lambda_k)$ using any unconstrained minimization algorithm, with starting point x_{k-1}.

Step 3. Compute a new estimate λ_{k+1} of $\lambda^\star$.

Step 4. Set $\epsilon_{k+1} = \beta\epsilon_k$, with $\beta = 1$ if $\|g(x_{k+1})\| \leq \frac{1}{4}\|g(x_k)\|$ or $\beta < 1$ otherwise.

Step 5. Set $k = k + 1$ and go to **Step 2**.

In **Step 3** it is necessary to construct a new estimate λ_{k+1} of λ_k. This can be done with proper considerations on the function $L_a(x(\lambda), \lambda)$, introduced in the above discussion. However, without providing the formal details, we mention that one of the most used update laws for λ are described by the equations

$$\lambda_{k+1} = \lambda_k + \frac{2}{\epsilon_k} g(x_k), \tag{3.18}$$

or

$$\lambda_{k+1} = \lambda_k - \left[\nabla^2 L_a(x(\lambda_k), \lambda_k)\right]^{-1} g(x_k), \tag{3.19}$$

whenever the indicated inverse exists.
Note that the convergence of the sequence $\{x_k\}$ to $x^\star$ is limited by the convergence of the sequence $\{\lambda_k\}$ to $\lambda^\star$. It is possible to prove that, if the update law (3.18) is used then the algorithm as linear convergence, whereas if (3.19) is used the convergence is superlinear.

Remark. Similar considerations can be done for problem P_2. For, recall that problem P_2 can be recast, increasing the number of variables, as an optimization problem with equality

constraints, *i.e.* problem $\tilde{P}_2$. For such an *extended* problem, consider the augmented Lagrangian

$$L_a(x, y, \rho) = f(x) + \sum_{i=1}^{p} \rho_i \left(h_i(x) + y_i^2 \right) + \frac{1}{\epsilon} \sum_{i=1}^{p} \left(h_i(x) + y_i^2 \right)^2,$$

and note that, in principle, it would be possible to make use of the results developed with reference to problem P_1. However, the function L_a can be analytically minimized with respect to the variables y_i. In fact, a simple computation shows that the (global) minimum of L_a as a function of y is attained at

$$y_i(x, \rho) = \sqrt{-\min\left(0, h_i(x) + \frac{\epsilon}{2}\rho_i\right)}.$$

As a result, the augmented Lagrangian function for problem P_2 is given by

$$L_a(x, \rho) = f(x) + \rho' h(x) + \frac{1}{\epsilon}\|h(x)\|^2 - \frac{1}{\epsilon} \sum_{i=1}^{p} \left(\min(0, h_i(x) + \frac{\epsilon}{2}\rho_i) \right)^2.$$

◇

3.4.3 Exact penalty functions

An exact penalty function, for a given constrained optimization problem, is a function of the same variables of the problem with the property that its unconstrained minimization yields a solution of the original problem. The term *exact* as opposed to *sequential* indicates that only one, instead of several, minimization is required.
Consider problem P_1, let $x^\star$ be a local solution and let $\lambda^\star$ be the corresponding multiplier. The basic idea of exact penalty functions methods is to determine the multiplier λ appearing in the augmented Lagrangian function as a function of x, *i.e.* $\lambda = \lambda(x)$, with $\lambda(x^\star) = \lambda^\star$. With the use of this function one has[8]

$$L_a(x, \lambda(x)) = f(x) + \lambda(x)' g(x) + \frac{1}{\epsilon}\|g(x)\|^2.$$

The function $\lambda(x)$ is obtained considering the necessary condition of optimality

$$\nabla_x L_a(x^\star, \lambda^\star) = \nabla f(x^\star) + \frac{\partial g(x^\star)'}{\partial x}\lambda^\star = 0 \tag{3.20}$$

and noting that, if $x^\star$ is a regular point for the constraints then equation (3.20) can be solved for $\lambda^\star$ yielding

$$\lambda^\star = -\left(\frac{\partial g(x^\star)}{\partial x} \frac{\partial g(x^\star)'}{\partial x} \right)^{-1} \frac{\partial g(x^\star)}{\partial x} \nabla f(x^\star).$$

[8] As in previous sections we omit the argument ϵ.

This equality suggests to define the function $\lambda(x)$ as

$$\lambda(x) = -\left(\frac{\partial g(x)}{\partial x}\frac{\partial g(x)'}{\partial x}\right)^{-1}\frac{\partial g(x)}{\partial x}\nabla f(x),$$

and this is defined at all x where the indicated inverse exists, in particular at $x^\star$.
It is possible to show that this selection of $\lambda(x)$ yields and exact penalty function for problem P_1. For, consider the function

$$G(x) = f(x) - g(x)'\left(\frac{\partial g(x)}{\partial x}\frac{\partial g(x)'}{\partial x}\right)^{-1}\frac{\partial g(x)}{\partial x}\nabla f(x) + \frac{1}{\epsilon}\|g(x)\|^2,$$

which is defined and differentiable in the set

$$\tilde{\mathcal{X}} = \{x \in I\!R^n \parallel \text{rank}\frac{\partial g(x)}{\partial x} = m\}, \tag{3.21}$$

and the following statements.

Theorem 19 *Let $\bar{\mathcal{X}}$ be a compact subset of $\tilde{\mathcal{X}}$. Assume that $x^\star$ is the only global minimum of f in $\mathcal{X} \cap \bar{\mathcal{X}}$ and that $x^\star$ is in the interior of $\bar{\mathcal{X}}$. Then there exists $\bar{\epsilon} > 0$ such that, for any $\epsilon \in (0, \bar{\epsilon})$, $x^\star$ is the only global minimum of G in $\bar{\mathcal{X}}$.*

Theorem 20 *Let $\bar{\mathcal{X}}$ be a compact subset of $\tilde{\mathcal{X}}$. Then there exists $\bar{\epsilon} > 0$ such that, for any $\epsilon \in (0, \bar{\epsilon})$, if $x^\star$ is a unconstrained minimum of $G(x)$ and $x^\star \in \bar{\mathcal{X}}$, then $x^\star$ is a constrained local minimum for problem P_1.*

Theorems 19 and 20 show that it is legitimate to search solutions of problem P_1 minimizing the function G for sufficiently small values of ϵ. Note that it is possible to prove stronger results, showing that there is (under certain hypotheses) a one to one correspondence between the minima of problem P_1 and the minima of the function G, provided ϵ is sufficiently small.
For problem P_2 it is possible to proceed as discussed in Section 3.4.2, *i.e.* transforming problem P_2 into problem $\tilde{P}_2$ and then minimizing analytically with respect to the new variables y_i. However, a different and more direct route can be taken. Consider problem P_2 and the necessary conditions

$$\nabla_x L(x^\star, \rho^\star) = \nabla f(x^\star) + \frac{\partial h(x^\star)'}{\partial x}\rho^\star = 0 \tag{3.22}$$

and

$$\rho_i^\star h_i(x^\star) = 0, \tag{3.23}$$

for $i = 1, \cdots, p$. Premultiplying equation (3.22) by $\frac{\partial h(x^\star)}{\partial x}$ and equation (3.23) by $\gamma^2 h_i(x^\star)$, with $\gamma > 0$, and adding, yields

$$\left(\frac{\partial h(x^\star)}{\partial x}\frac{\partial h(x^\star)'}{\partial x} + \gamma^2 H^2(x^\star)\right)\rho^\star + \frac{\partial h(x^\star)}{\partial x}\nabla f(x^\star) = 0,$$

where

$$H(x^\star) = \mathrm{diag}(h_1(x^\star), \cdots, h_p(x^\star)).$$

As a result, a natural candidate for the function $\rho(x)$ is

$$\rho(x) = -\left(\frac{\partial h(x)}{\partial x}\frac{\partial h(x)'}{\partial x} + \gamma^2 H^2(x)\right)^{-1}\frac{\partial h(x)}{\partial x}\nabla f(x),$$

which is defined whenever the indicated inverse exists, in particular in the neighborhood of any regular point. With the use of this function, it is possible to define an exact penalty function for problem P_2 and to establish results similar to those illustrated in Theorems 19 and 20.

The exact penalty functions considered in this section provide, in principle, a theoretically sound way of solving constrained optimization problem. However, in practice, they have two major drawbacks. Firstly, at each step, it is necessary to invert a matrix with dimension equal to the number of constraint. This operation is numerically ill conditioned if the number of constraints is large. Secondly, the exact penalty functions may not be sufficiently regular to allow the use of unconstrained minimization methods with fast speed of convergence, *e.g.* Newton method.

3.4.4 Exact augmented Lagrangian functions

An exact augmented Lagrangian function, for a given constrained optimization problem, is a function, defined on an augmented space with dimension equal to the number of variables plus the number of constraint, with the property that its unconstrained minimization yields a solution of the original problem. Moreover, in the computation of such a function it is not necessary to invert any matrix.

To begin with, consider problem P_1 and recall that, for such a problem, a sequential augmented Lagrangian function has been defined adding to the Lagrangian function a term, namely $\frac{1}{\epsilon}\|g(x)\|^2$, which penalizes the violation of the necessary condition $g(x) = 0$. This term, for ϵ sufficiently small, renders the function L_a convex in a neighborhood of $x^\star$. A *complete* convexification can be achieved adding a further term that penalizes the violation of the necessary condition $\nabla_x L(x, \lambda) = 0$. More precisely, consider the function

$$S(x, \lambda) = f(x) + \lambda' g(x) + \frac{1}{\epsilon}\|g(x)\|^2 + \eta\|\frac{\partial g(x)}{\partial x}\nabla_x L(x, \lambda)\|^2, \tag{3.24}$$

with $\epsilon > 0$ and $\eta > 0$. The function (3.24) is continuously differentiable and it is such that, for ϵ sufficiently small, the solutions of problem P_1 are in a one to one correspondence with the points (x, λ) which are local minima of S, as detailed in the following statements.

Theorem 21 *Let $\bar{\mathcal{X}}$ be a compact set. Suppose $x^\star$ is the unique global minimum of f in the set $\mathcal{X} \cap \bar{\mathcal{X}}$ and $x^\star$ is an interior point of $\bar{\mathcal{X}}$. Let $\lambda^\star$ be the multiplier associated to $x^\star$. Then, for any compact set $\Lambda \subset \mathbb{R}^m$ such that $\lambda^\star \in \Lambda$ there exists $\bar{\epsilon}$ such that, for all $\epsilon \in (0, \bar{\epsilon})$, $(x^\star, \lambda^\star)$ is the unique global minimum of S in $\mathcal{X} \times \Gamma$.*

Theorem 22 *Let*[9] $\mathcal{X} \times \Lambda \subset \tilde{\mathcal{X}} \times I\!R^m$ *be a compact set. Then there exists* $\bar{\epsilon} > 0$ *such that, for all* $\epsilon \in (0, \bar{\epsilon})$*, if* $(x^\star, \lambda^\star)$ *is a unconstrained local minimum of* S*, then* $x^\star$ *is a constrained local minimum for problem* P_1 *and* $\lambda^\star$ *is the corresponding multiplier.*

Theorems 21 and 22 justify the use of a unconstrained minimization algorithm, applied to the function S, to find local (or global) solutions of problem P_1.
Problem P_2 can be dealt with using the same considerations done in Section 3.4.2.

3.5 Recursive quadratic programming

Recursive quadratic programming methods have been widely studied in the past years. In this section we provide a preliminary description of such methods. For, consider problem P_1 and suppose that $x^\star$ and $\lambda^\star$ are such that the necessary conditions (3.7) hold. Consider now a series expansion of the function $L(x, \lambda^\star)$ in a neighborhood of $x^\star$, *i.e.*

$$L(x, \lambda^\star) = f(x^\star) + \frac{1}{2}(x - x^\star)' \nabla^2_{xx} L(x^\star, \lambda^\star)(x - x^\star) + ...$$

a series expansion of the constraint, *i.e.*

$$0 = g(x) = g(x^\star) + \frac{\partial g(x^\star)}{\partial x}(x - x^\star) + ...$$

and the problem

$$\widetilde{PQ}_1 \begin{cases} \min\limits_x f(x^\star) + \frac{1}{2}(x - x^\star)' \nabla^2_{xx} L(x^\star, \lambda^\star)(x - x^\star) \\ \frac{\partial g(x^\star)}{\partial x}(x - x^\star) = 0. \end{cases}$$

Note that problem $\widetilde{PQ}_1$ has the solution $x^\star$, and the corresponding multiplier is $\lambda = 0$, which is not equal (in general) to $\lambda^\star$. This phenomenon is called *bias* of the multiplier, and can be avoided by modifying the objective function and considering the new problem

$$PQ_1 \begin{cases} \min\limits_x f(x^\star) + \nabla f(x^\star)(x - x^\star) + \frac{1}{2}(x - x^\star)' \nabla^2_{xx} L(x^\star, \lambda^\star)(x - x^\star) \\ \frac{\partial g(x^\star)}{\partial x}(x - x^\star) = 0, \end{cases} \tag{3.25}$$

which has solution $x^\star$ with multiplier $\lambda^\star$. This observation suggests to consider the sequence of quadratic programming problems

$$PQ_1^{k+1} \begin{cases} \min\limits_\delta f(x_k) + \nabla f(x_k)\delta + \frac{1}{2}\delta' \nabla^2_{xx} L(x_k, \lambda_k)\delta \\ \frac{\partial g(x_k)}{\partial x}\delta = 0, \end{cases} \tag{3.26}$$

[9] The set $\tilde{\mathcal{X}}$ is defined as in equation (3.21).

where $\delta = x - x_k$, and x_k and λ_k are the current estimates of the solution and of the multiplier. The solution of problem PQ_1^{k+1} yields new estimates x_{k+1} and λ_{k+1}. To assess the local convergence of the method, note that the necessary conditions for problem PQ_1^{k+1} yields the system of equations

$$\begin{bmatrix} \nabla^2_{xx} L(x_k, \lambda_k) & \dfrac{\partial g(x_k)'}{\partial x} \\ \dfrac{\partial g(x_k)}{\partial x} & 0 \end{bmatrix} \begin{bmatrix} \delta \\ \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) \\ g(x_k) \end{bmatrix}, \tag{3.27}$$

and this system coincides with the system arising from the application of Newton method to the solution of the necessary conditions for problem P_1. As a consequence, the solutions of the problems PQ_1^{k+1} converge to a solution of problem P_1 under the same hypotheses that guarantee the convergence of Newton method.

Theorem 23 *Let $x^\star$ be a strict constrained local minimum for problem P_1, and let $\lambda^\star$ be the corresponding multiplier. Suppose that for $x^\star$ and $\lambda^\star$ the sufficient conditions of Theorem 15 hold. Then there exists an open neighborhood $\Omega \subset I\!R^n \times I\!R^m$ of the point $(x^\star, \lambda^\star)$ such that, if $(x_0, \lambda_0) \in \Omega$, the sequence $\{x_k, \lambda_k\}$ obtained solving the sequence of quadratic programming problems PQ_1^{k+1}, with $k = 0, 1, \cdots$, converges to $(x^\star, \lambda^\star)$.*
Moreover, the speed of convergence is superlinear, and, if f and g are three times differentiable, the speed of convergence is quadratic.

Remark. It is convenient to solve the sequence of quadratic programming problems PQ_1^{k+1}, instead of solving the equations (3.27) with Newton method, because, for the former it is possible to exclude converge to maxima or saddle points. ◇

In the case of problem P_2, using considerations similar to the one above, it is easy to obtain the following sequence of quadratic programming problems

$$PQ_2^{k+1} \begin{cases} \min\limits_{\delta} f(x_k) + \nabla f(x_k)\delta + \dfrac{1}{2}\delta' \nabla^2_{xx} L(x_k, \lambda_k)\delta \\ \dfrac{\partial h(x_k)}{\partial x}\delta + h(x_k) \leq 0. \end{cases} \tag{3.28}$$

This sequence of problems has to be solved iteratively to generate a sequence $\{x_k, \lambda_k\}$ that, under hypotheses similar to those of Theorem 23, converges to a strict constrained local minimum of problem P_2.
The method described are the basis for a large class of iterative methods.
A first disadvantage of the proposed iterative schemes is that it is necessary to compute the second derivatives of the functions of the problem. This computation can be avoided, using the same philosophy of quasi-Newton methods.
A second disadvantage is in the fact that, being based on Newton algorithm, only local convergence can be guaranteed. However, it is possible to combine the method with global convergent methods: these are used to generate a pair $(\tilde{x}, \tilde{\lambda})$ sufficiently close to $(x^\star, \lambda^\star)$

and then recursive quadratic programming methods are used to obtain fast convergence to $(x^\star, \lambda^\star)$.
A third disadvantage is in the fact that there is no guarantee that the sequence of admissible sets generated by the algorithm is non-empty at each step.
Finally, it is worth noting that, contrary to most of the existing methods, quadratic programming methods do not rely on the use of a penalty term.

Remark. There are several alternative recursive quadratic programming methods which alleviate the drawbacks of the methods described. These are (in general) based on the use of quadratic approximation of penalty functions. For brevity, we do not discuss these methods. ◇

3.6 Concluding remarks

In this section we briefly summarize advantages and disadvantages of the nonlinear programming methods discussed.
Sequential penalty functions methods are very simple to implement, but suffer from the ill conditioning associated to large penalties (*i.e.* to small values of ϵ). As a result, these methods can be used if approximate solutions are acceptable, or in the determination of initial estimates for more precise, but only locally convergent, methods. Note, in fact, that not only an approximation of the solution $x^\star$ can be obtained, but also an approximation of the corresponding multiplier $\lambda^\star$. For example, for problem P_1, a (approximate) solution $\bar{x}$ is such that

$$\nabla F_{\epsilon_k}(\bar{x}) = \nabla f(\bar{x}) + \frac{2}{\epsilon_k}\frac{\partial g(\bar{x})}{\partial x} g(\bar{x}) = 0,$$

hence, the term $\frac{2}{\epsilon_k} g(\bar{x})$ provides an approximation of $\lambda^\star$.
Sequential augmented Lagrangian functions do not suffer from ill conditioning, and yield faster speed of convergence then that attainable using sequential penalty functions.
The methods based on exact penalty functions do not require the solution of a sequence of problems. However, they require the inversion of a matrix of dimension equal to the number of constraints, hence their applicability is limited to problems with a small number of constraints.
Exact augmented Lagrangian functions can be built without inverting any matrix. However, the minimization has to be performed in an extended space.
Recursive quadratic programming methods require the solution, at each step, of a constrained quadratic programming problem. Their main problem is that there is no guarantee that the admissible set is non-empty at each step.
We conclude that it is not possible to decide which is the best method. Each method has its own advantages and disadvantages. Therefore, the selection of a nonlinear programming method has to be driven by the nature of the problem: and has to take into consideration the number of variables, the regularity of the involved functions, the required precision, the computational cost, ...

Chapter 4

Global optimization

4.1 Introduction

Given a function $f : \mathbb{R}^n \to \mathbb{R}$, global optimization methods aim at finding the global minimum of f, *i.e.* a point $x^\star$ such that

$$f(x^\star) \leq f(x)$$

for all $x \in \mathbb{R}^n$. Among these methods it is possible to distinguish between deterministic methods and probabilistic methods.
In the following sections we provide a very brief introductions to global minimization methods. It is worth noting that this is an active area of research.

4.2 Deterministic methods

4.2.1 Methods for Lipschitz functions

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ and suppose it is Lipschitz with constant $L > 0$, *i.e.*

$$|f(x_1) - f(x_2)| \leq L\|x_1 - x_2\|, \tag{4.1}$$

for all $x_1 \in \mathbb{R}^n$ and $x_2 \in \mathbb{R}^n$. Note that equation (4.1) implies that

$$f(x) \geq f(x_0) - L\|x - x_0\| \tag{4.2}$$

and that

$$f(x) \leq f(x_0) + L\|x - x_0\|, \tag{4.3}$$

for all $x \in \mathbb{R}^n$ and $x_0 \in \mathbb{R}^n$, see Figure 4.1 for a geometrical interpretation.

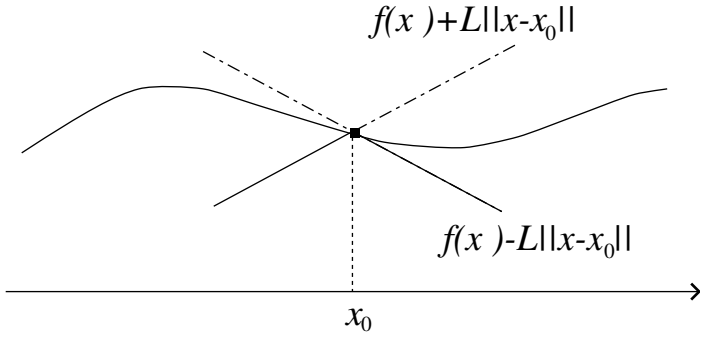

Figure 4.1: Geometrical interpretation of the Lipschitz conditions (4.2) and (4.3).

Methods for Lipschitz functions are suitable to find a global solution of the problem

$$\min_x f(x),$$

with

$$x \in I_n = \{x \in \mathbb{R}^n \mid A_i \leq x_i \leq B_i\},$$

and $A_i < B_i$ given, under the assumptions that the set I_n contains a global minimum of f, the function f is Lipschitz in I_n and the Lipschitz constant L of f in I_n is known. Under these assumptions it is possible to construct a very simple global minimization algorithm, known as Schubert-Mladineo algorithm, as follows.

Step 0. Given $x_0 \in I_n$ and $\tilde{L} > L$.

Step 1. Set $k = 0$.

Step 2. Let

$$F_k(x) = \max_{j=0,\cdots,k} \{f(x_j) - \tilde{L}\|x - x_j\|\}$$

and compute x_{k+1} such that

$$F_k(x_{k+1}) = \min_{x \in I_n} F_k(x).$$

Step 4. Set $k = k + 1$ and go to **Step 2**.

Remark. The functions F_k in **Step 2** of the algorithm have a very special form. This can be exploited to construct special algorithms solving the problem

$$\min_{x \in I_n} F_k(x)$$

in a finite number of iterations. ◇

For Schubert-Mladineo algorithm it is possible to prove the following statement.

Theorem 24 *Let $f^\star$ be the minimum value of f in I_n, let $x^\star$ be such that $f(x^\star) = f^\star$ and let $F_k^\star$ be the minima of the functions F_k in I_n. Let*

$$\Phi = \{x \in I_n \mid f(x) = f^\star\}$$

and let $\{x_k\}$ be the sequence generated by the algorithm. Then

- $\lim_{k \to \infty} f(x_k) = f^\star$;
- *the sequence $\{F_k^\star\}$ is non-decreasing and $\lim_{k \to \infty} F_k^\star = f^\star$;*
- $\lim_{k \to \infty} \inf_{x \in \Phi} \|x^\star - x_k\| = 0$;
- $f(x_k) \geq f^\star \geq F_{k-1}(x_k)$.

Schubert-Mladineo algorithm can be given, if $x \in I_1 \subset I\!R$, a simple geometrical interpretation, as shown in Figure 4.2.
The main advantage of Schubert-Mladineo algorithm is that it does not require the computation of derivatives, hence it is also applicable to functions which are not everywhere

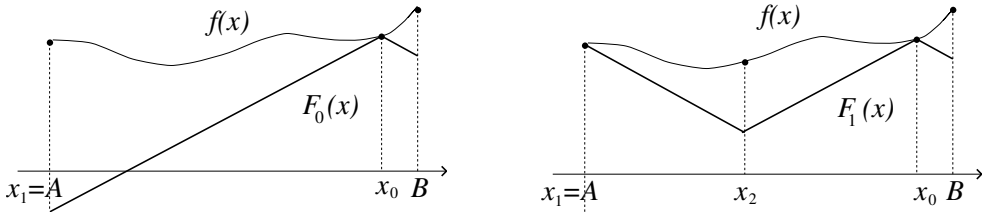

Figure 4.2: Geometrical interpretation of Schubert-Mladineo algorithm.

differentiable. Moreover, unlike other global minimization algorithms, it is possible to prove the convergence of the sequence $\{x_k\}$ to the global minimum. Finally, it is possible to define a simple *stopping* condition. For, note that if $\{x_k\}$ and $\{F_k^\star\}$ are the sequences generated by the algorithm, then

$$f(x_k) \geq f^\star \geq F_k^\star$$

and

$$f(x_k) \geq f^\star \geq f(x_k) + r_k,$$

where $r_k = F_k^\star - f(x_k)$ and $\lim_{k\to\infty} r_k = 0$. As a result, if $|r_k| < \epsilon$, for some $\epsilon > 0$, the point x_k gives a good approximation of the minimum of f.
The main disadvantage of the algorithm is in the assumption that the set I_n contains a global minimum of f in $\mathbb{R}^n$. Moreover, it may be difficult to compute the Lipschitz constant L.

4.2.2 Methods of the trajectories

The basic idea of the global optimization methods known as methods of the trajectories is to construct trajectories which go through all local minima. Once all local minima are determined, the global minimum can be easily isolated. These methods have been originally proposed in the 70's, but only recently, because of increased computer power and of a reformulation using tools from differential geometry, they have proved to be effective.
The simplest and first method of the trajectories is the so-called Branin method. Consider the function f and assume ∇f is continuous. Fix x_0 and consider the differential equations

$$\frac{d}{dt}\nabla f(x(t)) = \pm\nabla f(x(t)). \tag{4.4}$$

The solutions $x(t)$ of such differential equations are such that

$$\nabla f(x(t)) = \nabla f(x_0)e^{\pm t},$$

i.e. $\nabla f(x(t))$ is parallel to $\nabla f(x_0)$ for all t. Using these facts it is possible to describe Branin algorithm.

Step 0. Given x_0.

Step 1. Compute the solution $x(t)$ of the differential equation

$$\frac{d}{dt}\nabla f(x(t)) = -\nabla f(x(t))$$

with $x(0) = x_0$.

Step 2. The point $x^\star = \lim_{t\to\infty} x(t)$ is a stationary point of f, in fact $\lim_{t\to\infty} \nabla f(x(t)) = 0$.

Step 3. Consider a perturbation of the point $x^\star$, *i.e.* the point $\tilde{x} = x^\star + \epsilon$ and compute the solution $x(t)$ of the differential equation

$$\frac{d}{dt}\nabla f(x(t)) = \nabla f(x(t)).$$

Along this trajectory the gradient $\nabla f(x(t))$ increases, hence the trajectory escapes from the *region of attraction* of x_0.

Step 4. Fix $\bar{t} > 0$ and assume that $x(\bar{t})$ is sufficiently away from x_0. Set $x_0 = x(\bar{t})$ and got to **Step 1**.

Note that, if the perturbation ϵ and the time $\bar{t}$ are properly selected, at each iteration the algorithm generates a new stationary point of the function f.

Remark. If $\nabla^2 f$ is continuous then the differential equations (4.4) can be written as

$$\dot{x}(t) = \pm \left[\nabla^2 f(x(t))\right]^{-1} \nabla f(x(t)).$$

Therefore Branin method is a continuous equivalent of Newton method. Note however that, as $\nabla^2 f(x(t))$ may become singular, the above equation may be meaningless. In such a case it is possible to modify Branin method using ideas borrowed from quasi-Newton algorithms. ◇

Branin method is very simple to implement. However, it has several disadvantages.

- It is not possible to prove convergence to the global minimum.
- Even if the method yields the global minimum, it is not possible to know how many iterations are needed to reach such a global minimum, *i.e.* there is no stopping criterion.
- The trajectories $x(t)$ are attracted by all stationary points of f, *i.e.* both minima and maxima.
- There is not a systematic way to select ϵ and $\bar{t}$.

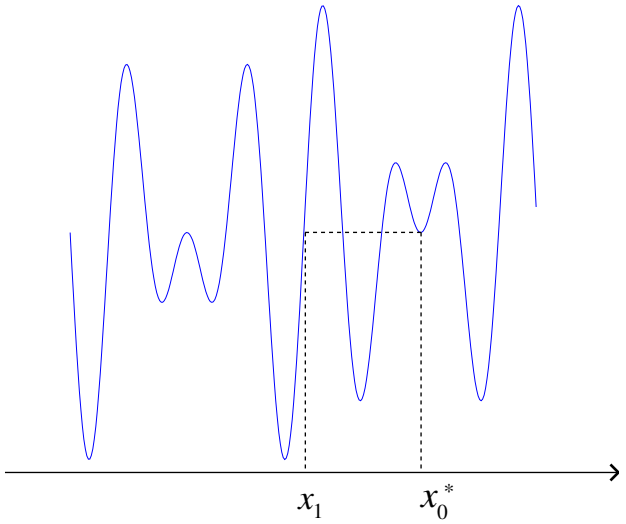

Figure 4.3: Interpretation of the tunneling phase.

4.2.3 Tunneling methods

Tunneling methods have been proposed to find, in an efficient way, the global minimum of a function with several (possibly thousands) of local minima.
Tunneling algorithms are composed of a sequence of cycles, each having two phases. The first phase is the minimization phase, *i.e.* a local minimum is computed. The second phase is the tunneling phase, *i.e.* a new starting point for the minimization phase is computed.

Minimization phase

Given a point x_0, a local minimization, using any unconstrained optimization algorithm, is performed. This minimization yields a local minimum $x_0^\star$.

Tunneling phase

A point $x_1 \neq x_0^\star$ such that

$$f(x_1) = f(x_0^\star)$$

is determined. See Figure 4.3 for a geometrical interpretation.

In theory, tunneling methods generate a sequence $\{x_k^\star\}$ such that

$$f(x_{k+1}^\star) \leq f(x_k^\star)$$

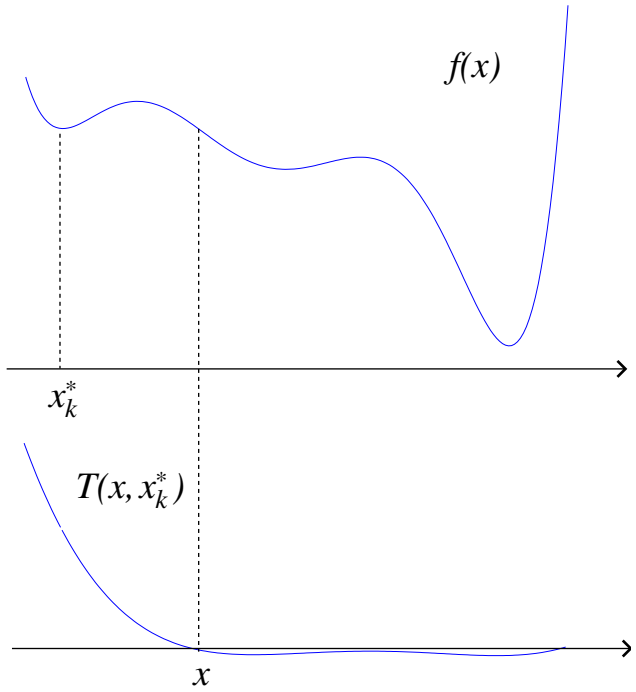

Figure 4.4: The functions $f(x)$ and $T(x, x_k^\star)$.

and the sequence $\{x_k^\star\}$ converges to the global minimum without *passing* through all local minima. This is the most important advantage of tunneling methods. The main disadvantage is the difficulty in performing the tunneling phase. In general, given a point $x_k^\star$ a point x such that $f(x) = f(x_k^\star)$ is constructed searching for a zero of the function (see Figure 4.4)

$$T(x, x_k^\star) = \frac{f(x) - f(x_k^\star)}{\|x - x_k^\star\|^{2\lambda}},$$

where the parameter $\lambda > 0$ has to be selected such that $T(x_k^\star, x_k^\star) > 0$.
Finally, it is worth noting that tunneling methods do not have a stopping criterion, *i.e.* the algorithm attempts to perform the tunneling phase even if the point $x_k^\star$ is a global minimum.

4.3 Probabilistic methods

4.3.1 Methods using random directions

In this class of algorithms at each iteration a randomly selected direction, having unity norm, is selected. The theoretical justification of such an algorithm rests on Gaviano theorem. This states that the sequence $\{x_k\}$ generated using the iteration

$$x_{k+1} = x_k + \alpha_k d_k,$$

where d_k is randomly selected on a unity norm sphere and α_k is such that

$$f(x_k + \alpha_k d_k) = \min_{\alpha} f(x_k + \alpha d_k),$$

is such that for any $\epsilon > 0$ the probability that

$$f(x_k) - f^\star < \epsilon,$$

where $f^\star$ is a global minimum of f, tends to one as $k \to \infty$.

4.3.2 Multistart methods

Multistart methods are based on the fact that for given sets D and A, with measures $m(D)$ and $m(A)$, and such that

$$1 \geq \frac{m(A)}{m(D)} = \alpha \geq 0,$$

the probability that, selecting N random points in D, one of these points is in A is

$$P(A, N) = 1 - (1 - \alpha)^N.$$

As a result

$$\lim_{N \to \infty} P(A, N) = 1.$$

Therefore, if A is a neighborhood of a global minimum of f in D, we conclude that, selecting a sufficiently large number of random points in D, one of these will (almost surely) be close to the global minimum. Using these considerations it is possible to construct a whole class of algorithms, with similar properties, as detailed hereafter.

Step 0. Set $f^\star = \infty$.

Step 1. Select a random point $x_0 \in I\!R^n$.

Step 2. If $f(x_0) > f^\star$ go to **Step 1**.

Step 3. Perform a local minimization starting from x_0 and yielding a point $x_0^\star$. Set $f^\star = f(x_0^\star)$.

Step 4. Check if $x_0^\star$ satisfies a stopping criterion. If not, go to **Step 1**.

4.3.3 Stopping criteria

The main disadvantage of probabilistic algorithms is the lack of a theoretically sound stopping criterion. The most promising and used stopping criterion is based on the construction of a *probabilistic approximation* $\tilde{P}(w)$ of the function

$$P(w) = \frac{m(\{x \in D \mid f(x) \leq w\})}{m(D)}.$$

Once the function $\tilde{P}(w)$ is known, a point $x^\star$ is regarded as a good approximation of the global minimum of f if

$$\tilde{P}(f(x^\star)) \leq \epsilon \ll 1.$$