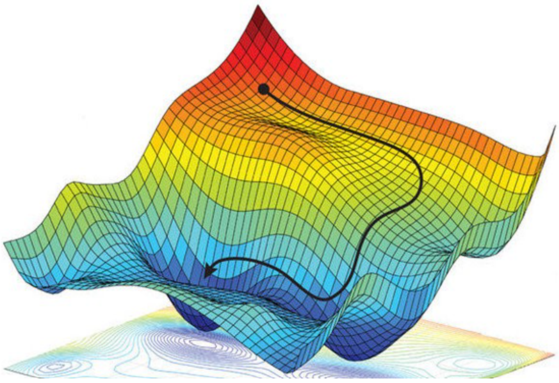
École Nationale Supérieure des Mines de Nancy

Optimization

Theory and numerics

2023-2024

Christophe ZHANG



Institut Élie Cartan de Lorraine (Inria SPHINX)
Université de Lorraine
email : christophe.zhang@inria.fr

Preamble

These lecture notes lay out theoretical and numerical fundamentals of optimization.

The first chapter contains reminders on differential calculus, and a summary of the terminology frequently used in optimization.

The second chapter covers some classical theoretical results on the existence and uniqueness of solutions, and on necessary (and sometimes sufficient) optimality conditions, both with and without constraints.

These notions will allow us to understand how optimization algorithms can be designed and studied, which will be the aim of chapters 3 (unconstrained optimization) and 4 (constrained optimization).

These lecture notes are purposedly lapidary in places. The motivated reader can find some of the missing proofs in further resources, or in class.

Contents

I.

Introduction

I.1 The vocabulary of optimization

Let V be a normed vector space equipped with the norm $\|\cdot\|$. In this course, we are interested in the following problem:

$$\inf_{x\in K} f(x) \tag{I.1}$$

where $K \subset V$, and $f : K \longrightarrow \mathbb{R}$ is a function, called the cost function or criterion.

- If $K = V$, we say that (I.1) is an unconstrained optimization problem.
- If $K \subsetneq V$, we say that (I.1) is a constrained optimization problem.
- If $\dim K < +\infty$ (resp. $\dim K = +\infty$), we say that (I.1) is a finite-dimensional optimization problem (resp. infinite-dimensional optimization problem).

Note that this formalism encompasses all optimization problems, including maximization problems since maximizing a quantity is equivalent to minimizing its negation.

In the context of this course, we will mainly study finite-dimensional optimization. We will adopt the following convention: if we want to indicate that the minimum value is achieved, we will write

$$\min_{x\in K} f(x)$$

while we will use the notation "inf" when we do not know *a priori* whether the lower bound is reached.

We will say that

- x^* is a global minimizer of problem (I.1) if x^* is a solution to this problem.
- x^* is a local minimizer of problem (I.1) if there exists a neighborhood $\mathscr{V}$ of x^* such that x^* is a solution to the problem

$$\inf_{x\in\mathscr{V}} f(x).$$

Finally, recall that any non-empty bounded subset of $\mathbb{R}$ has a lower bound, characterized as follows:

Proposition I.1.1. Minimizing Sequences

Let X be a non-empty bounded subset of $\mathbb{R}$.
Then, the following statements are equivalent:

(i) $m = \inf\{x, x \in X\}$;

(ii) $\forall \varepsilon > 0,\ \exists x \in X \mid m \leq x < m + \varepsilon$;

(iii) m is a lower bound for X, and there exists a sequence $(x_n)_{n \in \mathbb{N}} \in X^{\mathbb{N}}$, called a "minimizing sequence," converging to m.

As a result, here are the questions that will naturally arise when you encounter an optimization problem:

- Does this problem have a solution?
- *First scenario.*
  If this problem has a solution, we will seek to characterize it (for example, is it unique?) or, even better, to determine it when possible. For this purpose, we will exploit the necessary optimality conditions (first and second order).
- *Second scenario.*
  If this problem does not have a solution, we will try to exhibit a minimizing sequence, i.e., a sequence $(x_k)_k$ from the set K such that $(f(x_k))_k$ converges to $\inf\{f(x), x \in K\}$.
- Finally, when we cannot explicitly determine the solutions of the optimization problem, we will ask ourselves about the choice of suitable numerical methods to find the minimum and its minimizers.

Some Examples of Optimization Problems

Problem 1. (Finite Dimension) : Find the maximum volume of a rectangular parallelepiped among those with an external surface area of 6
By introducing a, b, and c, the lengths of the sides of the parallelepiped, we reduce the problem to finding the solution of the problem

$$\begin{cases} \max & abc \\ s.t. & 2(ab + bc + ca) = 6. \end{cases}$$

This problem is a finite-dimensional optimization problem.

Problem 2. (Infinite Dimension) : Find the function $u_0 \in L^2(\Omega)$ such that

$$\begin{cases} \inf & \|\nabla u\|^2_{L^2(\Omega)} \\ \text{s.t.} & -\Delta u = f \ \text{ in } \Omega, \\ & u = g \ \text{ on } \partial\Omega. \end{cases}$$

This problem is infinite-dimensional, and the space of constraints is non-empty, as we have $\|g\|^2_{H^1(\partial\Omega)} \leq \|u_0\|^2_{L^2(\Omega)}$.

Problem 3. (No Solution) : Consider the optimization problem

$$\begin{cases} \min & x^2 \\ \text{s.t.} & x > 0, \\ & x^2 + 1 = 0. \end{cases}$$

Problem 3 has no solution since $x^2 + 1 > 0$ for all $x \in \mathbb{R}$.

Problem 4. (Minimizing Sequence) : Consider the problem

$$\begin{cases} \min & x^2 \\ \text{s.t.} & x \in [0,1). \end{cases}$$

The set $K = [0,1)$ is not compact, and the infimum of the problem is 0. The sequence $(x_n)_n = (1/n)_n$ satisfies $\lim_{n\to+\infty} x_n = 0$ and $\lim_{n\to+\infty} f(x_n) = 0$. This sequence is a minimizing sequence for problem 4.

I.2 Some Reminders of Differential Calculus

Let us start with the concept of **differentiability**.

> **Definition I.2.1. Differentiability**
>
> Let E and F be two real normed vector spaces. Let U be an open set in E, and $x_0 \in U$. We say that a function $f : U \longrightarrow F$ is **differentiable at** x_0 or has a first-order Taylor expansion at x_0 if there exists $df_{x_0} \in \mathscr{L}(E,F)$ (continuous), such that
>
> $$f(x_0 + h) - f(x_0) = df_{x_0}(h) + o(\|h\|_E).$$

Some immediate remarks:

- In infinite dimensions, the differentiability of a function depends on the norms used for the spaces E and F. This is not the case in finite dimensions since all norms are equivalent.

- By definition, the map df_{x_0} is continuous. However, this may not be the case for the map $df : \begin{array}{ccc} U & \longrightarrow & L(E,F) \\ x_0 & \longmapsto & df_{x_0} \end{array}$. If it is the case, we say that f is of class $\mathscr{C}^1$ in the neighborhood of x_0.

- *How to compute a differential in practice?*

  If we have previously demonstrated that f is differentiable at x_0, then for all $h \in E$, we can write

  $$df_{x_0}(h) = \lim_{\substack{\varepsilon\to 0 \\ \varepsilon\in\mathbb{R}}} \frac{f(x_0 + \varepsilon h) - f(x_0)}{\varepsilon}.$$

  The advantage of such a notation is that we have reduced it to calculating the limit of a function of a real variable. The previous limit is called the *directional derivative of f at x_0 in the direction of h*, or *Gâteaux differential of f at x_0 in the direction h*. Note that if f is differentiable, it is easy to show that f has a directional derivative along any vector h, but the converse is not necessarily true.

Let us summarize in the form of a diagram the implications between these different properties.

$$\begin{tikzcd}
f \text{ is } \mathscr{C}^1 \text{ at } x_0 \arrow[r, Rightarrow] & f \text{ is differentiable at } x_0 \arrow[r, Rightarrow] \arrow[d, Rightarrow] & f \text{ is } \mathscr{C}^0 \text{ at } x_0 \\
& \begin{array}{c} f \text{ is differentiable along} \\ \text{every vector } h \text{ at } x_0 \end{array} &
\end{tikzcd}$$

The unwritten implications are not necessarily true, meaning that counterexamples can be found.

Remark I.2.2 Higher Order Differentiability

Let V be a Hilbert space and $f : V \longrightarrow \mathbb{R}$. If f is assumed to be differentiable, starting from the expansion

$$f(x_0 + h) - f(x_0) = df_{x_0}(h) + o(\|h\|_V),$$

by using the Riesz representation theorem, we can identify $df_{x_0}(h)$ as $\langle \nabla f(x_0), h\rangle$, where $\nabla f(x_0) \in V$. This is how we generalize the concept of gradient, which we will detail in the next section. Stating that f is twice differentiable means that there exists a linear map $L(x_0) : V \longrightarrow V'$ such that

$$df_{x_0+\xi} = df_{x_0} + L(x_0)\xi + o(\|\xi\|_V) \in V',$$

where V' denotes the *topological dual of* V, which is the set of continuous linear functionals on V. Recall that when V is a Hilbert space, the map

$$\begin{array}{rccc} f : & V & \longrightarrow & V' \\ & y & \longmapsto & f_y \end{array} \qquad \text{where for all } x \in V,\ f_y(x) = \langle x, y\rangle_V$$

is an isometry, which allows us to identify V' with V.

The second derivative of f, denoted as $d^2 f_{x_0}$, is then the map $L(x_0) : V \longrightarrow V'$. It is challenging to evaluate it in practice because $L(x_0)\xi$ is an element of V'. Furthermore, if the application $x_0 \mapsto d^2 f_{x_0}$ is continuous at x_0, we say that f is of class $\mathscr{C}^2$ at x_0.

In the case of finite dimension ($V = V' = \mathbb{R}^n$), these formulas take on a particularly friendly form, as the second derivative is identified as the Hessian matrix when f is twice differentiable (see the next section for more details).

I.3 Digression to Finite Dimension

We will complete the concepts we have just discussed in this particular case. In what follows, we will denote $(e_1, \cdots, e_n)$ as the canonical basis of $\mathbb{R}^n$, and we equip $\mathbb{R}^n$ with its usual Euclidean structure.

Definition I.3.1. Functions of class $\mathscr{C}^k$

Let $i \in \{1, \cdots n\}$ and $k \geq 2$. We say that a function $f : U \subset \mathbb{R}^n \longrightarrow \mathbb{R}$

(i) has a partial derivative of index i at x_0 if it is differentiable at x_0 in the direction of e_i;

(ii) is of class $\mathscr{C}^k$ if all its partial derivatives up to order k exist and are continuous on U.

We will now focus on the specific case of a function $f : U \subset \mathbb{R}^n \longrightarrow \mathbb{R}$, with U an open subset of $\mathbb{R}^n$. Let $x_0 \in K$.

- Suppose that f is differentiable at x_0. Then, for all $h \in \mathbb{R}^n$,

  $$\boxed{f(x_0 + h) - f(x_0) = \langle \nabla f(x_0), h \rangle + \underset{h \to 0}{o}(\|h\|)}$$

  where $\nabla f(x_0)$ is the gradient of f at x_0, i.e., the vector $(\frac{\partial f}{\partial x_1}(x_0), \cdots, \frac{\partial f}{\partial x_n}(x_0))$. The notion of gradient is not intrinsic; it depends on the chosen inner product. The general definition of $\nabla f(x)$ results from Riesz's representation theorem applied to the differential of f at x. However, in finite dimensions, the standard inner product is usually chosen, and the formulas above define the gradient and the Hessian equally well.

- Suppose that f is twice differentiable at x_0. Then, for all $h \in \mathbb{R}^n$,

  $$\boxed{f(x_0 + h) - f(x_0) = \langle \nabla f(x_0), h \rangle + \frac{1}{2} \langle \mathrm{Hess} f(x_0) h, h \rangle + \underset{h \to 0}{o}(\|h\|^2)}$$

  where $\mathrm{Hess} f(x_0)$ is the $n \times n$ matrix of second derivatives of f evaluated at x_0, i.e.,

  $$\mathrm{Hess} f(x_0) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x_0) \right)_{1 \leq i,j \leq n}$$

  Note that if f is twice differentiable, due to the Schwarz theorem, $\mathrm{Hess} f(x_0)$ is a real symmetric matrix (keep in mind also the Peano counterexample when the function is not twice differentiable).

I.4 Formulae

As a reminder, let us go over the different Taylor formulas and the minimal regularity assumptions they require. We will write them here up to the second order only, as it follows the logic of this course, but these formulas can be extended to all orders with suitable hypotheses.

$\hookrightarrow$ **Taylor formula with integral remainder.**

Suppose that f is $\mathscr{C}^2$ in an open set U of $\mathbb{R}^n$ into $\mathbb{R}$. If the line segment $[a, a + h]$ is contained in U, then

$$f(x_0 + h) - f(x_0) = \langle \nabla f(x_0), h \rangle + \frac{1}{2} \int_0^1 \frac{(1-t)^2}{2} \langle \mathrm{Hess} f(x_0 + th) h, h \rangle dt$$

↪ Taylor formula with Lagrange remainder.

Suppose that f is twice differentiable in an open set U of $\mathbb{R}^n$ into $\mathbb{R}$. If the line segment $[a,a+h]$ is contained in U and suppose that there exists a constant $C > 0$ such that

$$\forall t \in [0,1], \ |\langle \mathrm{Hess} f(x_0 + th)h,h\rangle| \leq C\|h\|^2.$$

Then,

$$|f(x_0 + h) - f(x_0) - \langle \nabla f(x_0),h\rangle| \leq \frac{C}{2}\|h\|^2.$$

↪ Taylor Mac-Laurin formula.

Let $f : [\alpha,\beta] \longrightarrow \mathbb{R}$ be a function that is $N+1$ times differentiable. Then, there exists $\gamma \in]\alpha,\beta[$ such that

$$f(\beta) = f(\alpha) + \sum_{k=1}^{N} \frac{(\beta-\alpha)^k}{k!} f^{(k)}(\alpha) + \frac{(\beta-\alpha)^{N+1}}{(N+1)!} f^{(N+1)}(\gamma).$$

The Taylor Mac-Laurin formula is a generalization of the mean value theorem.

↪ Product of functions

Let $x \in \mathbb{R}^n$, and let $f,g : \mathbb{R}^n \to \mathbb{R}$ be two functions that are differentiable in x. Then fg is differentiable in x and

$$D(fg)(x) = g(x)Df(x) + f(x)Dg(x).$$

↪ Ratio of functions

Let $x \in \mathbb{R}^n$, and $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable in x, and such that $f(x) \neq 0$. Then, $1/f$ is differentiable in x and:

$$D\left(\frac{1}{f}\right)(x) = -\frac{1}{f(x)^2} Df(x).$$

↪ Chain rule

Let $x \in \mathbb{R}^n$. Let $f : \mathbb{R}^m \to \mathbb{R}^p$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ be two functions such that g is differentiable in x, and f is differentiable in $g(x)$. Then, $f \circ g : \mathbb{R}^n \to \mathbb{R}^p$ is differentiable in x and:

$$D(f \circ g)(x) = Df(g(x)) \circ Dg(x).$$

↪ Inverse of a function

Let $x \in \mathbb{R}^n$, and $f : \mathbb{R}^n \to \mathbb{R}^n$ such that $Df(s)$ is invertible for s in a neighborhood of x. Then, f is locally invertible around x, and its local inverse f^{-1} is differentiable in $f(x)$.

$$D(f^{-1})(f(x)) = Df(x)^{-1}.$$

II. Generalities and Theoretical Study of Optimization Problems

II.1 Introduction

In this course, we are interested in problems of the following type: “finding the minimum and the minimizer of a function with or without constraint(s).” From a mathematical perspective, the problem is formulated as follows:

- **Unconstrained problem**
$$\inf_{x\in\mathbb{R}^n} J(x),$$
- **Constrained problem(s)**
$$\inf_{x\in C} J(x),$$
where $C \subsetneq \mathbb{R}^n$.

The problem can also be posed in infinite dimension. In this case, it is about minimizing a functional J over a vector space of *infinite* dimension (for example, a Hilbert space, but not exclusively). However, in this elementary course, we will focus on optimization problems in *finite* dimensions. The main reason for this is that in practice, we discretize the continuous problem, which involves projecting it onto a finite-dimensional space.

In this chapter, we will provide answers to the following questions:

- how to demonstrate the existence and uniqueness of solutions to an optimization problem?
- how to determine minimizers/maximizers using local optimality conditions?

We will mainly consider two types of constraints:

$$\begin{array}{rcll} C &=& \{x\in\mathbb{R}^n \text{ such that: } \varphi_i(x)\le 0, \forall i\in I\} & \text{(inequality constraints)},\\ C &=& \{x\in\mathbb{R}^n \text{ such that: } \varphi_i(x)= 0, \forall i\in I\} & \text{(equality constraints)}, \end{array}$$

where the functions φ_i are continuous (at least) from $\mathbb{R}^n$ to $\mathbb{R}$.

Sometimes, you may encounter a classification of optimization problems, which allows you to choose a suitable resolution algorithm. For example, one talks about:

- linear programming when J is linear, and C is a (convex) polyhedron defined by
$$C = \{x\in\mathbb{R}^n, Bx \le b\},$$
where B is an $m\times n$ matrix, and b is an element of $\mathbb{R}$,

- quadratic programming when J is of the form

$$J(x) = \frac{1}{2}\langle Ax, x\rangle_{\mathbb{R}^n} + \langle b, x\rangle_{\mathbb{R}^n} + c,$$

  where A is a symmetric matrix, b an element of $\mathbb{R}$, and C is still generally a convex polyhedron,

- convex programming when the functional J and the set C are convex (with C still being polyhedral),

- differentiable optimization when J and the functions φ_i are once or twice differentiable.

- nondifferentiable optimization when J or the functions φ_i are not differentiable (e.g., the minimization of $x \mapsto |x|$ over $\mathbb{R}$).

II.2 Existence and Uniqueness Results

II.2.1 Existence

Most existence theorems are variations of the following classical theorem.

Theorem II.2.1. Extreme value theorem

Let f be a continuous function on a compact set $K \subset \mathbb{R}^n$. Then, it reaches its minimum and maximum values on K.

More generally, we can state the following:

Theorem II.2.2. Existence

Let J be a continuous function on a closed subset C of $\mathbb{R}^n$. We assume that

- either C is bounded,
- or C is unbounded, and $\lim\limits_{\|x\|\to+\infty} J(x) = +\infty$ (we then say that J is coercive),

then J has a minimum on C.

There are two ingredients in the proofs of the usual existence results:

(i) **Compactness** ensures that minimizing sequences converge to a point in C. This limit is the potential minimizer of the function.

(ii) **Continuity** ensures that the values of f converge as well, so that the minimum can be reached.

II.2.2 Some remarks on the case of Hilbert spaces

In infinite dimension, compactness is typically more difficult to come by. Indeed, closed bounded sets are usually not compact, unlike in finite-dimensional spaces. We will provide some insights on how this can be dealt with in Hilbert spaces. The motivated reader will find proofs and more detailed results in any functional analysis course, or advanced optimization course.

Focusing on Hilbert spaces, we introduce a weaker topology to circumvent the aforementioned difficulty, by defining the following:

> **Definition II.2.3. Weak convergence**
>
> Let H be a Hilbert space. We say that a sequence $(x_n) \in H^{\mathbb{N}}$ converges weakly to $x \in H$ if
>
> $$\langle x_n, y \rangle \xrightarrow[n\to\infty]{} \langle x, y \rangle, \quad \forall y \in H,$$
>
> and we denote this **weak convergence** by

$$x_n \underset{n\to\infty}{\rightharpoonup} x.$$

This convergence corresponds to the so-called **weak topology**, named in contrast to the usual topology defined by the norm, which we will now call the **strong topology**, with its associated **strong convergence**. There are several ways to understand this name:

- *Strong convergence clearly implies weak convergence*, for any sequence.
- Topologically speaking, the **weak topology** has *fewer open sets* (recall that a topology is defined by its open sets).
- If there are fewer open sets, there are also *fewer closed sets*. Moreover, *if a set is weakly closed, then it is strongly closed.* The implication here is inverted and may seem counterintuitive, but one should keep in mind that in the weak topology there are more converging sequences than in the strong topology. Therefore, *being weakly closed is a stronger requirement than being strongly closed.*

Weakening topologies has a considerable advantage: indeed, if there are fewer open sets, this means that there are **more compact sets**. Thus, one has a better chance of proving existence results, as these results rely heavily on compactness.

All of this depends nonetheless on the continuity of the function f one seeks to minimize. Since we want to work in the weak topology in order to ensure compactness, f has to be **continuous for the weak topology**. This calls for caution: indeed weak convergence is less demanding, but this means that there are more converging sequences, with which f must behave accordingly. Thus *weak continuity is a stronger requirement than strong continuity* (again, beware of the unintuitive order).

Finally, we state a crucial property, which illustrates the importance of convex functions in optimization:

Theorem II.2.4. Continuity of convex functions

Let H be a Hilbert space. A convex function on $H \to \mathbb{R}$ is strongly continuous if and only if it is weakly continuous.

This result relies on this fundamental lemma:

Lemma II.2.5. Banach-Alaoglu-Bourbaki

Let H be a Hilbert space, and $C \subset H$ be a convex set. Then, C is strongly closed if and only if it is weakly closed.

Combining all of the above, we get a fundamental existence result in Hilbert spaces:

Theorem II.2.6. Optimization in a Hilbert space

Let H be a Hilbert space, and $J : H \to \mathbb{R}$ a convex, coercive, and strongly continuous functional. Then, J is bounded from below on H, and there exists $x^* \in C$ such that $J(x^*) = \inf_{x \in C} J(x)$.

Finally, let us mention a weaker notion of continuity, for which all of the above existence results actually hold as well:

Definition II.2.7. lower semi-continuity

Let H be a Hilbert space. We say that a function $f : H \to \mathbb{R}$ is lower semi-continuous for a given topology (*e.g.* the weak or the strong topology) if for all $\lambda \in \mathbb{R}$, the set

$$\{x \in H, \quad f(x) \leq \lambda\}$$

is closed.

Of course, continuity implies lower semi-continuity. But semi continuity allows for discontinuities: consider for example the Heaviside function.

II.2.3 Convex sets and functions

Beyond the above considerations on infinite-dimensional problems, convexity plays an extremely important role in finite-dimensional optimization since it generally allows us to study the uniqueness of solutions to an optimization problem.

We will also further on that convexity allows us to obtain necessary and *sufficient* optimality conditions.

Definition II.2.8. Convex Set/Function

(i) A set C is called convex if, for all points x and y in C, the line segment $[x;y]$ is included in C, i.e., for any $t \in [0;1]$, the point $tx + (1-t)y$ belongs to C.

(ii) A function J defined on a convex set C is called convex if

$$\forall (x,y) \in C^2, \quad \forall t \in [0;1], \quad J(tx + (1-t)y) \leq tJ(x) + (1-t)J(y).$$

The function is called strictly convex if

$$\forall (x,y) \in C^2, \quad x \neq y, \quad \forall t \in]0;1[, \quad J(tx + (1-t)y) < tJ(x) + (1-t)J(y).$$

When a convex function is differentiable, the following characterization is useful.

Proposition II.2.9.

Let J be a differentiable function defined on a convex set C of $\mathbb{R}^n$ and taking real values. The following conditions are equivalent to the convexity of J

$$\begin{array}{c} (i)\ \forall (x,y) \in C^2,\ \langle \nabla J(x), y - x \rangle_{\mathbb{R}^n} \leq J(y) - J(x) \\ (ii)\ \forall (x,y) \in C^2,\ \langle \nabla J(y) - \nabla J(x), y - x \rangle_{\mathbb{R}^n} \geq 0. \end{array}.$$

Remark II.2.10

A similar characterization holds for strictly convex functions: one simply replaces convex by strictly convex and the inequality by a strict inequality in Proposition II.2.9.

Proposition II.2.11.

Let J be a twice differentiable function defined on a convex set C of $\mathbb{R}^n$ and taking real values. For any $x \in C$, let $D^2J(x)$ denote the Hessian matrix, defined by the coefficients $\dfrac{\partial^2 J}{\partial_{x_i} \partial_{x_j}}(x)$.

(i) J is convex if and only if the symmetric matrix $D^2J(x)$ is positive for all $x \in C$.

(ii) If $D^2J(x)$ is positive definite for all $x \in C$, then J is strictly convex.

Remark II.2.12

It should be noted that the second point is a sufficient condition, but not necessary. The function $x \mapsto x^4$ provides a simple counterexample in 1-D.

The following result illustrates the importance of convexity in optimization problems.

Proposition II.2.13. Uniqueness

Let J be a convex function defined on a convex set C of $\mathbb{R}^n$. Then,

- any local minimizer of J on C is a global minimizer,
- if J is strictly convex, there is at most one global minimizer.

II.3 Optimality Conditions for unconstrained problems

Throughout this section, we assume that $J : \mathbb{R}^N \to \mathbb{R}$ is a function that is once or twice differentiable. We denote x^* as a (local) minimizer of J.

The following remains valid in the case where $J : C \subsetneq \mathbb{R}^N \to \mathbb{R}$, when the minimizer x^* is in the interior of the set of constraints C.

II.3.1 Necessary conditions

Theorem II.3.1. Necessary Optimality Conditions

Let x^* be a local minimizer of the problem

$$\inf_{x\in\mathbb{R}^n} J(x).$$

Then, x^* must satisfy the following:

- First-order condition: if J is differentiable at x^*, then $\nabla J(x^*) = 0$. One says that x^* is a critical point.
- Second-order condition: if J is twice differentiable at x^*, then the quadratic form $D^2J(x^*)$ is semi-definite positive, i.e.,

$$\forall y \in \mathbb{R}^n, \quad \left\langle D^2J(x^*)y, y\right\rangle_{\mathbb{R}^n} \geq 0.$$

Keep in mind that the conditions above are merely necessary: they are useful to narrow down the search of local minimizers. However, not all critical points are local minimizers, and we will see below a more detailed classification of critical points of a differentiable function.

There are however some general (and quite useful) sufficient conditions for a point x^* to be a local minimizer, assuming that J is sufficiently smooth.

II.3.2 Sufficient conditions

Theorem II.3.2. Sufficient Optimality Conditions

Let J be a $\mathscr{C}^1$ function defined on $\mathbb{R}^n$. We assume that: $\nabla J(x^*) = \mathbf{0}$ and that J is twice differentiable at x^*. Then, if one of the following conditions is met, x^* is a local minimizer of J:

(i) $D^2J(x^*)$ is positive definite,

(ii) $\exists r > 0$ such that J is twice differentiable on $B(x^*,r)$ and, for all $x \in B(x^*,r)$, the quadratic form $D^2J(x)$ is semi-definite positive.

In the case where J is convex, the sufficient condition is expressed much more straightforwardly. Indeed, due to Proposition II.2.9, we have the following:

Proposition II.3.3. Sufficient Condition, Convex Case

Let J be a convex function of class $\mathscr{C}^1$, defined on $\mathbb{R}^n$, and x^* be a point in $\mathbb{R}^n$. Then, x^* is a (global) minimizer of J if and only if $\nabla J(x^*) = \mathbf{0}$.

II.3.3 An important example: quadratic functions

In $\mathbb{R}^n$, quadratic functions are polynomial functions which can be written in the form

$$f(X) = \frac{1}{2}\langle X, AX\rangle - \langle b, X\rangle - c, \quad \forall X \in \mathbb{R}^n,$$

where $c \in \mathbb{R}$, $b \in \mathbb{R}^n$, and $A \in S_n(\mathbb{R})$.

Many of their properties hinge on the matrix A. Indeed, one has

$$\nabla f(X) = AX - b, \quad \forall X \in \mathbb{R}^n,$$

and

$$\operatorname{Hess} f(X) = A, \quad \forall X \in \mathbb{R}^n.$$

Thus, using Proposition II.2.11, we get that f **is convex if and only if** A **is positive semi-definite, and strictly convex if and only if** A **is positive definite.**

Moreover, using the Cauchy Schwarz inequality and properties of positive definite matrices, one easily proves that f **is coercive if and only if** A **is positive definite.**

Finally, recall that A is diagonalizable thanks to the spectral theorem for real symmetric matrices. We denote $(e_i, \lambda_i)_{1\leq i\leq n}$ its eigenpairs, with

$$\lambda_1 \leq \cdots \leq \lambda_n$$

- If $\lambda_1 > 0$ then A is positive definite, and one can use Theorem II.2.2, as f is coercive and continuous on $\mathbb{R}^n$, so it reaches its minimum. As it is moreover strictly convex, its minimizer is unique.

  We now simply compute the critical point of f, which by Theorem II.3.3 will be the unique minimizer of f:
  $$\nabla f(X) = 0 \iff AX = b.$$
  As A is positive definite, it is invertible, so the above equation has a unique solution $X^* = A^{-1}b$. It is therefore the unique minimizer of f and we have
  $$\min_{\mathbb{R}^n} f = -\frac{1}{2}\langle b, A^{-1}b\rangle - c.$$

- If $\lambda_1 < 0$, then, for $r > 0$,
  $$f(re_1) = \frac{1}{2}r^2\lambda_1 - r\langle b, e_1\rangle - c \xrightarrow[r\to\infty]{} -\infty.$$
  Hence, f has no minimizer and $\inf f = -\infty$.

- Finally, if $\lambda_1 = 0$, then A is positive semi-definite. Thus f is convex and we know that it suffices to find its critical points, which are all global minimizers of f.

  From the computations above we know that the critical points of f are given by the solutions to the linear equation
  $$AX = b.$$

  - If $b \in \operatorname{Im} A$, then this equation has infinitely many solutions, given that A is not injective. By Theorem II.3.3, all these critical points are minimizers of f. Let X_0 be a solution to the above equation (*i.e.,* a critical point). Then
    $$f(X_0) = -\frac{1}{2}\langle b, X_0\rangle - c.$$
    Note that any other minimizer X^* of f satisfies
    $$X^* - X_0 \in \operatorname{Ker} A = \left(\operatorname{Im}(A^\top)\right)^\perp = (\operatorname{Im}(A))^\perp. \tag{II.1}$$
    Hence
    $$\langle b, X^* - X_0\rangle = 0.$$
  - Otherwise, f has no critical points. Given that it is convex, we deduce that it has no mimizers. Moreover, taking $e_1 \in \operatorname{Ker} A$, we know from (II.1) that $\langle b,e_1\rangle \neq 0$. We then have
    $$f(-r\operatorname{sgn}(\langle b, e_1\rangle)e_1) = -r|\langle b, e_1\rangle| - c \xrightarrow[r\to\infty]{} -\infty,$$

hence $\inf f = -\infty$.

Example II.3.4 Linear Regression

Consider a cloud of m points in $\mathbb{R}^2$: $M_i = (t_i,x_i)$, for $i \in \{1,\cdots,m\}$. These data are often the result of measurements, and we seek to describe the overall behavior of this cloud. In general, these points are not aligned, but if there are good reasons to believe they should be (a physical, biological model, etc., may give some intuition), one may wonder what is the line that best approximates these points.

The method of least squares consists in searching for the line such that the sum of the squares of the distances from the points in the cloud to this line is minimal.

In other words, we seek to solve

$$\inf_{(\alpha,\beta)\in\mathbb{R}^2} f(\alpha,\beta) \qquad \text{where} \qquad f(\alpha,\beta) = \sum_{i=1}^{n}(x_i - \alpha t_i - \beta)^2.$$

Let $X = (\alpha,\beta)^\top$. Then, we can write that

$$f(\alpha,\beta) = \|AX - b\|^2_{\mathbb{R}^n}, \text{ with } A = \begin{pmatrix} t_1 & 1 \\ \vdots & \vdots \\ t_m & 1 \end{pmatrix}, \ b = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$

Interpreting this problem as a projection problem, we can easily show the existence of solutions. Furthermore, the optimality conditions are written as

$$\begin{cases} S_{t^2}\alpha + S_t\beta = S_{xt} \\ S_t\alpha + m\beta = S_x \end{cases}$$

where we have set $S_t = \sum_{i=1}^m t_i$, $S_x = \sum_{i=1}^m x_i$, $S_{xt} = \sum_{i=1}^m x_i t_i$, and $S_{t^2} = \sum_{i=1}^m t_i^2$. Assuming that we are not in the situation where "$t_1 = \cdots = t_m$" (which can be found by calculating the determinant of the system and finding a case of Cauchy-Schwarz inequality), this system has the following solution

$$\alpha = \frac{S_x S_t - m S_{xt}}{(S_t)^2 - m S_{t^2}} \text{ and } \beta = \frac{S_{xt} S_t - S_x S_{t^2}}{(S_t)^2 - m S_{t^2}}.$$

II.3.4 Classification of critical points

As we have mentioned earlier, critical points play an important role in optimization. For convex functions, being a critical point is equivalent to being a global minimizer.

On the other hand, in the general case of a differentiable function, while a local minimizer is always a critical point, the converse is not true.

Indeed, taking Theorem II.3.1 with f, one sees that the local minimizers of $-f$, that is, the local maximizers of f, are also critical points.

Thus critical points can be local minimizers, but also local maximizers. Moreover, there are critical points which are neither, which are called **saddle points**.

Let us give a partial classification of critical points of twice differentiable functions, using the Hessian matrix.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable function, and $x^* \in \mathbb{R}^n$ be a critical point of f.

(i) If $\operatorname{Hess} f(x^*)$ is positive definite, we know from Theorem II.3.2 that x^* is a local maximizer.

(ii) Similarly, if $\operatorname{Hess} f(x^*)$ is negative definite (that is, is $-\operatorname{Hess} f(x^*)$ is positive definite), then x^* is a local maximizer.

(iii) If $\operatorname{Hess} f(x^*)$ does not have a sign (*i.e.*, is neither positive semi-definite nor negative semi-definite), from Theorem II.3.1 we know that x^* is neither a local maximizer nor minimizer. It is thus a saddle point.

(iv) Finally, if $\operatorname{Hess} f(x^*)$ is positive semi-definite, then x^* is either a local minimizer, or a saddle point.

From Theorem II.3.2, if one can prove that $\operatorname{Hess} f(x)$ stays positive semi-definite for x in a neighborhood of x^*, then x^* is a local minimizer of f.

Otherwise, one can push the analysis of $\operatorname{Hess} f(x^*)$ further to determine whether x^* is indeed a local minimizer.

Recall that $\operatorname{Hess} f(x^*)$ has positive eigenvalues, and a nontrivial kernel, which contains the directions which could prevent x^* from being a local minimizer. One way to find out if x^* is a saddle point is to study the behavior of

$$s \mapsto f(sd)$$

around 0, where $d \in \ker \operatorname{Hess} f(x^*)$. If one can prove that the above function has a local maximum at $s = 0$ then x^* is not a local minimizer and is thus a saddle point.

(v) If $\operatorname{Hess} f(x^*)$ is negative semi-definite, the above applies, replacing f by $-f$.

Going further on saddle points, we can distinguish different types of saddle points, according to the eigenvalues of $\operatorname{Hess} f(x^*)$.

The easiest case is when $\operatorname{Hess} f(x^*)$ has only nonzero, both positive and negative, eigenvalues. One can easily see, using appropriate Taylor expansions on the eigenspace corresponding to positive eigenvalues of $\operatorname{Hess} f(x^*)$, that x^* is a local minimizer of f in these directions. Similarly, x^* is a local maximizer of f on the eigenspace corresponding to negative eigenvalues. It is thus a **min-max**, as illustrated by the typical example in Figure II.1.

Other subtler situations can occur with saddle points. These occur when $\operatorname{Hess} f(x^*)$ has a nontrivial kernel. In such a direction d, the function

$$\varphi_d : s \mapsto f(sd)$$

satisfies

$$\phi'(0) = \phi''(0) = 0.$$

The point 0 could thus be a local minimizer or maximizer of φ, or an *inflection* point. In the latter case, x^* is not a local minimizer of f as it does not minimize f along d. However, it is also not a min-max. We illustrate this special case of saddle point in Figure II.2.

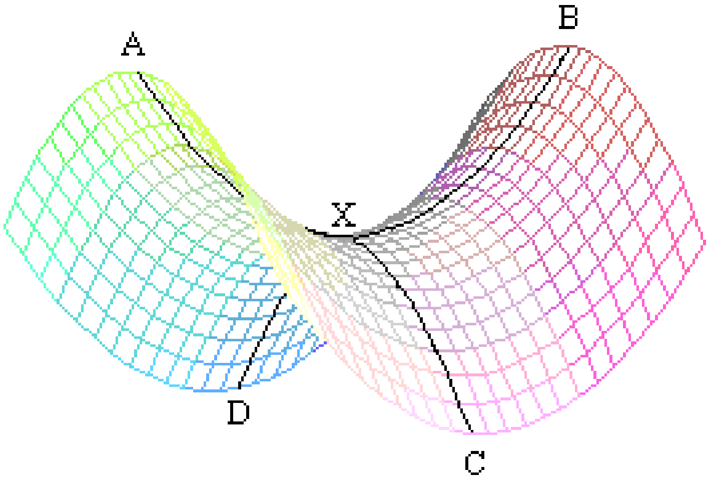


Figure II.1 : *a typical min-max saddle point.*

II.4 Optimality conditions for constrained problems

In this second, more complex situation, let us first provide a strong result, in a favorable framework:

Proposition II.4.1. Optimality Conditions under Constraints

II.4.1 General remarks and results

Let J be a convex function of class $\mathscr{C}^1$, defined on a convex set $C \subseteq \mathbb{R}^n$, and x^* be a point in C. Then, x^* is a (global) minimizer of J on C if and only if

$$\forall y \in C,\ \langle \nabla J(x^*), y - x \rangle_{\mathbb{R}^n} \geq 0.$$

The intuition behind this result is clear: for x^* to be a global minimizer, it is necessary and sufficient for J to increase locally *along the directions that stay within the set of constraints* C. In the case where C is convex, this is expressed more easily because any line segment with endpoints in a convex set remains within that convex set.

In the case of a general set of constraints, the notion of directions that stay (locally) within the set of constraints, called *admissible directions*, is defined as follows:

Definition II.4.2. Cone of Admissible Directions

We say that a curve $\gamma : [0; \varepsilon] \to \mathbb{R}^n$ is admissible at x^* if $\gamma(0) = x^*$ and there exists $\varepsilon' \leq \varepsilon$ such that $\gamma(t) \in C$ for $t \in [0, \varepsilon')$. We call **admissible directions at point** x^* the tangent vectors at point x^* of admissible curves in x^*. We denote by $C_{\mathsf{ad}}(x^*)$ the cone of admissible directions at point x^*.

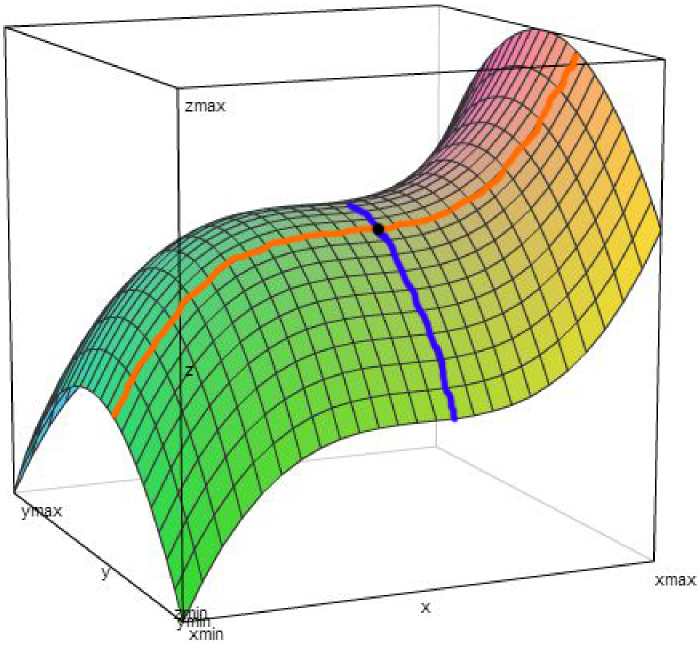


Figure II.2 : *a saddle point presenting an inflection point in the x direction.*

We then have the following result, more general but weaker, since admissible directions are not described as explicitly as in the convex case:

Proposition II.4.3. Euler Inequality

If J is differentiable at x^* and x^* is a minimizer of the function J on C, then

$$\langle \nabla J(x^*), y\rangle_{\mathbb{R}^n} \geq 0, \quad \forall y \in C_{ad}(x^*).$$

II.4.2 The particular case of equality and inequality constraints: Lagrange and Karush-Kuhn-Tucker conditions

We will now focus on constraints expressed in a more quantitative manner (although still implicit), which will allow us to provide necessary (and possibly sufficient) optimality conditions that are more powerful and usable.

We will distinguish between two types of constraints.

- Equality constraints, where the constraint set is defined as

$$C = \{x \in \mathbb{R}^n, \quad f_i(x) = 0, \quad i \in \{1, \cdots, p\}\},$$

where $p \leq n$, and the f_i functions are of class $\mathscr{C}^1$.

- Inequality constraints, where the constraint set is defined as

$$C = \{x \in \mathbb{R}^n, \quad g_i(x) \leq 0, \quad i \in \{1, \cdots, p\}\},$$

where $p \leq n$, and the g_i functions are of class $\mathscr{C}^1$.

A crucial concept for the study of optimality conditions is to determine if a constraint is "fully satisfied"; we will then say that it is "active" or "saturated":

Definition II.4.4. Active/Saturated Constraint

Let an inequality constraint $g_i(x) \leq 0$ and x_0 be a point in $\mathbb{R}^n$.
If x_0 satisfies $g_i(x_0) < 0$, we say that the constraint is inactive at x_0, while if x_0 satisfies $g_i(x_0) = 0$, we say that the constraint is active or saturated at x_0, and we denote $I_0(x^*)$ as the set of active constraints at point x^*, i.e.,

$$I_0(x^*) = \{i \in \{1, \cdots, m\} \mid g_i(x^*) = 0\}.$$

Intuitively, an unsaturated constraint at $x \in C$ is "invisible", meaning there is a whole neighborhood of x in C where it remains satisfied. When the constraint is saturated, on the contrary, we are, so to speak, "on the boundary of C," so there will be directions that need attention to avoid leaving the constraint set C.

Remark II.4.5

Equality constraints are always saturated.

Equality Constraints

As mentioned earlier, the central point is to understand how admissible directions can be characterized. This, combined with Euler's inequality, will allow us to give useful optimality conditions,.

First, we have the following, virtually obvious, property:

Proposition II.4.6.

If y is an admissible direction at point $x^* \in C$, then

$$\langle \nabla f_i(x^*), y \rangle = 0, \quad \forall i \in \{1, \cdots, p\}. \tag{II.2}$$

To have a complete characterization of $C_{ad}(x^*)$, the necessary condition (II.2) would also need to be sufficient. We then speak of "qualification of constraints at point x^*" (see Definition II.4.11), or we say that x^* is a "regular point." This is not always the case (trivial counterexample: $\nabla f_i(x^*) = 0$).

That being said, there are simple criteria for the constraints to be qualified. The most important one is the following:

Proposition II.4.7.

Let $x^* \in C$. We assume that $\nabla f_i(x^*)$ are linearly independent. Then,

$$d \in C_{ad}(x^*) \quad \Longleftrightarrow \quad \langle \nabla f_i(x^*), d\rangle = 0, \quad \forall i \in \{1, \cdots, p\}$$

In other words, when the constraints are qualified at x^*, we can write

$$(C_{ad}(x^*))^{\perp} = \text{Vect}(\nabla f_i(x^*),\ i \in \{1, \cdots, p\}).$$

Now, Euler's inequality can be written as

$$\nabla J(x^*) \in (C_{ad}(x^*))^{\perp}.$$

We can then deduce the following theorem:

Theorem II.4.8. Lagrange (Equality Constraints)

Let J and f_i, $i \in I = \{1, \cdots, p\}$, be $\mathscr{C}^1$ functions. We assume that the constraints are qualified at point x^*. Then, a necessary condition for x^* to be a minimizer of J on the set $C = \{x \in \mathbb{R}^n, f_i(x) = 0, i \in I\}$, is that there exist positive numbers $\lambda_1, \cdots, \lambda_p$ (called Lagrange multipliers) such that

$$\nabla J(x^*) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x^*) = \mathbf{0}$$

Inequality Constraints

Now, we assume that the set C over which we want to minimize J is given by inequality constraints

$$C = \{x \in C,\ g_i(x) \leq 0,\ i = 1, \cdots, m\},$$

where the g_i functions are $\mathscr{C}^1$ functions from $\mathbb{R}^n$ to $\mathbb{R}$.

Similar to equality constraints, we want to exploit Proposition II.4.3 by specifying admissible conditions.

Analogous to Proposition II.4.6, we have the following property, almost immediate:

Proposition II.4.9.

If y is an admissible direction at point x, then

$$\langle \nabla g_i(x), y \rangle \leq 0, \quad \forall i \in I_0(x). \tag{II.3}$$

The set of directions satisfying (II.3) is called the **linearizing cone at** x, denoted as $C_\ell(x)$. So, we have

$$C_{ad}(x) \subset C_\ell(x), \quad \forall x \in C.$$

This is an extension of Proposition II.4.6: note that equality becomes a wide inequality, so the characterization is weaker, and instead of a vector subspace, we obtain a cone.

It can be seen that this linearizing cone is much easier to describe than the cone of admissible directions, which is defined more implicitly. Nevertheless, as in the case of equality constraints, the above inclusion is not always reciprocal.

Example II.4.10

Let $S \in \mathbb{R}^2$ be the set defined by the inequality constraint

$$S = \left\{(x_1, x_2) \in \mathbb{R}^2, \quad g(x) := x_1^2 - x_2^2 \leq 0\right\}.$$

It can be easily verified that at 0, the constraint is saturated. Moreover, $\nabla g(0) = 0$, so in particular

$$C_\ell(0) = \mathbb{R}^2$$

However, the set of admissible directions is equal to S (draw a diagram).

This counterexample illustrates the considerable difference that can exist between $C_{ad}(x^*)$ and $C_\ell(x^*)$, even in a very simple case. There exist nonetheless some simple and general criteria for Proposition II.4.9 to be necessary **and sufficient**. Actually, $C_{ad}(x^*)$ depends on the set C, but $C_\ell(x^*)$ depends on the choice of the functions g_i describing C! Choosing the g_i wisely (or rather, avoiding poor choices) one can still make it work. When that is the case, the constraints are said to be *qualified in* x^* (or that x^* is a *regular point*), as in the case of equality constraints:

Definition II.4.11. Constraint qualification for inequality constraints

Let $x \in C$. The constraints are said to be **qualified in** $x \in C$ (or that x is a regular point) if

$$C_{ad}(x) = C_\ell(x)$$

In that case, the admissible directions are completely described by the inequality (II.3).

Among the criteria for constraint qualification, the most important is the following:

Proposition II.4.12.

Let $x^* \in C$. The constraints are qualified at x^* if and only if there exists a direction $d \in \mathbb{R}^n$ such that for all $i \in I_0(x^*)$,

$$\langle \nabla g_i(x^*), d \rangle_{\mathbb{R}^n} < 0. \tag{II.4}$$

It can also be shown that the constraints are qualified at point x^* if:

- Either the functions g_i are affine.
- Or the vectors $\nabla g_i(x^*)$, $i \in I_0$, are linearly independent.

Now that admissible directions can be described more quantitatively, recall that according to Euler's Inequality (Proposition II.4.3), any local minimizer x^* is such that

$$\langle \nabla J(x^*), d \rangle \geq 0, \quad \forall d \in C_{ad}(x^*),$$

knowing that

$$\langle \nabla g_i, d \rangle \leq 0, \quad \forall d \in C_{ad}(x^*), \quad \forall i \in I_0(x^*).$$

In the case of equality constraints, the two relations above were inequalities. We then obtained an analytical expression of $\nabla J(x^*)$ in terms of $\nabla g_i(x^*)$ by using the fact that for any vector subspace V of $\mathbb{R}^n$, $(V^\perp)^\perp = V$.

In the case of inequality constraints, there is actually a generalization of this result, sometimes called Farkas' Lemma. It is a beautiful result in convex geometry, existing in several forms, the proof of which exceeds the scope of this course. Here is one analytical version:

Lemma II.4.13. Farkas' Lemma

Let $e, v_1, \cdots, v_k \in \mathbb{R}^n$. Then,

$$\{d \in \mathbb{R}^n, \langle v_i, d \rangle \leq 0, \forall i \in \{1, \cdots, k\}\} \subset \{d \in \mathbb{R}^n, \langle e, d \rangle \leq 0\}$$

if and only if e is a linear combination with positive coefficients of $(v_i)_i$.

By applying this lemma with $v_i = -\nabla g_i(x^*)$ and $e = \nabla J(x^*)$, and combining Propositions II.4.3 and II.4.9, we then have the fundamental theorem due to Kuhn-Tucker.

Theorem II.4.14. Karush-Kuhn-Tucker conditions (Inequality Constraints)

Let J and g_i, $i \in I = \{1, \cdots, m\}$, be $\mathscr{C}^1$ functions. We assume that the constraints are qualified at point x^*. Then, a necessary condition for x^* to be a minimizer of J on the set $C = \{x \in \mathbb{R}^n, g_i(x) \leq 0, i \in I\}$, is that there exist positive numbers $\lambda_1, \cdots, \lambda_m$ (called Kuhn-Tucker or generalized Lagrange multipliers) such that

$$\begin{cases} \nabla J(x^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x^*) = \mathbf{0}, \\ \text{with} \quad \lambda_j g_j(x^*) = 0, \quad 1 \leq j \leq m. \end{cases}$$

Please note that this result is not an equivalence but a necessary (and local) condition for optimality. However, it becomes an equivalence when considering a convex function:

Theorem II.4.15.

We reiterate the hypotheses of the Karush-Kuhn-Tucker theorem and further assume that J and the g_i functions are convex. Then, x^* is a minimizer of J on $C = \{x \in \mathbb{R}^n, g_i(x) \leq 0, i \in I\}$ if and only if there exist positive numbers $\lambda_1, \cdots, \lambda_m$ such that

$$\begin{cases} \nabla J(x^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x^*) = \mathbf{0}, \\ \text{with} \quad \lambda_j g_j(x^*) = 0, \quad 1 \leq j \leq m. \end{cases}$$

To summarize all these results, let us state the following theorem, which gives a necessary optimality condition known as the KKT conditios.

Theorem II.4.16. Karush-Kuhn-Tucker, Lagrange

Let J, h_i, $i \in \{1, \cdots, m\}$, g_j, $j \in \{1, \cdots, p\}$, be $\mathscr{C}^1$ functions. We introduce the set of constraints

$$C = \{x \in \mathbb{R}^n, h_i(x^*) = 0, \quad i \in \{1, \cdots, m\}, \quad g_i(x^*) \leq 0, \quad i \in \{1, \cdots, p\}\}.$$

We assume the constraints are qualified at point x^*. Then, a necessary condition for x^* to be a minimizer of J on C is that there exist positive numbers $\lambda_1, \cdots, \lambda_m$, and real numbers $\mu_1, \cdots, \mu_p$, such that

$$\begin{cases} \nabla J(x^*) + \sum_{j=1}^{m} \lambda_j \nabla h_j(x^*) + \sum_{i=1}^{p} \mu_i \nabla g_i(x^*) = 0, \\ \text{with} \quad \mu_j g_j(x^*) = 0, \quad 1 \leq j \leq p. \end{cases} \tag{II.5}$$

II.4.3 Second order Optimality Conditions

The Lagrangian formalism

We begin by introducing a crucial tool in the analysis of constrained optimization problems:

Definition II.4.17. Lagrangian of problem $(\mathscr{P})$

The function defined on $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p_+$ by:

$$\mathscr{L}\left(x,\lambda,\mu\right) = J\left(x\right) + \sum_{j=1}^{m} \lambda_i h_i\left(x\right) + \sum_{i=1}^{p} \mu_i \; g_i\left(x\right)$$

Straightforward computations lead to the following identities:

$$\begin{aligned} \nabla_x \mathscr{L}(x,\lambda,\mu) = \nabla J(x) + \sum_{i=0}^{m} \lambda_i \nabla h_i(x) + \sum_{j=0}^{p} \mu_p \nabla g_j(x), \\ \nabla_\lambda \mathscr{L}(x,\lambda,\mu) = (h_i(x))^\top_{1\leq i\leq m} \\ \nabla_\mu \mathscr{L}(x,\lambda,\mu) = (g_j(x))^\top_{1\leq j\leq p}. \end{aligned}$$

Thus, the Lagrangian encompasses, in an unconstrained function, a great amount of information on the unconstrained problem.

In particular, equality (II.5) can be written as

$$\nabla_x \mathscr{L}\left(x^*,\lambda^*,\mu^*\right) = 0,$$

and the equality constraints translate as

$$\nabla_\lambda \mathscr{L}(x^*,\lambda^*,\mu^*) = 0,$$

and finally, the inequality constraints write

$$\begin{aligned} \frac{\partial \mathscr{L}}{\partial \mu_j}(x^*,\lambda^*,\mu^*) = 0, \quad \forall i \in I_0(x^*), \\ \frac{\partial \mathscr{L}}{\partial \mu_j}(x^*,\lambda^*,\mu^*) < 0, \quad \forall i \notin I_0(x^*). \end{aligned}$$

We clearly see that, up to the unsaturated constraints, points that satisfy the KKT conditions are critical points of the Lagrangian.

Necessary condition

Now, to draw conclusions about whether the obtained solution is indeed a minimizer, further information is required. For this, we can narrow down the number of candidates using a necessary second-order condition.

Theorem II.4.18. Necessary second-order condition for constrained optimization

Assuming J, h, and g are $\mathscr{C}^2$, x^* is a (local) minimizer of J on C, and point x^* is regular (i.e., the constraints are qualified at this point). Denote by λ^* (resp. μ^*) the Lagrange multipliers associated with equality (resp. inequality) constraints. Then, for any direction $d \in \mathbb{R}^n$ satisfying

$$\langle \nabla h_i(x^*), d\rangle_{\mathbb{R}^n} = 0, \ i = 1, \cdots, m$$

$$\langle \nabla g_j\left(x^*\right), d\rangle_{\mathbb{R}^n} = 0, j \in I_0^+\left(x^*\right) \quad \text{and} \quad \langle \nabla g_j\left(x^*\right), d\rangle_{\mathbb{R}^n} \leqslant 0, \ j \in I_0\left(x^*\right) \backslash I_0^+\left(x^*\right)$$

where $I_0^+\left(x^*\right) = \left\{j \mid 1 \leqslant j \leqslant p, \ g_j\left(x^*\right) = 0 \text{ and } \mu_j^* > 0\right\}$, we have

$$\langle \nabla_{xx}^2 \mathscr{L}\left(x^*, \mu^*, \lambda^*\right) d, d\rangle_{\mathbb{R}^n} \geqslant 0,$$

where $\nabla_{xx}^2 \mathscr{L}\left(x, \mu, \lambda\right)$ denotes the second derivative of $\mathscr{L}$ at point $\left(x, \mu, \lambda\right)$.

Definition II.4.19.

The set $I_0^+\left(x^*\right)$ represents the set of strongly active constraints. When $I_0^+\left(x^*\right) = I_0\left(x^*\right)$, that is, $0 = g_j\left(x^*\right) \iff \lambda_j^* > 0$, we say there is strict complementarity.

The directions satisfying the conditions of the above theorem are a particular subset of admissible directions: they are precisely the admissible directions $d \in C_{ad}(x^*)$ such that

$$\langle \nabla J(x^*), d\rangle = 0.$$

In other words, they are the directions that do not give us enough information to conclude on the nature of x^*. Recall indeed that in the unconstrained case we encountered the same problem: critical points can be local minimizers but also local maximizers, or even worse, saddle points.

The sign of the second derivative is therefore a crucial additional condition for the function to increase locally in the direction d.

A sufficient condition

On the other hand, the Lagrangian formalism allows us to formulate sufficient second-order optimality conditions. We need, for the subsequent discussion, the notion of a saddle point for the Lagrangian.

Definition II.4.20. Saddle Point of the Lagrangian

A saddle point of $\mathscr{L}$ on $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}_+^m$ is any triplet $\left(x^*, \mu^*, \lambda^*\right) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}_+^m$ that satisfies

$$\mathscr{L}\left(x^*, \mu, \lambda\right) \leqslant \mathscr{L}\left(x^*, \mu^*, \lambda^*\right) \leqslant \mathscr{L}\left(x, \mu^*, \lambda^*\right) \forall \left(x, \mu, \lambda\right) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}_+^m$$

Theorem II.4.21. Sufficient second-order condition

Assume that J, f, and g are $\mathscr{C}^1$ functions and that the triplet $(x^*,\mu^*,\lambda^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m_+$ is a saddle point of $\mathscr{L}$ on $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m_+$. Then, this triplet satisfies the Kuhn-Tucker conditions.

In the convex case, we have a characterization of Lagrangian saddle points through the Kuhn-Tucker conditions.

Theorem II.4.22.

Suppose J, f, and g are convex and $\mathscr{C}^1$. Then, the triplet $(x^*,\mu^*,\lambda^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m_+$ is a saddle point of $\mathscr{L}$ on $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m_+$ if and only if it satisfies the Kuhn-Tucker conditions.

III.
Algorithms for Unconstrained Optimization

A large class of algorithms that we will consider for optimization problems has the following general form:

$$\text{Given } x^{(0)}, \text{ compute } x^{(k+1)} = x^{(k)} + \rho^{(k)} d^{(k)}. \tag{III.1}$$

The vector $d^{(k)}$ is called the descent direction, and $\rho^{(k)}$ is the step size at the k-th iteration. In practice, we aim to satisfy the inequality

$$J(x^{(k+1)}) \leq J(x^{(k)}).$$

Such algorithms are often called descent methods. Essentially, the difference between these algorithms lies in the choice of the descent direction $d^{(k)}$. Once this direction is chosen, we are more or less reduced to a one-dimensional problem to determine $\rho^{(k)}$. For these reasons, let us begin by analyzing what happens in the one-dimensional case.

III.1 One-Dimensional Algorithms or line Search

Let $\rho \mapsto q(\rho)$ be the cost function that we seek to minimize. For example, one can consider

$$q(\rho) = J(x^{(k)} + \rho d^{(k)})$$

to apply the ideas to the descent method. Suppose we know an interval $[a; b]$ containing the minimizer ρ^* of q and such that q is decreasing on $[a; \rho^*]$ and increasing on $]\rho^*; b]$ (q is then called a unimodal function).

III.1.1 Golden Section Method

We construct a decreasing sequence of intervals $[a_i; b_i]$, all of which contain the minimizer ρ^*. To move from $[a_i; b_i]$ to $[a_{i+1}; b_{i+1}]$, we proceed as follows. We introduce two numbers a' and b' in the interval $[a_i; b_i]$, with $a' < b'$. Then, we compute the values $q(a')$ and $q(b')$. There are three possibilities: If $q(a') < q(b')$, then the minimizer ρ^* necessarily lies to the left of b'. This defines the new interval by setting $a_{i+1} = a_i$ and $b_{i+1} = b'$. Now, consider the case where $q(a') > q(b')$. In this second case, it is clear that the minimizer is to the right of a'. We set: $a_{i+1} = a'$ and $b_{i+1} = b_i$. Finally, the last case is when $q(a') = q(b')$. Then, the minimizer lies in the interval $[a'; b']$. We restrict ourselves to $a_{i+1} = a'$ and $b_{i+1} = b'$.

The following question arises: how do we choose a' and b' in practice? In general, we prioritize two aspects:

(i) We want the reduction factor τ, representing the ratio of the new interval to the previous one, to be constant (or at least bounded). This will ensure the convergence of the algorithm, as the length of the intervals will converge to 0.

(ii) We want to reuse the point that was not chosen in the previous iteration to reduce computational costs.

It can be shown that the simultaneous fulfillment of these two constraints leads to a unique choice of parameters a' and b'. More precisely, suppose q is unimodal. Then we obtain the algorithm in Table III.1, known as the golden section method, where the method is named after the value of the parameter τ.

```
Set $\tau = \frac{1+\sqrt{5}}{2}$
Set $a_0 = a$
Set $b_0 = b$
For $i = 0,...,N^{\max}$
    Set $a' = a_i + \frac{1}{\tau^2}(b_i - a_i)$
    Set $b' = a_i + \frac{1}{\tau}(b_i - a_i)$
    If $(q(a') < q(b'))$
        Set $a_{i+1} = a_i$
        Set $b_{i+1} = b'$
    Else if $(q(a') > q(b'))$
        Set $a_{i+1} = a'$
        Set $b_{i+1} = b_i$
    Else if $(q(a') = q(b'))$
        Set $a_{i+1} = a'$
        Set $b_{i+1} = b'$
    End if
End for i
```

Table III.1 : *Golden Section Method, naive algorithm.*

Here, $N^{\max}$ is the maximum number of iterations we set for the algorithm. To this end, we must validate a stopping criterion of the form: $|b_{i+1} - a_{i+1}| < \varepsilon$, where ε is the error (or tolerance) we allow on the solution ρ^* of the problem.

Recall that the Golden Section is defined by the following geometric problem, which dates back to Euclid at least: *is there a ratio between the length and width of a rectangle such that, when removing the square with sides of the same length as the width of the rectangle, one obtains a rectangle with the same ratio between length and width?*

Taking a,b as the width and length respectively, this simply translates as

$$\frac{b}{a} = \frac{a}{b-a}, \ \textit{i.e.,} \ \frac{a}{b} = \frac{b}{a} - 1, \ \textit{i.e.,} \ \left(\frac{b}{a}\right)^2 - \frac{b}{a} - 1 = 0.$$

Thus the golden section is the positive root of the polynomial

$$X^2 - X - 1.$$

The motivated reader can then check the following statements, which lead to a more efficient algorithm in terms of memory storage:

(i) If $q(a') = q(b')$, then the ratio of the new interval to the previous interval is equal to $1/\tau^3$.

(ii) Otherwise, the ratio of the new interval to the previous interval is exactly $1/\tau$.

(iii) If $a_{i+1} = a_i,\ b_{i+1} = b'$ is chosen, then a' plays the role b' at the next iteration.

(iv) Symmetrically, if $a_{i+1} = a',\ b_{i+1} = b_i$ is chosen, then b' plays the role a' at the next iteration.

This leads to an algorithm requiring fewer computations:

```
Set $\tau = \dfrac{1+\sqrt{5}}{2}$
Set $a_0 = a$ and $b_0 = b$
Set $a' = a + \frac{1}{\tau^2}(b-a)$ and $b' = a + \frac{1}{\tau}(b-a)$
For $i = 0,\ldots,N^{\max}$
    If $(f(a') < f(b'))$
        Set $a_{i+1} = a_i$ and $b_{i+1} = b'$
        Set $b' = a'$
        Set $a' = a_{i+1} + \dfrac{1}{\tau^2}(b_{i+1} - a_{i+1})$
    Else If $(f(a') > f(b'))$
        Set $a_{i+1} = a'$ and $b_{i+1} = b_i$
        Set $a' = b'$
        Set $b' = a_{i+1} + \dfrac{1}{\tau}(b_{i+1} - a_{i+1})$
    Else If $(f(a') = f(b'))$
        Set $a_{i+1} = a'$ and $b_{i+1} = b'$
        Set $a' = a_{i+1} + \dfrac{1}{\tau^2}(b_{i+1} - a_{i+1})$
        Set $b' = a_{i+1} + \dfrac{1}{\tau}(b_{i+1} - a_{i+1})$
    End If
End For i
```

III.1.2 Parabolic Interpolation Method

The main idea of the parabolic interpolation method is to replace the cost function q with its second-order interpolation polynomial p (hence the name parabolic interpolation) at three points x_0, y_0, and z_0 in the interval $[a;b]$. These points are chosen such that: $q(x_0) \geq q(y_0)$ and $q(z_0) \geq q(y_0)$. It can be shown that if we set

$$q[x_0;y_0] = \frac{q(y_0) - q(x_0)}{y_0 - x_0},$$

and

$$q[x_0;y_0;z_0] = \frac{q[z_0;y_0] - q[x_0;y_0]}{z_0 - x_0},$$

then the minimizer is given by

$$\bar{y} = \frac{x_0 + y_0}{2} - \frac{q[x_0;y_0]}{2q[x_0;y_0;z_0]}.$$

It is clear that $\rho^* \in [x_0;z_0]$ according to the previous choices. We then choose the three new points as follows:

- if $\bar{y} \in [x_0; y_0]$, we set $x_1 = x_0$, $y_1 = \bar{y}$, and $z_1 = y_0$, since $\rho^* \in [x_0; y_0]$,
- if $\bar{y} \in [y_0; z_0]$, we set $x_1 = y_0$, $y_1 = \bar{y}$, and $z_1 = z_0$ because $\rho^* \in [y_0; z_0]$.

Then we repeat this process. This leads to the algorithm given in Table III.2.

```
Choose $x_0$, $y_0$, and $z_0$ in $[a;b]$ such that $q(x_0) \geq q(y_0)$ and $q(z_0) \geq q(y_0)$
For $i = 0,...,N^{\max}$
    Set $q[x_i;y_i] = \dfrac{q(y_i) - q(x_i)}{y_i - x_i}$
    Set $q[x_i;y_i;z_i] = \dfrac{q[x_i;z_i] - q[x_i;y_i]}{z_i - x_i}$
    Set $y_{i+1} = \dfrac{x_i + y_i}{2} - \dfrac{q[x_i;y_i]}{2q[x_i;y_i;z_i]}$
    If $y_{i+1} \in [x_i;y_i]$
        Set $x_{i+1} = x_i$
        Set $z_{i+1} = y_i$
    Else if $y_{i+1} \in [y_i;z_i]$
        Set $x_{i+1} = y_i$
        Set $z_{i+1} = z_i$
    End if
End for i
```

Table III.2 : *Parabolic Interpolation Method Algorithm.*

It can be shown that the method is of order 1.3. In fact, we can show that there exists a strictly positive constant C, independent of i, such that we have the inequality

$$|y_{i+1} - \rho^*| \leq C|y_i - \rho^*|^{1.3}.$$

Stating that the method is of order 1.3 means that if the error is 10^{-2} at one step, it will be of the order of $(10^{-2})^{1.3} \approx 2.5 \times 10^{-3}$ at the next step.

One of the difficulties is the initialization of the algorithm. In practice, you can proceed as follows: Choose a point α_0 in the interval $[a;b]$ and a positive displacement step δ. Then, compute $q(\alpha_0)$ and $q(\alpha_0 + \delta)$. There are two situations:

- if $q(\alpha_0) \geq q(\alpha_0 + \delta)$, then q is decreasing, and therefore, ρ^* is to the right of α_0. Continue to compute $q(\alpha_0 + 2\delta)$, $q(\alpha_0 + 3\delta)$, and so on, until you find an integer k such that q is increasing: $q(\alpha_0 + k\delta) > q(\alpha_0 + (k-1)\delta)$, with $k \geq 2$. Then, set
$$x_0 = \alpha_0 + (k-2)\delta, y_0 = \alpha_0 + (k-1)\delta, z_0 = \alpha_0 + k\delta.$$
- if $q(\alpha_0) < q(\alpha_0 + \delta)$, then ρ^* is to the left of $\alpha_0 + \delta$. Take $-\delta$ as the step until you find an integer k such that: $q(\alpha_0 - k\delta) \geq q(\alpha_0 - (k-1)\delta)$. Then, set
$$x_0 = \alpha_0 - k\delta, y_0 = \alpha_0 - (k-1)\delta, z_0 = \alpha_0 - (k-2)\delta.$$

III.2 Some Concepts About Algorithms

Let us now focus on the development of numerical algorithms for solving minimization problems intended to be implemented on computers. We will only cover basic algorithms here, as optimization is a vast field of research and applications. In this discussion, we will only consider local optimization, as the search for a global extremum is beyond the scope of this introduction. Furthermore, we'll assume differentiability throughout this discussion, and non-differentiable optimization is not covered here.

So, what is an algorithm?

Definition III.2.1. Iterative Algorithm

An iterative algorithm is defined by a vector-valued function $\mathbb{A} : \mathbb{R}^n \to \mathbb{R}^n$, which generates a sequence of vector fields $(x^{(k)})_{k\geq 0}$ through a typical construction of the form:

```
Choose $x^{(0)}$ (algorithm initialization phase)
compute $x^{(k+1)} = \mathbb{A}(x^{(k)})$ (the $k$-th iteration)
```

Of course, what we hope for is that the sequence $(x^{(k)})_{k\geq 0}$ converges to a limit x^*, which will be our relative minimizer. We say the algorithm converges to the solution of the minimization problem if this is the case. When we have a given algorithm, two important measures of its effectiveness are:

(i) Convergence speed

(ii) Computational complexity

Convergence speed measures how quickly the sequence $(x^{(k)})_{k\geq 0}$ converges to the point x^*. Computational complexity measures the cost of the operations required to obtain an iteration, with the overall cost being the cost of one iteration multiplied by the number of iterations needed to achieve the desired solution with a pre-defined precision ε. Generally, the following terms are used:

- Linear Convergence: If the Euclidean norm of the vector error in the solution $e^{(k)} = x^* - x^{(k)}$ decreases linearly, then we say that the convergence speed is linear. More precisely, this property can be expressed as:

$$\exists C \in [0,1),\ \exists k_0 \in \mathbb{N} \text{ such that } \forall k \geq k_0, \|e^{(k+1)}\|_{\mathbb{R}^n} \leq C\|e^{(k)}\|_{\mathbb{R}^n}.$$

- Superlinear Convergence: If $\|e^{(k+1)}\|_{\mathbb{R}^n} \leq \gamma^{(k)}\|e^{(k)}\|_{\mathbb{R}^n}$ and $\lim_{k\to+\infty}\gamma^{(k)} = 0$ for $\gamma^{(k)} \geq 0$ for all $k \geq 0$, we say that the convergence speed is superlinear.

- Geometric Convergence: If the sequence $(\gamma^{(k)})_k$ is a geometric sequence, the method is said to have geometric convergence.

- Order-p Method: A method is said to be of order p if it satisfies:

$$\exists C \in [0,1),\ \exists k_0 \in \mathbb{N} \text{ such that } \forall k \geq k_0, \|e^{(k+1)}\|_{\mathbb{R}^n} \leq C\|e^{(k)}\|^p_{\mathbb{R}^n}.$$

  If $p = 2$, we say that the convergence speed is quadratic.

- Local Convergence: If convergence only happens for initial values of $x^{(0)}$ that are close to x^*, we call it local convergence; otherwise, it is global.

III.3 Gradient Methods

III.3.1 Fixed or Optimal Step Gradient

The gradient method is part of a class of methods known as descent methods. The underlying idea behind these methods is to start with an initial point $x^{(0)}$ and seek to minimize a function J. We aim to have $J(x^{(1)}) < J(x^{(0)})$. One straightforward approach is to choose $x^{(1)}$ such that the vector $x^{(1)} - x^{(0)}$ is collinear with a descent direction $d^{(0)} \neq 0$. This can be written as $x^{(1)} - x^{(0)} = \rho^{(0)} d^{(1)}$, where $\rho^{(0)}$ is the step size. We can iterate in this manner by setting $x^{(k)}$, $d^{(k)}$, and $\rho^{(k)}$ to obtain $x^{(k+1)}$ as follows:

$$\begin{gathered}\texttt{Choose } x^{(0)} \\ \texttt{compute } x^{(k+1)} = x^{(k)} + \rho^{(k)} d^{(k)}\end{gathered}$$

There are many choices for $d^{(k)}$ and $\rho^{(k)}$. One common choice for $d^{(k)}$ is to set it as $-\nabla J(x^{(k)})$ since it ensures that $J(x^{(k+1)}) < J(x^{(k)})$ if $\rho^{(k)}$ is sufficiently small. This approach leads to the method known as the gradient method. Typically, when working on a numerical solution to a problem, a stopping criterion is set based on $\left\|x^{(k+1)} - x^{(k)}\right\| < \varepsilon$, and a maximum number of iterations $k^{\max}$ is imposed. The gradient method is presented as shown in Table III.3.

```
set $k = 0$
choose $x^{(0)}$
while $(\left\|x^{(k+1)} - x^{(k)}\right\|_{\mathbb{R}^n} \geq \varepsilon)$ and $(k \leq k^{\max})$ do
     compute $d^{(k)} = -\nabla J(x^{(k)})$
     compute $\rho^{(k)}$
     set $x^{(k+1)} = x^{(k)} + \rho^{(k)} d^{(k)}$
end while
```

Table III.3 : *Gradient Method Algorithm.*

Even though these methods are conceptually simple and can be implemented directly, they can be slow in practice. They converge, but often under complex convergence conditions. For example, consider the following result.

Theorem III.3.1. Variable Step Gradient Algorithm

Let J be a $\mathscr{C}^1$ function from $\mathbb{R}^n$ to $\mathbb{R}$, and let x^* be a minimizer of J. Suppose that:

(i) J is α-elliptic, which means there exists $\alpha > 0$ such that for all $(x, y) \in (\mathbb{R}^n)^2$, $\langle \nabla J(x) - \nabla J(y), x - y \rangle_{\mathbb{R}^n} \geq \alpha \|x - y\|^2_{\mathbb{R}^n}$.

(ii) The gradient map ∇J is Lipschitz, which means there exists $M > 0$ such that for all $(x, y) \in (\mathbb{R}^n)^2$, $\|\nabla J(x) - \nabla J(y)\|_{\mathbb{R}^n} \leq M \|x - y\|_{\mathbb{R}^n}$.

If there exist two real numbers a and b such that $\rho^{(k)}$ satisfies $0 < a < \rho^{(k)} < b < \frac{2\alpha}{M^2}$ for all $k \geq 0$, then the gradient method defined by

$$x^{(k+1)} = x^{(k)} - \rho^{(k)} \nabla J(x^{(k)})$$

converges for any choice of $x^{(0)}$ geometrically. In other words,

$$\exists \beta \in]0;1[, \left\| x^{(k+1)} - x^* \right\|_{\mathbb{R}^n} \leq \beta^k \left\| x^{(0)} - x^* \right\|_{\mathbb{R}^n}.$$

The choice of step size $\rho^{(k)}$ can be done in the following ways:

- Fixed Step: Set $\rho^{(k)} = \rho$ as a constant value.
- Optimal Step: Choose $\rho^{(k)}$ as the minimizer of the function $q(\rho) = J(x^{(k)} - \rho \nabla J(x^{(k)}))$; this is the optimal step gradient method.
- Computed Step: compute $\rho^{(k)}$ using methods presented earlier.

In the case of the optimal step gradient method, we have the same convergence result as previously under weak assumptions about J.

Theorem III.3.2. Optimal Step Gradient Algorithm

Let J be a $\mathscr{C}^1$ function from $\mathbb{R}^n$ to $\mathbb{R}$, and x^* be a minimizer of J. Assume J is α-elliptic. Then, the optimal step gradient method converges for any choice of the initial vector $x^{(0)}$.

Remark III.3.3

Even in the case of the optimal step gradient method, which is theoretically the best in terms of convergence speed, it can be slow due to the poor conditioning of the Hessian matrix of J. Additionally, one can consider convergence criteria on the gradient of J at $x^{(k)}$: $\left\| \nabla J(x^{(k)}) \right\| < \varepsilon_1$.

III.3.2 Other rules

In practice, it is not necessary to implement a complete method for finding a minimizer when the problem in one dimension is just a step in a more complicated minimization algorithm in dimension N. The goal is simply to find a "reasonable" step to ensure the global convergence of the descent method considered elsewhere.

We are still in the same framework: the objective is to find a "reasonable" step $t > 0$ to decrease the function of a single variable $q(t) = J(X + tD)$ (where X is the point obtained in the previous iteration and D is the descent direction). Since $q'(0) = \nabla J(X) \cdot D < 0$, we are sure to decrease the function q by taking t a little to the right of 0. However, we are faced with two contradictory requirements:

(a) t should not be too large because there is a risk that the function q will not decrease or that the behavior of q will be somewhat oscillatory.

(b) t should not be too small, as the algorithm will not progress quickly enough.

There are two classical ways to try to satisfy these two objectives:

Goldstein Rule (1967):

Choose two numbers m_1 and m_2 with $0 < m_1 < m_2 < 1$ and search for a value of t that satisfies:

(a) $q(t) \leq q(0) + m_1 t q'(0)$

(b) $q(t) \geq q(0) + m_2 t q'(0)$

Wolfe Rule (1969):

Choose two numbers m_1 and m_2 with $0 < m_1 < m_2 < 1$ (for example, $m_1 = 0.1$ and $m_2 = 0.7$) and search for a value of t that satisfies:

(a) $q(t) \leq q(0) + m_1 t q'(0)$

(b) $q'(t) \geq m_2 q'(0)$

Implementation of the Previous Rules in a General Algorithm Using Descent Directions:

Let J be the function (from $\mathbb{R}^N$ to $\mathbb{R}$) to be minimized. At iteration k, $X = X_k$ (the point obtained in the previous step) and $D = D_k$ (the descent direction) have been computed elsewhere. The goal is to find $X_{k+1} = X_k + tD_k$ for a certain step size t, which needs to be determined. The local cost function is $q(t) = J(X + tD)$. Two constants m_1 and m_2 are fixed with $0 < m_1 < m_2 < 1$.

(i) Initialization: $t_g = 0$, $t_d = +\infty$, $q'(0) = \nabla J(X) \cdot D$. Choose an initial value of t in $]t_g, t_d[$.

(ii) If $q(t) \leq q(0) + m_1 t q'(0)$, go to **3**; otherwise, set $t_d = t$ and go to **5**.

(iii) Goldstein Rule:

$$\begin{cases} \text{If } q(t) \geq q(0) + m_2 t q'(0), \textbf{stop} \\ \text{Otherwise, go to } \mathbf{4}. \end{cases}$$

Wolfe Rule:

$$\begin{cases} q'(t) = \nabla J(X + tD) \cdot D \\ \text{If } q'(t) \geq m_2 q'(0), \textbf{stop} \\ \text{Otherwise, go to } \mathbf{4}. \end{cases}$$

(iv) Set $t_g = t$.

(v) Initialize with a value of t in $]t_g, t_d[$ and return to **2**.

IV.

Algorithms for Constrained Optimization

In the previous chapter, we dealt with unconstrained minimization problems. We now consider minimization problems under constraints.

All of the algorithms presented in this chapter draw from the same general idea: to find a way to reformulate constrained optimization problems into unconstrained problems, in order to apply the methods of the previous chapter.

IV.1 Algorithms

IV.1.1 Projected Gradient Method

Recall that in the unconstrained case, the gradient algorithm, which is a descent method, is formulated as follows:

$$\left\{\begin{array}{l} x^{(0)} \in \mathbb{R}^n \text{ given.} \\ x^{(k+1)} = x^{(k)} + \rho^{(k)} d^{(k)}, d^{(k)} \in \mathbb{R}^n \setminus \{0\}, \rho^{(k)} \in \mathbb{R}^{+*} \end{array}\right.$$

where $f^{(k)}$ and $d^{(k)}$ are chosen such that $J\left(x^{(k+1)}\right) \leqslant J\left(x^{(k)}\right)$. When minimizing over a set of constraints C assumed closed, it is not guaranteed that $x^{(k)}$ remains in C. Therefore, it is necessary to project it onto C. This is done through a projection on C, denoted by

$$\begin{array}{rccc} \Pi_C : & \mathbb{R}^n & \rightarrow & C \\ & x & \longmapsto & \Pi_C(x) \end{array}.$$

This naturally leads to the projected gradient algorithm, which is identical to the gradient algorithm except for the projection.

```
Initialization
$k = 0$ ; choice of $x_0$ and $g_0 > 0$
Iteration $k$
While termination criterion not satisfied
        $\widehat{x}^{(k+1)} = x^{(k)} - \rho^{(k)} \nabla J\left(x^{(k)}\right)$
        $x^{(k+1)} = \Pi_C \widehat{x}^{(k+1)}$
        $k = k + 1$
End
```

Table IV.1 : *Projected Gradient Algorithm.*

Theorem IV.1.1. Projected Gradient Algorithm

Let J be a $\mathscr{C}^1$ function from $\mathbb{R}^n$ to $\mathbb{R}$. Assume J is α-elliptic with M-Lipschitz derivative (see theorem III.3.1). Then, if we choose the step $\rho^{(k)}$ in an interval $[\beta_1;\beta_2]$ such that $0<\beta_1<\beta_2<\frac{2\alpha}{M^2}$, the sequence $\left(x^{(k)}\right)_k$ defined by the projected gradient method converges to the solution of problem $(\mathscr{P})$.

This approach may seem straightforward at first. However, it is essential to remember that one must know the projection operator on C, which is not straightforward a priori. It is clearly out of the question to solve the minimization problem for fixed $x\in\mathbb{R}^n$

$$\inf_{y\in C}\|y-x\|^2_{\mathbb{R}^n}$$

which itself is a minimization problem on the same set of constraints. In some special cases, one can explicitly express the projection operator.

Suppose C is an intersection of half-spaces of the type

$$C=\{x=(x_1,\ldots,x_n)\,,\,x_i\geqslant a_i,\,i\in I,\,x_j\leqslant b_j,\,j\in J\}$$

where I and J are sets of indices not necessarily disjoint (this notably includes the case of boxes).

Regarding the case of a constraint of the form $x_i\geqslant a_i$, it can be easily seen that the i-th coordinate of $\Pi_C x$ will be x_i if $x_i\geqslant a_i$ and a_i otherwise. The same reasoning applies for the case of a constraint of the form $x_j\leqslant b_j$. This can be summarized as

$$(\Pi_C x)_i=\max\left(x_i,a_i\right)\ \text{ or }\ (\Pi_C x)_j=\min\left(x_j,b_j\right).$$

If the set of constraints is $C=\{x\in\mathbb{R}^n\mid x\in B_f(x_0,R)\}$, where $B_f(x_0,R)$ represents the closed ball of center x_0 and radius $R>0$ (for the Euclidean norm), the projection Π_C is then written as

$$\Pi_C x=\left\{\begin{array}{ll} x & ,\text{ if } x\in C\ ;\\ x_0+R\frac{x-x_0}{\|x-x_0\|} & ,\text{ if } x\notin C\end{array}\right.$$

IV.1.2 Penalty Methods

Penalty methods are commonly used in practice because they are quite simple. They start from the following principle: we replace the problem with constraints

$$(\mathscr{P})\qquad\inf_{x\in C\subset\mathbb{R}^n}J(x)$$

by an unconstrained problem

$$(\mathscr{P}_\varepsilon)\qquad\inf_{x\in\mathbb{R}^n}J(x)+\frac{1}{\varepsilon}\alpha(x),$$

where $\alpha:\mathbb{R}^n\to\mathbb{R}$ is a penalty function for the constraints, and $\varepsilon>0$. Ideally, one wishes to find functions α such that the problems $(\mathscr{P})$ and $(\mathscr{P}_\varepsilon)$ are equivalent, meaning they have the same solutions. In this case, we say the penalty is exact. For example, one can choose

$$\alpha(x)=\left\{\begin{array}{ll}0 & \text{if } x\in C\\ +\infty & \text{if } x\notin C\end{array}\right.$$

This function does not have good mathematical properties (especially differentiability) to apply unconstrained resolution techniques.

Generally, we perform what is called an *inexact penalty*, where the problem $(\mathscr{P})$ has solutions that are not solutions of $(\mathscr{P}_\varepsilon)$; the set of solutions of $(\mathscr{P}_\varepsilon)$ does not cover the entire set of solutions of $(\mathscr{P})$. In this case, one can find functions α that are differentiable, allowing the use of unconstrained minimization results.

Intuitive Principle of Penalty Methods: We hope that when ε becomes small, the term $\frac{1}{\varepsilon}\alpha(x)$ will be very large if $x \notin C$, so that the constraint will naturally be (almost) satisfied at the optimum for the problem $(\mathscr{P}_\varepsilon)$.

Let us give some examples of penalty functions α, assuming that α satisfies the following properties:

(i) α is continuous on $\mathbb{R}^n$,

(ii) $\forall x \in \mathbb{R}^n$, $\alpha(x) \geqslant 0$,

(iii) $\alpha(x) = 0 \Leftrightarrow x \in C$.

- Constraint $x \leqslant 0$: the function α is $\alpha(x) = \|x^+\|^2_{\mathbb{R}^n}$,
- Constraint $h(x) = 0$: the function α is $\alpha(x) = \|h(x)\|^2_{\mathbb{R}^n}$,
- Constraint $g(x) \leqslant 0$: the function α is $\alpha(x) = \|g(x)^+\|^2_{\mathbb{R}^n}$,

where $x^+ = (x_1^+, \cdots, x_n^+)$. We then have the following convergence result:

Theorem IV.1.2. Penalty Algorithm

Let J be a continuous and coercive function. Let C be a non-empty closed set. Assume α satisfies the following conditions:

(i) α is continuous on $\mathbb{R}^n$.

(ii) $\forall x \in \mathbb{R}^n, \alpha(x) \geqslant 0$.

(iii) $\alpha(x) = 0 \Leftrightarrow x \in C$.

Then,

- $\forall \varepsilon > 0$, $(\mathscr{P}_\varepsilon)$ has at least one solution x_ε,
- The family $(x_\varepsilon)_{\varepsilon>0}$ is bounded,
- Any convergent subsequence of $(x_\varepsilon)_{\varepsilon>0}$ converges to a solution of $(\mathscr{P})$ as $\varepsilon \searrow 0$.

We then get the following algorithm for exterior penalty.

```
Initialization
 $k = 1$
Choose $x^{(0)} \in \mathbb{R}^n, \varepsilon^{(1)} > 0$
Iteration $k$ while the termination criterion is not satisfied:
      a) Solve the sub-problem $(\mathscr{P}_{\varepsilon^{(k)}})$
         $\begin{cases} \min J(x) + \frac{1}{\varepsilon^{(k)}}\alpha(x) \\ x \in \mathbb{R}^n \end{cases}$
         with $x^{(k-1)}$ as the initialization point.
      b) $k \leftarrow k + 1$, take $\varepsilon^{(k+1)} < \varepsilon^{(k)}$.
```

Table IV.2 : *Exterior penalty Algorithm*

Remark IV.1.3 Interior point methods

This algorithm is called *exterior penalty* because the x_ε lie outside the constraint set, in other words, they converge (in terms of subsequences) to the real minimizer on C from outside.

There exist inside penalty methods as well, called *interior point methods*. They consist in adding *barrier functions* which are defined on the interior of C and go to $+\infty$ close to the boundary of C.

IV.1.3 Duality Method: Uzawa's Algorithm

The technique presented here comes from the optimization area called convex duality theory. The general idea is to consider the Lagrangian $\mathscr{L}$ instead of the function J; this choice is motivated (at least) by two reasons:

- The Lagrangian function encompasses both the function J and the constraints h and g, well representing the problem.
- Secondly, we have seen that a necessary first-order condition for x^* to be a minimizer of J with constraints is that x^* (associated with Lagrange multipliers) is a critical point of $\mathscr{L}$.

In this paragraph, we will consider the particular case where

$$C = \left\{x \in \mathbb{R}^N,\ h_i\left(x\right) = 0,\ i = 1,\ldots,m,\ g_j\left(x\right) \leqslant 0,\ j = 1,\ldots,p\right\}.$$

. Recall that the Lagrangian of the problem is

$$\mathscr{L}\left(x,\mu,\lambda\right) = J\left(x\right) + \sum_{i=1}^{m} \lambda_i f_i\left(x\right) + \sum_{j=1}^{p} \mu_j g_j\left(x\right)$$

Recalling Definition II.4.20 and Theorem II.4.21, we can design an algorithmic procedure for finding the minimizer. We will seek a triplet $(x^*,\mu^*,\lambda^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m_+$ satisfying the Kuhn-Tucker conditions as follows:

(i) For fixed (μ^*,λ^*) in $\mathbb{R}^n \times (\mathbb{R}^+)^m$, we will search for the unconstrained minimizer (over $\mathbb{R}^n$) of the function $x \longmapsto \mathscr{L}\left(x,\mu^*,\lambda^*\right)$.

(ii) For fixed x^* in $\mathbb{R}^n$, we seek the maximum over $\mathbb{R}^p \times \mathbb{R}^m_+$ (i.e., simple bound constraints) of the function $(\mu,\lambda) \longmapsto \mathscr{L}\left(x^*,\mu,\lambda\right)$

We perform these two calculations simultaneously. This leads us to the Uzawa algorithm.

```
Initialization:  $k = 0$, choose $\mu^{(0)} \in \mathbb{R}^p$ and $\lambda^{(0)} \in \mathbb{R}^m_+$
Iteration.  While the termination criterion is not satisfied:
        a) compute $x^{(k)} \in \mathbb{R}^n$ solution of
        $\left(\mathscr{P}^{(k)}\right) \begin{cases} \min\ \mathscr{L}\left(x,\mu^{(k)},\lambda^{(k)}\right) \\ x \in \mathbb{R}^n \end{cases}$
        b) compute $\mu^{(k+1)}$ and $\lambda^{(k+1)}$ with
        $\begin{cases} \mu_i^{(k+1)} = \mu_i^{(k)} + \rho\ f_i\left(x^{(k)}\right),i = 1,...,p \\ \lambda_j^{(k+1)} = \max\left(0,\lambda_j^{(k)} + \rho g_j\left(x^{(k)}\right)\right),\ j = 1,\dots,m \end{cases}$
        where $\rho > 0$ is a fixed real (set by the user).
```

Table IV.3 : *Uzawa's Algorithm*

The first step involves solving:

$$\nabla_x \mathscr{L}\left(x,\mu^{(k)},\lambda^{(k)}\right) = \nabla J\left(x\right) + \sum_{j=1}^{p} \mu_j^{(k)} \nabla f_j\left(x\right) + \sum_{i=1}^{m} \lambda_i^{(k)} \nabla g_i\left(x\right) = 0$$

The second step is straightforward.
We then have the following convergence theorem.

Theorem IV.1.4. Uzawa's Algorithm

Assuming that J is $\mathscr{C}^1$ and elliptic, f is affine, g is convex of class $\mathscr{C}^1$, and h and g are Lipschitz. Moreover, assuming that the Lagrangian $\mathscr{L}$ has a saddle point (x^*,μ^*,λ^*) in $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m_+$.
Then, there exist ρ_1, ρ_2, with $0 < \rho_1 < \rho_2$, such that for all $\rho \in [\rho_1,\rho_2]$, the sequence $\left(x^{(k)}\right)_{k>0}$ generated by Uzawa's algorithm converges to x^*.