OPT : an Introduction to Numerical and Combinatorial Optimization

E. Fabre, DistribCom team, IRISA

24th September 2007

Abstract

Master 2, research in Computer Science, '07

These notes correspond to the first lectures of the OPT module. They propose a rapid classification of optimization problems, then focus on optimization methods in continous domains, so-called numerical optimization. The following lectures will address combinatorial optimization (R. Andonov), and some aspects of game theory (S. Pinchinat).

The objective of this first part is to explain the main principles of numerical optimization methods, in order to help the student understand what type of problem he is facing, what is its difficulty, and what method he should try. Technical aspects related to the convergence properties of the algorithms are not detailed. In the same way, for the sake of simplicity, we assume the functions to optimize are regular (continuous, differentiable, etc.), and some theorem proofs are not given. We rather insist on the geometrical interpretations. The reference list proposed at the end should allow the reader to fill the holes that correspond to the particular case he has to deal with. Caution : notations are not standardized in this domain, and each book generally proposes its own notations.

Background to understand these notes : derivation of functions of several variables, geometry in $\mathbb{R}^d$ and basics of linear algebra.

Contents

1 Classes of optimization problems: an overview

1.1 What is it all about?

Let $f : \mathcal{D} \to \mathbb{R}$ be a real valued function defined over some domain $\mathcal{D}$. An optimization problem consists in computing

$$x^* \quad = \quad \arg\min_{x \in \mathcal{D}} f(x) \tag{1}$$

so that $\forall x \in \mathcal{D},\ f(x^*) \leq f(x)$. In general one is simply interested in obtaining *one* point x^* among the possibly many points that achieve the minimum of f. The function f is called the **cost function**. Optimization problems sometimes appear with a *max* instead of the *min*, and f is then called the **objective function**. This doesn't change the formalism : maximizing f is equivalent to minimizing $-f$.

The **parameters** x to optimize may lie in a continuous domain, *i.e.* a subset of $\mathbb{R}^d$, or in a discrete domain. This establishes the distinction between **numerical optimization** and **combinatorial optimization**, that use very different techniques, although some connections exist.

In numerical optimization, it is often convenient to further distinguish between unconstrained problems, for which x can take any value in $\mathbb{R}^d$, and constrained ones, for which the range of x is limited by extra conditions (for example $\|x\|^2 = 1$).

Notations. The coordinates of an element $x \in \mathbb{R}^d$ will be denoted as $x_1, ..., x_d$. By abuse of notations, we sometimes write $f(x) = f(x_1, ..., x_d)$ to stress that f is a function of d variables. To simplify the expression of some functions, we use the matrix notation, where x and other vectors are represented as **column vectors** (or matrices $d \times 1$) : $x = [x_1, ..., x_d]^t$. The superscript t denotes the transposition operation on matrices.

1.2 Numerical optimization

In continuous domains, problems can be classified as follows, by order of complexity.

1.2.1 Unconstrained problems

Here x is free to explore $\mathbb{R}^d$ (but the dimension d can be large). f is generally continuous, and often C^n, *i.e.* n times differentiable, for some $n > 0$.

Quadratic problem. Here $f : \mathbb{R}^d \to \mathbb{R}$ is a quadratic form, *i.e.* can be expressed as $f(x) = x^t A x + b^t x$ where A is a symmetric matrix of $\mathbb{R}^{d \times d}$ and $b \in \mathbb{R}^d$ a vector. We study this case in detail below (section 2.1). These problems can be solved exactly and their solutions admit a closed form expression.

We recall that $b^t x = \sum_{i=1}^d b_i x_i$ is the *scalar product* of vectors in $\mathbb{R}^d$. The product Ax of a matrix by a vector yields a vector, and so $x^t A x$ is a scalar.

Convex problem. This situation refers to cost functions that satisfy

$$\forall\, 0 \leq \alpha \leq 1,\ \ \forall\, x, y \in \mathcal{D}, \qquad f[\alpha x + (1-\alpha) y] \ \leq\ \alpha f(x) + (1-\alpha) f(y) \tag{2}$$

Let us say that x is a **local minimum** of f iff there exist some $\epsilon > 0$ such that $\|y - x\| < \epsilon$ implies $f(x) \leq f(y)$. In other words, inside a small ball of radius ϵ around x, there is no smaller value of f than $f(x)$. Convex functions satisfy the following nice property: *every local minimum of f is also a global minimum.* This is very convenient in practice: this means that optimization methods based on a **descent scheme** (find around the current x some y for which f is smaller) will not converge until they reach a global minimum.

Non linear problem. This refers to all other cases. However the most regular (=differentiable) f is, the simpler the optimization problem in general. There are few cases for which closed form expressions of the optimum exist. In general, one must deploy numerical search methods, that try to improve a current estimate x of the min by making a small step around x. Such methods often get trapped in local minima of f. Imagine for example f having the shape of a bended egg box.

1.2.2 Problems with linear constraints

Here we assume $x \in \mathbb{R}^d$ and also satisfies

$$\forall\, 1 \leq j \leq n, \quad \theta_j(x) \ = \ {b_j}^t\, x + c_j \ \leq \ 0 \tag{3}$$

where $b_j \in \mathbb{R}^d$ is a vector, ${b_j}^t\, x$ denotes the scalar product (of column vectors), and $c_j \in \mathbb{R}$.

Linear program. Corresponds to the case where the objective function f is itself a linear function $f(x) = b^t x$, $b \in \mathbb{R}^d$. These problems are often presented as a maximization issue, and one generally finds the constraints $x_i \geq 0$ among the θ_j. Each constraint $\theta_j(x)$ defines a half-space of $\mathbb{R}^d$, and the legal values of x are thus contained in the intersection of these half-spaces, which is a *convex* volume. The boundary of this volume is called a **simplex** (Fig. 1).

The equations $f(x) = t$ for $t \in \mathbb{R}$ define parallel hyperplanes, that are *orthogonal* to the vector b. In effect, let x^1, x^2 satisfy $f(x^1) = f(x^2) = t$, then $b^t(x^1 - x^2) = 0$, which means that $x^1 - x^2$ is orthogonal to b. We will see later that b is the *gradient* of f, which indicates the direction in which f increases the most.

Some of the hyperplanes $f(x) = t$ cross the simplex, others don't, and some of them are just "at the limit." The problem thus consists in finding the maximal value of t for which $f(x) = t$ still touches the simplex, and to isolate the intersection points x^* of this hyperplane with the simplex. Due to the linearity of f and of the boundaries of the simplex, the contact is either on a corner of the simplex, or on an edge, or on a face. In any case, an optimum x^* is found on a corner.

The **simplex method** thus amounts to exploring only the corners of the simplex to find an optimum. In that sense, it is related to combinatorial methods. It is found

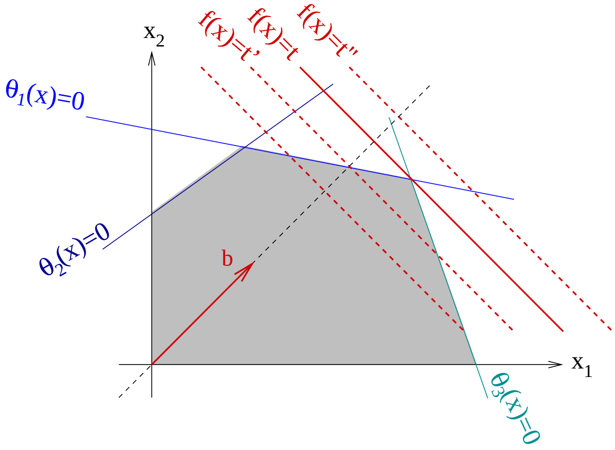

Figure 1: *A simplex (border of the gray zone) in* $\mathbb{R}^2$ *defined by 5 linear constraints, and different level planes of the cost function,* $t' < t < t''$*.*

in the literature under the name **linear programming**.[1] The **interior points methods** rather explore the interior of the simplex and try to get closer to the faces, while optimizing f. They are mostly applied when f is not linear.

Example. Products A, B and C are sold at prices p_A, p_B and p_C per unit, respectively. To produce each unit of these products, one consumes quantities m_A, m_B, m_C of matter, and needs t_A, t_B, t_C time. For a given amount of matter, and a limited time, find the right quantities to produce in order to maximize the income.

Quadratic program. This corresponds to the case where f is quadratic, and constraints are linear. Addressed by **quadratic programming** methods.

Non-linear program. All other functions. Again when f is convex, things are simpler.

1.2.3 Problems with non-linear constraints

Here, the difficulty also comes from the nature of constraints. Implicitly, it is supposed that constraints are not strong enough to transform domain $\mathcal{D}$ into a discrete domain... The classification is essentially the same as above.

1.3 Combinatorial optimization

Here x ranges over a discrete domain $\mathcal{D}$. There can exist a topology in $\mathcal{D}$, for example $\mathcal{D} = \mathbb{N}^d$, which gives sense to the notion of local minimum. Or $\mathcal{D}$ can be more complex, for example the set of paths in a graph.

Again we classify them according to their difficulty.

[1] The term "programming" doesn't mean programming a computer, but refers to programming tasks or actions to achieve an objective. This term comes from the early military applications of optimization methods, which were also at the origin of the discipline called "operational research."

1.3.1 Easy problems

This refers to problems for which an exact solution can be found in polynomial time. Domains $\mathcal{D}$ are often defined from a graph. For example :

Dynamic program. Consider a graph $G = (V, E, c)$ where V is the vertex set, $E \subseteq V \times V$ represent the edges (non-oriented for simplicity), and $c : E \to \mathbb{R}$ is a value on each edge. One wants to determine the shortest path from $s \in V$ to $s' \in V$, where the length of a path is obtained by summing the lengths $c(e)$ over all edges e composing this path.

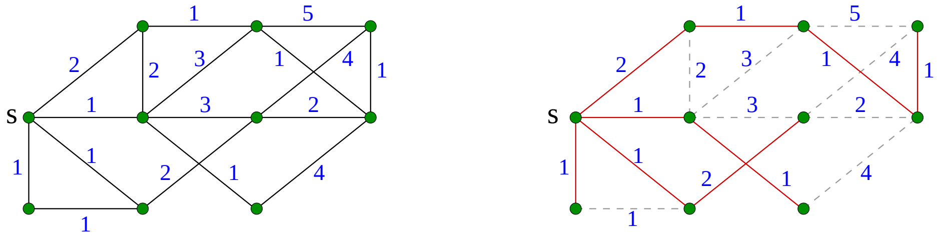

Figure 2: *Left : A graph with edge costs. Right : shortest paths from s to all nodes.*

Observing that if the shortest path goes through s'', then the section from s to s'' is also the shortest, one derives a recursion on shortest paths from s to all other nodes (Fig. 2). Which opens the way to a recursive resolution called **dynamic programming**.

Exercise 1 *Write a dynamic programming algorithm, that computes the minimal distance of s to all nodes, and prove its convergence. Hint : introduce a set of active nodes, that have been newly reached, or for which the shortest distance from s has just been modified. Initialize this set with the neighbours of s. Build a recursion by exploring the neighbuors of one node in the active set (that will then becom inactive).*

Optimal covering tree. Here the problem is to select edges in G that reach all nodes and has minimal weight. A trivial greedy procedure gives the solution (Prim or Kruskal algorithms), see Fig. 3. The idea is to sort edges by increasing weights, and to take them in order. When an edge closes a cycle, it is rejected.

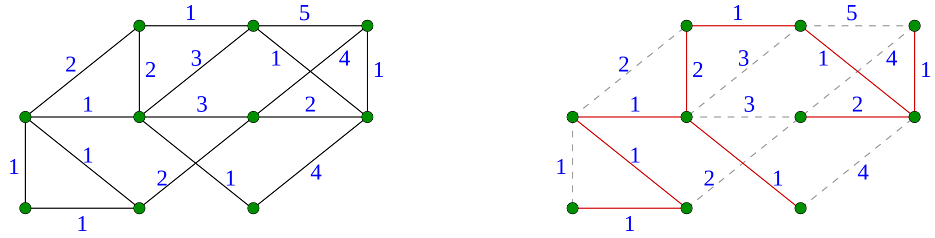

Figure 3: *Left : A graph with edge costs. Right : An optimal covering tree.*

Exercise 2 *Write the algorithm and prove its convergence.*

Flow problems. Now $c(e)$ represents the capacity of an edge, seen as a pipe. The problem is to compute the maximal flow (of "liquid," say) from a node s to a node s', such that the flow on each edge remains lower than its capacity.

Again, a simple greedy procedure gives the solution (the Ford-Fulkerson algorithm) : find a path from s to s', maximize the flow on it, then reduce the capacity of each edge on this path by the value of this maximal flow. Repeat until s and s' are disconnected (an edge of capacity 0 is equivalent to its absence).

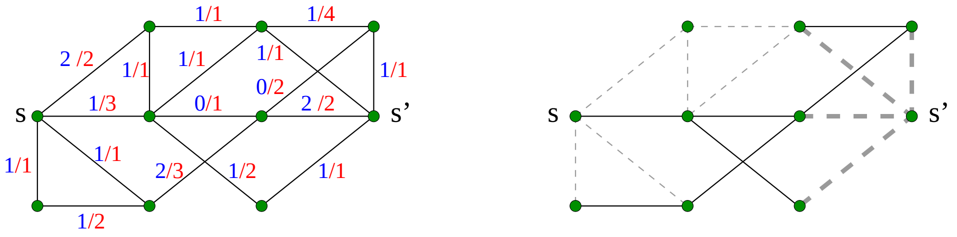

Figure 4: *Left : A maximal flow from s to s'. On each edge, the first value (blue) indicates the flow, the second one (red) the capacity. Right : edges reaching their capacity are in dashed grey, thick edges represent a minimal cut, of value 5.*

Exercise 3 *Write the algorithm and prove its convergence.*

Recall that finding the value of the **maximal flow** is equivalent to finding the value of a **minimal cut** in the graph, where a cut is a set of edges that separates s from s', and where the price of a cut is the sum of the $c(e)$ over this set of edges (see Fig. 4).

Multi-flow problems, where one wants to maximize several flows over the same graph, are different in nature and more complex to solve.

1.3.2 Integer linear program

As for a linear program, the objective function f is linear. The domain $\mathcal{D}$ is defined by linear inequalities $\theta_j(x) = {b_j}^t x + c_j \leq 0$ and by $x \in \mathbb{N}^d$ (see Fig. 5). By rounding the optimal solution of a linear program, one can get an approximation (or initial guess) of the optimum of an integer linear program. The quality of this approximation varies with the nature of constraints : when the simplex is very flat, the best integer solution may be far from solutions in $\mathbb{R}^d$. In general, such problems are NP hard.

Example : the **knapsack problem.** Consider a limited volume knapsack, and a collection of objects O_k with volume v_k and utility u_k. The objective is to maximize the global utility of your knapsack, by selecting objects, under the volume constraint.

1.3.3 Complex problems

Very rapidly, combinatorial optimization problems become NP hard or NP complete, and the search for the optimum is hopeless. But good approximations may be accessible. For example :

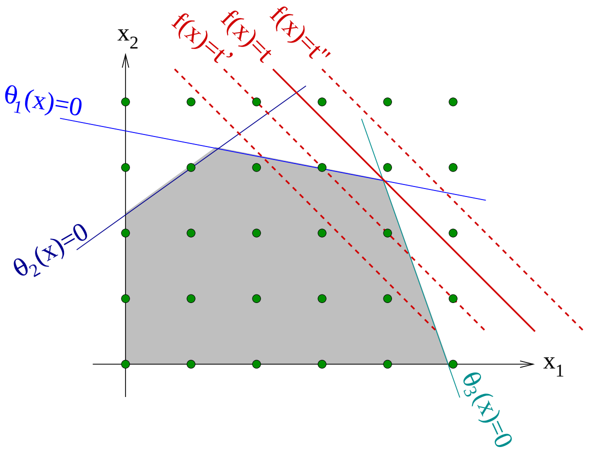

Figure 5: *Integer linear programming : only points inside the simplex and with integer coordinates are permitted (represented as green dots).*

- **Traveling salesman.** Consists in finding an optimal hamiltonian path in G, *i.e.* a cycle that goes once and only once through each vertex. The cost function is of course the sum of the edge costs $c(e)$ in this path. This problem is NP complete, but there exist efficient **heuristics** to build "good" paths, in particular when the edge costs correspond to Euclidean distances.

- **Maximal coupling.** A coupling in a graph G is a set of edges that have no vertex in common (we "mary" nodes). Finding the largest coupling in G is an NP complete problem.

- **Maximal clique.** Similarly, a clique of G is a subset of nodes that are pairwise connected by edges of E. Finding the largest clique is NP complete as well.

In the above cases, solutions can only be obtained by smartly exploring the domain $\mathcal{D}$ of possible solutions, with a **branch and bound** technique. The latter consists in exploring all possibilities by taking local decisions, arranged under the form of a decision tree. The main idea is to quickly decide that one branch will not lead to an optimum x^*, or at least to an acceptable solution, in order to quickly backtrack and explore a more promising branch. The art is to design heuristics that allow us to estimate reliably that one branch is promising or not.

In some cases, one can approximate a combinatorial problem by a continuous domain problem, to which numerical methods can then be applied. For example the integer linear programming above can be addressed by first dropping the constraint of an interger solution.

There exist cases where the problem is already to find an acceptable solution in $\mathcal{D}$! When $\mathcal{D}$ is defined by a huge collection of constraints, one is already happy if the problem has a solution, *i.e.* if $\mathcal{D}$ is non empty... Finding the "best" one is a secondary objective. These problems are addressed by constraint solving methods, or other ad hoc techniques (temporal logics, for ex.). For example :

- **Puzzles, games.** Like Eternity, sudokus, SAT problems. Or like the association of classrooms, teachers and students in a college to cover the program of each class over the week.

- **Planning problems.** One is given a set of resources, and a set of actions that consume some resources and produce others. Resources can be understood as binary variables (present/absent). The objective is to arrange the actions in order to reach a given global state, *i.e.* the simultaneous presence of a collection of resources. As a secondary objective, one may try to minimize the number of actions.

1.4 A few variations

Optimization multi-criterion. Some optimization problems involve several cost functions and want to minimize at the same time $f_1(x)$ and $f_2(x)$ in the same variable x. Imagine for example doing some shopping with several children to satisfy and a limited budget... A straigtforward solution consists in combining all criteria in a single one, like in $f(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x)$, but this is not always meaningful.

Game problems. Let us divide the parameter vector $x \in \mathbb{R}^d$ into sub-vectors $x^t = [{x_1}^t, ..., {x_K}^t]$ with $x_k \in \mathbb{R}^{d_k}$ and $\sum_k d_k = d$. Instead of globally optimizing x, assume that we have K **players**, and that player k is in charge of tuning the sub-vector x_k of parameters.

The objective can be to jointly minimize the same cost function $f(x)$, in which case we have a **cooperative game**. This can also be considered as a distributed optimization problem. The issue is often to understand what players should know, what they should communicate, when, and how they should behave.

In many cases, one has an **competitive game**. Each player tries independently to minimize its own cost function $f_k(x)$, that depends on the full vector of parameters x, while player k only pilots part of it.

Functional optimization. (also called **variations calculus**). There exists optimization problems where the parameter x to optimze doesn't lie in $\mathbb{R}^d$ but in an infinite dimensional space. Consider for example the question of designing the shape of a toboggan, with fixed height and length, in order to minimize the travel time of a child (the brachistochrone problem). Consider also the determination of the shape of a hanging rope, fixed at its two extremities, which minimizes its potential energy. In both cases, the "parameter" to adjust is a *function*, not a vector. Such problems are more involved mathematically, and won't be covered in these notes.

2 Some problems that can be solved analytically

This section examines some situations where the objective function is relatively simple and allows us to derive an exact expression of its minimum. These cases are frequent... because people state problems in a way they can solve them easily!

2.1 Quadratic forms and vectorial notation

Assume $f : \mathbb{R} \to \mathbb{R}$ is quadratic, *i.e.* $f(x) = ax^2 + bx + c$, then its minimum is easy to obtain: Compute the derivative $f'(x) = 2ax + b$ and solve $f'(x) = 0$, then check that this point $x^* = -\frac{b}{2a}$ is indeed a minimum, not a maximum, which is granted if $a > 0$. Notice that this amounts to putting f under the form $f(x) = a(x - x^*)^2 + c'$. The general case where $x \in \mathbb{R}^d$ can be solved in the same way, up to a few extra technicalities.

$f : \mathbb{R}^d \to \mathbb{R}$ is a **quadratic form** iff it is a polynomial in the coordinates x_i of x, and monomials have order 2 at maximum. For example, assuming $d = 3$, $f(x) = f(x_1, x_2, x_3) = {x_1}^2 + {x_3}^2 + 2x_1x_2 + 2x_2 + x_3 + 1$. Quadratic forms can always be expressed as

$$f(x) \quad = \quad x^t A x + b^t x + c \tag{4}$$

where $A \in \mathbb{R}^{d\times d}$ is a $d \times d$ symmetrical matrix ($A^t = A$), $b \in \mathbb{R}^d$ is a vector and $c \in \mathbb{R}$ a constant. Observe that each term in this sum is a scalar (check matrix dimensions rules). In particular, in this notation $b^t x$ is the **scalar product** of vector b with vector x, since $b^t x = \sum_{i=1}^{d} b_i x_i$. And in particular $x^t x = \sum_i {x_i}^2$ yields the square **norm** of vector x.

If A **is diagonal**, which we denote by $A = \Delta = diag(a_1, ..., a_d)$, f can be minimized very easily: $f(x) - c = \sum_i (a_i\, {x_i}^2 + b_i x_i) = \sum_i f_i(x_i)$ so it suffices to minimize independently the scalar functions f_i. The minimum is unique iff all a_i are positive: f has a "bowl" shape (elliptic paraboloid) when $d = 2$ (Fig. 6, left). The minimum is not unique when at least one of the a_i is negative, since f can go to $-\infty$. For $d = 2$, when the a_i have different signs, f has a saddle shape (hyperbolic paraboloid, Fig. 6, right), or is a reversed bowl when all a_i are negative. In the particular case where one of the $a_i = 0$ for some i, the surface defined by f is a "gutter," with a parabolic profile.

When A is a positive diagonal matrix, observe that one can re-express f as

$$f(x) \quad = \quad (x - x^*)^t A (x - x^*) + c' \tag{5}$$

where $x^* = [x_1^*, ..., x_d^*]^t$ is the vector of optimal values.

In the general case where A **is not diagonal**, the problem is the same... up to a change of coordinates. As a symmetric matrix, A can be diagonalized as

$$A \quad = \quad P \Delta P^t \tag{6}$$

where P is formed by juxtaposing the (column) eigenvectors $p_i \in \mathbb{R}^d$ of A: $Ap_i = \lambda_i p_i$ and $\Delta = diag(\lambda_1, ..., \lambda_d)$ is the diagnonal matrix containing the eigenvalues λ_i

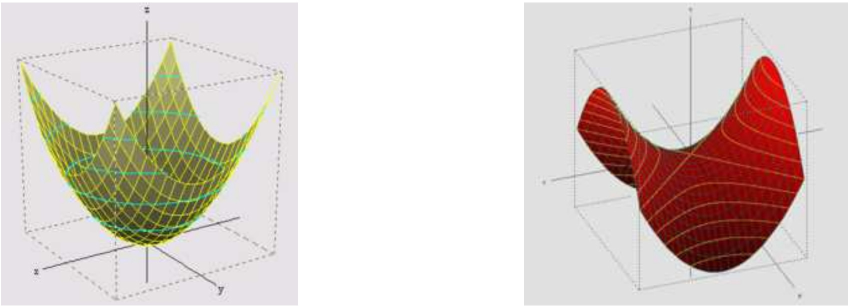

Figure 6: For $d = 2$, two typical shapes quadratic forms. The bowl (left) when all eigenvalues of A are positive, and the saddle (right) when eigenvalues of A have opposite signs.

of A ($\lambda_i \in \mathbb{R}$). The p_i form an orthonormal basis of $\mathbb{R}^d$, *i.e.* ${p_i}^t p_i = 1$ and ${p_i}^t p_j = 0$ for $i \neq j$. So P is a unitary matrix : $PP^t = \mathbb{1}$ and P^t is the inverse of P. P^t must be read as a change of coordinates : one goes from the canonical basis of $\mathbb{R}^d$ to the orthonormal basis formed by the p_i.

If P is available, let us change variables by taking $x = Py$ or $y = P^t x$ (we express x in the new orthonormal basis of the p_i, since $x = Py = \sum_i y_i p_i$, a linear combination of column vectors). One gets

$$f(x) = f(Py) = \bar{f}(y) \quad = \quad y^t P^t P \Delta P^t P y + b^t P y + c = y^t \Delta y + (P^t b)^t y + c \tag{7}$$

Since $\bar{f}(y)$ is diagonal, its minimum y^* is easily computed, and yields the minimum $x^* = Py^*$ for $f(x)$.

In practice, however, diagonalizing A is as complex as inverting it (complexity=d^3). It is much simpler to derive x^* directly. The idea is to derive f on all its coordinates, which yields the **gradient**[2] of f, and to find the point where this gradient vanishes.

The gradient $g(x) = \nabla f(x)$ is a (column) vector $[g_1(x), ..., g_d(x)]^t$ defined by

$$g_i(x) \quad = \quad \frac{\partial f(x)}{\partial x_i} \tag{8}$$

Denoting by $a_{i,j}$ the entries of A, one easily checks that $\frac{\partial f(x)}{\partial x_i} = 2 \sum_j a_{i,j}\, x_j + b_i$, which yields

$$g(x) \quad = \quad 2Ax + b \tag{9}$$

So the (candidate) minimum of f is obtained for

$$x^* \quad = \quad -\frac{1}{2} A^{-1} b \tag{10}$$

when A is invertible[3] (*i.e.* has only non-null eigenvalues). Solving a linear system is much simpler than inverting A (complexity=d^2 instead of d^3). When all eigenvalues of A are positive, x^* is indeed the unique minimum of f. Otherwise, x^* corresponds to the “saddle point” of f (see Fig. 6, right).

[2] More details are given later on the notion of gradient.

[3] We'll see below how to proceed with pseudo inverses when A is not invertible.

2.2 Example 1: Curve fitting

Assume we are given N points $p(1), ..., p(N)$ in $\mathbb{R}^2$, and we want to draw a line that best describes these points. This line $\mathcal{L}$ is defined by a linear equation

$$\mathcal{L}: \quad x_1\, p_1 + x_2\, p_2 + x_3 \quad = \quad 0 \tag{11}$$

where (p_1, p_2) denotes the coordinates of a point in $\mathbb{R}^2$ and $x = [x_1, x_2, x_3]^t$ defines the desired parameters of line $\mathcal{L}$. Observe that the x_i are defined up to a constant, so in reality one of the x_i can be set to 1.

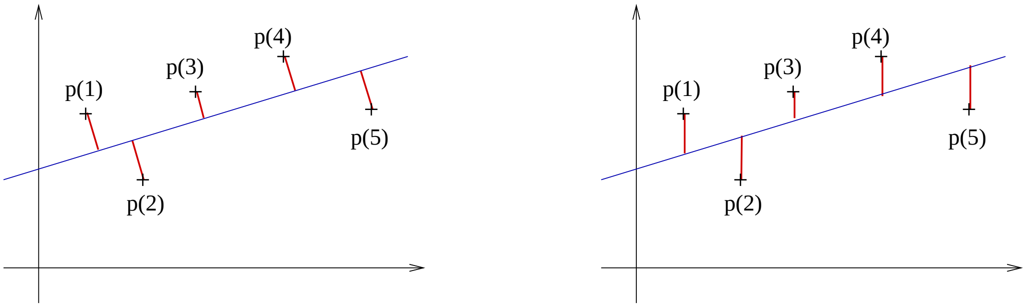

Figure 7: *Two criteria to adjust a line through a cloud of points.*

What criterion should we use ? A first guess would be to minimize the distance of all points $p(n)$ to line $\mathcal{L}$ (Fig. 7, left).

$$\arg\min_x f(x) \quad \text{with} \quad f(x) = \sum_{n=1}^{N} D(p(n), \mathcal{L}) \tag{12}$$

One easily shows that the distance of a point p to $\mathcal{L}$ is given by (exercise)

$$D(p, \mathcal{L}) \quad = \quad \frac{x_1\, p_1 + x_2\, p_2 + x_3}{\sqrt{{x_1}^2 + {x_2}^2}} \tag{13}$$

which is highly non linear in x, and so not easy to minimize... The difficulty remains if we consider the sum of square distances : we still have a quotient of quadratic functions to minimize.

In practice, people prefer to use the “vertical” distance to the line (Fig. 7, right), given by (exercise)

$$D_v(p, \mathcal{L}) \quad = \quad \left| \frac{x_1\, p_1 + x_2\, p_2 + x_3}{x_2} \right| \tag{14}$$

assuming that $x_2 \neq 0$, *i.e.* that the line $\mathcal{L}$ isn't vertical. If we normalize (11) to have $x_2 = 1$, and choose to minimise the sum of these square distances, we get a nice quadratic form in the remaining parameters x_1, x_3.

Exercise 4 *Give the expression of this criterion under the form (4).*

Exercise 5 *How does the criterion change if we choose to set $x_3 = 1$ instead of $x_2 = 1$ in (11)? Is it still quadratic form in x?*

This quadratic criterion is extremely useful in practice. It generalizes very easily :

- For example, one can try to fit a parabola

$$\mathcal{P}: \quad p_2 \quad = \quad x_1\, p_1 + x_2\, {p_1}^2 + x_3 \tag{15}$$

  to a cloud of points. Or any polynomial of p_1 in fact. The criterion remains quadratic in the unknown parameters x.

- In the same way, in dimension $d > 2$ this time, one can try to adjust a hyperplane

$$\mathcal{P}: \quad x_1\, p_1 + x_2\, p_2 + ... + x_d\, p_d + x_{d+1} \quad = \quad 0 \tag{16}$$

  to a cloud of points (where again the x_i are defined up to a constant).

The weakness of this criterion, however, is its high sensitivity to **outliers** : if all points are aligned excepted one that is far away, the latter will deviate the line from the orientation given by all the others, resulting in a meaningless interpolation. This is in fact a bad feature of most curve fitting methods, which therefore must introduce special treatments to detect outliers and correct their effects.

Exercise 6 *Compute the best line that fits points* $p(1) = [-1, 1]^t, p(2) = [0, 0]^t, p(3) = [2, 4]^t$ *in the plane.*

Exercise 7 *Compute the best parabola that fits the same points.*

2.3 Example 2: Best point correspondence between two images

Consider two consecutive images I_1, I_2 in a sequence. They represent the same scene but a camera move took place between I_1 and I_2, so that objects appear shifted in I_2. We want to estimate this apparent move. Assume N characteristic points $p(1), ..., p(N)$ have been identified in I_1, that correspond to points $q(1), ..., q(N)$ in I_2. This association is assumed to be exact (which is rarely the case in practice).

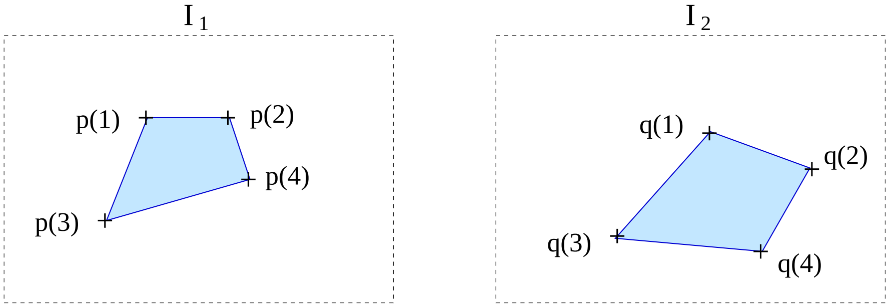

Figure 8: *Matching points between two images.*

We model the transform from I_1 to I_2 as a similitude, a linear transform composed of a rotation, a dilation and a translation. Formally, this map writes

$$q = Hp + T \quad \text{where} \quad H = \begin{bmatrix} x_1 & x_2 \\ -x_2 & x_1 \end{bmatrix} \quad \text{and} \quad T = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \tag{17}$$

T is the translation vector and H combines a rotation and a dilation (by factor $\sqrt{{x_1}^2 + {x_2}^2}$). So we have an optimization problem in $\mathbb{R}^4$.

Once again, we can choose to minimize the square distances between the $q(n)$ and the images by (17) of the $p(n)$, which yields:

$$\arg\min_x f(x) \quad \text{with} \quad f(x) = \sum_n \|Hp(n) + T - q(n)\|^2 \tag{18}$$

which is easily seen to be a quadratic criterion in x.

Exercise 8 *Give the expression of this criterion under the form (4).*

2.4 SVD and pseudo-inverse

Before extending the list of examples, let us introduce two very useful notions.

Consider a matrix $M \in \mathbb{R}^{n \times d}$, where possibly $n \neq d$. M can always be decomposed as

$$M = U \Delta V^t \quad \text{with} \quad U \in \mathbb{R}^{n \times n},\ \Delta \in \mathbb{R}^{n \times d},\ V \in \mathbb{R}^{d \times d} \tag{19}$$

where both U and V are unitary matrices ($U^t U = \mathbb{1}_n$ and $V^t V = \mathbb{1}_d$ [4]), and Δ is a (non-square) diagonal matrix containing the **singular values** of M (see Fig. 9). This **singular value decomposition** (SVD) generalizes the diagonalization of symmetric matrices.

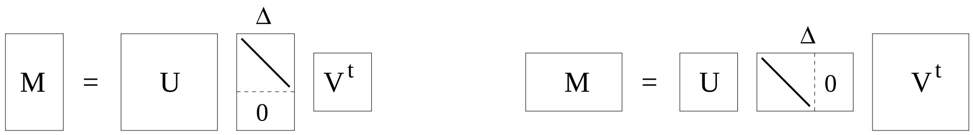

Figure 9: *Illustration of matrix dimensions in the SVD of $M \in \mathbb{R}^{n \times d}$, for $n \geq d$ (left) and $n \leq d$ (right).*

Denoting by $u(i)$ and $v(i)$ the i-th column in U and V respectively, and by λ_i the diagnonal elements of Δ, (19) expresses that vector $v(i)$ is mapped by M to $\lambda_i\, u(i)$:

$$Mv(i) \quad = \quad \lambda_i\, u(i) \tag{20}$$

There is a direct relation between SVD and diagonalization. Observe that MM^t is a (positive semi-definite) symmetric matrix, and thus admits a diagonal form. (19) reveals that

$$MM^t \quad = \quad U \Delta \Delta^t U^t \tag{21}$$

so the $u(i)$ are the eigenvectors of MM^t, with eigenvalues ${\lambda_i}^2$ (all non-negative). And symmetrically the $v(i)$ are the eigenvectors of $M^t M$, with the same eigenvalues. This is in fact how (19) is derived.

[4] $\mathbb{1}_n$ denotes the identity matrix of dimension n. We omit the subscript n when it is obvious.

The SVD provides the clearest way of understanding the notion of **pseudo-inverse**. The pseudo-inverse of $M \in \mathbb{R}^{n\times d}$ is a matrix $M^\dagger \in \mathbb{R}^{d\times n}$ that satisfies

$$\begin{aligned} MM^\dagger M &= M & (22)\\ M^\dagger MM^\dagger &= M^\dagger & (23)\\ (MM^\dagger)^t &= MM^\dagger & (24)\\ (M^\dagger M)^t &= M^\dagger M & (25) \end{aligned}$$

so $M^\dagger$ "almost" behaves as an inverse for M, but $MM^\dagger \neq 1\!\!1_n$ and $M^\dagger M \neq 1\!\!1_d$ (in general). In the case of a diagonal matrix like Δ, the pseudo-inverse $\Delta^\dagger$ is also diagonal, with reversed dimensions. Its diagonal terms are given by

$$\lambda_i^\dagger = \begin{cases} \frac{1}{\lambda_i} & \text{when } \lambda_i \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

For example

$$\Delta = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \Longrightarrow \quad \Delta^\dagger = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{27}$$

The general case is obtained from the SVD:

$$M = U\Delta V^t \quad \Longrightarrow \quad M^\dagger = V\Delta^\dagger U^t \tag{28}$$

and it is easy to check that $M^\dagger$ satisfies the four conditions above (exercise). Moreover, one has the following properties:

$$\begin{aligned} (M^\dagger)^\dagger &= M & & (29)\\ (M^\dagger)^t &= (M^t)^\dagger & & (30)\\ M^\dagger &= M^{-1} & \text{for } M \text{ invertible} & \quad (31)\\ (M^tM)^\dagger M^t &= M^\dagger & & (32)\\ M^tMM^\dagger &= M^t & & (33) \end{aligned}$$

Exercise 9 *Prove the above relations, using the SVD of M when necessary.*

2.5 Example 3: Best solution to a linear system

Back to optimization problems. In many situations, one has to solve a linear system

$$Mx = y \tag{34}$$

in the unknown vector $x \in \mathbb{R}^d$, and with n equations: $M \in \mathbb{R}^{n\times d}$, $y \in \mathbb{R}^n$. When $n < d$, this system has an infinite number of solutions, and one may want to find the one with minimal norm, which looks like an optimization problem under the constraint (34). But when $n > d$, the system can be overconstrained and may not have exact solutions.

Both cases are captured by the following approach, called a **least squares** estimation. Let us look for the best match of y with Mx, *i.e.* let us minimize the norm of the error $e = y - Mx$:

$$\arg\min_x f(x) \quad \text{with} \quad f(x) = \|Mx - y\|^2 = (Mx - y)^t(Mx - y) \tag{35}$$

This quadratic criterion can of course be put under the form of (4) : $A = M^tM$ is a $d \times d$ positive matrix, and $b = -2M^ty$, $c = y^ty$. The definition of A imposes non-negative eigenvalues, so f has either a bowl shape or a gutter shape. Its minima are characterized by a vanishing gradient $g(x) = 2Ax + b = 0$, which corresponds to

$$M^tMx \;=\; M^ty \tag{36}$$

If M has full rank ($Rank(M) = Rank(M^tM) = d$), then M^tM is invertible, and there is a *unique* minimum to f given by

$$x^* \;=\; (M^tM)^{-1}M^ty \;=\; M^\dagger y \tag{37}$$

In the general case, $Rank(M) < d$ and so M^tM is *not* invertible (gutter case), and there exist an infinite number of minima to f. To characterize them, let us plug in (36) the SVD $M = U\Delta V^t$. One gets

$$\begin{aligned} (V\Delta^tU^t)(U\Delta V^t)x &= (V\Delta^tU^t)y \\ \Delta^t\Delta(V^tx) &= \Delta^t(U^ty) \\ \Delta^t\Delta\bar{x} &= \Delta^t\bar{y} \end{aligned} \tag{38}$$

so it suffices to solve the last equation in the new variable $\bar{x} = V^tx$ (with $\bar{y} = U^ty$). Choosing

$$\bar{x}^* = \Delta^\dagger\bar{y} \quad \text{or equivalently} \quad x^* = M^\dagger y \tag{39}$$

gives the solution to (38) with minimal norm $\bar{x}^t\bar{x}$, and so the solution to (36) with minimal norm (recall that as a unitary matrix, V^t preserves the norm). Observe that $\bar{x}^*_i = 0$ whenever $\lambda_i = 0$ in the diagonal matrix Δ. All other solutions are obtained by setting these components $\bar{x}^*_i$ to any value (which of course augments the norm). This corresponds to adding to x^* any linear combination of the column vectors $v(i)$ of V for which $\lambda_i = 0$. All these values of x^* define the "bottom of the gutter."

Notice that the SVD and the pseudo-inverses give insight on this minimization problem, but shouldn't be used as such for large dimension problems : they have the same complexity as matrix diagonalization or matrix inversion. In practice, one will solve (36) by triangulation, *i.e.* using an LU decomposition or Choleski decomposition of A. The latter are easy to build by successive products of A with suitable Householder matrices (see [1]).

As we have seen, it is very convenient to minimize quadratic forms. One may prefer other criteria, for example minimize the L_1-norm of the error vector e, *i.e.* $\sum_i |e_i|$, instead of the L_2-norm. The L_1-norm better favors sparse vectors, *i.e.* minimizes the number of non-null entries, or equivalently maximizes the number of linear equations that are satisfied. This may be a desirable feature, but it imposes more complex optimization techniques...

2.6 Example 4: Intersection of N lines

A typical application of the previous section. Consider N lines in $\mathbb{R}^2$, given by

$$\mathcal{L}(n): \quad a_{n,1}\, x_1 + a_{n,2}\, x_2 = a_{n,0} \qquad\qquad 1 \leq n \leq N \tag{40}$$

We want to compute their intersection point x. There is little chance that it exists... but we may want to find the best point that matches these relations (Fig. 10). A standard problem in navigation, implemented in all GPS receivers.

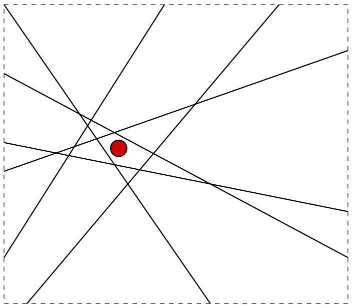

Figure 10: *Best intersection of N lines in $\mathbb{R}^2$.*

The problem of course generalizes directly to the intersection of N hyperplanes in dimension d.

2.7 Example 5: Estimation in presence of random noise

We consider the case where an unknown parameter $x \in \mathbb{R}^d$ is observed through a linear equation perturbed by a random (Gaussian) noise:

$$Y \;=\; Hx + V \quad \text{ where } \quad H \in \mathbb{R}^{n\times d},\; V \sim \mathcal{N}(0, R) \tag{41}$$

The observation noise V, a random vector in $\mathbb{R}^n$, follows a centered Gaussian law with covariance matrix $R \in \mathbb{R}^{n\times n}$. If we observe/measure the value y for Y, what is the most likely value of x?

2.7.1 Minimal background on Gaussian vectors

$U \in \mathbb{R}^n$ is a Gaussian vector of mean $m_u \in \mathbb{R}^n$ and covariance matrix $P_U \in \mathbb{R}^{n\times n}$, denoted by $U \sim \mathcal{N}(m_U, P_U)$, iff its probability density is given by

$$p(u) \;=\; \frac{1}{(2\pi)^{n/2}(\det P_u)^{1/2}}\; \exp[\, -\frac{1}{2}(u - m_U)^t {P_U}^{-1}(u - m_U)\,] \tag{42}$$

One has the following remarkable expected values

$$\begin{aligned}
\mathbb{E}(1) &= \int_{\mathbb{R}^n} p(u)du \;=\; 1 && (43)\\
\mathbb{E}(U) &= \int_{\mathbb{R}^n} u\, p(u)du \;=\; m_u && (44)\\
\mathbb{E}[\,(U-m_U)(U-m_U)^t\,] &= \int_{\mathbb{R}^n} (u-m_U)(u-m_U)^t\, p(u)du \;=\; P_U && (45)\\
\mathbb{E}(\|U-m_U\|^2) &= \int_{\mathbb{R}^n} (u-m_U)^t(u-m_U)\, p(u)du \;=\; Tr(P_U) && (46)
\end{aligned}$$

where $Tr(P_U)$ is the *trace* of matrix P_U, *i.e.* the sum of its diagonal entries. Assume we want to estimate U with a constant $c \in \mathbb{R}^n$, with the following criterion

$$\arg\min_c \; \mathbb{E}(\,\|U-c\|^2\,) \tag{47}$$

The best c is called the **minimum mean square estimate** (MMSE) of the random vector U.

Exercise 10 *Prove that $c^* = m_U$.*
Hint: show that $\mathbb{E}[\,(U-c)^t(U-c)\,] = Tr(P_U) - 2\,{m_U}^t c + c^t c$.

Exercise 11 *Prove that the most likely value of U,* i.e. $\arg\max_u p(u)$*, is also given by $u^* = m_U$.*

So, for a Gaussian vector U, its MMSE estimate as well are its maximum likelihood estimate are both equal to its mean m_U.

2.7.2 Maximum likelihood (ML) estimate of x

Back to (41). When Y is fixed to its observed value y, what is the most likely value of x? More precisely, what is the most likely value of the (non observed) measurement noive V? It can be determined by adjusting x in order to maximize the likelihood $p(v)$ for $v = y - Hx$. In other words,

$$\begin{aligned}
x^* &= \arg\min_x \; \|y-Hx\|^2_{R^{-1}}\\
&= \arg\min_x \; (y-Hx)^t R^{-1}(y-Hx) && (48)
\end{aligned}$$

This is of course a quadratic program.

Exercise 12 *Prove that*

$$x^* \;=\; (H^t R^{-1} H)^\dagger H^t R^{-1} y \tag{49}$$

Compe this expression with (39), taking into account (32), and observe that it assigns different weights to the components of the measured vector y, to account for the different noise levels. Imagine for example that R is diagonal: coordinates of y perturbed by a high level of noise are damped in proportion, so the estimate of x relies less on them. When R is proportional to the identity, it disappears from (49).

2.7.3 Maximum *a posteriori* (MAP) estimate of X

In some cases, X is itself a random vector, $X \sim \mathcal{N}(m_X, P_X)$, and the measurement $Y = HX + V$, as a linear combination of Gaussian vectors, remains a Gaussian vector. It is generally assumed that X and V are independent, which is equivalent to decorrelation in the Gaussian case, so $\mathbb{E}(XV^t) = 0$.

Exercise 13 *Show the relations*

$$\begin{aligned} m_Y &\triangleq \mathbb{E}(Y) \;=\; H\,m_X && (50)\\ P_{X,Y} &\triangleq \mathbb{E}[(X-m_X)(Y-m_Y)^t] \;=\; P_X H^t && (51)\\ P_Y &\triangleq \mathbb{E}[(Y-m_Y)(Y-m_Y)^t] \;=\; HP_XH^t + R && (52)\end{aligned}$$

Hint: use (44,45) and the independence of X and V.

The MAP estimate of X consists in computing

$$\begin{aligned} \hat{X}(y) &= \arg\max_x p(x|Y=y)\\ &= \arg\max_x p(x|Y=y)p(Y=y)\\ &= \arg\max_x p(x,y) && (53)\end{aligned}$$

which is a function of the observation y. Notice that optimizing the conditional density $p(x|Y=y)$ or the joint density $p(x,y)$ gives the same estimate, since y is fixed[5].

This joint density is obtained by noting the independence of X and V, so $p(x,v) = p(x)p(v)$, and operating the change of variables $V = Y - HX$, so

$$p(x,y) \;\propto\; \exp[\,-\frac{1}{2}(x-m_X)^tP_X^{-1}(x-m_X) - \frac{1}{2}(y-Hx)^tR^{-1}(y-Hx)\,] \quad (54)$$

and its maximum in x for a fixed y is again a quadratic program with

$$f(x) \;=\; (x-m_X)^tP_X^{-1}(x-m_X) + (y-Hx)^tR^{-1}(y-Hx) \quad (55)$$

Exercise 14 *Prove that the minimum is given by*

$$\hat{X}(y) - m_X \;=\; (P_X^{-1} + H^tR^{-1}H)^{-1}H^tR^{-1}(y - m_Y) \quad (56)$$

Observe that this relation is very similar to (49) when $m_X = 0$: some extra ponderation by P_X has been added to account for the *a priori* knowledge one has on the components of X.

$\hat{X}$, as a linear function of the Gaussian variable Y, and so is a Gaussian variable itself (when y varies). Its mean is given by m_X, and its variance by the quite complex expression $P_{\hat{X}} = (P_X^{-1} + H^tR^{-1}H)^{-1}H^tR^{-1}P_YR^{-1}H(P_X^{-1} + H^tR^{-1}H)^{-1}$, that we derive from (56).

These relations can be derived in a different way, using a minimum mean square estimation. Assume for simplicity that $d = 1$, so X is a scalar Gaussian random

[5] Recall Bayes relation, that defines the conditional density: $p(x|y) = p(x,y)/p(y)$.

variable, and that X and Y have zero mean. Let us build an estimate $\hat{X}$ of X as a *linear combination* of components of Y, $\hat{X} = m^t Y$, $m \in \mathbb{R}^n$, in order to minimize the variance of the estimation error $\tilde{X} = X - \hat{X}$:

$$\begin{aligned} m^* &= \arg\min_m \mathbb{E}[(X - m^t Y)^2] \\ &= \arg\min_m \mathbb{E}[(X - m^t Y)(X - Y^t m)] \\ &= \arg\min_m m^t P_Y m - 2 P_{X,Y} m + P_X \end{aligned} \tag{57}$$

Again this is a quadratic program in vector m, and the optimum is given by

$$(m^t)^* = P_{X,Y} P_Y^{-1} = P_X H^t (H P_X H^t + R)^{-1} \tag{58}$$

Back to the general case, where X is a non centered random vector. Doing this estimation for each coordinate of X yields

$$\hat{X} - m_X = P_{X,Y} P_Y^{-1}(Y - m_Y) = P_X H^t (H P_X H^t + R)^{-1}(Y - m_Y) \tag{59}$$

The variance of $\hat{X}$ is thus $P_{\hat{X}} = P_X H^t (H P_X H^t + R)^{-1} H P_X$. The estimation error $\tilde{X} = X - \hat{X}$ is thus centered with variance $P_{\tilde{X}} = P_X - P_{\hat{X}} = P_X - P_X H^t P_Y^{-1} H P_X$.

There are two expressions for $P_{\hat{X}}$ and thus for $P_{\tilde{X}}$: a complex one derived from (56), and another derived from (59). They can be shown to be identical using a matrix inversion lemma for $A + CBC^t$.

3 Optimization in $\mathbb{R}^d$

Notations. We now need to handle series of vectors $x \in \mathbb{R}^d$. We denote them with superscripts $x^0, x^1, x^2, ..., x^n, ...$ in order to avoid the confusion with $x_1, x_2, ..., x_d$ that represent the coordinates of x.

3.1 Taylor expansion

Consider a function $f : \mathbb{R} \to \mathbb{R}$ that is C^2, *i.e.* that can be differentiated twice, with continuous derivatives. Around any point[6] $x^0 \in \mathbb{R}$, one can write the second order *Taylor expansion* of f

$$f(x) \quad = \quad f(x^0) + f'(x^0)(x - x^0) + \frac{1}{2}f''(x^0)(x - x^0)^2 + o(x - x^0)^2 \qquad (60)$$

where the displacement $x - x^0$ is small ($|x - x^0| \leq \epsilon$), and where $o(x - x^0)^2$ denotes an error term that is negligeable compared to $(x - x^0)^2$, when x gets closer to x^0. This expression approximates f by a second order polynomial around x^0, *i.e.* by a parabola (Fig. 11).

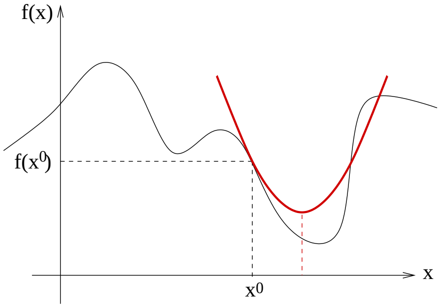

Figure 11: *Second order approximation of f around x^0.*

For a function $f : \mathbb{R}^d \to \mathbb{R}$, there exists a similar expression :

$$\begin{aligned} f(x) \quad &= \quad f(x^0) + \nabla f(x^0)^t \, (x - x^0) \\ &\quad + \frac{1}{2}(x - x^0)^t \, \nabla^2 f(x^0) \, (x - x^0) + o(\|x - x^0\|^2) \end{aligned} \qquad (61)$$

where x is the column vector $[x_1, ..., x_d]^t$ (and $x^0 = [x^0_1, ..., x^0_d]^t$). This formula deserves several comments.

- In the first order term, $\nabla f(x^0)$ denotes the **gradient**[7] of f at point x^0, *i.e.*

[6] To avoid heavy formulae, we adopt notation x^0 as well as $(x^n)_{n\geq 0}$ and $(d^n)_{n\geq 0}$ for series of vectors. This superscript is not the "power" of the vector ! Also, mind the difference between the vectors d^n that we use in the sequel, and the dimension d of $x \in \mathbb{R}^d$. In general, the context will resolve any ambiguity.

[7] Some authors simply write $f'(x^0)$ or $\frac{\partial f(x^0)}{\partial x}$. We prefer to avoid these motations to stress the vector nature of the gradient.

the (column) vector

$$\nabla f(x^0) \quad = \quad \begin{bmatrix} \vdots \\ \frac{\partial f(x^0)}{\partial x_i} \\ \vdots \end{bmatrix} \tag{62}$$

The expression $\nabla f(x^0)^t \, (x - x^0)$ is thus the scalar product of the gradient by the displacement $x - x^0$.

- In the second order term, we denote by $\nabla^2 f(x^0)$ the **Hessian** of f at point x^0, *i.e.* the symmetric $d \times d$ matrix which entries are the second order derivatives of f

$$\nabla^2 f(x^0) \quad = \quad \begin{bmatrix} & \vdots & \\ \cdots & \frac{\partial^2 f(x^0)}{\partial x_i \partial x_j} & \cdots \\ & \vdots & \end{bmatrix} \tag{63}$$

  The term $(x - x^0)^t \, \nabla^2 f(x^0) \, (x - x^0)$ thus returns a scalar.

- The last term means that the difference between f and its approximation is negligeable (in norm) compared to $\|x - x^0\|^2$, when x goes to x^0.

(61) gives the quadratic form that best approximates f around x^0. Observe that when $f(x)$ is simply a sum of functions $f_i : \mathbb{R} \to \mathbb{R}$ that only depend on the coordinate x_i of x, then (61) is simply the sum of the Taylor expansions (60) of each f_i around x_i^0.

To illustrate (61), let us assume $d = 2$ (Fig. 12). f can thus be represented as a surface in 3D, or by means of **level lines** in 2D, like the lines of equal altitude on a geographical map, on like the lines of equal pressure on a meteorological map.

Definition 1 *The* **level line** *at point x^0 is the set of points x such that $f(x) = f(x^0)$.*

Consider the first order Taylor expansion of f at x^0

$$f(x) \quad = \quad f(x^0) + \nabla f(x^0)^t \, (x - x^0) + o(\|x - x^0\|) \tag{64}$$

A point x satisfies $f(x) = f(x^0)$ iff $\nabla f(x^0)^t \, (x - x^0) = 0$, *i.e.* iff the displacement $x - x^0$ is orthogonal to the gradient $\nabla f(x^0)$ of f at x^0. Equivalently, *at each point the level line is perpendicular to the gradient.* Assume now that we want to make a small step around x^0, with $\|x - x^0\| = \epsilon$, in order to maximize $f(x)$. What direction should we take ? Again, since we have to maximize the scalar product $\nabla f(x^0)^t \, (x - x^0)$, we should make a step in the direction of the gradient : $x - x^0 = \epsilon \; \nabla f(x^0) \,/\, \|\nabla f(x^0)\|$. In other words, *the gradient indicates the direction of the maximal (increasing) slope of f at x^0.* Recall that the closest are the level lines, the largest is the norm of the gradient.

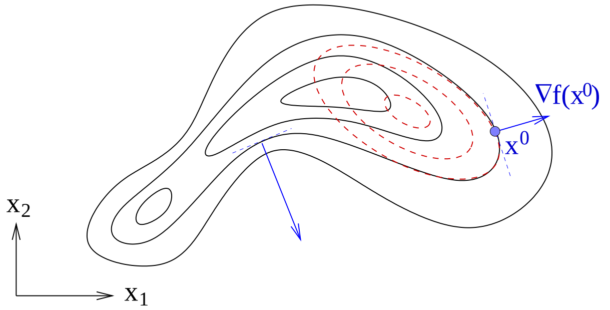

Figure 12: *Level lines of $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, and position of the gradient. In red dashed lines, level lines of the best approximation of f by a quadratic form around x^0.*

The second order Taylor expansion of f at x^0 can also be illustrated by its level lines. As a quadratic form, they are concentric ellipses if the Hessian of f has positive eigenvalues (see the previous chapter). The ellipse that goes through x^0 is tangent to the level line of f.

Exercise 15 *Consider the function $g(x) = f(x^0) + \nabla f(x^0)^t\,(x - x^0)$, obtained from first order Taylor expansion of $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. How do its level lines look like ?*

Exercise 16 *Let q be the quadratic form that best approximates f at x^0. Consider the level lines of f and q at x^0. Prove that they are tangent, and that they even have identical curvatures.*

Exercise 17 *Compute the 2nd order Taylor expansion of $\phi(t) = f(x^0 + tu)$ where t is a scalar and u a unit vector in $\mathbb{R}^d$. What does the 3rd order expansion look like ?*

Coming back to the general case where $d \in \mathbb{N}$, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ can be understood as defining a hyper-surface[8] in $\mathbb{R}^{d+1}$: the first d coordinates are given by x, the last one by $z = f(x)$. The first order Taylor expansion (64) allows us to define the notion of **tangent hyperplane** to this hyper-surface at point (x^0, z^0) with $z^0 = f(x^0)$. This hyperplane is simply defined by

$$\mathcal{H} \; : \qquad z \;=\; f(x^0) + \nabla f(x^0)^t\,(x - x^0) \tag{65}$$

This equation also writes $(f(x^0) - z) + \nabla f(x^0)^t\,(x - x^0) = 0$ which reveals the the tangent hyperplane is defined as the plane containing $(x^0, f(x^0))$ and perpendicular to vector $[-1, \nabla f(x^0)^t]^t$.

Exercise 18 *Assume $d = 1$ and make a figure explaining this (f defines a curve in $\mathbb{R}^2$, and $\mathcal{H}$ is a tangent line). Assume $d = 2$ and do the same (f defines a surface in $\mathbb{R}^3$, and $\mathcal{H}$ is a tangent plane).*

[8] the exact term is **differentiable manifold** in $\mathbb{R}^{d+1}$.

3.2 Optimality conditions

Here we assume f is C^2.

Definition 2 *x^0 is* **stationary point** *of f iff $\nabla f(x^0) = 0$.*

This is a necessary condition for x^0 to be a minimum of f. If this is not true, a small step around x^0 in the direction opposite to the gradient allows us to decrease the value of f, so x^0 is clearly not an optimum.

Definition 3 *A stationary point x^0 of f is a* **local minimum** *of f iff there exists an open neighborhood $N(x^0) \subseteq \mathbb{R}^d$ around x^0 such that $\forall x \in \mathcal{N}(x^0)$, $f(x^0) \leq f(x)$.*

Lemma 1 *(Necessary condition) If x^0 is a local minimum of f, then $\nabla^2 f(x^0)$ is a semi-definite positive matrix (all eigenvalues are non-negative : $\lambda_i \geq 0$).*
(Sufficient condition) If $\nabla^2 f(x^0)$ is a positive matrix (all eigenvalues positive : $\lambda_i > 0$) then x^0 is a local minimum of f.

Proof. At a stationary point, the first order term of the Taylor expansion vanishes, so the second order term becomes dominant. If $\nabla^2 f(x^0)$ is a positive matrix, this second order term is always positive, whatever the direction of the small displacement around x^0. If $\nabla^2 f(x^0)$ has a negative eigenvalue, then a step in the direction of the corresponding eigenvector decreases f (see the saddle point situation in Fig. 6). If $\nabla^2 f(x^0)$ has a null eigenvalue, then, in the direction of the corresponding eigenvector, the third order term of the Taylor expansion becomes dominant, which allows us to decrease f if it is non null. But if this term vanishes, the fourth order term becomes dominant, etc. So one need to check possibly many derivatives of f to conclude. □

For convex functions, these necessary and sufficient conditions simplify.

Lemma 2 *$f \in C^1$ is convex iff*

$$\forall x, x^0, \quad f(x) \quad \geq \quad f(x^0) + \nabla f(x^0)^t (x - x^0) \tag{66}$$

Let $f \in C^2$, if $\nabla^2 f$ is semi-definite positive at all point, then f is convex.

Theorem 1 *For a convex function, any local minimun is a global minimum.*
If f is a C^1 convex function, then every stationary point is a global minimum.

Exercise 19 *Prove the lemma and the theorem above.*

Exercise 20 *Let f be a convex function in C^1. Show that its global minima form a (closed) convex set.*

3.3 Optimization scheme

Numerical optimization methods all proceed in the same way. The idea is to start from an *initial guess* x^0, that forms a reasonable approximation of the minimum, and to progressively refine it by local search around it. We therefore build a series x^n that hopefully converges to an optimum x^*. Specifically

1. Initialization : by x^0, an initial guess of a minimum x^*.
2. Recursion : until a convergence criterion is satisfied at x^n
   - From the current value x^n, determine of a search direction $d^n \in \mathbb{R}^d$.
   - Linear search along the semi-line $x^n + t\, d^n,\ \ t \in \mathbb{R}^+$, to determine x^{n+1}. This amounts to minimizing $\phi(t) = f(x^n + t\, d^n)$ in $t > 0$.

Different stop criteria can be used. First of all, we must be at (or close to) a stationary point (in practice $\|\nabla f(x^n)\| \leq 10^{-6}\|\nabla f(x^0)\|$). Then we must check that this stationary point is indeed a minimum, and not a saddle point. For example by checking the Hessian. Notice that in practice iterations are also stopped when the steps become negligeable ($\|x^{n+1} - x^n\| \leq 10^{-6}\|x^n\|$).

The research methods differ by the information they use from f. Either only $f(x)$ is accessible, or the gradient $\nabla f(x)$ is also known, or the gradient and the Hessian are known. According to what is available, it may be necessary to estimate the missing elements. For example to estimate the gradient by taking values of f around x.

3.4 Optimization in $\mathbb{R}$: linear search

Finding a minimum of $f : \mathbb{R} \to \mathbb{R}$ may look simple since we are used to plot such real functions, which gives a global view of them in one shot. The problem is different in practice, because knowing the value of f at many points x is expensive : optimization methods that require many values rapidly become unaffordable in complexity. The right picture is rather that one knows values (or even derivatives) of f at a few points, and has to decide where to look for more information on f in order to corner the minimum.

Dichotomy. Here we assume that f is convex on some interval $[a, b]$ where we look for a minimum, or at least unimodular[9]. The idea is to divide the interval by 2 at each step, and remove the part that doesn't contain the minimum.

Let's take $x^1 = a, x^2 = b$ and $x^3 = (x^1 + x^2)/2$, and compare the values of f at these points. Three cases can occur :

1. if $f(x^1) > f(x^3) > f(x^2)$, the minimum is necessarily in $[x^3, x^2]$,
2. if $f(x^1) < f(x^3) < f(x^2)$, the minimum is necessarily in $[x^1, x^3]$,
3. otherwise we can't conclude.

[9] Unimodularity : let x^* be a minimum of f on $[a, b]$, if $x^1 < x^2 < x^*$ then $f(x^1) \geq f(x^2) \geq f(x^*)$, and if $x^* < x^1 < x^2$ then $f(x^*) \leq f(x^1) \leq f(x^1)$.

In the first two cases, we can repeat the procedure with the selected interval, twice as small (Fig. 13, left). In the last case(Fig. 13, right), let us introduce $x^4 = (x^1+x^3)/2$ and $x^5 = (x^3 + x^2)/2$. The values of f at these new points allow us to select, once again, an interval that is twice as small and contains the minimum (exercise ; consider the 3 possible subcases).

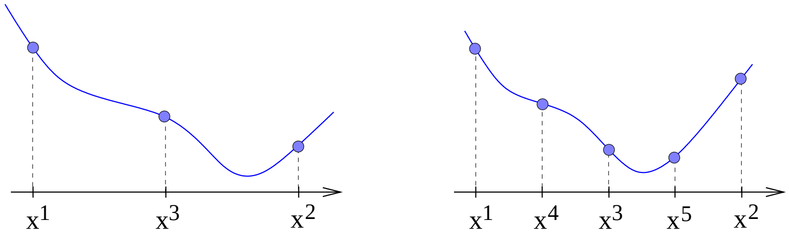

Figure 13: *Dichotomic search : cases 1 (left) and 3 (right) for a unimodular function.*

Newton-Raphson method. This method assumes that f is C^2. Considering its second order Taylor expansion (60) around x^0, the idea is to take x^1 as the minimum of the quadratic form (*i.e.* the parabola) that best fits f at x^0. And to repeat the procedure from x^1.

Equivalently, the method consists in progressively finding a zero of f'. Considering the first order Taylor expansion of f' at x^0,

$$f'(x) \quad = \quad f'(x^0) + f''(x^0)(x - x^0) + o(x - x^0) \tag{67}$$

this amounts to taking as x^1 the zero of this linear form, given by

$$x^1 \quad = \quad x^0 - \frac{f'(x^0)}{f''(x^0)} \tag{68}$$

and to repeat the procedure from x^1, as illustrated in Fig. 14.

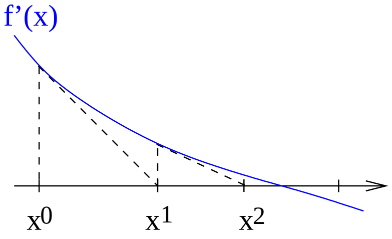

Figure 14: *Newton search of a zero for $\phi'(x)$.*

Secant method. The principle is the same as above, excepted that the second derivatives of f are not available. So we replace f'' by an approximation of it. Specifically, knowing two points x^0 and x^1, we replace $f''(x^1)$ by $\frac{f'(x^1)-f'(x^0)}{x^1-x^0}$. This allows us to compute x^2 as above :

$$x^2 \quad = \quad x^1 - \frac{x^1 - x^0}{f'(x^1) - f'(x^0)} f'(x^1) \tag{69}$$

and we proceed similarly to compute x^3 from x^2 and x^1, etc.

Exercise 21 *Explain this recursion on a plot of f, as in the previous figures.*

Wolfe's method. For functions defined over $\mathbb{R}^d$, it may not be efficient to spend a lot of energy to perform an exact linear search before changing search direction. In practice, it is more efficient to stop when f has "reasonably" decreased in the current search direction.

Wolfe proposed the following method. Let $0 < m_1 < \frac{1}{2} < m_2 < 1$ be two parameters, and, assuming $f'(x^0) < 0$, let $[a = x^0, b]$ the search interval. The point $x \in]a, b[$ is an acceptable estimate of the minimum of f if it satisfies

$$f(x) \leq f(x^0) + m_1(x - x^0)f'(x_0) \tag{70}$$
$$f'(x) \geq m_2 f'(x^0) \tag{71}$$

The second condition imposes to decrease the derivative sufficiently, and the first one prevents from going too far (Fig. 15). When the first condition is violated (x is too far), we simply take $b = x$ and restart the procedure. And symmetrically when the second is violated (x is not far enough), we do $a = x$ and restart.

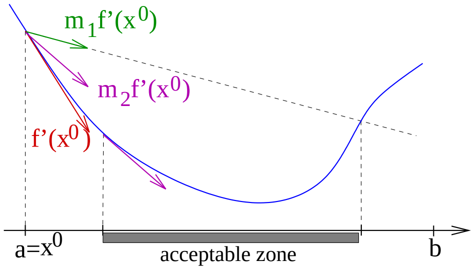

Figure 15: *Acceptable zone for Wolfe's linear search.*

3.5 Gradient descent

Also called **steepest descent**. This is a **first order method** (f and ∇f known). The idea is to progress in the direction of the maximum slope of f at x^n :

$$d^n = -\nabla f(x^n) \tag{72}$$

as descent direction. The **optimal step**[10] $t(n)$ is such that it minimizes in $t \in \mathbb{R}^+$ the function

$$\phi(t) = f(x^n + t\, d^n) \quad \text{so} \quad \phi'(t) = \nabla f(x^n + t\, d^n)^t\, d^n \tag{73}$$

The optimal step $t(n)$ leads to a new point x^{n+1} such that $\nabla f(x^{n+1})^t d^n = 0$, so the new search direction d^{n+1} will be perpendicular to the previous one (Fig. 16).

This method is *simple*, but *very slow*, even for a simple quadratic form. Is is penalized by badly conditioned Hessians. The conditioning number $r = |\lambda_1/\lambda_d|$ of

[10] Here we write $t(n)$ and not t^n to avoid confusion with t to the power n, since t is a scalar.

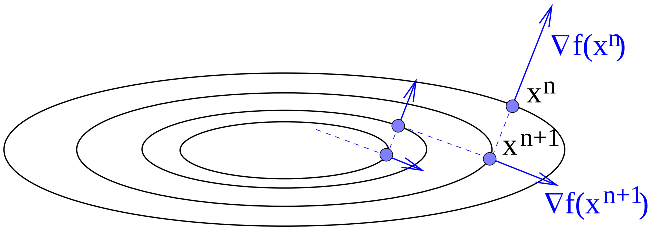

Figure 16: *Gradient search, or steepest descent, with optimal steps.*

a symmetric matrix is the ratio of the largest eigenvalue by the smallest. A high value of r means very flat ellipses as level lines, and so numerous steps to reach the minimum.

In practice, the **linear search** that determines $t(n)$ doesn't look for the optimal step, which can be costly, but rather considers a step that "reasonably" decreases f. A variant consists in choosing a fix step t at each iteration. In both cases, there exist convergence result when f is strongly convex.

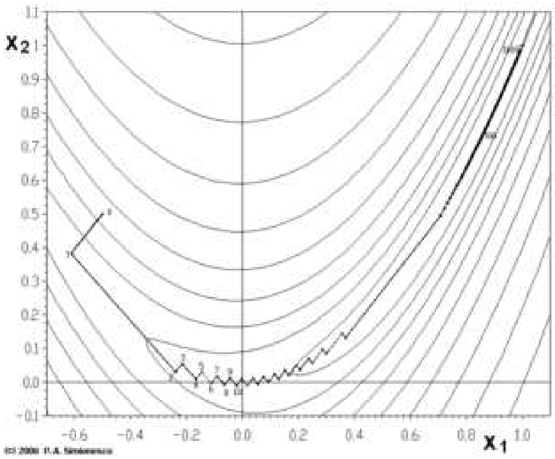

Figure 17: *Gradient search on* $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$*, function known as Rosenbrock's banana. Its unique global minimum is* $x_1 = x_2 = 1$*.*

3.6 Newton method

This is a **second order method** (f, ∇f and $\nabla^2 f$ known). The idea is to consider the 2nd order Taylor expansion of f and to jump directly to the/a stationary point of this quadratic form (*i.e.* to the bottom of the paraboloid in Fig. 12). Specifically, from (61) one has

$$\begin{aligned}
\phi(x) &= f(x^n) + \nabla f(x^n)^t\,(x - x^n) \\
&\quad + \frac{1}{2}(x - x^n)^t\,\nabla^2 f(x^n)\,(x - x^n) && (74) \\
\nabla\phi(x) &= \nabla f(x^n) + \nabla^2 f(x^n)\,(x - x^n) && (75)
\end{aligned}$$

so cancelling the gradient of ϕ suggests to take x^{n+1} such that

$$\nabla^2 f(x^n)\,(x^{n+1} - x^n) \;\;=\;\; -\nabla f(x^n) \tag{76}$$

Notice that we didn't invert the Hessian. First of all, it may not be invertible. But even if it is, we simply have a linear system to solve, which is less expensive than a matrix inversion. The resolution is often performed *via* a *Choleski factorization* of the Hessian.

Comments :

- In general the method is faster, since the search direction is better chosen. Convergence in one step for a quadratic form (compared to the slow convergence of the gradient descent).
- The Newton method tries to cancel the gradient of $\phi(x)$, and so looks for a *stationary point* of f. The latter may very well be a saddle point, or a maximum (!), so one still has to check that the method yields a minimum. In particular, there is no guarantee that the Hessian of f is always a semi-definite positive matrix.
- There is no guarantee that the new point x^{n+1} is better than x^n, or even that $-(\nabla^2 f)^{-1}\nabla f$ is a descent direction... Some authors suggest to take this search direction, when it is valid, and to perform a linear search along it.
- When the Hessian of f is not positive (or ill-conditioned), the Levenberg-Marquardt technique consists in slightly augmenting all eigenvalues of $\nabla^2 f(x^n)$ to get this positivity :

  $$[\nabla^2 f(x^n) + \mu \mathbb{1}]\,(x^{n+1} - x^n) \;\;=\;\; -\nabla f(x^n) \tag{77}$$

  The corrective term $\mu \mathbb{1}$ (with $\mu > 0$) operates as a *regularisation term*, just like P_X^{-1} in (56) or R in (58). It reduces the conditioning number of $\nabla^2 f$, and thus improves the numerical stability of the linear system resolution. Graphically, this amounts to slightly bending up all slopes of $\phi(x)$.

3.7 Conjugate gradient

This first order method is a slight (but smart !) variation on the gradient method. The latter is very slow even on simple functions like quadratic forms, which are extremely important in practice because of (61). The explanation lies in a bad choice of the descent directions when the matrix A of the quadratic form (or the Hessian of f) is badly conditioned (see Fig. 16). In order to better point to the direction of the minimum, the idea is to slightly tilt the descent direction with respect to the gradient, in order to better aim at the bottom of the paraboloid (Fig. 18).

We want to minimize

$$f(x) \;=\; \frac{1}{2}x^t A x + b^t x \quad \text{with} \quad \nabla f(x) \;=\; Ax + b \tag{78}$$

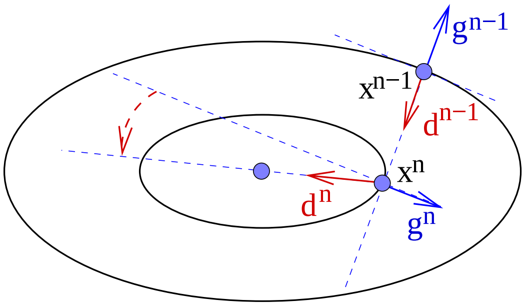

Figure 18: *Conjugate gradient search: deviates from the steepest slope to better aim at the optimum of f.*

where $A \in \mathbb{R}^{d \times d}$ is a symmetric positive matrix and $b \in \mathbb{R}^d$. This amounts to determining x^* such that $Ax^* + b = 0$.

Assume that we are running a gradient descent algorithm, starting from x^0 and with initial search direction $d^0 = -g^0 \triangleq -\nabla f(x^0)$. At point x^n, instead of looking for the optimum x^{n+1} along $d^n = -g^n \triangleq -\nabla f(x^n)$, we look for the optimum in the whole affine space

$$\mathcal{W}_{n+1} \quad = \quad x^0 + sp\{d^0, d^1, ..., d^{n-1}, g^n\} \tag{79}$$

defined by all previous descent directions plus the last gradient, and containing x^0 (and by definition containing also $x^1, ..., x^n$).

Lemma 3 *x^{n+1} is the minimum of f in $\mathcal{W}_{n+1}$ implies g^{n+1} is orthogonal to $\mathcal{W}_{n+1}$.*

Proof. Consider function $h(a) = f(x^{n+1}+a_0 d^0+a_1 d^1+...+a_{n-1}d^{n-1}+a_n g^n)$, where $a = [a_0, ..., a_n]^t$. One has $\frac{\partial h(a)}{\partial a_i} = \nabla f(x^{n+1} + \sum_j a_j d^j + a_n g^n)^t\, d^i$ for $0 \leq i \leq n-1$ and similarly for $i = n$. So h has a stationary point at $a = 0$ implies $\nabla f(x^{n+1})$ is orthogonal to the d^i and to g^n.

Notice that this is true for any f (we didn't use the assumption f quadratic). □

As a first consequence of this lemma, if we perform an exact search of the minimum at each descent, the dimension of the search space $\mathcal{W}_n$ augments by one. Observe also that $\mathcal{W}_{n+1} = x_0 + sp\{g^0, g^1, ..., g^{n-1}, g^n\}$, and that the g^n form an orthogonal family.

Lemma 4 *If x^n is the minimum[11] of f in $\mathcal{W}_n$, d^n is the direction of the minimum in $\mathcal{W}_{n+1}$ iff $(d^n)^t A d^i = 0$ for $0 \leq i \leq n-1$. The direction d^n is said to be* **conjugate** *to the other descent directions d^i with respect to A.*

Proof. Let $x^{n+1} = x^n + td^n$ be the minimum of f in $\mathcal{W}_{n+1}$, with $t \geq 0$, so

$$g^{n+1} \quad \triangleq \quad \nabla f(x^{n+1}) \;=\; Ax^{n+1} + b \;=\; g^n + tAd^n \tag{80}$$

[11] Here we say "minimum" for clarity, assuming that A is a positive matrix. In reality, these results only characterize *stationary* points of f, since they only consider vanishing points of the gradient.

By the previous lemma, g^{n+1} satisfies

$$\begin{aligned}(g^{n+1})^t g^n &= \|g^n\|^2 + t(d^n)^t A g^n \;=\; 0 && (81)\\ (g^{n+1})^t d^i &= (g^n)^t d^i + t(d^n)^t A d^i \;=\; 0 \qquad \text{for } 0 \le i \le n-1 && (82)\end{aligned}$$

In the second equation, $(g^n)^t d^i = 0$ since x^n was the optimum in $\mathcal{W}_n$. The first equation gives $t > 0$. Reporting this in the second allows us to conclude. □

How can we determine this search direction d^n, conjugate to all the previous ones ? Observe that $g^{i+1} - g^i = A(x^{i+1} - x^i) \propto Ad^i$, and so d^n conjugate to $d^0, ..., d^{n-1}$ iff $(d^n)^t g^i$ is a constant for $0 \le i \le n$. Since the g^i form an orthogonal family, this suggests to take

$$d^n \;=\; -g^n + c_n d^{n-1} \tag{83}$$

which corresponds to the steepest slope slightly corrected by the previous descent direction. One easily checks that

$$\begin{aligned}(d^n)^t g^i &= -(g^n)^t g^i + c_n (d^{n-1})^t g^i\\ &= c_n (d^{n-1})^t g^i\\ &= \textit{constant} \qquad \text{for } 0 \le i \le n-1 && (84)\end{aligned}$$

since d^{n-1} was already conjugate to all the previous d^i. So we only have to adjust c_n to extend this property to $i = n$. One has

$$\begin{aligned}(d^n)^t g^n &= -\|g^n\|^2 + c_n (d^{n-1})^t g^n\\ &= -\|g^n\|^2 && (85)\\ (d^n)^t g^{n-1} &= -(g^n)^t g^{n-1} + c_n (d^{n-1})^t g^{n-1}\\ &= c_n (d^{n-1})^t g^{n-1}\\ &= -c_n \|g^{n-1}\|^2 && (86)\end{aligned}$$

so taking

$$c_n \;=\; \frac{\|g^n\|^2}{\|g^{n-1}\|^2} \tag{87}$$

ensures this property. (87) was proposed by Fletcher and Reeves in 1964. In the literature, one may also find other expressions for c_n. For example, one directly derived from the conjugation property of d^n.

$$\begin{aligned}(d^n)^t A d^{n-1} &= -(g^n)^t A d^{n-1} + c^n (d^{n-1})^t A d^{n-1} \;=\; 0 && (88)\\ & && (89)\end{aligned}$$

which entails

$$c_n \;=\; \frac{(g^n)^t A d^{n-1}}{\|d^{n-1}\|_A^2} \tag{90}$$

Exercise 22 *Take this last expression for c_n, and prove directly that d^n is conjugate to all the d^i, not only d^{n-1}.*
Hint: compute the optimal step t_i in $x^{i+1} = x^i + t_i d^i$.

Comments:

- In the case of a quadratic form in $\mathbb{R}^d$, the conjugate gradient converges in d steps to the minimum of f. The method can be understood as a recursive way of solving $Ax + b = 0$.

- It is remarkable that this method is a very small perturbation of the gradient search, with identical complexity, but with much better convergence properties, in particular for quadratic forms.

- The method can of course be applied to non quadratic forms, provided the Hessian doesn't change much when we move from x^n to x^{n+1}. Otherwise, it may happen that the directions d^n become meaningless, or even are not admissible descent directions! Therefore, one should test the validity of d^n, and regularly "reset" the method by taking $d^n = -g^n$.

- Polak and Ribière proposed a variant of (87) (in 1971)

  $$c_n = \frac{(g^n - g^{n-1})^t g^n}{\|g^{n-1}\|^2} \tag{91}$$

  which seems to give better results for non quadratic forms, in particular when g^n is close to g^{n-1}. This entails that c_n is close to 0, which automatically resets the algorithm by taking $d^n = -g^n$.

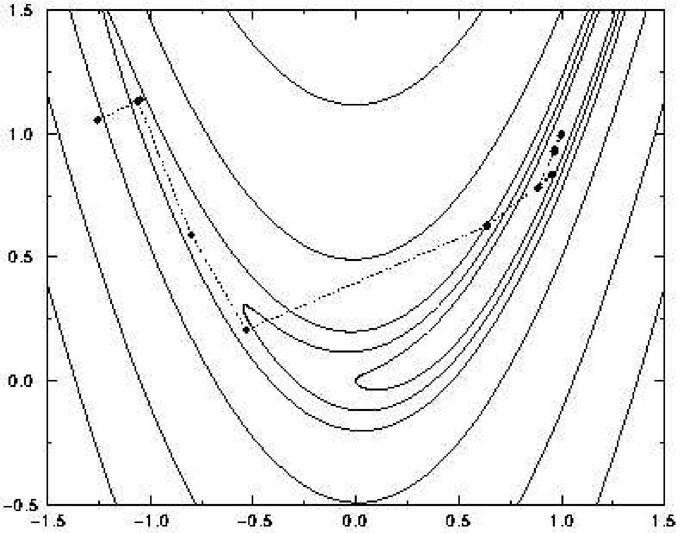

Figure 19: *Conjugate gradient search on Rosenbrock's banana.*

3.8 Quasi-Newton method

The efficiency of the Newton method is very appealing, but the price to pay may be discouraging. Indeed, it requires to know the Hessian $\nabla^2 f$ and to resolve the possibly large linear system (76). Quasi-Newton methods are first order methods that try to mimic the behavior of a Newton method, while keeping the complexity low.

The idea is to compute at each step an estimate K_n of the inverse Hessian $\nabla^2 f(x^n)^{-1}$, and to replace the theoretical optimal step by

$$x^{n+1} \;=\; x^n - t\, K_n\, \nabla f(x^n) \tag{92}$$

where the step t is obtained by a linear search along the direction $K_n \nabla f(x^n)$. Considering the first order Taylor expansion of ∇f given by (75), this matrix K_n must satisfy the **quasi-Newton condition**

$$\begin{aligned} x^{n+1} - x^n &= K_{n+1}\,[\nabla f(x^{n+1}) - \nabla f(x^n)] \\ &= K_{n+1}\,(g^{n+1} - g^n) \end{aligned} \tag{93}$$

In reality, one would like to have this relation satisfied by K_n, but K_n is used to compute x^{n+1}... So we will impose that the relation be satisfied at the next step.

All the subtlety of quasi-Newton methods consists in building matrices K_n in a simple manner. They all proceed recursively by

$$K_{n+1} \;=\; K_n + C_n \tag{94}$$

where the correction C_n is chosen to satisfy (93). The objective is of course to rapidly converge to the true inverse Hessian, in particular when f is quadratic... The exercise below shows that this is possible.

Exercise 23 *Let $u^1, ..., u^d$ be a family of pairwise conjugate vectors wrt the positive symmetric matrix A. Consider the matrices*[12] $K_n = \sum_{i=1}^n \frac{1}{\|u^i\|_A^2}\, u^i (u^i)^t$*, $n \leq d$, that can be built recursively by a formula like (94). Check that $K_d = A^{-1}$.*

Corrections of rank 1. Let us adopt notations

$$u^n \;\triangleq\; x^n - x^{n-1} \tag{95}$$

$$v^n \;\triangleq\; g_n - g^{n-1} \tag{96}$$

so that (93) becomes $u^n = K_n v^n$. This method proposes to update K_n with a correction $C_n = \alpha_n w^n w^{nt}$ with w^n a (column) vector in $\mathbb{R}^d$. Plugging this in (93) yields

$$u^{n+1} \;=\; K_{n+1} v^{n+1} \;=\; K_n v^{n+1} + \alpha_n w^n [(w^n)^t v^{n+1}] \tag{97}$$

so one should take $w_n \propto u^{n+1} - K_n v^{n+1}$. By computing the resulting α_n, this yields

$$K_{n+1} \;=\; K_n + \frac{w^n (w^n)^t}{(w^n)^t\, v^{n+1}} \quad \text{where} \quad w^n \;=\; u^{n+1} - K_n v^{n+1} \tag{98}$$

[12] Observe that each $u^i (u^i)^t$ is indeed a matrix, of rank one.

The drawback of this method is that the recursion doesn't guarantee the positivity of K_{n+1} because the coefficient α_n may be negative. The denominator may also be close to zero, which can cause instabilities.

Exercise 24 *Prove that $K_n v^i = u^i$ also for $i < n$ (as it is the case for most quasi-Newton methods). If $u^1, ..., u^d$ are independent vectors, show that K_d becomes invertible after these d steps. And in the case of a quadratic form, show that $K_d A = \mathbb{1}$.*

DFP. This construction of K_n was proposed by Davidon, Fletcher and Powell. It is based on a correction of rank 2 :

$$K_{n+1} \quad = \quad K_n + \frac{u^{n+1}(u^{n+1})^t}{(u^{n+1})^t\, v^{n+1}} - \frac{K_n v^{n+1}(v^{n+1})^t K_n}{(v^{n+1})^t K_n\, v^{n+1}} \tag{99}$$

The verification of (93) is left as an (easy) exercise. The interest of this method is that the successive descent directions u^n are conjugate when f is quadratic. This allows us to prove that K_n converges in d steps to A^{-1}. However, the DFP is sensitive to the precision of the linear search.

Exercise 25 *When the matrix K_0 is set to identity, show that the DFP coincides with the conjugate gradient method.*

BFGS. This construction of K_n was proposed by Broyden, Fletcher, Goldfarb and Shanno in 1970, and is considered as the best at this time. The correction of K_n is given by

$$\begin{aligned} K_{n+1} \quad = \quad & K_n - \frac{u^{n+1}(v^{n+1})^t K_n + K_n v^{n+1}(u^{n+1})^t}{(u^{n+1})^t\, v^{n+1}} \\ & + \left(1 + \frac{(v^{n+1})^t K_n v^{n+1}}{(u^{n+1})^t v^{n+1}}\right) \frac{u^{n+1}(u^{n+1})^t}{(u^{n+1})^t v^{n+1}} \end{aligned} \tag{100}$$

We leave again the verification of (93) as an exercise.

Comments :

- The K_n computed by all the above formulae don't always determine a descent direction, so this should always be checked.

- For the BFGS, if the scalar product $(u^{n+1})^t v^{n+1}$ is positive, then K_{n+1} remains positive. This ensures that it indicates a correct descent direction. However, it should be checked that this product doesn't become too small, which may cause instabilities.

- As for the conjugate gradient, it is important in practice to regularly reinitialize K_n to identity.

- It is also possible to mix BFGS and DFP, by taking a convex combination of their corrective terms. This defines the Broyden family of quasi-Newton methods, which have been shown to perform well on vast families of C^1 functions.

4 Optimization in $\mathbb{R}^d$ with constraints

In practice, most (numerical) optimization problems limit the possible values of x to a subset $\mathcal{D}$ of $\mathbb{R}^d$. Here, we consider the case where this domain is bounded by C^1 functions $\theta_j(x)$. We will have equality constraints like $\theta_j(x) = 0$, that impose x to live on a manifold, as well as inequality constaints like $\theta_j(x) \leq 0$, that represent one "side" of the manifold $\theta_j(x) = 0$. Compared to the previous chapter, the difficulty now is that we must take into account not only the "geometry" of the cost function f, but also the "geometry" of the boundaries defined by the θ_j. As before, we rather insist on these geometrical interpretations than on detailed convergence properties of the algorithms.

4.1 Equality constraints

The problem we consider here still consists in minimizing $f(x)$, but subject to m constraints θ_j, that limit the possible values of x:

$$\min_x f(x) \quad \text{s.t.} \quad \theta_j(x) = 0, \;\; 1 \leq j \leq m \tag{101}$$

where $f, \theta_j : \mathbb{R}^d \to \mathbb{R}$. We will often write constraints in vector form $\theta(x) = 0 \in \mathbb{R}^m$, by aggregating the m functions θ_j in a column vector. Notice that one must take $m < d$, otherwise the **domain** $\mathcal{D} = \{x : \theta(x) = 0\}$ **of admissible solutions** may reduce to a few points or be empty.

4.1.1 Introduction to the Lagrange multipliers method

To help intuition, let us start with a simple example. Assume we want to determine the dimensions of a pan (radius r of the bottom and height h of its side) in order to minimize its surface (*i.e.* the quantity of metal used to make the pan) for a fixed volume of 1 litre. Formally

$$\begin{aligned} f(x) &= \pi {x_1}^2 + 2\pi x_1 x_2 & (102) \\ \theta(x) &= \pi {x_1}^2 x_2 - 1 & (103) \end{aligned}$$

where x_1 is the radius of the pan and x_2 its height (Fig. 20).

The constraint is bothering, so let us "relax" it: we introduce it into the cost function, with a penalty factor λ. Specifically, we condider the new cost function

$$L(x, \lambda) = f(x) + \lambda \theta(x) \tag{104}$$

that is called the **Lagrangian**[13] of the problem. Intuitively, if the constraint θ always took positive values, this would amount to assigning a relative weight to the constraint in the new cost function.

[13] Joseph Louis, count of Lagrange (in Italian Giuseppe Lodovico Lagrangia), born in Turin in 1736 and dead in Paris 1813, is an Italian mathematician and astronomer.

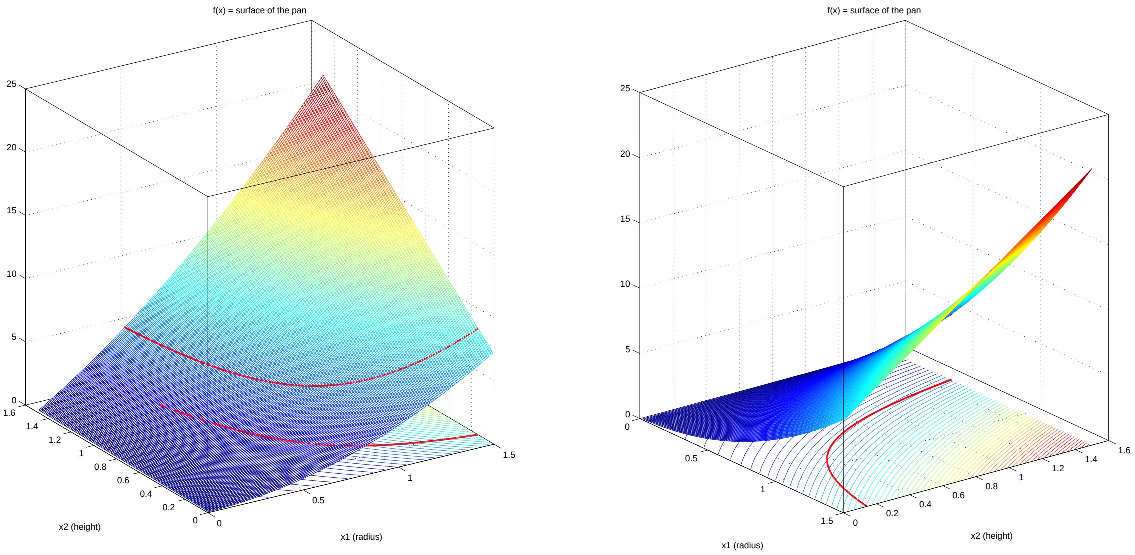

Figure 20: *Plots of f, surface of the pan, as a function of the diameter and of the height. Red curves : points x satisfying the constraint $\theta(x) = 0$ on the volume of the pan, and their image by f.*

Observe that, for any fixed λ, f and L have the same minima (and maxima) on the domain defined by $\theta(x) = 0$. So let us minimize this Lagrangian in x, for any value of λ. A stationary point is obtained for $\nabla_x L = 0$, *i.e.*

$$\frac{\partial L(x,\lambda)}{\partial x_1} = 2\pi x_1 + 2\pi x_2 + \lambda 2\pi x_1 x_2 = 0 \qquad (105)$$

$$\frac{\partial L(x,\lambda)}{\partial x_2} = 2\pi x_1 + \lambda \pi {x_1}^2 = 0 \qquad (106)$$

we obtain[14] $x_1^* = x_2^* = -\frac{2}{\lambda}$.

The next step consists in adjusting λ in order to satisfy the constraint, *i.e.* to ensure that the $x^*(\lambda)$ that we find is in the desired domain (where f and L have identical minima). This gives $\lambda^* = -\frac{\pi^{1/3}}{2}$ and so $x_1^* = x_2^* = \pi^{-1/3}$.

We conclude by the following reasoning : for this particular value λ^* of λ, we have looked for stationary points of $L(x, \lambda^*)$ in the whole space $\mathbb{R}^d$, and the $x^*(\lambda)$ we found belongs to the domain $\mathcal{D} = \{x : \theta(x) = 0\}$. Since f and L have the same stationary points in this domain, then x^* is also stationary for f.

As a last step, one must of course check that the stationary point x^* is a (local) minimum of f, and not a (local) maximum...

This resolution method can of course be generalized to m constraints. In the next section, we develop the theory that justifies this approach.

[14] In passing, notice this remarkable fact that the height of an optimal pan should be equal to its radius. We invite the reader to check the optimality of his own pans...

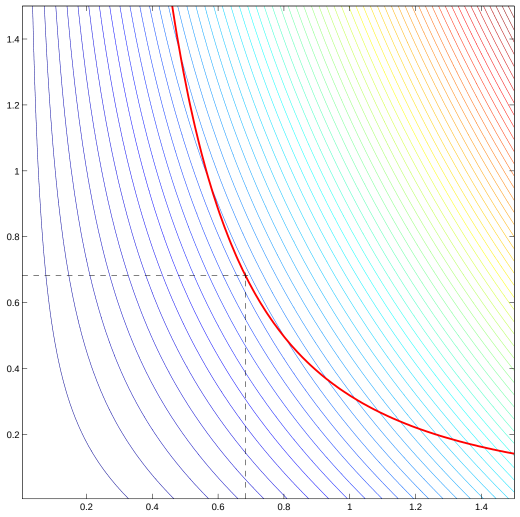

Figure 21: *Level lines of f. The red curve corresponds to $\theta(x) = 0$. The minimum of f on the red curve is represented by its coordinates.*

Exercise 26 *In $\mathbb{R}^2$, consider the circle defined by $\theta(x) = (x_1-1)^2+(x_2-1)^2-1 = 0$. Use the method of Lagrange multipliers to compute the point of this circle that is the closest to the origin (which amounts to minimizing $f(x) = {x_1}^2 + {x_2}^2$). Observe that the Lagrangian has two stationary points.*

Exercise 27 *In $\mathbb{R}^3$, compute the radius of the sphere that is tangent to the plane $x_1 + x_2 + x_3 = 1$. This amounts to minimizing $f(x) = \|x\|^2$ s.t. $b^t x = 1$ where $b^t = [1, 1, 1]$.*

4.1.2 Lagrange optimality conditions

The constraints $\theta_j(x) = 0$, with $1 \leq j \leq m$, define a **differentiable manifold** $\mathcal{D}$ in $\mathbb{R}^d$. Consider one of these constraints θ_j, and its gradient $\nabla\theta_j(x^0)$ at x^0. The **tangent hyperplane** to the manifold $\theta_j(x) = 0$ at point x^0 is defined by the (affine) equation (recall section 3.1).

$$\nabla\theta_j(x^0)^t(x - x^0) \quad = \quad 0 \tag{107}$$

and so the **tangent space** to domain $\mathcal{D}$ at x^0 is the intersection of these tangent hyperplanes. It is common to gather the functions θ_j in a vector of $\mathbb{R}^m$, and to denote by $\nabla\theta(x^0)$ the juxtaposition of the (column) vectors $\nabla\theta_j(x^0)$, which results in a $d \times m$ matrix. The tangent space is then defined by

$$\nabla\theta(x^0)^t(x - x^0) \quad = \quad 0 \tag{108}$$

Definition 4 x^0 *is a* **regular point** *of $\mathcal{D}$ if the gradients $\nabla\theta_j(x^0)$ are linearly independent, or equivalently if $\nabla\theta(x^0)$ is of rank m. The manifold $\mathcal{D}$ is regular iff all its points are regular.*

Notice that this excludes situations where the gradient vanishes. In particular $\theta_j(x) = 0$ can not have double points (Fig. 22). In the sequel, we limit ourselves to regular manifolds, or at least study them at regular points.

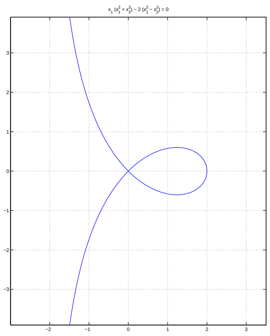

Figure 22: *The curve defined by $x_1(x_1^2 + x_2^2) - 2(x_1^2 - x_2^2) = 0$ has a multiple point at $x_1 = x_2 = 0$, where the gradient is necessarily null. This point is not regular.*

The following result gives necessary conditions on the extrema of f in $\mathcal{D}$.

Theorem 2 *Let x^* be a regular point of $\mathcal{D} = \{x : \theta(x) = 0\}$. If x^* is a local extremum of f in $\mathcal{D}$, there exists a (unique) vector $\lambda^* \in \mathbb{R}^m$ of* **Lagrange multipliers** *such that*

$$\nabla f(x^*) + \sum_{j=1}^{m} \lambda_j^* \; \nabla\theta_j(x^*) \quad = \quad 0 \tag{109}$$

Proof. (sketch of)
Consider the vector space $\mathcal{V} = sp\{\nabla\theta_1(x^*), ..., \nabla\theta_m(x^*)\}$, and let us project $\nabla f(x^*)$ on $\mathcal{V}$. In other words, $\nabla f(x^*)$ decomposes as

$$\nabla f(x^*) \quad = \quad \sum_{j=1}^{m} -\lambda_j^* \; \nabla\theta_j(x^*) + u \tag{110}$$

where u is orthogonal to $\mathcal{V}$, *i.e.* to all the $\nabla\theta_j(x^*)$. The remainder u belongs to the tangent space to $\mathcal{D}$ at x^*. If $u \neq 0$, then making a small step in the direction of $-u$ from x^* will decrease the value of f (the first order term in the Taylor expansion of f is negative), and leave the constraints unchanged (the first order term of the taylor expansion of θ_j vanishes). In other words, a small step on the manifold $\mathcal{D}$ allows us to decrease f, which contradicts the stationarity of x^*. Symmetrically, a small step in the direction of u would increase f and leave the constraints unchanged (Fig. 23). □

The attentive reader will have noticed that the regularity condition of x^* didn't appear in the above arguments. The true (more technical) proof actually reasons on *admissible* directions u of the tangent space that would effectively leave the

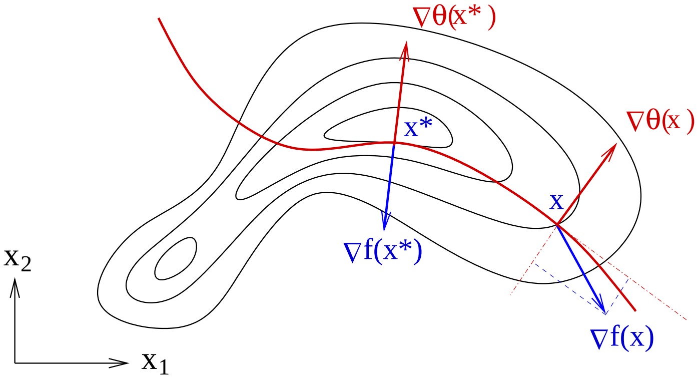

Figure 23: *f represented by its level lines, and the constraint $\theta(x) = 0$ represented as a red curve. An extremum of f s.t. θ is necessarily obtained at a point where the level line of f is tangent to the manifold $\mathcal{D}$ defined by the constraint.*

constraints unchanged (and would possibly change f). They are obtained as limits of directions $\frac{x^n - x^*}{\|x^n - x^*\|}$ where $x^n \in \mathcal{D}$ and $\lim_n x^n = x^*$. When gradients $\nabla\theta_j(x^*)$ are not linearly independent, one may find directions in the tangent space that can't be expressed as such a limit, *i.e.* that are not admissible. We give an example in appendix A. We also provide below counter-examples to (109) when the regularity[15] condition is violated.

Definition 5 *The* **Lagrangian** *of the non-linear program* $(f, \{\theta_j\}_{1\leq j\leq m})$ *is the function* $L : \mathbb{R}^{d+m} \to \mathbb{R}$

$$L(x,\lambda) \quad = \quad f(x) + \sum_{j=1}^{m} \lambda_j \, \theta_j(x) \tag{111}$$

Remarks

- The conditions (109) together with constraints $\theta_j(x^*) = 0$ form a set of $d+m$ non-linear equations that may be sufficient to determine the $d+m$ unknowns (x^*, λ^*). This is actually how we solved the example in 4.1.1. Of course, when an analytic resolution is not possible, the numerical methods we describe later will try to solve these equations.

- Observe that solving the $d+m$ equations above exactly amounts to finding a stationary point of the Lagrangian. (109) corresponds to $\nabla_x L(x,\lambda) = 0$, and the constraints to $\nabla_\lambda L(x,\lambda) = 0$, where the notation $\nabla_y L$ represents the partial gradient formed by the partial derivatives in the components of y.

- In practice, it is often simpler to determine first the λ^*, and then compute the x^* as a function of λ^*. See for example exercise 27. This is the principle of the resolution by duality that we describe later.

[15] Instead of "regularity" of x^*, the expression "qualification of constraints" $\nabla\theta_i(x^*)$ is often used. There exist weaker forms than the linear independence. See [1] for details.

- The theorem only characterizes stationary points of f (or of the Lagrangian). One still has to check that the points found correspond to a minimum (see exercise 26). We give sufficient conditions in the next section.

To illustrate the importance of the regularity condition, consider $f(x) = \|x\|^2$ with $\theta(x) = {x_2}^2 - (x_1 - 1)^3$. One easily checks (graphically) that the minimum is $x_1^* = 1, x_2^* = 0$. But the gradient of θ is null at this point, so x^* is not regular and theorem 2 doesn't hold. The reader can check (exercise) that it is not possible to find a λ^* that satisfies (109).

Exercise 28 *In $\mathbb{R}^3$, compute the minimum of $f(x) = x_1x_2x_3$ s.t. $\theta(x) = x_1 + x_2 + x_3 - 3 = 0$, and $x_i \geq 0$. Check that there exist several extremal points.*

An important specific case concerns quadratic forms under affine constraints.

Exercise 29 *Consider min $f(x) = \frac{1}{2}x^tAx + b^tx$ under the constraints $\theta(x) = Cx - c = 0$, with $A \in \mathbb{R}^{m\times d}$. If A is invertible, and C is of maximal rank (m), prove that the unique stationary point of the Lagrangian is given by*

$$\lambda^* = -(CA^{-1}C^t)^{-1}[c - CA^{-1}b] \tag{112}$$

$$x^* = A^{-1}\{b - C^t(CA^{-1}C^t)^{-1}[c - CA^{-1}b]\} \tag{113}$$

Under what conditions x^* is a minimum ?

Exercise 30 *Show that the projection of x^0 on the affine manifold defined by $Cx = c$, $C \in \mathbb{R}^{m\times d}$, is given by $x^* = x^0 - C^t(CC^t)^{-1}(Cx^0 - c)$, when C has full rank.*

4.1.3 Second order necessary/sufficient conditions

In the unconstrained case, the “sign” of the Hessian of f allows us to check whether a stationary point of f is a local minimum or maximum. This extends to the constrained case, with an important light difference however : the second derivatives of the constraints θ_j must also be taken into account. (This regularity assumption on the manifold $\mathcal{D}$ is rarely met in practice.)

Theorem 3 *Let x^* be a (regular) stationary point of f on $\mathcal{D} = \{x : \theta(x) = 0\}$, with associated vector of Lagrange multipliers $\lambda^* \in \mathbb{R}^m$ (see theorem 2). Consider the Hessian in the variable x of the Lagrangian $L(x, \lambda)$ at point (x^*, λ^*) :*

$$\nabla_x^2 L(x^*, \lambda^*) = \nabla^2 f(x^*) + \sum_{j=1}^{m} \lambda_j^* \nabla^2\theta_j(x^*) \tag{114}$$

NC : If x^* is a minimum of f on $\mathcal{D}$, then $\nabla_x^2 L(x^*, \lambda^*)$ defines a positive quadratic form on the kernel of matrix $\nabla\theta(x^*)^t$.
SC : If $\nabla_x^2 L(x^*, \lambda^*)$ is strictly positive on this space, then x^* is a local minimum of f on $\mathcal{D}$.

The kernel of $\nabla\theta(x^*)^t$ simply represents the space of vectors that are orthogonal to the gradients of all constraints, *i.e.* directions of the tangent space at x^*. Therefore the condition reads

$$u^t\,[\nabla_x^2\,L(x^*,\lambda^*)]\,u\geq 0 \qquad \forall u\in\mathbb{R}^d\;:\;\theta_1(x^*)^t u=\ldots=\theta_m(x^*)^t u=0 \quad (115)$$

The proof of this theorem is detailed in appendix B. It is important to notice that *the Hessian of f is replaced by the Hessian of the Lagrangian* in the constrained case. The reason is that the second order Taylor expansion of f on $\mathcal{D}$ requires to know the second order expansion of the constraints (the tangent space is not sufficient anymore, one must take into account the curvature of the constraint space). The example below illustrates this fact.

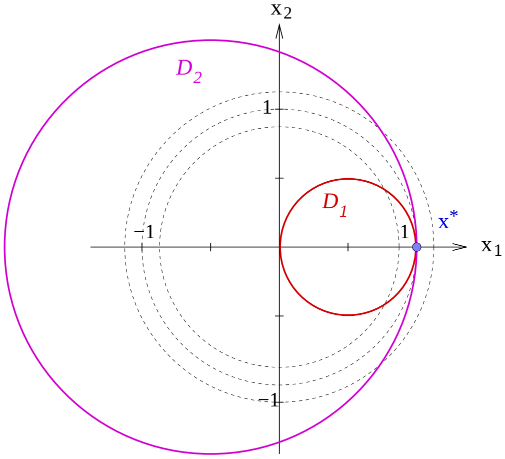

Figure 24: *Two domains, $\mathcal{D}_1$ and $\mathcal{D}_2$, and the level lines of f (in dotted lines). The point $x^*=[1,0]^t$ is a min of f on $\mathcal{D}_2$ but a max of f on $\mathcal{D}_1$. This can only be seen by checking the Hessian of the Lagrangian.*

Example. Consider $f:\mathbb{R}^2\to\mathbb{R}$ defined by $f(x)={x_1}^2+{x_2}^2-1$ and the domain $\mathcal{D}_1$ defined as the circle $\theta(x)=(x_1-\frac{1}{2})^2+{x_2}^2-\frac{1}{4}=0$. The point $x^*=[1,0]^t$ is a stationnary point of f on $\mathcal{D}_1$, with Lagrange multiplier $\lambda^*=-2$. In effect, one has $\nabla f(x^*)=[2,0]^t$ and $\nabla\theta(x^*)=[1,0]^t$. Since $\nabla^2 f(x^*)=2\cdot\mathbb{1}=\nabla^2\,\theta(x^*)$, we observe that $\nabla_x^2\,L(x^*,\lambda^*)=-2\cdot\mathbb{1}$ is a negative symmetric matrix, and conclude that x^* is a maximum of f on $\mathcal{D}_1$ (see Fig. 24).

By constrast, consider the domain $\mathcal{D}_2$ defined as the circle $\theta(x)=(x_1+\frac{1}{2})^2+{x_2}^2-\frac{9}{4}=0$. This time one has $\nabla\theta(x^*)=[3,0]^t$ and so $\lambda^*=-\frac{2}{3}$. This induces that $\nabla_x^2\,L(x^*,\lambda^*)=\frac{2}{3}\cdot\mathbb{1}$ is a strictly positive matrix, and so x^* is now a minimum of f on $\mathcal{D}_2$.

Exercise 31 *Consider now the domain $\mathcal{D}_3$ defined as the circle $\theta(x)={x_1}^2+{x_2}^2-1=0$. What can we say about x^* ?*

Exercise 32 *Taking for f an hyperbolic paraboloid (*i.e. *a saddle) instead of a paraboloid, build an example where the positivity criterion only holds for vectors of the tangent space.*

4.2 Inequality constraints

We now consider the non-linear program

$$\min_x f(x) \quad \text{s.t.} \quad \theta_j(x) \leq 0, \;\; 1 \leq j \leq m \tag{116}$$

where $f, \theta_j : \mathbb{R}^d \to \mathbb{R}$. Of course, it would be possible to have both equality and inequality constraints, which we avoid here for the clarity of the presentation. The **domain of admissible solutions** now writes $\mathcal{D} = \{x : \theta(x) \leq 0\}$, using the vector form of constraints (all coordinates must be negative).

4.2.1 Cone of admissible directions

As for equality constraints, the difficulty is to understand what directions of $\mathbb{R}^d$ one could explore, from a point x^0, in order to reduce f and *to stay inside the domain* $\mathcal{D}$. Let us first distinguish **active** from **inactive** constraints.

Definition 6 *The constraint θ_j is* **active** *at x^0 when $\theta_j(x^0) = 0$,* i.e. *when x^0 is on the border of $\mathcal{D}$ defined by θ_j. We denote by $\mathcal{A}(x^0) = \{j : \theta_j(x^0) = 0\}$ the set of active constrains.*

Definition 7 *For $x^0 \in \mathcal{D}$, the direction u is* admissible *iff there exists $\epsilon > 0$ such that $x^0 + \epsilon\, u \in \mathcal{D}$, or more precisely iff there exists a series $(x^n)_{n>0}$ in $\mathcal{D}$ such that $\lim_n x^n = x$ and $\lim_n \frac{x^n - x^0}{\|x^n - x^0\|} = \frac{u}{\|u\|}$. These directions form the* **cone of admissible directions** *$\mathcal{C}(x^0)$ at x^0.*

Definition 8 *A* **cone** *$\mathcal{C}$ of $\mathbb{R}^d$ is a subset such that $\forall u \in \mathcal{C},\ \forall \alpha \geq 0,\ \alpha u \in \mathcal{C}$. It is* **convex** *if $\forall u, v \in \mathcal{C},\ \forall 0 \leq \alpha \leq 1,\ \alpha u + (1-\alpha) v \in \mathcal{C}$.*

The cone of admissible directions $\mathcal{C}(x^0)$ is not always convex see a counter-example in Fig. 25, left. However, if the *active* constraints are regular[16] at x^0, one has the following result.

Theorem 4 *If the active constraints at x^0 are regular,* i.e. *if the $\nabla\theta_j(x^0),\ j \in \mathcal{A}(x^0)$ are linearly independent, then the cone of admissible directions at x^0 is convex and defined by*

$$\mathcal{C}(x^0) \quad = \quad \{u \in \mathbb{R}^d \; : \; \nabla\theta_j(x^0)^t\, u \leq 0,\, j \in \mathcal{A}(x^0)\} \tag{117}$$

This result is illustrated in Fig. 25. We admit it, although its interpretation is rather intuitive : admissible directions must decrease the values of active constraints, to have $x = x^0 + \epsilon\, u$ in $\mathcal{D}$ (inactive constraints impose no restriction on the possible displacement around x^0). Augmenting active constraints necessarily leads outside $\mathcal{D}$. Using the 1st order Taylor expansion of θ_j around x^0, this immediately translates

[16] The regularity of constraints can be weakened into a “qualification condition” on active constraints, see [1], chapter 9.2.

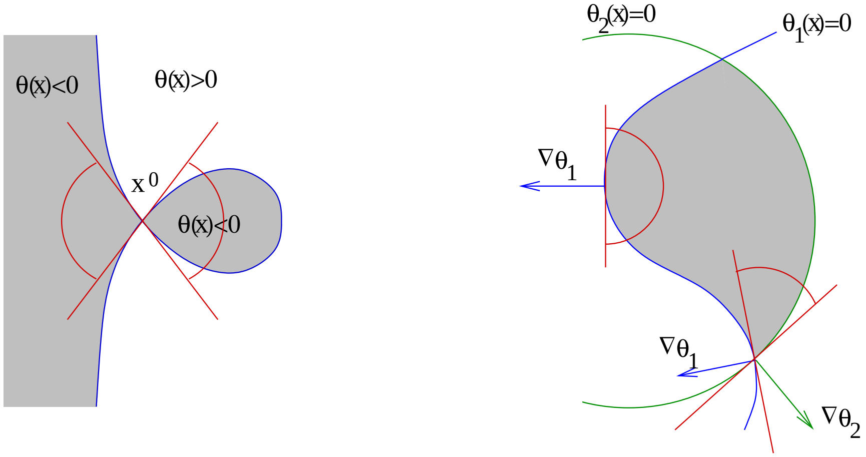

Figure 25: *Left: a non-convex cone of admissible directions, at a non regular point* x^0. *Right: (convex) cones of admissible directions, placed at regular points.*

into the negative scalar products of (117). The difficult part of the theorem thus relates to the convexity of $\mathcal{C}(x^0)$.

We now state a famous and useful result is associated to cones as defined in (117).

Lemma 5 (Farkas-Minkowski) *Let the* u_j *be vectors in* $\mathbb{R}^d$, *and consider the convex cone* $\mathcal{C} = \{x \in \mathbb{R}^d : u_j^t\, x \geq 0,\ 1 \leq j \leq m\}$. *Given a vector* $v \in \mathbb{R}^d$, *consider the half-space* $\{x : v^t\, x \geq 0\}$ *that it defines. One has*

$$\mathcal{C} \subseteq \{x : v^t\, x \geq 0\} \quad \textit{iff} \quad v = \sum_j \alpha_j\, u_j, \ \ \alpha_j \geq 0 \qquad (118)$$

The proof, not difficult, can be found in [1] (or may be done as an exercise). We prefer to insist here on the geometrical interpretation of the lemma, which is convincing enough to replace a detailed proof. C is the intersection of m half-spaces (see Fig. 26). This domain is contained in the half-space pointed by v iff v is inside the convex cone generated by the vectors u_j. Fig. 26 gives an example and a counter-example of this situation.

This result is sometimes expressed under a different form [3]. Given a cone C, the **dual cone** C' of C is defined by $C' = \{y : \forall x \in C, x^t y \leq 0\}$. The Farkas lemma thus states an NSC[17] for $-v$ to belong to the dual of $C = \{x \in \mathbb{R}^d : u_j^t\, x \geq 0,\ 1 \leq j \leq m\}$. Or equivalently states that $C' = \{-\sum_j \alpha_j\, u_j,\ \ \alpha_j \geq 0\}$.

Exercise 33 *Verify that these two expressions of the Farkas lemma are equivalent.*

Exercise 34 *Prove that the dual of a cone* C *is a convex cone (even if* C *is not convex). Make a drawing showing the dual of the convex cone generated by vectors* $u_1, ..., u_m$.

[17] NSC = necessary and sufficient condition

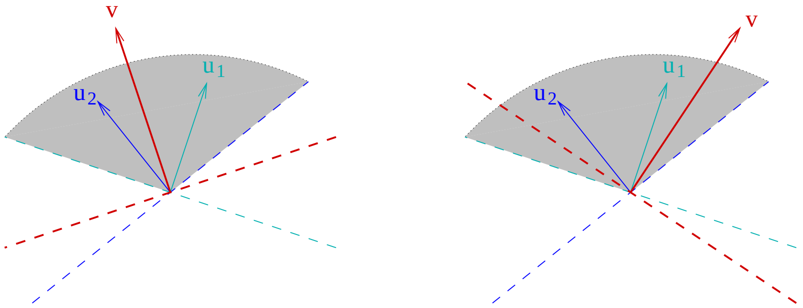

Figure 26: *As v rotates around the origin, the half-space it points to may capture or not the cone C (in gray). It captures it if and only if v stays within the convex cone bounded by the u_j.*

Exercise 35 *Prove that if C is a convex cone, then $(C')' = C$,* i.e. *the dual of the dual, is the cone itself.*

Exercise 36 *Let C be a convex cone, prove that any element $x \in \mathbb{R}^d$ decomposes uniquely as $x = u + v$ where $u \in C$ and $v \in C'$.*

4.2.2 Karush-Kuhn-Tucker optimality conditions

We now have enough material to establish the main theorem

Theorem 5 (Karush-Kuhn-Tucker conditions) *Let x^* be a regular point of domain $\mathcal{D} = \{x : \theta_j(x) \leq 0, 1 \leq j \leq m\}$. If x^* is a local minimum of f on $\mathcal{D}$, there exists a unique set of* **generalized Lagrange multipliers** *λ_j^* for $j \in \mathcal{A}(x^*)$ (the set of active constraints at x^*) such that*

$$\nabla f(x^*) + \sum_{j \in \mathcal{A}(x^*)} \lambda_j^* \nabla \theta_j(x^*) = 0 \quad \textit{and} \quad \lambda_j^* \geq 0, \;\; j \in \mathcal{A}(x^*) \tag{119}$$

Proof. Since x^* is regular, the cone of admissible directions is convex and given by $\mathcal{C}(x^*) = \{u \in \mathbb{R}^d : \nabla\theta_j(x^*)^t u \leq 0, \; j \in \mathcal{A}(x^*)\}$ (theorem 4). Since x^* is a local minimum, one has $\nabla f(x^*)^t u \leq 0$ for every u in $\mathcal{C}(x^*)$, otherwise a small step in direction u from x^* would allow us to decrease f while staying in $\mathcal{D}$. We conclude by Farkas' lemma, with the $-\nabla\theta_j(x^*)$ as the u_j. □

Corollary 1 *The Karush-Kuhn-Tucker conditions are sometimes expressed differently, in a form that resembles more theorem 2:*

$$\nabla f(x^*) + \sum_{j=1}^{m} \lambda_j^* \nabla \theta_j(x^*) = 0 \quad \textit{and} \quad \lambda_j^* \geq 0, \;\; 1 \leq j \leq m \tag{120}$$

with the extra **complementarity condition**

$$\sum_{j=1}^{m} \lambda_j^* \theta_j(x^*) \;\; = \;\; 0 \tag{121}$$

Proof. The difference is that *all* constraints appear in (120). Considering the positivity of the λ_j^* and the negativity of the $\theta_j(x^*)$, the complementarity condition automatically imposes $\lambda_j^* = 0$ for non active constraints. □

Remarks.

- An alternate "proof" of theorem 5 would be to apply theorem 2 to the active constraints, and then prove that the λ_j^* must be positive.
- When there is no active constraint, *i.e.* $\mathcal{A}(x^*) = \emptyset$, (119) reduces to $\nabla f(x^*) = 0$, which is obvious since x^* is strictly inside $\mathcal{D}$ (all directions are admissible).
- (120) and (121) are hard to use in practice because they define a set of (non-linear) *inequations*, by contrast with the set of equations in the case of equality constraints.
- It is possible to mix equality and inequality constraints in the definition of $\mathcal{D}$. Equality constraints thus automatically go into the set of active constraints. It is not wise to replace $\theta(x) = 0$ by the pair $\theta(x) \geq 0$, $\theta(x) \leq 0$ since this kills the regularity assumption.
- Assuming the set of active constraints at the optimum is known, (119) amounts to finding a stationary point of the Lagrangian under the constraints $\lambda_j \geq 0$, as in the case of equality constraints.

4.2.3 A resolution example

The nice closed-form of the Kuhn-Tucker conditions hides a practical difficulty : to be able to use them, one must guess what are the active constraints at the optimum.

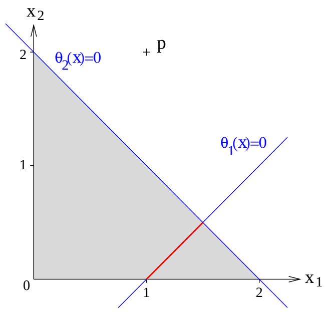

Figure 27: *Minimize the distance to point p in the domain define by the red segment.*

Consider $\min_x f(x) = (x_1-1)^2 + (x_2-2)^2$ s.t. $\theta_1(x) = x_1 - x_2 - 1 = 0$, $\theta_2(x) = x_1 + x_2 - 2 \leq 0$, $\theta_3(x) = -x_1 \leq 0$, $\theta_4(x) = -x_2 \leq 0$ (see Fig. 27). The Lagrangian is

$$\begin{aligned} L(x,\lambda) &= (x_1-1)^2 + (x_2-2)^2 + \lambda_1(x_1 - x_2 - 1) \\ &\quad + \lambda_2(x_1 + x_2 - 2) + \lambda_3(-x_1) + \lambda_4(-x_2) \end{aligned} \tag{122}$$

so the Kuhn-Tucker conditions on the gradient of the Lagrangian yield

$$\begin{aligned}
& & \frac{\partial L(x,\lambda)}{\partial x_1} &= x_1 - 1 + \lambda_1 + \lambda_2 - \lambda_3 = 0 && (123)\\
& & \frac{\partial L(x,\lambda)}{\partial x_2} &= x_2 - 2 - \lambda_1 + \lambda_2 - \lambda_4 = 0 && (124)\\
&\text{equality constraint} & \frac{\partial L(x,\lambda)}{\partial \lambda_1} &= x_1 - x_2 - 1 = 0 && (125)\\
&\text{inequality constraints} & & \lambda_2(x_1 + x_2 - 2) = 0, \;\; \lambda_2 \geq 0 && (126)\\
& & & -\lambda_3 x_1 = 0, \;\; \lambda_3 \geq 0 && (127)\\
& & & -\lambda_4 x_2 = 0, \;\; \lambda_4 \geq 0 && (128)
\end{aligned}$$

We assume first that no inequality constraint is active, which yields $\lambda_2^* = \lambda_3^* = \lambda_4^* = 0$. Solving the rest of the system yields $x_1^* = 2$ and $x_2^* = 1$ which violates $\theta_2(x) \leq 0$...

So we have to change our assumptions and introduce θ_2 in the list of saturated constraints. This yields $\lambda_3^* = \lambda_4^* = 0$. The other equations yield $x_1^* = 3/2$ and $x_2^* = 1/2$ which now is in the desired domain.

This example suggests how complex the resolution can be when there are many inequality constraints...

4.2.4 Convex case

In the unconstrained case, when f is convex, local and global minima coincide. This results extends to the constrained case

Theorem 6 *Let f and the constraints θ_j be convex functions, which defines a convex domain $\mathcal{D}$. Let x^* be a local minimum of f,* i.e. *a point satisfying the Karun-Kuhn-Tucker conditions. Then x^* is also a global minimum of f on $\mathcal{D}$.*

See thm 9.2-4 in [1] for a proof.

4.2.5 Second order necessary/sufficient conditions

This result extends theorem 3 to the case of inequality constraints.

Theorem 7 *Let x^* be a (regular) stationary point of f on $\mathcal{D} = \{x : \theta(x) \leq 0\}$, with associated vector of generalized Lagrange multipliers $\lambda^* \in \mathbb{R}^m$: (x^*, λ^*) satisfy the Kuhn-Tucher conditions (see theorem 5). Consider $\nabla_x^2 L(x^*, \lambda^*)$, the Hessian in the variable x of the Lagrangian $L(x, \lambda)$ at point (x^*, λ^*).*

NC: if x^* is a minimum of f in $\mathcal{D}$, then $u^t\,[\nabla_x^2 L(x^*, \lambda^*)]\,u \geq 0$ for every $u \in \mathbb{R}^d$ such that $\nabla\theta_j(x^*)^t u = 0,\ j \in \mathcal{A}(x^*)$ (the set of active constraints).

SC: if $u^t\,[\nabla_x^2 L(x^*, \lambda^*)]\,u > 0$ for every $u \in \mathbb{R}^d, u \neq 0$ such that $\nabla\theta_j(x^*)^t u = 0$ when $\lambda_j^* > 0$, then x^* is a local minimun of f on $\mathcal{D}$.

Proof. (sketch of) The necessary condition is actually the same as in theorem 3, where only active constraints are selected. Although the criterion is obviously necessary on the manifold that limits $\mathcal{D}$ at x^*, it is strange that nothing is said for the other admissible directions, that point toward the interior of $\mathcal{D}$.

The idea is that if u is an admissible direction such that $\nabla\theta_j(x)^t\, u \leq 0$ for $j \in \mathcal{A}(x^*)$, then $\nabla f(x^*)^t u = -\sum_j \lambda_j^* \nabla\theta_j(x)^t\, u \geq 0$. So the first order term dominates in the Taylor expansion of f and imposes the positivity. So there is no necessary condition imposed to the second order term by this direction u.

In the sufficient condition, observe that $\lambda_j^* > 0$ selects a *subset* of the active constraints, and so the space of admissible directions u that is tested is *larger* that in the necessary condition. □

4.3 Numerical methods

There exists an extremely vast family of numerical methods that address constrained optimization problems when their analytical resolution is too complex. This variety comes from the different situations that arise : linear (or affine) or non-linear constraints, equality or inequality constraints, etc. We briefly sketch some of them below, essentially to illustrate the geometrical intuition that motivates them. Their principle is similar to the unconstrained case : find a descent direction, and progress along this line. There are two essential difficulties to deal with :

- constraints limit the choice of admissible descent directions, and
- the progression along an admissible direction may meet the boundary of $\mathcal{D}$.

4.3.1 Penalty functions

This is the simplest and most natural idea to get rid of constraints. It corresponds to the intuition used in the introduction 4.1.1.

Exterior points. Let us consider equality constraints $\theta(x) = 0$ for example, so the domain of admissible points $\mathcal{D}$ is a manifold. Assume there exists a function $\psi : \mathbb{R}^d \to \mathbb{R}^+$, always positive and vanishing exactly on $\mathcal{D}$, for example $\psi(x) = \|\theta(x)\|^2$, or $\psi(x) = \sum_j |\theta_j(x)|$. We replace the constrained problem

$$\min_x f(x) \quad \text{s.t.} \quad \theta_j(x) = 0, \;\; 1 \leq j \leq m \tag{129}$$

by the unconstrained one

$$\min_x F_k(x), \qquad F_k(x) = f(x) + c_k\, \psi(x), \;\; c_k > 0 \tag{130}$$

Obviously, f and F_k have the same minima. This unconstrained problem can be addressed by standard numerical methods. If it admits a minimum in $\mathcal{D}$, then this is obviously a minimum of f in $\mathcal{D}$. Otherwise, at convergence, one progressively reinforces the penalty term by taking $c_{k+1} > c_k$, which makes the constraint more attractive, on so on with $\lim_k c_k = \infty$.

This method is sometimes called **Lagrangian relaxation**, because of the similarity of the new cost function (130) with the Lagrangian.

Interior points. Better suited to inequality constraints $\theta(x) \leq 0$. Instead of penalizing non-admissible solutions, the principle is to forbid them completely, and to penalize elements of the domain $\mathcal{D}$ that are close to the boundaries, in order to force a numerical method to stay "inside" $\mathcal{D}$.

Formally, let $\psi(x) : \mathbb{R}^d \to \mathbb{R}^+$ be such that $\psi(x) \to +\infty$ when $\theta_j(x) \to 0_-$ for some j. For example $\psi(x) = -\sum_j \frac{1}{\theta_j(x)}$. We replace the constrained problem

$$\min_x f(x) \quad \text{s.t.} \quad \theta_j(x) \leq 0, \;\; 1 \leq j \leq m \tag{131}$$

by the unconstrained one

$$\min_x F_k(x), \qquad F_k(x) = f(x) + c_k\, \psi(x), \;\; c_k > 0 \tag{132}$$

and minimize the successive F_k, letting c_k go to 0.

4.3.2 Projected gradient

This family of methods consist in exploring only admissible points, *i.e.* points in the domain $\mathcal{D}$ defined by constraints. They vary according to the nature of $\mathcal{D}$. We start with an instrumental result.

Lemma 6 *Let $C_1, ..., C_m \in \mathbb{R}^d$ be linearly independent (column) vectors, and let $C = [C_1, ..., C_m]$ be the $d \times m$ matrix obtained by juxtaposing these vectors. The projection $\pi_C(x)$ of $x \in \mathbb{R}^d$ on $sp\{C_1, ..., C_m\}$, the vector space generated by the C_j, is given by*

$$\pi_C(x) = Px \quad \textit{with} \quad P = C(C^tC)^{-1}C^t \tag{133}$$

Proof. Elements in $sp\{C_1, ..., C_m\}$ can be expressed as $C\alpha$ where $\alpha \in \mathbb{R}^m$, so the problem amounts to finding the optimal coefficients $\alpha^* = \arg\min_\alpha \|x - C\alpha\|^2$. The minimum of this quadratic form is given by $\alpha^* = (C^tC)^{-1}C^tx$ (see chapter 2), whence the result. Notice that C^tC is invertible iff C is of rank m, which we assumed[18]. □

Observe that $\pi_C^\perp(x) = (I - P)x$ is the projection on the orthogonal space of $sp\{C_1, ..., C_m\}$, *i.e.* the space defined by $C^tx = 0$. Observe also that $P = P^t$.

Affine equality constraints. Consider the problem $\min_x f(x)$ s.t. $\theta(x) = C^tx - c = 0$. Since we are only interested in the value of f on the affine space $\mathcal{D}$, let us replace the problem by $\min_x f(\pi_\mathcal{D}(x))$ where $\pi_\mathcal{D}$ is the orthogonal projection on $\mathcal{D}$. These two functions coincide on $\mathcal{D}$, and so have the same minimum. Observe however that $F(x) = f(\pi_\mathcal{D}(x))$ is now constant on the directions perpendicular to $\mathcal{D}$.

[18] When the m columns are not linearly independent, it suffices to replace the inversion by a pseudo-inverse to get the expression of the projector P. This can be check with the SVD of C for example.

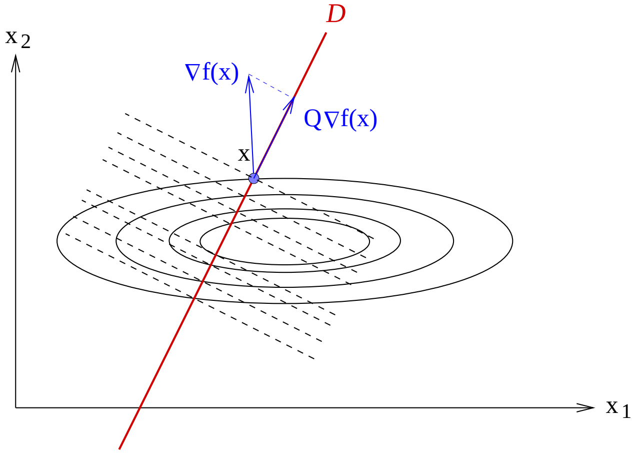

Figure 28: *Level lines of f and an affine domain $\mathcal{D}$ of $\mathbb{R}^2$ (in red). Dashed lines represent the corresponding level lines of F.*

The orthogonal projection on the vector space defined by $C^t x = 0$ is given by the projection matrix $Q = I - P$ (lemma 6). Let $x^0 \in \mathcal{D}$, then $x \in \mathcal{D}$ iff $C^t(x - x^0) = 0$, from which we deduce that the projection of x on $\mathcal{D}$ is given by $\pi_{\mathcal{D}}(x) = x^0 + \pi_C^{\perp}(x - x^0) = x^0 + Q(x - x^0)$. The problem becomes

$$\min_x F(x) = f[x^0 + Q(x - x^0)] \quad \text{s.t.} \quad C^t(x - x^0) = 0 \tag{134}$$

The constraint is now almost superfluous : given a local minimum x in $\mathbb{R}^d$, its projection on $\mathcal{D}$ will be a solution to our problem. So we can ignore the constraint. One has

$$\begin{aligned} \nabla F(x) &= Q \, \nabla f[x^0 + Q(x - x^0)] && (135) \\ \nabla^2 F(x) &= Q \, \nabla f[x^0 + Q(x - x^0)] \, Q && (136) \end{aligned}$$

which allows us to implement first order and second order methods. Observe that ∇F is obtained by projecting the gradient of f on the vector space $C^t x = 0$, whence the name of the method. As a consequence, a search method that starts at $x^0 \in \mathcal{D}$ will always stay in $\mathcal{D}$. This is why we can ignore the constraint. In the same way, the hessian is restricted to its projection on $C^t x = 0$, which makes it a singular matrix. But we can ignore this singularity in the formulae of the second order methods and replace inverses by pseudo-inverses.

Exercise 37 *Prove relations (135) and (136).*

Affine inequality constraints. Consider the problem $\min_x f(x)$ s.t. $\theta(x) = C^t x - c \leq 0$. Here the difficulty is the management of active constraints. This is best explained on an example.

Assume we are at point x (Fig. 29), where $\mathcal{A}(x) = \{1\}$, *i.e.* only θ_1 is active, and let us project the gradient $\nabla f(x)$ on the space defined by the gradients of active constraints, *i.e.* $\nabla\theta_1(x)$. To do so, we build matrix $C = [\nabla\theta_1(x)]$ and by

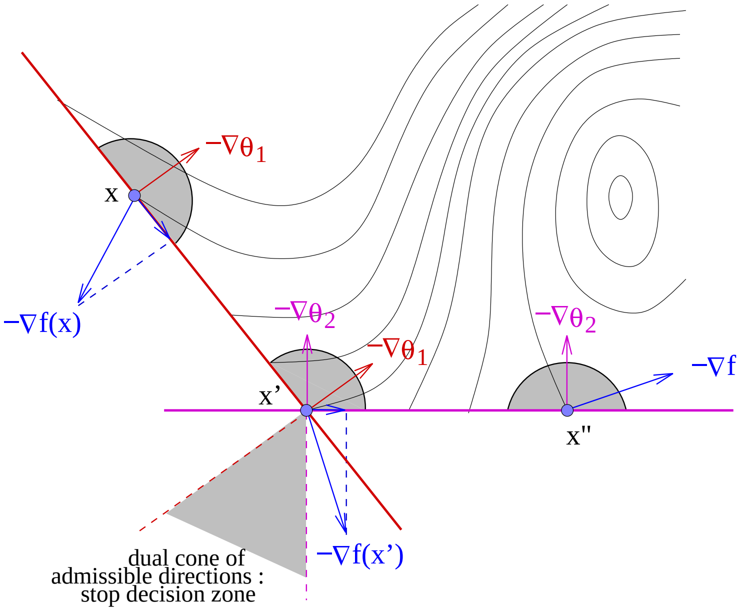

Figure 29: *Level lines of f and a domain $\mathcal{D}$ of $\mathbb{R}^2$ limited by affine lines (in red and purple). The gray sectors represent admissible directions.*

lemma 6 one gets $\pi_{\mathcal{A}(x)}[\nabla f(x)] = C\alpha$ where the vector of coefficients α is given by $\alpha = (C^tC)^{-1}C^t\,\nabla f(x)$. Here α is a negative scalar, which means that $-\nabla f(x)$ doesn't belong to the cone of admissible directions, therefore we have to project it on this cone, which means to project it on the manifold of active constraints. This projection is given by $\pi^{\perp}_{\mathcal{A}(x)}[\nabla f(x)] = \nabla f(x) - \pi_{\mathcal{A}(x)}[\nabla f(x)] = Q\,\nabla f(x)$ with $Q = I - C(C^tC)^{-1}C^t$. This gives a descent direction.

By constrast, consider point x'' where $\mathcal{A}(x'') = \{2\}$. At this point the coefficient α would be positive, which means that $-\nabla f(x'')$ points to a direction that also decreases $\theta_2(x)$. Therefore one can keep $-\nabla f(x'')$ as a descent direction and θ_2 must be removed from the set of active constraints.

Let us come back to point x. The linear search in the direction of the opposite to the projected gradient may stop in two cases.

1. either we have found a local minimum in this direction, and we have to select another descent direction (same as for unconstrained optimization),

2. or in our descent we are blocked by another constraint that becomes active.

This second case happens in the figure : the descent from x stops at point x' where θ_2 becomes active. We now have $\mathcal{A}(x') = \{1, 2\}$, and another direction must be selected. Let us again project $\nabla f(x')$ on the space defined by the gradients of active constraints, so here $C = [\nabla\theta_1(x'), \nabla\theta_2(x')]$. If the vector of coefficients α has only negative coordinates, then the Kuhn-Tucker conditions are met and we have a stationary point of f in $\mathcal{D}$ (this corresponds to $\nabla f(x')$ belonging to the dual cone in gray in the figure). Otherwise, some coordinates in α are positive, which corresponds to constraints that can be relaxed. Here θ_1 can be relaxed. We therefore take $\mathcal{A}(x') = \{2\}$ and reiterate the process of projecting the gradient, etc.

Non-linear constraints. The method of projected gradients can be extended to the case of non-linear constraints, where the projections are done on the tangent plane to the active constraints. However, one has to deal with the extra difficulty that following this tangent plane may lead outside $\mathcal{D}$. Therefore these methods must be coupled with a projection of the current point on $\mathcal{D}$.

4.3.3 Reduced gradient

In the simplex algorithm (linear programming), the optimization is performed by adjusting *a subset* of the coordinates of x, the *base variables*, the remaining ones being related to the former by the (affine) constraints of the problem. The reduced gradient method generalizes this idea to the non-linear case. We first examine conditions that allow to express some of the coordinates of x as functions of the others.

Implicit functions theorem. Assume domain $\mathcal{D}$ is a manifold defined by m equality constraints gathered in $\theta(x) = 0$ where $\theta : \mathbb{R}^n \to \mathbb{R}^m,\ m < n$. Without loss of generality, let us split x into $x = (u, v)$ where $u \in \mathbb{R}^{n-m}$ and $v \in \mathbb{R}^m$, and assume $x^0 = (u^0, v^0) \in \mathcal{D}$. We examine conditions that allow to express v as a function of u in $\mathcal{D}$.

Consider the $m \times m$ **Jacobian** matrix

$$\nabla_v \theta(x^0) \quad = \quad [\nabla_v \theta_1(x^0), ..., \nabla_v \theta_m(x^0)] \tag{137}$$

obtained by juxtaposing the *partial gradients* in v of the m constraint functions θ_j. If this matrix is invertible (*i.e.* if the partial gradients are linearly independent), there exists a small ball $\mathcal{B}(u^0, \epsilon) = \{u \in \mathbb{R}^{n-m} : \|u - u^0\| < \epsilon\}$ around u^0 and a function $\phi : \mathcal{B}(u^0, \epsilon) \to \mathbb{R}^m$ such that

$$x = (u, v) \in \mathcal{D} \quad \text{and} \quad u \in \mathcal{B}(u^0, \epsilon) \quad \Leftrightarrow \quad v = \phi(u) \tag{138}$$

In other words, domain $\mathcal{D}$ if defined by points $(u, \phi(u))$ around x^0.

Example : consider a single constraint $\theta(x) = x_1 - {x_2}^2 = 0$, and take $u = x_1, v = x_2$. One has $\nabla\theta(x) = [1, -2x_2]^t$, so $\nabla_v\theta(x) = -2x_2$. At $x^0 = (0, 0)$, this elementary Jacobian vanishes, so one can not express x_2 as a function of x_1 (given x_1, there is a positive and a negative x_2 that satisfies $\theta(x) = 0$). However, a plot of θ will convince the reader that one can locally express x_2 as a function of x_1 everywhere else on the curve.

Equality constraints. Given the m constraints, one can consider that in reality the optimization problem only has $n - m$ degrees of freedom. Using the implicit functions theorem, and by analogy with the simplex method, u will thus play the part of the base variables.

Let us define the reduced (unconstrained) problem $\min_u F(u)$ where $F(u) = f(u, \phi(u))$. Denoting $x^0 = (u^0, v^0)$ with $v^0 = \phi(u^0)$, the gradient of F at u^0 is given by[19]

$$\nabla F(u^0) \quad = \quad \nabla_u f(u^0, v^0) + \nabla\phi(u^0) \nabla_v f(u^0, v^0) \tag{139}$$

where $\nabla\phi(u^0) \in \mathbb{R}^{m\times m}$ is the **Jacobian** of ϕ, obtained by juxtaposing the gradients of the m scalar functions ϕ_j that compose ϕ: $\nabla\phi(u^0) = [\nabla\phi_i(u^0), ..., \nabla\phi_m(u^0)]$, and where $\nabla_u f$ and $\nabla_v f$ are partial gradients of f.

To be able to use this gradient in a descent scheme, one must determine the Jacobian of ϕ. This is simple in the case of **linear constraints** $\theta(x) = Cx + c = 0$, $C \in \mathbb{R}^{m\times n}$. Consider the partition $C = [B, A]$ where $B \in \mathbb{R}^{m\times(n-m)}$, $A \in \mathbb{R}^{m\times m}$. Without loss of generality, one can assume that A is invertible[20], which corresponds to the necessary condition in the implicit functions theorem. The constraints now write $Bu + Av + c = 0$, whence $v = -A^{-1}(Bu + c)$, and $\nabla\phi(u) = -B^t(A^{-1})^t$.

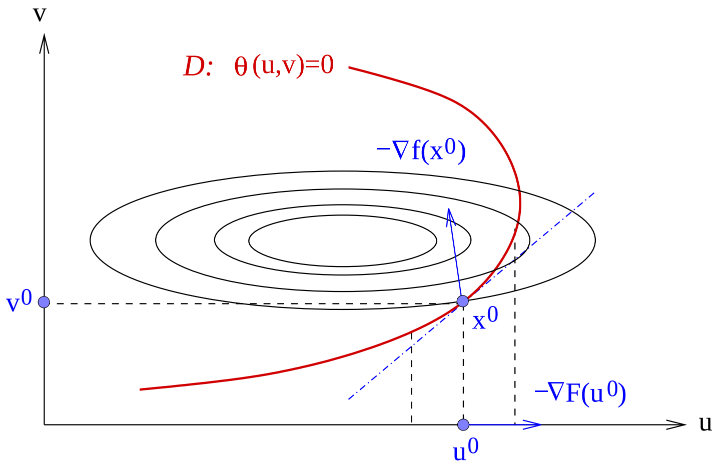

Figure 30: *Expressing* v *as a function of* u *in the neighborhood of* u^0 *allows to replace* $f(u, v)$ *by* $F(u) = f(u, \phi(u))$.

For **non-linear constraints**, the situation is slightly more complex. Assume $\nabla_v\,\theta(u^0, v^0)$ is invertible (up to a reordering of coordinates in x), which characterizes a *non degenerate* point x^0. The implicit functions theorem guarantees the existence of ϕ in a neighborhood of u^0. To determine $\nabla\phi$, we use the following trick : Let us consider the function $\Theta(u) = \theta(u, \phi(u)) = 0$, that implicitly defines ϕ. Its gradient is given as in (139) by

$$\nabla\Theta(u) \quad = \quad \nabla_u\,\theta[u, \phi(u)] + \nabla\phi(u)\nabla_v\,\theta[u, \phi(u)] \;=\; 0 \tag{140}$$

[19] To prove this relation, consider each $\frac{\partial F(u)}{\partial u_i}$ and use the formula for the derivative of composed functions.

[20] Otherwise, the invertibility of A is obtained after a reordering of the coordinates in x. If this is true for no reordering, this simply means that one of the affine constraints is redundant, or that the domain is empty (the verification of this statement is left as an exerise).

and vanishes for any value of u (in the neighborhood of u^0 where ϕ exists). This yields

$$\nabla\phi(u^0) \quad = \quad -\nabla_u\theta(u^0, v^0)[\nabla_v\theta(u^0, v^0)]^{-1} \tag{141}$$

which generalizes the expression $\nabla\phi(u^0) = -B^t(A^{-1})^t$ found in the linear case. This relation can now be injected in (139) to determine a descent direction.

This is not sufficient however since we also need to determine the values of F at a successive point $u = u^0 - t\nabla F(u^0)$, after a step in direction $-\nabla F(u^0)$, and therefore we need to compute $\phi(u)$. This amounts to adjusting the value of v^0 in $x = (u, v^0)$ in order to satisfy $\theta(x) = 0$. This operation, called the *projection* of x on the manifold, depends on the nature of constraints. It may also require a numerical resolution.

Inequality constraints. The method is similar to the approach by projected gradients. One considers active and inactive constraints. The reduction of the gradient is performed with respect to the active constraints. And the progression in one direction must also check that some constraints do not come active or inactive.

References

[1] Philippe G. Ciarlet, “Introduction à l’analyse numérique matricielle et à l’optimisation,” Ed. Masson, 1982.

[2] Claude Lemarechal, “Méthodes numériques d’optimisation,” INRIA, coll. didactique, 1989.

[3] Jean-Christophe Culioli, “Introduction à l’optimisation,” Ed. Ellipses, 1994.

[4] Michel Bierlaire, “Introduction a l’optimisation différentiable,” presses polytechniques et universitaires romandes.

A On the regularity assumption

What if constraint gradients are not linearly independent ? Here is one example that illustrates this importance of the regularity assumption for the Lagrange conditions to be valid.

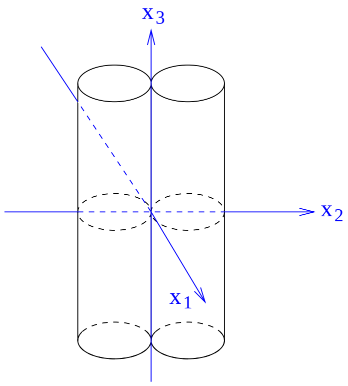

Figure 31: *All points of the domain defined by the intersection of these two cylinders are irregular.*

Consider constraints $\theta_1(x) = x_1^2+(x_2-1)^2-1 = 0$ and $\theta_2(x) = x_1^2+(x_2+1)^2-1 = 0$, that define two cylinders of radius 1, with the x_3 axis as intersection. At any point $x \in \mathcal{D}$ of the contact line, the gradient of θ_i is along the x_2 axis, so its tangent plane is defined by $x_2 = 0$. Since gradients are colinear (actually opposite one of another), x is not regular. And in effect there are directions of the tangent space that can not lead to a point in $\mathcal{D}$, *i.e.* that are not admissible. Consider for example moving in the direction of x_1 from the origin. This degenerate case happens when constraints $\theta_j(x) = 0$ are “tangent” at some point, and don’t intersect sharply.

B Proof of theorem 3 on second order conditions

Proof. As for the unconstrained case, the main idea is to consider the second order Taylor expansion of f around a stationary point. But since now x varies in a limited domain, this requires a second order characterization of the boundaries of the domain. The tangent space (first oder approximation) is not sufficient anymore.

Assume x varies along a curve in domain $\mathcal{D}$, which we express by a C^2 parametric curve $x :]-T,T[\to \mathcal{D} \subseteq \mathbb{R}^d$ where $x(0) = x^*$.

Let us first consider constraint θ_j, and define $\Theta_j(\tau) = \theta_j[x(\tau)] \equiv 0$ on $]-T,T[$. One has

$$\begin{aligned} \Theta_j'(\tau) &= \nabla\theta_j[x(\tau)]^t\, x'(\tau) \;\equiv\; 0 \qquad (142) \\ \Theta_j''(\tau) &= x'(\tau)^t\, \nabla^2\, \theta_j[x(\tau)]^t\, x'(\tau) + \nabla\theta_j[x(\tau)]^t\, x''(\tau) \;\equiv\; 0 \qquad (143) \end{aligned}$$

The first equation taken at $\tau = 0$ expresses that the "speed" $x'(0)$ of x is in the tangent space of the constraint θ_j at x^*. And the second one relates the speed of x to its acceleration x''. In the same manner, let us define $F :]-T,T[\to \mathbb{R}$ by $F(\tau) = f[x(\tau)]$. One has

$$\begin{aligned} F'(\tau) &= \nabla f[x(\tau)]^t\, x'(\tau) \qquad (144) \\ F''(\tau) &= x'(\tau)^t\, \nabla^2 f[x(\tau)]^t\, x'(\tau) + \nabla f[x(\tau)]^t\, x''(\tau) \qquad (145) \end{aligned}$$

Since the first equation vanishes at $\tau = 0$, we recover that $\nabla f(x^*)$ is orthogonal to the speed $x'(0)$, whatever the value it takes in the tangent space to the constraints at x^*. So we recover $\nabla f(x^*) + \sum_j \lambda_j^*\, \nabla\theta_j(x^*) = 0$ for some λ_j^*. Let us inject this expression in (145) taken at $\tau = 0$:

$$F''(0) \;=\; x'(0)^t\, \nabla^2 f(x^*)^t\, x'(0) - \sum_j \lambda_j^*\, \nabla\theta_j(x^*)^t\, x''(0) \qquad (146)$$

where the acceleration $x''(0)$ can be removed using (143) taken at $\tau = 0$. This yields

$$F''(0) \;=\; x'(0)^t\, [\, \nabla^2 f(x^*)^t\, x'(0) + \sum_j \lambda_j^*\, \nabla^2\, \theta_j(x^*)\,]\, x'(0) \qquad (147)$$

To conclude, consider the second order Taylor expansion of F at $\tau = 0$:

$$\begin{aligned} F(\tau) &= F(0) + \tau F'(0) + \frac{1}{2}\tau^2 F''(0) + o(\tau^2) \\ f[x(\tau)] &= f(x^*) + \frac{1}{2}\,\tau^2 F''(0) + o(\tau^2) \qquad (148) \end{aligned}$$

If x^* is a local minimum of f on $\mathcal{D}$, then necessarily 0 is a local minimum of F,*i.e.* $F''(0) \geq 0$. This in turn implies that (147) must be non-negative, *i.e.* that the quadratic form defined by the symmetric matrix $\nabla^2 f(x^*)^t\, x'(0) + \sum_j \lambda_j^*\, \nabla^2\, \theta_j(x^*)$ is positive for all possible values of the speed $x'(0)$, *i.e.* on the tangent space to the constraints at x^*. In the same way, the strict positivity of this expression is sufficient to state that x^* is a local minimum. □

Exercise 38 *Express in what space all the functions appearing above live, and check that matrix dimensions agree in all equations. Prove all the formula above for the expressions of derivatives.*