NUMERICAL OPTIMISATION

IDRISS MAZARI-FOUQUER, MAXIME CHUPIN

This document is to be used as a set of lecture notes — the lectures given might differ in places.

Contents

Date: February 16, 2026.

General introduction

The overall goal of this class is to provide a concise introduction to classical and modern methods in classical numerical optimisation. It goes without saying that such a class cannot be fully self-contained, and the students are expected to have basic working knowledge of the following fields of mathematics:

(1) Calculus and measure theory,
(2) Linear algebra,
(3) Dynamical systems,
(4) Basics of optimisation.

This class consists of two main parts: the first one covers the basics of numerical optimisation, with a strong emphasis on gradient and Newton methods. The second part of the class deals with more modern topics, and seeks to offer an overview of new perspectives in numerical optimisation.

None of the material presented here is original, and there are numerous good references available, both online (several exercises were drawn from the lectures of C. Royer, B. Bogosel, Y. Privat, D. Gontier, A. Frouvelle, F. Bach, R. Herbin, E. Herberg, etc.) and in the university library. In particular, I drew inspiration from the following sources:

References

[1] Arthur Albert. *Regression and the Moore-Penrose pseudoinverse*, volume 94 of *Math. Sci. Eng.* Elsevier, Amsterdam, 1972.

[2] Nicolas Boumal, Dmitriy Drusvyatskiy, and Quentin Rebjock. Gradient descent can converge to any isolated saddle point.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

[4] Guillaume Carlier. *Classical and modern optimization.* Adv. Textb. Math. Hackensack, NJ: World Scientific, 2022.

[5] J. E. jun. Dennis and Jorge J. More. Quasi-Newton methods, motivation and theory. *SIAM Rev.*, 19:46–89, 1977.

[6] Guillaume Garrigos and Robert M. Gower. Handbook of Convergence Theorems for (Stochastic) Gradient Methods. Preprint, arXiv:2301.11235 [math.OC] (2023), 2023.

[7] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an o(1/t) convergence rate for the projected stochastic subgradient method, 2012.

[8] Svein Linge and Hans Petter Langtangen. *Programming for computations – Python. A gentle introduction to numerical simulations with Python 3.6*, volume 15 of *Texts Comput. Sci. Eng.* Cham: Springer, 2nd revised and extended edition edition, 2020.

1. A primer on gradient descent

1.1. Introduction: basic concepts, optimality conditions

Throughout this entire chapter, unless stated otherwise, we consider a fixed, $\mathscr{C}^2$ function $f:\mathbb{R}^d\to\mathbb{R}$.

1.1.1. First definitions

The goal of this section is to fix the terminology, as well as some notations.

Definition 1.1. *A point $x^*\in\mathbb{R}^d$ is called:*

(1) A global minimiser of f *if*

$$\forall x\in\mathbb{R}^d\,,f(x^*)\leqslant f(x).$$

(2) A local minimiser of f *if*

$$\exists\varepsilon>0\,,\forall x\in\mathbb{R}^d\,,\|x-x^*\|\leqslant\varepsilon\Rightarrow f(x^*)\leqslant f(x).$$

In the remainder of this document, we will adopt the following notational conventions:

(1) $M_d(\mathbb{R})$ denotes the set of $d\times d$ matrices, $M_{p,q}(\mathbb{R})$ denotes the set of $p\times q$ matrices.
(2) $S_d(\mathbb{R})$ denotes the set of symmetric matrices in $M_d(\mathbb{R})$. The transpose of a matrix M is written M^T.
(3) $S_d^+(\mathbb{R})$, resp. $S_d^{++}(\mathbb{R})$, resp. $S_d^-(\mathbb{R})$, resp. $S_d^{--}(\mathbb{R})$ denotes the set of symmetric positive, resp. definite positive, resp. negative, resp. symmetric negative, matrices.

1.1.2. The optimisation problem under consideration

The goal of this class is to study the optimisation problem

$$\min_{x\in\mathbb{R}^d}f(x). \tag{1.1}$$

A first question is whether or not a solution x^* actually exists. To this end, let us recall the definition of *coercivity*:

Definition 1.2. *We say that a continuous function f is* coercive *if for any $M\in\mathbb{R}$ the sub-level set $\{x\in\mathbb{R}^d:\ f(x)\leqslant M\}$ is bounded.*

It is a classical result (see Exercise 1.2) that if f is coercive, then the optimisation problem (1.1) has a solution $x^*\in\mathbb{R}^d$. With a slight abuse of terminology, we will say that x^* solves (1.1), and dub it a *minimiser* of f in $\mathbb{R}^d$.

Now, consider (1.1), and assume that x^* is a solution. Naturally, one would like to have either an explicit or a good enough numerical approximation of the minimiser x^*. Unless we are quite lucky and an easy comparison argument provides an explicit value, this is a hopeless endeavour. The best we can do is to rely on *optimality conditions*. Optimality conditions allow to reduce the search of a minimiser to the resolution of a non-linear system of equations.

There are two optimality conditions. The first-order optimality condition reads

$$\nabla f(x^*)=0 \tag{1.2}$$

while the second-order optimality condition writes

$$\nabla^2 f(x^*)\in S_d^+(\mathbb{R}).$$

Assume for simplicity that (1.2) has a finite number of solutions $x_1, \dots, x_N$, which have tractable expressions. This still does not provide any conclusion, and we need to compute the Hessian of f at each x_k, $k = 1, \dots, N$. There are several possibilities:

1. If $\nabla^2 f(x_k) \in S_d^{++}(\mathbb{R})$, then x_k is a local minimiser of f.
2. If $\nabla^2 f(x_k) \in S_d^{--}(\mathbb{R})$, then x_k is a local maximiser of f.
3. If $\nabla^2 f(x_k)$ has at least one negative and one positive eigenvalue, x_k is a *saddle* point: there exist two orthogonal directions $\vec{e}_1\,, \vec{e}_2$ such that $t^* = 0$ is a local minimiser of $t \mapsto f(x+te_1)$, and a local maximiser of $t \mapsto f(x+te_2)$.
4. If $\nabla^2 f(x_k) \in S_d^{+}(\mathbb{R})$, but not in $S_d^{++}(\mathbb{R})$, then we cannot conclude and further analysis is required.

The proof of this result is very quick; we refer to Exercise 1.4. The exercises of this chapter (in particular exercise 1.5) contain several examples of optimisation problems that can be solved by hand. Such examples are usually limited to dimension 2 or 3, unless the problem has a very specific structure.

For the sake of future references, let us single out the following definition:

Definition 1.3. *Let $f \in \mathscr{C}^1(\mathbb{R}^d;\mathbb{R})$. A point $x^* \in \mathbb{R}^d$ is called a* critical point *of f if*

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

In the first part of this class, we will focus on *finding critical points*.

Most real-life applications of optimisation cannot be tackled analytically, and one must be satisfied with good numerical approximation of the optimisers. In this class, we will be focusing on *iterative methods*, that is, methods which can be written as

$$\begin{cases} \text{Start from an initial guess } x_0, \\ \text{Supposing } x_0, \dots, x_k \text{ are built, set } x_{k+1} = x_k + G_k(x_k) \end{cases}$$

for some function G_k, the definition of which might depend on the previous iterates $x_0, \dots, x_k$. Most of the time, this will not be the case, and the iteration map G_k will not depend on the index k. The goal is to obtain algorithms that produce sequences that converge at a “good enough” rate — most of the time, we will be satisfied with *linear convergence*, in the following sense:

Definition 1.4. *Let $\{x_k\}_{k\in\mathbb{N}} \in (\mathbb{R}^d)^{\mathbb{N}}$ and $x^* \in \mathbb{R}^d$. We say that $\{x_k\}_{k\in\mathbb{N}}$ converges linearly, at rate $\alpha \in [0;1)$, to x^*, if there exists a constant C such that*

$$\forall k \in \mathbb{N}\,, \|x_k - x^*\| \leqslant C\alpha^k.$$

At any rate, in order to tackle (1.1), we will rely on the true and tried method of *gradient descent* and its variants. Recall that *our main objective, from now on, is to locate critical points of f.*

1.2. Definition of the gradient descent and basic properties

The gradient descent is a *local* algorithm that essentially relies on a Taylor expansion of the function f: assume that you are starting from an initial guess $x_0 \in \mathbb{R}^d$, and you want to solve (1.1). We look for a point that is close enough to x_0, say at distance at most d_0, and such that $f(x_1) < f(x_0)$ (if that is possible). In that case, a natural idea is to replace f by its first-order Taylor approximation

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

so that, at first order, we are solving the minimisation problem

$$\min_{\|x-x_0\|\leqslant d_0} \langle \nabla f(x_0), x-x_0\rangle. \tag{1.3}$$

At this stage, two things may happen:

(1) Either the gradient vanishes ($\nabla f(x_0)=0$), in which case we stop, as we are satisfied with what we already have. Now, if we wanted to go further in the analysis, we should note that two other possibilities arise: either f is convex, in which case this implies that x_0 is a global minimiser of the function f, or f is not convex and we would need to do something different to investigate the local optimality of x_0. This will very often not be the case, and we will see, in Chapter **??**, why this is not a problem in practice.

(2) Either the gradient does not vanish, so that the pseudo-optimisation problem (1.3) has a unique solution

$$x_1=-\frac{d_0}{\|\nabla f(x_0)\|}\nabla f(x_0).$$

Now, the question remains of choosing the parameter d_0. Of course, if we already know that the gradient is small enough in norm, it makes no sense to look for a point that would be far away, and this naturally leads to choosing d_0 as $d_0=\tau\|\nabla f(x_0)\|$ for some $\tau>0$.

Overall, we define the sequence of iterates of the gradient descents as follows:

$$\begin{cases} x_0\in\mathbb{R}^d\,, \\ \forall k\in\mathbb{R}^d\,, x_{k+1}=x_k-\tau\nabla f(x_k). \end{cases}$$

The main questions under consideration from now on are:

(1) The **convergence** of the generated sequence $\{x_k\}_{k\in\mathbb{N}}$.
(2) The **convergence of the sequence of values** $\{f(x_k)\}_{k\in\mathbb{N}}$.
(3) The **convergence of the gradient of the objective function** $\{\nabla f(x_k)\}_{k\in\mathbb{N}}$.

Of course, the convergence of $\{x_k\}_{k\in\mathbb{N}}$ implies the convergence of the values and of the gradient; the convergence of the values, on the other hand, does not imply the convergence of the sequence itself. It is also important to note that, in general, the presentation of gradient descent assumes, from the get-go, some strong convexity of f, which gives a positive answer to all the questions above. On the other hand, it is extremely important, both in practice and in theory, to distinguish these different steps and this is what we will do. At any rate, here is a simple result:

Proposition 1.1. *Assume that $f\in\mathscr{C}^1$ and that the gradient descent with fixed step size τ converges in the sense that $\{x_k\}_{k\in\mathbb{N}}$ converges to some x^*. Then $\nabla f(x^*)=0$.*

Proof of Proposition 4.1. If the sequence converges then, passing to the limit in $x_{k+1}=x_k-\nabla f(x_k)$ yields $\nabla f(x^*)=0$. □

Of course the next question is, if we assume that $\{x_k\}_{k\in\mathbb{N}}$ converges to some x^*, is it true that x^* is, in fact, a minimiser of f? The answer is no in general. Consider for instance the function

$$f:x\mapsto\frac{x^3}{3},$$

a fixed $1>\tau>0$ and an initialisation $x_0=1$. The sequence of iterates of the gradient descent is given by

$$\forall k\in\mathbb{N}\,, x_{k+1}=x_k(1-\tau x_k).$$

Now, if $\tau \in (0;1)$, a simple reasoning by induction shows that the sequence $\{x_k\}_{k\in\mathbb{N}}$ is positive and decreasing; in particular it is converging so that by Proposition 4.1 it converges to 0, which is not a minimiser of f. Of course, one might argue that this is cheating, as the function f is not coercive. Nevertheless, it is easy to adapt this example: simply modify f on $(-\infty;-1]$ to have a globally smooth, coercive function.

1.2.1. **Convergence of the gradient descent: first considerations**

In this first paragraph, we investigate in a formal manner the constraints we should put on the step size and on the function f to obtain a converging sequence, where the parameters should be chosen uniformly with respect to the initial condition. We begin with the regularity of the function. Let us consider the case of a $\mathscr{C}^1$, but not $\mathscr{C}^{1,1}$ function, for instance, in two variables

$$f:(x,y)\mapsto \frac{2}{3}\left(x^2+2y^2\right)^{\frac{3}{4}}.$$

It is fairly easy to show that the function f is $\mathscr{C}^1$, but not $\mathscr{C}^{1,1}$ at 0 (this is left as an exercice): simply observe that

$$|f(x,y)|\leqslant C\|(x,y)\|^{\frac{3}{2}}.$$

Furthermore, 0 is the unique minimiser of f. For a given parameter $\tau>0$, the sequence of iterates is given explicitly by

$$\begin{cases} x_{k+1}=x_k\left(1-\frac{\tau}{(x_k^2+2y_k^2)^{\frac{1}{4}}}\right) \\ y_{k+1}=y_k\left(1-\frac{2\tau}{(x_k^2+2y_k^2)^{\frac{1}{4}}}\right). \end{cases}$$

Although this example will be studied more in detail (or rather, illustrated) in the computer sessions, observe that, at a formal level, if the sequence converges, then it must converge to 0. Thus, we “should” be able to write that

$$(x_{k+1},y_{k+1})\sim\frac{\tau}{(x_k^2+2y_k^2)^{\frac{1}{4}}}\left(-x_k,-2y_k\right).$$

Defining

$$z_k:=x_k^2+2y_k^2$$

we deduce that (asymptotically)

$$z_{k+1}\geqslant Cz_k^{\frac{1}{2}}\,,C=\tau^2.$$

Now, let us assume that this inequality is, in fact, satisfied for all $k\in\mathbb{N}$. This would give the lower bound

$$z_{k+1}\geqslant CC^{\frac{1}{2}}z_{k-1}^{\frac{1^2}{2^2}}\geqslant\cdots\geqslant C^{\sum_{i=0}^k\left(\frac{1}{2}\right)^i}z_0^{2^{-k}},$$

and cannot converge to 0.

We continue with an investigation of the step size; here the computations are much easier, as it suffices to consider, in the one-dimensional case, the function

$$f:x\mapsto\frac{\mu}{2}x^2.$$

Then, for any initialisation x_0 and any fixed step size $\tau>0$, the sequence of iterates is given by

$$x_{k+1}=x_k(1-\mu\tau)=x_0(1-\mu\tau)^k.$$

Thus, the method converges if, and only if, $0 < \tau < \frac{1}{\mu}$. As μ quantifies the steepness of f', or, put otherwise, the average variation of the gradient, we fairly easily understand that the wilder the gradient of a function, the smaller the step size needs to be.

1.2.2. **The gradient descent is a descent method**

In this section, we consider a function $f \in \mathscr{C}^1(\mathbb{R}^d)$ with a μ-Lipschitz gradient in the sense that

$$\forall x\,,y \in \mathbb{R}^d\,, \|\nabla f(x) - \nabla f(y)\| \leqslant \mu\|x-y\|. \tag{1.4}$$

We do not make any assumption on the coercivity of f, or on the existence of a minimiser. Our first result is the following:

Theorem 1.1. *For any $x_0 \in \mathbb{R}^d$, for any $\tau > 0$, the sequence generated by the gradient descent initialised at x_0 with step size τ satisfies*

$$\forall k \in \mathbb{N}\,, f(x_{k+1}) - f(x_k) \leqslant \tau\left(\tau\mu - 1\right)\|\nabla f(x_k)\|^2.$$

In particular, if $\tau \in \left(0; \frac{1}{\mu}\right)$ then the sequence $\{f(x_k)\}_{k\in\mathbb{N}}$ is strictly decreasing unless it is stationary. Finally, for any $\tau \in \left(0; \frac{1}{2\mu}\right)$ there holds

$$\forall k \in \mathbb{N}\,, f(x_{k+1}) - f(x_k) \leqslant -\frac{\tau}{2}\|\nabla f(x_k)\|^2.$$

Proof of Theorem 1.1. It suffices to write that for any $k \in \mathbb{N}$ there holds

$$f(x_{k+1}) = f(x_k - \tau\nabla f(x_k)).$$

From the mean-value theorem, there exists $\xi \in \mathbb{B}(x_k; \|x_{k+1} - x_k\|)$ such that

$$f(x_{k+1}) = f(x_k) + \langle \nabla f(\xi), -\tau\nabla f(x_k)\rangle.$$

This rewrites

$$\begin{aligned} f(x_{k+1}) - f(x_k) &= -\langle \nabla f(\xi) - \nabla f(x_k), \tau\nabla f(x_k)\rangle - \tau\|\nabla f(x_k)\|^2 \\ &\leqslant \tau\|\nabla f(\xi) - \nabla f(x_k)\| \cdot \|\nabla f(x_k)\| - \tau\|\nabla f(x_k)\|^2 \\ &\leqslant \tau\mu\|x_{k+1} - x_k\| \cdot \|\nabla f(x_k)\| - \tau\|\nabla f(x_k)\|^2 \\ &= \tau\left(\tau\mu - 1\right)\|\nabla f(x_k)\|^2. \end{aligned}$$

The conclusion follows. □

We highlight once again that we did not require any information other than the regularity of ∇f.

1.2.3. **Convergence of the gradient descent II: the Zoutendijk theorem**

In this section, we seek to answer the question:*do the gradients converge?* As it turns out, they do! Provided, once again, the step size τ is chosen properly enough. Here again, we let $\mu > 0$ be such that (1.4) is satisfied.

Theorem 1.2. *Assume that f is bounded from below and that $\tau \in \left(0; \frac{1}{2\mu}\right)$. Then there holds*

$$\sum_{k=0}^{\infty} \|\nabla f(x_k)\|^2 < \infty.$$

In particular, $\{\nabla f(x_k)\}_{k\in\mathbb{N}}$ converges to 0.

Proof of Theorem 1.2. An easy consequence of Theorem 1.1 is that

$$\forall k \in \mathbb{N}\,, f(x_k) - f(x_{k+1}) \geqslant \frac{\tau}{2}\|\nabla f(x_k)\|^2.$$

Summing the previous inequalities yields

$$\forall N \in \mathbb{N}\,, \frac{\mu}{2}\sum_{k=0}^N \|\nabla f(x_k)\|^2 \leqslant f(x_0) - f(x_{N+1}).$$

Thus,

$$\mu \sum_{k=0}^N \|\nabla f(x_k)\|^2 \leqslant 2\left(f(x_0) - \inf f\right).$$

The conclusion follows. □

This still does not allow to conclude regarding the convergence of the sequence itself. We refer to Exercise 1.8.

1.2.4. **Convergence of the gradient descent III: convex functions**

Recall the definition of a convex function:

Definition 1.5. *We say that a function $\varphi : \mathbb{R}^d \to \mathbb{R}$ is* convex *if*

$$\forall x\,, y \in \mathbb{R}^d\,, \forall t \in [0;1]\,, \varphi((1-t)x + ty) \leqslant (1-t)\varphi(x) + t\varphi(y).$$

We say that φ is strictly convex *if*

$$\forall x \neq y \in \mathbb{R}^d\,, \forall t \in (0;1)\,, \varphi((1-t)x + ty) < (1-t)\varphi(x) + t\varphi(y).$$

A classical result states the following:

Proposition 1.2. *Let $\varphi \in \mathscr{C}^1(\mathbb{R}^d;\mathbb{R})$. Then φ is convex (resp. strictly convex) if, and only if, its gradient is monotone:*

$$\forall x\,, y \in \mathbb{R}^d\,, \langle \nabla\varphi(x) - \nabla\varphi(y), x - y\rangle \geqslant 0\,,$$

resp. strictly monotone:

$$\forall x \neq y \in \mathbb{R}^d\,, \langle \nabla\varphi(x) - \nabla\varphi(y), x - y\rangle > 0.$$

In addition, if $\varphi \in \mathscr{C}^2(\mathbb{R}^d;\mathbb{R})$ and is convex, then for any $x \in \mathbb{R}^d$ $\nabla^2\varphi(x) \in S_d^+(\mathbb{R})$. If, for any $x \in \mathbb{R}^d$, $\nabla^2\varphi(x) \in S_d^{++}(\mathbb{R})$, then φ is strictly convex.

We now make one stronger assumption on the function f. Namely, we assume that f still satisfies (1.4) for some constant $\mu > 0$ and that f is convex.

Theorem 1.3. *Assume that f is convex and satisfies (1.4) for some $\mu > 0$. Finally, assume that f has a minimiser x^*. For any $\tau \in \left(0; \frac{1}{2\mu}\right)$, for any initialisation x_0, the gradient descent with fixed step size τ, initialised at x_0, satisfies*

$$\forall k \in \mathbb{N}\,, f(x_{k+1}) - f(x^*) \leqslant \frac{f(x_0) - f(x^*)}{k+1}.$$

Proof of Theorem 1.3. Recall that from Theorem 1.1 we have

$$\forall k \in \mathbb{N}\,, f(x_{k+1}) - f(x_k) \leqslant -\frac{\tau}{2}\|\nabla f(x_k)\|^2.$$

However, by convexity of f,

$$f(x_k) \leqslant f(x^*) + \langle \nabla f(x_k), x_k - x^*\rangle.$$

Consequently,

$$\begin{aligned}
f(x_{k+1}) &\leqslant f(x_k) - \frac{\tau}{2}\|\nabla f(x_k)\|^2 \\
&\leqslant f(x^*) + \langle \nabla f(x_k), x_k - x^*\rangle - \frac{\tau}{2}\|\nabla f(x_k)\|^2 \\
&= f(x^*) + \frac{2}{\tau}\left(\frac{\tau}{2}\langle \nabla f(x_k), x_k - x^*\rangle - \frac{\tau^2}{4}\|\nabla f(x_k)\|^2\right) \\
&= f(x^*) - \frac{2}{\tau}\left(\|\frac{\tau}{2}\nabla f(x_k) - \frac{1}{2}(x_k - x^*)\|^2 - \frac{1}{4}\|x_k - x^*\|^2\right) \\
&= f(x^*) - \frac{2}{\tau}\left(\frac{1}{4}\|x_{k+1} - x^*\|^2 - \frac{1}{4}\|x_k - x^*\|^2\right).
\end{aligned}$$

We thus deduce that

$$k\left(f(x_k) - f(x^*)\right) \leqslant \sum_{i=1}^{k}(f(x_i) - f(x^*)) \leqslant \frac{1}{2\tau}\|x_0 - x^*\|^2.$$

The conclusion follows.

□

1.2.5. **Convergence of the gradient descent IV: quadratic functions**

We saw in the previous paragraph that, in the case of convex functions, we could get a convergence rate (algebraic, as it turns out) for the gradient descent. The goal of this section is to provide a finer convergence rate in the special case of quadratic functions.

Definition 1.6. *We say that a function $f : \mathbb{R}^d \to \mathbb{R}$ is quadratic if there exists $A \in M_d(\mathbb{R})$ and $b \in \mathbb{R}^d$ such that*

$$\forall x \in \mathbb{R}^d\,, f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

When f is quadratic, we say that f is represented by (A, b).

A straightforward computation shows that

$$\nabla f(x) = \frac{A + A^T}{2}x - b.$$

In particular, when A is symmetric,

$$\nabla f(x) = Ax - b$$

and x^* is a critical point of f if, and only if, x^* is a solution to $Ax^* = b$.

Theorem 1.4. *Let $A \in S_d^{++}(\mathbb{R})$, $b \in \mathbb{R}^d$ and f be the quadratic function represented by (A, b). Letting $0 < \lambda_1 \leqslant \lambda_d(A)$ be the eigenvalues of A, for any $\tau \in \left(0; \frac{2}{\lambda_d(A)}\right)$, for any $x_0 \in \mathbb{R}^d$, the gradient descent initialised at x_0 with fixed step size $\tau > 0$ converges linearly to the unique solution of $Ax^* = b$ and, more specifically,*

$$\forall k \in \mathbb{N}\,, \|x_k - x^*\| \leqslant \alpha(\tau)^k\|x_0 - x^*\|$$

with $\alpha(\tau) = \max_{i=1,\dots,d}|1 - \tau\lambda_i(A)|$. Finally,

$$\min_{\tau\in\left(0;\frac{2}{\lambda_d(A)}\right)}\alpha(\tau) = \alpha\left(\frac{1}{\lambda_1(A) + \lambda_d(A)}\right) = \frac{\mathrm{cond}(A) - 1}{\mathrm{cond}(A) + 1} \textit{ with } \mathrm{cond}(A) = \frac{\lambda_d(A)}{\lambda_1(A)}.$$

Proof of Theorem 1.4. Observe that as $A \in S_d^{++}(\mathbb{R})$ all the eigenvalues of A are positive. Furthermore, A induces a bijection, whence x^* is uniquely defined. Additionally, as A is symmetric, $\nabla f(x) = Ax - b$. Now we explicitly obtain, for any $k \in \mathbb{N}$,

$$x_{k+1} = x_k - \tau A x_k + \tau b$$

so that, defining $y_k := x_k - x^*$,

$$\forall k \in \mathbb{N}\,, y_{k+1} = y_k - \tau A x_k + \tau b = y_k - \tau A(x_k - x^*) = (\mathrm{Id} - \tau A) y_k.$$

The matrix $\mathrm{Id} - \tau A$ has eigenvalues $1 - \tau\lambda_d(A) \leqslant \cdots \leqslant 1 - \tau\lambda_1(A)$We deduce that

$$\forall k \in \mathbb{N}\,, \|y_{k+1}\| = \|(\mathrm{Id} - \tau A) y_k\| \leqslant \|\mathrm{Id} - \tau A\|_{\mathrm{op}} \cdot \|y_k\|.$$

Here, we used the operator norm on $\mathrm{Id} - \tau A$. By a straightforward iteration argument we deduce that

$$\forall k \in \mathbb{N}\,, \|y_k\| \leqslant \|\mathrm{Id} - \tau A\|_{\mathrm{op}}^k \|y_0\|.$$

However,

$$\|\mathrm{Id} - \tau A\|_{\mathrm{op}} = \max_{i=1,\dots,d} |1 - \tau\lambda_i(A)|.$$

We refer to Exercise 1.3. In particular, if $\tau > 0$ is chosen so that

$$\alpha(\tau) := \max_{i=1,\dots,d} |1 - \tau\lambda_i(A)| < 1 \tag{1.5}$$

we obtain

$$\forall k \in \mathbb{N}\,, \|x_k - x^*\| \leqslant \alpha(\tau)^k \|x_0 - x^*\|.$$

It remains to pick $\tau > 0$ so that $\alpha(\tau) < 1$. However

$$\alpha(\tau) < 1 \Leftrightarrow -1 < 1 - \tau\lambda_d(A) \leqslant 1 - \tau\lambda_1(A) < 1$$

which rewrites, in a compact form, as

$$\tau < \frac{2}{\lambda_d(A)}.$$

The conclusion follows. Finally, it is an easy exercise to see that α is minimised at τ^* such that

$$|1 - \tau^*\lambda_1(A)| = |1 - \tau^*\lambda_d(A)|.$$

Solving this equation explicitly in τ^* yields

$$\tau^* = \frac{2}{\lambda_1(A) + \lambda_d(A)}\,,\ \text{whence } \alpha(\tau^*).$$

□

Let us observe that the convergence rate of gradient descent is quantified by the conditioning number of the matrix A: if $\mathrm{cond}(A) \approx 1$ then the method converges extremely quickly if τ^* is chosen properly, while, if $\mathrm{cond}(A) \gg 1$ (which means that A, as a linear map, dilates much more in certain directions than in others), the method will *a priori* converge extremely slowly. It is important to have basic reflexes regarding the conditioning number of matrix. We refer to Exercise 1.14.

1.2.6. Convergence of the gradient descent V: the case of strongly convex functions

The purpose of this section is to generalise the results of the previous paragraph to the case of strongly convex functions.

Definition 1.7. *Let $\alpha > 0$. A function f is said to be* α-strongly convex *if*

$$\forall x\, y \in \mathbb{R}^d\,, \forall t \in [0;1]\,,$$
$$f((1-t)x+ty) \leqslant (1-t)f(x)+tf(y)-\alpha t(1-t)\|x-y\|^2.$$

Additionally, observe the following facts (see Exercise 1.12) if f is α-strongly convex and $\mathscr{C}^1$ then

$$\forall x\,, y \in \mathbb{R}^d\,, \langle \nabla f(x)-\nabla f(y), x-y\rangle \geqslant \alpha\|x-y\|^2. \tag{1.6}$$

Second, if ∇f is μ-Lipschitz and is minimal at x^*, then

$$\forall x \in \mathbb{R}^d\,, f(x)-f(x^*) \geqslant \frac{1}{2\mu}\|\nabla f(x)\|^2. \tag{1.7}$$

For this second inequality, we refer to Exercise 1.11.

With this inequality at hand, we can finally prove the following result:

Theorem 1.5. *Let f be a α-strongly convex, coercive, $\mathscr{C}^1$ function with a μ-Lipschitz gradient. Let x^* be the minimiser of f. Then, for any $x_0 \in \mathbb{R}^d$, for any $\tau \in \left(0; \frac{1}{2\mu}\right)$, the gradient descent initialised at x_0 with fixed step size τ converges linearly to x^* and, more precisely, we have*

$$\forall k \in \mathbb{N}\,, \|x_k-x^*\| \leqslant (1-\alpha\tau)^{\frac{k}{2}}\|x_0-x^*\|.$$

Proof of Theorem 1.5. We observe that, setting $y_k := x_k - x^*$, we have

$$\forall k \in \mathbb{N}\,, y_{k+1} = y_k - \tau\left(\nabla f(x_k)-\nabla f(x^*)\right).$$

Taking the squared norm on each side of this identity yields

$$\|y_{k+1}\|^2 = \|y_k\|^2 + \tau^2\|\nabla f(x_k)\|^2 - 2\tau\langle \nabla f(x_k)-\nabla f(x^*), x_k-x^*\rangle$$

Now observe that

$$\frac{\alpha}{2}\|x_k-x^*\|^2 + \langle \nabla f(x_k), x^*-x_k\rangle \leqslant f(x^*)-f(x_k)$$

so that

$$\|y_{k+1}\|^2 \leqslant (1-\alpha\tau)\|y_k\|^2 + 2\tau(f(x^*)-f(x_k)) + \tau^2\|\nabla f(x_k)\|^2$$

From (1.7)

$$\tau^2\|\nabla f(x_k)\|^2 \leqslant 2\tau^2\mu(f(x_k)-f(x^*)).$$

Consequently

$$\|y_{k+1}\|^2 \leqslant (1-\alpha\tau)\|y_k\|^2 + 2\tau\left(1-\tau\mu\right)\left(f(x^*)-f(x_k)\right) \leqslant (1-\alpha\tau)\|y_k\|^2.$$

□

1.3. **The Polyak-Lojasiewicz condition**

In this section, we finally consider the most general class of functions for which we can provide guarantee and estimate the convergence for the gradient descent. Such functions are characterised by an inequality:

Definition 1.8. *Let $f \in \mathscr{C}^1(\mathbb{R}^d;\mathbb{R})$ be bounded from below and $\alpha > 0$. We say that f satisfies the* Polyak-Lojasiewicz condition *with constant α if*

$$\forall x \in \mathbb{R}^d\,, f(x) - \inf_{\mathbb{R}^d} f \leqslant \frac{1}{2\alpha}\|\nabla f(x)\|^2.$$

This notion is strictly weaker than α-strong convexity. Indeed, we will see in Exercise 1.13 the following two facts:

(1) If f is α-strongly convex, then f satisfies a Polyak-Lojasiewicz inequality with constant α.
(2) There exist functions f satisfying a Polyak-Lojasiewicz condition that are not strictly convex.

The main takeaway of this section is that it suffices that the Polyak-Lojasiewicz condition is satisfied to ensure a proper behaviour of the gradient descent. More specifically, we have the following:

Theorem 1.6. *Let $f \in \mathscr{C}^1(\mathbb{R}^d;\mathbb{R})$ be bounded from below, satisfy the Polyak-Lojasiewicz condition with constant α and such that ∇f is μ-Lipschitz. For any $\tau \in \left(0;\frac{1}{\mu}\right)$ and any $x_0 \in \mathbb{R}^n$, letting $\{x_k\}_{k\in\mathbb{N}}$ be the sequence generated by the gradient descent initialised at x_0 with fixed step size τ, we have*

$$\forall k \in \mathbb{N}\,, f(x_{k+1}) - \inf f \leqslant (1-\tau\alpha)^{k+1}(f(x_0) - \inf f).$$

Proof of Theorem 1.6. First observe that

$$\begin{aligned} f(x_{k+1}) &= f(x_k) + \int_0^1 \langle \nabla f((1-t)x_k + tx_{k+1}), x_{k+1} - x_k\rangle dt \\ &\leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{1}{2}\mu\|x_{k+1} - x_k\|^2 \\ &= f(x_k) - \tau\|\nabla f(x_k)\|^2 + \frac{\mu}{2}\|x_{k+1} - x_k\|^2 \\ &= f(x_k) - \tau\|\nabla f(x_k)\|^2 + \frac{\mu}{2}\tau^2\|\nabla f(x_k)\|^2 \\ &\leqslant f(x_k) + \tau(-1 + \frac{\mu}{2}\tau)\|\nabla f(x_k)\|^2. \end{aligned}$$

Since $\tau\mu < 1$, $(-1 + \frac{\mu\tau}{2}) \leqslant \frac{1}{2}$ whence

$$f(x_{k+1}) \leqslant f(x_k) - \frac{\tau}{2}\|\nabla f(x_k)\|^2.$$

By the Polyak-Lojasiewicz inequality this yields

$$f(x_{k+1}) \leqslant f(x_k) - \alpha\tau(f(x_k) - \inf f) = (1-\alpha\tau)(f(x_k) - \inf f) + \inf f.$$

A straightforward iteration yields the conclusion. □

1.4. **The least square method**

1.4.1. **Presentation of the method**

A very important class of optimisation problems consist of *parameter identification* from observations. These problems can be found in a range of fields, including, most recently, machine learning and neural networks. While there are several efficient frameworks to tackle these problems from the numerical perspective, in this class, we will mostly focus on least square techniques, briefly presenting Ordinary Least Square (OLS) problems and Nonlinear Least Square. At any rate, the problem always reads as follows: letting $\Phi : \mathbb{R}^d \to \mathbb{R}^N$ be a (potentially non-linear) map and letting $Y \in \mathbb{R}^N$ be a fixed vector (the observation) solve

$$\min_{\theta\in\mathbb{R}^d} \frac{1}{2}\|\Phi(\theta) - Y\|^2. \tag{1.8}$$

In the upcoming paragraphs we will distinguish between cases where Φ is linear, and where it is truly non-linear (as is the case, typically in neural networks). There will not really be any proofs, mostly a discussion. We refer to the computer sessions for practical illustrations.

1.4.2. **Ordinary least squares**

The first case of importance, called Ordinary Least Squares (sometimes called linear regression), corresponds to a situation where there exists a matrix $Z \in M_{d,N}(\mathbb{R})$ such that

$$\forall \theta \in \mathbb{R}^d\,, \Phi(\theta) = Z\theta.$$

In that case, expanding the scalar product allows to boil this problem down to a quadratic problem. Namely,

$$\begin{aligned}\frac{1}{2}\|\Phi(\theta) - Y\|^2 &= \frac{1}{2}\langle Z\theta - Y, Z\theta - Y\rangle \\ &= \frac{1}{2}\langle (Z^TZ)\theta, \theta\rangle - \langle Z^TY, \theta\rangle + \frac{1}{2}\|Y\|^2 \\ &= \frac{1}{2}\langle A\theta, \theta\rangle - \langle b, \theta\rangle + \frac{1}{2}\|Y\|^2.\end{aligned}$$

However, one should be careful: although A is symmetric and non-negative, there is no guarantee that A is positive definite. There are two ways to establish the existence of a solution to (1.8) in this setting.

(1) Using the geometric interpretation of the problem: in the linear case, (1.8) can be interpreted as follows: find the orthogonal projection of the reference point Y on the hyperplane $\text{Im}(Z)$. As this hyperplane is closed and convex, it follows from the projection theorem that this orthogonal projection is well defined and unique. Be careful though: this means that there exists a unique $X \in \text{Im}(A)$ such that the orthogonal projection of Y is X, but there is not a unique θ^* such that $X = A\theta^*$.

(2) Using optimality conditions: letting $A := Z^TZ\,, b := Z^TY$, the functional

$$\theta \mapsto \frac{1}{2}\langle A\theta, \theta\rangle - \langle b, \theta\rangle$$

is convex. Thus any critical point is a minimiser. As A is not necessarily invertible, it is not clear whether the criticality equation

$$A\theta^* = b$$

actually has a solution. Nevertheless, finding a critical θ^* amounts to finding a solution to

$$Z^T(Z\theta^* - Y) = 0.$$

Now recall that in finite dimension we always have

$$\mathrm{Im}(Z) = \ker(Z^T)^\perp.$$

Thus

$$\mathrm{Im}(Z) \oplus \ker(Z^T) = \mathbb{R}^N$$

and, consequently, there exists $X \in \mathrm{Im}(Z)$ (and, consequently, $\theta^* \in \mathbb{R}^d$), $X_{\mathrm{ker}} \in \ker(Z^T)$ such that

$$Y = X + X_{\mathrm{ker}} = Z\theta^* + X_{\mathrm{ker}}.$$

The conclusion follows. Of course, we observe that the uniqueness of θ^* fails again, unless the matrix $Z^T Z$ is invertible.

When trying to solve ordinary least squares, a possibility is to use the so-called Moore-Penrose pseudo-inverse of a matrix; this is an important object, which is the topic of Exercise 1.15.

1.4.3. **Non-linear least squares**

We now move on to a brief discussion of the non-linear least square model. Recall that we work with a generic functional

$$f(\theta) = \frac{1}{2}\|\Phi(\theta) - Y\|^2.$$

We are interested in simple conditions on Φ that can guarantee a good behaviour of the gradient descent. In order to do so, we wish to find conditions on Φ to ensure that the Polyak-Lojasiewicz condition is satisfied. Assume that Φ is $\mathscr{C}^1$. Then

$$\begin{aligned}\|\nabla f(x)\|^2 &= \|(\nabla\Phi)^T(\theta)(\Phi(\theta) - Y)\|^2\\ &= \langle(\nabla\Phi(\theta)\nabla\Phi(\theta)^T)(\Phi(\theta) - Y), \Phi(\theta) - Y\rangle.\end{aligned}$$

Consequently, if the matrix $\nabla\Phi(\nabla\Phi)^T$ is uniformly positive, the Polyak-Lojasiewicz condition is satisfied.

1.5. **Line search and variable step size**

The aim of this section is to prepare for future methods, typically (quasi-)Newton methods, where one cannot expect to use a fixed step-size. Rather than delve into the general theory (we refer to the next chapter) we focus on the simple case of *line-search*, and we state a convergence result for quadratic functions; this result is proved in Exercise 1.16. The setting is the following: consider a matrix $A \in S_d^{++}(\mathbb{R})$ and $b \in \mathbb{R}^d$. Let f be the quadratic function represented by (A, b). For a given initialisation $x_0 \in \mathbb{R}^d$, we define, iteratively, a sequence $\{x_k\}_{k\in\mathbb{N}}$ by

$$\forall k \in \mathbb{N}\,, x_{k+1} = x_k - \tau_k \nabla f(x_k)$$

where

$$\tau_k = \underset{\tau>0}{\mathrm{argmin}}\, f(x_k - \tau\nabla f(x_k)).$$

The main result is the following:

Theorem 1.7. *The sequence generated by the line-search algorithm converges linearly at rate*

$$\frac{\operatorname{cond}(A) - 1}{\operatorname{cond}(A) + 1}.$$

In particular, observe that the guaranteed rate of convergence corresponds to the optimal rate of convergence for the fixed step-size gradient descent.

1.6. **Exercises of the chapter**

1.6.1. **Basics**

Exercise 1.1. [Fixed points] We consider a function $F : \mathbb{R}^d \to \mathbb{R}^d$ of class $\mathscr{C}^1$ and we let $x^* \in \mathbb{R}^d$ be such that $F(x^*) = x^*$. Assume that there exists $\delta > 0$ such that for all $y \in \mathbb{R}^d$ we have $\|\nabla F(x^*)y\| > (1+\delta)\|y\|$. Show that for any $\varepsilon > 0$, and any $x_0 \in \mathbb{B}(x^*;\varepsilon) \setminus \{x^*\}$, the iterative sequence $x_{k+1} = F(x_k)$ does not converge to x^*, or converges to x^* in a finite number of iterations. Show that if $F \in \mathscr{C}^2(\mathbb{R}^d;\mathbb{R}^d)$ and if $F(x^*) = x^*\,, \nabla F(x^*) = 0$ then, for $\varepsilon > 0$ small enough and any $x_0 \in \mathbb{B}(x^*;\varepsilon)$ the sequence $\{x_k\}_{k\in\mathbb{N}}$ converges to x^* quadratically: there exist constants $C \in \mathbb{R}\,, \alpha \in (0;1)$ such that

$$\forall k \in \mathbb{N}\,, \|x_k - x^*\| \leqslant C\alpha^{2^k}.$$

Exercise 1.2.

(1) Show that a coercive function on $\mathbb{R}^d$ admits a minimiser.
(2) Give an example of a function that admits a minimiser but that is not coercive.

Exercise 1.3. We let $M \in S_d(\mathbb{R})$ and $\lambda_1(M) \leqslant \cdots \leqslant \lambda_d(M)$ be its eigenvalues. Show that

$$\lambda_1(M) = \inf_{\|z\|=1} \langle Mz, z\rangle\,, \lambda_d(M) = \sup_{\|z\|=1} \langle Mz, z\rangle$$

and that

$$\|M\|_{\text{op}} = \max(|\lambda_1(M)|, |\lambda_d(M)|).$$

Exercise 1.4. Show that, if f is a $\mathscr{C}^2$ function, and if x^* is a critical point of f in the sense that $\nabla f(x^*) = 0$, then

(1) If $\nabla^2 f(x^*) \in S_d^{++}(\mathbb{R})$, then x^* is a local minimiser of f.
(2) If $\nabla^2 f(x^*) \in S_d^{--}(\mathbb{R})$, then x^* is a local maximiser of f.
(3) If $\nabla^2 f(x^*)$ has at least one negative and one positive eigenvalue, x^* is a *saddle* point: there exist two orthogonal directions $\vec{e}_1\,, \vec{e}_2$ such that $t^* = 0$ is a local minimiser of $t \mapsto f(x^* + te_1)$, and a local maximiser of $t \mapsto f(x^* + te_2)$.

Exercise 1.5. Classify the critical points (local minimisers, local maximisers, saddle points, indeterminate critical points) of the following functions:

(1) $f_1 : (x,y) \mapsto (x-y)^2 + (x+y)^3$,
(2) $f_2 : (x,y) \mapsto x^2 - 2y^2 + 3xy$,
(3) $f_3(x,y) \mapsto x^4 + y^3 - 3y - 2$.

1.6.2. **Quadratic functions and gradient descent**

Exercise 1.6.

(1) We let $A = \begin{pmatrix} 1 & 3 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ and f be represented by (A,b). Can the gradient descent initialised at a given $x_0 \in \mathbb{R}^d$ with fixed step size $\tau > 0$ converge?
(2) Assume that A is symmetric and that for any $b \in \mathbb{R}^d$, for any $x_0 \in \mathbb{R}^d$ there exists $\tau > 0$ such that the gradient descent generated at x_0 with step size $\tau > 0$ converges. Show that $A \in S_d^{++}(\mathbb{R})$.

Exercise 1.7. Let $f \in \mathscr{C}^1(\mathbb{R}^d)$ be a coercive function with an L-Lipschitz gradient, and assume that f only has one critical point. Show that for any initialisation

x_0 and any $\tau \in (0; 1/2L)$, the gradient descent initialised at x_0 with step size τ converges.

Exercise 1.8. Let $f \in \mathscr{C}^1(\mathbb{R}^d)$ be such that f is bounded from below, ∇f is μ-Lipschitz and that $\|\nabla f\|$ is coercive. We further assume that f only has isolated critical points. Show that, for any $\tau > 0$ small enough the sequence generated by the gradient descent with fixed step size τ converges. Does it necessarily converge to a local minimiser?

Exercise 1.9. We let $A \in S_d(\mathbb{R})$ be matrix with (at least) two eigenvalues of opposite signs. We let $b = 0$. Show that for any $\tau > 0$ the set $\{x_0 \in \mathbb{R}^d :$ the gradient descent initialised at x_0 with fixed step size τ converges$\}$ has measure zero.

Exercise 1.10. Give a strictly convex function φ such that the equation $\nabla^2\varphi = 0$ has an infinite number solutions.

1.6.3. **Convexity**

Exercise 1.11. (1) Show that if ∇f is μ-Lipschitz then

$$\forall x \in \mathbb{R}^d, f\left(x - \frac{1}{\mu}\nabla f(x)\right) - f(x) \leqslant -\frac{1}{2\mu}\|\nabla f(x)\|^2.$$

(2) Show that if ∇f is μ-Lipschitz and f is minimal at x^*, then

$$\forall x \in \mathbb{R}^d, f(x) - f(x^*) \geqslant \frac{1}{2\mu}\|\nabla f(x)\|^2. \tag{1.9}$$

Exercise 1.12. (1) Show that a function f is μ-strongly convex if, and only if, $f - \mu\|\cdot\|^2$ is convex.

(2) Show that if f is $\mathscr{C}^1$, it is μ-strongly convex if, and only if,

$$\forall x, y \in \mathbb{R}^d, \langle \nabla f(x) - \nabla f(y), x - y\rangle \geqslant \mu\|x - y\|^2.$$

Exercise 1.13 (Polyak-Lojasciewicz Inequality)**.** Let $f : \mathbb{R} \to \mathbb{R}$ be a λ-convex function and let x^* be a minimiser of f. First, prove that

$$\forall x \in \mathbb{R}^d, \|x - x^*\|^2 \leqslant \frac{2}{\lambda}(f(x) - f(x^*)).$$

Second, show the following inequality:

$$\forall x \in \mathbb{R}^d, f(x) - f(x^*) \leqslant \frac{1}{2\lambda}\left\|\nabla f(x)\right\|^2.$$

Finally, deduce that

$$\forall x \in \mathbb{R}^d, \|x - x^*\| \leqslant \frac{1}{\lambda}\|\nabla f(x)\|.$$

1.6.4. **Matrices, conditioning number and line-search algorithm**

Exercise 1.14. [Some basic properties of the conditioning number]

(1) Show that, for any symmetric positive definite matrix M, $\mathrm{cond}(M) \geqslant 1$.
(2) Show that for any symmetric definite positive matrix $\mathrm{cond}(M) = \|M\|_{\mathrm{op}} \cdot \|M^{-1}\|_{\mathrm{op}}$. We use this expression to define the conditioning number of any invertible matrix $M \in Gl_d(\mathbb{R})$.
(3) Show that for any $M \in Gl_d(\mathbb{R})$ $\mathrm{cond}(M) \geqslant 1$ and that, for any orthogonal matrix P, $\mathrm{cond}(PM) = \mathrm{cond}(M)$.
(4) For any $M \in Gl_d(\mathbb{R})$ show that $\|M\|_{\mathrm{op}} = \|M^T\|_{\mathrm{op}}$.

(5) Let $M \in Gl_d(\mathbb{R})$ be such that $\mathrm{Cond}(M) = 1$. Show that there exists $x \in \mathbb{R}^*$ such that xM is an orthogonal matrix.

Exercise 1.15. [The Moore-Penrose pseudo-inverse] In this exercise, we investigate the properties of the Moore-Penrose pseudo inverse. Namely, for any matrix $A \in M_{m,n}(\mathbb{R})$ we say that $B \in M_{n,m}(\mathbb{R})$ is a *pseudo-inverse* of A is

$$ABA = A. \tag{1.10}$$

Existence of pseudo-inverse matrices In this first part we show that any A admits a pseudo-inverse.

(1) A being fixed, why does there exists an integer r and two invertible matrices $U \in M_m(\mathbb{R})\,, V \in M_n(\mathbb{R})$ such that

$$A = U \begin{pmatrix} \mathrm{Id}_r & 0 \\ 0 & 0 \end{pmatrix} V.$$

(2) Show that $V^{-1} \begin{pmatrix} \mathrm{Id}_r & 0 \\ 0 & 0 \end{pmatrix} U^{-1}$ is a pseudo-inverse of A.

(3) Do we have uniqueness?

The Moore-Penrose pseudo-inverse The Penrose pseudo-inverse can be seen from two points of view. The first one is to require more symmetry and structural assumption on the pseudo-inverse, while the second one, which is more natural, is to perturb the system $Ax = y$ into $\tilde{A}x = y$ with $\tilde{A}$ close to A and invertible, and to pass to the limit in the solutions of this system.

(1) Let M be a $N \times N$ symmetric matrix. Show that

$$Q_M := \lim_{\delta \to 0} (M + \delta \mathrm{Id})^{-1} M = \lim_{\delta \to 0} M (M + \delta \mathrm{Id})^{-1}$$

always exists and is in fact the projection on $\mathrm{Im}(M)$.

(2) We now go back to our general (not necessarily symmetric) matrix A. Show that

$$A^{MP} := \lim_{\delta \to 0} (A^T A + \delta^2 \mathrm{Id})^{-1} A^T$$

is a pseudo-inverse of A.

(3) Recall why $\ker(A) \oplus \mathrm{Im}(A^T) = \mathbb{R}^n\,, \ker(A^T) \oplus \mathrm{Im}(A) = \mathbb{R}^m$. For any $z \in \mathbb{R}^m$, let $z_{\mathrm{Im}} + z_{\ker}$ be the associated decomposition.

(4) Show that there exists $x_0 \in \mathbb{R}^n$ such that

$$\lim_{\delta \to 0} (A^T A + \delta \mathrm{Id})^{-1} A^T z = \mathrm{Proj}_{\mathrm{Im}(A^T A)} x_0.$$

(5) Show that, for any z, $\hat{X} := A^{MP} z$ solves the optimisation problem

$$\min_{X \in \mathbb{R}^m\,, \|AX - z\|^2 = \min_{\theta \in \mathbb{R}^m} \|A\theta - x\|^2} \|X\|^2.$$

In other words, the Moore-Penrose inverse gives the minimal norm solution of the ordinary least-square problem.

Exercise 1.16. The goal of this exercise is to show the convergence of the line-search gradient descent for quadratic functions.

(1) Preliminary: Kantorovich inequality Let $A \in S_d^{++}(\mathbb{R})$ with eigenvalues $0 < \lambda_1 \leqslant \cdots \leqslant \lambda_d$. Show that

$$\forall x \in \mathbb{R}^d \setminus \{0\}\,, \|x\|^4 \leqslant \langle Ax, x \rangle \cdot \langle A^{-1} x, x \rangle \leqslant \frac{\|x\|^4}{4} \cdot \frac{(\lambda_1 + \lambda_d)^2}{\lambda_1 \lambda_d}.$$

(2) Let $A \in S_d^{++}(\mathbb{R})$ and $b \in \mathbb{R}^d$. Let $x \in \mathbb{R}^d$. Solve the optimisation problem[1]

$$\min_{\tau > 0} f(x - \tau \nabla f(x)).$$

(3) We now consider the sequence generated by the line search algorithm. Using the explicit expression of the step size obtained at the previous question and defining, for any $k \in \mathbb{N}$, $y_k := A(x_k - x^*)$, show that

$$\forall k \in \mathbb{N}\,, \langle y_{k+1}, x_{k+1} - x^* \rangle = \langle y_k, x_k - x^* \rangle \cdot \left(1 - \frac{\|y_k\|^4}{\langle A y_k, y_k \rangle \langle A^{-1} y_k, y_k \rangle}\right).$$

(4) Conclude the proof.

1.7. Correction of the exercises

Solution of Exercise 1.1. (1) Consider the iterative sequence $x_{k+1} = F(x_k)$. First of all, from the assumption, there exists a radius $\varepsilon > 0$ such that $\inf_{\mathbb{B}(x^*;\varepsilon)} \inf_{\|y\|=1} \|\nabla F(x) y\| \geqslant 1 + \frac{\delta}{2}$ and, consequently, x^* is the unique fixed point of F in $\mathbb{B}(x^*;\varepsilon)$. Fix this $\varepsilon > 0$. Now if there exists $x_0 \in \mathbb{B}(x^*;\varepsilon) \setminus \{x_0\}$ such that the sequence converges to x_0, it would follow that $\|x_{k+1} - x^*\| \geqslant (1 + \frac{\delta}{2}) \|x_k - x^*\|$, which immediately leads to a contradiction.

(2) It suffices to do a Taylor expansion to obtain that

$$x_1 - x^* = F(x_0) - F(x^*) = \langle \nabla^2 F(\xi)(x_0 - x^*), x_0 - x^* \rangle$$

for some ξ. A simple iteration argument (which needs to be detailed properly) allows to conclude.

Solution of Exercise 1.2. (1) Fix an arbitrary $x_0 \in \mathbb{R}^d$. Then it is trivial to see that

$$\inf_{x \in \mathbb{R}^d} f = \inf_{x \in \mathbb{R}^d\,, f(x) \leqslant f(x_0)} f.$$

The conclusion follows as $\{f \leqslant f(x_0)\}$ is closed by continuity of f and bounded by the coercivity assumption.

(2) Take $f(x) = \sin(x)$.

Solution of Exercise 1.3. It suffices to apply the spectral decomposition theorem.

Solution of Exercise 1.4. Immediate by Taylor expansions and the Rayleigh quotient formulation of eigenvalues (see the previous exercise).

Solution of Exercise 1.5. (1) The critical points are the points (x, y) such that

$$\begin{cases} 2(x-y) + 3(x+y)^2 = 0, \\ -2(x-y) + 3(x+y)^2 = 0. \end{cases}$$

In particular, (x, y) is critical if and only if $x = y = 0$. The hessian at $(0,0)$ is the matrix

$$\begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$$

which has eigenvalues $\{0, 4\}$. In particular, we can not conclude immediately. Nevertheless, observe that $f(-\varepsilon, -\varepsilon) < 0$ for ε small enough. Consequently, $(0,0)$ is a saddle point.

[1] In particular, show existence and uniqueness of the optimiser

(2) Likewise, (x, y) is critical if, and only if,

$$\begin{cases} 2x + 3y = 0 \\ -4y + 3x = 0. \end{cases}$$

This system has $(0, 0)$ as a unique solution. Furthermore, the hessian at $(0, 0)$ is

$$\begin{pmatrix} 2 & 3 \\ 3 & -4 \end{pmatrix}$$

which has two eigenvalues with opposite signs. Thus $(0, 0)$ is a saddle point.

(3) (x, y) is a critical point if, and only if,

$$\begin{cases} 4x^3 = 0\,, \\ y^2 - 1 = 0. \end{cases}$$

Similarly $(0, 1)$ and $(0, -1)$ are the only critical point. As the hessian at $(0, y)$ is $\begin{pmatrix} 0 & 0 \\ 0 & 6y \end{pmatrix}$ and $f(x, y) = -f(x, -y)$ we deduce that $(0, 1)$ is a local minimiser and $(0, -1)$ is a local minimiser.

Solution of Exercise 1.6. (1) Let $M := \frac{A+A^T}{2}$. If the gradient with fixed step size $\tau > 0$ converges, it converges to a solution x^* of $Mx^* = b$. However, $b \in \text{Im}(M)^\perp \setminus \{0\}$, so that no such x^* exists. Thus, the gradient descent can not converge.

(2) Without loss of generality, we can assume that A is diagonal, $A = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix}$.

If we assume that (for instance) $\lambda_1 < 0$, then, letting $x_0 = e_1$ be the associated first eigenvector and $b = 0$, we deduce that for any $k \in \mathbb{N}$ and any $\tau > 0$ there holds $x_k = (1 - \tau\lambda_1)^k x_0$, and it thus does not converge. Thus, for any i, $\lambda_i \geqslant 0$. If we assume, for instance, that $\lambda_1 = 0$, then taking $b = e_1$ and any x_0, the same reasoning as in the previous question shows that the gradient descent can not converge. Thus, for any i, $\lambda_i > 0$ and so $A \in S_d^{++}(\mathbb{R})$.

Solution of Exercise 1.7. It suffices to apply the Zoutendijk theorem; the fact that $\tau < 1/2L$, combined with the coercivity of f, implies the boundedness of the sequence of iterates. In particular, it suffices to show that the sequence only has one closure point. As any closure point is a critical point, the assumption allows to conclude.

Solution of Exercise 1.8. By the Zoutendijk theorem, for $\tau > 0$ small enough, $\|\nabla f(x_k)\| \underset{k\to\infty}{\to} 0$. By assumption, this implies that $\{x_k\}_{k\in\mathbb{N}}$ converges, up to a subsequence, to a critical point. To conclude, it suffices to show that the sequence has a unique closure point. First of all, observe that as the critical points of f are isolated and as $\|\nabla f\|$ is coercive, f only has finitely many critical points $x_1, \ldots, x_N$. For the sake of readability, assume that f only has two distinct critical points $x_1\,, x_2$ (the general case follows by an immediate adaptation).

Argue by contradiction and assume that $\{x_k\}_{k\in\mathbb{N}}$ has two distinct closure points, which are necessarily equal to $x_1\,, x_2$. We can fix $\varepsilon > 0$ such that $\mathbb{B}(x_1; \varepsilon) \cap \mathbb{B}(x_2; \varepsilon) =$

$\emptyset$ and such that x_i is the unique critical point of f in $\mathbb{B}(x_i;\varepsilon)$ $(i=1,2)$. As $x_{k+1}-x_k \underset{k\to\infty}{\to} 0$, it is easy to see that the set $\{k\in\mathbb{N}\,, x_k\notin(\mathbb{B}(x_1;\varepsilon)\cup\mathbb{B}(x_2;\varepsilon))\}$ is infinite. We can thus extract another converging subsequence, and obtain the existence of another closure point x_3, which is necessarily critical, a contradiction.

Using a suitable modification of $x\mapsto x^3$ with a positive initial condition x_0 and a small enough step size, we deduce that it does not necessarily converge to a local minimiser.

Solution of Exercise 1.9. Up to a change of basis, we can assume that $A=\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_d\end{pmatrix}$ with $\lambda_1<0$. Letting $x_{k,1}$ be the first coordinate of the k-th iterate we observe that $x_{k,1}=(1-\tau\lambda_1)^k x_{0,1}$. Thus, the set of initial conditions that guarantee convergence is included in $e_1^\perp$, which has Lebesgue measure zero, thereby concluding the proof.

Solution of Exercise 1.10. It suffices to take $\varphi(x)=\frac{x^2}{2}+\cos(x)$. Indeed, $\varphi''(x)=1-\cos(x)$, which has infinitely many zeroes. As these zero has zero Lebesgue measure the function φ' is strictly increasing, thereby giving the strict convexity of the function.

Solution of Exercise 1.11. (1) This basically follows the proof of the theorem on the gradient descent. Namely, we know (by the integral Taylor formula for instance) that

$$f\left(x-\frac{1}{\mu}\nabla f(x)\right)-f(x)\leqslant-\frac{1}{\mu}\|\nabla f(x)\|^2+\frac{1}{2\mu}\|\nabla f(x)\|^2.$$

The conclusion follows.

(2) Now, observe that for any $x\in\mathbb{R}^d$ we have

$$f(x^*)\leqslant f\left(x-\frac{1}{\mu}\nabla f(x)\right)\leqslant f(x)-\frac{1}{2\mu}\|\nabla f(x)\|^2.$$

The conclusion follows.

Solution of Exercise 1.12. (1) It suffices to observe that

$$\begin{aligned}&t(1-t)\|x-y\|^2\\&\qquad=-(1-t)^2\|x\|^2-t^2\|y\|^2-2t(1-t)\langle x,y\rangle+(1-t)\|x\|^2+t\|y\|^2\end{aligned}$$

and to plug this equality in the strong convexity inequality.

(2) We proceed with a Taylor expansion. Starting from the definition of strong convexity, we observe that for any $x\,,y\in\mathbb{R}^d$,

$$\begin{aligned}f(x)+t\langle\nabla f(x),y-x\rangle+o(t)&=f((1-t)x+ty)\\&\leqslant f(x)+t(f(y)-f(x))-\mu t\|x-y\|^2+o(t)\end{aligned}$$

whence

$$\langle\nabla f(x),y-x\rangle\leqslant f(y)-f(x)-\mu\|x-y\|^2.$$

By convexity of f,

$$f(y)-f(x)\leqslant\langle\nabla f(y),y-x\rangle.$$

Thus

$$\langle \nabla f(x) - \nabla f(y), y - x \rangle \leqslant -\mu \|x - y\|^2,$$

whence the conclusion.

Solution of Exercise 1.13. As f is λ-convex we know that for any $x, y \in \mathbb{R}^d$ we have

$$f(y) \geqslant f(x) + \langle \nabla f(x), y - x \rangle + \frac{\lambda}{2}\|y - x\|^2.$$

(1) We begin with the first inequality. Take $x = x^*$ in the previous inequality.

(2) Letting x be fixed, minimise the right-hand side of the inequality in y.We obtain a minimum at

$$y^* = -\frac{\nabla f(x)}{\lambda} + x$$

thereby yielding

$$f(y) \geqslant f(x) - \frac{1}{2\lambda}\|\nabla f(x)\|^2.$$

This yields the conclusion.

(3) The two previous inequalities combine to give the required result.

Solution of Exercise 1.14. (1) Trivial.

(2) It suffices to observe that, if M is positive definite, than $\lambda_d(M) = \|M\|_{\text{op}}$. Indeed, working in a diagonalisation basis,

$$\forall x \in \mathbb{R}^d, \|Mx\|^2 = \sum_{i=1}^d \lambda_i^2 x_i$$

and the conclusion follows.

(3) As $\|AB\|_{\text{op}} = 1$ we obtain $\text{cond}(M) \geqslant \|MM^{-1}\|_{\text{op}} = 1$. Likewise, as the operator norm is invariant by conjugation by an orthogonal matrix the second property follows.

(4)

$$\begin{aligned}\|M\|_{\text{op}}^2 = \sup_{\|x\|^2=1} \langle x, M^T M x \rangle \\ = \lambda_d(M^T M)\end{aligned}$$

by the Rayleigh quotient formulation of eigenvalues. Furthermore, $\det(AB - \text{Id}) = \det(BA - \text{Id})$ so that AB and BA have the same spectral radius. The conclusion follows.

(5) The conclusion is obvious if M is symmetric positive definite. In general let $A = MM^T$. By the previous questions we obtain $\text{cond}(A) = 1$. As A is symmetric, positive definite, $A = x^{-2}\text{Id}$ for some $x \in \mathbb{R}^*$, whence xM is orthogonal.

Solution of Exercise 1.15. (1) Rank theorem.

(2) Trivial.

(3) Any matrix of the form $V^{-1}\begin{pmatrix}\text{Id}_r & W \\ S & Q\end{pmatrix}U^{-1}$ is a pseudo-inverse of A.

(1) The fact that $M(M+\delta\mathrm{Id})^{-1}=(M+\delta\mathrm{Id})^{-1}M$ is a consequence of $(M+\delta\mathrm{Id})M=M(M+\delta\mathrm{Id})$. Now observe that for $\delta>0$ small enough we have $(M+\delta\mathrm{Id})$ invertible. This is a simple consequence of the spectral theorem: letting $\lambda_1,\dots,\lambda_n$ be the non-zero eigenvalues of M we have, in a diagonalisation basis of M,

$$(M+\delta\mathrm{Id})=\begin{pmatrix}(\lambda_i-\delta)_{1\leqslant i\leqslant n} & 0\\ 0 & -\delta\mathrm{Id}\end{pmatrix}.$$

Similarly, we obtain

$$M(M+\delta\mathrm{Id})^{-1}=\begin{pmatrix}\left(\frac{\lambda_i}{\lambda_i-\delta}\right)_{1\leqslant i\leqslant n} & 0\\ 0 & 0\end{pmatrix}.$$

Passing to the limit provides the conclusion.

(2) Rank theorem

(3) Consequence of the first part.

(4) Write $z_{\mathrm{Im}}=Ax_0$ for some $x_0\in\mathbb{R}^n$. Then

$$(A^TA+\delta\mathrm{Id})^{-1}A^Tz=(A^TA+\delta\mathrm{Id})^{-1}A^TAx_0.$$

It then suffices to invoke the first part.

(5) Observe that since we are working with projection, we have $A^TA\hat{X}=\hat{X}$. Let x_0 be another solution of

$$Ax_0=A\hat{X}.$$

Then

$$\|x_0\|^2=\|\hat{X}\|^2+\|x_0-\hat{X}\|^2+2\langle\hat{X},\hat{X}-x_0\rangle.$$

However,

$$\langle\hat{X},\hat{X}-x_0\rangle=\langle A^TA\hat{X},\hat{X}-x_0\rangle=0.$$

Solution of Exercise 1.16. (1) Assume $\|x\|=1$ and that $A=\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_d\end{pmatrix}$.

We have to show that for any $\{x_i\}^2_{i=1,\dots,d}$ such that $\sum x_k^2=1$ there holds

$$1\leqslant\sum_{i=1}^d\lambda_ix_i^2\sum_{j=1}^d\frac{x_i^2}{\lambda_i}\leqslant\frac{1}{4}\frac{(\lambda_1+\lambda_d)^2}{\lambda_1\lambda_d}.$$

The left-hand inequality is the Cauchy-Schwarz inequality. As for the right-hand inequality, observe that for any $\delta>0$

$$2\sqrt{\sum_{i=1}^d\lambda_ix_i^2\sum_{j=1}^d\frac{x_i^2}{\lambda_i}}\leqslant\sum_{i=1}^dx_i^2\left(\frac{1}{\delta\lambda_i}+\delta\lambda_i\right)=\sum_{i=1}^dx_i^2\phi(\delta\lambda_i)$$

with $\phi(x)=x+\frac{1}{x}$. As ϕ is convex, it is maximal at λ_1 or λ_d whence

$$2\sqrt{\sum_{i=1}^d\lambda_ix_i^2\sum_{j=1}^d\frac{x_i^2}{\lambda_i}}\leqslant\max\{\phi(\delta\lambda_1),\phi(\delta\lambda_d)\}.$$

Choosing $\delta>0$ such that $\phi(\delta\lambda_1)=\phi(\delta\lambda_d)$ provides the answer.

(2) We obtain after explicit computations (the existence and uniqueness of $\tau^* > 0$ being guaranteed by the strict convexity of the underlying functional) that

$$\tau^* = \frac{\|Ax - b\|^2}{\langle A(Ax - b), Ax - b\rangle}.$$

(3) Explicit computations.

(4) A simple iterative argument allows to conclude.

1.8. Computer session

The goal of this first computer session is to review some basic manipulations in Python and to get down to business regarding basic optimisation algorithms. The basic packages we will be using throughout are the `numpy` and `matplotlib` packages; do not forget to load them.

```python
import numpy
import matplotlib.pyplot as plt
```

1.8.1. Basics in Python

The `numpy` library deals with arrays and matrices, and is particularly useful for linear algebra.

Exercise 1.17. Consider the matrix A defined as:

$$\begin{pmatrix} 4 & 6 & -2 & 3 \\ 2 & -1 & 0 & 1 \\ -7 & 0 & 1 & 12 \end{pmatrix}.$$

(1) Define A as a `numpy.array()`.
(2) Print the first line and the second column of A.
(3) Create a new matrix `Ac` as a copy of A. Modify it by multiplying the first two lines by 2 and (then) divide its last column by 3.
(4) Define the new matrix B

$$\begin{pmatrix} 4 & 5 & 6 \\ 5 & 10 & 15 \\ 1 & 1 & 1 \end{pmatrix},$$

(5) Go back to the initial matrix A. Create a matrix $C \in M_{33}\mathbb{R}$ as a sub-matrix of A: for any $1 \leqslant i, j \leqslant 3$, $c_{ij} = a_{ij}$.
(6) Matrix product
- Create $D = BA$ (using `numpy.dot()`).
- Create $E = B \cdot C$, where $\cdot$ denotes the Hadamard product of matrices:

$$\forall 1 \leqslant i, j \leqslant 3, \quad e_{ij} = c_{ij} b_{ij}.$$

(7) Compute the sum of all elements of E, and create the vector $Y \in \mathbb{R}^3$ whose coordinates are given, for $1 \leqslant i \leqslant 3$, by $y_i = \sum_{j=1}^{4} d_{ij}$.

1.8.2. Plotting curves

The basic command for plotting curves is `plot(x,y)`. In this expression, `x` et `y` are (`array`) with the same size which can be either declared or generated. You should use `linspace(a,b,N)` to represent as a list the interval (a, b) with N (uniform) discretisation points.

The functions `title, axis, legend, x/ylabel` are useful for the presentation of the graphs.The typical code will write as follows:

```python

def f(x) : return .... # The function

xx=linspace(a,b,N) #
```

```
plot(xx, f(xx),'color') # 'color'  is optional but allows to choose the color of the curve

axis('equal') # the two axis are at the same scale
title("Graph")
legend("f")
xlabel("\$ x\$-axis")
ylabel("\$ y\$-axis")
```

Exercise 1.18. (1) Plot the graph of $f : x \mapsto x^2 \cos(10 * x)$ on $(-\pi;\pi)$ in green.

(2) Plot, on the same graph, the functions $f_n : x \mapsto x^2 \cos(nx)$ pour $x \in (-\pi,\pi)$ for $n = 0, 1, \dots, 10$. Observe that if we create a graph using `plt.plot(xx,f(xx),...)`, we can call back the function `plt.plot(xx,g(xx),....)` to draw another graph on the same figure.

Finally, the `contour` function is extremely useful to plot level-sets of functions of several variables. To do so, we consider a function f of two variables and we generate an array containing all the values of f as follows: once f and the two domains of definition are give, we can write:

```
z=[[f(x,y) for x in ... ] for y in ...]
```

The function fonction `plt.contour(`x `domain,`y `domain,Z,N)` plots N level sets. The `plt.colorbar()` command allows for a tuning of the colour scheme.

Exercise 1.19. Plot 20 level sets of

$$f : (x,y) \mapsto e^{-x^2} \sin(\pi x - y)$$

fo $(x,y) \in (-4,4)^2$. Toy around with the options `cmap='inferno'`,`cmap='plasma'`...

1.8.3. Basics of gradient descent

Exercise 1.20. (1) We first consider the minimisation problem

$$\inf_{x\in\mathbb{R}^d} \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

We work in dimension 2. Write a function f that takes as arguments A, b and x and returns $\frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$.

(2) Choose any symmetric, positive definite matrix A and generate a random vector b. Solve the equation

$$Ax = b$$

using the gradient descent method (you are of course allowed to use, here, the explicit expression for the gradient). Can you illustrate the order of convergence of your method?

(3) Plot the level sets of the function f, and represent the successive iterates on the graph.

We now consider a function f of d variables.

(1) We aim at approximation, numerically, the gradient of f. In order to do so, we resort to a centred difference methods. in other words, we write

$$\frac{\partial f}{\partial e_i} \approx \frac{f(x+\delta e_i) - f(x-\delta)}{2\delta}$$

for $\delta > 0$ small enough where $\{e_i\}_{i=1,\dots,d}$ is the canonical basis of $\mathbb{R}^d$. First, define the vectors e_i in Python. Second, write a function `gradientF(F,d,delta=1e-5)` that returns the discrete gradient of F. You can test your code on the following toy function:

```python
# Test function
def F(x):
    return x[0]**2 + x[1]**2
```

(2) Write a code that takes, as arguments, x_0 and a step size τ and returns the sequence generated by the gradient descent initialised at x_0 with fixed step size τ. Test your code on `F`.

We now apply this to the least square problem to test our algorithm: fix $\alpha_0 = 3\,, \alpha_1 = 2$, an integer N and generate a random vector of size N by calling `Xi= numpy.random.rand(N)*2-1` (this generates numbers between -1 and 1). Generate a "noise" vector `Wi=numpy.random.rand(N)*2-1`. Finally, set `Yi=alpha0+alpha1*Xi+Wi` (the set of noisy observations we have access to). We want to see whether we can recover the values of α_0 and α_1 from the knowledge of `(Xi,Yi)`. To do so, we define, for an array $A = (a_0, a_1)$,

$$E(A) = \frac{1}{N}\sum_{k=1}^{N} |y_i - (a_0 + a_1 x_i)|^2.$$

(3) Write the function $A \mapsto E(A)$ in python, A being coded as an `numpy` array. The vectors `Xi` and `Yi` are `numpy` array defined globally.

(4) Plot 20 level lines of E. Is what you observe coherent with we might expect?

(5) Minimise E with a constant step-size gradient descent using the discretised gradient you computed earlier. Run your code with several values of τ and find the best value for τ.

(6) Check that we find a good approximation for (α_0, α_1). Show datas `(Xi,Yi)` and the line $y = a_0 + a_1 x$ that you obtain, for $x \in [-1, 1]$.

2. Newton and quasi-Newton methods

In this chapter, we investigate another very important class of methods, the family of (quasi-)Newton iterates. Here, rather than a descent approach, we focus, for the optimisation problem $\min_{x\in\mathbb{R}^d} f(x)$, on solving the criticality equation

$$\nabla f(x^*) = 0, \tag{2.1}$$

by seeing it as a general nonlinear equation

$$\Phi(x^*) = 0. \tag{2.2}$$

We will present two methods. The first one, the classical Newton method, is extremely efficient but has two difficulties, its computational cost and its delicate sensitivity to initialisation. The second one is much more useful in practice, but slightly slower — keep in mind it also has some limitations.

2.1. **The classical Newton method**

2.1.1. **The general framework**

We consider a function $\Phi : \mathbb{R}^d \to \mathbb{R}^d$, say of class $\mathscr{C}^1$. Assume we are given a reasonable estimate x_0 of a solution x^* to (2.2). In order to compute a better approximation of x^*, we do a Taylor expansion of Φ at order 2 at the point x_0, namely

$$\Phi(x_0+y) \underset{\|y\|\ll 1}{\approx} \Phi(x_0) + \nabla\Phi(x_0)\cdot y.$$

Keep in mind that in this expression $\nabla\Phi$ is the Jacobian matrix $\nabla\Phi = \left(\frac{\partial \Phi_i}{\partial x_j}\right)_{1\leqslant i,j\leqslant d}$. Thus, we replace (2.2) with the approximate equation

$$\text{Find } x_1 \text{ such that } \Phi(x_0) + \nabla\Phi(x_0)\cdot(x_1-x_0) = 0.$$

Provided the matrix $\nabla\Phi(x_0)$ is invertible, this leads to defining

$$x_1 = x_0 - (\nabla\Phi(x_0))^{-1}\Phi(x_0).$$

Two questions need to be addressed: first, is the method well-defined? We see that we need $\nabla\Phi(x_0)$ to be invertible for the previous iterate to make sense. Of course, if we assume that $\nabla\Phi(x^*) \in Gl_d(\mathbb{R})$ and that $\nabla\Phi$ is uniformly continuous, the fact that $Gl_d(\mathbb{R})$ is an open subset of $M_d(\mathbb{R})$ implies that $\nabla\Phi(x_0)$ is invertible as well if x_0 is close enough to x^*. Second, even if the method is well-defined, can we guarantee its convergence and estimate its rate?

The main result of this first part is the following local convergence result:

Theorem 2.1. *Assume that $\Phi \in \mathscr{C}^1(\mathbb{R}^d;\mathbb{R}^d)$ with a M_2-Lipschitz Jacobian. Let $x^* \in \mathbb{R}^d$ be such that*

$$\Phi(x^*) = 0\,, \nabla\Phi(x^*) \in Gl_d(\mathbb{R}).$$

There exists $\varepsilon > 0$ such that, for any $x_0 \in \mathbb{B}(x^*;\varepsilon)$, the iterative sequence

$$\forall k \in \mathbb{N}\,, x_{k+1} = x_k - (\nabla\Phi(x_k))^{-1}\Phi(x_k)$$

is well defined, and converges quadratically to x^*: there exist $C \in \mathbb{R}$ and $\alpha \in (0;1)$ such that

$$\forall k \in \mathbb{N}\,, \|x_k - x^*\| \leqslant C\alpha^{2^k}.$$

Furthermore, we can estimate ε in a finer way: it suffices to have

$$M_2\cdot\|x_0-x^*\|\cdot\|\nabla\Phi^{-1}\|_\infty < 1. \tag{2.3}$$

Some remarks are in order: first, if the method converges, it converges phenomenally fast. Second, the limitation $x_0 \in \mathbb{B}(x^*;\varepsilon)$ is tedious, and not so easy to check. In practice (we refer to the next paragraph), when solving optimisation problems, a possible strategy is to run a slow algorithm for some time, and then to switch to the Newton method at the end of the procedure in order to accelerate the computations. Second, even if at a theoretical level this method is very tempting, a hurdle to overcome is the computation of the inverse of the Jacobian. Unless this Jacobian has a nice structure, this inversion can take some time...although the Newton method itself can be used to compute the inverse of a matrix, see Exercise 2.7!

Proof of Theorem 2.1. We give two proofs.

(1) When Φ is $\mathscr{C}^2$: When the function is more regular, the quadratic convergence of $\{x_k\}_{k\in\mathbb{N}}$ is a consequence of the fixed point theorem, see Exercise 1.1. Indeed, recalling that the differential of $\mathrm{Inv} : Gl_d(\mathbb{R}) \ni A \mapsto A^{-1}$ is given by
$$d\mathrm{Inv}(A) : H \mapsto -A^{-1}HA^{-1}$$
we observe that, setting $F : x \mapsto x - (\nabla\Phi(x))^{-1}\Phi(x)$ we have
$$\nabla F(x^*) = \mathrm{Id} - \mathrm{Id} - (\nabla\Phi(x^*))^{-1}\nabla^2\Phi(x^*)(\nabla\Phi(x^*))^{-1}\Phi(x^*) = 0.$$
A word of caution here: $\nabla^2\Phi$ is not a matrix but rather a matrix of matrices, in the sense that $\nabla^2\Phi \in \mathscr{L}(\mathbb{R}^d;\mathscr{L}(\mathbb{R}^d;\mathbb{R}^d))$ (*i.e.* it is rather a matrix of matrices), but this is not an essential point.

(2) When $\nabla\Phi$ is M_2-Lipschitz: In that case, the idea is essentially the same. First of all, observe that as $Gl_d(\mathbb{R})$ is open, $\nabla\Phi$ is invertible in a small enough ball $\mathbb{B}(x^*;\varepsilon)$. Now we will prove that, up to reducing $\varepsilon > 0$, the sequence is well-defined and converges quadratically. We do not detail the entire induction steps. Assuming the sequence is well-defined at a step k, observe that
$$x_{k+1} - x_k = -(\nabla\Phi(x_k))^{-1}(\Phi(x_k)).$$
As $\Phi(x_k) = \Phi(x_k) - \Phi(x^*) = \int_0^1 \nabla\Phi((1-s)x^* + sx_k)(x_k - x^*)dx$ and as $\nabla\Phi$ is M_2-Lipschitz we derive
$$\begin{aligned}\|x_{k+1} - x^*\| &= \left\|x_k - x^* - \int_0^1 \nabla\Phi(x_k)^{-1}\nabla\Phi((1-s)x^* + sx_k)(x_k - x^*)dx\right\| \\ &\leqslant \left\|x_k - x^* - \int_0^1 (x_k - x^*)ds\right\| \\ &\quad + M_2\|\nabla\Phi(x_k)^{-1}\|_{\mathrm{op}} \cdot \|x_k - x^*\|^2 \cdot \left\|\int_0^1 (1-s)ds\right\| \\ &\leqslant M_2\|\nabla\Phi(x_k)^{-1}\|_{\mathrm{op}} \cdot \|x_k - x^*\|^2.\end{aligned}$$
Now, fixing $\varepsilon' > 0$, $M := \sup_{x\in\mathbb{B}(x^*;\varepsilon')} \|\nabla\Phi(x)^{-1}\|$ and choosing $0 < \varepsilon'' \leqslant \varepsilon$ such that for any $M_2M\varepsilon'' < 1$ we deduce that the sequence is well defined at any step. The quadratic convergence is an immediate consequence, and is guaranteed provided that
$$M_2 \sup_{x\in\mathbb{B}(x^*;\varepsilon')} \|\nabla\Phi(x)^{-1}\|_{\mathrm{op}} \cdot \|x_0 - x^*\| < 1. \tag{2.4}$$

□

2.1.2. Application to optimisation problems — the case of strongly convex functions

Recall that within the framework of optimisation, the function Φ of (2.2) is $\Phi = \nabla f$. In that case, the sequence of Newton iterates reads

For an initialisation $x_0 \in \mathbb{R}^d$, for any $k \in \mathbb{N}$,

$$\text{if } \nabla^2 f(x_k) \in Gl_d(\mathbb{R})\,, x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k).$$

The local convergence of the method to a local minimiser x^* is guaranteed by Theorem 2.1, provided $\nabla^2 f(x^*) \in Gl_d(\mathbb{R})$. Since x^* is assumed to be a local minimiser, this means that we have convergence bounds provided

$$\nabla^2 f(x^*) \in S_d^{++}(\mathbb{R})$$

or, in other words, if x^* is a non-degenerate local minimiser.

Let us now discuss the question of global convergence; in general **the convergence is not global**, even if the function f is strongly convex. Nevertheless, as strong convexity is a very hard assumption to assess in practice, a good strategy is the following:

(1) First, run the gradient descent for a “sufficient” number of iterations. The hope is that it gets you close enough to the minimiser.
(2) Second, run the Newton method, initialising it.

This is problematic: the radius $\varepsilon > 0$ provided by Theorem 2.1 depends on the distance of x_0 to...x^*. This is not good, as the only information available to us is x_0 and the values of f and its derivatives at x_0. To obtain a tractable rule, we would thus like to re-interpret the closeness of x_0 to x^* in terms of ∇f. This is, however, exactly what α-strongly convex functions allow us to do. Recall that if f is α-strongly convex then (see Exercise 1.13) then there holds

$$\|x - x^*\| \leqslant \frac{1}{\alpha} \|\nabla f(x)\|.$$

Consequently, we know from (2.3) that, if x_0 is such that

$$\frac{M_2}{\alpha} \|\nabla f(x_0)\| \cdot \|(\nabla^2 f)^{-1}\|_{\text{op}} < 1 \tag{2.5}$$

the method initialised at x_0 converges quadratically. What is more, if f is strongly α, it implies $\|(\nabla^2 f)^{-1}\|_{\text{op}} < \frac{1}{\alpha}$ so that it suffices to guarantee

$$\|\nabla f(x_0)\| \leqslant \frac{\alpha^2}{M_2} \tag{2.6}$$

to obtain a quadratic convergence.

(Dis)advantages of the Newton method

This still does not answer the question of the global convergence of the Newton method for optimisation. A reasonable guess would be that if f is strongly convex, the Newton method might converge. It is, after all, the right setting for gradient descent.

On the one hand, the Newton method works extremely well for quadratic function: namely, if we try to minimise the function

$$x \mapsto \frac{x^2}{2}$$

then, regardless of the initialisation x_0, the Newton method converges in one iteration; furthermore, the "more quadratic" the function f, the better the convergence rate–we refer to Exercise 2.5.

On the other hand, this very good behaviour for quadratic functions implies that in general the Newton method behaves very poorly: consider the function

$$f : x \mapsto \begin{cases} (x+1)^2 \text{ if } x \geqslant 1\,, \\ (x-1)^2 \text{ if } x \leqslant -1 \\ \text{ Any smooth convex extension on } [-1;1]. \end{cases}$$

Then, starting from $x_0 = 1$ we obtain $x_1 = 1 - 2 = -1\,, x_2 = 1 \dots$ and it is trivial to see that, for any k, $x_k = (-1)^k$. This is simply due to the fact that the Newton method for optimisation converges, for quadratic functions, in exactly one iteration so that, starting at 1, it sends us directly to the unique root of $x \mapsto (x+1)^2$.

Furthermore, it is computationally heavy (and error-ridden) to try and compute the Hessian of f and its inverse.

The remainder of this chapter is devoted to understanding how these two issues might be resolved: first, by considering the so-called *damped* Newton method, which seeks to avoid the overshooting issues that we saw above. Second, quasi-Newton methods, which aim at bypassing the computation of the "true" Hessian and of its inverse. A trigger warning, the rest of this chapter becomes quite technical, quite quickly.

2.1.3. Global convergence analysis: damped Newton method for strongly convex functions

There is a possibility to obtain global convergence of the Newton method for strongly convex and coercive functions, the so-called *damped Newton method.* The basic idea is similar to the standard *line-search* technique in gradient descent. Essentially, we consider, at a given $x \in \mathbb{R}^d$ where $\nabla^2 f(x) \in Gl_d(\mathbb{R})$ the Newton direction

$$d(x) := (\nabla^2 f(x))^{-1} \nabla f(x).$$

To some extent, the norm of $d(x)$ gives (at least when f is α-strongly convex) a notion of proximity to the minimiser x^* of f. The damped Newton method selects, when $\|d(x)\|$ is too large, a smaller step size t_x to avoid overshooting, while, if $\|d(x)\|$ is small enough, we can hope to be in a neighbourhood small enough to guarantee convergence of the method.

To be more specific, we make the following assumptions on f:

$$(H_{Newton}) \qquad \begin{cases} f \text{ is } \alpha-\text{strongly convex and bounded from below,} \\ \nabla f\,, \nabla^2 f \text{ are Lipschitz,} \\ \text{ with Lipschitz constants } M_1\,, M_2 \text{ respectively.} \end{cases}$$

Since we want to make smaller steps and to run the standard Newton method once we are close enough, recall from Theorem 2.1 and (2.6) that we can guarantee quadratic convergence if we start from a point x satisfying

$$\|\nabla f(x_0)\| \leqslant \frac{\alpha^2}{M_2}.$$

Now, as long as this condition is not satisfied, we shall make a smaller step $\bar{t}$ (which we would like to be fixed to not recompute it at every step), and we shall choose a step size that guarantees the fact that $f(x_1) < f(x_0)$. We thus need a finer estimate of $f(x_1) - f(x_0)$.

Finer estimate on $f(x_1) - f(x_0)$ for variable step size We let $\bar{t} > 0$ be a fixed parameter and we set

$$x_1 = x_0 - \bar{t}(\nabla^2 f(x_0))^{-1}\nabla f(x_0).$$

Consequently, from (H_{Newton}) we obtain

$$\begin{aligned} f(x_1) &= f(x_0 - \bar{t}(\nabla^2 f(x_0))^{-1}\nabla f(x_0)) \\ &\leqslant f(x_0) - \bar{t}\langle \nabla f(x_0), \nabla^2 f(x_0)^{-1}\nabla f(x_0)\rangle + \frac{M_1}{2}\|x_1 - x_0\|^2 \\ &\leqslant f(x_0) - \bar{t}\langle \nabla f(x_0), \nabla^2 f(x_0)^{-1}\nabla f(x_0)\rangle + \frac{M_1\bar{t}^2}{2}\|\nabla^2 f(x_0)^{-1}\nabla f(x_0)\|^2. \end{aligned}$$

Now observe that using the spectral theorem and the fact that any eigenvalue λ_k of $(\nabla^2 f(x_0))^{-1}$ satisfies $\lambda_k \leqslant \frac{1}{\alpha}$, we obtain that for any $X \in \mathbb{R}^d$

$$\|\nabla^2 f(x_0)^{-1}X\|^2 \leqslant \frac{1}{\alpha}\langle \nabla^2 f(x_0)^{-1}X, X\rangle$$

whence

$$f(x_1) - f(x_0) \leqslant \bar{t}\left(\frac{M_1\bar{t}}{2\alpha} - 1\right)\langle \nabla f(x_0), \nabla^2 f(x_0)^{-1}\nabla f(x_0)\rangle.$$

Thus, if we take

$$\bar{t} = \frac{\alpha}{M_1}$$

we obtain

$$f(x_1) - f(x_0) \leqslant -\frac{\alpha}{2M_1}\langle \nabla f(x_0), \nabla^2 f(x_0)^{-1}\nabla f(x_0)\rangle \leqslant -\frac{\alpha^2}{2M_1^3}\|\nabla f(x_0)\|^2.$$

In this last step, we used the fact that for any $X \in \mathbb{R}^d$

$$\|X\|^2 = \|(\nabla^2 f(x_0))(\nabla^2 f(x_0)^{-1}X)\|^2 \leqslant \frac{\|\nabla^2 f(x_0)\|_{\text{op}}^2}{\alpha}\langle X, \nabla^2 f(x_0)^{-1}X\rangle,$$

once again by the spectral decomposition theorem.

Summary of the damped Newton method

The damped Newton method is defined as follows: set $\gamma := \frac{\alpha^2}{M_2}$, $\bar{t} := \frac{\alpha}{M_1}$, $x_0 \in \mathbb{R}^d$. For any $k \in \mathbb{N}$, assuming x_k is constructed, define

$$x_{k+1} = \begin{cases} x_k - (\nabla^2 f(x_k))^{-1}\nabla f(x_k) \text{ if } d(x_k) = (\nabla^2 f(x_k))^{-1}\nabla f(x_k) \text{ satisfies} \\ \|\nabla f(x_k)\| < \gamma\,, \\ x_k - \bar{t}(\nabla^2 f(x_k))^{-1}\nabla f(x_k) \text{ else.} \end{cases}$$

Observe that the damped Newton method always enters the quadratic regime phase. Indeed, arguing by contradiction and assuming that for any $k \in \mathbb{N}$, $\|\nabla f(x_k)\| \geqslant \gamma$ we deduce that

$$\forall k \in \mathbb{N}\,, f(x_k) - f(x_0) \leqslant -\frac{\alpha k}{2M_1^2}\gamma^2,$$

in contradiction with the coercivity of f. We have thus obtained the following result:

Theorem 2.2. *Assume f satisfies (H_{Newton}) and has a minimiser at x^*. The damped Newton method converges quadratically for any initialisation $x_0 \in \mathbb{R}^d$.*

2.2. **Quasi-Newton methods**

There is a big drawback to the Newton method applied to the minimisation of a function, among others the fact that it relies on a computation of the second order derivative and on the computation of its inverse. As in practice this is done numerically, errors might pile up and affect the result substantially.

2.2.1. **The one-dimensional secant method**

We very briefly recall the secant method in the one-dimensional case. In 1-d, we can, rather than computing f'' at every iteration, approximate f'' with

$$f''(x_k) \approx \frac{f'(x) - f'(x_k)}{x - x_k}$$

whenever x is close enough to x_k, and this generates the sequence of iterates

$$\begin{cases} x_0 \in \mathbb{R} \text{ fixed,} \\ \forall k \in \mathbb{N}\,, x_{k+1} = x_k - f'(x) \cdot \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})}. \end{cases} \tag{2.7}$$

A tedious series of computations (which we recall in Exercise 2.8) shows that this method does perform *much better* than a brute gradient descent, in the sense that the local convergence rate is super-linear, but that it performs a priori worse than the Newton method. This suggests that such methods might be a good compromise and might mitigate the computational complexity of computing the hessian at every step.

2.2.2. **Generalities about quasi-Newton methods**

There is a whole range of quasi-Newton methods; in this class, we will present the general philosophy behind these methods and present more in details the two most standard quasi-Newton iterations, the DFP and the BFGS algorithms. The basic idea behind quasi-Newton method is to overcome the difficulty of finding $\nabla^2 f$; this refers both to computational mistakes and to computational complexity. Thus, rather than working on the quadratic model

$$f(y) \approx f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2}\langle \nabla^2 f(x)(y-x), y - x\rangle$$

we seek a "good enough" approximation. This means that, at step k, we look for a matrix B_{k+1} that should satisfy:

1. A good approximation quality: the quadratic function
$$y \mapsto f(x_k) + \langle \nabla f(x_k), y - x_k\rangle + \frac{1}{2}\langle B_k(y - x_k), y - x_k\rangle$$
should be a good approximation of the function f.
2. Good structural properties: B_k, as an approximate Hessian, should be symmetric and, if $\nabla^2 f \in S_d^{++}(\mathbb{R})$, should also be definite positive.
3. An efficient cost: B_{k+1} should be easily computable from B_k,

Let us begin with the first point. The least that one might require is that B_k should quantify how much the gradient varies between one iterate and the next one. In particular, it makes sense to require that

$$B_k\,(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1}). \tag{2.8}$$

This is sort of where we need to make a first stop. The two most famous Newton methods, the DFP and the BFGS methods, correspond to to ways of solving this equation for B_k. The DFP works on (2.8) directly, while the BFGS focuses on finding B_k^{-1} in an iterative way. Naturally, as working with B_k^{-1} directly avoids the numerical computation of an inverse, this one will be favoured, but we present both.

2.2.3. **Generalities about the selection of $S_d^{++}(\mathbb{R})$ matrices**

At any rate, in both cases, the general setting is the following: assume we have constructed the first matrices given by the method, say, for the sake of generality, $A_0, \dots, A_k$. In order to look for A_{k+1}, we take into account both a linear equation of the form

$$AX = Y \tag{2.9}$$

and a closeness constraint — this is implemented by requesting that A_{k+1} solves

$$\min_{A \in S_d^{++}(\mathbb{R})\,, AX=Y} \|\sqrt{C}(A - A_k)\sqrt{C}\|_F \tag{2.10}$$

where C is a suitably chosen matrix and F is the Frobenius norm $\|A\|_F = \sum_{i,j} a_{i,j}^2$. Since both DFP and BFGS share similarities in their underlying computations, we lay out the general framework here. We begin with the solvability of (2.9).

Lemma 2.1. *The equation* (2.9) *has a solution $A \in S_d^{++}(\mathbb{R})$ if, and only if,*

$$\langle X, Y\rangle > 0.$$

Proof of Lemma 2.1. If there exists $A \in S_d^{++}(\mathbb{R})$ such that $AX = Y$ then $\langle X, Y\rangle = \langle AX, X\rangle > 0$. Conversely, assume that

$$\langle X, Y\rangle > 0.$$

Let us show that there exists $A \in S_d^{++}(\mathbb{R})$ such that $AX = Y$. If $X = aY$ for some $a > 0$, it suffices to take $A = \frac{1}{a}\mathrm{Id}$. Else, consider $E := \mathrm{span}(X, Y)$. Since $\langle X, Y\rangle > 0$, up to a rotation, we can find a direct orthonormal basis (e_1, e_2) of E such that, in this basis, $x_1\,, x_2\,, y_1\,, y_2 > 0$. We can define $A_0 : (\alpha_1, \alpha_2) \mapsto (y_1\alpha_1/x_1, y_1\alpha_2/x_2) \in S_2^{++}(\mathbb{R})$. We then set $A = \begin{pmatrix} A_0 & 0 \\ 0 & \mathrm{I}_{d-2}\end{pmatrix}$ in an extended orthonormal basis and it is readily checked that it satisfies all requirements. □

We now analyse how to tackle minimisation problems of the type (2.10). Two things are noteworthy:

(1) First, the Frobenius norm is unitarily invariant; we refer to Exercise 2.9.
(2) Second, the matrix C is chosen to guarantee that all matrices $A \in \mathcal{S} = \{A \in S_d^{++}(\mathbb{R})\,, AX = Y\}$ have a common eigenvector and eigenvalue.

To fix the objects under consideration, we pick *any* matrix C such that

$$CY = X$$

and we let $\hat{A} := \sqrt{C}A\sqrt{C}\,, \hat{Y} = \sqrt{C}Y$. Thus, for any $A \in \mathcal{S}$, there holds

$$\hat{A}\hat{Y} = \sqrt{C}ACY = \sqrt{C}AX = \sqrt{C}Y = \hat{Y}.$$

In particular, $\hat{Y}$ is a common eigenvector (with common eigenvalue 1) to all matrices $\hat{A} \in \sqrt{C}\mathcal{S}\sqrt{C}$. Now, let $P = (\hat{Y}/\|\hat{Y}\| | \hat{Y}^\perp) \in O_d(\mathbb{R})$ where $\hat{Y}^\perp$ is an orthogonal complement to $\hat{Y}$, so that

$$P^T\hat{A}P=\begin{pmatrix}1 & 0 & \dots & 0\\ 0 & & & \\ \vdots & & \hat{M} & \\ 0 & & & \end{pmatrix}.$$

Write

$$P^T\hat{A}_kP=\begin{pmatrix}\alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,d}\\ \alpha_{2,1} & & & \\ \vdots & & \hat{M}_k & \\ \alpha_{d,1} & & & \end{pmatrix}.$$

As the Frobenius norm is unitarily invariant, as already noted, the minimisation problem (2.10) boils down to

$$\min_{\hat{M}\in S_d^{++}(\mathbb{R})}\|\hat{M}-\hat{M}_k\|_F$$

and we thus deduce that, at an optimal A^*, there holds

$$\hat{A}^*=P\begin{pmatrix}1 & 0 & \dots & 0\\ 0 & & & \\ \vdots & & \hat{M}_k & \\ 0 & & & \end{pmatrix}P^T \tag{2.11}$$

where $P=(\hat{Y}/\|\hat{Y}\||Y^\perp)$. This is not the most convenient way to write, and it can be condensed further. Indeed, introducing $\hat{u}:=\frac{\hat{Y}}{\|\hat{Y}\|}$ we obtain

$$\begin{aligned}\hat{A}^*&=(u|u^\perp)\begin{pmatrix}1 & 0 & \dots & 0\\ 0 & & & \\ \vdots & & \hat{M}_k & \\ 0 & & & \end{pmatrix}\begin{pmatrix}u^T\\ (u^\perp)^T\end{pmatrix}\\ &=uu^T+u^\perp(u^\perp)^T\hat{A}_ku^\perp(u^\perp)^T\\ &=u\otimes u+(\mathrm{Id}-u\otimes u)\hat{A}_k(\mathrm{Id}-u\otimes u).\end{aligned}$$

Here we used the fact that $\hat{M}_k=(u^\perp)^T\hat{A}_ku^\perp$, and that $\mathrm{Id}-u\otimes u=u^\perp(u^\perp)^T$ is the projection on $u^\perp$ (observe that $Id=PP^T$). In particular, this gives

$$A^*=\sqrt{C^{-1}}u\otimes u\sqrt{C^{-1}}+\sqrt{C^{-1}}(\mathrm{Id}-u\otimes u)\sqrt{C}A_k\sqrt{C}(\mathrm{Id}-u\otimes u)\sqrt{C^{-1}}$$

As

$$\sqrt{C^{-1}}u\otimes u\sqrt{C^{-1}}=\frac{1}{\|\hat{Y}\|^2}(\sqrt{C^{-1}}\hat{Y})(\sqrt{C^{-1}}\hat{Y})^T=\frac{1}{\|\hat{Y}\|^2}Y\otimes Y$$

and

$$\|\hat{Y}\|^2=\langle\sqrt{C^{-1}}Y,\sqrt{C^{-1}}Y\rangle=\langle C^{-1}Y,Y\rangle=\langle X,Y\rangle$$

we deduce that

$$A^*=\frac{Y\otimes Y}{\langle X,Y\rangle}+\left(\mathrm{Id}-\frac{Y\otimes X}{\langle X,Y\rangle}\right)A_k\left(\mathrm{Id}-\frac{X\otimes Y}{\langle X,Y\rangle}\right). \tag{2.12}$$

Before we conclude this section we need a basic considerations about matrices of the form (2.12). Namely, observe that A^* is a rank-1 perturbation of $\left(\mathrm{Id}-\frac{Y\otimes X}{\langle X,Y\rangle}\right)A_k\left(\mathrm{Id}-\frac{X\otimes Y}{\langle X,Y\rangle}\right)$.

Keeping in mind that A^* should be an approximation of the hessian matrix of f (in the case of the DFP method), it is important to compute its inverse. To this end, let us recall the Sherman-Morrison-Woodbury formula.

Proposition 2.1. *[Sherman– Morrison–Woodbury Formula] Let $A \in Gl_d(\mathbb{R})$ and let $u\,,v \in \mathbb{R}^d$. Then $A + u \otimes v \in Gl_d(\mathbb{R})$ if, and only if, $1 + \langle A^{-1}u, v\rangle \neq 0$. In this case*

$$(A + u \otimes v)^{-1} = A^{-1} - \frac{A^{-1}(u \otimes v)A^{-1}}{1 + \langle A^{-1}u, v\rangle}.$$

Proof of Proposition 2.1. We first prove the formula in the case $A = \mathrm{Id}$, namely, we consider $\mathrm{Id} + u_1 \otimes v_1$. Observe that

$$(\mathrm{Id} + u_1 \otimes v_1)u_1 = u_1 + \langle u_1, v_1\rangle u_1$$

Thus if $1 + \langle u_1, v_1\rangle = 0$ $\mathrm{Id} + u_1 \otimes v_1$ is not invertible. Assume that $1 + \langle u_1, v_1\rangle \neq 0$, so that u_1 is an eigenvector with non-zero eigenvalue. Furthermore, for any $w \in v^\perp$, $(\mathrm{Id} + u_1 \otimes v_1)w = w$, so that $\mathrm{Id} + u_1 \otimes v_1 \in Gl_d(\mathbb{R})$.

Now, to compute the inverse, it makes sense to look for $(\mathrm{Id} + u_1 \otimes v_1)^{-1}$ in the form $(\mathrm{Id} + \alpha u_1 \otimes v_1)$. Identifying, this gives

$$0 = (1 + \alpha)u_1 \otimes v_1 + \alpha(u_1 \otimes v_1)^2 = (u_1 \otimes v_1)(1 + \alpha + \alpha\langle u_1, v_1\rangle).$$

Thus we obtain

$$\alpha = \frac{-1}{1 + \langle u_1, v_1\rangle}.$$

Now, in the general case

$$(A + u \otimes v)^{-1} = (\mathrm{Id} + (A^{-1}u \otimes v))^{-1}A^{-1} = \left(A^{-1} - \frac{A^{-1}u \otimes vA^{-1}}{1 + \langle A^{-1}u, v\rangle}\right)$$

□

2.2.4. The DFP algorithm

We come back to the DFP algorithm, named after Davidon, Fletcher & Powell. In this framework, we seek to update the approximation of the Hessian B_k by solving

$$\min_{B \in S_d^{++}(\mathbb{R})\,, B(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k)} \|\sqrt{C}(B - B_k)\sqrt{C}\|_F^2$$

where C is any matrix such that

$$C(\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k.$$

We let $Y_k = \nabla f(x_{k+1}) - \nabla f(x_k)\,, X = x_{k+1} - x_k$. From the discussion of the previous section we deduce that, starting (for the sake of initialisation) at $B_0 = \mathrm{Id}$, we have

$$\forall k \in \mathbb{N}\,, B_{k+1} = \frac{Y_k \otimes X_k}{\langle X_k, Y_k\rangle} + \left(\mathrm{Id} - \frac{Y_k \otimes X_k}{\langle X_k, Y_k\rangle}\right) B_k \left(\mathrm{Id} - \frac{X_k \otimes Y_k}{\langle X_k, Y_k\rangle}\right).$$

Its inverse H_{k+1} is thus given, in compact form (exercise: check for yourselves!), by

$$H_{k+1} = B_{k+1}^{-1} = H_k - \frac{H_k Y_k \otimes Y_k H_k}{\langle Y_k, H_k Y_k\rangle} + \frac{X_k \otimes X_k}{\langle X_k, Y_k\rangle}.$$

2.2.5. **The BFGS algorithm**

In the case of the BFGS (Broyder-Fletcher-Goldfarb-Shanno) algorithm, which proved to be in practice more robust than the DFP update, rather than trying to find an approximation of the hessian, we rather focus on finding a good approximation of its inverse. In other words, starting from $H_0 = \mathrm{Id}$ we solve, iteratively (retaining the notations $X_k = x_{k+1} - x_k\,, Y_{k+1} = \nabla f(x_{k+1}) - \nabla f(x_k)$)

$$\min_{H \in S_d^{++}(\mathbb{R})\,, HY_k = X_k} \|\sqrt{C}(H - H_k)\sqrt{C}\|_F^2$$

where C is any symmetric matrix such that $CX_k = Y_k$. Similarly, we deduce that the iterates are given by

$$H_{k+1} = \frac{X_k \otimes X_k}{\langle X_k, Y_k\rangle} + \left(\mathrm{Id} - \frac{X_k \otimes Y_k}{\langle X_k, Y_k\rangle}\right) H_k \left(\mathrm{Id} - \frac{Y_k \otimes X_k}{\langle X_k, Y_k\rangle}\right). \tag{2.13}$$

In the next section we will show that the BFGS method defined as

$$x_0 \in \mathbb{R}^d\,, H_0 = \mathrm{Id}\,, x_1 = x_0 - \nabla f(x_0) \text{ and for any } k \in \mathbb{N} \begin{cases} H_{k+1} \text{ given by (2.13)} \\ x_{k+2} = x_{k+1} - H_{k+1}\nabla f(x_{k+1}) \end{cases}$$

converges superlinearly for certain classes of functions.

For the sake of completeness, observe that the Sherman-Morrison-Woodbury formula (Proposition 2.1) gives

$$B_{k+1} = H_{k+1}^{-1} = \frac{Y_k \otimes Y_k}{\langle X_k, Y_k\rangle} + B_k - \frac{B_k X_k \otimes X_k B_k}{\langle B_k X_k, X_k\rangle}. \tag{2.14}$$

2.3. **Super-linear convergence of the BFGS algorithm**

The last part of this chapter is devoted to understanding the superlinear convergence of the BFGS method under quite general assumptions. This does, however, require several preparatory steps: as we saw when discussing the damped Newton algorithm, it is necessary, for the standard Newton method, to adjust the step-size as we go along, and there is no reason the BFGS iterations should not be subjected to the same type of problems. This is the first paragraph, devoted to the understanding of the Wolfe criterion for step sizes and to the general understanding of Zoutendijk type results. Second, we will give a general result, the Dennis-Moré theorem, that characterises the super-linear convergence of Newton-like methods. Finally, we will give the complete proof of local superlinear convergence.

2.3.1. **Adaptative step-sizes and the Zoutendijk theorem**

In this first part we work on a general algorithm of the following form: f is a smooth, $\mathscr{C}^2$ function, x_0 is a given initialisation in $\mathbb{R}^d$, and, at every step k, we fix

(1) A direction $z_k \in \mathbb{R}^d$ such that $\langle z_k, \nabla f(x_k)\rangle < 0$ (such a direction is called a *descent direction*).
(2) A step size $\tau_k > 0$,

and we set

$$x_{k+1} = x_k + \tau_k z_k.$$

The question is: under the usual assumption (*e.g.* coercivity, convexity,...), can we guarantee a convergence of this algorithm? The answer is: not completely, but we can nevertheless garner enough information to provide a partial result. A first observation is that the step-sizes should naturally be neither too large (to prevent overshooting) nor too small (for obvious reasons). This leads to the Wolfe conditions:

Definition 2.1. *Let $x \in \mathbb{R}^d$ be such that $\nabla f(x) \neq 0$ and z be a descent direction at x. Let $0 < \gamma_1 < \gamma_2 < 1$ and $\alpha > 0$. We say that $(x, z, \alpha, \gamma_1, \gamma_2)$ satisfies the Wolfe condition if:*

(1) $f(x + \alpha z) \leqslant f(x) + \gamma_1 \alpha \langle z, \nabla f(x)\rangle$,
(2) $\langle \nabla f(x + \alpha z), z\rangle) \geqslant \gamma_2 \langle \nabla f(x), z\rangle$.

The Zoutendijk theorem (which generalises Theorem 1.2) gives a sufficient condition to ensure the convergence of the gradients $\{\nabla f(x_k)\}_{k\in\mathbb{N}}$ (see Remark 2.1).

Theorem 2.3. *Assume that f is $\mathscr{C}^1$, has a μ-Lipschitz gradient and is bounded from below. Let $x_0 \in \mathbb{R}^d$, $0 < \gamma_1 < \gamma_2 < 1$ and consider a sequence $\{\tau_k, z_k\}_{k\in\mathbb{N}}$ which satisfies the following property: the sequence $\{x_k\}_{k\in\mathbb{N}}$ defined iteratively by*

$$\forall k \in \mathbb{N}\,, x_{k+1} = x_k + \tau_k z_k$$

is such that for any $k \in \mathbb{N}$ $(x_k, z_k, \tau_k, \gamma_1, \gamma_2)$ satisfies the Wolfe condition. Then

$$\sum_{k=0}^{\infty} \frac{|\langle \nabla f(x_k), z_k\rangle|^2}{\|z_k\|^2} < \infty. \tag{2.15}$$

Proof of Theorem 2.3. First observe that

$$0 < (\gamma_2 - 1)\langle \nabla f(x_k), z_k\rangle < \langle \nabla f(x_{k+1}) - \nabla f(x_k), z_k\rangle \leqslant \mu \|x_{k+1} - x_k\| \cdot \|z_k\| = \tau_k \mu \|z_k\|^2.$$

Thus,

$$(1-\gamma_2)\, |\langle \nabla f(x_k), z_k\rangle| \leqslant \tau_k \mu \|z_k\|^2. \tag{2.16}$$

Furthermore,

$$f(x_{k+1}) - f(x_k) \leqslant -\gamma_1 \tau_k \, |\langle \nabla f(x_k), z_k\rangle|\,.$$

Thus (2.16) implies

$$f(x_{k+1}) - f(x_k) \leqslant -\frac{\gamma_1(1-\gamma_2)}{\mu} \cdot \frac{|\langle \nabla f(x_k), z_k\rangle|}{\|z_k\|^2}.$$

Summing this inequality in k yields the conclusion. □

Remark 2.1. *[A geometric interpretation of the Zoutendijk theorem] Equation (2.15) has a simple geometric interpretation and very nice consequences. Let θ_k be the angle between the chosen descent direction z_k and $\nabla f(x_k)$, so that*

$$|\langle \nabla f(x_k), z_k\rangle| = |\cos(\theta_k)| \cdot \|\nabla f(x_k)\| \cdot \|z_k\|.$$

(2.15) rewrites

$$\sum_{k=0}^{\infty} \cos(\theta_k)^2 \cdot \|\nabla f(x_k)\|^2 < \infty.$$

In particular, if the sequence $\{\theta_k\}_{k\in\mathbb{N}}$ is bounded away from $\pm\frac{\pi}{2}$, (2.15) ensures that

$$\lim_{k\to\infty} \|\nabla f(x_k)\|^2 = 0$$

and, provided f is strictly convex and coercive, this suffices to guarantee the convergence of the entire sequence.

This theorem will be applied to the BFGS method, where we will choose $z_k = -H_k\nabla f(x_k)$.

The only result that remains to be proved is the existence of step sizes satisfying the Wolfe condition.

Proposition 2.2. *Let $f \in \mathscr{C}^2(\mathbb{R}^d)$, $x \in \mathbb{R}^d$, $0 < \gamma_1 < \gamma_2 < 1$. Assume that $\nabla f(x) \neq 0$ and let z be a descent direction at x. There exists $\alpha > 0$ such that $(x, z, \alpha, \gamma_1, \gamma_2)$ satisfies the Wolfe condition.*

The proof is left as an exercise.

2.3.2. The Dennis-More theorem

We are now giving a general result on the convergence of "Newton-like" methods, which essentially states that the super-linear convergence of a Newton-like algorithm is linked to the way the iteration matrices approximate the Hessian of the function.

Throughout, $f \in \mathscr{C}^2$, x^* is a critical point of f and $\nabla^2 f(x^*) \in S_d^{++}(\mathbb{R})$. We consider a sequence $\{M_k\}_{k\in\mathbb{N}}$ of matrices, an initialisation $x_0 \in D$ where D is an open neighbourhood of x^* and the sequence iteratively defined as

$$\forall k \in \mathbb{N}\,, x_{k+1} = x_k - M_k^{-1}\nabla f(x_k). \tag{2.17}$$

Theorem 2.4. *Assume that the sequence $\{x_k\}_{k\in\mathbb{N}}$ defined by (2.17) remains in D and that, for any $k \in \mathbb{N}$, $x_k \neq x^*$. Then $\{x_k\}_{k\in\mathbb{N}}$ converges super-linearly to x^* in the sense that*

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \underset{k\to\infty}{\to} 0$$

if, and only if,

$$\frac{\left\|(M_k - \nabla^2 f(x^*))(x_{k+1} - x_k)\right\|}{\|x_{k+1} - x_k\|} \underset{k\to\infty}{\to} 0. \tag{2.18}$$

This theorem is due to Dennis & More [5]. It is important to note that (2.18) expresses, not the closeness of M_k to $\nabla^2 f(x^*)$ as linear forms, but only the closeness on the subset spanned by $x_{k+1} - x_k$.

Proof of Theorem 2.4. Assume that $x_k \underset{k\to\infty}{\to} x^*$ super-linearly. Observe that

$$\frac{\|x_{k+1} - x_k\|}{\|x_k - x^*\|} \underset{k\to\infty}{\to} 1. \tag{2.19}$$

Indeed,

$$\left|\frac{\|x_{k+1} - x_k\|}{\|x_k - x^*\|} - 1\right| \leqslant \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|},$$

which allows to conclude.

Furthermore,

$$\frac{\|\nabla f(x_{k+1})\|}{\|x_{k+1} - x_k\|} \underset{k\to\infty}{\to} 0. \tag{2.20}$$

Indeed,

$$\begin{aligned}\frac{\|\nabla f(x_{k+1})\|}{\|x_{k+1} - x_k\|} &= \frac{\|\nabla f(x_{k+1})\|}{\|x_k - x^*\|} \cdot \frac{\|x_k - x^*\|}{\|x_{k+1} - x_k\|} \\ &= \frac{\|\nabla f(x_{k+1}) - \nabla f(x^*)\|}{\|x_k - x^*\|} \cdot \frac{\|x_k - x^*\|}{\|x_{k+1} - x_k\|}\end{aligned}$$

$$\leqslant \mu \frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|}\cdot\frac{\|x_k-x^*\|}{\|x_{k+1}-x_k\|}.$$

However, by super-linear convergence, $\frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} \underset{k\to\infty}{\to} 0$ and (2.20) follows.

Finally,

$$\begin{aligned}(M_k-\nabla^2 f(x^*))(x_{k+1}-x_k) &= -\nabla f(x_k)-\nabla^2 f(x^*)(x_{k+1}-x_k)\\ &= \nabla f(x_{k+1})-\nabla f(x_k)-\nabla^2 f(x^*)(x_{k+1}-x_k)-\nabla f(x_{k+1})\\ &= \nabla^2 f(x^*)(x_{k+1}-x_k)-\nabla^2 f(x^*)(x_{k+1}-x_k)\\ &+ \underset{k\to\infty}{o}(\|x_{k+1}-x_k\|)-\nabla f(x_{k+1}) \text{ by continuity of the Hessian at } x^*\\ &= \underset{k\to\infty}{o}(\|x_{k+1}-x_k\|) \text{ by (2.20).}\end{aligned}$$

This concludes the proof.

Conversely, assume that (2.18) holds. Similarly to the previous computations (here we do not yet use (2.18)), we deduce

$$(M_k-\nabla^2 f(x^*))(x_{k+1}-x^*) = \underset{k\to\infty}{o}(\|x_{k+1}-x_k\|)-\nabla f(x_{k+1}).$$

Consequently, (2.18) implies

$$\frac{\|\nabla f(x_{k+1})\|}{\|x_{k+1}-x_k\|}\underset{k\to\infty}{\to} 0.$$

Since $\nabla^2 f(x^*)\in S_d^{++}(\mathbb{R})$ this gives

$$\frac{\|x_{k+1}-x^*\|}{\|x_{k+1}-x_k\|}\underset{k\to\infty}{\to} 0.$$

As $\|x_{k+1}-x_k\|\leqslant \|x_{k+1}-x^*\|+\|x_k-x^*\|$ this in turn yields

$$\frac{1}{1+\frac{\|x_k-x^*\|}{\|x_{k+1}-x^*\|}}\underset{k\to\infty}{\to} 0.$$

This is equivalent to the super-linear convergence of the sequence $\{x_k\}_{k\in\mathbb{N}}$.

□

2.3.3. Super-linear convergence of the BFGS method and approximation of the Hessian

In this section we assume the following: $f\in\mathscr{C}^2(\mathbb{R}^d)$, is α-strongly convex and has a unique minimiser at x^*. We define a sequence $\{x_k\}_{k\in\mathbb{N}}$ iteratively as follows: fix $0<\gamma_1<\gamma_2<1$. Starting from an initialisation $x_0\in\mathbb{R}^d$ and setting $H_0=\mathrm{Id}$, we let $\tau_0>0$ satisfy the Wolfe condition and we define

$$x_1=x_0-\tau_0 H_0\nabla f(x_0).$$

We then iterate the procedure as follows: $x_0,\dots,x_k$ being constructed, we let H_k be the BFGS update at x_k, we choose $\tau_k>0$ so that τ_k satisfies the Wolfe condition for the descent direction $-H_k\nabla f(x_k)$ and we define

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

The main theorem is the following:

Theorem 2.5. *The sequence of iterates $\{x_k\}_{k\in\mathbb{N}}$ thus constructed converges super-linearly to x^*.*

Proof of Theorem 2.5. The proof will follow several steps:

(1) We first prove that

$$\liminf_{k\to\infty}\|\nabla f(x_k)\| = 0. \tag{2.21}$$

This relies on the Zoutendijk theorem.

(2) We then show that (2.21) is sufficient to guarantee that

$$x_k \underset{k\to\infty}{\to} x^*.$$

(3) We then show that convergence to x^* implies super-linear convergence to x^*.

We begin with (2.21).

Proof of (2.21). We introduce the angle θ_k as

$$\cos(\theta_k) = \frac{\langle \nabla f(x_k), H_k\nabla f(x_k)\rangle}{\|\nabla f(x_k)\|\cdot\|H_k\nabla f(x_k)\|} = \frac{\langle X_k, B_kX_k\rangle}{\|X_k\|\cdot\|B_kX_k\|}.$$

We now prove that there exists a subsequence $\{\theta_{i_k}\}_{k\in\mathbb{N}}$ and $\underline{a}>0$ such that

$$\forall k\in\mathbb{N}\,, |\cos(\theta_{i_k})| \geqslant \underline{a}. \tag{2.22}$$

Theorem 2.3 implies, in that case, that (2.21) holds. We now argue by contradiction, and assume that

$$\cos(\theta_k) \underset{k\to\infty}{\to} 0.$$

Let us show that this would break the non-negativity of H_k. Introduce the function

$$\psi : S_d^+(\mathbb{R}) \ni M \mapsto \begin{cases}\mathrm{Tr}(M) - \ln(\det(M)) \text{ if } M\in S_d^{++}(\mathbb{R})\,,\\ +\infty \text{ else.}\end{cases}$$

Of course,

$$\psi(M) = \sum_{\lambda \text{ eigenvalue of } M} (\lambda - \ln(\lambda)) \geqslant 0.$$

Now, observe that, letting $B_k = H_k^{-1}$ we have from (2.13)

$$\mathrm{Tr}(B_{k+1}) = \mathrm{Tr}(B_k) + \frac{\|Y_k\|^2}{\langle X_k, Y_k\rangle} - \frac{\|B_kX_k\|^2}{\langle B_kX_k, X_k\rangle}. \tag{2.23}$$

In order to compute the determinant, use (2.14), and observe that for any four vectors $x_1\,, x_2\,, y_1\,, y_2$ we have

$$\det(\mathrm{Id} + x_1\otimes y_1 + x_2\otimes y_2) = (1+\langle x_1, y_1\rangle)(1+\langle x_2,y_2\rangle) - \langle x_1,y_2\rangle\langle x_2,y_1\rangle. \tag{2.24}$$

This immediately gives

$$\det(B_{k+1}) = \det(B_k)\cdot\frac{\langle Y_k, X_k\rangle}{\|X_k\|^2}\cdot\frac{\|X_k\|^2}{\langle X_k, B_kX_k\rangle} = \det(B_k)\frac{m_k}{q_k}. \tag{2.25}$$

Thus we deduce that

$$\begin{aligned}\psi(B_{k+1}) &= \psi(B_k) + \frac{\|Y_k\|^2}{\langle X_k, Y_k\rangle} - \frac{\|B_kX_k\|^2}{\langle B_kX_k, X_k\rangle} - \ln(m_k) + \ln(q_k)\\ &= \psi(B_k) + \left(\frac{\|Y_k\|^2}{\langle X_k, Y_k\rangle} - \ln(m_k) - 1\right)\\ &+ \left(1 - \frac{\|B_kX_k\|^2}{\langle B_kX_k, X_k\rangle} + \ln(q_k)\right)\end{aligned}$$

$$= \psi(B_k) + \left(\underbrace{\frac{\|Y_k\|^2}{\langle X_k, Y_k\rangle}}_{=:M_k} - \ln(m_k) - 1\right)$$
$$+ \left(1 - \frac{q_k}{\cos^2(\theta_k)} + \ln(q_k/\cos^2(\theta_k))\right) + \ln(\cos^2(\theta_k)).$$

A key factor is now that by strong convexity of f there holds

$$m_k \geqslant \alpha > 0\,, M_k \leqslant \overline{\mu} = \|f\|_{\mathscr{C}^2}.$$

Furthermore,

$$\forall x \geqslant 0\,, 1 - x + \ln(x) \leqslant 0.$$

Consequently, we deduce the existence of a constant C such that

$$\psi(B_k) \leqslant \psi(B_0) + Ck + \sum_{i=0}^{k} \ln(\cos^2(\theta_k)).$$

We immediately deduce that the sequence $\{\cos^2(\theta_k)\}_{k\in\mathbb{N}}$ cannot converge to zero whence the Zoutendijk theorem entails (2.21). □

Let us now prove that

$$x_k \underset{k\to\infty}{\to} x^*. \tag{2.26}$$

Proof of (2.26). Since the step sizes satisfy the Wolfe condition and since f is coercive, $\{x_k\}_{k\in\mathbb{N}}$ remains bounded, and we thus just have to show that x^* is its unique closure point. From (2.21) and the strong convexity of f we know that x^* is a closure point of the sequence. Let $\{x_{i_k}\}_{k\in\mathbb{N}}$ be a subsequence that converges to x^*. Since for any $k \in \mathbb{N}$ and any $j \geqslant i_k$ $f(x_j) \leqslant f(x_{i_k})$ we deduce that any other closure point x^{**} satisfies $f(x^{**}) \leqslant f(x^*)$, whence the conclusion. □

We finally show the superlinear convergence, that is,

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \underset{k\to\infty}{\to} 0. \tag{2.27}$$

Observe that for any $k \in \mathbb{N}$ we have, since $f \in \mathscr{C}^2$,

$$\begin{aligned}(\nabla f(x_{k+1}) - \nabla f(x_k)) - \nabla^2 f(x^*)(x_{k+1} - x_k) &= (\nabla f(x_{k+1}) - \nabla f(x_k)) - \nabla^2 f(x_k)(x_{k+1} - x_k)\\ &\quad + O(\|x_k - x^*\| \cdot \|x_{k+1} - x_k\|)\\ &= \underset{k\to\infty}{O}((\|x_k - x^*\| + \|x_{k+1} - x_k\|) \cdot \|x_{k+1} - x_k\|)\end{aligned}$$

In particular, from (2.26),

$$\|Y_k - \nabla^2 f(x^*)(x_{k+1} - x_k)\| = \underset{k\to\infty}{o}(\|x_{k+1} - x_k\|).$$

The conclusion follows by the Dennis-Moré theorem. □

2.4. **Exercises of the chapter**

2.4.1. **Newton method**

Exercise 2.1. We want to minimise the function $f : x \mapsto x^4$ using the Newton method. Does the method converge? Linearly? Quadratically?

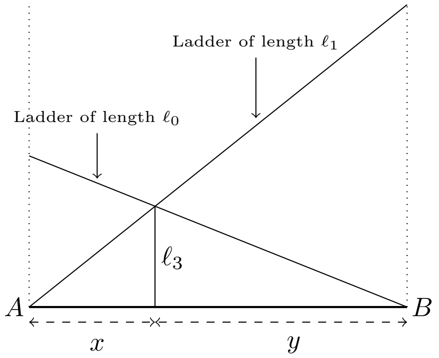
Exercise 2.2. We let $A \in S_d^{++}(\mathbb{R})\,, b \in \mathbb{R}^d$ and we consider $f : x \mapsto \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$. Show that the Newton method applied to the minimisation of f converges for any initialisation.

Exercise 2.3. We want to solve

$$\begin{cases} x^2 + 2xy = 0\,, \\ xy + 1 = 0. \end{cases}$$

(1) Solve the system explicitly.
(2) Write the Newton iterations for the resolution of this system.
(3) Can you expect the Newton iterates to converge locally?

Exercise 2.4. We want to solve a basic geometric problem: consider two walls, separated by a distance $d = AB$, which we want to compute. We are given two ladders, one of length ℓ_0 and a second one of length ℓ_1, with $\ell_0 \leqslant \ell_1$. We assume that these two ladders intersect at a point I at height ℓ_3. We represent the situation in the picture below:



(1) Show that $d = x + y$ where (x, y) solves

$$\begin{cases} \ell_1^2 x^2 = (x+y)^2(\ell_3^2 + x^2)\,, \\ \ell_0^2 y^2 = (x+y)^2(\ell_3^2 + y^2). \end{cases}$$

You should use the Pythagoras and Thales theorems.
(2) Write the system that should be solved to obtain the sequence of Newton iterates.

Exercise 2.5. (1) Let $\bar{x} \in \mathbb{R}$ and consider $f : x \mapsto (x - \bar{x})^2$. Show that for any initialisation x_0 the Newton method applied to the minimisation of f converges in exactly one iteration.
(2) We consider a function φ of class $\mathscr{C}^\infty$ and we assume that there exists $x^* \in \mathbb{R}\,, n \in \mathbb{N}\,, n \geqslant 2$ such that

$$\varphi(x^*) = 0\,, \varphi'(x^*) \neq 0 \text{ and, for any } k \in \{2, \dots, n\},\ \varphi^{(k)}(x^*) = 0.$$

Show that the Newton method locally converges at order $n+1$, in the sense that there exists $\varepsilon > 0\,, \alpha \in (0;1)$ and a constant C such that, for any $x_0 \in \mathbb{B}(x^*, \varepsilon)$,

$$\|x_k - x^*\| \leqslant C\alpha^{(n+1)^k}.$$

(3) Acceleration of the Newton method: Let φ be $\mathscr{C}^\infty$ and x^* be such that $\varphi(x^*) = 0\,, \varphi'(x^*) \neq 0$. We want to find a function f such that the map $\psi : x \mapsto \varphi(x)f(x)$ satisfies $\psi(x^*) = 0\,, \psi'(x^*) \neq 0\,, \psi''(x^*) = 0$. Show that the solution f to the ODE

$$\begin{cases} f(x^*) = 1\,, \\ 2\varphi'(x)f'(x) + \varphi''(x)f(x) = 0 \end{cases}$$

satisfies these requirements. Solve this equation, and deduce a way to accelerate the Newton method.

Exercise 2.6. We investigate in this exercise an important property of optimisation algorithms, the *affine invariance*. The point of affine invariance is to say that no change of coordinates can improve the behaviour, or convergence rate of the algorithm. To be more specific, consider a function f to be minimised, and $A \in Gl_d(\mathbb{R})$. Define $f_A : x \mapsto f(Ax)$. We say that an iterative method is affine invariant if, for any $A \in Gl_d(\mathbb{R})$, the sequence $\{x_k\}_{k\in\mathbb{N}}$ generated for the function f and the sequence $\{y_k\}_{k\in\mathbb{N}}$ generated for f_A with $y_0 = A^{-1}x_0$ satisfy

$$\forall k \in \mathbb{N}\,, x_k = Ay_k.$$

(1) Show that the gradient descent is not affine invariant (use, for instance, $f : x \mapsto \frac{1}{2}\langle Mx, x\rangle$ for a positive definite symmetric matrix M). Give a geometric interpretation of that result.
(2) Show that the Newton method applied to minimisation is affine invariant.

Exercise 2.7. The goal of this exercise is to compute the inverse of a matrix using only the Newton method.

(1) The case of scalars We first compute, for a real number $a \neq 0$, its inverse $\frac{1}{a}$ using only multiplications and additions. Using the function $\varphi : x \mapsto \frac{1}{x} - a$, show that the sequence of Newton iterates converges quadratically, and only requires multiplications and additions.
(2) The case of matrices
  (a) Recall why the set of invertible matrices is an open set of $M_d(\mathbb{R})$.
  (b) Compute the differential of the inversion map $\mathrm{Inv} : Gl_d(\mathbb{R}) \ni M \mapsto M^{-1}$.
  (c) Compute the Newton iterations $\{M_k\}_{k\in\mathbb{N}}$ associated with the Newton method applied to the map $\Phi : M \mapsto M^{-1} - A$.
  (d) Show that

$$\forall k \in \mathbb{N}\,, \mathrm{Id} - AM_{k+1} = (\mathrm{Id} - AM_k)^2.$$

Deduce that the sequence converges if, and only if, the spectral radius of $\mathrm{Id} - AM_0$ is strictly lower than 1.

2.4.2. **Quasi-Newton methods**

Exercise 2.8. In the one dimensional case, the most useful quasi-Newton method is called the *secant method* and it is easier to show its super-linear convergence. Namely, we consider a function $f : \mathbb{R} \to \mathbb{R}$, we let x^* be such that $f(x^*) = 0\,, f'(x^*) \neq 0$. We define, for a given initialisation $x_0\,, x_1 \in \mathbb{R} \setminus \{x^*\}$, the iterative sequence $\{x_k\}_{k\in\mathbb{N}}$ as

$$\forall k \in \mathbb{N}^*\,, x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}.$$

(1) Why is this a quasi-Newton method?

(2) Assume that f is $\mathscr{C}^2$ in a neighbourhood of x^* and introduce the sequence $\{y_k\}_{k\in\mathbb{N}}$ defined as $y_k = x_k - x^*$. Show that there exists a rational fraction $T : (x, y) \mapsto \frac{\alpha y}{\beta+\gamma(x+y)}$ such that

$$\forall k \in \mathbb{N}\,, |y_{k+1}| \leqslant |y_k| \cdot |T(y_k, y_{k-1})|.$$

(3) Deduce that $y_k \underset{k\to\infty}{\to} 0$.

(4) Show, in the case $f'(x^*) \neq 0$, that there exists a constant $C \neq 0$ such that

$$\frac{|y_{k+1}|}{|y_k y_{k-1}|} \underset{k\to\infty}{\to} C.$$

Is this consistent with the superlinear rate we expect?

(5) Introduce $z_k = -\ln(|y_k|)$. Show that the sequence $\{z_{k+1} - z_k - z_{k-1}\}_{k\in\mathbb{N}}$ is bounded from below by a constant $-A$. Finally, show that $\{z_k - A\}_{k\in\mathbb{N}}$ is subadditive. Conclude.

Exercise 2.9. Recall that the Frobenius norm of matrix A is given by

$$\|A\|_F^2 = \sum_{i,j=1}^{d} a_{i,j}^2.$$

(1) Give a geometric interpretation of the Frobenius norm if A is a symmetric matrix.

(2) Show that the Frobenius norm is unitary invariant in the sense that for any matrices $P, Q \in O_d(\mathbb{R})$ there holds

$$\|PAQ\|_F = \|A\|_F.$$

(3) We now show that a unitary invariant norm N is induced by two norms $(\|\cdot\|_1, \|\cdot\|_2)$ if, and only if, there exists a constant α such that $N(A) = \alpha \sup_{\lambda\in\mathbb{C}\,,\lambda \text{ eigenvalue of } A^TA} |\sqrt{\lambda}| = \alpha\rho_{\text{sing}}(A)$.

(a) We begin with a remainder on the Singular Value Decomposition of matrices (SVD for short). Show that if $A \in M_d(\mathbb{R})$ has rank m there exist two orthogonal matrices $U\,, V$ and $D = \text{diag}(\sigma_1, \dots, \sigma_m, 0, \dots, 0)$ with $\sigma_1, \dots, \sigma_m > 0$ such that

$$A = UDV.$$

Hint: work on A^TA.

(b) Show that if a norm N is unitary invariant, there exists $\alpha \in \mathbb{R}$ such that for any rank-1 matrix A we have

$$N(A) = \alpha\rho(A).$$

(c) For any two vectors $x\,,y \in \mathbb{R}^d$, wet let $y \otimes x = yx^T(y_ix_j)_{1\leqslant i,j\leqslant d}$. Show that if N is induced by $(\|\cdot\|_1, \|\cdot\|_2)$ then

$$N(y\otimes x) = \|y\|_2 \sup_{z\,,\|z\|_1=1} |\langle x, z\rangle|.$$

(d) Show that $y \otimes x$ is rank-1.

(e) Assume that a unitary invariant norm is induced by two norms; show that these two norms coincide with the euclidean norm.

(f) Show that the Frobenius norm is not induced by two norms.

2.5. **Corrections of the exercises**

Solution of Exercise 2.1. The Newton iterates read

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} = x_k - \frac{4x^3}{3\times 4x_k^2} = x_k - \frac{x_k}{3}.$$

We deduce that the method converges linearly.

Solution of Exercise 2.2. Recall that $\nabla f(x) = Ax - b\,, \nabla^2 f(x) = A$. Consequently, the Newton iterates read

$$x_{k+1} = x_k - A^{-1}(Ax_k - b) = A^{-1}b.$$

The method converges in exactly one iteration.

Solution of Exercise 2.3. (1) It is obvious that the only solutions are $\{(\pm\sqrt{2}, \mp\sqrt{\frac{1}{2}})\}$.

(2) Introduce the function $F : (x, y) \mapsto (x^2 + 2xy, xy + 1)$. Then the Jacobian matrix of F is

$$J_F(x, y) = \begin{pmatrix} 2x+2y & y \\ 2x & x \end{pmatrix}$$

Recall that the inverse of $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ (when this matrix is invertible) is given by $\frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$ and observe that $\det(J_F) \neq 0$ at any solution of the problem. In particular, the convergence theorem ensures the local quadratic convergence of the Newton method.

Solution of Exercise 2.4. (1) By the Pythagoras and Thales theorem, we obtain

$$\ell_3^2 + x^2 = AI^2 = \ell_1^2 \frac{x^2}{(x+y)^2}.$$

Similarly,

$$\ell_3^2 + y^2 = \ell_0^2 \frac{y^2}{(x+y)^2}.$$

(2) We introduce the function $F : (x, y) \mapsto (\ell_1^2 x^2 - (x+y)^2(\ell_3^2 + x^2), \ell_0^2 y^2 - (x+y)^2(\ell_3^2 + y^2))$.

Solution of Exercise 2.5. (1) The Newton method gives $x_{k+1} = x_k - (x_k - \bar{x}) = \bar{x}$ and thus always converges in exactly one iteration.

(2) This is a simple adaptation of the proof of the lecture notes.

(3) For a given f, we have $\psi'(x) = \varphi'(x)f(x) + \varphi(x)f'(x)$. Thus, if $f(x^*) = 1$, $\psi'(x^*) = \varphi'(x^*) \neq 0$. Furthermore, taking into account the fact that $\varphi(x^*) = 0$ we obtain $\psi''(x^*) = 2\varphi'(x^*)f'(x^*) + \varphi''(x^*)f(x^*)$. The conclusion follows.

(4) This equation is easily integrated: formally, we have

$$f'(x) = -\frac{\varphi''(x)}{2\varphi'(x)} f(x)$$

and we can thus choose

$$f(x) = e^{-\frac{1}{2}\ln(|\varphi'(x)|)}.$$

Typically, if $\varphi' > 0$ we may choose $f(x) = \frac{1}{\sqrt{\varphi'(x)}}$. This gives a function for which the Newton method converges at order 3.

Solution of Exercise 2.6. (1) Consider the function $f : x \mapsto \frac{1}{2}\langle Mx, x\rangle$ and $A \in Gl_d(\mathbb{R})$. Then the gradient descent (with step size $\tau > 0$ for f is the sequence of iterates

$$x_{k+1} = (\mathrm{Id} - \tau M)x_k.$$

On the other hand,

$$f_A(x) = \frac{1}{2}\langle MAx, Ax\rangle = \frac{1}{2}\langle A^T MA, x\rangle.$$

The sequence of iterates generated by gradient descent for f_A now reads

$$y_{k+1} = \left(\mathrm{Id} - \tau A^T MA\right) y_k.$$

In particular, in order for the method to be affine invariant, we would need to have

$$(\mathrm{Id} - \tau A^T MA)A^{-1}x_0 = y_1 = A^{-1}(\mathrm{Id} - \tau M)x_0,$$

and this should be valid for any x_0, whence the condition would read

$$A^{-1}M = A^T M$$

or, in other words, A would need to be orthogonal. Thus, the gradient descent is not affine invariant. Geometrically, this means that the gradient descent strongly depends on the underlying geometry (or metric) of the space, and choosing a right A is a particularly important question–although we do not touch on this topic specifically in this class, we mention that this is the field of “pre-conditioning”.

(2) We consider a $\mathscr{C}^2$ function f that needs to be minimised and $A \in Gl_d(\mathbb{R})$. As a first step, let us compute ∇f_A , $\nabla^2 f_A$; the easiest way is to go through Taylor expansion. Indeed,

$$\begin{aligned} f_A(x+\varepsilon z) &= f(A(x+\varepsilon z)) \\ &= f(Ax + \varepsilon Az) \\ &= f_A(x) + \varepsilon\langle A^T\nabla f(Ax), z\rangle + \frac{\varepsilon^2}{2}\langle A^T\nabla^2 f(Ax)Az, z\rangle + \underset{\varepsilon\to 0}{o}(\varepsilon^2). \end{aligned}$$

We deduce that

$$\nabla f_A(x) = A^T\nabla f(Ax)\,, \nabla^2 f_A(x) = A^T\nabla^2 f(Ax)A.$$

In particular, starting from $y_0 = A^{-1}x_0$ we obtain

$$\begin{aligned} y_1 &= y_0 - \nabla^2 f_A(y_0)^{-1}\nabla f_A(y_0) \\ &= A^{-1}x_0 - A^{-1}\nabla^2 f(x_0)^{-1}A^{-T}A^T\nabla f(x_0) = A^{-1}x_1 \end{aligned}$$

and the conclusion follows. In particular, the Newton method does not depend on the specific underlying geometry.

Solution of Exercise 2.7. (1) The Newton sequence generated by the function φ is the iterative sequence

$$x_{k+1} = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)} = x_k(2 - ax_k).$$

Thus, it only requires multiplications and additions. Furthermore, $\varphi'(1/a) = -a^2 \neq 0$, so that the Newton method converges locally quadratically.

(2) (a) The set $Gl_d(\mathbb{R})$ is the pre-image of $\mathbb{R}^*$ by a polynomial function and is thus open.

(b) The easiest way to compute the differential of Inv is to use Neumann series. Namely, for a given $A \in Gl_d(\mathbb{R})$ and any $H \in M_d(\mathbb{R})$, for ε small enough,

$$\begin{aligned}(A+\varepsilon H)^{-1} &= (\mathrm{Id} + \varepsilon A^{-1}H)^{-1}A^{-1} \\ &= \left(\sum_{k=0}^{\infty}(-1)^k\varepsilon^k(A^{-1}H)^k\right)A^{-1} \\ &= A^{-1} - \varepsilon A^{-1}HA^{-1} + o(\varepsilon).\end{aligned}$$

(c) The Newton iterates are given by

$$M_{k+1} = M_k - (d\Phi(M_k))^{-1}\Phi(M_k).$$

As

$$(d\Phi(M_k))^{-1}(H) = -M_kHM_k$$

this yields

$$\begin{aligned}M_{k+1} &= M_k + M_k(M_k^{-1} - A)M_k \\ &= M_k + M_k - M_kAM_k \\ &= M_k(2I_d - AM_k).\end{aligned}$$

(d) Observe that

$$\begin{aligned}\mathrm{Id} - AM_{k+1} &= \mathrm{Id} - 2AM_k - (AM_k)^2 \\ &= (\mathrm{Id} - AM_k)^2.\end{aligned}$$

Consequently, the sequence converges if, and only if, the spectral radius ρ is strictly lower than 1.

Solution of Exercise 2.8. (1) This is a quasi-Newton method as the quantity $\frac{x_k - x_{k-1}}{f(x_k)-f(x_{k-1}}$ is an approximation of $\frac{1}{f'(x_k)}$.

(2) For the sake of notational simplicity, we only write the proof in the case where f is a quadratic function, that is,

Solution of Exercise 2.9. (1) Diagonal of the image of the unit cube.

(2) $\|A\|_F^2 = Tr(A^TA) = Tr(A^TP^TPAQQ^T) = \|PAQ\|_F^2$.

(3) (a) A^TA is symmetric positive. Consequently, by virtue of the spectral decomposition theorem, we can fin $P \in O_d(\mathbb{R})$ such that $A^TA = P^T\mathrm{diag}(\lambda_1, \ldots, \lambda_m, 0, \ldots, 0)P$ where $\lambda_i \geqslant 0$ and $m = \mathrm{rank}(A)$. Let $V = P$ be the matrix of the eigenvectors of A^TA. Let $\{x_i\}_{i=1,\ldots,d}$ be the eigenvectors of A^TA. Observe that $\{y_i = Ax_i/\sqrt{\lambda_i}\}_{i=1,\ldots,m}$ is an

orthonormal basis of $\mathrm{Im}(A)$. The fact that it is an orthonormal family is immediate, and it suffices to observe that $\mathrm{rank}(A^T A) \leqslant \mathrm{rank}(A^T) = \mathrm{rank}(A)$. A dimensionality argument provides the conclusion. We complete $\{y_i\}_{i=1,\dots,m}$ to be an orthonormal basis of $\mathbb{R}^d$ and we let U be the matrix $(y_1|\dots|y_n)$. The conclusion follows as $Ax_i = \sqrt{\lambda_i}y_i$.

(b) Since A is rank 1, there exist U, V orthogonal and a real number σ such that $A = U\mathrm{diag}(\sigma, 0, \dots, 0)V = \sigma U E_{1,1} V$. The conclusion follows by setting $\alpha = N(E_{1,1})$, and by observing that $\sigma = \rho(A)$. This last identity comes from the Gelfand formula

$$\rho(A) = \lim_{k\to\infty} |||A^k|||^{1/k}.$$

(c) Trivial

(d) Trivial

(e) We know that if the Frobenius norm is induced by a norm then

$$\alpha\rho_{\mathrm{sing}}(y \otimes x) = \|y\|_2 \cdot \sup_{\|z\|_1=1} |\langle x, z\rangle|.$$

Let x be fixed. We deduce that there exists a constant a such that

$$\|y\|_2 = a\rho(y \otimes x).$$

Let us compute $\rho_{\mathrm{sing}}(y \otimes x)$.Observe that

$$\rho_{\mathrm{sing}}(y \otimes x) = \rho(x \otimes y \cdot y \otimes x) = \|x\|_{eucl}\|y\|_{eucl}.$$

We deduce that up to a constant $\|y\|_2$ is the euclidean norm of y and, similarly, that $\sup_{\|z\|_1=1} |\langle x, z\rangle| = \|x\|_{eucl}$. The conclusion follows.

(f) We have thus showed that if the Frobenius norm is induced by two norms, it coincides with the norm of the largest singular value (up to a multiplicative constant). It suffices to take $A = \mathrm{diag}(\lambda_1, \lambda_2, \dots)$ with λ_i reasonably chosen to obtain a contradiction.

2.6. **Computer session**

The goal of this computer class is to get a good feel of the Newton method and its variants. In a (maybe) surprising way, we actually start with the dichotomy method in the one-dimensional case.

2.6.1. **The dichotomy method in the one-dimensional case**

When trying to solve the equation $\phi(x) = 0$ in the one-dimensional case, the most naive method, which actually turns out to be quite efficient, is the *dichotomy method*. Namely, starting from an initial pair $(a_L, a_R) \in \mathbb{R}^2$ with $a_L < a_R$ such that $\phi(a_L)\phi(a_R) < 0$, we set $b := \frac{a_L+a_R}{2}$. If $\phi(b) = 0$, the algorithm stops. If $\phi(a_L)\phi(b) < 0$ we set $a_L \to a_L$ and $a_R \to b$. In this way, we obtain a linearly converging algorithm. In particular, it is globally converging.

Exercise 2.10. Write a function `Dichotomy(phi,aL,aR,eps)` that takes as argument a function `phi`, an initial guess `aL,aR` and a tolerance `eps` and that runs the dichotomy algorithm. Your argument should check that the condition $\phi(a_L)\phi(a_R) < 0$ is satisfied, stop when the function `phi` reaches a value lower than `eps` and return the number of iteration. Run your algorithm on the function $f = \tanh(x)$ with initial guesses $a_L = -20\,, a_R = 3$.

Solving one-dimensional equation with the Newton and the secant method We work again in the one-dimensional case with a function ϕ we want to find the zeros of.

Exercise 2.11. Write a function `Newton(phi,dphi,x0,eps)` that takes, as arguments, a function `phi`, its derivative `dphi`, an initial guess `x0` and a tolerance `eps` and that returns an approximation of the solutions of the equation $\phi(x) = 0$. The tolerance criterion should again be that $|\phi| \leqslant$`eps`. Your algorithm should return an error message in the following cases:

(1) If the derivative is zero (look up the `try` and `except` commands in Python).
(2) If the method diverges.

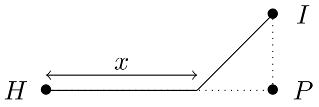
Apply this code to the minimisation of $x \mapsto \ln(e^x + e^{-x})$, with initial condition `x0=1.8`. Compare this with the results of Exercise 2.10.

Exercise 2.12. Write a function `Secant(phi,dphi,x0,x1,eps)` that takes, as arguments, a function `phi`, its derivative `dphi`, two initial positions `x0, x1` and a tolerance `eps` and that returns an approximation of the solutions of the equation $\phi(x) = 0$. The tolerance criterion should again be that $|\phi| \leqslant$`eps`. Apply this code to the minimisation of $x \mapsto \ln(e^x + e^{-x})$, with initial conditions `x0=1,x1=1.9`, then `x0=1,x1=2.3` and `x0=1,x1=2.4`. Compare with the results of Exercise 2.10.

Combining dichotomy and the Newton method A possibility to leverage the advantages of dichotomy (the global convergence of the method) and of the Newton method (the quadratic convergence rate) is to combine both: start from an initial interval `[aL,aR]` of length `InitialLength`with `phi(aL)phi(aR)<0` and fix a real number $s \in [0;1]$. Run the dichotomy algorithm until the new interval is of length `s*InitialLength`. From this point on, apply the Newton method.

Exercise 2.13. Implement this algorithm with $s = 0.1$. Include a possibility to switch back to the dichotomy method if, when switching to the Newton method, the new iterate falls outside of the computed interval `[aL,aR]`. Apply this to the minimisation of the function $f : x \mapsto \ln(e^x + e^{-x})$ with an initial condition that made the Newton method diverge. What you can say about the number of iterations?

Solving an optimisation problem using the Newton method An island (denoted by a point I below) is situated 2 kilometers from the shore (its projection on the shore is a point P). A guest staying at a nearby hotel H wants to go from the hotel to the island and decides that he will run at 8km/hr for a distance x, before swimming at speed 3km/hr to reach the island.



Taking into account the fact that there are 6 kilometers between the hotel and P, how far should the visitor run before swimming?

Exercise 2.14. Model the situation as a minimisation problem, and solve it numerically. Compare the efficiency of the dichotomy method and of the Newton algorithm.

2.7. The Newton method to solve boundary value problems

We consider the following non-linear ODE

$$y'' = f(x, y, y'), \quad x \in [a, b], \quad y(a) = \alpha, y(b) = \beta. \tag{2.28}$$

To classically integrate such an ODE, we usually don't have endpoints for y, but initial values for y and y'. So, we cannot start at $x = a$ and integrate up to $x = b$. This is a boundary value problem.

One approach is to approximate y by somme finite difference and then arrive at a system for the discrete values $y(x_i)$ and finally solve large linear systems.

Here, we will see how we can formulate the problem as a shooting method, and use Newton method so solve it.

The idea is to use a guess for the initial value of y'. Let s be a parameter for a fonction $y(\,\cdot\,; s)$ solution of (1) such that

$$y(a; s) = \alpha, \text{ and } y'(a; s) = s.$$

There is no chance that $y(b; s) = y(b) = \beta$ but we can adjust the value of s, refining the guess until it is (nearly equal to) the right value.

This method is known as shooting method in analogy to shooting a ball at a goal, determining the unknown correct velocity by throwing it too fast/too slow until it hits the goal exactly.

In practice

For the parameter s, we integrate the following ODE:

$$y'' = f(x, y, y'), \quad x \in [a, b], \quad y(a) = \alpha, y'(a) = s. \tag{2.29}$$

We denote $y(\,\cdot\,; s)$ solution of (2).

Let us now define the goal function. Here, we want that $y(b; s) = \beta$, hence, we define:

$$g : s \mapsto y(x; s)|_{x=b} - \beta$$

We seek s^* suth that $g(s^*) = 0$.

Note that computing $g(s)$ involves the integration of an ODE, so each evaluation of g is fairlà expensive. Newton's method seems then to be a good way due to its fast convergence.

To be able to code a Newton's method, we need to compute the derivative of g. For this purpose, let define

$$z(x;s) = \frac{\partial y(x;s)}{\partial s}.$$

Then by differentiating (2) with respect to s, we get

$$z'' = \frac{\partial f}{\partial y} z + \frac{\partial f}{\partial y'} z', \quad z(a;s) = 0, \text{ and } z'(a;s) = 1.$$

The derivative of g can now be expressed in term of z:

$$g'(z) = z(b;s).$$

Putting this together, we can code the Newton's method:

$$s_{n+1} = s_n - \frac{g(s_n)}{g'(s_n)}.$$

To sum up, a shooting method requires an ODE solver and a Newton solver.

Example

Apply this method to

$$y'' = 2y^3 - 6y - 2x^3, \quad y(1) = 2, y(2) = 5/2,$$

with standard library for integration, and your own Newton implementation.

Note that you may want to express this with one order ODE.

With python, you can use 'scipy.integrate.solve_ivp' function:

`https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html#scipy.i`

Here, we are going to use a different approach to solve the boundary value problem:

$$y'' = f(x, y, y'), \quad x \in [a,b], \quad y(a) = \alpha, y(b) = \beta. \tag{2.30}$$

This problem can be solved by the following direct process:

1. We discretize the domain choosing an integer N, grid points $\{x_n\}_{n=0,\dots,N}$ and we define the discrete solution $\{y_n\}_{n=0,\dots,N}$. 1. We discretize the ODE using derivative approximation with finite differences in the interior of the domain. 1. We inject the boundary conditions (here $y_0 = \alpha$ and $y_N = \beta$) in the discretized ODE. 1. Solve the system of equation for the unknows $\{y_n\}_{n=1,\dots,N-1}$.

We use here a uniform grid :

$$h := (b-a)/N, \quad \forall n = 0, \dots, N \quad x_n = hn.$$

If we use a centered difference formula for y'' and y', we obtain:

$$\forall n = 1, \dots, N-1, \quad \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} = f\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}\right).$$

The result is a system of equations for $\mathbf{y} = (y_1, \dots, y_{N-1})$:

$$G(\mathbf{y}) = 0, \quad G : \mathbb{R}^{N-1} \to \mathbb{R}^{N-1}.$$

This system can be solved using Newton's method. Note that the Jacobian $\partial G/\partial \curvearrowright$ is tridiagonal.

Example

Apply this method to

$$y'' = 2y^3 - 6y - 2x^3, \quad y(1) = 2, y(2) = 5/2.$$

Remark 2.2. *In the context of numerical optimal control, these two numerical methods are often called indirect method (for the shooting method) and direct method (for the finite difference method).*

3. Constrained optimisation problems

In this chapter, we start investigating constrained problems and their numerical approximation. We will see several methods:

(1) First and foremost, the case of inequality constraints, in which the basic methods we shall investigate are the penalisation method, the projected gradient descent and the Uzawa algorithm.

(2) Second, we will investigate the case of equality constraints, with a particular emphasis on linear constraints, as these problems are much harder to approximate numerically.

3.1. General framework and basics of constrained optimisation for inequality constraints

3.1.1. Generalities

The setting will *primarily* be the following:

(1) We let

$$K := \{g \leqslant 0\}$$

where g is a coercive, convex function and we assume that there exists $x_0 \in \mathbb{R}^d$ such that $g(x_0) < 0$.

(2) We let $f \in \mathscr{C}^2(\mathbb{R}^d;\mathbb{R})$.

The main problem under consideration is

$$\min_{x\in K} f(x). \tag{3.1}$$

Remark 3.1. *The existence of a solution to (3.1) follows from the compactness of K. Note that the convexity of g does not play a role here but will be instrumental when deriving the optimality conditions, see Exercise 3.5.*

Remark 3.2. *For the sake of readability, we shall mostly work with only one constraint–the generalisation to multiple constraints is usually not a problem. Of course, it is also useful to keep in mind the Lagrange multiplier rule in the case of several (in)equality constraints, and we recall them in section 3.2.1.*

Remark 3.3. *We will often go back and forth between the set K and the function g used to describe it; naturally, g is not unique. Depending on the situation, one description might be preferable to the other.*

Remark 3.4 (Regarding terminology)**.** *One often encounters in the literature the distinction between equality constrained problems and inequality constrained problems, which correspond, respectively, to taking $K = \{g = 0\}$ and $K = \{g \leqslant 0\}$. The distinction is only meaningful when some qualification conditions are satisfied; in order to clarify what we mean, for us, given a constraint set K, we will refer to it as an inequality constrained set when K is the closure of its interior, and to equality constraints when K has an empty interior. This is not a completely standard terminology but it will prove useful.*

3.1.2. Optimality conditions

Although the most general optimality conditions are the well-known Karush-Kuhn-Tucker (KKT) conditions, we will only work, for the time being, with the simpler Lagrange multiplier rules. We refer to Theorem 3.3 for the general KKT conditions.

Theorem 3.1. *Assume that f is convex. For any solution x^* of* (3.1) *there exists a Lagrange multiplier $\lambda(x^*) \geqslant 0$ such that*

$$\begin{cases}\nabla f(x^*) + \lambda(x^*)\nabla g(x^*) = 0\,, \\ \lambda(x^*)g(x^*) = 0.\end{cases} \tag{3.2}$$

There are several proofs and interpretations of this theorem; we refer to Exercise 3.2. Additionally, one should keep in mind the following facts:

(1) First, if $g(x^*) < 0$, one has $\nabla f(x^*) = 0$–this is natural as in this case x^* is in fact a critical point of f.

(2) Second, as always, the Lagrange multiplier rule is a first order optimality conditions; this cannot help to distinguish between saddle points, local minimisers and local maximisers, and one needs to check second-order optimality conditions. In the context of convex constrained optimisation problems, these second-order order optimality conditions would write

$$\begin{cases}\text{If } g(x^*) < 0 \text{ then } \nabla^2 f(x^*) \in S_d^+(\mathbb{R}) \\ \text{If } g(x^*) = 0 \text{ then for any } y \text{ such that } \langle \nabla g(x^*), y\rangle = 0 \\ \text{there holds } \langle (\nabla^2 f(x^*) + \lambda(x^*)\nabla^2 g(x^*))y, y\rangle \geqslant 0\end{cases} \tag{3.3}$$

The question for these optimisation problems is the same as the one for unconstrained optimisation: can we find an efficient iterative method that produces a good numerical approximation of the optimiser x^*?

3.1.3. First option: penalisation techniques

In this approach, we need to go from an inequality constraint to an equality constraint. Introduce, the function g being given,

$$h : x \mapsto \max(g(x), 0)^2.$$

Then

$$K = \{h = 0\},$$

and the function h is both $\mathscr{C}^1$ and convex. We can thus replace (3.1) with

$$\min_{h=0} f. \tag{3.4}$$

Now, let $\varepsilon > 0$ be a small parameter, and replace (3.4) with

$$\min_{x\in\mathbb{R}^d} F_\varepsilon(x) := \left(f(x) + \frac{1}{\varepsilon}h(x)\right). \tag{3.5}$$

As g is coercive, so is h and, provided f is bounded from below, this implies the coercivity of F_ε so that the optimisation problem above is well-posed. We can thus apply the algorithms we saw previously to provide a good (linearly converging) approximation of the minimiser x^*_ε of F_ε. Formally, we can expect that

$$x^*_\varepsilon \underset{\varepsilon\to 0}{\to} x^*,$$

a minimiser of f on K. It is in fact the case, and can be used to derive the multiplier rule, but that does not give a reasonable convergence rate. We refer to Exercise 3.2.

This approach needs to be tuned in a finer way, and we will do so when discussing the augmented Lagrangian method, which is closely related to duality methods.

3.1.4. **Second option for inequality constraints: projection method**

Another natural idea that works well in the case of convex inequality constraints is the projection method. Recall the following basic fact:

Proposition 3.1. *For any closed, convex non-empty set $C \subset \mathbb{R}^d$, for any $x \in \mathbb{R}^d$, there exists a unique $\Pi_C(x) \in C$ such that*

$$\|\Pi_C(x) - x\| = \min_{y \in C} \|y - x\|.$$

The operator Π_C is called the orthogonal projection on C, and is uniquely characterised as the solution of the variational inequality

$$\forall y \in C\,, \langle x - \Pi_C(x), x - y\rangle \geqslant 0.$$

We refer to Exercise 3.8 for a proof of this Proposition.

We thus devise a gradient descent in the following way: fix an initialisation $x_0 \in K$ and a step size $\tau \geqslant 0$. For any $k \in \mathbb{N}$, define

$$x_{k+1} = \Pi_K\left(x_k - \tau \nabla f(x_k)\right). \tag{3.6}$$

As always, there are two questions:

(1) First, does the sequence $\{x_k\}_{k\in\mathbb{N}}$ converge?
(2) If so, can we specify the rate of convergence?

Naturally, since if, x^* being a minimiser, $g(x^*) < 0$, and if $\tau > 0$ is small enough, this is nothing but the gradient descent in the unconstrained case, so that we can hope–at best–to get a linear convergence rate.

The main result is the following:

Theorem 3.2. *Assume that f is α-strongly convex and let ∇f be μ-Lipschitz in K. Then:*

(1) (3.1) has a unique solution x^*.
(2) For any $\tau \in \left(0; \frac{2\alpha}{\mu^2}\right)$, for any initialisation $x_0 \in K$, the sequence generated by the projected gradient algorithm converges linearly to x^*.

Proof of Theorem 3.2. (1) The fact that (3.1) has a solution x^* is immediate. Regarding the uniqueness, assume (3.1) has two distinct solutions x_1^*, x_2^* and consider $\phi : [0;1] \ni t \mapsto f\left((1-t)x_1^* + tx_2^*\right)$. Then ϕ is (strongly) convex. As for any $t \in [0;1]$ we have $(1-t)x_1^* + tx_2^* \in K$ we deduce that

$$\langle \nabla f(x_1^*), x_2^* - x_1^*\rangle\,, \langle \nabla f(x_2^*), x_1^* - x_2^*\rangle \geqslant 0$$

whence

$$\langle \nabla f(x_2^*) - \nabla f(x_1^*), x_2^* - x_1^*\rangle \leqslant 0$$

in contradiction with the monotonicity of the gradient.

(2) Recall (see Exercise 3.8) that Π_K is a 1-Lipschitz function. Consequently, for any $k \in \mathbb{N}$,

$$\begin{aligned}\|x_{k+1} - x_k\|^2 &\leqslant \|(x_k - x_{k-1} + \tau(\nabla f(x_{k-1}) - \nabla f(x_k))\|^2 \\ &= \|x_k - x_{k-1}\|^2 + \tau^2 \|\nabla f(x_k) - \nabla f(x_{k-1})\|^2 \\ &- 2\tau\langle x_k - x_{k-1}, \nabla f(x_k) - \nabla f(x_{k-1})\rangle \\ &\leqslant \|x_k - x_{k-1}\|^2 \left(1 + \tau^2\mu - 2\tau\alpha\right).\end{aligned}$$

In particular, provided

$$\left|1+\tau^2\mu^2-2\tau\alpha\right|<1, \tag{3.7}$$

we obtain linear convergence of the sequence. Observe that condition (3.7) amounts to

$$\begin{cases}\tau<\frac{\alpha}{2\mu^2}\,,\\ 2\tau\alpha-\tau^2\mu^2<2.\end{cases}$$

The second inequality is always satisfied whenever the first one is. Indeed, by the definitions of $\alpha\,,\mu$, $\alpha\leqslant\mu$. Thus, for any $\tau\in\left(0;\frac{\alpha}{2\mu^2}\right)$ there holds

$$2\tau\alpha<\frac{\alpha^2}{\mu^2}\leqslant 1.$$

This concludes the proof.

□

Remark 3.5. *Just as in gradient descent, the strong convexity assumption simplifies our queries, but can be weakened. We refer to Exercise 3.9.*

Although this might seem like the problem is more or less solved, as we have a linearly converging method, this is not satisfactory: indeed, this requires knowing the projection operator Π_K. While it is easy in particular cases, this is not the case for general constraints. A way to overcome this difficulty is to use the theory of duality.

3.2. **Duality & the Uzawa algorithm for inequality constraints**

In this section we will work (as this is now quite important) with N possibly non-linear constraints

$$\forall i\in\{1,\dots,N\}\,,g_i(x)\leqslant 0.$$

Nevertheless, so as not to bother with the qualification of constraints, the running assumption is

$$\text{For any } i,\ g_i \text{ is convex and coercive.} \tag{3.8}$$

We let

$$K:=\cap_{i=1}^N\{g_i\leqslant 0\}$$

and we assume that

$$\exists x_0\,,\forall i\in\{1,\dots,N\}\,,g_i(x_0)<0. \tag{3.9}$$

We consider

$$\min_{x\in K}f(x) \tag{3.10}$$

where f is, as usual, a $\mathscr{C}^1$ function (observe that for the time being we do not require that f be convex).

3.2.1. **Karush-Kuhn-Tucker optimality conditions**

We briefly recall the necessary optimality conditions for (3.10) and, most importantly, how to derive them. We will not go into advanced concepts like the qualification of constraints as these are mostly–for the purposes of this class– technical details that would obscure the presentation. Our goal is to prove the following generalisation of the Lagrange multiplier rule that is due to Karush, Kuhn & Tucker:

Theorem 3.3. *Under the standing assumptions* (3.8)-(3.9)*, if x^* solves* (3.10) *there exist $\lambda_1,\dots,\lambda_N\geqslant 0$ such that*

$$\begin{cases}\nabla f(x^*)+\sum_{i=1}^N\lambda_i\nabla g_i(x^*)=0\,,\\ \sum_{i=1}^N\lambda_i g_i(x^*)=0.\end{cases}$$

Proof of Theorem 3.3. The core of the proof amounts to determining the cone of admissible perturbations at the point x^* or, in other words, the set

$$C(x^*):=\{0\}\cup\left\{y\in\mathbb{R}^d\setminus\{0\}\,,\text{ there exists }\{x_k\}_{k\in\mathbb{N}}\in K^{\mathbb{N}}\,,x_k\underset{k\to\infty}{\to}x^*\,,\frac{x_k-x^*}{\|x_k-x^*\|}\underset{k\to\infty}{\to}\frac{y}{\|y\|}\right\}.$$

Assume that, up to relabelling, $g_1(x^*),\dots,g_n(x^*)<0\,,g_{n+1}(x^*)=\dots=g_N(x^*)=0$. Indeed, if no constraints are active, there is nothing to prove. Observe that

$$\forall y\in C(x^*)\,,\langle\nabla f(x^*),y\rangle\geqslant 0. \tag{3.11}$$

Remark 3.6. *Be careful: that a direction y is admissible does not mean that for any $t\geqslant 0$ small enough $x^*+ty\in K$. However, it means that there exist two sequences $\{t_k\}_{k\in\mathbb{N}}\in(0;+\infty)^{\mathbb{N}}\,,\{y_k\}_{k\in\mathbb{N}}$ such that*

$$t_k\underset{k\to\infty}{\to}0\,,y_k\underset{k\to\infty}{\to}y\text{ and for any }k\in\mathbb{N},\ x^*+t_ky_k\in K.$$

Indeed, letting $y\in C(x^*)\setminus\{0\}$ and choosing any sequence $\{x_k\}_{k\in\mathbb{N}}\in K^{\mathbb{N}}$ in the conditions of the definition, we have

$$f(x^*)\leqslant f(x_k)$$

whence

$$\langle\nabla f(x^*),x_k-x^*\rangle\geqslant 0.$$

Let us now prove that under (3.8)–(3.9) we have the explicit description of the tangent cone as

$$C(x^*)=\{y\in\mathbb{R}^d\,,\forall i\in\{1,\dots,n\}\,,\langle\nabla g_i(x^*),y\rangle\leqslant 0\}. \tag{3.12}$$

Define $\tilde{C}:=\{y\in\mathbb{R}^d\,,\forall i\in\{1,\dots,n\}\,,\langle\nabla g_i(x^*),y\rangle\leqslant 0\}$. The fact that $C(x^*)\subset\tilde{C}$ is immediate. Conversely, let $y\in\tilde{C}\setminus\{0\}$. Assume, up to relabelling, that

$$\langle\nabla g_i(x^*),y\rangle=0\,,1\leqslant i\leqslant n\,,\langle\nabla g_i(x^*),y\rangle<0\,,i\geqslant n.$$

As the g_i are convex Assumption (3.9) guarantees that $\nabla g_1(x^*),\dots,\nabla g_n(x^*)\neq 0$ and, x_0 being a point provided by (3.9), that is, such that

$$\forall i\in\{1,\dots,N\}\,,g_i(x_0)<0$$

we obtain

$$\forall i\in\{1,\dots,N\}\,,\langle\nabla g_i(x^*),x_0-x^*\rangle<0.$$

Indeed, convexity would otherwise imply $g_i(x_0) > 0$, which is absurd. Define, then, $w := x_0 - x^*$ and set, for $\varepsilon > 0$ small enough, $y_\varepsilon := y + \varepsilon w$. Thus, for any $t > 0$ small enough, $x_\varepsilon := x^* + ty_\varepsilon \in K$. Finally, observe that $C(x^*)$ is closed, as a consequence of a simple Cantor diagonal argument. Letting $\varepsilon \to 0$ we deduce that $y \in C(x^*)$, as claimed.

In particular, the optimality condition (3.11) rewrite

$$\{y \in \mathbb{R}^d\,, \forall i \in \{1, \dots, n\} \langle \nabla g_i(x^*), y\rangle \leqslant 0\} \subset \{y \in \mathbb{R}^d\,, \langle \nabla f(x^*), y\rangle \geqslant 0\}. \tag{3.13}$$

But we know from the Farkas-Minkowski lemma (see Exercise 3.8) that this implies the existence of a family $\lambda_1, \dots, \lambda_n \geqslant 0$ such that

$$\nabla f(x^*) + \sum_{i=1}^{n} \lambda_i \nabla g_i(x^*) = 0.$$

Setting $\lambda_{n+1} = \cdots = \lambda_N = 0$ concludes the proof.

□

3.2.2. The augmented Lagrangian method: a primer on duality

The Karush-Kuhn-Tucker conditions suggest looking at the so-called Lagrangian

$$\mathcal{L}(x, \lambda) := f(x) + \sum_{i=1}^{N} \lambda_i g_i(x).$$

A key observation is the following one, that relates the saddle points of the Lagrangian to the solution of the constrained optimisation problem:

Theorem 3.4. *Let x^* be a solution of* (3.10) *and $\lambda(x^*) := (\lambda_1(x^*)\,, \dots, \lambda_N(x^*))$ be the Lagrange multiplier given by Theorem 3.3. Then $(x^*, \lambda(x^*))$ is a saddle-point of $\mathcal{L}$ in the sense that*

$$\min_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda^*) = \mathcal{L}(x^*, \lambda(x^*)) = \max_{\lambda \geqslant 0} \mathcal{L}(x^*, \lambda)$$

where we use the notation $\lambda \geqslant 0$ to signify: for any i, $\lambda_i \geqslant 0$ and $\lambda(x^*) = (\lambda_1(x^*), \dots, \lambda_N(x^*))$.

Conversely, let (x, λ) be a saddle point of $\mathcal{L}$. Then $x \in K$, x solves (3.10) *and λ are the associated the Lagrange multipliers.*

Proof of Theorem 3.4. As all $\lambda_i(x^*)$ are non-negative, $x \mapsto \mathcal{L}(x, \lambda(x^*))$ is convex, so that the condition $\nabla_x \mathcal{L}(x^*, \lambda(x^*)) = 0$ implies the minimality of x^*. Furthermore, fix x^* and let $\mu \geqslant 0$. Assume, up to a relabelling,

$$g_1(x^*)\,, \dots, g_n(x^*) = 0\,, g_i(x^*) < 0\,, i \geqslant n + 1.$$

To check the optimality of $\lambda(x^*)$, it suffices to prove that

$$0 \leqslant \sum_{i=1}^{N} (\lambda_i(x^*) - \mu_i) g_i(x^*) = - \sum_{i=n+1}^{N} \mu_i g_i(x^*),$$

which is obvious.

Now, assume that (x, λ) is a saddle point of $\mathcal{L}$. The optimality in λ implies

$$\forall \mu \geqslant 0\,, \sum_{i=1}^{N} (\lambda_i - \mu_i) g_i(x) \geqslant 0.$$

This immediately implies (letting $\mu_i \to \infty$) that

$$\forall i \in \{1, \dots, N\}, g_i(x) \leqslant 0$$

or, in other words, that $x \in K$. Additionally, this gives

$$\sum_{i=1}^{N} \lambda_i g_i(x) \geqslant 0$$

and thus

$$\forall i \in \{1, \dots, N\}, \lambda_i g_i(x) = 0. \tag{3.14}$$

Furthermore, for any $x' \in K$, we have

$$f(x) = f(x) + \sum_{i=1}^{N} \lambda_i g_i(x) \leqslant f(x') + \sum_{i=1}^{N} \lambda_i g_i(x') \leqslant f(x'),$$

which gives the conclusion.

□

The main result of this part is the following:

Theorem 3.5. *Let x^* be a solution of (3.10) and $\lambda(x^*)$ be the associated Lagrange multiplier. Then*

$$\min_{x \in \mathbb{R}^d} \max_{\lambda \geqslant 0} \mathcal{L}(x, \lambda) = \mathcal{L}(x^*, \lambda(x^*)) = \max_{\lambda \geqslant 0} \min_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda).$$

Proof of Theorem 3.5. On the one hand, we have, for any (x, λ),

$$\inf_{y \in \mathbb{R}^d} \mathcal{L}(y, \lambda) \leqslant \mathcal{L}(x, \lambda) \leqslant \sup_{\mu \geqslant 0} \mathcal{L}(x, \mu)$$

so that

$$\sup_{\lambda \geqslant 0} \inf_{y \in \mathbb{R}^d} \mathcal{L}(y, \lambda) \leqslant \inf_{y \in \mathbb{R}^d} \sup_{\mu \geqslant 0} \mathcal{L}(x, \mu).$$

Furthermore,

$$\mathcal{L}(x^*, \lambda(x^*)) = \sup_{\lambda \geqslant 0} \mathcal{L}(x^*, \lambda)$$

so that

$$\inf_{y \in \mathbb{R}^d} \sup_{\lambda \geqslant 0} \mathcal{L}(y, \lambda) \leqslant \mathcal{L}(x^*, \lambda(x^*)).$$

Thus,

$$\mathcal{L}(x^*, \lambda(x^*)) = \inf_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda(x^*)) \leqslant \sup_{\lambda \geqslant 0} \inf_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda),$$

which provides the reverse inequality.

□

Using the terminology of convex optimisation, we say that convex problems have no duality gaps. At any rate, what matters most now is that we have replaced a potentially nasty optimisation problem with a (continuous) family of constrained optimisation problems, but where the constraints $\lambda \geqslant 0$ are particularly easy to handle (in particular, the projection operator is explicit). This is the basic idea behind the Uzawa algorithm, which will take up the rest of the chapter.

3.2.3. The Uzawa algorithm

The idea of the Uzawa algorithm is to generate two sequences $\{x_k\}_{k\in\mathbb{N}}, \{\lambda_k\}_{k\in\mathbb{N}}$ that should converge to x^* and $\lambda(x^*)$. We introduce the following notations:

$$\mathbb{R}_+^N := \{\lambda \in \mathbb{R}^N, \forall i \in \{1,\dots,N\}, \lambda_i \geqslant 0\}, g(x) = (g_1(x),\dots,g_N(x)).$$

An important observation is the following one: for any $\tau > 0$,

$$\lambda(x^*) = \Pi_{\mathbb{R}_+^N}\left(\lambda(x^*) + \tau g(x^*)\right). \tag{3.15}$$

The proof of (3.15) is straightforward: first, observe that for any $y \in \mathbb{R}^N$

$$\Pi_{\mathbb{R}_+^N}(y) = (\max(y_i, 0))_{i=1,\dots,N}.$$

Second, on each coordinate, either $g_i(x^*) < 0$ and $\lambda_i(x^*) = 0$, or $g_i(x^*) = 0$ and $\lambda_i(x^*) \geqslant 0$.

Back to the Uzawa algorithm, which goes as follows:

(1) Start from any $\lambda_0 \in \mathbb{R}_+^N$ and fix a step size $\tau > 0$.

(2) For any $k \in \mathbb{N}$, set

$$x_k := \underset{\mathbb{R}^d}{\operatorname{argmin}}\left(f(x) + \langle \lambda_k, g(x)\rangle\right)$$

and

$$\lambda_{k+1} := \Pi_{\mathbb{R}_+^N}\left(\lambda_k + \tau g(x_k)\right).$$

Now, can we expect to have convergence for this algorithm and, if so, at which rate? To do so, let us first remain at a formal level. Assume that for any $\lambda \in \mathbb{R}_+^N$ there exists a unique x_λ that solves

$$\ell(\lambda) := \min_{x\in\mathbb{R}^d}(f(x) + \langle \lambda, g(x)\rangle = \mathcal{L}(x,\lambda)).$$

Assume furthermore that the map $\lambda \mapsto x_\lambda$ is differentiable. Then, it appears that

$$\begin{aligned}\nabla\ell(\lambda) &= \nabla_x\mathcal{L}(x_\lambda,\lambda)\nabla_\lambda x_\lambda + \nabla_\lambda\mathcal{L}(x_\lambda,\lambda)\\ &= \nabla_\lambda\mathcal{L}(x_\lambda,\lambda) \text{ because of the optimality of } x_\lambda\\ &= g(x_\lambda),\end{aligned}$$

so that the Uzawa algorithm appears as a real projected gradient ascent. In particular, this means that we should expect convergence, and maybe even linear convergence. Unfortunately, while the convergence indeed holds under suitable assumption, the rate is not always known. We summarise this in the following theorem:

Theorem 3.6. *Assume that f is α-strongly convex and that g is convex and μ-Lipschitz. Then for any $\tau \in (0; \frac{2\alpha}{\mu^2})$, the sequence $\{x_k\}_{k\in\mathbb{N}}$ converges to the solution x^* of (3.10).*

Proof of Theorem 3.6. We proceed in several steps: first of all, observe that for any $\lambda \in \mathbb{R}_+^N$, letting x_λ be the unique solution of

$$\min_{x\in\mathbb{R}^d}\left(f(x) + \langle\lambda, g(x)\rangle\right)$$

there holds, for any $y \in \mathbb{R}^d$,

$$\langle\nabla f(x_\lambda), y - x_\lambda\rangle + \langle\lambda, g(y) - g(x_\lambda)\rangle \geqslant 0. \tag{3.16}$$

Proof of (3.16). This is a consequence of convexity: let $y \in \mathbb{R}^d$. Then for any $t \in [0,1]$ there holds

$$f((1-t)x_\lambda + ty) - f(x_\lambda) + \langle \lambda, g((1-t)x_\lambda + ty) - g(x_\lambda)\rangle \geqslant 0.$$

As g is convex and $\lambda \geqslant 0$ this implies

$$f((1-t)x_\lambda + ty) - f(x_\lambda) + \langle \lambda, -tg(x_\lambda) + tg(y)\rangle \geqslant 0.$$

Dividing by t and letting $t \to 0$ provides the required inequality. □

An important consequence of (3.16) is the following:

$$\begin{cases} \langle \nabla f(x_k), x^* - x_k\rangle + \langle \lambda_k, g(x^*) - g(x_k)\rangle \geqslant 0\,, \\ \langle \nabla f(x^*), x_k - x^*\rangle + \langle \lambda^*, g(x_k) - g(x^*)\rangle \geqslant 0. \end{cases}$$

Combining these two inequalities we deduce

$$\langle \nabla f(x^*) - \nabla f(x_k), x_k - x^*\rangle + \langle \lambda^* - \lambda_k, g(x_k) - g(x^*)\rangle \geqslant 0. \tag{3.17}$$

Using the α-strong convexity of f, (3.17) in turn implies

$$0 \leqslant \alpha \|x_k - x^*\|^2 \leqslant \langle \lambda^* - \lambda_k, g(x_k) - g(x^*)\rangle. \tag{3.18}$$

Now, observe that from (3.15) and the fact that projections are 1-Lipschitz we also have the estimate

$$\begin{aligned} \|\lambda_{k+1} - \lambda^*\|^2 &\leqslant \|\lambda_k - \lambda^* + \tau(g(x_k) - g(x^*))\|^2 \\ &= \|\lambda_k - \lambda^*\|^2 + \tau^2\mu^2\|x_k - x^*\|^2 + 2\tau\langle \lambda_k - \lambda^*, g(x_k) - g(x^*)\rangle \\ &\leqslant \|\lambda_k - \lambda^*\|^2 + \|x_k - x^*\|^2 \left(\tau^2\mu^2 - 2\alpha\tau\right). \end{aligned}$$

Thus, if τ is chosen so that

$$\tau < \frac{2\alpha}{\mu^2}$$

we obtain

$$\|\lambda_{k+1} - \lambda^*\|^2 \leqslant \|\lambda_k - \lambda^*\|^2.$$

The sequence $\{\|\lambda_k - \lambda^*\|^2\}_{k\in\mathbb{N}}$ is consequently non-increasing, and so it converges. Passing to the limit in

$$\|\lambda_{k+1} - \lambda^*\|^2 \leqslant \|\lambda_k - \lambda^*\|^2 + \|x_k - x^*\|^2 \left(\tau^2\mu^2 - 2\alpha\tau\right)$$

we obtain

$$\|x_k - x^*\|^2 \underset{k\to\infty}{\rightarrow} 0.$$

□

Observe that this proof does not give convergence to the solution of the dual problem; nevertheless, in certain specific cases, it is possible to show it. We refer to Exercise (3.11).

3.3. **Exercises of the chapter**

3.3.1. **Generalities on constrained optimisation**

Exercise 3.1. Let f be convex and K be a compact, convex set equal to the closure of its interior. Show that

$$\max_K f = \max_{\partial K} f$$

and that this maximum can always be reached at an extremal point of K.

Exercise 3.2. We consider the constrained optimisation problem

$$\min_{g(x)\leqslant 0} f(x)$$

where f, g are strictly convex functions and we assume that g is a coercive and satisfies $g(x_0) < 0$ for some $x_0 \in \mathbb{R}^d$. We want to give several proofs or interpretation of Theorem 3.1. We shall always assume that there exists a unique solution x^* to the underlying optimisation problem, that further satisfies

$$g(x^*) = 0.$$

(1) A geometric interpretation What does the Lagrange multiplier rule mean geometrically? If this is not clear for you, try to grasp it, following for instance the following reasoning:
   (a) For any $\varepsilon > 0$ consider the level set
   $$A_\varepsilon := \{f = f(x^*) - \varepsilon\}$$
   and assume that this is a smooth submanifold. Why do $A_\varepsilon \cap \{g = 0\} = \emptyset$?
   (b) What are the normal vectors to the surfaces A_ε and $\{g = 0\}$? Why is x^* the "first" intersection point between the two manifolds? What can you conclude?

(2) A proof by parametrised curves This is a more standard proof. Show that for any curve $\gamma : [-1; 1] \to \{g = 0\}$ such that $\gamma(0) = x^*$ there holds
$$\langle \nabla f(x^*), \gamma'(0)\rangle = 0$$
and conclude.

(3) A proof by penalisation This is yet another standard and important proof. Consider the optimisation problem
$$\min_{g(x)=0} f(x).$$
We assume that f , g, are strictly convex and coercive, and that there exists a point x_0 such that $g(x_0) < 0$. Finally, we assume that $\nabla f \neq 0$ on $\{g = 0\}$ and that there exists a unique minimiser to the constrained problem. We introduce for any $\varepsilon > 0$ the function
$$f_\varepsilon(x) = f(x) + \frac{1}{2\varepsilon} g(x)^2.$$
Show that f_ε admits a minimiser x_ε.

(4) Show that $\{g(x_\varepsilon)/2\varepsilon\}_{\varepsilon\to 0}$ is bounded.

(5) Conclude.

Exercise 3.3. Let K be a convex, compact subset of $\mathbb{R}^d$. Show that there exists a convex, coercive function g such that

$$K = \{g \leqslant 0\}$$

and such that there exists $x_0 \in \mathbb{R}^d$ with $g(x_0) < 0$.

Exercise 3.4. Using the Lagrange multiplier rule, solve the following constrained optimisation problems:

(1) Maximise $f(x, y) = xy$ subject to $x^2 + y^2 \leqslant 2$.
(2) Minimise $f(x, y) = (x-4)^2 + (y-4)^2$ subject to $x + 3y \leqslant 9$ and $x + y \leqslant 4$.

Exercise 3.5. Solve the optimisation problem: minimise $x^2 + y^2$ subject to $(x-1)^2 = 0$. Does the Lagrange multiplier rule hold? Why?

3.3.2. Theoretical aspects of constrained optimisation problems

Exercise 3.6. We want to prove two classical inequalities using constrained optimisation.

(1) The goal is the prove the Hölder inequality using the Lagrange multiplier rule. Fix $p\,, q \in (1; +\infty)$ such that $1/p + 1/q = 1$. Consider a vector $x \in \mathbb{R}^d \setminus \{0\}$. Study the optimisation problem

$$\max_{y \in \mathbb{R}^d\,, \sum_{i=1}^d |y_i|^q = 1} \langle x, y \rangle$$

and characterise the optimisers using the Lagrange multiplier rule.

(2) The goal is to prove the discrete rearrangement inequality using the Lagrange multiplier rule. Let $x \in \mathbb{R}^d$ and assume that all its components are non-negative: for any i, $x_i \geqslant 0$. Furthermore, assume that the components of x are non-increasing:

$$x_1 \geqslant x_2 \geqslant \cdots \geqslant x_d.$$

Let $y \in \mathbb{R}^d$ be a vector whose components are non-negative. Find the solution of

$$\max_{\sigma \in \mathfrak{S}_d} \sum_{i=1}^d x_i y_{\sigma(i)}$$

where $\mathfrak{S}_d$ is the group of permutations of a set with d elements. Find an easy condition on x to ensure the uniqueness of maximisers.

Hint: try a direct approach. Do not use the Lagrange multiplier rule as the set $\mathfrak{S}_d$ is discrete.

Exercise 3.7. The goal of this exercise is to study the isoperimetric inequality in the polygonal case and, more generally, to tackle geometric problems using constrained optimisation as a main tool.

(1) A first geometric optimisation problem We consider a convex polygon inscribed in the unit disk in $\mathbb{R}^d$. In other words, we consider $0 = \theta_1 \leqslant \theta_2 \leqslant \cdots \leqslant \theta_n < 2\pi$, and the associated polygon is the convex hull of $\{e^{i\theta_k}\}_{k=1,\dots n}$. Show that the perimeter of such a polygon P is maximal when

$$\theta_1 = 0\,, \theta_{k+1} - \theta_k = \frac{2\pi}{n}.$$

(2) The polygonal isoperimetric inequality (*) We want to solve the following problem: given an perimeter constraint L and a fixed number of sides N, what is the polygon with N sides that maximises the enclosed area? We represent a polygon in $\mathbb{R}^2$ by a collection of point $\{(x_i, y_i)\}_{i=1,\dots,N}$, ordered in a counterclockwise fashion.
   (a) What is the perimeter of such a polygon?
   (b) Show that an optimal polygon exists.
   (c) We admit that the area of this polygon is given by

$$A = \frac{1}{2}\sum_{k=1}^{N}(x_k y_{k+1} - y_k x_{k+1}).$$

   (d) Write down the optimality conditions for the optimisation problem.
   (e) Solve this optimality system and conclude.

3.3.3. **Numerical methods for constrained optimisation**

Exercise 3.8. [Existence and uniqueness of a projection, application] The goal of this exercise is to (re)prove Proposition 3.1. We let C be a closed, non-empty convex subset of a Hilbert space H.

(1) Show that for any $y \in \mathbb{R}^d$ there exists a unique $y_C \in C$ such that

$$\|y - y_C\| = \min_{x \in C} \|y - x\|.$$

(2) Show that y_C is the unique solution of the variational inequality

$$\forall x \in C\,, \langle y - y_C, x - y_C\rangle \leqslant 0.$$

(3) We define $\Pi_C(y) := y_C$. Show that Π_C is 1-Lipschitz.
(4) Deduce from the first two questions that for any closed subspace E of H there holds:

$$E \oplus E^{\perp} = H.$$

(5) Deduce from the previous question that for any continuous linear form $\ell \in \mathcal{L}(H, \mathbb{R})$ there exists $x_\ell \in H$ such that $\ell = \langle x_\ell, \cdot\rangle$.
(6) We now want to prove the Farkas-Minkowski lemma: let $a_1, \dots, a_n \in \mathbb{R}^d$ and $b \in \mathbb{R}^d$. Assume that

$$\cap_{i=1}^{n}\{\langle a_i, \cdot\rangle \geqslant 0\} \subset \{\langle b, \cdot\rangle \geqslant 0\}.$$

Then there exist $\lambda_1, \dots, \lambda_n \geqslant 0$ such that

$$b = \sum_{i=1}^{n} \lambda_i a_i.$$

   (a) Show the lemma when $n = 1$, using the projection Π_a on the set $\mathbb{R}_+ a$.
   (b) In the case $n = 2$, propose a geometric interpretation of this lemma.
   (c) Consider, in the general case, the cone

$$C := \{\sum_{i=1}^{n} \alpha_i a_i\,, \alpha_i \geqslant 0\}.$$

   Show that C is closed and convex when the vectors a_i are linearly independent, and conclude in that case.
   (d) Show the lemma in the general case.

Exercise 3.9. In this exercise, we study the projected gradient descent under a simple convexity assumption (*i.e.* we do not require the function f to be strongly convex). To be more precise, we consider a function f that we assume is convex and with a μ-Lipschitz gradient. We let $K \subset \mathbb{R}^d$ be a compact, convex subset of $\mathbb{R}^d$. We fix a step size $\tau > 0$ and, starting from any $x_0 \in K$, we define the sequence $\{x_k\}_{k\in\mathbb{N}}$ as

$$x_{k+1} := \Pi_K \left(x_k - \tau \nabla f(x_k)\right),$$

Π_K being the projection on K. The main goal of this exercise is to show that

$$f(x_k) - \min_K f \underset{k\to\infty}{\to} 0. \tag{3.19}$$

We set

$$y_k := x_k - x_{k+1} = x_k - \Pi_K(x_k - \tau\nabla f(x_k)).$$

(1) Show that x_{k+1} solves

$$x_{k+1} = \underset{x\in K}{\operatorname{argmin}} \left(\langle \nabla f(x_k), x - x_k\rangle + \frac{1}{2\tau}\|x - x_k\|^2 \right)$$

and deduce that

$$\forall x' \in K\,, \left\langle \nabla f(x_k) + \frac{1}{\tau}(x_{k+1} - x_k), x' - x_{k+1} \right\rangle \geqslant 0.$$

(2) Show that whenever $\tau \in \left(0; \frac{1}{\mu}\right)$

$$f(x_{k+1}) - f(x_k) \leqslant -\frac{1}{2\tau}\|y_k\|^2.$$

(3) Let x^* be a minimiser of f in K. Show that for any $k \in \mathbb{N}$ there holds

$$f(x_{k+1}) - f(x^*) \leqslant f(x_{k+1}) - f(x_k) - \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \langle \nabla f(x_k), x_{k+1} - x^*\rangle.$$

(4) Show that

$$\langle \nabla f(x_k), x_{k+1} - x^*\rangle \leqslant \frac{1}{2\tau}\left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 - \|x_{k+1} - x_k\|^2\right).$$

(5) Conclude that

$$f(x_{k+1}) - f(x^*) \leqslant \frac{1}{2\tau}\left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\right).$$

(6) Conclude.

(7) Assume that f is, in addition, strictly convex. Show that $\{x_k\}_{k\in\mathbb{N}}$ converges to x^*.

3.3.4. **Around the Uzawa method & duality**

Exercise 3.10. Let $f : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ be such that ∇f is locally Lipschitz. We assume that there exists $\alpha > 0$ such that

$$\forall (x, z) \in \mathbb{R}^n \times \mathbb{R}^d\,, \nabla_z^2 f(x, z) \geqslant \alpha \mathrm{Id}.$$

We define

$$\ell(x) := \min_{z\in\mathbb{R}^d} f(x, z).$$

(1) Show that ℓ is well-defined, and that the optimisation problem has a unique solution z_x.

(2) Show that the map $x \mapsto z_x$ is Lipschitz continuous.

(3) Show that ℓ is differentiable, and that

$$\nabla \ell(x) = \nabla_x f(x, z_x).$$

Exercise 3.11. We want to investigate certain situations where the Uzawa algorithm provides not only convergence of the approximations of the primal problem, but also of the dual problem. To this end, we consider two functions $f\,, g$ that satisfy the following assumptions:

(1) f and g are convex, M_0-Lipschitz with M_1-Lipschitz gradients.
(2) f is α-strongly convex.
(3) There exists $x_0 \in \mathbb{R}^d$ such that $g(x_0) < 0$.

We consider the constrained optimisation problem

$$\min_{x\,, g(x) \leqslant 0} f(x),$$

and we let $\{x_k, \lambda_k\}_{k\in\mathbb{N}}$ be the sequence generated by the Uzawa algorithm. Similarly, we let

$$\mathcal{L}(x, \lambda) := f(x) + \lambda g(x)$$

be the associated Lagrangian.

(1) Recall why, for any $\tau > 0$ small enough, the sequence generated by the Uzawa algorithm with step-size $\tau > 0$ converges to the solution of the primal problem.
(2) Let $\lambda\,, \mu \geqslant 0$ be fixed, and let $x_\lambda\,, x_\mu$ minimise $\mathcal{L}(\cdot, \lambda)$ and $\mathcal{L}(\cdot, \mu)$ respectively. Show that there exists a constant M_2 (independent of λ and μ) such that

$$\|x_\lambda - x_\mu\| \leqslant M_2|\lambda - \mu|.$$

(3) Show that $\lambda \mapsto \ell(\lambda) := \min_{x\in\mathbb{R}^d} \mathcal{L}(x, \lambda)$ is either strictly concave or linear, and has a Lipschitz continuous gradient.
(4) Conclude.

3.4. **Correction of the exercises**

Solution of Exercise 3.1. We first do it assuming that f is a linear map, that is

$$f(x) = \langle a, x\rangle.$$

In that case, if f is minimised at some interior x^* then for any $y \in K$ $\langle a, y\rangle = 0$, which is impossible as K has non-empty interior, unless $a = 0$.

To come back to the general case, let $x \in K$ and let y solve

$$\max y \in K\langle \nabla f(x), y\rangle.$$

This is a linear optimisation problem, a solution of which is an extreme point. By convexity, $f(y) \geqslant f(x)$.

Solution of Exercise 3.2. (1) (a) Argue by contradiction: if for some $\varepsilon > 0$ there exists $x_\varepsilon \in A_\varepsilon \cap \{g = 0\}$ $f(x_\varepsilon) = f(x^*) - \varepsilon < f(x^*)$, in contradiction with the definition of x^*.

(b) The normal vector to A_ε at a point z is $\frac{\nabla f(z)}{\|\nabla f(z)\|}$. Similarly, the normal vector to $\{g = 0\}$ at a point x is $\frac{\nabla g(x)}{\|\nabla g(x)\|}$. Furthermore, as $\varepsilon \to 0$, we claim that the two level sets must touch tangentially; as is obvious on a drawing. Consequently, the two normal vectors should be equal, yielding the conclusion.

(2) (a) It suffices to apply to chain rule to the mapping $\phi : t \mapsto f \circ \gamma(t)$, as

$$\phi'(t) = \langle \nabla f(\gamma(t)), \dot{\gamma}(t) \rangle.$$

In particular, as $\phi'(0) = 0$, this gives the conclusion. Now, observe that this means that the vector $\nabla f(x^*)$ is orthogonal to the entire tangent space at x^*, so that it has to be normal. As the normal vector is $\frac{\nabla g(x^*)}{\|\nabla g(x^*)\|}$ we derive the required conclusion.

(3) (a) As f is coercive, so is f_ε.

(b) As

$$f_\varepsilon(x_\varepsilon) \leqslant f_\varepsilon(x^*) = f(x^*),$$

the conclusion follows.

(c) At a point of minimum,

$$\nabla f(x_\varepsilon) + \frac{g(x_\varepsilon)}{\varepsilon} \nabla g(x_\varepsilon) = 0.$$

Now, taking any closure point of the sequence $\{x_\varepsilon\}_{\varepsilon \to 0}$ we deduce that any closure point is a minimiser of the constrained problem. Furthermore, if we assume that the minimiser is unique, we can pass to the limit in the equation above: as $\nabla g(x^*) \neq 0$ we have

$$\left| \frac{g(x_\varepsilon)}{\varepsilon} \right| = \frac{\|\nabla f(x_\varepsilon)\|}{\|\nabla g(x_\varepsilon)\|}$$

and we can then pass to the limit.

Solution of Exercise 3.3. It suffices to consider the Minkowski gauge: up to a translation, we might assume that $0 \in \mathrm{Int}(K)$. We then define

$$p_K(x) := \inf\{\lambda \geqslant 0 : \, x \in \lambda K\}$$

and to define

$$g(x) := p_K(x) - 1.$$

Solution of Exercise 3.4. (1) We let $f(x, y) = xy \, , g(x, y) = x^2 + y^2 - 2$. The set $\{g \leqslant 0\}$ is compact, so that the existence of a minimiser is guaranteed. At a point of minimum (x^*, y^*) there exists $\lambda \geqslant 0$ such that

$$\begin{cases} y + 2\lambda x = 0 \, , \\ x + 2\lambda y = 0 \, , \\ \lambda(x^2 + y^2 - 2) = 0. \end{cases}$$

In particular, either $x = y = \lambda = 0$, but this is impossible as 0 is not a local minimiser, or $x = -y \, , \lambda = \frac{1}{2}$. In this case, the problem becomes:

$$\min_{x \, , x^2 \leqslant 1} -x^2,$$

so that the only two solutions are $x = \pm 1 \, , y = \mp 1$.

(2) We set $f(x, y) = (x-4)^2 + (y-4)^2 \, , g_1(x, y) = x + 3y - 9 \, , g_2(x, y) = x + y - 4$. The function f is coercive so that a minimum exists. At a minimum (x^*, y^*)

there exist $\lambda_1\,,\lambda \geqslant 0$ such that

$$\begin{cases}2(x-4)+\lambda_1+\lambda_2=0\,,\\ 2(y-4)+3\lambda_1+\lambda_2=0\,,\\ \lambda_1(x+3y-9)=0\,,\\ \lambda_2(x+y-4)=0.\end{cases}$$

We do a case by case analysis:

(a) If $\lambda_1=\lambda_2=0$ we obtain $x=y=4$, which violates the condition $x+y-4\leqslant 0$.

(b) If $\lambda_1\neq 0\,,\lambda_2=0$ we deduce that $x\approx 3.3\,,y\approx 1.7$, which violates the condition $x+y-4\leqslant 0$.

(c) If $\lambda_1\neq 0\,,\lambda_2\neq 0$ then we obtain $y=\frac{5}{2}\,,x=\frac{3}{2}$ and $\lambda_1+\lambda_2=5\,,3\lambda_1+\lambda=3$, which has obviously no non-negative solutions.

(d) If $\lambda_1=0\,,\lambda_2\neq 0$ we obtain $x=y$ and $x+y=4$ whence $x=y=2$. This is the only admisible solution, and is thus the global solution to the optimisation problem.

Solution of Exercise 3.5. Observe that the problem does not fulfil qualification conditions. Furthermore, if we apply the Lagrange multiplier rule then we must

$$2x+2\lambda(x-1)=0\,,y=0\,,\lambda(x-1)^2=0.$$

However, if $x=1$, then $x=0$ and if $\lambda=0$ $x=0$, which is not admissible. This is due to the fact that the gradient of the constraint is zero at any admissible point.

Solution of Exercise 3.6. (1) First of all observe that we can, without loss of generality, assume that x is component-wise non-negative: for any $i\in\{1,\dots,d\}$, $x_i\geqslant 0$, which means that, for any optimiser y^*, we can also assume without loss of generality that y^* is component-wise non-negative. By the Lagrange multiplier rule, we deduce that there exists a real number λ such that, for any $i\in\{1,\dots,d\}$,

$$x_i+\lambda(y_i^*)^{q-1}=0.$$

From the constraints $\sum(y_i^*)^q=1$ we obtain

$$\lambda=-\sum_{i=1}^N x_i^{\frac{q}{q-1}}=-\sum_{i=1}^N x_i^p\neq 0$$

so that finally, either $x_i=0$, in which case $y_i^*=0$, or $x_i\neq 0$, in which case $y_i^*=\frac{x_i^{\frac{1}{q-1}}}{\left(\sum_{i=1}^N x_i^p\right)^{\frac{1}{q-1}}}$. Thus, for any $y\,,\|y\|_{\ell^q}=1$,

$$\langle x,y\rangle\leqslant\frac{\sum_{i=1}^N x_i^{1+\frac{1}{q-1}}}{\left(\sum_{i=1}^N x_i^p\right)^{\frac{1}{q-1}}}=\left(\sum_{i=1}^N x_i^p\right)^{\frac{1}{p}},$$

which is exactly the Hölder inequality.

(2) We let σ^* be an optimal perturbation. Let $i\neq j$ be two distinct integers and let $\tau=(i\,j)$ be the permutation associated with these two indices. As

$$\sum_{i=1}^d x_iy_{\sigma^*(i)}\geqslant\sum_{i=1}^d x_iy_{\sigma^*\circ\tau(i)}$$

we deduce that

$$x_i y_{\sigma^*(i)} + x_j y_{\sigma^*(j)} \geqslant x_i y_{\sigma^*(j)} + x_j y_{\sigma^*(i)}.$$

This rewrites

$$(x_i - x_j)(y_{\sigma^*(i)} - y_{\sigma^*(j)}) \geqslant 0.$$

In particular, σ^* must be chosen so that

$$y_{\sigma^*(1)} \geqslant \cdots \geqslant y_{\sigma^*(d)}.$$

The uniqueness is guaranteed if we have

$$x_1 > \cdots > x_d,$$

and this is actually a necessary and sufficient condition.

Solution of Exercise 3.7. (1) We begin by computing the perimeter of a polygon described by its angles. By a simple computation, we obtain

$$P(\theta_1, \dots, \theta_N) = 2\sum_{i=1}^{N} \left| \sin\left(\frac{\theta_{i+1} - \theta_i}{2}\right) \right|.$$

Furthermore, we have

$$\theta_{i+1} - \theta_i \leqslant 2\pi$$

whence $0 \leqslant \frac{\theta_{i+1}-\theta_i}{2} \leqslant \pi$ and so

$$P(\theta_1, \dots, \theta_N) = 2\sum_{i=1}^{N} \sin\left(\frac{\theta_{i+1} - \theta_i}{2}\right).$$

At a point of maximum, we observe (by a simple drawing) that

$$\theta_{i+1} - \theta_i > 0$$

so that the only active constraint is

$$\sum_{i=1}^{N} \theta_i = 2\pi.$$

Thus, there exists a Lagrange multiplier λ such that

$$\cos\left(\frac{\theta_{i+1} - \theta_i}{2}\right) = -\lambda.$$

As $\frac{\theta_{i+1}-\theta_i}{2} \leqslant \pi$ we deduce that the quantity $\theta_{i+1} - \theta_i$ does not depend on the index i, which gives the result.

(a) We have

$$P = \sum_{k=1}^{N} \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}.$$

(b) We fix $(x_1, y_1) = (1, 0)$ (up to a translation). The perimeter bound $P = L$ gives the compactness of the design set.

(c) At an optimal polygon, there exists a constant λ such that, first,

$$\forall k \in \{1, \dots, N\}, \lambda \left(\frac{x_k - x_{k-1}}{\sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}} - \frac{x_{k+1} - x_k}{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}} \right) + (y_{k+1} - y_{k-1}) = 0,$$

which we rewrite, introducing

$$p_k := \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}$$

$$\lambda \left(\frac{x_k - x_{k-1}}{p_k} - \frac{x_{k+1} - x_k}{p_{k+1}} \right) + (y_{k+1} - y_{k-1}) = 0.$$

Furthermore, we also have

$$\lambda \left(\frac{y_k - y_{k-1}}{p_k} - \frac{y_{k+1} - y_k}{p_{k+1}} \right) + (x_{k+1} - x_{k-1}) = 0.$$

(d) We introduce

$$z_k := (x_{k+1} - x_k) + i(y_{k+1} - y_k)$$

so that the previous system rewrites

$$z_k + z_{k-1} - \lambda i \left(\frac{z_{k-1}}{p_k} - \frac{z_k}{p_{k+1}} \right) = 0,$$

whence

$$z_k \left(1 - \frac{\lambda i}{p_{k+1}} \right) = -z_{k-1} \left(1 + \frac{\lambda i}{p_k} \right)$$

Taking the modulus on both sides yields

$$p_{k+1}^2 + \lambda^2 = p_k^2 + \lambda^2.$$

Thus, all p_k's are equal and the optimal has sides of the same length $\bar{p}$. Furthermore, this implies

$$\frac{z_{k+1}}{z_k} = -\frac{\bar{p} + i\lambda}{\bar{p} - i\lambda} = e^{i\theta}$$

for a given angle, and so the optimal polygon is regular.

Solution of Exercise 3.8. (1) The existence follows from a standard compactness argument. Introduce the function

$$f : C \ni x \mapsto \|y - x\|^2.$$

f is strongly convex and, in particular, has a unique minimiser on C; this follows from arguments seen in the lectures.

(2) Define, for $x \in C$, $\varphi : t \mapsto f((1-t)y_c + tx)$. Thus, $\varphi'(0) \geqslant 0$. However,

$$\varphi'(0) = 2\langle x - y_C, y_C - y \rangle.$$

Conversely, if a point y_C satisfies this inequality then, for any x, the function φ (defined above) being convex, $f(x) \geqslant f(y_C)$.

(3) We consider two points y, z. From the previous question, we deduce

$$\begin{cases} \langle y - y_C, z_c - y_C \rangle \leqslant 0 \\ \langle z - z_C, y_C - z_C \rangle \leqslant 0 \end{cases}$$

whence we obtain

$$\langle (y - z) + (z_C - y_C), z_C - y_C \rangle \leqslant 0.$$

As a consequence,

$$\|z_C - y_C\|^2 \leqslant \langle z-y, z_C-y_C\rangle \leqslant \|z-y\|\cdot\|y_C-z_C\|$$

and the conclusion follows.

(4) We consider Π_E the projection onto E. Then we obtain, from the optimality conditions, that for any $x\in E$ there holds

$$\langle y-\Pi(E), x-y\rangle \leqslant 0.$$

Consequently, the linear map

$$x\mapsto \langle x, y-\Pi(E)\rangle$$

is bounded, and is thus equal to 0, whence $y-\Pi(E)\in E^\perp$ and the conclusion follows.

(5) We now prove the Riesz theorem. Assume that $\ell\neq 0$ and write

$$\mathbb{R}^d = \ker(\ell)\oplus\ker(\ell)^\perp.$$

Let $x\in\ker(\ell)^\perp\setminus\{0\}$. Of course, $\ell(x)\neq 0$. Now, for any $z\in\mathbb{R}^d$,

$$\ell(z-\ell(z)/\ell(x)x)=0$$

whence, by orthogonality,

$$\langle z-\frac{\ell(z)}{\ell(x)}x, x\rangle = 0.$$

We deduce the result

(6) (a) We suppose that $b\neq 0$ otherwise there is nothing to prove. When $n=1$, we define

(b)

(c)

(d)

Solution of Exercise 3.9. (1) One needs to complete the square. Indeed,

$$\begin{aligned}\langle\nabla f(x_k), x-x_k\rangle + \frac{1}{2\tau}\|x-x_k\|^2 &= \frac{1}{2\tau}\left(\|x-x_k\|^2 + 2\tau\langle\nabla f(x_k), x-x_k\rangle\right)\\ &= \frac{1}{2\tau}\left(\|x-x_k+\tau\nabla f(x_k)\|^2 - \tau^2\|\nabla f(x_k)\|^2\right).\end{aligned}$$

The definition of the orthogonal projection yields the desired result. The last inequality is the optimality condition at x_{k+1}: consider any $x'\in K$ and let $x_t := (1-t)x_{k+1}+tx'$. Define

$$\varphi : t\mapsto \langle\nabla f(x_k), x_t - x_{k+1}\rangle + \frac{1}{2\tau}\|x_t-x_k\|^2.$$

We have $\varphi'(0)\geqslant 0$ whence

$$\langle\nabla f(x_k), x'-x_{k+1}\rangle + \frac{1}{\tau}\langle x'-x_{k+1}, x_{k+1}-x_k\rangle \geqslant 0.$$

(2) By standard estimates, we have

$$\begin{aligned}f(x_{k+1})-f(x_k) &\leqslant \langle\nabla f(x_k), x_{k+1}-x_k\rangle + \frac{\mu}{2}\|x_{k+1}-x_k\|^2\\ &\leqslant \langle\nabla f(x_k), x_{k+1}-x_k\rangle + \frac{1}{2\tau}\|x_{k+1}-x_k\|^2\\ &\leqslant \langle\nabla f(x_k), x_{k+1}-x_k\rangle + \frac{1}{\tau}\|x_{k+1}-x_k\|^2 - \frac{1}{2\tau}\|x_{k+1}-x_k\|^2\end{aligned}$$

$$\leqslant \frac{1}{\tau}\langle \tau\nabla f(x_k)+x_{k+1}-x_k, x_{k+1}-x_k\rangle - \frac{1}{2\tau}\|x_{k+1}-x_k\|^2.$$

Introduce $z := x_k - \tau\nabla f(x_k)$ and observe that

$$\begin{aligned}\langle \tau\nabla f(x_k)+x_{k+1}-x_k, x_{k+1}-x_k\rangle &= \langle \tau\nabla f(x_k)+\Pi_K(z)-x_k, \Pi_K(z)-x_k\rangle\\ &= \langle \Pi_K(z)-z, \Pi_k(z)-x_k\rangle\\ &\leqslant 0\end{aligned}$$

by the fundamental property of orthogonal projections. This concludes the proof.

(3) Since f is convex, we have

$$\begin{aligned}f(x_{k+1})-f(x^*) &\leqslant f(x_{k+1})-f(x_k)-\langle \nabla f(x_k), x^*-x_k\rangle\\ &\leqslant f(x_{k+1})-f(x_k)-\langle \nabla f(x_k), x_{k+1}-x_k\rangle - \langle \nabla f(x_k), x^*-x_{k+1}\rangle.\end{aligned}$$

(4) Observe that from the first question

$$\begin{aligned}\langle \nabla f(x_k), x_{k+1}-x^*\rangle &\leqslant \frac{1}{\tau}\langle x_{k+1}-x_k, x^*-x_{k+1}\rangle\\ &= \frac{1}{2\tau}\left(\|x_k-x^*\|^2-\|x_{k+1}-x_k\|^2-\|x_{k+1}-x^*\|^2\right).\end{aligned}$$

(5) All the previous estimates lead to

$$\begin{aligned}f(x_{k+1})-f(x^*) \leqslant f(x_{k+1})-f(x_k)-\langle \nabla f(x_k), x_{k+1}-x_k\rangle - \frac{1}{2\tau}\|x_{k+1}-x_k\|^2\\ +\frac{1}{2\tau}\left(\|x_k-x^*\|^2-\|x_{k+1}-x^*\|^2\right)\\ \leqslant \frac{1}{2\tau}\left(\|x_k-x^*\|^2-\|x_{k+1}-x^*\|^2\right).\end{aligned}$$

(6) We can conclude as in the lecture notes: the sequence $\{\|x_k-x^*\|^2\}_{k\in\mathbb{N}}$ is non-increasing, and thus converges, which implies the convergence of $\{f(x_k)\}_{k\in\mathbb{N}}$ to $f(x^*)$.

(7) As f is strictly convex, it has at most one minimiser and, from the previous questions, any closure point x_∞ of $\{x_k\}_{k\in\mathbb{N}}$ is a minimiser. Consequently, the sequence has a unique closure point, and thus converges.

Solution of Exercise 3.10. (1) By strong convexity of $z \mapsto f(x,z)$, the existence and uniqueness of z_x are guaranteed.

(2) Let $x_1, x_2 \in \mathbb{R}^n$ and z_1, z_2 be such that $\ell(x_i) = f(x_i, z_i)$ $(i=1,2)$. Then

$$\begin{aligned}\frac{\alpha}{2}\|z_1-z_2\|^2+\langle \nabla_z f(x_1,z_1), z_2-z_1\rangle &\leqslant f(x_1,z_2)-f(x_1,z_1)\\ &= f(x_2,z_1)-f(x_2,z_1)+f(x_1,z_2)-f(x_1,z_1)\\ &\leqslant f(x_2,z_1)-f(x_2,z_2)+f(x_1,z_2)-f(x_1,z_1).\end{aligned}$$

Introducing the function $F : x \mapsto f(x,z_1)-f(x,z_2)$, the right-hand side rewrites $F(x_2)-F(x_1)$, so that there exists $\xi \in [x_1;x_2]$ satisfying

$$F(x_2)-F(x_1) = \langle \nabla F(\xi), x_2-x_1\rangle.$$

As $\nabla F(\xi) = \nabla_x f(x,z_1)-\nabla_x f(x,z_2)$ we have

$$\nabla F(\xi) \leqslant L\|z_1-z_2\|$$

and the conclusion follows by the Cauchy-Schwarz inequality.

(3) We let h be a vector. Then

$$\begin{aligned}\ell(x+h)-\ell(x) &\geqslant f(x,z_{x+h})-f(x,z_{x+h})\\ &= \langle \nabla_x f(x,z_{x+h}),h\rangle + o(\|h\|).\end{aligned}$$

Observe that $z_{x+h} \underset{h\to 0}{\to} z_x$. Similarly,

$$\begin{aligned}\ell(x+h)-\ell(x) &\leqslant f(x+h,z_x)-f(x,z_x)\\ &= \langle \nabla_x f(x,z_x),h\rangle + o(\|h\|).\end{aligned}$$

Passing to the limit provides the required result.

Solution of Exercise 3.11. (1) We refer to the lecture notes.

(2) Observe that by strong convexity of f and convexity of g we have

$$\frac{\alpha}{2}\|x_\lambda - x_\mu\|^2 + \langle \nabla f(x_\lambda), x_\mu - x_\lambda\rangle + \lambda\langle \nabla g(x_\lambda), x_\mu - x_\lambda\rangle \leqslant \mathcal{L}(\lambda, x_\mu) - \mathcal{L}(\lambda, x_\lambda).$$

By optimality of x_λ, $\nabla f(x_\lambda)+\lambda\nabla g(x_\lambda)=0$ whence

$$\frac{\alpha}{2}\|x_\lambda - x_\mu\|^2 \leqslant \mathcal{L}(\lambda, x_\mu) - \mathcal{L}(\lambda, x_\lambda).$$

However,

$$\begin{aligned}\mathcal{L}(\lambda, x_\mu) &= f(x_\mu)+\lambda g(x_\mu)\\ &= f(x_\mu)+\mu g(x_\mu)+(\lambda-\mu)g(x_\mu)\\ &\leqslant f(x_\lambda)+\mu g(x_\lambda)+(\lambda-\mu)g(x_\mu)\\ &= \mathcal{L}(\lambda,x_\lambda)+(\lambda-\mu)(g(x_\mu)-g(x_\lambda))\end{aligned}$$

so that

$$\frac{\alpha}{2}\|x_\lambda - x_\mu\|^2 \leqslant |\lambda-\mu|\cdot|g(x_\mu)-g(x_\lambda)|.$$

The Lipschitzianity of g allows to conclude.

(3) The concavity of ℓ follows from the following observation: let $\lambda_1, \lambda_2 \geqslant 0$, let x_i be the minimiser associated with λ_i $(i=1,2)$, define $\lambda_t := (1-t)\lambda_1 + t\lambda_2$ and let $x_t \in \mathbb{R}^d$ be the unique minimiser of $x \mapsto f(x)+\lambda_t g(x)$. We have

$$\begin{aligned}\ell(\lambda_t) &= f(x_t)+\lambda_t g(x_t)\\ &= (1-t)f(x_t)+tf(x_t)+(1-t)\lambda_1 g(x_t)+t\lambda_2 g(x_t)\\ &= (1-t)\left(f(x_t)+\lambda_1 g(x_t)\right)+t\left(f(x_t)+\lambda_2 g(x_t)\right)\\ &\geqslant (1-t)\left(f(x_1)+\lambda_1 g(x_1)\right)+t\left(f(x_2)+\lambda_2 g(x_2)\right).\end{aligned}$$

In particular, we deduce that:

- ℓ is concave,
- And that, if ℓ is not strictly concave, we can find $\lambda_1, \lambda_2 \geqslant 0$, $\lambda_1 \neq \lambda_2$ and $t \in (0;1)$ such that x_1 minimisers $f+\lambda_t g$. In particular, we obtain, for this t, two sets of optimality conditions: on the one-hand,

  $$\nabla f(x_1)+\lambda_1\nabla g(x_1)=0$$

  and, on the other,

  $$\nabla f(x_1)+\lambda_t\nabla g(x_1)=0.$$

  In particular, substracting these two equations, we obtain

  $$(\lambda_1-\lambda_2)\nabla g(x_1)=0.$$

This implies that $\nabla f(x_1) = 0$ and, in turn, that for any $\mu \geqslant 0$,

$$\nabla f(x_1) + \mu \nabla g(x_1) = 0.$$

Thus, x_1 minimises $f + \mu g$ for any $\mu \geqslant 0$ and so

$$\ell(\lambda) = f(x_1) + \lambda g(x_1)$$

and is in particular linear.

Furthermore, by the Danskin theorem,

$$\ell'(\lambda) = g(x_\lambda),$$

and as $\lambda \mapsto x_\lambda$ is Lipschitz, so is ℓ'.

(4) It suffices to apply Exercise 9 in the case of a strictly concave function. If ℓ is, on the other hand, linear, it follows from a direct computation.

3.5. **Computer session**

3.5.1. **Computing a projection using the Uzawa algorithm**

We consider a compact, convex subset $K \subset \mathbb{R}^d$

3.5.2. **Geometric problems**

In this first exercise, you are left completely free. Your goal is to solve, numerically, the two geometric problems of Exercise 3.7. Namely, your function must take, as an argument, the geometric quantities of the problem: in the first case, you should fix the number of sides or of angles. Then, computing the perimeter, minimise it under constraints, using (for instance) a penalised method, as in Exercise 3.2.

3.5.3. **The obstacle problem**

Let g be a given continuous function on the interval $[0,1]$. We consider an *obstacle problem*: find a function $u : [0,1] \longrightarrow \mathbb{R}$ such that:

$$\begin{cases} -u''(x) \geqslant 1 & x \in (0,1) \\ u(x) \geqslant g(x) & x \in (0,1) \\ (-u''(x) - 1)\,(u(x) - g(x)) = 0 & x \in (0,1) \\ u(0) = u(1) = 0 \end{cases}$$

The first equation represents a minimum concavity for the function u, the second equation represents the obstacle: u must remain above g. The third equation expresses the fact that we must satisfy at least one of the two previous equations with equality: either we solve $-u''(x) = 1$, or $u(x) = g(x)$, and we are on the obstacle.

Associated minimization problem

We discretize this problem by introducing a uniform mesh: $x_j = jh$, where h is the step size of the mesh, and $j \in \{0, \ldots, n+1\}$, with $n \geqslant 1$ an integer and $h = \frac{1}{n+1}$. Let $g_j = g(x_j)$ for $j \in \{0, \ldots, n+1\}$. We seek values $u_j = u(x_j)$ for $j \in \{0, \ldots, n+1\}$ such that:

$$\begin{cases} -\dfrac{u_{j-1} - 2u_j + u_{j+1}}{h^2} \geqslant 1 & j \in \{0, \ldots, n+1\} \\ u_j \geqslant g_j & j \in \{0, \ldots, n+1\} \\ \left(-\dfrac{u_{j-1} - 2u_j + u_{j+1}}{h^2}\right)(u_j - g_j) = 0 & j \in \{0, \ldots, n+1\} \\ u_0 = u_{n+1} = 0 \end{cases}$$

Recall that $-\frac{u_{j-1}-2u_j+u_{j+1}}{h^2}$ is the finite difference approximation of $-u''(x_j)$. We introduce the matrix $A \in S_n(\mathbb{R})$, defined by:

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \ldots & 0 \\ -1 & 2 & -1 & \vdots & \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \ldots & 0 & -1 & 2 \end{pmatrix}.$$

We also define the column vectors b and g of $\mathbb{R}^n$ as follows:

$$b = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \quad g = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}.$$

Recall that if $u = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}$, then

$$u \text{ is a solution of (2)} \Longleftrightarrow u \text{ is a solution of } \left\{ \begin{array}{l} \min\limits_{v \in K} \left\{ \dfrac{1}{2}(Av, v) - (b, v) \right\} \\ K = \{v \in \mathbb{R}^n : v \geqslant g\} \end{array} \right.$$

We define the function J on $\mathbb{R}^n$ by:

$$J(v) = \frac{1}{2}(Av, v) - (b, v).$$

Problem Resolution

Consider the specific case where $g(x) = \max\left(0, 1 - 100(x - 0.7)^2\right)$.

We want to solve this problem using the projected gradient algorithm. Denote by $\Pi_K = \{v \geqslant g\}$ the projection onto the convex set $K =$. Without proving it, we can use the fact that:

$$\Pi_K(v) = \left(\max(v_i, g_i)\right)_{1 \leqslant i \leqslant n}.$$

(1) Write a gradient method with a fixed step size to find the minimum of J over $\mathbb{R}^n$. Test this method and verify that it converges to the desired result.

(2) Adapt the previous program to implement the projected gradient method with a constant step size. For instance, you can choose the optimal step size $\rho_{\text{opt}} = \frac{2}{\lambda_1(A) + \lambda_n(A)}$ of the gradient method without constraints. Also, don't forget to set a maximum number of iterations and a relevant stopping criterion.

(3) Test the program for different values of n. Represent the graph of the solution and the graph of the obstacle on the same plot. Verify that if $u(x) \neq g(x)$, then $-u''(x) = 1$ (using the finite difference approximation).

(4) Implement the Uzawa algorithm to solve this problem and compare the efficiency of the methods implemented. Test with different choices of obstacles.

Regarding the end of the lectures

At this point, you should have mastered the basics in (un)constrained optimisation. The remainder of this class will be devoted to the exploration, of course in less details, of more recent topics. In particular, these last lectures will give a very brief overview of the following:

(1) The stochastic gradient descent, which is by now a central algorithm in machine learning (you will touch on that in the exercise session for this chapter).

(2) The link between gradient descent and continuous time differential equations, which will be used in two major directions:

  (a) The first one will be the study of another stochastic method, the simulated annealing.

  (b) The second one will be the study of acceleration in gradient descent.

4. Stochastic gradient descent for finite sum optimisation & Neural Networks

4.1. **Finite sum optimisation problems & stochastic gradient descent**

4.1.1. **General introduction**

In this chapter, we will be studying problems with a particular structure, the so-called finite sum functions. Namely, we consider a function f that writes

$$f : x \mapsto \frac{1}{N}\sum_{k=1}^{N} f_k(x), \tag{4.1}$$

where each f_k is a scalar valued function. The basic problem under consideration is

$$\min_{x} f(x).$$

The stochastic gradient descent has a simple objective: although we know that under mild assumptions the constant step-size gradient descent converges, the numerical computation of the gradient might be extremely costly. The idea behind the stochastic gradient descent is to just compute one (or a few) of the gradients ∇f_k, these being chosen at random. Before we proceed with the mathematical formalism, let us briefly explain where these types of problems naturally pop up.

An important example

A fundamental example is that of Neural Networks. For the sake of simplicity, we only discuss shallow Neural Networks with one layer (we refer to the following chapter for deep Neural Networks). A neural network is an application from $\mathbb{R}^d$ to $\mathbb{R}^m$ which consists of a triplet (σ, A, b) where

(1) $A \in M_{m,d}(\mathbb{R})$ is the set of *linear weights*,
(2) $b \in \mathbb{R}^m$ is the *bias*,
(3) $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ is the *activation function*.

In this very basic presentation we do not talk about either depth or width. A neural network takes an entry $x \in \mathbb{R}^d$ and outputs a result $y \in \mathbb{R}^m$ through the relation

$$y = \sigma\left(Ax + b\right).$$

The point is that we are given a (very large) number of data points $\{x_i, y_i\}_{i=1,\dots,N}$, and we want to find the best linear weights A and bias b such that

$$y_i \approx \sigma\left(Ax_i + b\right),$$

and to then uses these parameters A and b to predict the result for another entry x not already present in the data set. This can be used, for instance, in classification of data or in model prediction. The basic way to approach this is to solve the minimisation problem

$$\min_{A\,,b} \frac{1}{N}\sum_{i=1}^{N} \|y_i - \sigma(Ax_i + b)\|^2,$$

which has the exact shape of (4.1). There are, as always, two main questions when faced with such approximation problems:

(1) The first one is the approximation property: can we approximate the targets arbitrarily close? This is the problem of universal approximation, which is discussed in two of the exercises.

(2) The second one is the question of numerical approximation of the optimal parameters of the network.

We will tackle these questions in the reverse order.

4.1.2. The stochastic gradient descent I: the classical setting

The basic idea

The principle of the stochastic gradient descent is simple: fix a step-size $\tau > 0$ (which, in this chapter, we will call the learning rate) and proceed according to the following procedure:

(1) Start from a $x_0 \in \mathbb{R}^d$.
(2) For $k \in \mathbb{N}$, pick an index $j(\omega)$ at random in $\{1, \dots, N\}$ according to the uniform probability.
(3) Set $x_{k+1} := x_k - \tau \nabla f_{j(\omega)}(x_k)$.

The standing assumptions on $\{f_k\}_{k=1,\dots,N}$ are as follows:

$$\begin{cases} \text{For any } k \in \{1, \dots, N\},\ f_k \text{ is convex,} \\ f \text{ is coercive,} \\ \text{There exists } \mu \text{ such that for any } k \in \{1, \dots, N\},\ \nabla f_k \text{ is } \mu\text{-Lipschitz.} \end{cases} \tag{4.2}$$

The question is then: what type of behaviour can we expect? Of course, nothing as good as for the full gradient descent; in particular, observe that in general the stochastic gradient descent is *not* a descent method, although it is on average. In order to prove this, we need some preparatory material. We begin with the following basic proposition.

Proposition 4.1. *The following properties hold*

(1) The random vector $\nabla f_{j(\cdot)}(x)$ is a unbiased estimator of $\nabla f(x)$:

$$\mathbb{E}\left(\nabla f_{j(\cdot)}(x)\right) = \nabla f(x).$$

(2) Under assumption (4.2), the stochastic gradient descent satisfies

$$\mathbb{E}\left(f(x_{k+1})\right) \leqslant f(x_k) - \tau \|\nabla f(x_k)\|^2 + \frac{\mu\tau^2}{2}\mathbb{E}\left(\|\nabla f_{j(\cdot)}\|^2\right).$$

Proof of Proposition 4.1. (1) The first point is immediate.

(2) Likewise, this is quickly proved:

$$\begin{aligned} \mathbb{E}\left(f(x_{k+1})\right) &= \frac{1}{N^2}\sum_{i,j=1}^N f_i\left(x_k - \tau\nabla f_j(x_k)\right) \\ &= \frac{1}{N^2}\sum_{i,j=1}^N \left(f_i(x_k) + \int_0^1 \langle \nabla f_i(x_k - t\tau\nabla f_j(x_k)), -\tau\nabla f_j(x_k)\rangle dt\right) \\ &\leqslant f(x_k) - \tau\frac{1}{N^2}\sum_{i,j=1}^N \langle \nabla f_i(x_k), \nabla f_j(x_k)\rangle + \frac{\mu\tau^2}{2N^2}\sum_{i,j=1}^N \langle \nabla f_j(x_k), \nabla f_i(x_k)\rangle \\ &= f(x_k) - \tau\|\nabla f(x_k)\|^2 + \frac{\mu\tau^2}{2}\mathbb{E}\left(\|\nabla f_{j(\cdot)}(x_k)\|^2\right). \end{aligned}$$

□

If we want to proceed further, we need to estimate $\mathbb{E}(\|\nabla f_{j(\cdot)}(x_k)\|^2)$ in a finer way. The key tool to do so is the notion of *variance* of a finite-sum function f.

Variance of a finite-sum function

Definition 4.1. *Let f be of the form (4.1) and assume that f is strongly convex and coercive. Let x^* be the unique minimiser of f. We define the* variance *of f as*

$$\sigma_f := \mathbb{E}(\|\nabla f_{j(\cdot)}(x^*)\|^2).$$

The key lemma is the following:

Lemma 4.1. *Under Assumption (4.2), for any $x \in \mathbb{R}^d$,*

$$\mathbb{E}\left(\|\nabla f_{j(\cdot)}(x)\|^2\right) \leqslant 4\mu\left(f(x) - \min f\right) + 2\sigma_f.$$

Proof of Lemma 4.1. Fix $i \in \{1, \dots, N\}$. As f_i is convex, we deduce that for any $x\,, y \in \mathbb{R}^d$,

$$\frac{1}{2\mu}\|\nabla f_i(x) - \nabla f_i(y)\|^2 \leqslant f_i(y) - f_i(x) - \langle \nabla f_i(x), y - x\rangle. \tag{4.3}$$

Proof of (4.3). □

Let x^* be a minimiser of f. Observe that for any $x \in \mathbb{R}^d$, for any $i \in \{1, \dots, N\}$, there holds

$$\begin{aligned}\|\nabla f_i(x)\|^2 &\leqslant 2\|\nabla f_i(x) - \nabla f_i(x^*)\|^2 + 2\|\nabla f_i(x^*)\|^2 \\ &\leqslant 4\mu\left(f_i(x) - f_i(x^*)\right) - 4\mu\langle \nabla f_i(x^*), x - x^*\rangle + 2\|\nabla f_i(x)\|^2.\end{aligned}$$

Taking the expectation provides the conclusion. □

We are now ready to state our convergence result for stochastic gradient descent.

Convergence of the stochastic gradient algorithm

The main theorem of this section is the following:

Theorem 4.1. *Assume (4.2), that f is α-strongly convex and minimal at x^*, and let $x_0 \in \mathbb{R}^d$. Then, for any $\tau \in \left(0; \frac{1}{2\mu}\right)$, $\{x_k\}_{k\in\mathbb{N}}$ being the sequence generated by the stochastic gradient descent, there holds*

$$\mathbb{E}\left[\|x_k - x^*\|^2\right] \leqslant (1 - 2\tau\alpha)^k \|x_0 - x^*\|^2 + \frac{\tau}{\mu}\sigma_f. \tag{4.4}$$

Remark 4.1 (Regarding the assumptions)**.** *As usual, we can obtain different classes of results depending on the type of assumptions we enforce on the function f (for instance, we could obtain weaker convergence results if f is merely assumed to have a smooth gradient, or we can relax some assumptions to f satisfying a Polyak-Lojasiewicz inequalities). As this is just an introduction to stochastic gradient descent, we detail some of these cases in the exercises.*

Proof of Theorem 4.1. We know that

$$\|x_{k+1} - x^*\|^2 = \|x_k - x^*\|^2 - 2\tau\langle \nabla f_{j(\cdot)}(x_k), x_k - x^*\rangle + \tau^2\|\nabla f_{j(\cdot)}(x_k)\|^2$$

so that, taking the expectation with respect to x_k, we obtain

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1} - x^*\|^2\right] &= \|x_k - x^*\|^2 - 2\tau\langle \nabla f(x_k), x_k - x^*\rangle + \tau^2\mathbb{E}\left[\|\nabla f_{j(\cdot)}(x_k)\|^2\right] \\ &\leqslant \|x_k - x^*\|^2 - 2\tau\alpha\|x_k - x^*\|^2 - 2\tau(f(x_k) - f(x^*)) \\ &\quad + 4\tau^2\mu(f(x_k) - f(x^*)) + 2\tau^2\sigma_f\end{aligned}$$

$$\begin{aligned}&\text{from Lemma 4.1}\\ &\leqslant (1-2\tau\alpha)\left\|x_k-x^*\right\|^2+(4\tau^2\mu-2\tau)(f(x_k)-f(x^*))+2\tau^2\sigma_f.\end{aligned}$$

Thus, if $\tau \in \left(0; \frac{1}{2\mu}\right)$ we obtain

$$\mathbb{E}\left[\|x_{k+1}-x^*\|^2\right] \leqslant (1-2\tau\alpha)\left\|x_k-x^*\right\|^2+2\tau^2\sigma_f.$$

Taking the expectation with respect to $x_1,\dots,x_{k-1}$ this yields

$$\begin{aligned}\mathbb{E}\left[\|x_k-x^*\|^2\right] &\leqslant (1-2\tau\alpha)^k\|x_0-x^*\|^2+2\tau^2\sigma_f\sum_{i=0}^{k-1}(1-2\tau\mu)^i\\ &\leqslant (1-2\tau\alpha)^k\|x_0-x^*\|^2+\frac{\tau\sigma_f}{\mu}.\end{aligned}$$

□

4.1.3. The Stochastic Gradient Descent II: mini-batches and variance reduction

Still continuing our exploration of stochastic method, we focus on another aspect, variance reduction techniques. To be more specific, inspecting the estimate provided by Theorem 4.1 we observe that the error that is made is of order σ_f, which quantifies the deviation between the condition $\nabla f = 0$ and the individual conditions $\nabla f_k = 0$. A possibility, in order to overcome this hurdle, is to pick a family of random directions rather than just one. This is the purpose of mini-batching.

Definition 4.2. *Let f be of the form (4.1) and let $B \subset \{1,\dots,N\}$. The mini-batch gradient of f associated with B is the vector*

$$\nabla f_B = \frac{1}{|B|}\sum_{i\in B}\nabla f_i.$$

Now, we fix a batch size n_B. The mini-batch stochastic gradient descent algorithm reads as follows:

(1) Start from a $x_0 \in \mathbb{R}^d$.
(2) For $k \in \mathbb{N}$, pick a subset $B \subset \{1,\dots,N\}$ at random according to the uniform probability on all subsets of size n_B of $\{1,\dots,N\}$.
(3) Set $x_{k+1} := x_k - \tau\nabla f_{B(\omega)}(x_k)$.

Of course, when $n_B = 1$ the mini-batch and the standard Stochastic Gradient Descent coincide. When $n_B = N$n this is just the standard gradient descent.

Analogous to Definition 4.1, we can introduce the variance associated with a mini-batch:

Definition 4.3. *We assume that f is coercive and α-strictly convex. Let x^* be the unique minimiser of f and n_B be a mini-batch size. The variance of f at x^* is defined as*

$$\sigma_{f,n_B} = \mathbb{E}\left[\|\nabla f_{B(\cdot)}(x^*)\|^2\right].$$

Similar to Lemma 4.1, we have the following estimate:

Lemma 4.2. *Assume f satisfies (4.2), is coercive and α strictly convex. Then, for any $x \in \mathbb{R}^d$,*

$$\mathbb{E}\left[\|\nabla f_{B(\cdot)}(x)\|^2\right] \leqslant 4\mu(f(x)-f(x^*))+2\sigma_{f,n_B}.$$

The main theorem of this section is the following convergence result:

Theorem 4.2. *Assume (4.2), that f is α-strongly convex and minimal at x^*, and let $x_0 \in \mathbb{R}^d$. Then, for any $\tau \in \left(0; \frac{1}{2\mu}\right)$, $\{x_k\}_{k\in\mathbb{N}}$ being the sequence generated by the stochastic gradient descent, there holds*

$$\mathbb{E}\left[\|x_k - x^*\|^2\right] \leqslant (1-2\tau\alpha)^k \|x_0 - x^*\|^2 + \frac{\tau}{\alpha}\sigma_{f,n_B}. \tag{4.5}$$

Remark 4.2. *Naturally, $\sigma_{f,N} = 0$, in which case we recover the usual rate of the (deterministic) gradient descent. In fact, there is a very simple expression for σ_{f,n_B}, which allows to show that $n \mapsto \sigma_{f,n_B}$ is non-increasing (and decreasing unless $\sigma_f = 0$). We refer to Exercise 4.4.*

Proof of Theorem 4.2. The proof is essentially the same as that of Theorem 4.1. Namely, we let x^* be the unique minimiser of f. We have

$$\begin{aligned}\|x_{k+1} - x^*\|^2 &= \|x_k - x^* - \tau\nabla f_B(x_k)\|^2 \\ &= \|x_k - x^*\|^2 - 2\tau\langle\nabla f_B(x_k), x_k - x^*\rangle + \tau^2\|\nabla f_B(x_k)\|^2.\end{aligned}$$

Taking the expectation, this provides

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1} - x^*\|^2\right] &= \mathbb{E}\left[\|x_k - x^*\|^2\right] + \tau^2\mathbb{E}[\|\nabla f_{B(\cdot)}(x_k)\|^2] - 2\tau\langle\nabla f(x_k), x_k - x^*\rangle \\ &\leqslant \mathbb{E}\left[\|x_k - x^*\|^2\right] - 2\alpha\tau\|x_k - x^*\|^2 - 2\alpha\mu\tau(f(x_k) - f(x^*)) \\ &+ 4\mu\tau^2\left(f(x_k) - f(x^*)\right) \\ &+ 2\tau^2\sigma_{f,n_B}.\end{aligned}$$

The conclusion follows exactly as before. □

4.1.4. The stochastic Gradient Descent III: Projected Stochastic Gradient Descent

To conclude this (ever so short) introduction to the stochastic gradient descent, we present a similar result for the case of constrained optimisation, under a strong convexity assumption on the function f.

As always, we assume that f is a finite sum function satisfying (4.2). We fix a compact, convex set $K \subset \mathbb{R}^d$, and we are interested in the problem

$$\min_{x\in K} f. \tag{4.6}$$

We let Π_K denote the projection on the set K. Finally, we consider a sequence $\{\tau_k\}_{k\in\mathbb{N}}$ of step-sizes, and we define the Projected Stochastic Gradient Descent, starting from some $x_0 \in K$, as

$$x_{k+1} = \Pi_K\left(x_k - \tau_k\nabla f_{i(\cdot)}(x_k)\right).$$

Theorem 4.3. *Assume that f is α strongly convex and define the sequence $\{\tau_k\}_{k\in\mathbb{N}}$ as*

$$\forall k \in \mathbb{N}, \tau_k = \frac{1}{\alpha k}.$$

Then, letting, for any $k \in \mathbb{N}$,

$$\langle x\rangle_k = \frac{1}{k}\sum_{i=1}^k x_i$$

there holds

$$\mathbb{E}\left[f(\langle x\rangle_k) - f(x^*)\right] \leqslant M\frac{1+\log(k)}{\alpha^2 k}$$

with

$$M = \sup_{i=1,\dots,N} \|f_i\|^2_{\mathscr{C}^1(K)}.$$

Proof of Theorem 4.3. Observe that, defining M as in the statement of the theorem we have, for any $x \in K$,

$$\mathbb{E}[\|\nabla f_{i(\cdot)}(x)\|^2] \leqslant M.$$

Now, using the fact that Π_K is 1-Lipschitz, we obtain, for any $k \in \mathbb{N}$,

$$\begin{aligned}\mathbb{E}[\|x_{k+1} - x^*\|^2] &\leqslant \|x_k - x^*\|^2 - 2\tau_k\langle \nabla f(x_k), x_k - x^*\rangle + \tau_k^2 \mathbb{E}[\|\nabla f_{i(\cdot)}(x_k)\|^2] \\ &\leqslant \|x_k - x^*\|^2 - 2\tau_k\left(f(x_k) - f(x^*) - \frac{\alpha}{2}\|x_k - x^*\|^2\right) + \tau_k^2 M.\end{aligned}$$

Rearranging all these terms, we deduce that

$$2(f(x_k) - f(x^*)) \leqslant \frac{1-\alpha\tau_k}{\tau_k}\|x_k - x^*\|^2 - \frac{1}{\tau_k}\mathbb{E}[\|x_{k+1} - x^*\|^2] + \tau_k M.$$

Summing and using the convexity of f, we obtain

$$\begin{aligned}2\left(\mathbb{E}f\left(\frac{1}{k}\sum_{i=1}^k x_i\right) - f(x^*)\right) &\leqslant \frac{2}{k}\left(\sum_{i=1}^k (\mathbb{E}(f(x_i) - f(x^*))\right) \\ &\leqslant \frac{2}{k}\sum_{i=0}^k\left(\frac{1-\alpha\tau_i}{\tau_i}\mathbb{E}[\|x_i - x^*\|^2] - \frac{1}{\tau_i}\mathbb{E}[\|x_{i+1} - x^*\|^2] + \tau_k M\right) \\ &\leqslant \frac{2}{k}\sum_{i=0}^k\left(\frac{1-\alpha\tau_i}{\tau_i}\mathbb{E}[\|x_i - x^*\|^2] - \frac{1}{\tau_{i+1}}\mathbb{E}[\|x_{i+1} - x^*\|^2] + \tau_k M\right) \\ &\leqslant \frac{2\|x_0 - x^*\|^2}{k} + \frac{2}{k}\sum_{i=0}^k \tau_i M.\end{aligned}$$

The conclusion follows by the usual equivalent of the harmonic series. □

4.2. **Exercises of the chapter**

4.2.1. **Around neural networks**

In this first set of exercises, we investigate some basic aspects of Neural Networks To alleviate notations, we first consider a single layer (or shallow) Neural Networks, seen as an application from $\mathbb{R}^d$ to $\mathbb{R}^m$ and which consists of a triplet (σ, A, b) where

1. $A \in M_{m,d}(\mathbb{R})$ is the set of *linear weights*,
2. $b \in \mathbb{R}^m$ is the *bias*,
3. $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ is the *activation function*.

A neural network takes an entry $x \in \mathbb{R}^d$ and outputs a result $y \in \mathbb{R}^m$ through the relation

$$y = \sigma\left(Ax + b\right).$$

Depending on the type of applications we have in mind, the dimensions and the choice of activation functions will be crucial.

In these exercises, we will see some basic examples of Neural Networks and we will provide some universality properties.

Exercise 4.1. [The Perceptron Leaning Algorithm] The first example of Neural Network we study is the Perceptron, which is widely used in *classification* of data sets. In this setting, we have a set of data $X \subset \mathbb{R}^d$ that we want to label with the signs + or $-$ (typically, if the set X represents images, we want to assign a + to the image if the image represents a rabbit or a $-$ if it does not).

In terms of Neural Networks, we have $m = 1$ and the activation function is the Heaviside function

$$H : x \mapsto \mathbb{1}_{\mathbb{R}_+^*}(x).$$

We say that a function $f : X \to \{+, -\}$ is a *linear classification* of X if there exists a vector $w \in \mathbb{R}^d\,, b \in \mathbb{R}^d$ such that $\{f = +\} \subset \{\langle w, \cdot\rangle > 0\} + b$ and $\{f = -\} \subset \{\langle w, \cdot\rangle < 0\} + b$. In that case, we say that w generates this classification (and we will always assume $b = 0$).

The goal of this exercise is to show that there exists an algorithm *that converges in a finite number of sets* to a vector w_f that generates this classification.

1. *Preliminary* For any given set C we let $\mathrm{Conv}(C)$ denote the convex hull of C. Prove the Carathéodory theorem: for any $C \subset \mathbb{R}^d$ and any $x \in \mathrm{Con}(C)$, there exist $c_1, \dots, c_{d+1} \in C$ and $\alpha_1, \dots, \alpha_{d+1} \in [0; 1]$ such that
$$\sum_{i=1}^{d+1} \alpha_i = 1\,, x = \sum_{i=1}^{d+1} \alpha_i c_i.$$
2. Let $f : X \mapsto \{+, -\}$. Show that the following statements are equivalent:
   (a) f is a linear classification,
   (b) $\mathrm{Conv}(f^-{-}1(\{+\})) \cap \mathrm{Conv}(f^{-1}(\{-\})) = \emptyset$.
3. We now assume that f is a linear separation of a dataset $X = \{x_1, \dots, x_N\}$. We extend it to a sequence $X \in (\mathbb{R}^d)^{\mathbb{N}}$ by setting $x_{N+j} = x_j$ for $j \in \{1, \dots, N\}$. We let $\tau > 0\,, w_0 \in \mathbb{R}^d \setminus \{0\}$ and we define the sequence $\{w_j\}_{j\in\mathbb{N}}$ by setting
   (a) If $f(x_j)\langle w_{j-1}, x_j\rangle > 0$ then $w_j = w_{j-1}$,
   (b) Else, $w_j = w_{j-1} + \tau f(x_j) x_j$.
   The goal of this question is to prove that there exists $J \in \mathbb{N}^*$ such that for any $j \geqslant J$,
$$f(x_j)\langle w_J, x_j\rangle > 0.$$

(a) Why does it suffice to prove that the set

$$A := \{j \in \mathbb{N}^*\,, f(x_j)\langle w_{j-1}, x_j\rangle \leqslant 0\}$$

is finite?

(b) We introduce the following notations: n_j is the cardinal $\#(A \cap [\![1;j]\!])$ with $n_0 = 0$, $R := \sup_{x\in X}\|x\|$, and, for any $i \in \mathbb{N}$, $\delta_i := f(x_i)\frac{\langle w, x_i\rangle}{\|w\|} \geqslant \delta > 0$. Show by induction that for any $j \in \mathbb{N}$ there holds

$$\begin{cases}\langle w, w_0\rangle + \tau\delta n_j\|w\| \leqslant \langle w, w_j\rangle \\ \|w_j\|^2 \leqslant \|w_0\|^2 + \tau^2R^2n_j.\end{cases}$$

(c) Conclude that n_j is bounded and prove the result.

Exercise 4.2 (Universality of Neural Networks I)**.** In this exercise, we begin our investigation of the so-called “universality” of Neural Networks, starting with the one-dimensional case. We focus on *shallow* neural networks, which write, for $a \in \mathbb{R}^N\,, b \in \mathbb{R}^N\,, c \in \mathbb{R}^N$,

$$F(a, b, c; \cdot) : [0; 1] \ni x \mapsto \sum_{i=1}^N c_i\sigma(a_ix + b),$$

where σ is the Rectified Linear Unit (ReLu):

$$\sigma : y \mapsto y_+ := \max(0; y).$$

The number N is called the *width* of the Neural Network. We want to prove the following fact: given a function $f \in \mathscr{C}^0([0; 1])$ and $\varepsilon > 0$, there exist $N\,, a\,, b\,, c$ such that

$$\|f - F(a, b, b; \cdot)\|_{L^\infty([0;1])} \leqslant \varepsilon$$

(1) Why does it suffice to do it for a piecewise affine function?

(2) We let f be a piecewise linear function, parameterised by

$$0 = x_1 < x_2 < \cdots < x_m = 1$$

and we assume that $f' \equiv \alpha_i$ on $(x_i; x_{i+1})$. Show that there exist parameters $N\,, a\,, b\,, c$ such that

$$f = F(a, b, c; \cdot).$$

Exercise 4.3 (Universality of Neural Networks II)**.** The goal of this exercise is to prove a general universality theorem, which essentially state that shallow Neural Networks of arbitrary widths can approximate any function. We present the proof of the oldest of these approximation results, which is due to Cybenko. We let σ, for the time being, be a $\mathscr{C}^0$ function from $\mathbb{R}$ to $\mathbb{R}$. We say that σ has the *universal approximation property* if the following holds: for any $f \in \mathscr{C}^0([0; 1]^d; \mathbb{R})$ and any $\varepsilon > 0$ there exist $N \in \mathbb{N}$, $\{y_i\}_{i=1,\dots,N} \in (\mathbb{R}^d)^N$, $\{\theta_i\}_{i=1,\dots,N} \in \mathbb{R}^N$, $\{\alpha_i\}_{i=1,\dots,N}$, such that

$$\left\| f - \sum_{k=1}^N \alpha_k\sigma\left(\langle y_k, \cdot\rangle + \theta_k\right)\right\|_{L^\infty([0;1]^d)} \leqslant \varepsilon. \tag{4.7}$$

(1) We assume the following discrimination property on σ: for any probability measure μ on $[0; 1]^d$,

$$\left(\forall y \in \mathbb{R}^d\,, \forall\theta \in \mathbb{R}\,, \int_{[0;1]^d} \sigma\left(\langle y, x\rangle + \theta\right) d\mu(x) = 0\right) \Rightarrow \mu = 0.$$

Show that under this assumption σ has the universal approximation property. Is this an equivalence?

(2) We now let σ be such that

$$\sigma(x) \underset{x\to\infty}{\to} 0\,,\sigma(x) \underset{x\to\infty}{\to} 1.$$

We want to show that σ is discriminatory.

(a) Let μ be such that

$$\forall y \in \mathbb{R}^d\,, \forall \theta \in \mathbb{R}\,, \int_{[0;1]^d} \sigma\left(\langle y, x\rangle + \theta\right) d\mu(x) = 0.$$

Show that for any $y \in \mathbb{R}^d\,, \varphi \in \mathbb{R}\,, \theta \in \mathbb{R}$,

$$\mu\left(\{\langle y, \cdot\rangle > \theta\}\right) + \sigma(\varphi)\mu\left(\{\langle y, \cdot\rangle = \theta\}\right) = 0.$$

Hint: Consider, $y\,,\phi$ and θ being fixed, the behaviour of $\sigma(\lambda(\langle x, y\rangle + \theta) + \varphi)$ as $\lambda \to \infty$...

(b) Deduce that for all $y \in \mathbb{R}^d\,, \theta \in \mathbb{R}$,

$$\mu\left(\{\langle y, \cdot\rangle \geqslant \theta\}\right) = 0.$$

(c) Prove that this implies $\mu \equiv 0$.

(3) Give an example of a discriminatory σ.

4.2.2. **Around the stochastic gradient descent**

Exercise 4.4 (Interpolation and variance for finite-sum optimisation problems)**.** We consider a finite-sum function

$$f : x \mapsto \frac{1}{N}\sum_{i=1}^N f_i(x),$$

which is assumed to be coercive, and such that each of the f_i is bounded from below, convex, and has a μ-Lipschitz gradient.

(1) Interpolation for finite-sum problems We define

$$\delta_f := \min f - \frac{1}{N}\sum_{k=1}^N \inf_{\mathbb{R}^d} f_i.$$

(a) Show that $\delta_f \geqslant 0$, with equality if, and only if there exists $x^* \in \mathbb{R}^d$ such that for any $i \in \{1, \dots, N\}$, $f_i(x^*) = \min_{\mathbb{R}^d} f_i(x)$.

(b) We let X^* be the set of minimisers of f and we define, for any $x^* \in X^*$, $\sigma_f(x^*) := \mathbb{E}[\|\nabla f_{i(\cdot)}(x^*)\|^2]$. Show that for any $x\,, y \in X^*$, $\sigma_f(x) = \sigma_f(y)$. We call this common value σ_f.

(c) Show that for any $x \in \mathbb{R}^d$ there holds

$$\mathbb{E}[\|\nabla f_{i(\cdot)}(x)\|^2] \leqslant 4\mu\left(f(x) - \min f\right) + 2\sigma_f.$$

(2) Variance for mini-batches We assume that f is strictly convex and we let $n_B \in \{1, \dots, N\}$ be a mini-batch size. Show that

$$\sigma_{f,n_B} = \frac{N - n_b}{n_B(N-1)}\sigma_f.$$

Exercise 4.5 (Convergence results for the Stochastic Gradient Descent)**.** We explore in this exercise some further aspects of the stochastic gradient descent that we did not touch upon in the lecture notes.

(1) **Polyak-Lojasiewicz functions** We assume that a function f writes $f : x \mapsto \frac{1}{N}\sum_{i=1}^N f_i(x)$ with each ∇f_i μ-Lipschitz. Furthermore, we assume that f satisfies a Polyak-Lojasiewicz inequality with constant α:

$$\forall f \in \mathbb{R}^d \,, f(x) - \inf_{\mathbb{R}^d} f \leqslant \frac{\|\nabla f(x)\|^2}{2\alpha}.$$

Finally, we consider the sequence generated by the stochastic gradient descent. Show that, for any step size τ small enough, there holds

$$\mathbb{E}[f(x_k) - \inf_{\mathbb{R}^d} f] \leqslant (1-\tau\alpha)^k (f(x_0) - \inf_{\mathbb{R}^d} f) + \frac{\tau\mu^2}{\alpha}\delta_f$$

where $\delta_f = \inf_{\mathbb{R}^d} f - \frac{1}{N}\sum_{i=1}^N \inf_{\mathbb{R}^d} f_i$. You can assume for the sake of simplicity that f has a minimiser x^* and start by proving that

$$\forall x \in \mathbb{R}^d \,, \mathbb{E}[\|\nabla f_{i(\cdot)}(x)\|^2] \leqslant 2\mu\left(f(x) - f(x^*)\right) + 2\mu\delta_f.$$

(2) **The case of smooth functions** Consider a function f that writes $f = \frac{1}{N}\sum_{i=1}^N f_i$, where each of the f_i is assumed to have a μ-Lipschitz gradient, fix an integer K and the step size $\tau_K := \sqrt{\frac{2}{K\mu^2}}$. We let $\{x_k\}_{k\in\mathbb{N}}$ be a sequence generated by the stochastic gradient descent with step size K.

(a) Show that, for any k,

$$\tau_K \mathbb{E}[\|\nabla f_{i(\cdot)}(x_k)\|^2] \leqslant (1+\tau_K^2\mu^2)\mathbb{E}[f(x_k) - \inf_{\mathbb{R}^d} f] - \mathbb{E}[f(x_{k+1}) - \inf_{\mathbb{R}^d} f] + \tau_K^2\mu^2\delta_f.$$

(b) Consider the sequence

$$\alpha_k := (1+\tau_K^2\mu^2)^{-1-k}$$

and show that

$$\tau_K \sum_{k=0}^{K-1} \alpha_k \mathbb{E}[\|\nabla f(x_k)\|^2] \leqslant (f(x_0) - \inf_{\mathbb{R}^d} f) + \tau_K^2\mu^2\delta_f \sum_{k=0}^{K-1}\alpha_k.$$

(c) Deduce that

$$\min_{k\in\{0,\dots,K-1\}} \mathbb{E}[\|\nabla f(x_k)\|^2] \leqslant \sqrt{\frac{2\mu^2}{K}}\left(2(f(x_0) - \inf_{\mathbb{R}^d} f) + \delta_f\right).$$

4.3. **Solution of the exercises**

Solution of Exercise 4.1. (1) We let $x \in \mathrm{Conv}(C)$ so that we can find $\{x_i\}_{i=1,\dots,k} \in C^k \,, \{\alpha_i\}_{i=1,\dots,k} \in [0;1]^k \,, \sum_{i=1}^k \alpha_i = 1$ and

$$x = \sum_{i=1}^k \alpha_i x_i.$$

Either $k \leqslant d+1$, in which case there is nothing to prove, or $k > 1+d$. In that case, the family $\{x_i - x_1\}_{i=1,\dots,k}$ is linearly dependent, whence there exist $\{\mu_1,\dots,\mu_k\} \in \mathbb{R}^k \setminus \{(0,\dots,0)\}$ such that

$$\sum_{i=1}^k \mu_i(x_i - x_1) = 0.$$

Setting

$$\lambda_1 = -\sum_{i=2}^{k} \mu_i \,, \lambda_i = \mu_i$$

we obtain

$$\sum \lambda_i = 0 \,, \sum \lambda_i x_i = 0.$$

Set

$$\gamma := \min\{\frac{\alpha_i}{\lambda_i} \,, \lambda_i > 0\} \,, \beta_i = \alpha_i - \gamma\lambda_i.$$

The family $\{\beta_i\}_{i=1,\dots,d}$ is still an admissible linear weights. Furthermore, letting j be such that $\gamma = \frac{\alpha_j}{\lambda_j}$, $\beta_j = 0$ and finally

$$x = \sum_{i=1}^{k} \alpha_i x_i = \sum_{i=1}^{k} \beta_i x_i,$$

and we have thus expressed it as a combination of $k-1$ elements. Iterating provides the conclusion.

(2) The implication $(b) \Rightarrow (a)$ is the Hahn-Banach theorem. Now, assume that we can find $x \in \mathrm{Conv}(f^-{-1}(\{+\})) \cap \mathrm{Conv}(f^{-1}(\{-\}))$ where f is a linear separation of the dataset X, represented, say, by some vector w. Then it implies that we can write $x = \sum \alpha_i x_{i,+} = \sum \beta_i x_{i,-}$ with $x_{i,\pm} \in f^{-1}(\{\pm\})$. The contradiction follows by taking the scalar product with w.

(3) (a) If the set A is finite then, setting $J := 1 + \max(A)$, if follows that the sequence $\{w_j\}_{j\in\mathbb{N}}$ is stationary as off the rank J, which thus provides the conclusion.

(b) When $j = 0$, there is nothing to prove. Assume the inequalities are satisfied at a certain rank $j-1$ Two possibilities might occur:

- Either $f(x_j)\langle w_{j-1}, x_j\rangle > 0$, in which case $n_j = n_{j-1}$ and $w_j = w_{j-1}$. There is nothing to prove.
- Or $f(x_j)\langle w_{j-1}, x_j\rangle \leqslant 0$, in which case $n_j = n_{j-1} + 1$ and $w_j = w_{j-1} + \tau f(x_j)x_j$. Thus,

$$\tau\delta n_j\|w\| = \tau\delta n_{j-1}\|w\| + \tau\delta\|w\| \leqslant \langle w, w_{j-1}\rangle + \tau f(x_j)\langle w, x_j\rangle = \langle w, w_j\rangle.$$

Similarly

$$\begin{aligned}\|w_j\|^2 &= \|w_{j-1}\|^2 + 2\tau f(x_j)\langle w_{j-1}, x_j\rangle + \tau^2\|x_j\|^2\\ &\leqslant \|w_{j-1}\|^2 + \tau^2\|x_j\|^2\\ &\leqslant \|w_{j-1}\|^2 + \tau^2 R^2.\end{aligned}$$

This suffices to conclude.

(c) Using the Cauchy-Schwarz inequality and arguing by contradiction, this implies the existence of two constants $A \,, B$ such that

$$n_j \leqslant A + B\sqrt{n_j}$$

which entails that $\{n_j\}_{j\in\mathbb{N}}$ is bounded. As the sequence is non-decreasing, it is eventually stationary, which provides the conclusion.

Solution of Exercise 4.2. (1) This is simply because of the density of piecewise affine function.

(2) The function f is linear. It suffices to define $N = m + 1$ and

$$a_N = 0 \,, a_1, \dots, a_{N-1} = 1 \,, b_N = 1 \,, b_i = -x_i \,, c_1 := \alpha_1 \,, c_i = \alpha_i - \alpha_{i-1}.$$

Solution of Exercise 4.3. (1) We let X be the set

$$X := \left\{\sum_{k=1}^N \alpha_k \sigma\left(\langle y_k, \cdot\rangle + \theta_k\right)\,, N \in \mathbb{N}\,, \alpha\,, \theta \in \mathbb{R}^N\,, y \in (\mathbb{R}^d)^N\right\}$$

and we show that X is dense in $\mathscr{C}^0([0;1]^d)$. Should this not be the case, by the Hahn-Banach theorem, there exists a linear form $\ell \in \left(\mathscr{C}^0([0;1]^d)\right)'$ such that that $\ell \not\equiv 0$ on $\mathscr{C}^0([0;1]^d)$, but such that $\ell \equiv 0$ on X. By the Riesz representation theorem, there exists a measure μ on $[0;1]^d$ such that

$$\ell(f) = \int f d\mu.$$

As σ is discriminatory, this implies $\mu \equiv 0$, a contradiction. This is clearly an equivalence, for the non-discriminatory character of σ also implies that X thus defined is not dense.

(2) (a) We follow the hint and we observe that

$$\sigma(\lambda(\langle x, y\rangle + \theta) + \varphi) \underset{\lambda\to\infty}{\to} \begin{cases} 0 \text{ if } \langle x, y\rangle + \theta < 0\,, \\ \sigma(\varphi) \text{ if } \langle x, y\rangle + \theta = 0\,, \\ 1 \text{ if } \langle x, y\rangle + \theta > 0. \end{cases}$$

The conclusion follows by passing to the limit.

(b) Take the limit $\phi \to +\infty$.

(c) This is the Fourier transform, along with the fact that $\hat{\mu} \equiv 0 \Rightarrow \mu \equiv 0$, that gives the result: indeed, from the condition of question (b), we deduce that for any continuous function h,

$$\forall x \in \mathbb{R}^d\,, \int_{\mathbb{R}^d} h(\langle x, y\rangle) d\mu(y) = 0,$$

which entails that the Fourier transform of μ is zero.

(3) Trivial.

Solution of Exercise 4.4. (1) (a) We let x^* denote the minimiser of f. Then

$$\delta_f = \frac{1}{N}\sum_{i=1}^N \left(f_i(x^*) - \inf_{\mathbb{R}^d} f_i\right) \geqslant 0$$

with equality if, and only if, for any i, $f_i(x^*) = \inf_{\mathbb{R}^d} f_i$.

(b) We let $x\,, y \in X^*$. From the estimate (4.3), we know that for any $i \in \{1, \dots, N\}$,

$$\frac{1}{2\mu}\|\nabla f_i(x) - \nabla f_i(y)\|^2 \leqslant -\langle \nabla f_i(x), y - x\rangle.$$

In particular,

$$\mathbb{E}\left[\|\nabla f_i(x) - \nabla f_i(y)\|^2\right] \leqslant 2\mu\langle \mathbb{E}\left[\nabla f_i(x)\right], x - y\rangle = 0.$$

Thus,

$$\mathbb{E}\left[\|\nabla f_i(x)\|^2\right] = \mathbb{E}\left[\|\nabla f_i(y)\|^2\right]$$

and the conclusion follows.

(c) This follows from the same reasoning as in the lecture notes.

(2) This is a straightforward computation: let x^* be the unique minimiser of f. Then

$$\begin{aligned}
\sigma_{f,n_B} &= \frac{1}{\binom{N}{n_B}} \sum_{B\subset\{1,\dots,N\}\,,|B|=n_B} \left\| \sum_{i\in B} \nabla f_i(x^*) \right\|^2 \\
&= \frac{1}{\binom{N}{n_B}} \sum_{B\subset\{1,\dots,N\}\,,|B|=n_B} \sum_{i,j\in B} \langle \nabla f_i(x^*)\,, \nabla f_j(x^*)\rangle \\
&= \frac{1}{\binom{N}{n_B}} \sum_{B\subset\{1,\dots,N\}\,,|B|=n_B} \left(\sum_{i\in B} \|\nabla f_i(x^*)\|^2 + \sum_{i\neq j\,,i\,,j\in B} \langle \nabla f_i(x^*), \nabla f_j(x^*)\rangle \right) \\
&= \frac{1}{\binom{N}{n_B}} \sum_{B\subset\{1,\dots,N\}\,,|B|=n_B} \sum_{i\in B} \|\nabla f_i(x^*)\|^2 \\
&+ \frac{1}{\binom{N}{n_B}} \sum_{i\neq j} \sum_{B\,,|B|=n_B\,,(i,j)\in B} \langle \nabla f_i(x^*)\,, \nabla f_j(x^*)\rangle \\
&= \frac{\binom{N-1}{n_B-1}}{\binom{N}{n_B}} N\sigma_f + \frac{\binom{N-2}{n_B-2}}{\binom{N}{n_B}} \sum_{i\neq j} \langle \nabla f_i(x^*), \nabla f_j(x^*)\rangle \\
&= \frac{\binom{N-1}{n_B-1}}{\binom{N}{n_B}} N\sigma_f + \frac{\binom{N-2}{n_B-2}}{\binom{N}{n_B}} \left(\sum_{i=1}^N \langle \nabla f_i(x^*)\,, \nabla f_j(x^*)\rangle - \sum_i \|\nabla f_i(x^*)\|^2 \right) \\
&= \left(\frac{\binom{N-1}{n_B-1}}{\binom{N}{n_B}} - \frac{\binom{N-2}{n_B-2}}{\binom{N}{n_B}} \right) N\sigma_f \\
&= \frac{N-n_B}{n_B(N-1)}\sigma_f.
\end{aligned}$$

Solution of Exercise 4.5. (1) We begin by proving the estimate. Recall that, as each of the f_i's has a μ-Lipschitz gradient, we have

$$\forall i \in \{1,\dots,N\}\,, \forall x \in \mathbb{R}^d\,, f_i(x) - \inf_{\mathbb{R}^d} f_i \geqslant \frac{1}{2\mu}\|\nabla f_i(x)\|^2.$$

In particular, we obtain

$$\|\nabla f_i(x)\|^2 \leqslant 2\mu(f_i(x) - \inf_{\mathbb{R}^d} f_i) = 2\mu(f_i(x) - f_i(x^*)) + 2\mu(f_i(x^*) - \inf_{\mathbb{R}^d} f_i).$$

Passing to the expectation provides the conclusion, and we deduce

$$\forall x \in \mathbb{R}^d\,, \mathbb{E}[\|\nabla f_{i(\cdot)}(x)\|^2] \leqslant 2\mu\,(f(x) - f(x^*)) + 2\mu\delta_f. \tag{4.8}$$

We then argue exactly as in the lecture notes. Indeed, observe that by the same reasoning as in the proof of Theorem 4.1, for any $k \in \mathbb{N}$, we have

$$f(x_{k+1}) \leqslant f(x_k) - 2\tau\langle \nabla f(x_k), \nabla f_{i(\cdot)}(x_k)\rangle + \frac{\tau^2\mu}{2}\|\nabla f_{i(\cdot)}(x_k)\|^2$$

whence, passing to the expectation, we deduce

$$\begin{aligned}\mathbb{E}[f(x_{k+1})] &\leqslant f(x_k) - 2\tau\|\nabla f(x_k)\|^2 + \frac{\tau^2\mu}{2}\mathbb{E}[\|\nabla f_{i(\cdot)}(x_k)\|^2] \\ &\leqslant f(x_k) - \tau\alpha\left(f(x_k) - f(x^*)\right) + \frac{\tau^2\mu}{2}\mathbb{E}[\|\nabla f_{i(\cdot)}(x_k)\|^2]\text{by the Polyak-Lojasiewicz inequality} \\ &\leqslant f(x_k) - \tau\alpha\left(f(x_k) - f(x^*)\right) + \frac{\tau^2\mu}{2}\left(2\mu(f(x_k) - f(x^*)) + 2\mu\delta_f\right) \text{ by (4.8).}\end{aligned}$$

Thus,

$$\mathbb{E}[f(x_{k+1}) - f(x^*)] \leqslant (1 - \tau\alpha + 2\tau^2\mu^2)\mathbb{E}[f(x_k) - f(x^*)] + \tau^2\mu^2\delta_f.$$

Iterating, we deduce that

$$\mathbb{E}[f(x_{k+1}) - f(x^*)] \leqslant (1 - \tau\alpha + 2\tau^2\mu^2)(f(x_0) - f(x^*)) + \tau^2\mu^2\sum_{i=0}^{k}(1 - \tau\alpha + 2\tau^2\mu^2)^i,$$

whence the conclusion.

(2) (a) These are exactly the same computations as before.
(b) It suffices to multiply the previous inequality by α_k and to recognise a telescopic sum.
(c) It suffices to study the inequality

$$\tau_K \min_{i=0,\dots,K-1} \mathbb{E}[\|\nabla f(x_i)\|^2] \sum_{j=0}^{K-1} \alpha_j \leqslant f(x_0) - \inf_{\mathbb{R}^d} f) + \tau_K^2\mu^2\delta_f \sum_{k=0}^{K-1} \alpha_k$$

to conclude.

Appendix A. Exams

A.1. **Midterm (2024-2025)**

Exercise A.1 (A bit of Newton method)**.**

1. Let $A \in S_d^{++}(\mathbb{R})$ and $b \in \mathbb{R}^d$; write the iterations for the Newton method applied to the minimisation of the function
$$f : x \mapsto \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$
2. Propose an example of a function that is strongly convex but for which, depending on the initialisation, the Newton method does not converge.

Solution of Exercise A.1.

1. See the lecture notes.
2. Idem.

Exercise A.2 (The Newton method for the computation of eigenvalues)**.** Let $A \in M_d(\mathbb{R})$ and let $\lambda \in \mathbb{R}$ be the smallest eigenvalue of A. We assume λ is a *simple* eigenvalue of A. We let x_λ be an associated eigenvector, with $\|x\| = 1$. We want to approximate (λ, x_λ) numerically. To this end, we introduce the function
$$\Phi : (x, \xi) \mapsto (Ax - \xi x, \|x\|^2 - 1).$$

1. Prove that
$$\lambda = \min_{y\,, \|y\|=1} \frac{\langle Ay, y\rangle}{\|y\|^2}.$$
2. Show that the Jacobian J_Φ of Φ at (x_λ, λ) is invertible (Hint: to check a linear map is invertible, it suffices to check it is injective... Also, what does the Courant-Fischer formula give you?).
3. Which rate of convergence should we expect for the Newton method? Which operations should be carried out and how would you carry them out numerically?

Solution of Exercise A.2.

1. See exercise sheet 1.
2. The Jacobian J_Φ at (x, ξ) is given by
$$J_\Phi(x, \xi) : (h, \mu) \mapsto (Ah - \xi h - \mu x, 2\langle x, h\rangle).$$
To check its invertibility, it suffices to check that it is injective. In other words, let us show that $J_\Phi(x_\lambda, \lambda)(h, \mu) = (0, 0)$ implies $h = 0\,, \mu = 0$. On the one hand, we have
$$Ah - \lambda h = \mu x_\lambda$$
and, on the other,
$$\langle x_\lambda, h\rangle = 0.$$
This implies (taking the scalar product of the first equation with x_λ)
$$\langle Ah, h\rangle = \lambda \|h\|^2.$$
As
$$\lambda = \min_{y\,, \|y\|=1} \frac{\langle Ay, y\rangle}{\|y\|^2}$$
this gives that h is an eigenvector associated with λ. However, as λ is a simple eigenvalue and as x_λ and h are orthogonal, this gives a contradiction.
3. We can expect a quadratic convergence rate as the Jacobian is invertible. However, at each step, we must solve a linear system. There are several possibilities to do so, typically a Gauss elimination procedure.

Exercise A.3. Let $f \in \mathscr{C}^2(\mathbb{R}^d;\mathbb{R})$ be coercive, assume that ∇f is μ-Lipschitz and that f has a unique critical point.

(1) Show that for any $x_0 \in \mathbb{R}^d$ and any $\tau > 0$ small enough, the sequence of iterates generated by constant step-size gradient descent remains in a compact subset of $\mathbb{R}^d$.

(2) Show that for any $x_0 \in \mathbb{R}^d$ and for such a τ, the sequence generated by gradient descent with constant step-size τ converges (you must identify the limit point).

Solution of Exercise A.3. (1) Any $\tau < \frac{1}{\mu}$ would fit.

(2) Here you can invoke the Zoutendijk theorem: any closure point is critical, and since there is a unique critical point, there is a unique closure point for the sequence. As the sequence remains in a compact set, this is sufficient to guarantee convergence to that critical point.

Exercise A.4 (Mirror gradient descent)**.** We study a variant of gradient descent based on the notion of Bregman divergence: For any strictly convex, $\mathscr{C}^1$ function $\varphi : \Omega \to \mathbb{R}$ (where Ω is a convex subset of $\mathbb{R}^d$), we define its Bregman divergence as

$$d_\varphi : \Omega \times \Omega \ni (x,y) \mapsto \varphi(x) - \varphi(y) - \langle \nabla\varphi(y), x-y\rangle.$$

(1) Preliminary

(a) Compute the Bregman divergence of $\varphi : x \mapsto \|x\|^2$ (here $\Omega = \mathbb{R}^d$) and of $\varphi : x \mapsto \sum_{i=1}^d x_i \ln(x_i)$ (here $\Omega = \{x \in [0;1]^d, \sum_{i=1}^d x_i = 1\}$. In the last case, show that the Bregman divergence is given by $(x,y) \mapsto \sum_{i=1}^d x_i \ln(x_i/y_i)$. Is the Bregman divergence symmetric in general?

(b) Show that if φ is strictly convex, then for any y, $x \mapsto d_\varphi(x,y)$ is strictly convex.

(c) Show that if φ is strictly convex then $d_\varphi(x,y) \geqslant 0$ with equality if, and only if, $x = y$.

(d) Consider a $\mathscr{C}^1$ function $f : \mathbb{R}^d \to \mathbb{R}$ with μ-Lipschitz gradient. Let $\tau \geqslant 0$, $x_0 \in \mathbb{R}^d$ and $\{x_k\}_{k\in\mathbb{N}}$ the sequence generated by gradient descent with constant step size $\tau > 0$. Show that for any $k \geqslant 0$ x_{k+1} is the minimiser of

$$y \mapsto \langle \nabla f(x_k), y - x_k\rangle + \frac{1}{2\tau}\|y - x_k\|^2.$$

You must show that this map admits a minimiser beforehand.

We now define the gradient descent with step-size $\tau > 0$, with respect to a function φ, as: given an initialisation $x_0 \in \mathbb{R}^d$, for any $k \in \mathbb{N}$, x_{k+1} is chosen as the minimiser of the map

$$y \mapsto \langle \nabla f(x_k), y - x_k\rangle + \frac{1}{2\tau} d_\varphi(x, x_k).$$

(2) Study of the Bregman descent

(a) Let ℓ be a $\mathscr{C}^1$, convex function bounded from below, $x \in \mathbb{R}^d$. Assume that φ is coercive and strictly convex. Let y^* be defined as the solution of

$$\min_{y\in\mathbb{R}^d} \ell(y) + \frac{1}{2\tau} d_\varphi(y,x).$$

Show that for any $y \in \mathbb{R}^d$,

$$\ell(y) + \frac{1}{2\tau} d_\varphi(y,x) \geqslant \ell(y^*) + \frac{1}{2\tau} d_\varphi(y^*,x) + \frac{1}{2\tau} d_\varphi(y,y^*).$$

Hint: you can start by showing that for any $(a,b,c) \in (\mathbb{R}^d)^3$ there holds

$$d_\varphi(c,b) - \langle \nabla\varphi(a) - \nabla\varphi(b), c-b\rangle = d_\varphi(c,a) - d_\varphi(b,a).$$

This procedure, applied iteratively to ℓ with a step size $\tau > 0$, is called the Bregman descent.

We say that ℓ is L-smooth with respect to φ if $L\varphi - \ell$ is convex.

(b) Assume that f is L-smooth with respect to φ, convex, coercive and $\mathscr{C}^1$. Let $\{x_k\}_{k\in\mathbb{N}}$ be the sequence generated by Bregman descent with a constant step-size $\tau > 0$, defined as

$$x_{k+1} = \operatorname{argmin}\left(f(x_k) + \langle \nabla f(x_k), y - x_k\rangle + \frac{1}{2\tau} d_\varphi(y,x_k)\right). \tag{A.1}$$

We want to show that if $\tau \in (0;1/2L)$, the sequence $\{x_k\}_{k\in\mathbb{N}}$ satisfies

$$f(x_k) - \min_{\mathbb{R}^d} f \leqslant \frac{1}{k\tau} d_\varphi(x^*, x_0)$$

where x^* is a minimiser of f (you can assume it to be unique).

(i) First, show that for any x, y,

$$f(y) \leqslant f(x) + \langle \nabla f(x), y - x\rangle + L d_\varphi(y,x).$$

(ii) Write down the optimality conditions for the optimisation problem (A.1).

(iii) Using the strict convexity of φ, show that $\{f(x_k)\}_{k\in\mathbb{N}}$ is decreasing unless it is stationary.

(iv) Prove that

$$f(x_{k+1}) - f(x^*) \leqslant \langle \nabla f(x_k), x_{k+1} - x^*\rangle + L d_\varphi(x_{k+1}, x_k).$$

(v) Prove that

$$\langle \nabla f(x_k), x_{k+1} - x^*\rangle \leqslant \frac{1}{2\tau}\left(d_\varphi(x^*, x_k) - d_\varphi(x^*, x_{k+1}) - d_\varphi(x_{k+1}, x_k)\right).$$

(vi) Conclude.

(c) Assume that, besides the assumptions of the previous question, f is α-strongly convex relative to φ in the sense that $f - \alpha\varphi$ is convex. Show, with the same assumptions on the step-size $\tau > 0$, that

$$d_\varphi(x^*, x_k) \leqslant (1-\alpha\tau)^k d_\varphi(x^*, x_0).$$

Solution of Exercise A.4. (1) (a) For $\varphi_0 : x \mapsto \|x\|^2$, we obtain

$$d_{\varphi_0}(x,y) = \|x\| - \|y\|^2 - 2\langle y, x-y\rangle = \|x-y\|^2.$$

For $\varphi_1 : x \mapsto \sum_{i=1}^d x_i \ln(x_i)$,

$$\begin{aligned} d_{\varphi_1}(x,y) &= \sum_i x_i \ln(x_i) - \sum_i y_i \ln(y_i) - \langle \begin{pmatrix} 1+\ln(y_1) \\ \vdots \\ 1+\ln(y_d)\end{pmatrix}, x-y\rangle \\ &= \sum_{i=1}^d x_i \ln\left(\frac{x_i}{y_i}\right). \end{aligned}$$

(b) A sum of a linear function and of a strictly convex function is strictly convex.
(c) Definition of the strict convexity.
(d) Introduce the function

$$g : y \mapsto \langle \nabla f(x_k), y - x_k \rangle + \frac{1}{2\tau} ||y - x_k||^2.$$

The coercivity and the strong convexity of g are immediate, which gives the existence and uniqueness of a minimiser, characterised by

$$\nabla g(y^*) = 0 \Leftrightarrow \nabla f(x_k) + \frac{1}{\tau}(y^* - x_k) = 0.$$

This is exactly the definition of the sequence generated by gradient descent.

(2) (a) We begin with the hint. Letting $a\,,b\,,c$ be three points in $\mathbb{R}^d$, we have

$$\begin{aligned} d_\varphi(c,b) - \langle \nabla\varphi(a) - \nabla\varphi(b), c - b\rangle &= \varphi(c) - \varphi(b) - \langle \nabla\varphi(b), c-b\rangle - \langle \nabla\varphi(a), c-b\rangle + \langle \nabla\varphi(b), c-b\rangle \\ &= \varphi(c) - \varphi(b) - \langle \nabla\varphi(a), c-a\rangle - \langle \nabla\varphi(a), a-b\rangle \\ &= \varphi(c) - \varphi(a) - \langle \nabla\varphi(a), c-a\rangle \\ &- (\varphi(b) - \varphi(a) - \langle \nabla\varphi(a), b-a\rangle \\ &= d_\varphi(c,a) - d_\varphi(b,a). \end{aligned}$$

Moving back to the optimisation problem, we obtain, at an optimiser y^*,

$$\nabla \ell(y^*) = \frac{\nabla\varphi(x) - \nabla\varphi(y^*)}{2\tau}.$$

Consequently, as ℓ is convex,

$$\begin{aligned} \ell(y) &\geqslant \ell(y^*) + \langle \nabla\ell(y^*), y - y^*\rangle \\ &= \ell(y^*) + \frac{1}{2\tau}\langle \nabla\varphi(x) - \nabla\varphi(y^*), y - y^*\rangle \\ &= \ell(y^*) + \frac{1}{2\tau}\left(d_\varphi(y,y^*) + d_\varphi(y^*,x) - d_\varphi(y,x)\right). \end{aligned}$$

This gives the required conclusion.

(b) (i) This is simply a convexity inequality.
(ii) The optimality conditions write

$$\nabla f(x_k) + \frac{1}{2\tau}\left(\nabla\varphi(x_{k+1} - \nabla\varphi(x_k)\right) = 0.$$

(iii) We have

$$\begin{aligned} f(x_{k+1}) &\leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + L d_\varphi(x_{k+1}, x_k) \\ &\leqslant f(x_k) + \frac{1}{2\tau}\langle \nabla\varphi(x_k) - \nabla\varphi(x_{k+1}), x_{k+1} - x_k\rangle + L d_\varphi(x_{k+1}, x_k) \\ &\leqslant f(x_k) + \frac{1}{2\tau}\langle \nabla\varphi(x_k) - \nabla\varphi(x_{k+1}), x_{k+1} - x_k\rangle + L\left(\varphi(x_{k+1}) - \varphi(x_k) - \langle \nabla\varphi(x_k), x_{k+1} - x_k\rangle\right) \\ &\leqslant f(x_k) + \frac{1}{2\tau}\langle \nabla\varphi(x_k) - \nabla\varphi(x_{k+1}), x_{k+1} - x_k\rangle + L\left(\langle \nabla\varphi(x_{k+1}) - \nabla\varphi(x_k), x_{k+1} - x_k\rangle\right), \end{aligned}$$

and the last inequality is strict unless $x_k = x_{k+1}$. Taking $\tau < 1/2L$ yields the conclusion.

(iv) It suffices to use the convexity of f:

$$\begin{aligned}f(x_{k+1})-f(x^*)&\leqslant f(x_k)-f(x^*)+\langle\nabla f(x_k),x_{k+1}-x_k\rangle+Ld_\varphi(x_{k+1},x_k)\\&\leqslant\langle\nabla f(x_k),x_k-x^*\rangle+\langle\nabla f(x_k),x_{k+1}-x_k\rangle+Ld_\varphi(x_{k+1},x_k)\\&=\langle\nabla f(x_k),x_{k+1}-x^*\rangle+Ld_\varphi(x_{k+1},x_k)\end{aligned}$$

(v) It suffices to apply Question (a).

(vi) We obtain

$$f(x_{k+1})-f(x_k)\leqslant\frac{1}{2\tau}\left(d_\varphi(x^*,x_k)-d_\varphi(x^*,x_{k+1})\right).$$

We conclude as in the lectures.

(c) Observe that in that case

$$\alpha d_\varphi\leqslant d_f$$

so that, writing

$$f(x)-f(y)=\langle\nabla f(x),x-y\rangle-d_f(y,x)$$

the same steps provide

$$\begin{aligned}f(x_{k+1})-f(x^*)&\leqslant f(x_k)-f(x^*)+\langle\nabla f(x_k),x_{k+1}-x_k\rangle+Ld_\varphi(x_{k+1},x_k)\\&=\langle\nabla f(x_k),x_k-x^*\rangle-d_f(x^*,x_k)+\langle\nabla f(x_k),x_{k+1}-x_k\rangle+Ld_\varphi(x_{k+1},x_k)\end{aligned}$$

and we conclude as in the lectures.