Optimisation numérique et science des données

Emmanuel Trélat[1]

1. Sorbonne Université, CNRS, Université de Paris, Inria, Laboratoire Jacques-Louis Lions (LJLL), F-75005 Paris, France (emmanuel.trelat@sorbonne-universite.fr).

Table des matières

Résumé. Ce cours permet d'acquérir les outils mathématiques théoriques et pratiques de pointe en optimisation numérique et science des données. L'objectif est d'apprendre à modéliser et résoudre des problèmes complexes d'optimisation, avec ou sans contraintes, et d'apprendre à mettre en oeuvre divers algorithmes innovants efficaces pour l'approximation numérique des solutions. Dans ce cours, on apprendra les méthodes classiques d'optimisation : existence, conditions de premier et de second ordre, diverses variantes de méthodes de gradient, conditions de Karush-Kuhn-Tucker, dualité Lagrangienne, puis on fera une ouverture à la science des données : gradient stochastique, gradient coordonnée par coordonnée, gradient non lisse, TensorFlow. Des TD et TP (en Python) viendront compléter la formation, ainsi qu'une introduction aux méthodes les plus à la pointe : différentiation automatique (AMPL) couplée aux outils experts (IpOpt). Elles seront illustrées sur divers exemples, comme l'analyse d'images ou le machine learning.

Motivation. Les algorithmes de machine learning et deep learning sont basés sur des techniques d'optimisation. Les problèmes sous-jacents sont souvent posés en très grande dimension, ce qui rend leur résolution numérique difficile. En grande dimension, les approximations convexes sont particulièrement utiles, car le problème associé a une solution unique et les algorithmes sont plus efficaces. Mais en toute généralité, les problèmes sont non convexes et de plus comportent des contraintes d'égalité ou d'inégalité.

Dans les techniques d'apprentissage, les problèmes de type "moindres carrés" sont d'une grande importance. Ce sont des problèmes de minimisation convexe

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_2^2$$

où $A \in \mathbb{R}^{p\times n}$, $b \in \mathbb{R}^p$, qui sont souvent posés avec n et/ou p grands. On rencontre beaucoup les variantes

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_2^2 + \alpha\|x\|_2^2$$

où on pénalise avec une régularisation par la norme euclidienne, et

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_2^2 + \alpha\|x\|_1$$

où on pénalise avec la norme ℓ^1, cette dernière ayant l'avantage de promouvoir les solutions *parcimonieuses* ("sparse" en anglais), i.e., la solution x n'a qu'un nombre minimal de composantes actives. Cette dernière est beaucoup exploitée dans le domaine de l'analyse d'image, depuis les années 2000 où on a découvert son efficacité et su l'exploiter par des algorithmes d'analyse convexe et non lisse. Des variantes plus élaborées des moindres carrés comportent aussi des contraintes.

De manière générale, les problèmes de minimisation étudiés dans ce cours s'écrivent

$$\boxed{\min_{x\in C} f(x)} \tag{1}$$

où $f : E \to \mathbb{R}$ est une fonction sur un espace vectoriel E topologique (on la prendra souvent de classe C^1 ou C^2 par la suite, avec E Banach ou Hilbert, mais la fonction pourrait être peu régulière) et $C \subset E$ est un sous-ensemble de E représentant des contraintes. On parle de minimisation sans contrainte si $C = E$, et de minimisation sous contraintes lorsque $C \subsetneq E$.

On dit que $x^* \in C$ est un minimiseur global de f sur C si

$$f(x^*) = \min_{x\in C} f(x).$$

On note aussi

$$x^* = \operatorname*{argmin}_{x\in C} f(x).$$

On dit que x^* est un minimiseur local de f s'il existe un voisinage ouvert U de x^* dans E tel que

$$f(x^*) = \min_{x \in C \cap U} f(x)$$

autrement dit, x^* est un minimiseur de la fonction f restreinte à U, sous contraintes C.

Bien entendu, un minimiseur global est aussi un minimiseur local, mais pas réciproquement.

Dans ce cours, on donnera des théorèmes d'existence de minimiseur local ou global, qui se démontrent par des raisonnements de compacité. L'unicité, lorsqu'elle peut être obtenue, l'est souvent sous des conditions de convexité, ainsi que le caractère global du minimiseur.

Dans le cas sans contrainte, les conditions d'optimalité sont faciles à obtenir : le gradient de f doit être nul en un minimiseur (local). C'est une condition purement locale. On rappellera et on développera les algorithmes de type gradient (ou descente), en donnant diverses variantes dont certaines sont particulièrement adaptées aux problèmes de grande dimension.

Dans le cas avec contrainte, pour être capable d'obtenir des conditions nécessaires d'optimalité permettant de calculer le minimum (lorsqu'il existe), on suppose souvent que C est donné par des égalités et inégalités :

$$C = \{x \in \mathbb{R}^n \ \mid \ g_1(x) \leqslant 0, \dots, g_q(x) \leqslant 0, \quad h_1(x) = \cdots, h_p(x) = 0\}.$$

On donnera notamment les conditions de Karush-Kuhn-Tucker (KKT), qui généralisent le théorème des multiplicateurs de Lagrange (qui concerne les problèmes d'optimisation sous contrainte d'égalité seulement). Ces conditions conduisent à divers algorithmes de calcul dont on verra plusieurs variantes.

Toutes ces conditions, qui sont de type différentiel, sont locales et permettent donc au mieux de caractériser les minimiseurs locaux. Caractériser un minimiseur global est en général très difficile, à moins que la fonction soit convexe.

Il est intéressant de noter que le problème (1) d'optimisation sous contrainte s'exprime comme le problème d'optimisation sans contrainte

$$\min_{x \in \mathbb{R}^n} (f(x) + \chi_C(x))$$

où

$$\chi_C(x) = \begin{cases} 0 & \text{if } x \in C \\ +\infty & \text{sinon} \end{cases}$$

Cette remarque, formelle pour l'instant, conduit à la notions de pénalisation et aux importantes méthodes de dualité en optimisation.

Chapitre 1

Calcul différentiel

Ce chapitre contient des rappels en calcul différentiel. Le concept essentiel est la notion de différentielle (de Fréchet, de Gateaux) d'une fonction.

Bien que, en vue de développer des algorithmes de calcul, on s'intéresse prioritairement à des fonctions $f : \mathbb{R}^n \to \mathbb{R}$ (donc, sur un espace de dimension finie, éventuellement grande), autant que possible on s'attachera à donner des énoncés généraux, tant que cela ne complique pas les énoncés.

La notion la plus classique de différentielle, déjà vue dans des cours antérieurs, est la différentielle de Fréchet d'une fonction $f : E \to F$. Elle s'exprime dans des espaces de Banach. On rappelle que E est un espace (vectoriel) de Banach s'il est muni d'une norme $\| \ \|_E$ qui le rend complet.

La notion moins forte de différentielle de Gateaux nécessite simplement un espace vectoriel normé (pas forcément complet).

Ici, tous les espaces vectoriels qu'on va considérer sont réels. On rappelle qu'une norme $\| \ \|$ sur un espace vectoriel E est une application de E dans $[0, +\infty)$ qui est homogène ($\|\lambda x\| = |\lambda| \|x\|$ pour tout $\lambda \in \mathbb{R}$ et tout $x \in E$), sous-additive (inégalité triangulaire $\|x + y\| \leqslant \|x\| + \|y\|$ pour tous $x, y \in E$) et qui a la propriété de séparation (pour tout $x \in E$, $\|x\| = 0 \Rightarrow x = 0$).

Certaines notions nécessiteront d'avoir un produit scalaire. Or, une norme n'est pas nécessairement associée à un produit scalaire. Par exemple, dans $\mathbb{R}^n$, la norme $\| \ \|_2$ est la norme euclidienne, issue du produit scalaire euclidien. Mais les normes $\| \ \|_1$ et $\| \ \|_\infty$ ne sont pas des normes associées à un produit scalaire.

Un espace de Hilbert H est un espace de Banach dont la norme découle d'un produit scalaire, par la formule $\|x\| = \sqrt{\langle x, x \rangle}$. On rappelle qu'un produit scalaire est une forme bilinéaire symétrique et définie positive. Pour tous $x, y \in H$, on a

- $\|x + y\|^2 = \|x\|^2 + 2\langle x, y \rangle + \|y\|^2$ (théorème de Pythagore) ;
- $|\langle x, y \rangle| \leqslant \|x\| \|y\|$ (inégalité de Cauchy-Schwarz).

1.1 Différentielle

Différentielle de Fréchet. Soient E et F sont des espaces de Banach. Soit U un ouvert de E, soit $x \in U$, et soit $f : U \to F$ une application.

On dit que f est différentiable au sens de Fréchet en x s'il existe une application linéaire continue $df(x) : E \to F$ telle que, pour tout $h \in E$ tel que $x + h \in U$,

$$\boxed{f(x + h) = f(x) + df(x).h + \mathrm{o}(\|h\|_E)}$$

Autrement dit, f est Fréchet-différentiable en x si et seulement si f a un développement limité à l'ordre 1 en x.

L'application linéaire continue $df(x) : E \to F$ s'appelle la différentielle de Fréchet en x.

Notons que si f est Fréchet différentiable en x alors elle est continue en x.

Différentielle de Gateaux. Soient E et F sont des espaces vectoriels normés. Soit U un ouvert de E, soit $x \in U$, et soit $f : U \to F$ une application.

Soit $h \in E$. On dit que f a une dérivée directionnelle en x suivant h si la limite

$$\boxed{f'(x;h) = \lim_{\substack{t \to 0 \\ t \neq 0}} \frac{f(x+th) - f(x)}{t}}$$

existe. On dit que f est différentiable au sens de Gateaux en x si sa dérivée directionnelle en x existe selon toute direction et si l'application $h \mapsto f'(x;h)$ est linéaire continue.

Avec un abus de notation, on note aussi $df(x).h = f'(x;h)$.

Remarque 1. Si f est Fréchet différentiable en x alors f est Gateaux différentiable en x, et $f'(x,h) = df(x).h$. La réciproque est fausse : la fonction $f : \mathbb{R}^2 \to \mathbb{R}$ définie par $f(x,y) = 1$ si $y = x^2$, et 0 sinon, est Gateaux différentiable en $(0,0)$ (et $f'(x;h) = 0$ pour tout $h \in \mathbb{R}^2$) mais elle n'est pas Fréchet différentiable car elle n'est pas continue.

Une fonction f peut donc être Gateaux différentiable sans être continue. La dérivée au sens de Gateaux est une notion purement directionnelle, alors que la Fréchet différentiabilité est une dérivabilité pas seulement le long de toute direction, mais aussi le long de toute courbe notamment.

Cependant, la fonction $f : \mathbb{R}^2 \to \mathbb{R}$ définie par $f(x,y) = \frac{x^2}{|x|+y^2}$ si $(x,y) \neq (0,0)$, et $f(0,0) = 0$, a des dérivées directionnelles en $(0,0)$ dans toutes les directions, mais elle n'est pas Gateaux différentiable car l'application $h \mapsto f'((0,0),h)$ n'est pas linéaire.

Dans la suite, quand on parlera de différentielle sans préciser, cela sous-entend différentielle au sens de Fréchet.

Remarque 2. Pour $f : \mathbb{R} \to \mathbb{R}$, la dérivée usuelle est définie par

$$f'(x) = \lim_{\substack{h \to 0 \\ h \neq 0}} \frac{f(x+h) - f(x)}{h}$$

ce qui est équivalent au développement limité d'ordre 1 en x.

Remarque 3. Si $f : E \to F$ est linéaire, sa différentielle au sens de Fréchet est elle-même : $df(x) = f$, pour tout x.

Exemple 1. On note $\mathcal{M}_n(\mathbb{R})$ l'ensemble des matrices carrées d'ordre n, à coefficients réels, et $GL_n(\mathbb{R})$ l'ensemble des matrices inversibles.

- La différentielle au sens de Fréchet en $A \in \mathcal{M}_n(\mathbb{R})$ de l'application $f : \mathcal{M}_n(\mathbb{R}) \to \mathcal{M}_n(\mathbb{R})$ définie par $f(M) = M^2$ est $df(A).H = AH + HA$.
- La différentielle au sens de Fréchet en $A \in \mathcal{M}_n(\mathbb{R})$ de l'application $f : \mathcal{M}_n(\mathbb{R}) \to \mathcal{M}_n(\mathbb{R})$ définie par $f(M) = M^k$ est $df(A).H = A^{k-1}H + A^{k-2}HA + \cdots + HA^{k-1}$.
- La différentielle au sens de Fréchet en $A \in GL_n(\mathbb{R})$ de l'application $f : GL_n(\mathbb{R}) \to GL_n(\mathbb{R})$ définie par $f(M) = M^{-1}$ est $df(A).H = -A^{-1}HA^{-1}$.
- La différentielle au sens de Fréchet en $a \in \mathbb{R}^n$ de l'application $f : \mathbb{R}^n \to \mathbb{R}$ définie par $f(x) = \|x\|_2^2$ est $df(a).h = 2\langle a, h\rangle$.

- La différentielle au sens de Fréchet en $a \in \mathbb{R}^n \setminus \{0\}$ de la fonction $f : \mathbb{R}^n \to \mathbb{R}$ définie par $f(x) = \|x\|_2$ est $df(a).h = \langle \frac{a}{\|a\|_2}, h \rangle$ (produit scalaire euclidien). La fonction f n'est pas différentiable en 0.
- Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice symétrique. La différentielle au sens de Fréchet en $a \in \mathbb{R}^n$ de l'application $f : \mathbb{R}^n \to \mathbb{R}$ définie par $f(x) = x^\top Ax$ est $df(a).h = a^\top Ah + h^\top Aa = 2a^\top Ah$.

Dérivées partielles. Comme dit ci-dessus, si f est différentiable au sens de Fréchet en x (ou, seulement, Gateaux différentiable en x) et si h est un vecteur de E, la dérivée directionnelle de f dans la direction h est

$$f'(x;h) = df(x).h = \lim_{t \to 0} \frac{f(x+th) - f(x)}{t}.$$

Lorsque $E = \mathbb{R}^n$ et $F = \mathbb{R}$, en notant $(e_1, \ldots, e_n)$ la base canonique, on a

$$\frac{\partial f}{\partial x_j}(x) = df(x).e_j = f'(x;e_j) = \lim_{\substack{t \to 0 \\ t \neq 0}} \frac{f(x_1, \ldots, x_{j-1}, x_j + t, x_{j+1}, \ldots, x_n) - f(x)}{t}$$

et la différentielle de Fréchet de f en x s'écrit

$$\boxed{df(x).h = \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x) h_j}$$

Notons que l'existence de dérivées partielles n'implique pas forcément la différentiabilité. Par exemple la fonction $f : \mathbb{R}^2 \to \mathbb{R}$ définie par $f(x,y) = \frac{xy}{x^2+y^2}$ si $(x,y) \neq (0,0)$ et $f(0,0) = 0$ admet des dérivées partielles nulles en $(0,0)$, mais n'est pas différentiable en $(0,0)$ car elle n'y est pas continue.

Jacobienne. Prenons $E = \mathbb{R}^n$ et $F = \mathbb{R}^p$. L'application f a alors p composantes f_i, pour $i = 1, \ldots, p$. La représentation matricielle de la différentielle de Fréchet $df(x)$ (qui est une application linéaire) dans la base canonique est la Jacobienne de f en x

$$J_f(x) = \left(\frac{\partial f_i}{\partial x_j}(x) \right)_{\substack{1 \leqslant i \leqslant p \\ 1 \leqslant j \leqslant n}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial f_p}{\partial x_1}(x) & \cdots & \frac{\partial f_p}{\partial x_n}(x) \end{pmatrix}$$

On a $df(x).h = J_f(x)h$.

Un cas particulier important est lorsque $p = 1$, i.e., pour une fonction $f : \mathbb{R}^n \to \mathbb{R}$. En notant $\langle\ ,\ \rangle$ le produit scalaire euclidien de $\mathbb{R}^n$, on a

$$\boxed{df(x).h = \langle \nabla f(x), h \rangle = \nabla f(x)^\top h = h^\top \nabla f(x)}$$

où

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix}$$

est le gradient de f en x.

De manière plus générale, le gradient est défini dès que E est un espace de Hilbert (il faut un produit scalaire) : pour une fonction $f : E :\to \mathbb{R}$ différentiable, avec E Hilbert, pour tout $x \in E$ le vecteur $\nabla f(x)$ est l'unique vecteur de E tel que $df(x).h = \langle \nabla f(x), h \rangle$ pour tout $h \in E$, où $\langle\ ,\ \rangle$ est le produit scalaire sur E.

Propriétés.

- Si f et g sont deux applications différentiables en x, alors pour tous $\lambda, \mu \in \mathbb{R}$, l'application $\lambda f + \mu g$ est aussi différentiable en x, et

$$d(\lambda f + \mu g)(x) = \lambda df(x) + \mu dg(x).$$

- Si $f : E \to F$ est différentiable en x et $g : F \to G$ est différentiable en $f(x)$, alors $g \circ f$ est différentiable en x et

$$d(g \circ f)(x) = dg(f(x)) \circ df(x).$$

  En particulier, lorsque $E = \mathbb{R}^n$, $F = \mathbb{R}^p$ et $G = \mathbb{R}^m$, en représentant les différentielles par leurs matrices jacobiennes, la composition ci-dessus s'exprime par le produit matriciel.

Applications de classe C^1. Si f est différentiable en tout $x \in U \subset E$, on dit que f est différentiable sur U, et on peut considérer l'application

$$\begin{array}{rcl} df : U & \to & L(E,F) \\ x & \mapsto & df(x) \end{array}$$

On dit que f est de classe C^1 en x si elle est différentiable sur un voisinage ouvert U de x et si l'application $df : U \to L(E,F)$ est continue en x. Ici, l'espace $L(E,F)$ des applications linéaires continues de E dans F est équipé de la norme d'opérateur définie par

$$\|\ell\|_{L(E,F)} = \sup_{x \in E \setminus \{0\}} \frac{\|\ell(x)\|_F}{\|x\|_E} = \sup_{\substack{x \in E \\ \|x\|_E = 1}} \|\ell(x)\|_F \qquad \forall \ell \in L(E,F).$$

Théorème des accroissements finis.

Théorème 1. *Soient E et F des Banach. Soit $f : U \subset E \to F$ différentiable sur l'ouvert U et soient $x, y \in U$ tels que le segment $[x,y] = \{tx + (1-t)y \mid t \in [0,1]\}$ est inclus dans U. Alors*

$$\|f(y) - f(x)\|_F \leqslant \left(\sup_{z \in [x,y]} \|df(z)\|_{L(E,F)} \right) \|y - x\|_E$$

Comme conséquence, si f est C^1 alors f est localement Lipschitzienne.

Remarque 4. Soit $f : U \subset \mathbb{R}^n \to F$ et soit $x \in U$. L'application f est de classe C^1 en x si et seulement si les dérivées partielles de f existent et sont continues en x.

1.2 Dérivées d'ordre supérieur

Soient E et F des Banach. On dit que $f : E \to F$ est deux fois différentiable (au sens de Fréchet) en $x \in E$ si f est différentiable sur un voisinage ouvert U de x et si l'application $df : U \to L(E,F)$ est différentiable en x. On note $d^2f(x) = d(df)(x)$ la différentielle seconde de f en x.

L'application $d^2f(x)$ est un élément de $L(E, L(E,F))$ qu'on identifie comme un élément de $L(E \times E, F)$, i.e., une application bilinéaire.

Théorème 2. *(Théorème de Schwarz) Si f est deux fois différentiable en x alors $d^2f(x)$ est une application bilinéaire symétrique, i.e.,*

$$d^2f(x).(h_1,h_2) = d^2f(x).(h_2,h_1) \qquad \forall h_1, h_2 \in E.$$

Un cas particulier important est lorsque $E = \mathbb{R}^n$ et $F = \mathbb{R}$. Alors $d^2f(x)$ est une forme bilinéaire symétrique (donc, associée à une forme quadratique) qu'on représente par une matrice appelée la Hessienne de f en x, qui est la matrice symétrique

$$\boxed{H_f(x) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x)\right)_{1 \leqslant i,j \leqslant n}}$$

et on a

$$\boxed{d^2f(x).(h,h) = h^\top H_f(x) h \qquad \forall h \in \mathbb{R}^n}$$

De façon générale, par itération, on dit que $f : E \to F$ est k fois différentiable en x si elle est $(k-1)$ fois différentiable sur un voisinage ouvert U de a et si sa différentielle $(k-1)^{\text{ème}}$ $d^{k-1}f : U \to L(E^{k-1}, F)$ est différentiable en x. On note $d^kf(x).(h_1, \ldots, h_k)$ l'action de $d^kf(x)$ sur $(h_1, \ldots, h_k) \in E^k$. Le théorème de Schwarz se généralise à l'ordre k : si f est k fois différentiable en x alors $d^kf(x)$ est une application k-linéaire symétrique.

On dit que f est de classe C^k si l'application d^kf est continue.

Formules de Taylor. On rappelle d'abord la formule de Taylor avec reste intégral. Soit U est un ouvert de E, soit $f : U \to F$ une application de classe C^{k+1}, soient $x \in U$ et $h \in E$ tels que le segment $[x, x+h]$ est contenu dans U. Alors

$$f(x+h) = f(x) + df(x).h + \frac{1}{2}d^2f(x).(h,h) + \cdots + \frac{1}{k!}d^kf(x).(h,\ldots,h) + \frac{1}{k!}\int_0^1 (1-t)^k d^{k+1}f(x+th).(h,\ldots,h)\,dt$$

Ce résultat est très facile à obtenir en se ramenant à la dimension 1, le long du segment $[x, x+h]$. Comme nous allons beaucoup utiliser cette technique par la suite, nous la rappelons ici. On pose

$$\varphi(t) = f(x+th)$$

de façon à ce que $\varphi(0) = f(x)$ et $\varphi(1) = f(x+h)$. On part de $\varphi(1) = \varphi(0) + \int_0^1 \varphi'(t)\,dt$. En intégrant par parties, on dérive φ' et on intègre 1 en $-(1-t)$, on obtient $\varphi(1) = \varphi(0) + \varphi'(0) + \int_0^1 (1-t)\varphi''(t)\,dt$, puis en itérant,

$$\varphi(1) = \varphi(0) + \varphi'(0) + \cdots + \frac{1}{k!}\varphi^{(k)}(0) + \frac{1}{k!}\int_0^1 (1-t)^k \varphi^{(k+1)}(t)\,dt$$

Il reste à noter que $\varphi'(0) = df(x).h$, $\ldots$, $\varphi^{(k)}(0) = d^kf(x).(h,\ldots,h)$ et $\varphi^{(k+1)}(t) = d^{k+1}f(x+th).(h,\ldots,h)$.

Rappelons la formule de Taylor avec reste en o. Si f est k fois différentiable en x alors

$$f(x+h) = f(x) + df(x).h + \frac{1}{2}d^2f(x).(h,h) + \cdots + \frac{1}{k!}d^kf(x).(h,\ldots,h) + \mathrm{o}(\|h\|_E^k)$$

lorsque $h \to 0$. Attention, pour $k > 1$, f peut avoir un développement limité d'ordre k en x, et pour autant, ne pas être k fois différentiable en x. L'équivalence n'est vraie que pour $k = 1$.

Un cas particulier important est lorsque $E = \mathbb{R}^n$ et $k = 2$: si $f : \mathbb{R}^n \to \mathbb{R}$ est deux fois différentiable en x alors

$$\boxed{\begin{aligned} f(x+h) &= f(x) + df(x).h + \frac{1}{2}d^2f(x).(h,h) + \mathrm{o}(\|h\|^2) \\ &= f(x) + \nabla f(x)^\top h + \frac{1}{2}h^\top H_f(x) h + \mathrm{o}(\|h\|^2) \end{aligned}}$$

Exemple 2. Un exemple qu'on utilisera par la suite est la fonction $f : \mathbb{R}^n \to \mathbb{R}$ définie par

$$f(x) = \frac{1}{2} x^\top A x - b^\top x \qquad \forall x \in \mathbb{R}^n$$

où A est une matrice symétrique réelle, et $b \in \mathbb{R}^n$. On a ici

$$df(x).h = x^\top A h - b^\top h, \quad \text{i.e.,} \quad \nabla f(x) = Ax - b \qquad \text{et} \qquad H_f(x) = A$$

1.3 Théorème d'inversion locale, des fonctions implicites

Théorème d'inversion locale. L'objectif du théorème d'inversion locale est de résoudre localement une équation $f(x) = y$. Dans le cas linéaire en dimension finie, ce problème est bien connu : si $A \in GL_n(\mathbb{R})$ alors l'unique solution de $Ax = y$ est $x = A^{-1}y$. Pour une application $f : E \to F$ avec E et F Banach, un résultat similaire est vrai localement sous la condition que la différentielle soit inversible.

Théorème 3. *Soient E et F des Banach. Soit $f : E \to F$ une application de classe C^1 au voisinage de $a \in E$. On suppose que $df(a) \in L(E, F)$ est inversible. Alors il existe un voisinage ouvert U de a dans E et un voisinage ouvert V de $b = f(a)$ dans F tels que l'application $f : U \to V$ (restreinte à U au départ, restreint à V à l'arrivée) est un C^1-difféomorphisme, ce qui veut dire qu'elle est bijective, de classe C^1, et que son inverse est aussi de classe C^1. On peut noter que $df^{-1}(f(a)) = df(a)^{-1}$.*

Lorsque f est de classe C^p, on obtient un C^p-difféomorphisme local.

Remarque 5. Lorsque E est de dimension finie n, ce théorème ne peut s'appliquer que si F est aussi de dimension finie n. Prenons $E = F = \mathbb{R}^n$. L'hypothèse $df(a)$ inversible veut dire que la Jacobienne de f en a est une matrice inversible (inutile de dire "continue" puisque, en dimension finie, toute application linéaire est continue).

Remarque 6. De manière approchée, le théorème d'inversion locale revient à faire une approximation linéaire de f au voisinage de a, en négligeant les termes de reste dans le développement limité à l'ordre 1. Sachant que $f(a) = b$, résolvons $f(x) = y$ avec $x \simeq a$ et $y \simeq b$. On pose $x = a + \delta x$ et $y = b + \delta y$ avec $\|\delta x\|_E$ et $\|\delta y\|_F$ suffisamment petits. On doit donc résoudre

$$f(a + \delta x) = b + \delta y.$$

En toute rigueur le théorème d'inversion locale stipule que $a + \delta x = f^{-1}(b + \delta y)$. En faisant l'approximation à l'ordre 1 :

$$f(a) + df(a).\delta x \simeq b + \delta y$$

(on néglige les termes en $o(\|\delta x\|_E)$, comme $f(a) = b$ on obtient $\delta y \simeq df(a).\delta x$ et donc

$$\delta x \simeq df(a)^{-1}.\delta y.$$

Comme $f^{-1}(b + \delta y) = f^{-1}(b) + df^{-1}(b).\delta y + o(\|\delta y\|_F)$, on retrouve bien le fait que $df^{-1}(f(a)) = df(a)^{-1}$.

Théorème des fonctions implicites. L'objectif est de résoudre une équation $f(x, y) = 0$, où $f : E \times F \to G$, et d'exprimer y comme fonction de x (localement). Le théorème des fonctions implicites est en fait équivalent au théorème d'inversion locale.

Par exemple, en dimension finie, supposons que f soit linéaire : $f(x, y) = A_1 x + A_2 y$ où A_1 et A_2 sont des matrices. Pour résoudre $A_1 x + A_2 y = 0$ en exprimant y comme fonction de x, on a besoin de supposer que A_2 est inversible, et alors on trouve $y = -A_2^{-1} A_1 x$. Dans le cas général non linéaire, A_2 est la différentielle de f par rapport à y et on a le théorème suivant.

Théorème 4. *Soient E, F et G des Banach. Soit $f : E \times F \to G$ de classe C^1, et soit $(a, b) \in E \times F$ tel que $f(a, b) = 0$. On suppose que $\frac{\partial f}{\partial y}(a, b) \in L(F, G)$ (qui est la différentielle de f, uniquement par rapport à la variable $y \in F$) est inversible. Alors il existe un voisinage ouvert V de a dans E, un voisinage ouvert W de b dans F, et une (unique) application $\varphi : V \to W$ de classe C^1 telle que $f(x, \varphi(x)) = 0$ pour tout $x \in V$, avec $\varphi(a) = b$.*

Autrement dit, on a résolu localement autour de (a, b) l'équation $f(x, y) = 0$ par rapport à x (en exprimant, localement, y comme fonction de x).

Si f est de classe C^k avec $k \geqslant 1$ alors φ est C^k.

Pour exprimer la différentielle de φ, il suffit de dériver par rapport à x l'égalité $f(x, \varphi(x)) = 0$. On obtient $\frac{\partial f}{\partial x}(x, \varphi(x)) + \frac{\partial f}{\partial y}(x, \varphi(x)) \circ d\varphi(x) = 0$, d'où

$$d\varphi(x) = -\left(\frac{\partial f}{\partial y}(x, \varphi(x))\right)^{-1} \frac{\partial f}{\partial x}(x, \varphi(x)).$$

Cette formule est exactement $y = -A_2^{-1} A_1 x$ dans le cas linéaire.

Cela était attendu, car comme dans la remarque 6 ci-dessus, pour $\|\delta x\|_E$ et $\|\delta x\|_E$ suffisamment petits, de manière approchée on résout l'équation

$$f(a + \delta x, b + \delta y) = 0$$

en faisant une approximation du premier ordre en (a, b) :

$$\frac{\partial f}{\partial x}(a, b).\delta x + \frac{\partial f}{\partial y}(a, b).\delta y \simeq 0$$

(on néglige les termes en $o(\|\delta x\|_E)$ et $o(\|\delta y\|_F)$), et donc

$$\delta y = -\left(\frac{\partial f}{\partial y}(a, b)\right)^{-1} \frac{\partial f}{\partial x}(a, b).\delta x$$

qui est exactement $\delta y = d\varphi(a).\delta x$, comme attendu puisque, en toute rigueur on sait que $b + \delta y = \varphi(a + \delta x)$.

Chapitre 2

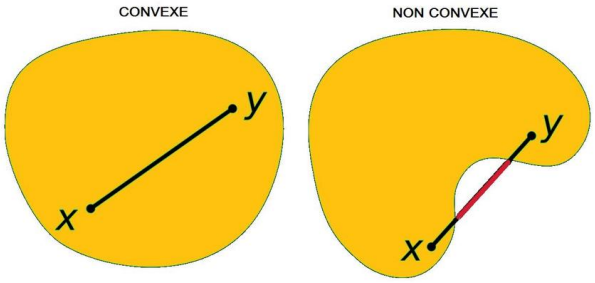
Convexité

2.1 Définition et propriétés

Définition 1. *Soit E un espace vectoriel. Un sous-ensemble $C \subset E$ est dit convexe si*

$$\boxed{\forall x, y \in C \qquad \forall t \in [0,1] \qquad tx + (1-t)y \in C}$$

Autrement dit, le segment $[x, y]$ est inclus dans C, pour tous points $x, y \in C$.



Dans $\mathbb{R}$, les ensembles convexes sont exactement les intervalles. Une union disjointe d'intervalles n'est pas convexe.

Définition 2. *Soit $C \subset E$ un sous-ensemble convexe de l'espace vectoriel E. Une fonction $f : C \to \mathbb{R} \cup \{+\infty\}$ est dite convexe si*

$$\boxed{\forall x, y \in C \cap \mathrm{Dom}(f) \quad \forall t \in [0,1] \qquad f(tx + (1-t)y) \leqslant tf(x) + (1-t)f(y)}$$

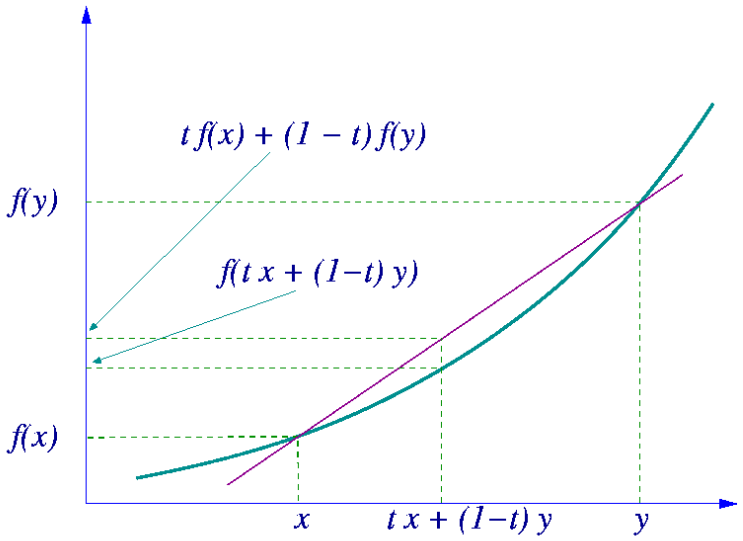
où $\mathrm{Dom}(f) = \{x \in E \ \mid \ f(x) < +\infty\}$.

(autrement dit, f est en dessous de ses "cordes", voir la figure)

Notons que cela est équivalent à

$$\forall k \in \mathbb{N}^* \qquad \forall x_1, \ldots, x_k \in C \cap \mathrm{Dom}(f) \qquad \forall \lambda_1, \ldots, \lambda_k \in [0,1] \ \mid \ \sum_{i=1}^{k} \lambda_i = 1$$
$$f\left(\sum_{i=1}^{k} \lambda_i x_i\right) \leqslant \sum_{i=1}^{k} \lambda_i f(x_i)$$

autrement dit la valeur prise par f sur toute combinaison convexe de points x_i est plus petite que la combinaison convexe des valeurs de f aux points x_i.



On dit que f est strictement convexe si

$$\boxed{\forall x, y \in C \cap \mathrm{Dom}(f),\ x \neq y \quad \forall t \in]0,1[\qquad f(tx + (1-t)y) < tf(x) + (1-t)f(y)}$$

On suppose que E est un espace vectoriel normé. Pour $\alpha > 0$, la fonction f est dite α-convexe (ou fortement convexe de module α) si

$$\boxed{\forall x, y \in C \cap \mathrm{Dom}(f) \quad \forall t \in [0,1] \qquad f(tx + (1-t)y) \leqslant tf(x) + (1-t)f(y) - \frac{\alpha}{2}t(1-t)\|x-y\|_E^2}$$

Dans la littérature existante, on trouve parfois le vocabulaire suivant à propos des fonctions convexes : lorsque $\mathrm{Dom}(f) \neq \emptyset$, on dit que f est propre.

On a les implications : α-convexe $\Rightarrow$ strictement convexe $\Rightarrow$ convexe.

Exemple 3.

- Dans un espace vectoriel normé, la fonction $f(x) = \|x\|$ est convexe.
  Dans $\mathbb{R}^n$, la fonction $f(x) = \|x\|_2^2$ est strictement convexe. La fonction $f(x) = \|x\|_\infty^2$ ou $\|x\|_1^2$ est convexe mais pas strictement convexe.
- Toute application linéaire ou affine est convexe (mais pas strictement convexe).
- Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice carrée symétrique positive d'ordre n. Soit $b \in \mathbb{R}^n$. La fonction

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x$$

  est convexe. Elle est strictement convexe lorsque A est définie positive.
  Plus généralement, dans un Hilbert H, soit $A : H \to H$ un opérateur linéaire autoadjoint. On suppose que A est monotone, i.e., $\langle A(x-y), x-y\rangle_H \geqslant 0$ pour tous $x, y \in H$ (de manière équivalente, puisque A est linéaire : $\langle Ax, x\rangle \geqslant 0$ pour tout $x \in H$; on dit aussi que $-A$ est dissipatif). Soit $b \in H$. La fonction

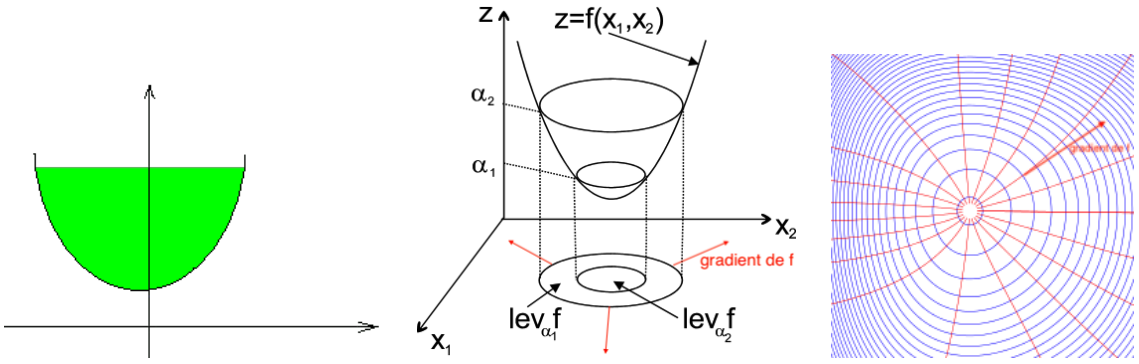
$$f(x) = \frac{1}{2}\langle x, Ax\rangle_H - \langle b, x\rangle_H$$

  est convexe.

Remarque 7. Une fonction f est convexe si et seulement si son épigraphe

$$\mathrm{Epi}(f) = \{(x, \alpha) \in \mathrm{Dom}(f) \times \mathbb{R} \ \mid \ f(x) \leqslant \alpha\}$$

(c'est-à-dire, ce qui est au-dessus du graphe de f) est convexe.



Si une fonction est convexe, alors, pour tout $\alpha \in \mathbb{R}$, l'ensemble de "sous-niveau"

$$\{x \in \mathrm{Dom}(f) \ \mid \ f(x) \leqslant \alpha\}$$

("sublevel set" en anglais) est convexe. La réciproque est fausse : par exemple la fonction $f(x) = \sqrt{|x|}$ n'est pas convexe, mais ses ensembles de sous-niveau sont convexes (on dit qu'elle est "quasi-convexe").

Remarque 8. Pour les fonctions $f : \mathbb{R} \to \mathbb{R}$, on rappelle les propriétés suivantes :

- lorsque f est dérivable, f est (strictement) convexe si et seulement si f' est (strictement) croissante, si et seulement si le graphe de f est (strictement, sauf au point de tangence) au-dessus de ses tangentes ;
- lorsque f est deux fois dérivable, f est convexe si et seulement si $f'' \geqslant 0$; f est strictement convexe si et seulement si $f'' \geqslant 0$ et ne s'annule que sur un ensemble d'intérieur vide.
  (NB : $f(x) = x^4$ est strictement convexe mais $f''(0) = 0$)

Remarque 9. Comme on l'a déjà vu pour la formule de Taylor, on peut démontrer de nombreuses propriétés en multi-D à partir des propriétés en 1D. La technique est toujours la même : étant donnée une fonction $f : E \to \mathbb{R}$ (où E est un espace vectoriel), étant donnés deux points $x, y \in E$, on regarde la fonction f le long du segment $[x, y]$ en posant

$$\varphi(t) = f((1-t)x + ty) = f(x + t(y-x)) \qquad \forall t \in [0,1].$$

On a $\varphi(0) = f(x)$ et $\varphi(1) = f(y)$.

Montrons que f est convexe si et seulement si φ est convexe pour tous $x, y \in E$.

Si f est convexe, alors, pour tout $\lambda \in [0,1]$,

$$\begin{aligned}\varphi(\lambda t_1 + (1-\lambda)t_2) &= f(x + (\lambda t_1 + (1-\lambda)t_2)(y-x)) \\ &= f(\lambda(x + t_1(y-x)) + (1-\lambda)(x + t_2(y-x))) \\ &\leqslant \lambda f(x + t_1(y-x)) + (1-\lambda) f(x + t_2(y-x)) = \lambda\varphi(t_1) + (1-\lambda)\varphi(t_2)\end{aligned}$$

donc φ est convexe.

Réciproquement, si φ est convexe, alors en prenant $t_1 = 0$ et $t_2 = 1$, on a $\varphi(1-\lambda) \leqslant \lambda\varphi(0) + (1-\lambda)\varphi(1)$, ce qui donne $f(\lambda x + (1-\lambda)y) \leqslant \lambda f(x) + (1-\lambda) f(y)$ donc f est convexe.

Continuité des fonctions convexes.

Théorème 5. *Soit E un espace vectoriel topologique, et soit $f : E \to \mathbb{R} \cup \{+\infty\}$ une fonction convexe. Si f est localement bornée autour d'un point $x \in E$ (qui est dans l'intérieur de* $\mathrm{Dom}(f)$*) alors f est continue en x.*

Comme conséquence, si E est de dimension finie alors f est continue sur l'intérieur de $\mathrm{Dom}(f)$.

Démonstration. Sans perte de généralité, on suppose que 0 est dans l'intérieur de $\mathrm{Dom}(f)$ et que $f(0) = 0$. Soit V un voisinage ouvert convexe de 0 et soit $a > 0$ tel que $f(x) \leqslant a$ pour tout $x \in V$. L'ouvert $W = V \cap (-V)$ est aussi un voisinage ouvert convexe de 0, qui est de plus symétrique par rapport à 0). Soit $\varepsilon \in]0, 1[$.

Pour tout $x \in \varepsilon W$ (i.e., $\frac{x}{\varepsilon} \in W$), x s'écrit comme la combinaison convexe $x = (1-\varepsilon)0 + \varepsilon\frac{x}{\varepsilon}$, donc par convexité de f,

$$f(x) \leqslant (1-\varepsilon)f(0) + \varepsilon f\Big(\frac{x}{\varepsilon}\Big) = \varepsilon f\Big(\frac{x}{\varepsilon}\Big) \leqslant \varepsilon a.$$

Par ailleurs, on a $-\frac{x}{\varepsilon} \in W$, et comme 0 s'écrit comme la combinaison convexe $0 = \frac{1}{1+\varepsilon}x + \frac{\varepsilon}{1+\varepsilon}\frac{-x}{\varepsilon}$, par convexité de f,

$$0 = f(0) \leqslant \frac{1}{1+\varepsilon}f(x) + \frac{\varepsilon}{1+\varepsilon}f\Big(\frac{-x}{\varepsilon}\Big)$$

donc $f(x) \geqslant -\varepsilon f\big(\frac{-x}{\varepsilon}\big) \geqslant -\varepsilon a$. De ces deux inégalités, on déduit que $|f(x)| \leqslant \varepsilon a$ pour tout $x \in \varepsilon W$. On conclut que f est continue en 0.

En dimension finie, si l'intérieur de $\mathrm{Dom}(f)$ est non vide, il contient $n+1$ points x_i affinement indépendants. Par convexité, on trouve que

$$\forall \lambda_1, \dots, \lambda_{n+1} > 0 \ \mid \ \sum_{i=1}^{n+1} \lambda_i = 1 \qquad f\left(\sum_{i=1}^{n+1} \lambda_i x_i\right) \leqslant \sum_{i=1}^{n+1} \lambda_i f(x_i) \leqslant \max_{1 \leqslant i \leqslant n+1} f(x_i).$$

On en déduit que f est localement bornée en tout point de l'intérieur de $\mathrm{Dom}(f)$. □

Fonctions convexes différentiables et leurs tangentes.

Théorème 6. *Soit E un espace vectoriel normé et soit $C \subset E$ un sous-ensemble convexe non vide. Soit $f : C \to \mathbb{R}$ (i.e., $C \subset \mathrm{Dom}(f)$) une fonction Gateaux différentiable. Alors :*

f est convexe

$\Leftrightarrow$ $\boxed{\forall x, y \in C \quad f(y) \geqslant f(x) + df(x).(y-x)}$ *(on devrait plutôt noter $f'(x; y-x)$)*

autrement dit, le graphe de f est au-dessus de ses tangentes.

(lorsque E est un espace de Hilbert et f est différentiable, on a $df(x).(y-x) = \langle \nabla f(x), y-x \rangle$)

$\Leftrightarrow$ $\forall x, y \in C \quad (df(x) - df(y)).(x-y) \geqslant 0$ *(i.e.., df est monotone)*

et

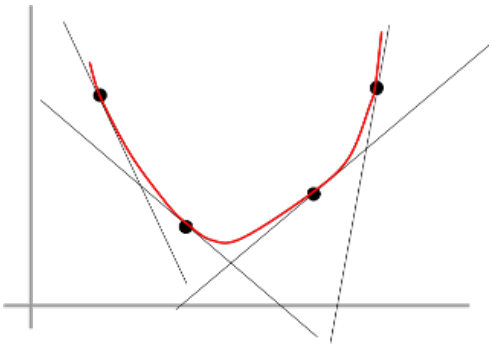
f est strictement convexe

$\Leftrightarrow$ $\forall x, y \in C,\ x \neq y, \quad f(y) > f(x) + df(x).(y-x)$

autrement dit, le graphe de f est strictement au-dessus de ses tangentes (sauf au point de tangence).

$\Leftrightarrow$ $\forall x, y \in C,\ x \neq y, \quad (df(x) - df(y)).(x-y) > 0$ *(i.e., df est strictement monotone)*

On parle de *minorantes affines* de f.



Démonstration. Suivant la remarque 9, en posant $\varphi(t) = f(x + t(x - y))$, f est convexe si et seulement si φ est convexe pour tous $x, y \in E$. On a $\varphi'(t) = df(x + t(x - y)).(x - y)$. En 1D, on sait que φ est convexe si et seulement si φ' est croissante, or $\varphi'(1) \geqslant \varphi'(0)$ si et seulement si $df(y).(y - x) \geqslant df(x).(y - x)$, ce qui donne la propriété de monotonie. D'autre part, en 1D, on sait que φ est convexe si et seulement si φ est au-dessus de ses tangentes, or $\varphi(1) \geqslant \varphi(0) + \varphi'(0)$ si et seulement si $f(y) \geqslant f(x) + df(x).(y - x)$.

Donnons toutefois une preuve directe de la propriété d'être au-dessus de ses tangentes. Si f est convexe alors on écrit l'inégalité de convexité légèrement différemment :

$$\forall t \in]0, 1] \qquad f(x + t(y - x)) - f(x) \leqslant t(f(y) - f(x)),$$

on divise par t et on fait tendre $t \to 0$ pour obtenir l'inégalité du théorème. Réciproquement, on applique l'inégalité d'être au-dessus de ses tangentes en remplaçant le couple (x, y) par :

- $(x + t(y - x), x)$, d'où $f(x) \geqslant f(x + t(y - x)) + t\, df(x + t(y - x)).(y - x)$
- $(x + t(y - x), y)$, d'où $f(y) \geqslant f(x + t(y - x)) + (1 - t)\, df(x + t(y - x)).(y - x)$

puis on multiplie la première inégalité par $(1 - t)$, la seconde par t, on somme et on obtient $(1 - t)f(x) + tf(y) \geqslant f(x + t(y - x))$, ce qui est l'inégalité de convexité. □

Remarque 10. Voici quelques compléments (admis).
On note $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$. Soit $x \in E$ tel que $f(x) < +\infty$. Alors :

- Pour tout $h \in E$, l'application $t > 0 \mapsto \frac{f(x+th)-f(x)}{t} \in \overline{\mathbb{R}}$ est croissante.
- Toute dérivée directionnelle existe dans $\overline{\mathbb{R}}$: pour tout $h \in E$, $f'(x; h) \in \overline{\mathbb{R}}$, et on a $f'(x; h) = +\infty \;\Leftrightarrow\; \forall t > 0 \quad x + th \notin \mathrm{Dom}(f)$.
- $f'(x; h) \geqslant -f'(x; -h)$: en particulier, si l'une des deux vaut $-\infty$ alors l'autre vaut $+\infty$;
- l'application $h \in E \mapsto f'(x; h)$ est sous-linéaire. Elle est linéaire si et seulement si $f'(x, h) < +\infty$ pour tout $h \in E$, si et seulement si $f'(x; h) = -f'(x; -h)$ pour tout $h \in E$.
- (Theorème de Mazur) On suppose que E est un Banach séparable. Soit $U \subset E$ un ouvert convexe et soit $f : U \to \mathbb{R}$ une fonction convexe continue. Alors f est Gateaux différentiable sur un sous-ensemble dense de U.
- Toute fonction convexe $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ est Fréchet différentiable presque partout sur l'intérieur de $\mathrm{Dom}(f)$.

Caractérisation de la convexité par la Hessienne.

Théorème 7. *Soit E un espace vectoriel et soit $f : E \to \mathbb{R}$ une fonction deux fois différentiable sur $\mathrm{Dom}(f)$. La fonction f est convexe si et seulement si, pour tout $x \in \mathrm{Dom}(f)$, $d^2f(x)$ est une forme quadratique positive. Si $d^2f(x)$ est définie positive en tout $x \in \mathrm{Dom}(f)$ alors f est strictement convexe (la réciproque est fausse : prendre $f(x) = x^4$).*

Lorsque E est de dimension finie, cela s'exprime sous la forme $\boxed{H_f(x) \geqslant 0}$, i.e., f est convexe si et seulement si sa Hessienne est en tout point une matrice symétrique positive. Si la Hessienne est définie positive en tout point alors f est strictement convexe.

Démonstration. Suivant la remarque 9, en posant $\varphi(t) = f(x + t(x - y))$, f est convexe si et seulement si φ est convexe pour tous $x, y \in E$. On a $\varphi''(t) = d^2f(x + t(y - x)).(y - x, y - x)$. Or, en 1D, φ est convexe si et seulement si $\varphi'' \geqslant 0$. Le résultat s'ensuit. □

Exemple 4. Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice symétrique réelle et soit $b \in \mathbb{R}^n$. La fonction

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x$$

est convexe (resp., strictement convexe) si et seulement si $A \geqslant 0$ (resp., $A > 0$).

Théorème 8. *Soit H un Hilbert, soit $C \subset H$ un sous-ensemble convexe et soit $f : C \to \mathbb{R}$ une fonction différentiable sur C. Soit $\alpha \geqslant 0$.*

$$\begin{aligned}
&f \text{ est } \alpha\text{-convexe sur } C \\
\Leftrightarrow\ &x \mapsto f(x) - \tfrac{\alpha}{2}\|x\|^2 \text{ est convexe sur } C \\
\Leftrightarrow\ &\forall x, y \in C \qquad f(y) \geqslant f(x) + df(x).(y - x) + \tfrac{\alpha}{2}\|y - x\|^2 \\
\Leftrightarrow\ &\forall x, y \in C \qquad (df(x) - df(y)).(x - y) \geqslant \alpha\|x - y\|^2
\end{aligned}$$

Si f est deux fois différentiable, alors f est α-convexe si et seulement si $d^2f(x).(h, h) \geqslant \alpha\|h\|^2$ pour tout $x \in C$, $h \in H$. En dimension finie, cette condition s'exprime sous la forme $\boxed{H_f(x) \geqslant \alpha I_n}$.

La troisième propriété stipule que non seulement le graphe de f est au-dessus de ses tangentes, mais que, en plus, entre les deux on peut placer une parabole. Cela veut donc dire que le graphe de f s'éloigne (au-dessus) de ses tangentes au moins comme un carré.

Démonstration. Posons $g(x) = f(x) - \frac{\alpha}{2}\|x\|^2$. La fonction g est convexe si et seulement si $tg(x) + (1 - t)g(y) - g(tx + (1 - t)y) \geqslant 0$ pour tout $t \in [0, 1]$ et tous $x, y \in C$. Or,

$$\begin{aligned}
tg(x) + (1 - t)g(y) - g(tx + (1 - t)y) = {} & tf(x) + (1 - t)f(y) - f(tx + (1 - t)y) \\
& \underbrace{-\frac{\alpha}{2}t\|x\|^2 - \frac{\alpha}{2}(1 - t)\|y\|^2 + \frac{\alpha}{2}\|tx + (1 - t)y\|^2}_{-\frac{\alpha}{2}t(1-t)\|x-y\|^2}
\end{aligned}$$

cette dernière égalité car

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

et ainsi on trouve que g est convexe si et seulement si f est α-convexe. Le reste est facile. □

2.2 Théorème de projection sur un convexe fermé

Soit H un espace de Hilbert. On rappelle tout d'abord l'identité du parallélogramme :

$$\forall x, y \in H \qquad \left\| \frac{x+y}{2} \right\|^2 + \left\| \frac{x-y}{2} \right\|^2 = \frac{1}{2} \left(\|x\|^2 + \|y\|^2 \right)$$

(qu'on retrouve immédiatement en développant par Pythagore).

Théorème 9. *Soit $C \subset H$ un convexe fermé non vide. Pour tout $x \in H$ il existe un unique $x^* \in C$ tel que*

$$\forall z \in C \qquad \langle x - x^*, z - x^* \rangle \leqslant 0 \tag{2.1}$$

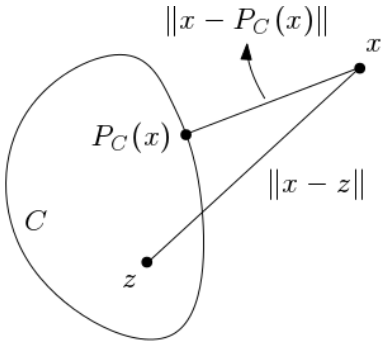
et on a

$$\|x - x^*\| = \min_{z \in C} \|x - z\|. \tag{2.2}$$

On note $x^* = P_C(x)$, appelé la projection de x sur C. On a (2.1) $\Leftrightarrow$ (2.2).

De plus, P_C est 1-Lipschitzienne, i.e.,

$$\|P_C(x) - P_C(y)\| \leqslant \|x - y\| \qquad \forall x, y \in H.$$



Ainsi, $x^* = P_C(x)$ est l'unique minimiseur du problème de minimisation $\min_{z \in C} \|x - z\|$.

Démonstration. Il s'agit de montrer que $d = \inf_{z \in C} \|x - z\|$ est atteint, i.e., que c'est un minimum. Soit $(x_n)_{n \in \mathbf{N}}$ une suite (dite minimisante) de C telle que $\|x - x_n\| \to d = \inf_{z \in C} \|x - z\|$. Montrons que la suite est de Cauchy. Par l'identité du parallélogramme, on a

$$\left\| x - \frac{x_n + x_m}{2} \right\|^2 + \left\| \frac{x_n - x_m}{2} \right\|^2 = \frac{1}{2} \left(\|x - x_n\|^2 + \|x - x_m\|^2 \right)$$

et comme $d \leqslant \left\| x - \frac{x_n + x_m}{2} \right\|^2$ (car $\frac{x_n + x_m}{2} \in C$ par convexité de C) on obtient

$$\left\| \frac{x_n - x_m}{2} \right\|^2 \leqslant \frac{1}{2} \left(\underbrace{\|x - x_n\|^2}_{\downarrow d} + \underbrace{\|x - x_m\|^2}_{\downarrow d} - 2d \right) \to 0$$

donc la suite est de Cauchy, donc elle converge (car l'espace est complet) vers un $x^* \in C$ (car C est fermé), et on obtient (2.2).

Montrons que (2.2) $\Rightarrow$ (2.1). Pour tout $t \in]0, 1]$, on a, par la propriété de minimisation,

$$\|x - x^*\|^2 \leqslant \|x - (1-t)x^* - tz\|^2 = \|x - x^* + t(x^* - z)\|^2$$

donc en développant, $0 \leqslant 2t\langle x - x^*, x^* - z\rangle + t^2\|x^* - z\|^2$, puis on divise par t et on fait tendre $t \to 0$ et on obtient (2.1).

Montrons que (2.1) $\Rightarrow$ (2.2). On a

$$\|x - x^*\|^2 - \|x - z\|^2 = \|x - x^*\|^2 - \|x - x^* + x^* - z\|^2 = 2\langle x - x^*, z - x^*\rangle - \|x^* - z\|^2 \leqslant 0$$

d'où (2.2).

Montrons que P_C est 1-Lipschitzienne. On note $x^* = P_C(x)$ et $y^* = P_C(y)$. Par (2.1), on a

$$\langle x - x^*, z - x^*\rangle \leqslant 0 \qquad \text{et} \qquad \langle y - y^*, z - y^*\rangle \leqslant 0 \qquad \forall z \in C.$$

On prend $z = y^*$ dans la première inégalité, et $z = x^*$ dans la seconde, et on somme. On obtient

$$\|x^* - y^*\|^2 \leqslant \langle x - y, x^* - y^*\rangle \qquad \forall x, y \in H$$

or par l'inégalité de Cauchy-Schwarz on a $|\langle x - y, x^* - y^*\rangle| \leqslant \|x - y\|\|x^* - y^*\|$, d'où finalement $\|x^* - y^*\| \leqslant \|x - y\|$. □

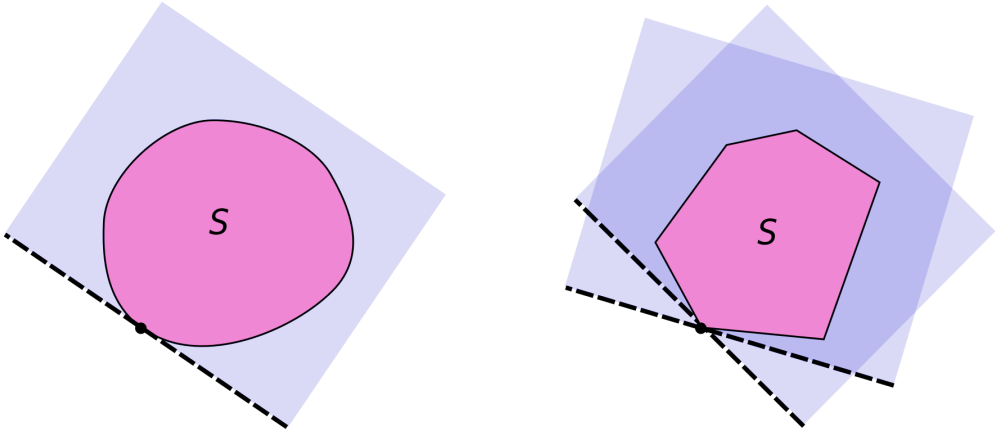
Corollaire 1. *Soit $M \subset H$ un sous-espace vectoriel fermé. Alors P_M est linéaire et, pour tout $x \in H$, $x^* = P_M x$ est caractérisé par : $x^* \in M$ et $\langle x - x^*, y\rangle = 0$ pour tout $y \in M$. Autrement dit, x^* est le projeté orthogonal de x sur M.*

Démonstration. On a $\langle x - x^*, ty - x^*\rangle \leqslant 0$ pour tout $y \in M$ et tout $t \in \mathbb{R}$, d'où $\langle x - x^*, y\rangle = 0$ pour tout $y \in M$. La réciproque est triviale. □

Remarque 11. Soit $C \subset H$ convexe et soit $x \in \partial C = \bar{C} \setminus \mathring{C}$. Alors il existe (au moins) un hyperplan séparant x et C au sens large :

$$\exists p \in H \setminus \{0\} \;\mid\; \forall y \in C \quad \langle p, y\rangle \leqslant \langle p, x\rangle$$

Ce théorème de séparation dans un Hilbert est un cas particulier du théorème de Hahn-Banach (dans un espace de Banach).



Remarque 12. Soit $A \subset H$ un sous-ensemble. On appelle $\overline{\text{Conv}}(A)$ l'enveloppe convexe fermée de A, autrement dit, le plus petit convexe fermé contenant A. D'après la propriété précédente, $\overline{\text{Conv}}(A)$ est l'intersection de tous les demi-espaces fermés contenant A.

2.3 Sous-différentiabilité des fonctions convexes

Soit H un espace de Hilbert.

Définition 3. *Soit $f : H \to \mathbb{R} \cup \{+\infty\}$ une fonction convexe. Un vecteur $p \in H$ est appelé sous-gradient de f au point $x \in \mathrm{Dom}(f)$ si*

$$\forall y \in \mathrm{Dom}(f) \qquad f(y) \geqslant f(x) + \langle p, y - x \rangle$$

Lorsque f est Gateaux différentiable, cela est équivalent à

$$\forall h \in H \qquad f'(x; h) \geqslant \langle p, h \rangle$$

On appelle sous-différentiel de f en x l'ensemble

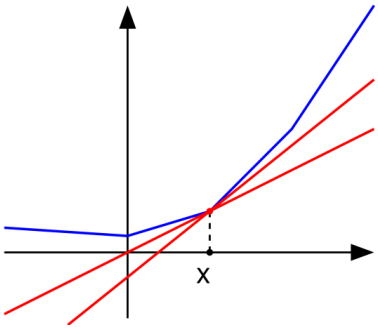
$$\partial f(x) = \{p \in H \ \mid \ \forall y \in \mathrm{Dom}(f) \quad f(y) \geqslant f(x) + \langle p, y - x \rangle\},$$

de tous les sous-gradients de f en x.

On dit que f est sous-différentiable en x lorsque $\partial f(x) \neq \emptyset$.

L'interprétation géométrique du sous-différentiel est la suivante : c'est l'ensemble de toutes les normales aux "hyperplans d'appui" du graphe de f, i.e., les hyperplans qui passent par le point $(x, f(x))$ et qui sont sous le graphe de f (rappelons que, comme f est convexe, le graphe de f est au-dessus de ses tangentes là où f est différentiable).

Par exemple, si $f : \mathbb{R} \to \mathbb{R}$ est définie par $f(x) = |x|$, on a $\partial f(x) = \{-1\}$ si $x < 0$, $\{+1\}$ si $x > 0$, et $[-1, 1]$ si $x = 0$.



Lorsque la fonction convexe f est différentiable au sens de Fréchet en $x \in H$, alors $\partial f(x)$ est le singleton :

$$\partial f(x) = \{\nabla f(x)\}.$$

Lemme 1. *Pour tout $x \in \mathrm{Dom}(f)$, $\partial f(x)$ est un ensemble convexe fermé.*

Démonstration. Soient $p_1, p_2 \in \partial f(x)$. Alors, pour tout $y \in \mathrm{Dom}(f)$, $f(y) \geqslant f(x) + \langle p_1, y - x \rangle$ et $f(y) \geqslant f(x) + \langle p_2, y - x \rangle$, d'où, pour tout $\lambda \in [0, 1]$, $f(y) \geqslant f(x) + \langle \lambda p_1 + (1 - \lambda) p_2, y - x \rangle$, et donc $\lambda p_1 + (1 - \lambda) p_2 \in \partial f(x)$. Le caractère fermé est évident. □

La notion de sous-différentiel n'est pas restreinte aux espaces de Hilbert. Lorsque H est un espace vectoriel topologique, on remplace dans la définition le produit scalaire par le crochet de dualité $\langle \ , \ \rangle_{H',H}$.

La notion de sous-différentiel n'est pas restreinte aux fonctions convexes. On peut définir la sous-différentiabilité d'une fonction dans un cadre très général (théorie d'analyse non lisse). Mais, dans cette courte section introductive, on se limite aux fonctions convexes.

Définition 4. *Pour E espace topologique, une fonction $f : E \to \mathbb{R}$ est semi-continue inférieurement (sci) en $x \in E$ si pour tout $\varepsilon > 0$ il existe un voisinage ouvert U de x dans E tel que, pour tout $y \in U$, $f(y) \geqslant f(x) - \varepsilon$.*

La fonction f est sci sur E si et seulement si, pour tout $\alpha \in \mathbb{R}$, l'ensemble $\{x \in E \mid f(x) \leqslant \alpha\}$ est fermé, si et seulement si l'épigraphe $\mathrm{Epi}(f) = \{(x, \alpha) \in \mathrm{Dom}(f) \times \mathbb{R} \mid f(x) \leqslant \alpha\}$ est fermé.

Lorsque E est un espace métrique, f est sci en x si et seulement si

$$f(x) \leqslant \liminf_{y \to x} f(y).$$

Bien que ce ne soit pas utile ici, précisons qu'on peut définir, de même, la notion de fonction semi-continue supérieurement (scs) : simplement, f est scs si et seulement si $-f$ est sci (ce qui donne, dans le cas métrique, $f(x) \geqslant \limsup_{y \to x} f(y)$).

Théorème 10. *Toute fonction convexe sci $f : H \to \mathbb{R} \cup \{+\infty\}$ est sous-différentiable sur l'intérieur de son domaine : pour tout x appartenant à l'intérieur de $\mathrm{Dom}(f)$, l'ensemble $\partial f(x)$ est un convexe non vide. Il est de plus borné si H est de dimension finie.*

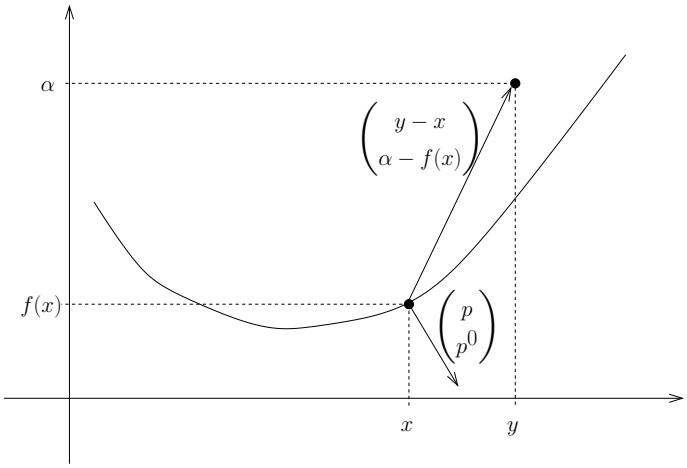
Démonstration. Comme f est convexe sci, $\mathrm{Epi}(f)$ est un ensemble convexe fermé. Pour $x \in \mathrm{Dom}(f)$, le point $(x, f(x))$ appartient à la frontière de $\mathrm{Epi}(f)$. Donc par le théorème de séparation convexe (voir remarque 11 et voir la figure), il existe $(p, p^0) \in H \times \mathbb{R} \setminus \{(0,0)\}$ (défini à scalaire multiplicatif > 0 près) tel que

$$\left\langle \begin{pmatrix} p \\ p^0 \end{pmatrix}, \begin{pmatrix} y - x \\ \alpha - f(x) \end{pmatrix} \right\rangle \leqslant 0 \qquad \forall (y, \alpha) \in \mathrm{Epi}(f)$$

de sorte que

$$\langle p, y - x \rangle + p^0(\alpha - f(x)) \leqslant 0.$$

En prenant $y = x$ et $\alpha = f(x) + t$, $t \geqslant 0$, on voit que, forcément, $p^0 \leqslant 0$.



Montrons par l'absurde que $p^0 < 0$. Si $p^0 = 0$ alors $\langle p, y - x \rangle \leqslant 0$ pour tout $y \in \mathrm{Dom}(f)$. Comme x est dans l'intérieur de $\mathrm{Dom}(f)$, il existe $\varepsilon > 0$ tel que la boule fermée $\bar{B}(x, \varepsilon)$ soit incluse dans l'intérieur de $\mathrm{Dom}(f)$. En prenant $y = x + h$ avec $h \in \bar{B}(x, \varepsilon)$, on obtient donc $\langle p, h \rangle = 0$ pour tout $h \in \bar{B}(x, \varepsilon)$. Mais cela implique $p = 0$ (en effet, prendre $h = tp$ avec t positif et négatif et $|t|$ assez petit). Autrement dit, $(p, p^0) = (0, 0)$, ce qui est une contradiction car $(p, p^0) \neq (0, 0)$.

Comme le couple (p, p^0) peut être multiplié par un scalaire strictement positif sans changer les inégalités ci-dessus, quitte à le multiplier par $-1/p^0$ on se ramène à $p^0 = -1$. Ainsi, on a obtenu $\langle p, y - x\rangle \leqslant \alpha - f(x)$ pour tout $(y, \alpha) \in \mathrm{Epi}(f)$. En particulier, pour $\alpha = f(y)$, cela donne $f(y) \geqslant f(x) + \langle p, y - x\rangle$. On a bien obtenu l'existence d'un $p \in \partial f(x)$, i.e., $\partial f(x) \neq \emptyset$.

Il reste à montrer que $\partial f(x)$ est borné si H est de dimension finie. Par l'absurde, s'il n'est pas borné alors il existe une suite $(p_k)_{k\in\mathbb{N}}$ d'éléments de $\partial f(x)$, tels que $\|p_k\| \to +\infty$. Par définition, on a $f(y) \geqslant f(x) + \langle p_k, y - x\rangle$ pour tout $y \in \mathrm{Dom}(f)$, pour tout $k \in \mathbb{N}$. Divisons par $\|p_k\|$ et posons $\Psi_k = \frac{p_k}{\|p_k\|}$. On a

$$\frac{1}{\|p_k\|} f(y) \geqslant \frac{1}{\|p_k\|} f(x) + \langle \Psi_k, y - x\rangle \qquad \forall y \in \mathrm{Dom}(f).$$

Le vecteur Ψ_k appartient à la sphère unité de H qui est compacte car H est de dimension finie, donc à sous-suite près on a $\Psi_k \to \Psi$ avec $\|\Psi\| = 1$. Comme $\|p_k\| \to +\infty$, on obtient en passant à la limite $\langle \Psi, y - x\rangle = 0$ pour tout $y \in \mathrm{Dom}(f)$. Comme x est dans l'intérieur de $\mathrm{Dom}(f)$, on en déduit comme précédemment que $\Psi = 0$, ce qui est absurde puisque $\|\Psi\| = 1$. □

La notion de sous-différentiabilité généralise la notion de différentiabilité (y compris pour des fonctions non convexes). Lorsqu'une fonction $f : H \to \mathbb{R}$ est différentiable au sens de Fréchet en $x \in H$, alors $\partial f(x) = \{\nabla f(x)\}$. Il existe une théorie très développée de la sous-différentiabilité (appelée "analyse non lisse"), et beaucoup de propriétés des fonctions sous-différentiables, avec tout un "calcul sous-différentiel" (modules d'analyse convexe ou analyse non lisse, généralement étudiés au niveau M2).

Ici, on se contente d'une brève introduction, en se restreignant de plus aux fonctions convexes, mais on mentionne le fait très simple suivant : le sous-différentiel permet notamment de caractériser les minimiseurs des fonctions convexes.

Théorème 11. *Soit $f : H \to \mathbb{R}$ une fonction convexe sous-différentiable en $x^* \in H$. Le point $x^* \in H$ est un minimiseur de la fonction f sur H si et seulement si $0 \in \partial f(x^*)$:*

$$\boxed{f(x^*) = \min_{x\in H} f(x) \quad \Leftrightarrow \quad 0 \in \partial f(x^*)}$$

Démonstration. Si $0 \in \partial f(x^*)$ alors par définition,

$$f(x) \geqslant f(x^*) + \langle 0, x - x^*\rangle = f(x^*)$$

pour tout $x \in \mathrm{Dom}(f)$. Réciproquement, si $f(x) \geqslant f(x^*)$ pour tout $x \in \mathrm{Dom}(f)$ alors $0 \in \partial f(x^*)$ par définition. □

Remarque 13. Soit $f : H \to \mathbb{R} \cup \{+\infty\}$ une fonction convexe telle que $\mathrm{Dom}(f) \neq \emptyset$. Soit $x \in \mathrm{Dom}(f)$. Les propriétés suivantes sont équivalentes :

- $p \in H$ est un sous-gradient de f en x ;
- $\forall y \in H \quad f(y) \geqslant f(x) + \langle p, y - x\rangle$;
- $\forall h \in H \quad f'(x; h) \geqslant \langle p, h\rangle$;
- x est un minimiseur de la fonction $y \in H \mapsto f(y) - \langle p, y\rangle$;
- $f(x) + f^*(p) \geqslant \langle p, x\rangle$;
- $f(x) + f^*(p) = \langle p, x\rangle$.

En particulier,

$$f'(x; h) = \sup_{p\in\partial f(x)} \langle p, h\rangle$$

Dans les deux derniers items, en anticipant légèrement, on a utilisé la conjuguée convexe f^*, définie dans la section suivante.

2.4 Conjuguée convexe (transformée de Fenchel)

La fonction conjuguée, appelée aussi transformée de Fenchel, est utilisée pour calculer le sous-différentiel d'une fonction convexe, caractériser des problèmes duaux (voir plus loin dans ce cours), ou encore, en utilisant la biconjuguée, pour convexifier une fonction non convexe (voir ci-dessous).

Définition 5. *Soit H un espace de Hilbert. Soit $f : H \to \mathbb{R} \cup \{+\infty\}$ une fonction quelconque. La fonction conjuguée (ou transformée de Fenchel) f^* de f est définie par*

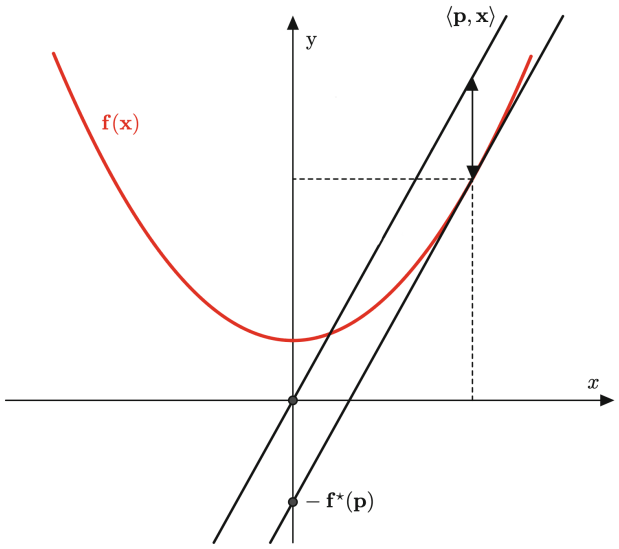
$$\boxed{f^*(p) = \sup_{x \in H} \left(\langle p, x \rangle - f(x) \right)} \qquad \forall p \in H$$

La conjuguée f^* est une fonction sur H à valeurs dans $\mathbb{R} \cup \{+\infty\}$. Son domaine est défini par $\mathrm{Dom}(f^*) = \{p \in H \ \mid \ f^*(p) < +\infty\}$.

De même que pour le sous-différentiel, la notion de conjuguée convexe se généralise lorsque H est un espace vectoriel topologique. Dans ce cas, la fonction f^* est définie sur H', et le produit scalaire ci-dessus est remplacé par le crochet de dualité.

Par définition, on a l'inégalité de Fenchel

$$\boxed{\forall x \in H \quad \forall p \in H \qquad \langle p, x \rangle \leqslant f(x) + f^*(p)}$$



Donnons deux interprétations géométriques du scalaire $f^*(p)$ (voir la figure).

Tout d'abord, par définition, $f^*(p)$ est le supremum de la différence verticale entre le graphe de f et le graphe de l'hyperplan vectoriel $x \mapsto \langle p, x \rangle$.

Ensuite, $p \in H$ étant fixé, par définition, on a $f(x) \geqslant \langle p, x \rangle - f^*(p)$ pour tout $x \in H$, ce qui signifie que le graphe de f est au-dessus de l'hyperplan affine $x \mapsto \langle p, x \rangle - f^*(p)$ (on parle de *minorante affine*) dont la normale est donnée par p (lorsque $H = \mathbb{R}$, comme sur la figure, p est une pente). Et alors, $f^*(p)$ est le plus petit des $\beta \in \mathbb{R} \cup \{+\infty\}$ tels que $x \mapsto \langle p, x \rangle - \beta$ est une minorante affine de f. En effet, on a

$$\langle p, x \rangle - \beta \leqslant f(x) \quad \forall x \in H \ \Leftrightarrow \ \beta \geqslant \langle p, x \rangle - f(x) \quad \forall x \in H \ \Leftrightarrow \ \beta \geqslant \sup_{x \in H} \left(\langle p, x \rangle - f(x) \right) = f^*(p).$$

Autrement dit, à p fixé, l'hyperplan affine $x \mapsto \langle p, x\rangle - f^*(p)$ est la plus grande minorante affine de f. Cet hyperplan a un point de contact avec le graphe de f (cf la proposition 1 ci-dessous).

Exemple 5. Pour $f(x) = \frac{1}{q}\|x\|^q$, on a $f^*(p) = \frac{1}{q'}\|x\|^{q'}$ pour $q \in]1, +\infty[$ et $\frac{1}{q} + \frac{1}{q'} = 1$. L'inégalité de Fenchel donne alors $\langle x, p\rangle \leqslant \frac{1}{q}\|x\|^q + \frac{1}{q'}\|p\|^{q'}$ pour tous $x, p \in H$, ce qui est l'inégalité de Young généralisée (bien connue pour $q = q' = 2$). Ainsi, l'inégalité de Fenchel permet d'obtenir de nouvelles inégalités.

La propriété suivante montre que les concepts de sous-différentiel et de transformée de Fenchel sont étroitement liés, et caractérise le cas d'égalité dans l'inégalité de Fenchel (on n'a pas besoin de supposer f convexe, à condition de garder toutefois la même définition d'un sous-gradient).

Proposition 1. *Pour $x \in \mathrm{Dom}(f)$, on a* $\boxed{p \in \partial f(x) \quad \Leftrightarrow \quad f(x) + f^*(p) = \langle p, x\rangle}$.

Démonstration. On a

$$\begin{aligned} p \in \partial f(x) \quad &\Leftrightarrow \quad \forall y \in H \quad f(y) \geqslant f(x) + \langle p, y - x\rangle \\ &\Leftrightarrow \quad \forall y \in H \quad \langle p, y\rangle - f(y) \leqslant \langle p, x\rangle - f(x) \quad \text{avec égalité pour } y = x \\ &\Leftrightarrow \quad \underbrace{\sup_{y \in H} \left(\langle p, y\rangle - f(y)\right)}_{f^*(p)} \leqslant \langle p, x\rangle - f(x) \end{aligned}$$

et on a égalité dans cette dernière (prendre $y = x$). □

Proposition 2. *Pour toute fonction $f : H \to \mathbb{R} \cup \{+\infty\}$ (pas forcément convexe), la transformée de Fenchel $f^* : H \to \mathbb{R} \cup \{+\infty\}$ est convexe sci.*

Démonstration. Par définition, $f^*(p)$ est un supremum de fonctions affines. Elle est donc convexe sci (exercice : le sup d'une famille de fonctions affines est toujours convexe sci ; de même l'inf d'une familles de fonctions affines est toujours concave scs). □

Ainsi, la transformation de Fenchel “convexifie".

Définition 6. *Soit $f : H \to \mathbb{R} \cup \{+\infty\}$ une fonction. Sa biconjuguée f^{**} est la conjuguée de f^*, i.e., est définie par*

$$\boxed{f^{**}(x) = \sup_{p \in H} \left(\langle p, x\rangle - f^*(p)\right)} \qquad \forall x \in H.$$

Comme f^*, la fonction f^{**} est une fonction sur H à valeurs dans $\mathbb{R} \cup \{+\infty\}$, et elle est convexe sci (comme supremum de fonctions affines, ou bien, parce que c'est une conjuguée).

Théorème 12. *(Fenchel-Moreau) On a toujours*

$$f^{**} \leqslant f$$

et en fait, la biconjuguée f^{**} est la plus grande fonction convexe sci inférieure ou égale à f. On a

$$\mathrm{Epi}(f^{**}) = \overline{\mathrm{Conv}}(\mathrm{Epi}(f)).$$

On dit que la fonction f^{**} est la convexifiée de la fonction f.

De plus, on a $f^{**} = f$ si et seulement si f est convexe sci.

On rappelle que $\overline{\mathrm{Conv}}(A)$ est l'enveloppe convexe fermée de A, autrement dit, le plus petit convexe fermé contenant A (voir remarque 12).

Démonstration. On définit l'ensemble $\mathscr{S}$ des minorantes affines de f :

$$\begin{aligned}\mathscr{S} &= \{(p,\beta) \in H \times (\mathbb{R} \cup \{+\infty\}) \ \mid \ \forall x \in H \quad f(x) \geqslant \langle p, x\rangle - \beta\} \\ &= \{(p,\beta) \in H \times (\mathbb{R} \cup \{+\infty\}) \ \mid \ \beta \geqslant \sup_{x \in H} (\langle p, x\rangle - f(x))\} \\ &= \{(p,\beta) \in H \times (\mathbb{R} \cup \{+\infty\}) \ \mid \ \beta \geqslant f^*(p)\}\end{aligned}$$

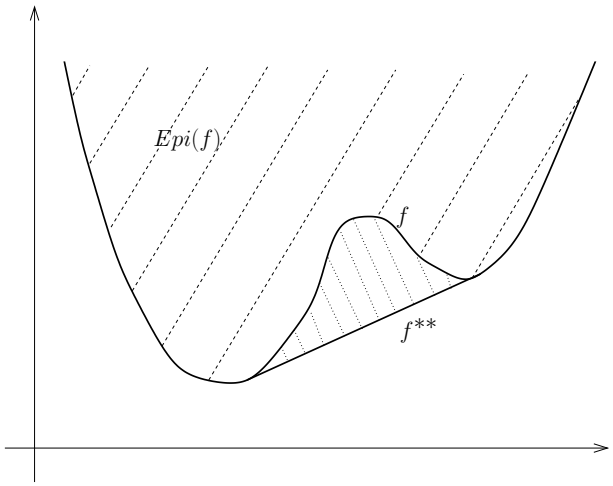
Pour tout $x \in H$ fixé, on a, par définition, $\langle p, x\rangle - \beta \leqslant f(x)$ pour tout $(p,\beta) \in \mathscr{S}$, donc en passant au sup, on a

$$\sup_{(p,\beta)\in\mathscr{S}} (\langle p, x\rangle - \beta) \leqslant f(x).$$

En fait, ce sup vaut exactement $f^{**}(x)$, ce qui montre donc que $f^{**}(x) \leqslant f(x)$. En effet :

$$\sup_{(p,\beta)\in\mathscr{S}} (\langle p, x\rangle - \beta) = \sup_{\substack{(p,\beta)\in H\times(\mathbb{R}\cup\{+\infty\}) \\ -\beta\leqslant -f^*(p)}} (\langle p, x\rangle - \beta) = \sup_{p\in H} (\langle p, x\rangle - f^*(p)) = f^{**}(x) \tag{2.3}$$

Pour montrer que f^{**} est la plus grande fonction convexe sci inférieure à f, il suffit de montrer la propriété sur les épigraphes. Pour tout $(p,\beta) \in \mathscr{S}$, comme $x \mapsto \langle p, x\rangle - \beta$ est une minorante affine de f, son épigraphe est un demi-espace fermé contenant $\mathrm{Epi}(f)$. Comme, d'après (2.3), la fonction convexe f^{**} est le sup sur $(p,\beta) \in \mathscr{S}$ de ces minorantes affines, son épigraphe $\mathrm{Epi}(f^{**})$ est l'intersection de tous ces demi-espaces fermés contenant $\mathrm{Epi}(f)$, autrement dit, c'est exactement $\overline{\mathrm{Conv}}(\mathrm{Epi}(f))$ (voir remarque (12)). □



Enfin, on a la propriété suivante, qui vient compléter la proposition 1.

Proposition 3. *Soit $f : H \to \mathbb{R}$ une fonction convexe sci. On a*

$$\forall x, p \in H \qquad p \in \partial f(x) \ \Leftrightarrow \ x \in \partial f^*(p)$$

(propriété d'échange).

Démonstration. On sait déjà, par la proposition 1, que $p \in \partial f(x) \ \Leftrightarrow \ f(x) + f^*(p) = \langle p, x\rangle$. Mais comme f est convexe sci, on a $f = f^{**}$, donc c'est aussi équivalent à $f^*(p) + f^{**}(x) = \langle p, x\rangle$. En appliquant la proposition 1 à f^*, cela est équivalent à $x \in \partial f^*(p)$. □

De même que pour la sous-différentiabilité, la théorie de Fenchel est très développée et a des applications importantes en optimisation. On imagine en effet très bien l'intérêt de convexifier une fonction non convexe ! Beaucoup d'autres développements (notamment algorithmiques) peuvent être vus dans des cours plus spécialisés.

Chapitre 3

Minimisation sans contraintes

Dans ce chapitre on étudie les problèmes de minimisation sans contrainte

$$\boxed{\min_{x\in E} f(x)}$$

où $f : E \to \mathbb{R} \cup \{+\infty\}$ est une fonction sur un espace vectoriel normé E.

En vue de l'implémentation numérique, c'est le cas $E = \mathbb{R}^n$ qui nous intéresse en priorité. Toutefois, on donnera des énoncés généraux chaque fois qu'on le pourra, en gardant les mêmes notations que dans les chapitres précédents : le plus souvent, E désigne un espace vectoriel normé et H un espace de Hilbert.

On dit que $x^* \in E$ est un *minimiseur global* de f si $f(x^*) \leqslant f(x)$ pour tout $x \in E$. On dit que $x^* \in E$ est un *minimiseur local* de f s'il existe un voisinage ouvert U de x^* tel que $f(x^*) \leqslant f(x)$ pour tout $x \in U$.

3.1 Existence et unicité

L'existence d'un minimiseur est triviale lorsque f est continue et qu'on considère un problème de minimisation avec contraintes $\min_{x\in C} f(x)$ où C est compact (car toute fonction continue sur un compact atteint son minimum). Lorsqu'il n'y a pas de contrainte, on a besoin d'ajouter des hypothèses à l'infini.

Définition 7. *On dit que f est infinie à l'infini si*

$$\boxed{\lim_{\|x\|\to+\infty} f(x) = +\infty}$$

On dit aussi parfois que f est propre (par définition, une application est propre si l'image réciproque de tout compact est un compact), mais comme on l'a vu le mot propre peut avoir plusieurs significations. On préfère ici utiliser le vocabulaire “infinie à l'infini".

Par exemple, $f(x) = \|x\|$ est infinie à l'infini. La fonction $f : \mathbb{R}^n \to \mathbb{R}$ définie par $f(x) = \frac{1}{2}x^\top Ax - b^\top b$ où A est une matrice symétrique définie positive est infinie à l'infini. Mais par exemple la fonction $f : \mathbb{R}^2 \to \mathbb{R}$ définie par $f(x) = x_1^2$ ne l'est pas.

Théorème 13. *Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction continue infinie à l'infini. Alors le problème*

$$\boxed{\min_{x\in\mathbb{R}^n} f(x)}$$

admet au moins un minimiseur (global), i.e., il existe au moins un point $x^* \in \mathbb{R}^n$ tel que $f(x^*) = \min_{x\in\mathbb{R}^n} f(x)$. On note aussi $x^* = \operatorname{argmin}_{x\in\mathbb{R}^n} f(x)$.

Démonstration. Soit $A > \inf_{x\in\mathbb{R}^n} f(x)$. Comme f est infinie à l'infini, il existe $R > 0$ tel que $f(x) \geqslant A$ pour tout $x \in \overline{B}(0,R) = \{x \in \mathbb{R}^n \mid \|x\| \geqslant R\}$. Cela implique que $\inf_{x\in\mathbb{R}^n} f(x) = \inf_{x\in\overline{B}(0,R)} f(x)$. Or, comme on est en dimension finie, $\overline{B}(0,R)$ est compacte, donc f, qui est continue, atteint son minimum. □

Il n'y a pas unicité du minimiseur en général. On peut aussi avoir des minimiseurs locaux qui ne sont pas globaux. Pour avoir unicité du minimiseur global, un bon moyen est de supposer la convexité.

Théorème 14. *Soit E un espace vectoriel. Si $f : E \to \mathbb{R}$ est strictement convexe alors elle a au plus un minimiseur (global).*

Démonstration. En effet, par l'absurde, s'il y avait deux minimiseurs x_1 et x_2 (i.e., $f(x_1) = f(x_2) = \min_{x\in E} f(x)$), alors, par convexité stricte, on aurait $f(\frac{x_1+x_2}{2}) < \frac{1}{2}(f(x_1) + f(x_2)) = \min_{x\in E} f(x)$ ce qui est absurde. □

Par contre, la convexité (même stricte) de f n'implique pas forcément l'existence d'un minimiseur. Par exemple, $f(x) = e^x$ est strictement convexe mais n'a pas de minimiseur. Son infimum sur $\mathbb{R}$ vaut 0 mais n'est pas atteint.

Corollaire 2. *Si $f : \mathbb{R}^n \to \mathbb{R}$ est continue, infinie à l'infini et strictement convexe, alors il existe un unique minimiseur global de f.*

Remarque 14. Si $f : \mathbb{R}^n \to \mathbb{R}$ est α-convexe (avec $\alpha > 0$) alors f est infinie à l'infini (appliquer le théorème 8), continue et strictement convexe, donc elle admet un unique minimiseur.

En pratique, lorsqu'on minimise une fonction f sur $\mathbb{R}^n$, il est rare que celle-ci soit convexe. Mais, souvent, on est capable de dire qu'elle est au moins dans une région donnée, et alors, dans cette région, on arrive à caractériser un unique minimiseur. C'est sur cette vision "locale" que se basent les algorithmes qu'on verra par la suite et qui sont basés sur des conditions nécessaires d'optimalité du premier ordre (conditions de dérivée nulle).

Obtenir numériquement un minimiseur global, alors qu'il existe plusieurs (éventuellement, beaucoup de) minimiseurs locaux, est difficile et fait appel à d'autres techniques ("optimisation globale") qui sortent du cadre de ce cours.

3.2 Conditions d'optimalité

3.2.1 Conditions nécessaires d'optimalité du premier ordre

Théorème 15. *Soit E un espace vectoriel normé. Soit $f : E \to \mathbb{R}$ une fonction Gateaux différentiable. Si $x^* \in E$ est un minimiseur (local ou global) de f alors $df(x^*) = 0$ (on devrait plutôt écrire : $f'(x;h) = 0$ pour tout $h \in E$).*

Lorsque E est un Hilbert et f est différentiable, on obtient $\nabla f(x^*) = 0$.

Démonstration. Comme x^* est un minimiseur (au moins local), pour tout $h \in E$, on a $f(x^*+th) \geqslant f(x^*)$ pour tout $t \in \mathbb{R}$ tel que $|t|$ est assez petit. En passant tout du même côté de l'inégalité, en divisant par $t \neq 0$ et en faisant tendre $t \to 0$, on obtient $f'(x;h) \geqslant 0$ pour tout $h \in E$, donc $f'(x;h) = 0$ par linéarité par rapport à h (appliquer l'inégalité à h et à $-h$). □

En général, un point qui annule la différentielle de f s'appelle un point extremum. Ainsi, tout minimiseur local est un point extremum.

Notons bien qu'il s'agit d'une condition nécessaire d'optimalité. Le fait que la dérivée de f s'annule en x^* n'implique pas que x^* soit un minimiseur (local) : ce serait être un maximiseur, ou bien, un extremum qui n'est ni un minimum ni un maximum (comme l'est 0 pour la fonction $f(x) = x^3$).

La condition est nécessaire et suffisante dans le cas convexe.

Théorème 16. *Soit $f : E \to \mathbb{R}$ une fonction convexe Gateaux différentiable. Alors $x^* \in E$ est un minimiseur global si et seulement si $df(x^*) = 0$.*

Plus généralement, on a déjà vu que si f est convexe et sous-différentiable, alors x^* est minimiseur global de f si et seulement si $0 \in \partial f(x^*)$.

Démonstration. Comme f est convexe, le graphe de f est au-dessus de ses tangentes : $f(x^* + h) \geqslant f(x^*) + df(x^*).h$ pour tout $h \in E$, donc si $df(x^*) = 0$ alors x^* est un minimiseur global de f. □

Remarque 15. Même pour f convexe, on n'a pas forcément unicité du minimiseur global : la fonction f pourrait en effet être constante au voisinage de x^*.

Lorsque f est convexe, l'ensemble des minimiseurs (globaux) est un sous-ensemble convexe de E. En effet, si x et y sont deux minimiseurs de f, i.e., $f(x) = f(y) = \min f$, alors pour tout $\lambda \in [0, 1]$, $f(\lambda x + (1-\lambda)y) \leqslant \lambda f(x) + (1-\lambda) f(y) = \min f$ donc en fait on a égalité et $\lambda x + (1-\lambda)y$ est aussi un minimiseur.

3.2.2 Conditions nécessaires et/ou suffisantes d'optimalité du deuxième ordre

Théorème 17. *Soit E un espace vectoriel normé, soit $x^* \in E$ et soit $f : E \to \mathbb{R}$ une fonction deux fois différentiable en x^*.*

— *Condition nécessaire du deuxième ordre : si x^* est un minimiseur (local ou global) de f alors $df(x^*) = 0$ et $d^2 f(x^*) \geqslant 0$, i.e., la Hessienne de f en x^* est une forme quadratique positive :*

$$\forall h \in E \qquad d^2 f(x^*).(h, h) \geqslant 0.$$

— *Condition suffisante du deuxième ordre : si $df(x^*) = 0$ et si la Hessienne de f en x^* est une forme quadratique vérifiant*

$$\exists \alpha > 0 \ \mid \ \forall h \in E \qquad d^2 f(x^*).(h, h) \geqslant \alpha \|h\|^2$$

(on dit que $d^2 f(x^*)$ est une forme quadratique coercive, ou α-elliptique) alors x^* est un minimiseur local strict de f.

Minimiseur local strict signifie que $f(x^* + h) > f(x^*)$ pour tout $h \neq 0$ de norme assez petite.

Démonstration. On se ramène au cas 1D en posant $\varphi(t) = f(x^* + th)$, pour $h \in E$ et $t \in \mathbb{R}$. Si x^* est un minimiseur de f, on a $\varphi(t) \geqslant \varphi(0)$, au moins dans un voisinage de $t = 0$. En appliquant la formule de Taylor à l'ordre deux, on a $\varphi(t) = \varphi(0) + t\varphi'(0) + \frac{1}{2}t^2\varphi''(0) + \mathrm{o}(t^2)$, lorsque $t \to 0$. Or, $\varphi'(0) = df(x^*).h = 0$ (on l'a déjà vu) et $\varphi''(0) = d^2 f(x^*).(h, h)$. On obtient donc $\frac{1}{2}t^2 d^2 f(x^*).(h, h) + \mathrm{o}(t^2) \geqslant 0$, et en divisant par t^2 et en faisant tendre $t \to 0$, on obtient le résultat.

Montrons maintenant la condition suffisante. On fait un développement de Taylor de f au second ordre en x^*, et comme $df(x^*) = 0$, on a

$$f(x^* + h) = f(x^*) + \frac{1}{2} d^2 f(x^*).(h, h) + \mathrm{o}(\|h\|^2)$$

lorsque $h \to 0$. Comme $d^2 f(x^*)$ est α-elliptique, il existe $\varepsilon > 0$ petit tel que pour tout $h \in E$ vérifiant $\|h\| \leqslant \varepsilon$, $\frac{1}{2} d^2 f(x^*).(h, h) + \mathrm{o}(\|h\|^2) \geqslant \frac{\alpha}{2} \|h\|^2$: autrement dit, pour $\|h\|$ assez petit on peut absorber le terme en o. On obtient alors

$$f(x^* + h) \geqslant f(x^*) + \frac{\alpha}{2} \|h\|^2$$

pour tout $h \in E$ tel que $\|h\| \leqslant \varepsilon$. Donc x^* est un minimiseur local strict de f. □

En fait, dans la condition suffisante, on a obtenu un résultat plus fort : non seulement le minimiseur local est strict, mais de plus, la différence $f(x^* + h) - f(x^*)$ est au moins quadratique.

Remarque 16. Rappelons que, lorsque $E = \mathbb{R}^n$, la Hessienne de f en x^* est la matrice symétrique réelle $H_f(x^*)$ (carrée d'ordre n) formée par les dérivées partielles $\frac{\partial^2 f}{\partial x_i \partial x_j}(x^*)$, $i, j = 1, \ldots, n$. On rappelle que $H_f(x^*)$ est positive si

$$y^\top H_f(x^*) y \geqslant 0 \qquad \forall y \in \mathbb{R}^n,$$

et $H_f(x^*)$ est définie positive si elle est positive et de plus

$$y^\top H_f(x^*) y = 0 \;\Rightarrow\; y = 0.$$

En dimension finie, il est toujours vrai que si $H_f(x^*)$ est définie positive alors elle est α-elliptique, pour un $\alpha > 0$. En effet, la matrice $H_f(x^*)$ est symétrique réelle, donc elle est diagonalisable en base orthonormée. En notant $\lambda_1 \geqslant \cdots \geqslant \lambda_n$ ses valeurs propres (qui sont réelles), on obtient donc

$$\forall y \in \mathbb{R}^n \qquad y^\top H_f(x^*) y \geqslant \lambda_n \|y\|^2.$$

Ainsi, si $H_f(x^*)$ est définie positive, alors $\lambda_n > 0$ et on conclut que $H_f(x^*)$ est α-elliptique avec $\alpha = \lambda_n = \min \mathrm{Spec}(H_f(x^*))$.

En dimension finie, la condition suffisante du deuxième ordre peut donc s'écrire : si $\nabla f(x^*) = 0$ et si $H_f(x^*) > 0$ alors x^* est un minimiseur local strict de f.

Mais l'implication "définie positive $\Rightarrow$ α-elliptique" n'est pas vraie en dimension infinie, et c'est pourquoi, dans le théorème, on a supposé que $d^2 f(x^*)$ est α-elliptique. En effet, il suffit d'imaginer une Hessienne de taille infinie, et telle que, même si elle est positive, ses valeurs propres forment une suite de réels strictement positifs décroissant vers 0. La conclusion du théorème peut alors être mise en défaut.

Remarque 17. Même en dimension finie, si la Hessienne est seulement positive (mais pas α-elliptique), la conclusion de minimiseur local est fausse. Par exemple, la fonction $f : \mathbb{R}^2 \to \mathbb{R}$ définie par

$$f(x, y) = x^2 - y^4$$

est telle que

$$\nabla f(0, 0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad H_f(0, 0) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

(donc, la Hessienne est positive mais pas définie positive) et $(0, 0)$ est un point extremum qui n'est ni un minimiseur ni un maximiseur. En effet, $f(x, 0) = x^2$ donc dans la direction x on a un minimum à l'origine. Mais par ailleurs, $f(0, y) = -y^4$ donc dans la direction y on a un maximum à l'origine.

Remarque 18. Par le théorème 8, la condition d'α-ellipticité de la Hessienne en x^* implique en fait que, dans un voisinage de x^*, la fonction f est α-convexe. C'est la raison pour laquelle on trouve que non seulement x^* est un minimiseur local strict, mais que, de plus, $f(x^*+h)-f(x^*)$ est au moins quadratique en h (pour $\|h\|$ petit).

Remarque 19. Les conditions obtenues ci-dessus sont locales, mais bien entendu, elles deviennent globales si f est de plus convexe sur E.

Exemple 6. Un exemple important, déjà vu, est le cas de la fonction $f:\mathbb{R}^n\to\mathbb{R}$ définie par

$$f(x)=\frac{1}{2}x^\top Ax-b^\top x$$

où $A\in\mathcal{M}_n(\mathbb{R})$ est une matrice symétrique réelle définie positive et $b\in\mathbb{R}^n$. L'unique minimiseur $x^*\in\mathbb{R}^n$ est déterminé par la condition $\nabla f(x^*)=0$, i.e., $Ax^*=b$, donc, $x^*=A^{-1}b$.

3.3 Problème des moindres carrés

3.3.1 Définition et résolution

Le problème des moindres carrés est certainement l'un des problèmes les plus importants en analyse numérique, avec un nombre immense d'applications et de variantes. L'origine de ce problème est la recherche de solutions d'un système linéaire $Ax=b$ dans lequel la matrice A est non inversible ou non carrée. Le problème des moindres carrés consiste alors à chercher la ou les solutions du problème de minimisation sans contrainte

$$\boxed{\min_{x\in\mathbb{R}^p}\|Ax-b\|^2}$$

où $A\in\mathcal{M}_{n,p}(\mathbb{R})$ est une matrice réelle ayant n lignes et p colonnes, et $b\in\mathbb{R}^n$. Dans toute cette section, $\|\ \|$ est la norme euclidienne de $\mathbb{R}^n$.

Il s'agit d'un problème de minimisation sans contrainte, pour la fonction $f:\mathbb{R}^p\to\mathbb{R}$ définie par

$$f(x)=\|Ax-b\|^2.$$

La fonction f est de classe C^∞, et en développant $f(x+h)=f(x)+2\langle Ax-b,Ah\rangle+\|Ah\|^2$, on trouve

$$\nabla f(x)=2A^\top(Ax-b)\qquad\text{et}\qquad H_f(x)=2A^\top A.$$

On voit aussi que f est convexe puisque $H_f(x)\geqslant 0$ (en effet, $h^\top H_f(x)h=2\|Ah\|^2$), mais elle n'est pas forcément strictement convexe. La fonction f n'est pas forcément infinie à l'infini (en effet on peut avoir $\ker A\neq\{0\}$). Malgré cela, il est remarquable que le problème des moindres carrés a toujours au moins une solution, comme l'affirme le résultat ci-dessous.

Théorème 18. *Le problème des moindres carrés admet au moins une solution. Les solutions du problème des moindres carrés sont exactement les solutions de "l'équation normale"*

$$A^\top Ax=A^\top b.$$

De plus, si $\ker A=\{0\}$ *(i.e., si A est injective) alors il existe une unique solution $x\in\mathbb{R}^p$, qui est*

$$\boxed{x=A^\#b=(A^\top A)^{-1}A^\top b}$$

Démonstration. Comme f est convexe, f a un minimiseur x si et seulement si $\nabla f(x) = 0$, i.e., $A^\top Ax = A^\top b$. Cela montre que les solutions du problème des moindres carrés, si elles existent, sont exactement les solutions de l'équation normale.

Notons que $\ker A = \{0\} \Leftrightarrow A^\top A$ inversible $\Leftrightarrow A^\top A > 0$ (en effet, il suffit de remarquer que $x^\top A^\top Ax = \|Ax\|^2$ pour tout $x \in \mathbb{R}^p$). Sous cette condition, on a $H_f(x) = 2A^\top A > 0$, donc f est strictement convexe et donc le minimiseur, s'il existe, est unique.

Il reste à montrer que l'équation normale a toujours au moins une solution. Pour cela, il suffit de montrer que $\operatorname{Im} A^\top \subset \operatorname{Im} A^\top A$. Or, cela est équivalent à $\ker A^\top A \subset \ker A$ (en effet, on a $(\operatorname{Im} A^\top)^\perp = \ker A$ et $(\operatorname{Im} A^\top A)^\perp = \ker A^\top A$), inclusion qui est vraie car si $x \in \ker A^\top A$ alors $A^\top Ax = 0$ donc $\|Ax\|^2 = x^\top A^\top Ax = 0$. En fait, on a $\ker A = \ker A^\top A$. □

Dans ce théorème, lorsque $\ker A = \{0\}$, on a introduit ce qu'on appelle la *pseudo-inverse* (de Moore-Penrose) de la matrice A :

$$\boxed{A^\# = (A^\top A)^{-1} A^\top}$$

On a $A^\# \in \mathcal{M}_{p,n}(\mathbb{R})$. Lorsque A est carrée inversible, on a $A^\# = A^{-1}$. Dans les autres cas, en supposant A injective, la pseudo-inverse généralise l'inverse de A. Le cas typique est lorsqu'on cherche résoudre un système de n équations à p inconnues

$$Ax = b$$

avec $n > p$ (il y a plus d'équations que d'inconnues) et avec $\ker A = \{0\}$ (il n'y a pas d'équation redondante : les contraintes sont indépendantes) : un tel système n'admet évidemment pas de solution ! En pratique, il y a quantité de problèmes qui n'admettent pas de solutions (car on impose trop de contraintes indépendantes et on n'a pas assez de degrés de liberté), mais on n'aime pas trop cela... et on cherche tout de même "le meilleur compromis possible", i.e., on cherche à résoudre

$$Ax \simeq b$$

et mathématiquement il est alors raisonnable de chercher la solution x qui minimise la norme de l'écart $Ax - b$. On vient de voir que cette solution, unique, est donnée par $x = A^\# b$ où $A^\#$ est la pseudo-inverse. La solution x réalise le meilleur compromis possible, au sens du problème des moindres carrés. C'est pourquoi ce problème a une telle importance en pratique (même en politique !).

Remarque 20. Ci-dessus, on a considéré la pseudo-inverse d'une matrice ayant plus de lignes que de colonnes. On peut aussi considérer le cas où A a plus de colonnes que de lignes, i.e., $p > n$, et de plus, $\operatorname{rg}(A) = p$ (A surjective). Ce cas est aussi d'intérêt, car cette fois, le système linéaire $Ax = b$ a une infinité de solutions. Mais, parmi cette infinité de solutions, on peut chercher celle qui est de plus petite norme euclidienne. En refaisant la théorie développée ci-dessus, on trouve $x = A^\# b$ où la pseudo-inverse de A est, cette fois, définie un peu différemment, par $A^\# = A^\top (AA^\top)^{-1}$.

Remarque 21. De manière plus générale, on peut définir la pseudo-inverse $A^\#$ de A sans aucune hypothèse sur A : étant donné une matrice $A \in \mathcal{M}_{n,p}(\mathbb{R})$, $A^\#$ est l'unique matrice de $\mathcal{M}_{p,n}(\mathbb{R})$ telle que

$$AA^\# A = A, \quad A^\# AA^\# = A^\#, \quad (AA^\#)^\top = AA^\#, \quad (A^\# A)^\top = A^\# A.$$

De plus,

$$A^\# = \lim_{\varepsilon \to 0} \left(A^\top A + \varepsilon I_p\right)^{-1} A^\top = \lim_{\varepsilon \to 0} A^\top \left(AA^\top + \varepsilon I_n\right)^{-1}.$$

Remarque 22 (Résolution numérique)**.** Du fait de l'importance considérable du problème des moindres carrés, il est important de disposer d'algorithmes efficaces de résolution, notamment en grande dimension.

Lorsque $\ker A = \{0\}$, on peut résoudre l'équation normale $AA^\top x = A^\top b$ par la méthode de Cholesky, la matrice $AA^\top$ étant symétrique définie positive (et donc, elle s'écrit sous la forme $LL^\top$ avec L triangulaire inférieure).

Une autre méthode d'analyse numérique matricielle est la méthode QR : la décomposition QR de la matrice A consiste à écrire $A = QR$ où $Q \in \mathcal{M}_n(\mathbb{R})$ est une matrice orthogonale et $R \in \mathcal{M}_{n,p}(\mathbb{R})$ est une matrice triangulaire supérieure. Le problème des moindres carrés est alors équivalent à

$$\min_{x \in \mathbb{R}^p} \|Rx - Q^\top b\|^2$$

dont l'unique solution (dans le cas $n \geqslant p$, $\ker A = \{0\}$) est

$$x = (R_{1 \leqslant i,j \leqslant p})^{-1} (Q^\top b)_{1 \leqslant i \leqslant p}$$

Mais, certainement, la méthode la plus efficace est de faire une décomposition SVD de la matrice A. Etant donné l'importance de la décomposition SVD, et du fait qu'elle n'est hélas généralement pas traitée en licence, cela vaut le coup d'y consacrer une section.

3.3.2 Décomposition en valeurs singulières (SVD)

SVD est le sigle anglais signifiant “Singular Value Decomposition".

Soit $A \in \mathcal{M}_{n,p}(\mathbb{R})$. On commence par noter que $A^\top A$ est une matrice symétrique réelle positive de taille p, donc diagonalisable en base orthonormée, et ses n valeurs propres sont réelles et positives.

Définition 8. *Les valeurs singulières d'une matrice $A \in \mathcal{M}_{n,p}(\mathbb{R})$ sont les racines carrées des valeurs propres de $A^\top A$.*

Par convention, les valeurs singulières $\sigma_i \geqslant 0$, $i = 1, \ldots, p$ de A sont classées par ordre décroissant :

$$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0$$

et on note r le nombre de valeurs singulières strictement positives. On a $r = \mathrm{rg}(A)$.

Théorème 19 (Décomposition SVD)**.** *Soit $A \in \mathcal{M}_{n,p}(\mathbb{R})$ une matrice ayant r valeurs singulières strictement positives. Il existe deux matrices orthogonales $U \in \mathcal{M}_p(\mathbb{R})$ et $V \in \mathcal{M}_n(\mathbb{R})$ telles que*

$$A = V \Sigma U^\top \qquad \textit{avec} \qquad \Sigma = \left(\begin{array}{ccc|c} \sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_r & \\ \hline & 0 & & 0 \end{array}\right) \in \mathcal{M}_{n,p}(\mathbb{R})$$

Remarque 23. Comme on l'a déjà remarqué, on a $\mathrm{rg}(A) = r$, le nombre de valeurs singulières non nulles. Par ailleurs, on a (d'après la remarque 21)

$$A^\# = U \Sigma^\# V^\top \qquad \text{avec} \qquad \Sigma^\# = \left(\begin{array}{ccc|c} 1/\sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & 1/\sigma_r & \\ \hline & 0 & & 0 \end{array}\right) \in \mathcal{M}_{n,p}(\mathbb{R})$$

Connaissant la décomposition SVD de A, il est donc très facile de calculer $A^\#$.

Remarque 24. Comme $A = V\Sigma U^\top$, premièrement, on a $A^\top A = U\Sigma^\top \Sigma U^\top$ (avec $U^\top = U^{-1}$), donc la matrice U est exactement la matrice de passage qui diagonalise la matrice symétrique réelle $A^\top A$, elle est donc composée par les vecteurs propres de $A^\top A$. La matrice carrée $\Sigma^\top \Sigma$ est diagonale, et les éléments de la diagonale sont exactement les valeurs propres de $A^\top A$ (carrés des valeurs singulières).

Deuxièmement, on a $AA^\top = V\Sigma\Sigma^\top V^\top$ (avec $V^\top = V^{-1}$), donc la matrice V est exactement la matrice de passage qui diagonalise la matrice symétrique réelle $AA^\top$, elle est donc composée par les vecteurs propres de $AA^\top$. La matrice carrée $\Sigma\Sigma^\top$ est diagonale, et les éléments de la diagonale sont exactement les valeurs propres de $AA^\top$ (carrés des valeurs singulières) : rien d'étonnant car en fait les valeurs propres non nulles de $A^\top A$ et de $AA^\top$ sont les mêmes.

Cette remarque donne presque la preuve du théorème.

Démonstration. Montrons le théorème pour $n \geqslant p$ (pour $n \leqslant p$, il suffit d'appliquer la décomposition SVD à $A^\top$). On note u_i les vecteurs propres de la matrice symétrique réelle $A^\top A$, associés aux valeurs propres σ_i^2, $i = 1, \dots, p$. On définit la matrice orthogonale $U \in \mathcal{M}_p(\mathbb{R})$ dont les colonnes sont les u_i. On a alors exactement $A^\top A = U\Sigma^\top \Sigma U^\top$.

En particulier, cette égalité donne $\langle Au_j, Au_i \rangle = u_j^\top A^\top A u_i = \sigma_i^2 \delta_{i,j}$ (avec $\delta_{i,j} = 1$ si $i = j$ et 0 sinon) pour $i = 1, \dots, p$; notons que $Au_i = 0$ pour $i = r+1, \dots, p$ (car $\sigma_i = 0$). Donc, en posant $v_i = \frac{1}{\sigma_i} Au_i$ pour $i = 1, \dots, r$, la famille $(v_1, \dots, v_r)$ est orthonormée. On la complète en une base orthonormée $(v_1, \dots, v_n)$ de $\mathbb{R}^n$ (de manière quelconque) et on définit la matrice orthogonale $V \in \mathcal{M}_n(\mathbb{R})$ dont les colonnes sont les v_i. On a $Au_i = \sigma_i v_i$ pour $i = 1, \dots, p$. Cela donne exactement $AU = V\Sigma$. □

Remarque 25. On peut aussi montrer que

- $\operatorname{Im} A = \operatorname{Vect}(v_1, \dots, v_r)$.
- $\ker A = \operatorname{Vect}(u_{r+1}, \dots, u_n)$.
- $v_i = \frac{1}{\sigma_i} Au_i$ pour $i = 1, \dots, r$, et $u_i = \frac{1}{\sigma_i} A^\top v_i$ pour $i = 1, \dots, r$.
- $\|A\|_F = \sqrt{\sum_{i=1}^r \sigma_i^2}$ (norme Frobenius de A), $\|A\|_2 = \sigma_1$, et $\min \frac{\|Ax\|_2}{\|x\|_2} = \sigma_p$ $(n \geqslant p)$.
- Géométriquement, l'image par A de la sphère S^{n-1} est un ellipsoïde de demi-axes $\sigma_i e_i$.
- Pour tout k, on pose $U_k = (u_1, \dots, u_k)$ (matrice de taille $p \times k$) et $V_k = (v_1, \dots, v_k)$ (matrice de taille $n \times k$). Alors :
  - $U_k^\top U_k = I_k$, $V_k^\top V_k = I_k$.
  - $U_k U_k^\top = \sum_{i=1}^k u_i u_i^\top$ est le projecteur orthogonal (car symétrique) sur $\operatorname{Vect}(u_1, \dots, u_k)$.
  - $V_k V_k^\top = \sum_{i=1}^k v_i v_i^\top$ est le projecteur orthogonal (car symétrique) sur $\operatorname{Vect}(v_1, \dots, v_k)$.
  - En particulier, en posant $U = (U_r, \tilde{U}_r)$ et $V = (V_r, \tilde{V}_r)$, on a :
    $V_r V_r^\top$ est le projecteur orthogonal sur $\operatorname{Im} A$.
    $\tilde{V}_r \tilde{V}_r^\top$ est le projecteur orthogonal sur $(\operatorname{Im} A)^\perp = \ker A^\top$.
    $U_r U_r^\top$ est le projecteur orthogonal sur $(\ker A)^\perp = \operatorname{Im} A^\top$.
    $\tilde{U}_r \tilde{U}_r^\top$ est le projecteur orthogonal sur $\ker A$.

Le théorème de décomposition SVD montre que, à transformations orthogonales près, toute matrice peut s'écrire sous forme "diagonale" ! Ce peut être vu comme une réduction de A à la matrice Σ contenant les valeurs singulières de A.

En ayant en tête que $r = \operatorname{rg}(A)$, une application très importante de la SVD est de donner des approximations de A de rang petit. Supposons que n et p soient très grands (c'est le cas si A est la matrice d'une image par exemple, comportant $n \times p$ pixels) et, pour k donné, vérifiant

$k \leqslant \min(n,p)$, cherchons une matrice A_k de rang k qui soit une "bonne approximation" de la matrice A. La matrice

$$A_k = V\Sigma_k U^\top \qquad \text{avec} \qquad \Sigma_k = \left(\begin{array}{ccc|c} \sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_k & \\ \hline & 0 & & 0 \\ & & & \end{array}\right) \in \mathcal{M}_{n,p}(\mathbb{R})$$

est la meilleure approximation de A parmi les matrices de rang k (on n'a pas forcément unicité) au sens de la norme $\|\ \|_2$ (qui est la norme matricielle subordonnée à la norme euclidienne), et on a $\|A - A_k\|_2 = \sigma_{k+1}$. Si A est la matrice d'une image, son approximation de rang k est une compression de l'image. Amusez-vous à coder cela (en Matlab, Scilab ou Python), et prenez diverses valeurs de k. Vous constaterez que, même pour k petit on arrive à avoir une image de bonne qualité.

Avant l'invention des normes jpeg, la décomposition SVD a longtemps servi pour la compression d'image. Elle est désormais supplantée, en efficacité, par les ondelettes.

3.3.3 Application à la régression

De manière générale, le problème de régression (en statistiques) est le suivant. On dispose de mesures

$$(t_i, b_i),\ i = 1, \ldots, n$$

avec $t_i \in \mathbb{R}$ (tous distincts), $b_i \in \mathbb{R}$, et généralement n est grand. Soit $p \in \mathbb{N}^*$ (généralement, assez petit). L'objectif est de trouver le "meilleur polynôme" $P \in \mathbb{R}_{p-1}[X]$, de degré $p-1$, tel que

$$P(t_i) \simeq b_i,\ i = 1, \ldots, n.$$

Bien entendu, lorsque $n \leqslant p$, on peut trouver un polynôme qui réalise exactement toutes les égalités (on utilise la théorie de l'interpolation de Lagrange). Mais, ici, on a plutôt $p \ll n$ et les égalités ne peuvent pas toutes être satisfaites. On cherche donc un meilleur compromis, en cherchant le polynôme $P \in \mathbb{R}_{p-1}[X]$ qui minimise

$$f(P) = \sum_{i=1}^{n} (P(t_i) - b_i)^2.$$

Il s'agit d'un problème de moindres carrés. En effet, soit $(\phi_j)_{0 \leqslant j \leqslant p-1}$ une base de $\mathbb{R}_{p-1}[X]$ (on n'est pas obligé de considérer la base canonique). On cherche

$$P = \sum_{j=0}^{p-1} a_j \phi_j.$$

En posant

$$x = \begin{pmatrix} a_0 \\ \vdots \\ a_{p-1} \end{pmatrix} \in \mathbb{R}^p, \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^n, \qquad A = \Big(\phi_j(t_i)\Big)_{\substack{1 \leqslant i \leqslant n \\ 0 \leqslant j \leqslant p-1}} \in \mathcal{M}_{n,p}(\mathbb{R})$$

et en identifiant $\mathbb{R}_{p-1}[X] \simeq \mathbb{R}^p$, on a exactement

$$f(x) = \|Ax - b\|^2.$$

La matrice A est injective car les t_i sont tous distincts. Ce problème de moindres carrés admet donc une unique solution, qui est $x = A^{\#}b$.

Exemple 7. Prenons $p = 2$, $\phi_0 = 1$ et $\phi_1 = X$ (base canonique), et retrouvons les formules connues de régression linéaire. On a ici

$$A = \begin{pmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_n \end{pmatrix}, \qquad A^\top = \begin{pmatrix} 1 & \cdots & 1 \\ t_1 & \cdots & t_n \end{pmatrix}, \qquad A^\top A = \begin{pmatrix} n & \sum_{i=1}^n t_i \\ \sum_{i=1}^n t_i & \sum_{i=1}^n t_i^2 \end{pmatrix}, \qquad A^\top b = \begin{pmatrix} \sum_{i=1}^n b_i \\ \sum_{i=1}^n t_i b_i \end{pmatrix}$$

et comme $x = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = (AA^\top)^{-1}A^\top b$, on trouve

$$a_0 = \frac{\sum_{i=1}^n t_i^2 \sum_{i=1}^n b_i - \sum_{i=1}^n t_i \sum_{i=1}^n t_i b_i}{n\sum_{i=1}^n t_i^2 - \left(\sum_{i=1}^n t_i\right)^2}, \qquad a_1 = \frac{-\sum_{i=1}^n t_i \sum_{i=1}^n b_i + n\sum_{i=1}^n t_i b_i}{n\sum_{i=1}^n t_i^2 - \left(\sum_{i=1}^n t_i\right)^2}$$

3.4 Algorithmes d'optimisation sans contraintes

Dans cette section, on donne des algorithmes de résolution numérique du problème de minimisation sans contrainte

$$\boxed{\min_{x\in\mathbb{R}^n} f(x)}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est une fonction qui est au moins de classe C^1.

On suppose qu'il existe un minimiseur $x^* \in \mathbb{R}^n$ (au moins local) de f. On a donc $\nabla f(x^*) = 0$.

3.4.1 Méthodes de descente

3.4.1.1 Principe des méthodes de descente

Le principe d'une méthode de descente est de faire, à partir d'un point initial $x_0 \in \mathbb{R}^n$, les itérations

$$\boxed{x_{k+1} = x_k + \rho_k d_k, \quad \rho_k > 0, \quad d_k \in \mathbb{R}^n}$$

en assurant que

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

(en tout cas, tant que $x_k \neq x^*$). Le vecteur d_k est appelé direction de descente et le réel $\rho_k > 0$ est appelé pas de descente à l'itération k. Généralement, on choisit ρ_k assez petit (au moins asymptotiquement), ce qui incite à considérer le développement limité au premier ordre

$$f(x_{k+1}) = f(x_k + \rho_k d_k) = f(x_k) + \rho_k \langle \nabla f(x_k), d_k\rangle + \mathrm{o}(\rho_k).$$

Cela nous amène naturellement à choisir $\rho_k > 0$ petit et une direction de descente $d_k \in \mathbb{R}^n$ vérifiant

$$\boxed{\langle \nabla f(x_k), d_k\rangle < 0}$$

par exemple, $d_k = -\nabla f(x_k)$. Avec un tel choix, si ρ_k est assez petit alors on a bien $f(x_{k+1}) < f(x_k)$, et on peut alors espérer que, lorsque $k \to +\infty$, on ait $x_k \to x^*$, à condition, bien sûr, d'avoir choisi x_0 pas trop loin de x^* (car, si f n'est pas convexe, elle peut avoir d'autres minimiseurs locaux).

Une façon de choisir le pas $\rho_k > 0$ est de minimiser la fonction $\varphi(t) = f(x_k + td_k)$:

$$\rho_k = \underset{t>0}{\operatorname{argmin}}\, \varphi(t)$$

Un tel minimum existe par exemple si f est infinie à l'infini : en effet, d'une part, $\varphi'(0) = df(x_k).d_k = \langle \nabla f(x_k), d_k \rangle < 0$, donc φ décroît à partir de $t = 0$, au moins sur un petit intervalle $[0, \varepsilon]$, et d'autre part, $\varphi(t) \to +\infty$ lorsque $t \to +\infty$, donc φ a un minimum. En un tel minimiseur ρ_k, on doit avoir

$$\varphi'(\rho_k) = \langle \nabla f(x_k + \rho_k d_k), d_k \rangle = 0$$

ce qui nous conduira, un peu plus loin, à définir la méthode de gradient à pas optimal.

Sans chercher, pour le moment, un pas optimal, on peut au moins choisir un pas $\rho_k = \rho > 0$ fixe, assez petit. Cela donne la méthode du gradient à pas fixe, qu'on va analyser ci-après. On peut aussi choisir des pas variables.

On peut choisir des directions de descente autres que $d_k = -\nabla f(x_k)$. Par exemple, dans la méthode de gradient conjugué, on construira une suite de directions de descente, ayant certaines propriétés d'orthogonalité. Cette construction astucieuse assure une convergence en nombre fini d'itérations pour des fonctions quadratiques.

Les variantes sont nombreuses, on va en voir et en analyser quelques-unes.

De manière générale, l'algorithme d'une méthode de descente, ainsi que définie ci-dessus, est le suivant :

1. Initialisation, $k = 0$: choisir $x_0 \in \mathbb{R}^n$ et $\varepsilon > 0$.
2. Itération k : ayant choisi un pas ρ_k et une direction de descente d_k, on calcule

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

3. Critère d'arrêt : on stoppe les itérations lorsque

$$\|\nabla f(x_k)\| \leqslant \varepsilon$$

c'est-à-dire, lorsque le gradient de f est suffisamment petit ; ou bien (variante), lorsque

$$\|x_{k+1} - x_k\| \leqslant \varepsilon$$

c'est-à-dire, lorsque les itérés ne progressent plus suffisamment ; ou bien (variante), lorsque

$$\|f(x_{k+1}) - f(x_k)\| \leqslant \varepsilon$$

c'est-à-dire, lorsque les valeurs de f ne changent plus suffisamment.

Le choix du critère d'arrêt est important. Il sera testé et discuté lors des séances de TP.

3.4.1.2 Méthode de gradient (à pas variable ou fixe)

Par définition, une méthode de gradient est une méthode de descente dans laquelle

$$d_k = -\nabla f(x_k) \qquad \forall k \in \mathbb{N}.$$

On a le résultat général de convergence suivant.

Théorème 20. *Soit U un ouvert convexe borné de $\mathbb{R}^n$ sur lequel f a un unique minimiseur x^*. On suppose que, sur U, la fonction f est de classe C^1, strictement convexe*[1]*, et que f est à gradient Lipschitz*[2]*, i.e., qu'il existe $M > 0$ tel que*

$$\forall x, y \in U \qquad \|\nabla f(x) - \nabla f(y)\| \leqslant M\|x - y\|. \tag{3.1}$$

Soient $0 < \beta_1 < \beta_2 < \frac{2}{M}$. Soit $\eta > 0$ tel que $V = \{x \in U \mid f(x) \leqslant f(x^*) + \eta\} \subset U$. Si, pour tout $k \in \mathbb{N}$, le pas ρ_k vérifie $\rho_k \in [\beta_1, \beta_2]$, alors la méthode de gradient, initialisée à un $x_0 \in V$ arbitraire, converge vers x^* : la suite $(x_k)_{k\in\mathbb{N}}$ définie par

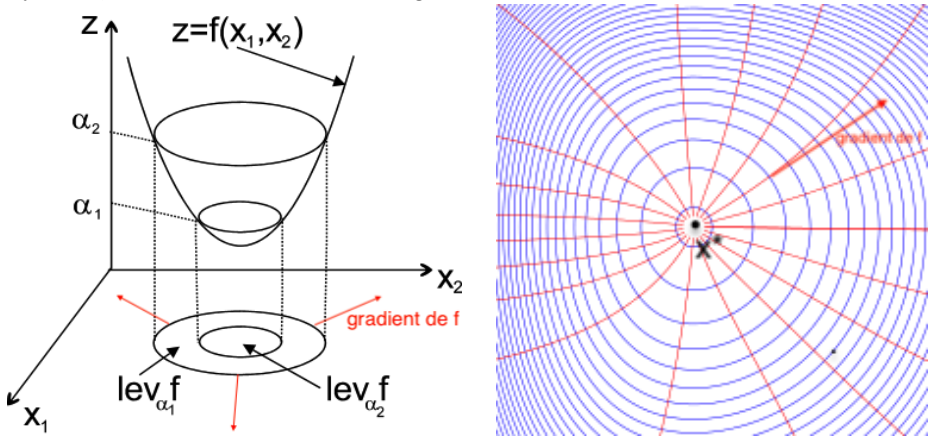
$$x_{k+1} = x_k - \rho_k \nabla f(x_k), \quad x_0 \in V$$

reste dans V et converge vers x^*.

On appelle cette méthode la méthode de gradient à pas variable. Le choix du pas à chaque itération peut être fait selon différents critères (on en verra plus loin). Lorsque le pas $\rho_k = \rho$ est fixe, on parle de méthode de gradient à pas fixe.

Bien entendu, en pratique on ne connait pas à l'avance un ouvert convexe U sur lequel f aurait un unique minimiseur. La recherche globale de telles "régions favorables" est souvent faite de manière heuristique (à moins évidemment que f est globalement convexe). Souvent, également, on reste un gradient à pas fixe, on choisit une initialisation un peu au hasard et on espère que l'algorithme va converger. Mais, lorsqu'on a plus d'informations sur la fonction, il faut essayer de s'en servir !

Remarque 26. On a vu à la remarque 7 (dont on remet les figures ici) que, comme f est supposée convexe sur U, les ensembles de sous-niveau $\{x \in U \mid f(x) \leqslant \alpha\}$ sont convexes. De plus, si f est C^1 en x et si $f(x) = \alpha$, le gradient $\nabla f(x)$ est un vecteur qui est orthogonal à l'ensemble de niveau $f = \alpha$ (qui est une hypersurface au voisinage de x). Cette interprétation du gradient est à la base de la méthode de gradient : le gradient est orthogonal aux surfaces de niveau, et est orienté dans le sens où f croît, comme on le voit sur la figure.



Sur la figure de droite, les ensembles de niveau $f = \alpha$ sont en bleu (les ensembles de sous-niveau sont leur intérieur), au centre se trouve le minimiseur x^*. En rouge sont représentées les courbes intégrales du gradient de f. Ce sont les courbes qui sont, en approximation, suivies par les itérations x_k.

L'ensemble V du théorème est un ensemble de sous-niveau compact (fermé borné) de f, contenu dans U (et contenant le minimiseur x^*), il est donc convexe fermé. Son intérieur $\mathring{V} = \{x \in U \mid f(x) < f(x^*) + \eta\}$ est un voisinage ouvert convexe de x^*.

1. Cela est vrai si f est C^2 et $H_f(x) \geqslant \alpha I_n$ pour tout $x \in U$, pour un $\alpha > 0$.
2. Cela est vrai si $H_f(x) \leqslant M I_n$ pour tout $x \in U$.

Démonstration. L'unique minimiseur x^* de f sur U est caractérisé par $\nabla f(x^*) = 0$ (par stricte convexité). L'ensemble V est convexe compact et contient x^* en son intérieur. Supposons être à l'itération k, avec $x_k \in V$ (c'est le cas pour $k = 0$, ce qui initialise la récurrence). On pose $x_{k+1} = x_k - \rho_k \nabla f(x_k)$, et on suppose que $\nabla f(x_k) \neq 0$ (sinon il n'y a rien à faire). Attention, on ne sait pas encore que $x_{k+1} \in V$: en effet, il se pourrait que $\rho_k > 0$ soit trop grand et que x_{k+1} sorte du convexe V. On va voir que ce n'est pas le cas grâce au choix de ρ_k, mais pour prendre en compte cette difficulté, on pose

$$\forall \rho > 0 \qquad x(\rho) = x_k - \rho \nabla f(x_k)$$

et bien sûr on a $x(\rho_k) = x_{k+1}$. Notons que, comme $\nabla f(x_k) \neq 0$, on a $x(\rho) \neq x_k$ pour tout $\rho > 0$. Il existe $\varepsilon > 0$ assez petit, tel que, pour tout $\rho \in]0, \varepsilon]$, on ait $x(\rho) \in V$ (voir figure dans la remarque ci-dessus) : pour le montrer rigoureusement, il suffit d'écrire un développement limité au premier ordre, $f(x(\rho)) = f(x_k - \rho \nabla f(x_k)) = f(x_k) - \rho \|\nabla f(x_k)\|^2 + o(\rho) < f(x_k)$ pour $\rho > 0$ assez petit, et donc $x(\rho) \in V$, et même, $x(\rho) \in \mathring{V}$. On pose

$$\beta = \max\{\rho \in]0, \beta_2] \;\mid\; x(\rho) \in V\}$$

(le supremum est bien un maximum car V est fermé). On vient de voir que $\beta > 0$, et notre objectif est de montrer que $\beta = \beta_2$.

Soit $\rho \in]0, \beta]$ arbitraire. Suivant la technique déjà largement utilisée, posons

$$\varphi(t) = f(x_k + t(x(\rho) - x_k))$$

ce qui correspond à regarder la fonction f le long du segment $[x_k, x(\rho)]$: on a $\varphi(0) = f(x_k)$, $\varphi(1) = f(x(\rho))$ et $\varphi'(t) = \langle \nabla f(x_k + t(x(\rho) - x_k)), x(\rho) - x_k \rangle$. On a $\varphi(1) = \varphi(0) + \int_0^1 \varphi'(t)\, dt$, ce qui donne

$$\begin{aligned} f(x(\rho)) &= f(x_k) + \int_0^1 \langle \nabla f(x_k + t(x(\rho) - x_k)), x(\rho) - x_k \rangle \, dt \\ &= f(x_k) + \langle \nabla f(x_k), x(\rho) - x_k \rangle + \int_0^1 \langle \nabla f(x_k + t(x(\rho) - x_k)) - \nabla f(x_k), x(\rho) - x_k \rangle \, dt \end{aligned}$$

D'une part, comme $x(\rho) = x_k - \rho \nabla f(x_k)$, on a $\nabla f(x_k) = -\frac{x(\rho) - x_k}{\rho}$. D'autre part, comme $x_k \in V$ et $x(\rho) \in V$, par convexité de V on a aussi $x_k + t(x(\rho) - x_k) \in V$ pour tout $t \in [0, 1]$ (c'est justement ce point qui nous empêche d'appliquer tout ce raisonnement directement à $x(\rho_k) = x_{k+1}$), et on peut utiliser l'inégalité (3.1). En utilisant l'inégalité de Cauchy-Schwarz, on obtient

$$f(x(\rho)) - f(x_k) \leqslant -\frac{1}{\rho} \|x(\rho) - x_k\|^2 + M \int_0^1 t \, dt \, \|x(\rho) - x_k\|^2 = \left(\frac{M}{2} - \frac{1}{\rho} \right) \|x(\rho) - x_k\|^2$$

et comme $\rho \leqslant \beta_2$, on a

$$f(x(\rho)) - f(x_k) \leqslant \underbrace{\left(\frac{M}{2} - \frac{1}{\beta_2} \right)}_{<0} \underbrace{\|x(\rho) - x_k\|^2}_{\rho^2 \|\nabla f(x_k)\|^2 > 0} < 0$$

donc $f(x(\rho)) < f(x_k)$ (et donc $x(\rho) \in \mathring{V}$).

En prenant $\rho = \beta$, on a donc $f(x(\beta)) < f(x_k)$. Comme la fonction $\rho \mapsto f(x(\rho))$ est continue, on en déduit qu'il existe $\delta > 0$ petit tel que $f(x(\beta + \delta)) < f(x_k)$, et donc $x(\beta + \delta) \in \mathring{V}$: cela montre (par l'absurde) qu'on a forcément $\beta = \beta_2$.

Ainsi, tout le raisonnement ci-dessus peut être appliqué à $\rho = \rho_k$, puisque, par hypothèse, $\rho_k \leqslant \beta_2$. On a donc

$$f(x_{k+1}) - f(x_k) \leqslant \underbrace{\left(\frac{M}{2} - \frac{1}{\beta_2}\right)}_{<0} \underbrace{\|x_{k+1} - x_k\|^2}_{\rho_k^2 \|\nabla f(x_k)\|^2 > 0} < 0$$

dont on déduit deux choses. Premièrement, la suite $(f(x_k))_{k\in\mathbb{N}}$ est décroissante. Comme elle est minorée (f a un minimum sur U), elle converge. Donc la suite de réels positifs $f(x_k) - f(x_{k+1})$ converge vers 0. Deuxièmement, en renversant l'inégalité, on a

$$\|x_{k+1} - x_k\|^2 \leqslant \underbrace{\frac{1}{\frac{1}{\beta_2} - \frac{M}{2}}}_{>0} (f(x_k) - f(x_{k+1}))$$

et donc $x_{k+1} - x_k$ converge vers 0 (dans $\mathbb{R}^n$). Or, $\nabla f(x_k) = \frac{x_k - x_{k+1}}{\rho_k}$, et par hypothèse, $\rho_k \geqslant \beta_1$ (il est important ici d'assurer que ρ_k ne converge pas vers 0), donc $\nabla f(x_k)$ converge vers 0.

Montrons finalement que x_k converge vers x^*. La suite $(x_k)_{k\in\mathbb{N}}$ restant dans le compact V, soit $\bar{x} \in V$ une valeur d'adhérence de cette suite (ce qui veut dire que $\bar{x}$ est la limite d'une sous-suite). Comme $\nabla f(x_k) \to 0$, en passant à la limite (comme f est C^1), on a $\nabla f(\bar{x}) = 0$. Donc $\bar{x} = x^*$ puisque x^* est le seul point possible de U qui annule le gradient. Ainsi, toute valeur d'adhérence de $(x_k)_{k\in\mathbb{N}}$ est égale à x^*. On conclut donc que la suite $(x_k)_{k\in\mathbb{N}}$ converge vers x^*. □

Remarque 27. Si dans le théorème on suppose que f est strictement convexe sur $\mathbb{R}^n$ tout entier, que f est infinie à l'infini, et que l'hypothèse (3.1) est globale sur $\mathbb{R}^n$, alors on n'a pas besoin de faire le raisonnement avec $x(\rho)$: on peut l'appliquer directement à x_{k+1}.

Le théorème 20 est plus fort et s'applique aux minimiseurs locaux. Il montre aussi que, pour assurer la convergence de la méthode de gradient, on a intérêt à avoir une idée a priori d'où se situent les minimiseurs locaux (et en effet, en pratique, on a souvent une connaissance intuitive du problème qui permet de faire une première localisation grossière) et à initialiser la méthode en un point x_0 en lequel on a la propriété locale de convexité.

Remarque 28 (Méthode du gradient à pas fixe)**.** Lorsque $\rho_k = \rho$ est constant, on parle de méthode de gradient à pas fixe. C'est certainement la méthode d'optimisation sans contrainte la plus simple.

Il est intéressant de noter que la méthode de gradient à pas fixe s'interprète comme une méthode de point fixe. En effet, en posant

$$F_\rho(x) = x - \rho\nabla f(x),$$

on constate que

$$x_{k+1} = x_k - \rho\nabla f(x_k) \quad \Leftrightarrow \quad x_{k+1} = F_\rho(x_k)$$

Autrement dit, les itérations de la méthode de gradient à pas fixe sont exactement les itérations de la méthode du point fixe de Picard.

Or, on sait que la méthode du point fixe converge lorsque l'application F_ρ est K-contractante, i.e., K-Lipschitzienne avec $0 \leqslant K < 1$. Vérifions que c'est bien le cas pour F_ρ, sous la condition (3.1) du théorème 20 (i.e., f est à gradient Lipschitz) et sous la condition, plus forte que dans le

théorème, que f soit α-convexe pour un $\alpha > 0$: on a, pour tous $x, y \in V$,

$$\begin{aligned}\|F_\rho(x) - F_\rho(y)\|^2 &= \|x - y - \rho(\nabla f(x) - \nabla f(y))\|^2 \\ &= \|x-y\|^2 + \rho^2 \underbrace{\|\nabla f(x) - \nabla f(y)\|^2}_{\leqslant M^2\|x-y\|^2 \text{ par (3.1)}} - 2\rho \underbrace{\langle x - y, \nabla f(x) - \nabla f(y)\rangle}_{\geqslant \alpha\|x-y\|^2 \text{ par le théorème 8 car } f \text{ est } \alpha\text{-convexe}} \\ &\leqslant \underbrace{(1 - 2\rho\alpha + \rho^2 M^2)}_{K^2} \|x-y\|^2\end{aligned}$$

On a $K < 1$ si et seulement si $0 < \rho < \frac{2\alpha}{M^2}$. De plus, la valeur minimale de K est $K = 1 - \frac{\alpha^2}{M^2}$ et est obtenue pour $\rho = \frac{\alpha}{M^2}$.

Or, on sait que, dans la méthode de point fixe, on a $x_{k+1} - x^* = F_\rho(x_k) - F_\rho(x^*)$ (car $x^* = F_\rho(x^*)$), donc $\|x_{k+1} - x^*\| \leqslant K\|x_k - x^*\|$ et donc

$$\boxed{\|x_k - x^*\| \leqslant K^k \|x_0 - x^*\| \qquad \forall k \in \mathbb{N}}$$

ce qui veut dire que la méthode de gradient à pas fixe converge linéairement, avec vitesse de convergence K (voir le rappel ci-dessous).

Ainsi, dans le cas d'un gradient à pas fixe, si f est α-convexe et à gradient Lipschitz sur U, on a obtenu (certes, sous des hypothèses un peu plus fortes que dans le théorème 20) une preuve alternative, qui a l'avantage de donner une vitesse de convergence explicite : la méthode converge linéairement si $\rho < \frac{2\alpha}{M^2}$, à vitesse $K = \sqrt{1 - 2\rho\alpha + \rho^2 M^2}$. De plus, le meilleur pas fixe possible (celui qui donne le K le plus petit) est $\rho = \frac{\alpha}{M^2}$.

Notons que $\alpha < M$, et donc le seuil de convergence $\frac{2\alpha}{M^2}$ établi ici est bel et bien inférieur au seuil de convergence $\frac{2}{M}$ établi dans le théorème 20.

Remarque 29 (Vitesse de convergence)**.** On rappelle le vocabulaire suivant. Soit $(x_k)_{k\in\mathbb{N}}$ une suite convergeant vers x^* (on suppose ci-dessous que $x_k \neq x^*$ pour tout k).

On dit que $(x_k)_{k\in\mathbb{N}}$ converge à l'ordre $p \geqslant 1$ vers x^*, à vitesse $c \in]0, 1[$, si

$$\lim_{k\to+\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} = c. \tag{3.2}$$

La convergence est dite linéaire si $p = 1$, quadratique si $p = 2$, cubique si $p = 3$, etc. Lorsque (3.2) est vérifiée avec $p = 1$ et $c = 0$, on dit que la convergence est super-linéaire. Attention au mot "vitesse", car plus c est petit et meilleur c'est !

Dans la pratique, on a rarement une convergence meilleure que quadratique (on verra que c'est le cas de la méthode de Newton). Une vitesse de convergence quadratique correspond à un doublement de la précision numérique à chaque étape. Un algorithme, lorsqu'il converge quadratiquement, est donc très efficace ! (encore faut-il l'initialiser convenablement, de manière à ce qu'il converge)

3.4.1.3 Méthode de gradient à pas optimal

L'idée de la méthode de gradient à pas optimal est de faire une méthode de gradient à pas variable dans laquelle on choisit, à chaque itération, le pas qui permet de faire décroître la fonction f le plus possible :

$$\boxed{\begin{array}{ll} x_{k+1} = x_k + \rho_k d_k & \\ d_k = -\nabla f(x_k) & \text{où} \qquad \rho_k = \underset{\rho>0}{\operatorname{argmin}}\ f(x_k + \rho d_k) \end{array}}$$

Posons

$$\varphi(\rho) = f(x_k + \rho d_k).$$

Il s'agit donc, à chaque étape, de déterminer (s'il existe) le minimum de la fonction φ sur $]0, +\infty[$. Notons qu'un tel minimum existe si f est infinie à l'infini, car $\varphi'(0) = -\|d_k\|^2 < 0$ (pourvu que $d_k = \nabla f(x_k) \neq 0$ bien sûr) donc φ décroît pour $\rho > 0$ petit, et par ailleurs $\varphi(\rho) \to +\infty$ lorsque $\rho \to +\infty$. On a $\varphi'(\rho_k) = 0$.

Déterminer un tel minimum global peut toutefois s'avérer coûteux, et en pratique, on cherche donc une approximation du pas optimal $\rho_k > 0$. En supposant que ρ_k est petit, on écrit le développement limité

$$0 = \varphi'(\rho_k) = df(x_k + \rho_k d_k).d_k = df(x_k).d_k + \rho_k d^2 f(x_k).(d_k, d_k) + \mathrm{o}(\rho_k)$$

et en négligeant le terme en o, comme $d_k = -\nabla f(x_k)$, on obtient

$$\rho_k = \frac{\|d_k\|^2}{d_k^\top H_k d_k} \qquad \text{où} \qquad H_k = H_f(x_k).$$

Notons que ces formules sont exactes lorsque f est quadratique, i.e., lorsque $f(x) = \frac{1}{2}x^\top Ax - b^\top x$ avec A symétrique définie positive.

Même dans le cas non linéaire, on a l'habitude d'appeler méthode de gradient à pas optimal la méthode suivante :

$$\boxed{x_{k+1} = x_k - \rho_k \nabla f(x_k) \qquad \text{où} \qquad \rho_k = \frac{d_k^\top d_k}{d_k^\top H_k d_k} = \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^\top H_f(x_k) \nabla f(x_k)}}$$

Cette méthode (dont on ne fait pas l'analyse de convergence ici) est excellente en pratique mais peut toutefois s'avérer coûteuse en grande dimension car elle réclame de calculer la Hessienne de f à chaque étape. Or, dans des problèmes de sciences des données, cela peut être rhédibitoire. Il faut donc chercher d'autres alternatives.

On verra, dans les méthodes de type Newton, qu'on pourrait remplacer la Hessienne par une "quasi-Hessienne" (i.e., on remplace H_k par une suite dont on sait démontrer qu'elle constitue, pour k grand, une bonne approximation de la Hessienne).

Mais, même le temps de calcul du gradient peut être problématique lorsque le nombre de variables est très grand. Certains problèmes comportent des millions de variables, et si la fonction est un peu trop non linéaire, calculer le gradient complet à chaque étape est trop lent : en effet, si $n = 10^6$ par exemple, chaque itération k réclame de calculer $n = 10^6$ dérivées partielles (évaluées au point x_k). Il faut alors trouver des moyens de réduire la dimension, d'une manière ou d'une autre. Dans la section suivante, on décrit une méthode très simple de gradient coordonnée par coordonnée, ou bloc par bloc, qui permet, en fait, "d'avancer de manière un peu diagonale".

3.4.1.4 Méthode de descente coordonnée par coordonnée, bloc par bloc

Cette méthode, ancienne, est largement revenue à la mode avec l'avènement des sciences des données. L'idée est simple : au lieu de minimiser sur l'ensemble des $x \in \mathbb{R}^n$, on minimise coordonnée par coordonnée, à chaque étape, ce qui donne un algorithme du type

$$\min_{x \in \mathbb{R}} f(x_1^k, \ldots, x_{i-1}^k, x, x_{i+1}^k, \ldots, x_n^k), \quad \text{pour } i = 1, \ldots, n$$

qu'on appelle méthode de relaxations successives de type Jacobi, ou, comme variante plus utilisée :

$$\min_{x \in \mathbb{R}} f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x, x_{i+1}^k, \ldots, x_n^k), \quad \text{pour } i = 1, \ldots, n$$

qu'on appelle méthode de relaxations successives de type Gauss-Seidel. Ici, par commodité, l'indice d'itération k est mis en haut ; l'indice du bas étant la coordonnée.

Ces dénominations Jacobi / Gauss-Seidel proviennent des méthodes itératives de résolution de systèmes linéaires $Ax = b$, qui consistent à décomposer $A = D - E - F$ avec D diagonale, E triangulaire inférieure et F triangulaire supérieure (qu'on appelle de manière générale les méthodes de relaxation). La méthode de Gauss-Seidel est généralement préférée – plus efficace car, à l'étape k, au fur et à mesure des calculs sur $i = 1, \dots, n$, on se sert des valeurs de x_i^{k+1} déjà calculées.

Quoi qu'il en soit, l'idée est ici, à chaque étape k, de remplacer l'itération "globale sur x" (qui consiste, dans une méthode de descente, à calculer $x^{k+1} = x^k + \rho_k d_k$ où d_k est une direction de descente, par exemple $d_k = -\nabla f(x_k)$) par n calculs qui se font chacun, coordonnée par coordonnée. Par exemple, à l'étape k, pour chaque $i = 1, \dots, n$, on peut choisir une méthode de gradient (à pas ρ_k), ce qui donne, dans la méthode de Gauss-Seidel :

$$\boxed{x_i^{k+1} = x_i^k - \rho_k \frac{\partial f}{\partial x_i}(x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \dots, x_n^k), \qquad i = 1, \dots, n}$$

Notons que, par rapport à une méthode de gradient classique, on calcule le même nombre de dérivées partielles ! Mais les calculs et itérations ne sont pas faits dans le même ordre, et cela change tout. Dans la méthode de gradient classique, à chaque étape k, on calcule toutes les dérivées partielles pour $i = 1, \dots, n$ et on fait un pas de gradient. Alors que, dans la méthode coordonnée par coordonnée, à chaque étape k, on itère sur $i = 1, \dots, n$, et pour chaque i, on fait un pas de gradient par rapport à la coordonnée x_i. C'est différent ! On conçoit en effet que cela peut aller plus vite car, à chaque étape, on peut espérer avoir significativement avancé dans la convergence par rapport à la coordonnée x_i.

Au lieu de choisir une méthode de gradient, on peut bien entendu choisir toute autre méthode de minimisation de fonctions à une variable, ce qui engendre plein de variantes possibles.

D'autre part, ci-dessus, on s'est ramené à des minimisations de fonctions à une variable, mais on aurait pu aussi faire des regroupements de variables, des "blocs", et remplacer l'itération $i = 1, \dots, n$ qui est faite sur toutes les coordonnées, par une itération faite sur les blocs de variables. On appelle ce type de méthode, les méthodes de descente bloc par bloc. En fait la méthode s'écrit exactement de la même façon que ci-dessus pourvu de remplacer $x_i \in \mathbb{R}$ par $x_i \in \mathbb{R}^{n_i}$ où n_i est la dimension du bloc i.

Théorème 21. *On suppose que $f : \mathbb{R}^n \to \mathbb{R}$ est C^1, infinie à l'infini et strictement convexe. Alors la méthode de gradient coordonnée par coordonnée de type Gauss-Seidel converge vers l'unique minimiseur de f.*

La preuve (qu'on peut trouver, par exemple, dans le livre de Glowinski, Lions, Trémolières, 1981, Chapter 2) est du même type que celle du théorème 20.

Enfin, dernière remarque, ces méthodes ont connu un fort regain d'intérêt récemment avec l'avènement du deep learning. Le choix des coordonnées ou des blocs de descente peut être fait de manière aléatoire, par exemple avec probabilité uniforme. La convergence de ce type de méthode a été étudiée récemment.

3.4.1.5 Méthodes de recherche linéaire

Il existe une immense quantité de variantes possibles de méthodes de descente. L'objectif de ce cours est d'en voir quelques-unes, les plus classiques, afin de développer l'intuition. En pratique, chaque problème a ses particularités et lorsqu'on veut résoudre efficacement un problème spécifique on est souvent amené à combiner diverses méthodes. D'où l'intérêt de bien comprendre, pour chaque méthode, quels sont ses avantages et inconvénients.

Dans cette section, on discute de méthodes heuristiques permettant d'orienter le choix du pas ρ_k à chaque étape. On a déjà vu qu'une (très) bonne méthode pour choisir le pas ρ_k était la méthode du gradient à pas optimal. Mais elle présente le défaut de devoir calculer la Hessienne (ou au moins une approximation de la Hessienne) à chaque itération, ce qui peut être prohibitif si n est très grand.

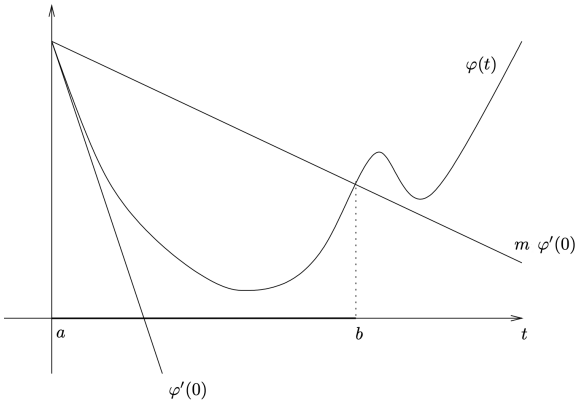
On va donc décrire ici des possibilités très simples qui permettent d'orienter le choix du pas ρ_k, dans une méthode de descente $x_{k+1} = x_k + \rho_k d_k$, de manière intelligente. Comme précédemment, à l'étape k, on pose

$$\varphi(\rho) = f(x_k + \rho d_k).$$

L'objectif d'une recherche linéaire est de ne pas chercher à calculer le minimum de φ (car cela peut être trop coûteux), et de déterminer un pas ρ_k permettant d'assurer que φ décroît suffisamment. En même temps, il faut assurer que ρ_k ne soit ni trop petit ni trop grand. Tout cela est très heuristique ! Et largement basé sur l'intuition.

Dans les différentes règles décrites ci-dessous, on décide de choisir ρ_k dans un intervalle $[a, b]$ appelé "intervalle de sécurité" : autrement dit, un réel $\rho < a$ est considéré comme étant trop petit, un réel $\rho > b$ est considéré comme étant trop grand, et lorsqu'on trouve un $\rho \in [a, b]$ on décide qu'il est convenable.

Règle d'Armijo. On fixe un réel $m \in]0, 1[$, et on décide que ρ convient si $\varphi(\rho) \leqslant \varphi(0) + m\varphi'(0)\rho$.



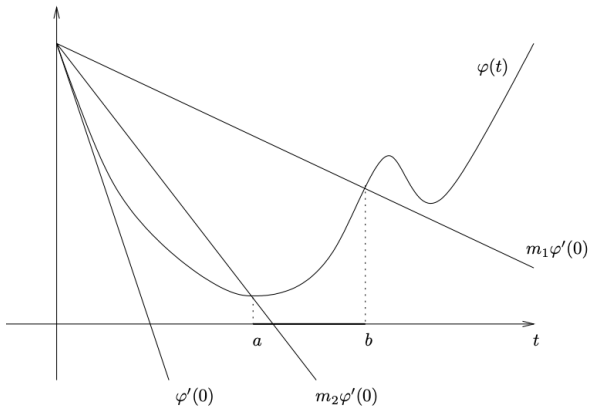
Autrement dit, ρ convient si $\varphi(\rho)$ est en dessous de la droite de pente $m\,\varphi'(0)$ passant par $(0, \varphi(0) = f(x_k))$ (rappelons que $\varphi'(0) < 0$ grâce au choix de la direction de descente).

Dans la règle d'Armijo, on a donc $a = 0$, et b est l'abscisse du premier point d'intersection entre la droite de pente $m\,\varphi'(0)$ ci-dessus et le graphe de φ. Mais comme $a = 0$, on n'a pas de borne inférieure sur le pas, et donc, souvent, on combine cette règle à d'autres règles.

Règle de Goldstein. On fixe deux réels $0 < m_1 < m_2 < 1$, et on décide que ρ convient si

$$\varphi(0) + m_2\varphi'(0)\rho \leqslant \varphi(\rho) \leqslant \varphi(0) + m_1\varphi'(0)\rho$$

autrement dit, si $\varphi(\rho)$ est en dessous de la droite de pente $m_1\,\varphi'(0)$ passant par $(0, \varphi(0) = f(x_k))$, et au-dessus de la droite de pente $m_2\,\varphi'(0)$ passant par $(0, \varphi(0) = f(x_k))$.

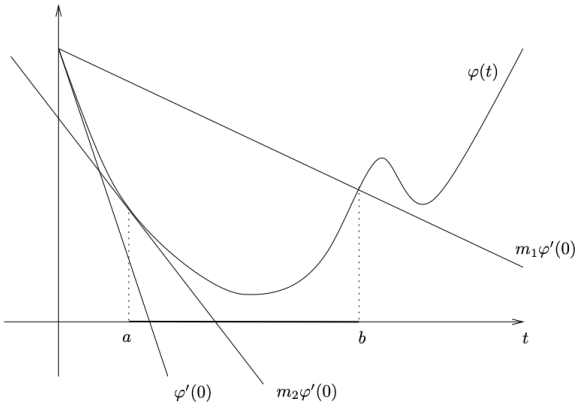


On choisit en général $m_2 \geqslant \frac{1}{2}$ de façon à ce que le pas optimal déterminé dans le cas quadratique appartienne à l'intervalle de sécurité (vérifiez cela par un calcul). Par exemple, $m_1 = 0.1$ et $m_2 = 0.7$.

Règle de Wolfe. On fixe deux réels $0 < m_1 < m_2 < 1$, et on décide que ρ convient si

$$m_2\varphi'(0) \leqslant \varphi'(\rho) \qquad \text{et} \qquad \varphi(\rho) \leqslant \varphi(0) + m_1\varphi'(0)\rho$$

autrement dit, si $\varphi(\rho)$ est en dessous de la droite de pente $m_1\,\varphi'(0)$ passant par $(0, \varphi(0) = f(x_k))$, et si la pente de φ en 0 est plus grande que $m_2\varphi'(0)$.



Dans cette règle, il faut estimer $\varphi'(0)$ et donc calculer $\nabla f(x_k)$ à chaque pas. Mais, si cela n'est pas trop coûteux, cette information différentielle est plus précise.

Bien sûr, on peut imaginer quantité d'autres variantes.

3.4.2 Méthodes de type Newton

Pour trouver un minimum de f, plutôt qu'une méthode de descente, on peut chercher à résoudre l'équation $\nabla f(x) = 0$, vérifiée (condition nécessaire) par x^*. Dans toute cette section, on pose

$$\boxed{F(x) = \nabla f(x)}$$

on suppose $f : \mathbb{R}^n \to \mathbb{R}$ de classe C^2, i.e., $F : \mathbb{R}^n \to \mathbb{R}^n$ de classe C^1, et on cherche à résoudre

$$\boxed{F(x) = 0} \tag{3.3}$$

qui est un système (non linéaire) de n équations à n inconnues.

Bien entendu, ce système étant non linéaire, il peut admettre plusieurs solutions. Nous sommes intéressés à trouver la solution x^*, où x^* est un minimiseur (au moins local) de f. Mais notons que tout minimiseur local, tout maximiseur local vérifie (3.3) – et pas seulement : on peut aussi avoir des solutions de (3.3) qui ne sont ni un minimiseur ni un maximiseur local (comme par exemple 0 pour l'équation $x^3 = 0$).

Le problème (3.3) ne peut donc être "bien posé" (i.e., admettre une unique solution) que dans un ouvert U de $\mathbb{R}^n$, éventuellement assez petit. Et encore, malgré ce caractère bien posé local, le fait d'être l'unique solution dans U de (3.3) ne distinguera pas entre un minimum, maximum ou extremum qui n'est aucun des deux.

3.4.2.1 Méthode classique de Newton

Etant donné un point $x_k \in \mathbb{R}^n$, supposé assez proche de la solution x^* (qui vérifie $F(x^*) = 0$), cherchons un nouveau point x_{k+1} qui soit "meilleur", au sens que $F(x_{k+1}) \simeq 0$. En posant $\Delta_k = x_{k+1} - x_k$, un développement limité à l'ordre 1 en x^* donne

$$0 \simeq F(x_{k+1}) = F(x_k + \Delta_k) \simeq F(x_k) + dF(x_k).\Delta_k$$

ce qui nous conduit à choisir x_{k+1} tel que $F(x_k) + dF(x_k).\Delta_k = 0$, i.e., lorsque c'est possible,

$$\boxed{x_{k+1} = x_k - dF(x_k)^{-1}.F(x_k)}$$

Ici, $dF(x_k)$, la différentielle de F au point x_k, s'identifie à une matrice carrée de taille n, et on doit donc assurer qu'elle soit inversible. Comme x_k est supposé proche de x^*, cela conduit naturellement à supposer que $dF(x^*)$ est inversible. Cette condition est l'une des conditions suffisantes assurant le caractère localement bien posé de la méthode de Newton.

Théorème 22. *On suppose que $F : \mathbb{R}^n \to \mathbb{R}^n$ est de classe C^1, que $F(x^*) = 0$, que $dF(x^*)$ est inversible et que dF est Lipschitzienne*[3] *au voisinage de x^*, i.e., il existe $r_0 > 0$ et $L > 0$ tel que*

$$\|dF(x) - dF(y)\|_{L(\mathbb{R}^n)} \leqslant L\|x - y\| \qquad \forall x, y \in B(x^*, r_0). \tag{3.4}$$

Alors il existe $r \in]0, r_0]$ tel que, pour tout $x_0 \in B(x^*, r)$, la suite $(x_k)_{k\in\mathbb{N}}$ définie par l'itération

$$\boxed{x_{k+1} = x_k - dF(x_k)^{-1}.F(x_k)}$$

partant de $x_0 \in B(x^*, r)$, est bien définie, reste dans la boule $B(x^*, r)$, et converge vers x^*. De plus, la convergence est quadratique (voir remarque 29), i.e., il existe $c > 0$ tel que

$$\boxed{\|x_{k+1} - x^*\| \leqslant c\|x_k - x^*\|^2 \qquad \forall k \in \mathbb{N}}$$

Dans (3.4), $\|\ \|$ est (par exemple) la norme Euclidienne de $\mathbb{R}^n$, la boule $B(x^*, r_0)$ est la boule Euclidienne de centre x^* et de rayon r_0. La norme $\|\ \|_{L(\mathbb{R}^n)}$ est la norme d'opérateur, dont on rappelle qu'elle est définie par

$$\|\ell\|_{L(\mathbb{R}^n)} = \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|\ell(x)\|}{\|x\|} = \sup_{\substack{x\in\mathbb{R}^n \\ \|x\|=1}} \|\ell(x)\|$$

pour toute application linéaire (continue) $\ell \in L(\mathbb{R}^n)$.

3. Cela est vrai si F est C^2 par le théorème des accroissements finis.

Démonstration. Comme F est C^1 et $dF(x^*)$ est inversible, il existe $0 < R < r_0$ tel que $dF(x)$ reste inversible pour tout $x \in \overline{B}(x^*, R)$. On pose alors

$$M = \max_{x \in \overline{B}(x^*,R)} \|dF(x)^{-1}\|_{L(\mathbb{R}^n)}.$$

Soit $x_0 \in \overline{B}(x^*, R)$. Par définition, on a $x_1 = x_0 - dF(x_0)^{-1}.F(x_0)$, et comme $F(x^*) = 0$ on peut écrire

$$\begin{aligned} x_1 - x^* &= x_0 - x^* - dF(x_0)^{-1}.(F(x_0) - F(x^*)) \\ &= -dF(x_0)^{-1}.(F(x_0) - F(x^*) - dF(x_0).(x_0 - x^*)) \end{aligned}$$

Mais d'une part on a $F(x_0) - F(x^*) = \int_0^1 dF(x^* + t(x_0 - x^*)).(x_0 - x^*)\, dt$ (formule de Taylor avec reste intégral à l'ordre 1), et d'autre part, trivialement, $dF(x_0).(x_0 - x^*) = \int_0^1 dF(x_0).(x_0 - x^*)\, dt$, donc

$$F(x_0) - F(x^*) - dF(x_0).(x_0 - x^*) = \int_0^1 \Big(dF(x^* + t(x_0 - x^*)) - dF(x_0)\Big).(x_0 - x^*)\, dt$$

et alors, comme dF est Lipschitzienne dans $B(x^*, r_0)$,

$$\begin{aligned} \|F(x_0) - F(x^*) - dF(x_0).(x_0 - x^*)\| &\leqslant \int_0^1 \|dF(x^* + t(x_0 - x^*)) - dF(x_0)\|_{L(\mathbb{R}^n)} \|x_0 - x^*\|\, dt \\ &\leqslant \int_0^1 L\|(1-t)(x_0 - x^*)\| \|x_0 - x^*\|\, dt \\ &\leqslant L\|x_0 - x^*\|^2 \int_0^1 (1-t)\, dt = \frac{L}{2}\|x_0 - x^*\|^2 \end{aligned}$$

et donc finalement,

$$\|x_1 - x^*\| \leqslant \frac{ML}{2}\|x_0 - x^*\|^2.$$

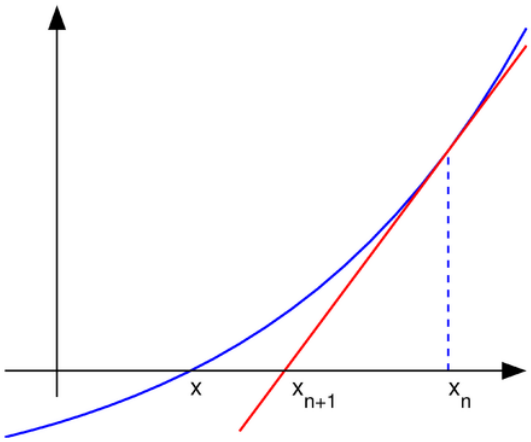
Posons alors $r = \min\left(\frac{1}{ML}, R\right)$. Si $x_0 \in B(x^*, r) \subset B(x^*, R)$ alors $ML\|x_0 - x^*\| < 1$ et donc $\|x_1 - x^*\| \leqslant \frac{1}{2}\|x_0 - x^*\|$. En particulier, $x_1 \in B(x^*, r)$.

Ce raisonnement peut être itéré et on obtient que la suite $(x_k)_{k \in \mathbf{N}}$ reste dans la boule $B(x^*, r)$ et $\|x_k - x^*\| \leqslant \frac{1}{2^k}\|x_0 - x^*\|$, donc la suite converge vers x^*. Pour la vitesse de convergence, on a même obtenu mieux, avec la convergence quadratique. □

Remarque 30. La méthode de Newton, lorsqu'elle converge, est très rapide puisqu'elle est à vitesse de convergence quadratique (voir remarque 29). Il faut toutefois décider d'un critère d'arrêt dans les itérations. Etant donné un $\varepsilon > 0$, des possibilités sont $\|x_{k+1} - x_k\| \leqslant \varepsilon$, ou bien $\|F(x_k)\| \leqslant \varepsilon$.

Remarque 31. Lorsque l'algorithme converge, on détermine donc un $x^* \in \mathbb{R}^n$ tel que $F(x^*) = \nabla f(x^*) = 0$. Rien ne garantit cependant que x^* soit un minimiseur local : ce pourrait être un maximiseur, ou même, un extremum local qui n'est ni un minimum ni un maximum. Lorsqu'on implémente la méthode de Newton, une fois obtenu x^*, on peut calculer la Hessienne $H_f(x^*)$ et vérifier des conditions de positivité : par exemple, cette matrice étant symétrique réelle donc diagonalisable à valeurs propres réelles, on peut calculer numériquement sa plus petit valeur propre et vérifier qu'elle est strictement positive. Si c'est le cas alors x^* est bien un minimiseur local de f.

Remarque 32. En dimension 1, la méthode de Newton s'interprète ainsi : étant donné le point x_n, on construit le point x_{n+1} comme étant l'intersection de l'axe des abscisses avec la tangente à f en x_n.



Remarque 33 (Rapport avec le point fixe de Banach)**.** Résoudre $F(x) = 0$ est équivalent à résoudre

$$G_M(x) = x - MF(x) = x$$

où $M \in GL_n(\mathbb{R})$ est une matrice inversible quelconque, autrement dit, à trouver un point fixe de G_M. L'algorithme du point fixe consiste à considérer la suite définie par récurrence

$$x_{k+1} = G_M(x_k) = x_k - MF(x_k)$$

On sait que cette suite converge vers x^* lorsque l'application G_M est contractante. Or, $dG_M(x) = I_n - MdF(x)$, et donc, pour assurer que G_M soit contractante dans un voisinage de x^*, on peut choisir $M = dF(x^*)^{-1}$; mais comme on ne connaît pas x^*, il est naturel de choisir $M = M_k$ dépendant de k, en posant $M_k = dF(x_k)^{-1}$, ce qui est exactement la méthode de Newton !

Remarque 34. A moins que la fonction F vérifie certaines conditions globales, la convergence de la méthode de Newton est locale. On a un domaine de convergence, qui est souvent assez petit. La difficulté majeure de la méthode de Newton est donc son initialisation : il faut en effet être capable de deviner un point initial x_0 qui est dans le domaine de convergence (donc, proche de x^*). Cela peut sembler paradoxal : pour déterminer x^* qui est inconnu, on a intérêt à déjà en connaître une approximation, si on veut assurer la convergence de la méthode de Newton. Ce genre de considération est inhérent à toute méthode locale.

De nombreuses méthodes existent pour rendre la méthode de Newton "un peu moins locale", et arriver à la faire converger même en n'ayant qu'une idée assez vague d'où se situe le point x^* recherché. Par exemple :

— Il existe des méthodes de Newton "globales", qui requièrent toutefois des hypothèses fortes sur la fonction F. On n'en parle pas ici.

— Avant d'appliquer une méthode de Newton, on peut, au préalable, appliquer une méthode de descente à la fonction

$$x \mapsto \|F(x)\|^2.$$

L'avantage est que cette méthode peut converger "plus facilement", avec une initialisation grossière. Ainsi, en quelques itérations, elle peut fournir un point un peu meilleur dont on peut se servir pour initialiser (on l'espère, avec succès) la méthode de Newton.

On appelle cela une méthode hybride. Il y a évidemment plein de manières d'hybridifier la méthode de Newton. A vous de combiner astucieusement les méthodes que vous connaissez, en fonction du problème !

— Une méthode très puissante pour faire converger la méthode de Newton est de la combiner à une *méthode de continuation* : on déforme l'équation $F(x) = 0$ à l'aide d'un paramètre (qui, souvent, est un ou plusieurs paramètres qui dans le problème rendent délicate sa résolution numérique, et qu'il s'agit alors, d'une manière ou d'une autre, de relaxer). Appelons λ ce paramètre, et supposons pour simplifier que $\lambda \in [0, 1]$ (mais il pourrait y avoir plusieurs paramètres !). On suppose maintenant que la fonction F dépend aussi du paramètre λ, i.e., on a $F(\lambda, x)$, avec, pour $\lambda = 1$, $F(1, x) = F(x)$ qui est la fonction de départ. On veut maintenant résoudre

$$F(\lambda, x) = 0$$

dont l'unique solution locale (sous des conditions qui assurent le caractère bien posé) est notée x_λ. Pour $\lambda = 1$, on a $x_1 = x^*$ qui est le minimiseur recherché. Pour $\lambda = 0$, on suppose que l'équation

$$F(0, x) = 0$$

est "facile à résoudre" (c'est ce qu'on doit être capable d'assurer en choisissant adéquatement le paramètre de continuation), sa solution étant x_0. C'est notre point de départ. A partir de ce point de départ, on résout maintenant $F(\lambda, x) = 0$ pour λ petit (par exemple, $\lambda = 0.1$), par une méthode de Newton initialisée à x_0 qui était la solution de $F(0, x_0) = 0$. Comme λ est petit, on peut s'attendre à ce que $x_\lambda \simeq x_0$, donc, à ce que la méthode converge. Si la méthode a convergé, on recommence avec λ un peu plus grand (par exemple, $\lambda = 0.2$), et ainsi de suite. En cas d'échec, on diminue l'incrément de λ.

On voit ainsi qu'au lieu de résoudre un seul problème de Newton, on en résout toute une série. Comme l'exécution de la méthode de Newton est quasi-instantanée (vitesse de convergence quadratique), cela n'est pas un problème et en général les continuations sur des problèmes de Newton sont rapides et efficaces. Il en existe de nombreuses variantes.

Remarque 35. Appliqué à $F(x) = \nabla f(x)$, l'algorithme de Newton s'écrit

$$\boxed{x_{k+1} = x_k - H_f(x_k)^{-1} \nabla f(x_k)}$$

On voit qu'il est nécessaire, à chaque itération, de calculer la Hessienne de f en x_k. Cela peut être coûteux, surtout si on est en grande dimension. Cela nous conduit aux méthodes de *quasi-Newton.*

3.4.2.2 Méthodes de quasi-Newton

L'idée des méthodes de quasi-Newton est de remplacer $H_f(x_k)$ (ou $H_f(x_k)^{-1}$) par une matrice $\tilde{H}_k$ (ou B_k) "moins lourde à calculer", et qui en est une bonne approximation, au moins lorsque k est grand. Ainsi, on cherche une suite de matrices $(\tilde{H}_k)_{k\in\mathbf{N}}$ (ou $(B_k)_{k\in\mathbf{N}}$), définie par récurrence, telle que $\tilde{H}_k \simeq H_f(x_k)$ (ou $B_k \simeq H_f(x_k)^{-1}$) lorsque k est grand. Dans la suite, on pose

$$\boxed{g_k = \nabla f(x_k)} \qquad \boxed{H_k = H_f(x_k)} \qquad \boxed{s_k = x_{k+1} - x_k} \qquad \boxed{y_k = g_{k+1} - g_k}$$

L'idée est de partir du développement limité $\nabla f(x_{k+1}) = \nabla f(x_k + x_{k+1} - x_k) = \nabla f(x_k) + H_f(x_k)(x_{k+1} - x_k) + \mathrm{o}(\|x_{k+1} - x_k\|)$, ce qui donne l'approximation

$$y_k \simeq H_k s_k \quad \Leftrightarrow \quad H_k^{-1} y_k \simeq s_k$$

Ainsi, on souhaite déterminer une suite de matrices $(\tilde{H}_k)_{k\in\mathbf{N}}$ ou $(B_k)_{k\in\mathbf{N}}$ telles que

$$y_k = \tilde{H}_k s_k \qquad \text{ou} \qquad B_k y_k = s_k.$$

Mais il faut déterminer de telles matrices, qui soient symétriques définies positives, pour tout k !

Algorithme DFP (Davidon, Fletcher, Powell). Cette formule de mise à jour est une formule de correction de rang 2 donnée par

$$B_{k+1} = B_k + \frac{s_k s_k^\top}{s_k^\top y_k} - \frac{B_k y_k y_k^\top B_k}{y_k^\top B_k y_k}$$

On peut montrer (c'est admis, ici) que, pour toute matrice B_0 symétrique définie positive (par exemple, $B_0 = I$), l'algorithme de quasi-Newton (dit DFP)

$$\boxed{x_{k+1} = x_k - B_k \nabla f(x_k), \qquad B_{k+1} = B_k + \frac{s_k s_k^\top}{s_k^\top y_k} - \frac{B_k y_k y_k^\top B_k}{y_k^\top B_k y_k}}$$

(qui est l'algorithme de Newton dans lequel on a remplacé H_k^{-1} par B_k), converge vers un minimum local x^* de f, sous les mêmes hypothèses que la méthode de Newton, et de plus,

$$\lim_{k \to +\infty} B_k = H_f(x^*)^{-1}.$$

Autrement dit la méthode se comporte asymptotiquement comme la méthode de Newton, avec l'avantage qu'on évite de calculer l'inverse de la Hessienne à chaque itération : les matrices B_k sont bien plus rapides à calculer, et donnent une approximation de H_k^{-1} pour k grand.

Algorithme BFGS (Broyden, Fletcher, Goldfarb, Shanno). C'est aussi une formule de correction de rang 2, qui consiste à intervertir les rôles de s_k et y_k dans la formule DFP. Elle fournit alors une approximation $\tilde{H}_k$ de la Hessienne H_k. L'itération est définie par

$$\tilde{H}_{k+1} = \tilde{H}_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{\tilde{H}_k s_k s_k^\top \tilde{H}_k}{s_k^\top \tilde{H}_k s_k}$$

On peut montrer (c'est admis, ici) que, pour toute matrice $\tilde{H}_0$ symétrique définie positive (par exemple, $\tilde{H}_0 = I$), l'algorithme de quasi-Newton (dit BFGS)

$$\boxed{x_{k+1} = x_k - \tilde{H}_k^{-1} \nabla f(x_k), \qquad \tilde{H}_{k+1} = \tilde{H}_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{\tilde{H}_k s_k s_k^\top \tilde{H}_k}{s_k^\top \tilde{H}_k s_k}}$$

(qui est l'algorithme de Newton dans lequel on a remplacé H_k par $\tilde{H}_k$), converge vers un minimum local x^* de f, sous les mêmes hypothèses que la méthode de Newton, et de plus,

$$\lim_{k \to +\infty} \tilde{H}_k = H_f(x^*).$$

Par rapport à la méthode DFP, la méthode BFGS nécessite de calculer l'inverse de la matrice $\tilde{H}_k$, et peut donc sembler moins intéressante. Toutefois, elle s'avère être en général meilleure et plus robuste que la méthode DFP. La méthode BFGS est la méthode de quasi-Newton la plus connue et la plus utilisée.

3.4.2.3 Méthode de Barzilai Borwein (1988)

Pour expliquer cette méthode, partons de la méthode de gradient à pas optimal. Pour rappel, elle consiste à considérer les itérations

$$x_{k+1} = x_k - \rho_k g_k \qquad \text{où} \qquad \rho_k = \operatorname*{argmin}_{\rho > 0} f(x_k - \rho g_k)$$

Comme la fonction $\rho \mapsto f(x_k - \rho g_k)$ a, par définition, un minimum en ρ_k, sa dérivée doit être nulle en ρ_k, donc $\nabla f(x_k - \rho_k g_k)^\top g_k = 0$, mais, par un développement limité à l'ordre 1, $\nabla f(x_k - \rho_k g_k) = \nabla f(x_k) - \rho_k H_f(x_k) g_k + \mathrm{o}(\rho_k)$, donc, en approximation, $(g_k - \rho_k H_k g_k)^\top g_k = 0$ ce qui conduit à prendre

$$\rho_k = \frac{\|g_k\|^2}{g_k^\top H_k g_k}$$

qui est ce qu'on a appelé le "pas optimal".

Dans la méthode de Barzilai Borwein, on considère l'itération $x_{k+1} = x_k - \rho_k g_k = x_k - (\rho_k I) g_k$ où I est la matrice identité, et on voudrait (c'est sérieusement gonflé !) pouvoir choisir le pas ρ_k tel que

$$\rho_k I \simeq H_k^{-1}$$

car cela nous rapprocherait de la méthode de Newton, connue pour converger très vite. Bien sûr, cette pseudo-égalité n'a aucun sens car la matrice H_k^{-1} n'est pas diagonale. Pourtant, un peu à la manière des moindres carrés, on va chercher à faire en sorte qu'elle soit vérifiée "au mieux". Ecrivons donc qu'on cherche ρ_k tel que $\rho_k H_k \simeq I$. En multipliant à droite par $s_{k-1} = x_k - x_{k-1}$, on a

$$\rho_k H_k s_{k-1} \simeq s_{k-1}$$

Mais $H_k s_{k-1} \simeq y_{k-1}$, car $y_{k-1} = g_k - g_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}) \simeq H_{k-1}(x_k - x_{k-1}) = H_{k-1} s_{k-1}$ (cette dernière approximation résultant d'un développement limité à l'ordre 1) et $H_{k-1} \simeq H_k$. Ainsi, on obtient

$$\rho_k y_{k-1} \simeq s_{k-1}.$$

Ces considérations heuristiques nous amènent à définir ρ_k par

$$\rho_k = \operatorname*{argmin}_{\rho > 0} \|\rho y_{k-1} - s_{k-1}\|^2$$

En minimisant ce trinôme $\rho^2 \|y_{k-1}\|^2 - 2\rho y_{k-1}^\top s_{k-1} + \|s_{k-1}\|^2$ (on écrit que sa dérivée est nulle en ρ_k), on trouve

$$\boxed{\rho_k = \frac{y_{k-1}^\top s_{k-1}}{\|y_{k-1}\|^2}}$$

La méthode de Barzilai Borwein consiste à faire les itérations $x_{k+1} = x_k - \rho_k g_k$ avec ce choix de pas ρ_k. De manière surprenante, cette méthode s'avère être bien plus efficace que la méthode de gradient à pas optimal ! A ce jour, on n'a toujours pas une explication complète de cette efficacité, en dimension quelconque.

Symétriquement, on peut prendre aussi $\rho_k = \frac{1}{\beta_k}$ où $\beta_k = \operatorname*{argmin}_{\beta > 0} \|y_{k-1} - \beta s_{k-1}\|^2$, ce qui donne la variante

$$\boxed{\rho_k = \frac{\|s_{k-1}\|^2}{s_{k-1}^\top y_{k-1}}}$$

3.4.2.4 Compléments : interprétation EDO

Dans cette section, donnons une interprétation des méthodes en termes d'équations différentielles ordinaires (EDO), qui conduit à de nouvelles variantes.

Pour minimiser f, dans la méthode de Newton, on a cherché à résoudre l'équation $F(x) = \nabla f(x) = 0$. Ecrivons l'EDO

$$x'(t) = F(x(t)) = -\nabla f(x(t)) \tag{3.5}$$

et notons que

$$\frac{d}{dt} f(x(t)) = -\|\nabla f(x(t))\|^2$$

autrement dit, f décroît le long de la trajectoire $x(t)$ solution de (3.5). Dans les conditions de convergence de la méthode de gradient ou de Newton, en fait, la trajectoire $x(t)$ converge, lorsque $t \to +\infty$, vers le minimiseur local x^*. Cela est lié à la théorie de Lyapunov.

Cette observation constitue en fait la version continue des algorithmes de descente qu'on a vus précédemment et donne un point de vue alternatif qui conduit à de nouvelles variantes d'algorithmes.

– En effet, appliquons le schéma de discrétisation d'Euler explicite à (3.5) avec un pas ρ_k : on obtient l'algorithme de descente

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

– La méthode de Newton standard pour résoudre $F(x) = \nabla f(x) = 0$ peut être vue de la manière suivante : on écrit l'approximation à l'ordre 2

$$f(x+h) \simeq f(x) + \nabla f(x)^\top h + \frac{1}{2} h^\top H_f(x) h$$

et on minimise ce trinôme en h, ce qui donne $h = H_f(x)^{-1} \nabla f(x)$. Puis on remplace h par $x_{k+1} - x_k$, ce qui conduit à

$$x_{k+1} = x_k - H_f(x_k)^{-1} \nabla f(x_k)$$

qui est bien la méthode de Newton.

– En appliquant le schéma de discrétisation d'Euler implicite à (3.5) avec un pas ρ_k, on obtient

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

puis, en faisant l'approximation à l'ordre 1 : $\nabla f(x+h) \simeq \nabla f(x) + H_f(x)h$, on obtient

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

donc $x_{k+1} = x_k - \rho_k \nabla f(x_k) - \rho_k H_f(x_k)(x_{k+1} - x_k)$ et donc

$$\boxed{x_{k+1} = x_k - \rho_k (I + \rho_k H_f(x_k))^{-1} \nabla f(x_k)} \tag{3.6}$$

qu'on appelle parfois la méthode d'Euler implicite linéarisée, ou encore modification de Levenberg-Marquardt de la méthode de Newton. On note sur (3.6) que :

— si ρ_k est petit alors $x_{k+1} \simeq x_k - \rho_k \nabla f(x_k)$: méthode de descente ;

— si ρ_k est grand alors $x_{k+1} \simeq x_k - H_f(x_k)^{-1} \nabla f(x_k)$: méthode de Newton. Autrement dit (3.6) réalise un compromis entre la méthode de descente et la méthode de Newton.

La méthode d'Euler implicite linéarisée (3.6) est à comparer à la variante

$$\boxed{x_{k+1} = x_k - \rho_k H_f(x_k)^{-1} \nabla f(x_k)}$$

qui s'appelle la "damped Newton method", car par rapport à la méthode standard de Newton on ajoute le "damping" ρ_k. Cela permet de gérer les cas où la dérivée est quasi-singulière.

3.4.3 Méthode de gradient conjugué

L'algorithme du gradient conjugué ci-dessous est dû à Fletcher et Reeves (1964) :

— Initialisation : on choisit $x_0 \in \mathbb{R}^n$ et $d_0 = -\nabla f(x_0)$.

— Itération k (tant que $\nabla f(x_k) \neq 0$) :

$$\rho_k = -\frac{\nabla f(x_k)^\top d_k}{d_k^\top H_f(x_k) d_k}$$
$$x_{k+1} = x_k + \rho_k d_k$$
$$d_{k+1} = -\nabla f(x_{k+1}) + \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2} d_k$$

— Critère d'arrêt.

Historiquement, la méthode du gradient conjugué a été introduite par Hestenes et Stiefel en 1952. Pour expliquer sa construction, considérons la fonction quadratique

$$f(x) = \frac{1}{2} x^\top A x - b^\top x$$

où A est une matrice carrée de taille n, symétrique définie positive, et $b \in \mathbb{R}^n$. La fonction f a un unique minimiseur $\bar{x} \in \mathbb{R}^n$, qui est caractérisé par $\nabla f(\bar{x}) = 0$, i.e., $A\bar{x} = b$.

Soit $x_0 \in \mathbb{R}^n$ un point initial quelconque. On va construire une suite $(x_k)_{k\in\mathbb{N}}$ ayant certaines propriétés et montrer qu'elle est définie par l'algorithme ci-dessus. Dans toute la suite, on pose

$$\forall k \in \mathbb{N} \qquad g_k = \nabla f(x_k) = Ax_k - b, \qquad s_k = x_{k+1} - x_k.$$

L'idée générale de la méthode du gradient conjugué est de construire la suite $(x_k)_{k\in\mathbb{N}}$ de façon à ce que la suite $(g_k)_{k\in\mathbb{N}}$ soit **orthogonale**, i.e., $g_k \perp g_j$ pour $k \neq j$. En effet, grâce à cette propriété, il existe un entier $K \leqslant n-1$ tel que $g_{K+1} = 0$ (car les $n+1$ vecteurs $g_0, g_1, \ldots, g_n$ ne peuvent pas être linéairement indépendants dans $\mathbb{R}^n$), i.e., $Ax_{K+1} = b$ et donc (par unicité) $x_{K+1} = \bar{x}$. On va voir que cet entier K se caractérise en analysant la suite des espaces de Krylov, définie ci-dessous.

Supposons que $g_0 = Ax_0 - b \neq 0$ (sinon, $x_0 = \bar{x}$ et il n'y a rien à faire). On pose $u_0 = g_0 = Ax_0 - b$ puis on itère par A en posant

$$\forall k \in \mathbb{N}^* \qquad u_k = Au_{k-1} = A^k u_0$$

et on considère la suite croissante des espaces vectoriels (appelés espaces de Krylov)

$$\forall k \in \mathbb{N} \qquad U_k = \mathrm{Vect}(u_0, u_1, \ldots, u_k) = \mathrm{Vect}(u_0, Au_0, \ldots, A^k u_0).$$

Notons que

$$\forall k \in \mathbb{N} \qquad U_k \subset U_{k+1} \qquad \text{et} \qquad AU_k \subset U_{k+1}. \tag{3.7}$$

Soit

$$K = \max\{j \in \mathbb{N} \mid \text{la famille } (u_0, \ldots, u_j) \text{ est libre}\}.$$

On a forcément $K \leqslant n-1$. Par définition de K, u_{K+1} est combinaison linéaire des u_i pour $i \leqslant K$. Donc $u_{K+2} = Au_{K+1}$ est combinaison linéaire des u_i pour $i \leqslant K+1$, donc combinaison linéaire des u_i pour $i \leqslant K$. Et ainsi de suite par récurrence. Par conséquent on a

$$U_0 \subsetneq U_1 \subsetneq \cdots \subsetneq U_{K-1} \subsetneq U_K = U_{K+1} = U_{K+2} = \cdots$$

autrement dit la suite d'espaces vectoriels U_k est strictement croissante jusqu'à $k = K$, puis stationnaire.

Pour tout $k \in \mathbb{N}^*$, soit $x_k \in U_{k-1}$ l'unique minimiseur de f sur le sous-espace affine $x_0 + U_{k-1}$. Comme tout point de U_{k-1} est combinaison linéaire de $u_0, \dots, u_{k-1}$, on a

$$x_k = \operatorname{argmin} \{f(x) \mid x \in x_0 + U_{k-1}\} = \operatorname{argmin} \left\{ f\Big(x_0 + \sum_{j=0}^{k-1} a_j u_j\Big) \quad \middle| \quad a_0, \dots, a_{k-1} \in \mathbb{R} \right\}$$

Nous allons établir que, pour la fonction f quadratique considérée, la suite $(x_k)_{k\in\mathbb{N}}$ converge en exactement $K+1$ itérations, ce qui est une propriété absolument remarquable ! Le problème est de calculer de manière algorithmique les points x_k. On va démontrer que ces points se calculent itérativement par l'algorithme de Fletcher et Reeves donné en début de section. Pour cela, analysons d'abord les propriétés de la suite $(x_k)_{k\in\mathbb{N}}$.

Faisons d'abord les remarques préliminaires suivantes. Comme $x_k \in x_0 + U_{k-1}$, il peut s'écrire $x_k = x_0 + \sum_{j=0}^{k-1} a_j u_j$. Comme $g_k = Ax_k - b$, on a donc

$$g_k = Ax_k - b = \underbrace{Ax_0 - b}_{u_0} + \sum_{j=0}^{k-1} a_j \underbrace{Au_j}_{u_{j+1}} = u_0 + \sum_{j=0}^{k-1} a_j u_{j+1}.$$

Par conséquent,

$$\forall k \in \mathbb{N} \qquad g_k \in U_k. \tag{3.8}$$

Par ailleurs, pour tout $k \in \mathbb{N}$, on a $g_{k+1} = Ax_{k+1} - b = As_k + Ax_k - b$, i.e.,

$$\forall k \in \mathbb{N} \qquad g_{k+1} = g_k + As_k. \tag{3.9}$$

Comme $x_{k+1} \in x_0 + U_k$ et $x_k \in x_0 + U_{k-1} \subset x_0 + U_k$, on en déduit que

$$\forall k \in \mathbb{N} \qquad s_k \in U_k. \tag{3.10}$$

Lemme 2. *(i) La suite $(g_k)_{k\in\mathbb{N}}$ est orthogonale, i.e., $g_k \perp g_j$ pour $k \neq j$, ou de manière équivalente,*

$$\forall k \in \mathbb{N}^* \qquad g_k \perp \operatorname{Vect}(g_0, \dots, g_{k-1}).$$

(ii) La suite $(s_k)_{k\in\mathbb{N}}$ est A-orthogonale (on dit aussi A-conjuguée), i.e., $As_k \perp s_j$ pour $k \neq j$, ou de manière équivalente,

$$\forall k \in \mathbb{N}^* \qquad As_k \perp \operatorname{Vect}(s_0, \dots, s_{k-1}).$$

(iii) On a $x_k \neq \bar{x}$ et (de manière équivalente) $g_k \neq 0$ pour tout $k \leqslant K$ et

$$x_{K+1} = x_{K+2} = \cdots = \bar{x}, \qquad g_{K+1} = g_{K+2} = \cdots = 0.$$

Démonstration. Pour tout $k \in \mathbb{N}^*$, par définition, x_k minimise la fonction $(a_0, \dots, a_{k-1}) \mapsto f\Big(x_0 + \sum_{j=0}^{k-1} a_j u_j\Big)$ sur $\mathbb{R}^k$, donc la différentielle de cette fonction en x_k (du moins, en le k-uplet de coefficients de x_k) est nulle, ce qui donne

$$\forall j \in \{0, \dots, k-1\} \qquad g_k^\top u_j = 0$$

et donc

$$\forall k \in \mathbb{N}^* \qquad g_k \perp U_{k-1}. \tag{3.11}$$

Montrons alors la propriété (i). D'après (3.8), on a $g_j \in U_j$ pour tout $j \in \mathbb{N}$. Soit $k \in \mathbb{N}^*$. Comme $U_j \subset U_{k-1}$ pour $j \leqslant k-1$, on a donc $\mathrm{Vect}(g_0, \dots, g_{k-1}) \subset U_{k-1}$. D'après (3.11), on a $g_k \perp U_{k-1}$. On obtient donc (i).

Montrons maintenant (ii). D'après (3.11), on a $g_{k+1} \perp U_k$ et $g_k \perp U_{k-1}$. Or, pour $j \leqslant k-1$, on a $U_j \subset U_{k-1} \subset U_k$, donc $g_{k+1} \perp U_j$ et $g_k \perp U_j$. Or, d'après (3.9), on a $As_k = g_{k+1} - g_k$, donc $As_k \perp U_j$. Mais, d'après (3.10), $s_j \in U_j$. D'où le résultat.

Montrons enfin (iii). Pour $k = K$, on a vu que $U_{K+1} = U_K$. Mais alors, par (3.11), $g_{K+1} \perp U_K$, et par (3.8), $g_{K+1} \in U_{K+1} = U_K$, on en déduit donc que $g_{K+1} = 0$, i.e., $Ax_{K+1} = b$ et donc (par unicité) $x_{K+1} = \bar{x}$. C'est bien le premier entier pour lequel cette égalité arrive, car pour $k \leqslant K$ la famille $(u_0, \dots, u_k)$ est libre et donc forcément $g_k \neq 0$, i.e., $x_k \neq \bar{x}$. □

D'après le lemme 2, (ii), on a, pour $k \neq j$, $s_k^\top As_j = 0$, or $As_j = g_{j+1} - g_j$ d'après (3.9), donc $s_k^\top(g_{j+1} - g_j) = 0$. Donc, en appliquant cette relation successivement pour $j = k-1, k-2, \dots, 0$, on obtient

$$s_k^\top g_k = s_k^\top g_{k-1} = \cdots = s_k^\top g_0 = \alpha_k$$

où le nombre réel $\alpha_k = s_k^\top g_j$ ainsi défini dépend de k, mais ne dépend pas de $j \in \{0, \dots, k\}$.

Grâce à ces relations, nous allons pouvoir déterminer x_{k+1} en fonction de x_k.

D'après le lemme 2, (i) et (iii), $(g_0, \dots, g_k)$ est une base orthogonale de U_k pour tout $k \leqslant K$. D'autre part, d'après (3.10), $s_k \in U_k$ donc on peut écrire s_k dans cette base :

$$s_k = \sum_{j=0}^{k} \frac{s_k^\top g_j}{\|g_j\|^2} g_j = \alpha_k \sum_{j=0}^{k} \frac{g_j}{\|g_j\|^2}$$

d'où

$$\forall k \in \{1, \dots, K\} \qquad \frac{s_k}{\alpha_k} = \frac{s_{k-1}}{\alpha_{k-1}} + \frac{g_k}{\|g_k\|^2}.$$

En posant

$$\forall k \in \{0, \dots, K\} \qquad d_k = -\frac{\|g_k\|^2}{\alpha_k} s_k = \frac{s_k}{\rho_k}, \qquad \rho_k = -\frac{\alpha_k}{\|g_k\|^2}$$

on obtient bien :

- $s_k = \rho_k d_k$, i.e., $x_{k+1} = x_k + \rho_k d_k$;
- $d_k = \frac{s_k}{\rho_k} = \frac{\alpha_k}{\rho_k}\frac{s_k}{\alpha_k} = \frac{\alpha_k}{\rho_k}\frac{s_{k-1}}{\alpha_{k-1}} + \frac{\alpha_k}{\rho_k}\frac{g_k}{\|g_k\|^2} = \frac{\alpha_k}{\rho_k}\frac{\rho_{k-1}d_{k-1}}{\alpha_{k-1}} - g_k = -g_k + \frac{\|g_k\|^2}{\|g_{k-1}\|^2} d_{k-1}$;

ce qui est l'algorithme écrit au début de la section, mais il reste encore à calculer ρ_k. Pour cela, on note que $g_{k+1} \perp g_k$ d'après le lemme 2, (i), et $g_{k+1} = g_k + As_k$ d'après (3.9), donc $0 = g_k^\top g_{k+1} = \|g_k\|^2 + g_k^\top As_k$, et comme $s_k = \rho_k d_k$, on obtient

$$\forall k \in \{0, \dots, K\} \qquad \rho_k = -\frac{\|g_k\|^2}{g_k^\top Ad_k}.$$

Or, d'après le lemme 2, (ii), la suite $(d_k)_{k\in\mathbb{N}}$ est A-orthogonale (puisque la suite $(s_k)_{k\in\mathbb{N}}$ l'est), et comme $g_k = -d_k + \frac{\|g_k\|^2}{\|g_{k-1}\|^2} d_{k-1}$, on déduit que $g_k^\top Ad_k = -d_k^\top Ad_k$. Par ailleurs, $g_k^\top d_k = -\frac{\|g_k\|^2}{\alpha_k} g_k^\top s_k = -\|g_k\|^2$ car $g_k^\top s_k = \alpha_k$. D'où finalement

$$\forall k \in \{0, \dots, K\} \qquad \rho_k = -\frac{g_k^\top d_k}{d_k^\top Ad_k}$$

ce qui est la formule écrite en début de section. On peut noter qu'on a aussi

$$\rho_k = \frac{\|g_k\|^2}{d_k^\top A d_k}$$

et donc en particulier $\rho_k > 0$.

Conclusion. On a donc montré que, pour une fonction f quadratique, l'algorithme du gradient conjugué de Fletcher et Reeves converge en exactement $K+1 \leqslant n$ itérations.

On utilise cet algorithme pour des fonctions f non quadratiques, en pariant sur le fait que, comme f est approchée à l'ordre deux par une fonction quadratique au voisinage d'un minimiseur $\bar{x}$, on s'attend à avoir un algorithme qui converge assez rapidement. Toutefois, à cause des termes d'ordre supérieur, la convergence n'est plus exacte dans le cas général : il faut donc se donner un critère d'arrêt, comme on l'a vu dans les sections précédentes.

Remarque 36. Notons que le pas ρ_k coïncide avec celui trouvé par la méthode du gradient à pas optimal : en effet, en posant

$$\rho_k = \operatorname*{argmin}_{\rho>0} \underbrace{f(x_k + \rho d_k)}_{\varphi(\rho)}$$

on a $0 = \varphi'(\rho_k) = \nabla f(x_k + \rho_k d_k)^\top d_k \simeq g_k^\top d_k + \rho_k d_k^\top H_k d_k$, d'où $\rho_k = -\frac{g_k^\top d_k}{d_k^\top H_k d_k}$.

Variante de Polak et Ribière (1969). Dans cette variante, on remplace la mise à jour de d_k par

$$\boxed{d_{k+1} = -\nabla f(x_{k+1}) + \frac{\nabla f(x_{k+1})^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}{\|\nabla f(x_k)\|^2} d_k}$$

Lorsque f est quadratique, les deux algorithmes coïncident (car $g_{k+1} \perp g_k$). Toutefois, autant il existe des résultats établissant la convergence de l'algorithme de Fletcher et Reeves, autant on ne sait pas démontrer la convergence de l'algorithme de Polak et Ribière pour des larges classes de fonctions !

L'algorithme du gradient conjugué de Polak et Ribière est le plus utilisé dans la pratique car on constate qu'il converge et est plus performant pour de plus grandes classes de fonctions que la version de Fletcher et Reeves (bien qu'on ne sache pas expliquer pourquoi, même 50 ans après sa découverte !). Vous voyez donc que, même sur des algorithmes classiques, on a encore des marges d'amélioration et de découvertes potentielles...

Dans les deux algorithmes, pour être sûr d'avoir une direction de descente raisonnable, on peut ajouter le test :

$$\frac{d_{k+1}^\top \nabla f(x_{k+1})}{\|d_{k+1}\| \|\nabla f(x_{k+1})\|} \leqslant -\alpha < 0$$

où $\alpha > 0$ est fixé, pas trop petit ; et si cela n'est pas vérifié, on prend simplement $d_{k+1} = -\nabla f(x_{k+1})$.

3.4.4 Conclusion

Dans cette section 3.4, on a vu diverses méthodes pour minimiser une fonction $f : \mathbb{R}^n \to \mathbb{R}$ suffisamment régulière. Les méthodes de type gradient consistent à suivre les lignes de gradient le long des itérations : $x_{k+1} = x_k - \rho_k \nabla f(x_k)$, où le pas ρ_k peut être astucieusement choisi. Dans les méthodes de type Newton, l'itération s'écrit $x_{k+1} = x_k - M_k \nabla f(x_k)$ où M_k est une matrice bien

choisie. Dans l'approche de gradient conjugué, la direction de descente est une combinaison linéaire de $\nabla f(x_k)$ et de $\nabla f(x_{k-1})$, astucieusement choisie de façon à assurer des propriétés d'orthogonalité.

On peut combiner ces méthodes (comme on l'a vu dans les interprétations EDO) et en imaginer d'autres, selon la classe de problèmes considérée : à vous d'être créatifs !

Chapitre 4

Minimisation sous contraintes

Dans ce chapitre on étudie les problèmes de minimisation sous contraintes

$$\boxed{\min_{x \in C} f(x)}$$

où $f : E \to \mathbb{R} \cup \{+\infty\}$ est une fonction sur un espace vectoriel E, et $C \subset E$ est un sous-ensemble non vide de E.

Comme dans le chapitre précédent, en vue de l'implémentation numérique, c'est le cas $E = \mathbb{R}^n$ qui nous intéresse en priorité. Toutefois, on donnera des énoncés généraux chaque fois qu'on le pourra, en gardant les mêmes notations que dans les chapitres précédents : le plus souvent, E désigne un espace vectoriel normé et H un espace de Hilbert.

Dans le cas où $E = \mathbb{R}^n$, l'ensemble de contraintes C sera, le plus souvent, soit un sous-ensemble convexe fermé non vide, soit un sous-ensemble défini par des égalités et des inégalités de fonctions suffisamment régulières : $C = \{x \in \mathbb{R}^n \ \mid \ g(x) = 0,\ h(x) \leqslant 0\}$.

4.1 Existence et unicité

Les énoncés qui suivent, et leurs arguments de preuve, font écho au chapitre précédent, section 3.1. On va donc vite.

Théorème 23. *Soit E un espace vectoriel topologique, soit $f : E \to \mathbb{R} \cup \{+\infty\}$ une fonction continue, et soit $C \subset E$ un sous-ensemble compact non vide. Alors le problème*

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

admet au moins un minimiseur (global), i.e., il existe au moins un point $x^* \in C$ tel que $f(x^*) = \min_{x \in C} f(x)$. On note aussi $x^* = \operatorname{argmin}_{x \in C} f(x)$.

Lorsque $E = \mathbb{R}^n$, et lorsque C est seulement fermé non vide (mais pas forcément borné), la conclusion est encore vraie si de plus f est infinie à l'infini.

On n'a pas unicité du minimiseur en général (on parle donc de minimiseur local et de minimiseur global). Pour avoir unicité, on peut supposer la convexité.

Théorème 24. *Soit E un espace vectoriel et $C \subset E$ un sous-ensemble non vide. Si $f : E \to \mathbb{R}$ est strictement convexe alors elle a au plus un minimiseur (global) sur C.*

4.2 Conditions d'optimalité

4.2.1 Conditions d'optimalité du premier ordre sur un ensemble convexe

Théorème 25. *Soit E un espace vectoriel normé, soit $C \subset E$ un sous-ensemble convexe fermé non vide, et soit $f : E \to \mathbb{R}$ une fonction Gateaux différentiable. Si $x^* \in C$ est un minimiseur (local ou global) de f alors*

$$\forall x \in C \qquad df(x^*).(x - x^*) \geqslant 0$$

(on devrait plutôt écrire : $f'(x^*; x - x^*) = 0$ pour tout $x \in E$). Dans le cas où f est convexe, la condition est nécessaire et suffisante (et x^* est un minimiseur global).

Lorsque E est un Hilbert et f est différentiable, cette condition s'écrit

$$\forall x \in C \qquad \langle \nabla f(x^*), x - x^* \rangle \geqslant 0.$$

Démonstration. Par convexité de C, $x^* + t(x - x^*) \in C$ pour tout $t \in [0, 1]$, et pour $t > 0$ assez petit on a $f(x^* + t(x - x^*)) - f(x^*) \geqslant 0$ car x^* est un minimiseur au moins local. On divise par t et on fait tendre t vers 0 pour obtenir la condition nécessaire.

Montrons qu'elle est suffisante si de plus f est convexe. Si f est convexe alors le graphe de f est au-dessus de ses tangentes, donc $f(x) \geqslant f(x^*) + df(x^*).(x - x^*)$, et donc, on en déduit que $f(x) \geqslant f(x^*)$, donc x^* est un minimiseur (global). □

Les conditions ci-dessus sont générales, sur un ensemble convexe, mais restent abstraites. Dans la section suivante, on traite le cas $E = \mathbb{R}^n$ et un ensemble de contraintes général non convexe, défini en termes d'égalités et d'inégalités de fonctions. Cela conduit à la notion de multiplicateurs de Lagrange.

4.2.2 Conditions d'optimalité du premier ordre : multiplicateurs de Lagrange

4.2.2.1 Contraintes d'égalité

Considérons le problème d'optimisation

$$\boxed{\min_{h(x)=0} f(x)}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ et $h : \mathbb{R}^n \to \mathbb{R}^p$ sont C^1. Ici, on a $h = (h_1, \ldots, h_p)$. On a donc un problème d'optimisation avec p contraintes d'égalité.

On a le théorème des multiplicateurs de Lagrange :

Théorème 26. *Si x^* est un minimiseur alors il existe $\lambda_0 \in \mathbb{R}$ et $\lambda = (\lambda_1, \ldots, \lambda_p) \in \mathbb{R}^p$, avec $(\lambda_0, \lambda) \neq 0$, tels que*

$$\boxed{\lambda_0 \nabla f(x^*) + \sum_{i=1}^{p} \lambda_i \nabla h_i(x^*) = 0}$$

Si de plus les gradients $\nabla h_i(x^*)$, $i = 1, \ldots, p$ sont indépendants (condition de qualification) alors on peut de plus choisir $\lambda_0 = 1$.

Remarque 37. En supposant que λ est un vecteur colonne, et en identifiant $dh(x^*)$ à la matrice jacobienne, la condition des multiplicateurs de Lagrange s'écrit aussi sous la forme

$$\boxed{\lambda_0 \, df(x^*)^\top + dh(x^*)^\top \lambda = 0}$$

Démonstration. La preuve est de nature géométrique et utilise le théorème des fonctions implicites. On définit l'application "augmentée" $F : \mathbb{R}^n \to \mathbb{R}^p \times \mathbb{R}$ par

$$F(x) = (h(x), f(x)).$$

Sur un dessin, représentons l'image de F, i.e., l'ensemble $F(\mathbb{R}^n)$: voir figure 4.1.

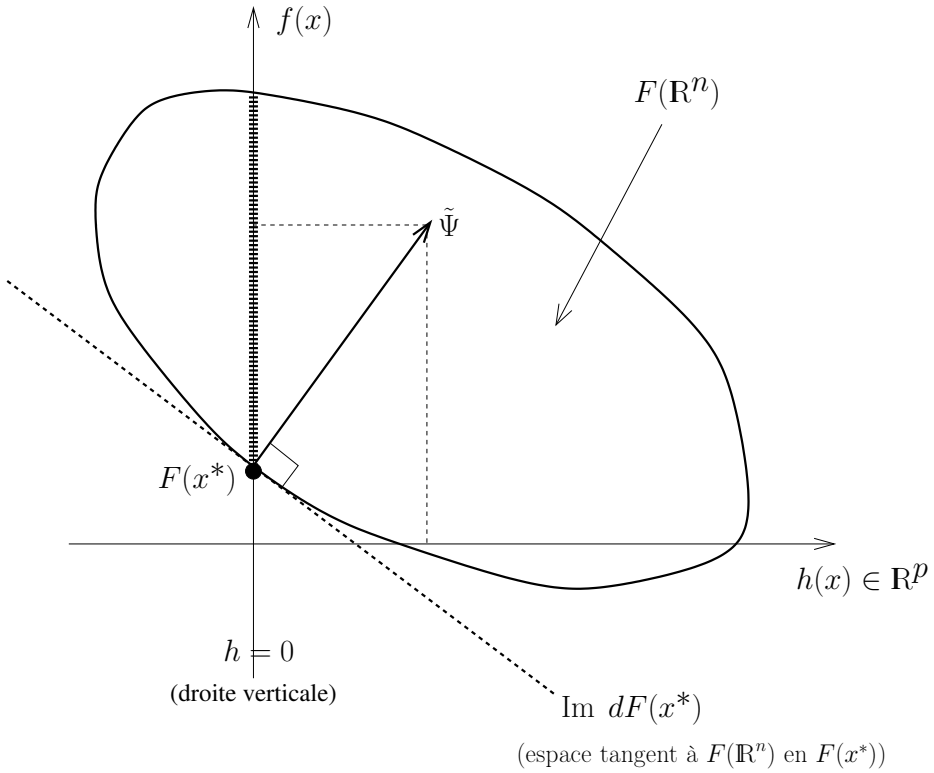


FIGURE 4.1 – Image de l'application augmentée F, et visualisation de la solution optimale : $F(x^*)$ est **au bord** de l'ensemble $F(\mathbb{R}^n)$.

On veut minimiser f sur l'ensemble $h = 0$ qui, sur le dessin, correspond à l'axe des ordonnées : on minimise donc f sur la "fibre" au-dessus de 0, qui intersecte l'ensemble $F(\mathbb{R}^n)$. Le minimum est obtenu au point le plus bas possible de cette fibre : c'est le point $F(x^*)$ qui est noirci sur la figure.

Le point clé est de remarquer que, nécessairement, $F(x^*)$ est au bord de l'ensemble $F(\mathbb{R}^n)$:

$$\boxed{F(x^*) \in \partial\,(F(\mathbb{R}^n))}$$

Ce point clé est le véritable "secret" de l'optimisation : dans tout problème d'optimisation, on a, d'une manière ou d'une autre, cette propriété "d'être au bord". En effet, être au bord, cela veut dire qu'on ne peut pas faire mieux !

Notons deux choses : d'une part ce qu'on dit ici ne préjuge d'aucune considération spécifique sur l'ensemble $F(\mathbb{R}^n)$ (qu'on ne suppose pas fermé, ouvert, ou quoi que ce soit) ; tout ce qu'on suppose c'est qu'un minimiseur existe, et alors, forcément, le point $F(x^*)$ est au bord de l'ensemble $F(\mathbb{R}^n)$. D'autre part, la réciproque est fausse : être au bord de l'ensemble ne signifie pas qu'on a

un minimum ; on pourrait avoir un maximum, ou bien, rien du tout (ni minimum ni maximum). N'oublions pas qu'on cherche ici une condition nécessaire d'optimalité.

Maintenant arrive l'étape du théorème des fonctions implicites : la fonction F est C^1 et, à cause de la propriété de bord ci-dessus, la fonction F n'est pas localement surjective en x^* (sinon, il existerait une petite boule ouverte autour du point $F(x^*)$ qui serait contenue dans $F(\mathbb{R}^n)$: mais c'est faux ! cf figure de nouveau). Donc, par contraposée du théorème des fonctions implicites (plus précisément, par contraposée du théorème de la submersion), la différentielle $dF(x^*) : \mathbb{R}^n \to \mathbb{R}^p \times \mathbb{R}$ n'est pas surjective, i.e.,

$$\text{Im } dF(x^*) \subsetneq \mathbb{R}^p \times \mathbb{R}$$

Comme on est en dimension finie, le sous-espace vectoriel strict Im $dF(x^*)$ est donc contenu dans un hyperplan [1], et donc il existe un vecteur $\tilde{\lambda} \in \mathbb{R}^p \times \mathbb{R} \setminus \{0\}$ (non trivial ! sinon on ne dit rien...) tel que $\tilde{\lambda} \perp \text{Im } dF(x^*)$, i.e., en notant $\tilde{\lambda}$ comme vecteur colonne,

$$dF(x^*)^\top \tilde{\lambda} = 0.$$

Mais comme $dF(x) = (df(x), dh(x))$, en posant $\tilde{\lambda} = (\lambda, \lambda_0)$ avec $\lambda \in \mathbb{R}^p$ et $\lambda_0 \in \mathbb{R}$, on en déduit la relation des multiplicateurs de Lagrange.

Sous la condition de qualification, on a forcément $\lambda_0 \neq 0$, car sinon, si $\lambda_0 = 0$ on aurait une relation de dépendance linéaire entre les gradients $\nabla h_i(x^*)$. On peut alors normaliser le multiplicateur de Lagrange de sorte que $\lambda_0 = 1$, grâce à la remarque ci-dessous. □

Remarque 38. Il est important de noter que le multiplicateur de Lagrange $\tilde{\lambda} = (\lambda, \lambda_0)$ construit ci-dessus, d'une part, est non trivial (sinon on écrit $0 = 0$! donc rien...), et d'autre part, est défini à scalaire multiplicatif près, autrement dit, pour tout $\alpha \neq 0$, $\alpha\tilde{\lambda}$ est aussi un multiplicateur de Lagrange.

Le réel λ_0 est appelé multiplicateur de Lagrange associé au coût.

Pour $i \in \{1, \ldots, p\}$, le réel λ_i est appelé multiplicateur de Lagrange associé à la contrainte $h_i = 0$.

On a deux cas possibles :

- Si $\lambda_0 \neq 0$, quitte à multiplier le multiplicateur de Lagrange $\tilde{\lambda}$ par $1/\lambda_0$, on peut supposer que $\lambda_0 = 1$. Ce cas s'appelle le *cas normal*.
- Mais il se peut que $\lambda_0 = 0$: on appelle ce cas le *cas anormal*.

Bien entendu, le cas anormal n'arrive pas sous la condition de qualification. Mais sinon, il pourrait arriver : par exemple considérons le problème d'optimisation (certes trivial) avec $n = 1$, $f(x) = x$ et $h(x) = x^2$; alors on a un cas anormal. On comprend ici que cela vient du fait que l'ensemble $h = 0$ est un point isolé. Dans le cas général, cela arrive lorsque l'ensemble $h = 0$ n'est pas une sous-variété de $\mathbb{R}^n$ au point x^* (point singulier).

Remarque 39. La condition de qualification s'exprime de manière équivalente en disant que la différentielle $dh(x^*) : \mathbb{R}^n \to \mathbb{R}^p$ est surjective.

Sous cette condition, le théorème des fonctions implicites implique que l'ensemble des contraintes $h = 0$ est une sous-variété de $\mathbb{R}^n$ (de codimension p).

C'est le sens des conditions de qualification : lorsque l'ensemble de contraintes est une sous-variété, alors il existe un multiplicateur de Lagrange normal ; mais lorsque l'ensemble de contraintes a un point singulier, on peut avoir un multiplicateur de Lagrange anormal.

Il faut prendre garde et ne pas oublier les multiplicateurs anormaux, lorsqu'on recherche les solutions optimales !

1. Notons que ce fait pourrait échouer en dimension infinie : on peut avoir un sous-espace vectoriel strict qui soit partout dense ! En dimension infinie, il faut alors trouver des hypothèses qui impliquent que Im $dF(x^*)$ est de plus fermé ; auquel cas l'argument de séparation marche encore...

Exemple 8. Voici un exemple où le multiplicateur de Lagrange est anormal. Prenons $n = 2$ et

$$f(x, y) = x + y, \qquad h_1(x, y) = (x-1)^2 + y^2 - 1, \qquad h_2(x, y) = (x-2)^2 + y^2 - 4.$$

L'ensemble $h_1 = 0$ est le cercle de centre $(1, 0)$ et de rayon 1. L'ensemble $h_2 = 0$ est le cercle de centre $(2, 0)$ et de rayon 2. Donc, l'ensemble $h = 0$, qui est leur intersection, est le singleton $\{(0, 0)\}$. Bien sûr, le minimum de f est alors $f(0, 0) = 0$! Mais appliquons tout de même la règle des multiplicateurs de Lagrange au point $x^* = (0, 0)$:

$$\lambda_0 \nabla f(0, 0) = \lambda_1 \nabla h_1(0, 0) + \lambda_2 \nabla h_2(0, 0)$$

ce qui donne, vu que $\nabla f(0, 0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\nabla h_1(0, 0) = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$, $\nabla h_2(0, 0) = \begin{pmatrix} -4 \\ 0 \end{pmatrix}$:

$$\lambda_0 = 0, \qquad \lambda_1 = -2\lambda_2.$$

Il est normal qu'il reste un degré de liberté, vu que le multiplicateur de Lagrange est défini à scalaire multiplicatif près. On peut prendre par exemple $\lambda_0 = 0$, $\lambda_1 = -2$, $\lambda_2 = 1$. Quoi qu'il en soit, il est anormal.

Formulation Lagrangienne. On peut formuler la condition de multiplicateurs de Lagrange sous la forme suivante. On définit la fonction $L : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^p$ (Lagrangien) par

$$L(x, \lambda_0, \lambda) = \lambda_0 f(x) + \lambda^\top h(x) = \lambda_0 f(x) + \sum_{i=1}^{p} \lambda_i h_i(x)$$

Avec cette fonction, la condition de multiplicateurs de Lagrange s'écrit

$$\boxed{\frac{\partial L}{\partial x}(x^*, \lambda_0, \lambda) = 0}$$

autrement dit, x^* est un point extrémal de $L(\cdot, \lambda_0, \lambda)$. Voir la section suivante pour plus d'éléments à ce sujet.

Lorsqu'on a un multiplicateur de Lagrange normal (i.e., $\lambda_0 > 0$), qu'on normalise à $\lambda_0 = 1$, souvent on note

$$L(x, \lambda) = L(x, 1, \lambda) = f(x) + \sum_{i=1}^{p} \lambda_i h_i(x).$$

4.2.2.2 Contraintes d'égalité et d'inégalité : conditions KKT

Considérons le problème d'optimisation

$$\boxed{\min_{\substack{h(x)=0 \\ g(x) \leqslant 0}} f(x)}$$

où $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^q$ et $h : \mathbb{R}^n \to \mathbb{R}^p$ sont C^1. Ici, la notation $g(x) \leqslant 0$ signifie que $g_j(x) \leqslant 0$ pour tout $j = 1, \ldots, q$.

Théorème de Karush-Kuhn-Tucker (KKT).

Théorème 27. *Si x^* est un minimiseur alors il existe $\lambda_0 \in \mathbb{R}$, $\lambda = (\lambda_1, \ldots, \lambda_p) \in \mathbb{R}^p$ et $\mu = (\mu_1, \ldots, \mu_q) \in \mathbb{R}^q$, avec $(\lambda_0, \lambda, \mu) \neq 0$ (multiplicateurs de Lagrange) tels que*

$$\boxed{\begin{array}{l} \lambda_0 \nabla f(x^*) + \displaystyle\sum_{i=1}^{p} \lambda_i \nabla h_i(x^*) + \sum_{j=1}^{q} \mu_j \nabla g_j(x^*) = 0 \\ h(x^*) = 0, \quad g(x^*) \leqslant 0 \\ \lambda_0 \geqslant 0, \qquad \mu_j \geqslant 0 \quad \forall j \in \{1, \ldots, q\} \\ \mu_j g_j(x^*) = 0 \quad \forall j \in \{1, \ldots, q\} \qquad \textit{(conditions de complémentarité)} \end{array}}$$

Si de plus les gradients $\nabla h_i(x^*)$, $i \in \{1, \ldots, p\}$, $\nabla g_j(x^*)$, $j \in I(x^*)$ (indices des contraintes actives) sont tous linéairement indépendants (famille libre), alors on peut de plus supposer que $\lambda_0 = 1$ ci-dessus. On parle de condition de qualification.

La contrainte $g_j \leqslant 0$ est dit active (ou saturée) si $g_j(x^*) = 0$, et inactive (ou non saturée) si $g_j(x^*) < 0$. On note $I(x^*)$ l'ensemble des indices actifs, i.e., des indices correspondant à une contrainte active.

Les conditions de complémentarité sont une manière équivalente de dire que, pour $j \in \{1, \ldots, q\}$, si la contrainte $g_j(x^*)$ est inactive en x^*, i.e., si $g_j(x^*) < 0$, alors $\mu_j = 0$: le multiplicateur de Lagrange correspondant est nul. Au contraire lorsque la contrainte est active, i.e., si $g_j(x^*) = 0$, alors la relation $\mu_j g_j(x^*) = 0$ est bien vérifiée (on pourrait avoir malgré tout $\mu_j = 0$, mais en général on aura $\mu_j > 0$).

Donnons deux démonstrations du théorème KKT.

Première démonstration de KKT. Le problème ci-dessus est équivalent à un problème d'optimisation comportant uniquement des contraintes d'égalité : ce sont les contraintes $h_i(x) = 0$ d'une part, avec $i = 1, \ldots, q$, et d'autre part les contraintes $g_j(x) = 0$ pour les indices actifs $j \in I(x^*)$.

Bien sûr, comme on ne connaît pas à l'avance le minimiseur x^*, on ne sait pas non plus à l'avance quelles sont les contraintes actives ! Tout cela est théorique.

Mais il n'empêche que, de fait, x^* est aussi une solution optimale du problème avec contraintes d'égalité

$$\min_{H(x)=0} f(x) \tag{4.1}$$

où $H : \mathbb{R}^n \to \mathbb{R}^p \times \mathbb{R}^{\mathrm{card}(I(x^*))}$ est donné par

$$H(x) = \begin{pmatrix} h(x) \\ g_j(x),\ j \in I(x^*) \end{pmatrix} \tag{4.2}$$

Notons que le problème est qualifié si et seulement si l'application linéaire $dH(x^*) : \mathbb{R}^n \to \mathbb{R}^p \times \mathbb{R}^{\mathrm{card}(I(x^*))}$ est surjective (condition sous laquelle le théorème des fonctions implicites implique que l'ensemble $\{x \in \mathbb{R}^n \mid H(x) = 0\}$ est une sous-variété de $\mathbb{R}^n$).

Par le théorème des multiplicateurs de Lagrange, il existe donc $\lambda_0 \in \mathbb{R}$ et un vecteur colonne $\Psi \in \mathbb{R}^p \times \mathbb{R}^{\mathrm{card}(I(x^*))}$, de coordonnées successives $\lambda_1, \ldots, \lambda_p, (\mu_j)_{j \in I(x^*)}$, tels que

$$\lambda_0\, df(x^*)^\top + dH(x^*)^\top \Psi = 0$$

ce qui donne exactement l'égalité KKT du théorème et les conditions de complémentarité.

Finalement, la seule nouveauté dans cet énoncé est la condition de signe $\lambda_0 \geqslant 0$ et $\mu_j \geqslant 0$ pour $j = 1, \ldots, q$. Mais en fait cette condition de signe peut se voir géométriquement, en reprenant la

preuve du théorème des multiplicateurs de Lagrange avec l'application augmentée qui, cette fois, est définie par $F(x) = (h(x), g(x), f(x))$. Sur un dessin, représentons, de même, l'image de F, i.e., l'ensemble $F(\mathbb{R}^n)$. Sur la figure 4.2, on représente la projection sur $\mathbb{R}^q \times \mathbb{R}$, autrement dit, la partie $(g(x), f(x))$. Sur ce dessin, l'ensemble où est recherché le minimum de f est l'ensemble hâchuré (ce n'est plus seulement une fibre comme précédemment, puisqu'on minimise sur $g \leqslant 0$).

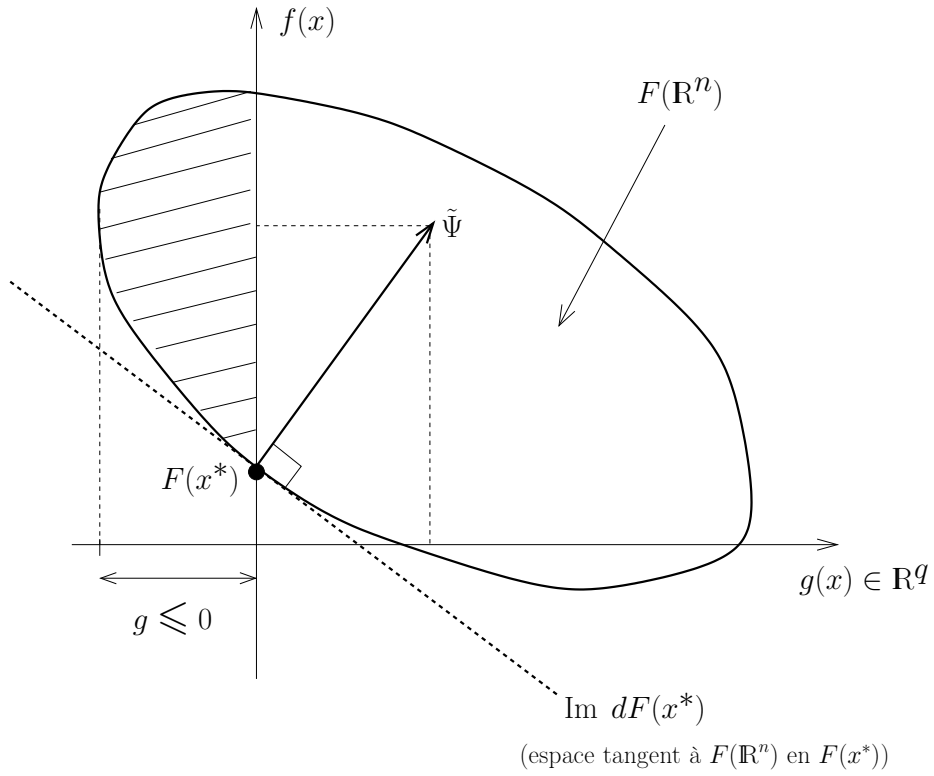


FIGURE 4.2 – Image de l'application augmentée $F(x) = (h(x), g(x), f(x))$ (on ne représente pas les coordonnées h), et visualisation de la solution optimale : $F(x^*)$ est au bord de l'ensemble $F(\mathbb{R}^n)$.

Dans la preuve des multiplicateurs de Lagrange, l'argument consiste à dire qu'il existe $\tilde{\Psi} \in \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}$ non trivial tel que $\tilde{\Psi} \perp \text{Im } dF(x^*)$, i.e., $dF(x^*)^\top \tilde{\Psi} = 0$. En posant ensuite $\tilde{\Psi} = (\lambda, \mu, \lambda_0)$, on trouve la relation des multiplicateurs de Lagrange. La considération de signe se voit alors très bien sur la figure 4.2 : par construction, le vecteur $\tilde{\Psi}$ est orthogonal à Im $dF(x^*)$ qui est l'hyperplan tangent à l'ensemble $F(\mathbb{R}^n)$ au point $F(x^*)$. On a deux orientations possibles pour ce vecteur $\tilde{\Psi}$: on le choisit ici de sorte qu'il pointe vers le haut, i.e., $\lambda_0 \geqslant 0$. On voit alors que forcément $\mu_j \geqslant 0$: il doit aussi pointer vers la droite, dans ses composantes correspondant à g_j (pour se convaincre, refaire un dessin avec un $\tilde{\Psi}$ qui pointerait vers la gauche : sur le dessin, la pente de la droite représentant Im $dF(x^*)$ serait positive et alors le point $F(x^*)$ ne serait plus au même endroit !). □

La preuve ci-dessus est une preuve par "argument de séparation", en conservant des considérations de signe.

Notons bien qu'on aurait pu prendre la convention opposée, et choisir $\lambda_0 \leqslant 0$: dans ce cas on aurait aussi $\mu_j \leqslant 0$.

Il existe d'autres preuves, comme celle ci-dessous qui est intéressante car elle ouvre la voie aux techniques de pénalisation (qu'on verra plus loin) :

Deuxième démonstration de KKT. Soit $R > 0$ quelconque, on note $\bar{B}(x^*, R)$ la boule fermée de $\mathbb{R}^n$ de centre x^* et de rayon R. Pour tout $\varepsilon > 0$, on définit la "fonction pénalisée" :

$$f_\varepsilon(x) = f(x) + \frac{1}{2\varepsilon}\sum_{i=1}^{p} h_i(x)^2 + \frac{1}{2\varepsilon}\sum_{j=1}^{q} \max(g_j(x), 0)^2 + \|x - x^*\|^2$$

L'idée ici est que, si on prend x tel que $h_i(x) \neq 0$ alors le terme $\frac{1}{2\varepsilon}h_i(x)^2$ devient très grand lorsque ε est petit. De même, si x est tel que $g_j(x) > 0$ alors le terme $\frac{1}{2\varepsilon}\max(g_j(x), 0)^2$ devient très grand lorsque ε est petit. On s'attend donc à ce que, lorsque $\varepsilon \to 0$, "le" (ou, "un") minimiseur x_ε de f_ε tend vers un point vérifiant les contraintes. Le terme supplémentaire $\|x - x^*\|^2$ force ce point à être égal au point désiré x^*.

Avec cette idée, faisons alors rigoureusement la preuve. On considère le problème

$$\min_{x \in \bar{B}(x^*, R)} f_\varepsilon(x).$$

Tout d'abord, ce problème admet au moins un minimiseur $x_\varepsilon \in \bar{B}(x^*, R)$, car f_ε est continue sur le compact $\bar{B}(x^*, R)$.

Comme $x_\varepsilon \in \bar{B}(x^*, R)$, à sous-suite près on peut supposer que $x_\varepsilon \to \bar{x} \in \bar{B}(x^*, R)$ lorsque $\varepsilon \to 0$. On va démontrer que, en fait, $\bar{x} = x^*$.

Tout d'abord, comme x_ε minimise f_ε, on a $f_\varepsilon(x_\varepsilon) \leqslant f_\varepsilon(x^*)$, et par ailleurs, comme $h(x^*) = 0$ et $g(x^*) \leqslant 0$, on observe que $f_\varepsilon(x^*) = f(x^*)$. Ainsi, on a $f_\varepsilon(x_\varepsilon) \leqslant f(x^*)$, et donc

$$\sum_{i=1}^{p} h_i(x_\varepsilon)^2 + \sum_{j=1}^{q} \max(g_j(x_\varepsilon), 0)^2 \leqslant 2\varepsilon \left(f(x^*) - f(x_\varepsilon) - \|x_\varepsilon - x^*\|^2\right) \leqslant 2\varepsilon M$$

pour un $M > 0$ car l'expression entre parenthèses est bornée (on est sur un compact). Donc, lorsque $\varepsilon \to 0$, on a $h_i(x_\varepsilon) \to 0$ et $\max(g_j(x_\varepsilon), 0) \to 0$. En passant à la limite, comme $x_\varepsilon \to \bar{x}$ on obtient donc

$$h_i(\bar{x}) = 0 \quad \forall i \in \{1, \dots, p\} \qquad\qquad g_j(\bar{x}) \leqslant 0 \quad \forall j \in \{1, \dots, q\}$$

autrement dit, $\bar{x}$ vérifie les contraintes.

Par ailleurs, par définition de f_ε on a $f(x_\varepsilon) + \|x_\varepsilon - x^*\|^2 \leqslant f_\varepsilon(x_\varepsilon)$ (puisqu'on ajoute des carrés), et on a vu que $f_\varepsilon(x_\varepsilon) \leqslant f(x^*)$, donc $f(x_\varepsilon) + \|x_\varepsilon - x^*\|^2 \leqslant f(x^*)$, et en passant à la limite on obtient $f(\bar{x}) + \|\bar{x} - x^*\|^2 \leqslant f(x^*)$. Mais comme x^* est un minimiseur du problème contraint et que $\bar{x}$ est un point qui vérifie les contraintes, on doit forcément avoir $f(x^*) \leqslant f(\bar{x})$, ce qui donne donc finalement $f(\bar{x}) + \|\bar{x} - x^*\|^2 \leqslant f(\bar{x})$ et donc $\bar{x} = x^*$.

On note que, comme ce raisonnement a été fait pour toute sous-suite convergente de x_ε, finalement, $x_\varepsilon \to x^*$ (pas seulement à sous-suite près).

Ecrivons maintenant les conditions nécessaires d'optimalité pour f_ε. Comme x_ε minimise f_ε sur la boule $\bar{B}(x^*, R)$, et comme x_ε est dans l'intérieur de la boule lorsque ε est assez petit (puisque $x_\varepsilon \to x^*$), la condition nécessaire d'optimalité est

$$\nabla f_\varepsilon(x_\varepsilon) = 0$$

pour tout $\varepsilon > 0$ assez petit, ce qui donne

$$\nabla f(x_\varepsilon) + \sum_{i=1}^{p} \frac{h_i(x_\varepsilon)}{\varepsilon} \nabla h_i(x_\varepsilon) + \sum_{j=1}^{q} \frac{\max(g_j(x_\varepsilon), 0)}{\varepsilon} \nabla g_j(x_\varepsilon) + 2(x_\varepsilon - x^*) = 0.$$

(notons ici qu'on a utilisé la formule $\frac{d}{dx} \max(x, 0)^2 = 2 \max(x, 0)$, facile à montrer)

Cette égalité commence à ressembler à la relation de multiplicateurs de Lagrange ! En effet si on pose

$$\lambda_i^\varepsilon = \frac{h_i(x_\varepsilon)}{\varepsilon}, \qquad \mu_j^\varepsilon = \frac{\max(g_j(x_\varepsilon), 0)}{\varepsilon}$$

on a

$$\nabla f(x_\varepsilon) + \sum_{i=1}^{p} \lambda_i^\varepsilon \nabla h_i(x_\varepsilon) + \sum_{j=1}^{q} \mu_j^\varepsilon \nabla g_j(x_\varepsilon) + 2(x_\varepsilon - x^*) = 0.$$

On veut maintenant faire tendre ε vers 0 dans cette égalité. Pour faire cela, on pose

$$\psi^\varepsilon = (1, \lambda_1^\varepsilon, \dots, \lambda_p^\varepsilon, \mu_1^\varepsilon, \dots, \mu_q^\varepsilon) \in \mathbb{R} \times \mathbb{R}^p \times \mathbb{R}^q$$

et on divise l'égalité ci-dessus par $\|\psi^\varepsilon\|$:

$$\frac{1}{\|\psi^\varepsilon\|} \nabla f(x_\varepsilon) + \sum_{i=1}^{p} \frac{\lambda_i^\varepsilon}{\|\psi^\varepsilon\|} \nabla h_i(x_\varepsilon) + \sum_{j=1}^{q} \frac{\mu_j^\varepsilon}{\|\psi^\varepsilon\|} \nabla g_j(x_\varepsilon) + \frac{2}{\|\psi^\varepsilon\|}(x_\varepsilon - x^*) = 0.$$

La famille de vecteurs $\frac{\psi^\varepsilon}{\|\psi^\varepsilon\|}$ est de norme 1, donc à sous-suite près, elle converge lorsque $\varepsilon \to 0$ vers un vecteur $(\bar\lambda_0, \bar\lambda, \bar\mu) \in \mathbb{R} \times \mathbb{R}^p \times \mathbb{R}^q$ de norme 1. En notant que $\|\psi^\varepsilon\| \geqslant 1$ et en passant à la limite on a donc

$$\bar\lambda_0 \nabla f(x^*) + \sum_{i=1}^{p} \bar\lambda_i \nabla h_i(x^*) + \sum_{j=1}^{q} \bar\mu_j \nabla g_j(x^*) = 0$$

ce qui est la relation des multiplicateurs de Lagrange, et de plus par la construction ci-dessus on a bien obtenu $\bar\mu_j \geqslant 0$, ce qui était la chose nouvelle à obtenir.

Notons que si $g_j(x^*) < 0$ (contrainte inactive) alors $g_j(x_\varepsilon) < 0$ pour ε assez petit et donc $\mu_j^\varepsilon = 0$ et donc $\bar\mu_j = 0$ par passage à la limite.

Comme dans la preuve des multiplicateurs de Lagrange, si les gradients sont indépendants alors $\bar\lambda_0 \neq 0$ (par l'absurde). □

Formulation Lagrangienne. On peut formuler les conditions KKT sous la forme suivante. On définit la fonction $L : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^p \times \mathbb{R}^q$ (Lagrangien) par

$$L(x, \lambda_0, \lambda, \mu) = \lambda_0 f(x) + \sum_{i=1}^{p} \lambda_i h_i(x) + \sum_{j=1}^{q} \mu_j g_j(x).$$

Avec cette fonction, la condition KKT s'écrit

$$\boxed{\frac{\partial L}{\partial x}(x^*, \lambda_0, \lambda, \mu) = 0}$$

autrement dit, x^* est un point extrémal de $L(\cdot, \lambda_0, \lambda, \mu)$. En fait, sous des conditions de convexité, x^* minimise cette fonction (ce qui ouvre aux méthodes dites de dualité qu'on va voir plus loin) :

Proposition 4. *Par rapport au théorème KKT, on suppose de plus que f est convexe, que les g_j sont convexes et les h_i sont affines. On suppose aussi que le problème est qualifié. Alors on a la réciproque : autrement dit, x^* est un minimiseur si et seulement si on a les conditions KKT.*

Démonstration. Avec les hypothèses de convexité, L est convexe par rapport à x (notons que $\mu_j \geqslant 0$), donc la condition $\frac{\partial L}{\partial x}(x^*, \lambda_0, \lambda, \mu) = 0$ est équivalente au fait que x^* minimise la fonction $L(\cdot, \lambda_0, \lambda, \mu)$:

$$L(x^*, \lambda_0, \lambda, \mu) \leqslant L(x, \lambda_0, \lambda, \mu) \qquad \forall x \in \mathbb{R}^n. \tag{4.3}$$

Soit alors $x \in \mathbb{R}^n$ un point vérifiant les contraintes $h(x) = 0$ et $g(x) \leqslant 0$. Comme $\mu_j \geqslant 0$, on a donc $\sum_{i=1}^p \lambda_i h_i(x) + \sum_{j=1}^q \mu_j g_j(x) \leqslant 0$, ce qui implique que $L(x, 1, \lambda, \mu) \leqslant f(x)$. Ainsi, en notant que $L(x^*, 1, \lambda, \mu) = f(x^*)$ (puisque $h(x^*) = 0$ et $\mu_j g_j(x^*) = 0$), l'inégalité (4.3) donne

$$f(x^*) \leqslant f(x)$$

ce qui montre que x^* est minimiseur du problème avec contraintes. □

Remarque 40 (Contraintes redondantes et multiplicateur anormal.)**.** Quand on résout un problème d'optimisation sous contraintes d'égalité et d'inégalité, prenons garde à éviter les contraintes redondantes, qui créent un multiplicateur anormal (i.e., tel que $\lambda^0 = 0$).

Cela signifie que, en notant $\mathcal{C} = \{x \in \mathbb{R}^n \mid h(x) = 0,\ g(x) \leqslant 0\}$, on essaie d'éliminer, autant que possible, toutes les contraintes qui sont redondantes : par exemple, si le problème comporte les contraintes $x_1 + x_2 = 1$ et $2x_1 + 2x_2 = 2$, on ne garde bien sûr que la première.

Si des contraintes sont redondantes, il existe toujours un multiplicateur de Lagrange anormal ! Par exemple, considérons le problème de minimiser $f : \mathbb{R}^n \to \mathbb{R}$ (de classe C^1, quelconque) sous les contraintes d'égalité $h_1(x) = b^\top x - 1 = 0$ et $h_2(x) = 2b^\top x - 2 = 0$ (où $b \in \mathbb{R}^n \setminus \{0\}$ est fixé), qui sont évidemment redondantes. On a $\nabla h_1 = b$ et $\nabla h_2 = 2b = 2\nabla h_1$, donc $0 \times \nabla f(x) + 2\nabla h_1 - \nabla h_2 = 0$ et donc l'égalité des multiplicateurs de Lagrange est vérifiée avec le multiplicateur anormal ($\lambda_0 = 0, \lambda_1 = 2, \lambda_2 = 1$).

Généralisation des conditions de qualification. Pour $x^* \in \mathbb{R}^n$ vérifiant $h(x^*) = 0$ et $g(x^*) \leqslant 0$, la condition de qualification donnée dans le théorème 27 est l'hypothèse suivante :

> **(LI)** Les gradients $\nabla h_i(x^*)$, $i \in \{1, \dots, p\}$, $\nabla g_j(x^*)$, $j \in I(x^*)$ (indices des contraintes actives) sont tous linéairement indépendants (famille libre).

Ici, le sigle **(LI)** signifie "linéairement indépendant".

Dans la démonstration de KKT vue ci-dessus, on voit qu'on aurait pu affaiblir cette hypothèse. Pour démontrer par l'absurde que $\lambda_0 \neq 0$, il suffit de faire l'hypothèse plus faible suivante :

> **(PI)** Les gradients $\nabla h_i(x^*)$, $i \in \{1, \dots, p\}$, $\nabla g_j(x^*)$, $j \in I(x^*)$ (indices des contraintes actives) sont $\mathbb{R}^p \times \mathbb{R}_+^{\mathrm{card}(I(x^*))}$-indépendants, au sens suivant :
>
> $$\forall \lambda = (\lambda_1, \dots, \lambda_p) \in \mathbb{R}^p \qquad \forall \mu = (\mu_j)_{j \in I(x^*)} \in \mathbb{R}_+^{\mathrm{card}(I(x^*))}$$
> $$\sum_{i=1}^p \lambda_i \nabla h_i(x^*) + \sum_{j \in I(x^*)} \mu_j \nabla g_j(x^*) = 0 \quad \Longrightarrow \quad \lambda = 0,\ \mu = 0.$$

Ici, le sigle **(PI)** signifie, avec un léger abus, "positivement indépendant".

La condition **(PI)** est suffisante, dans le théorème KKT, pour assurer l'existence d'un multiplicateur normal ($\lambda_0 = 1$). Elle est plus générale que la condition **(LI)**, i.e., **(LI)** $\Rightarrow$ **(PI)**. Par exemple

si $p = 0$ (i.e., si on n'a que des contraintes d'inégalité), on peut avoir une infinité de vecteurs qui sont "positivement indépendants" au sens ci-dessus : il suffit de prendre des vecteurs qui pointent tous dans le quadrant positif (c'est-à-dire, dont toutes les coordonnées sont positives) ; et bien sûr, dès que ces vecteurs sont trop nombreux ils ne peuvent pas être linéairement indépendants.

Il existe de nombreuses autres conditions dans la littérature, assurant l'existence d'un multiplicateur normal. Ci-dessous, on en donne deux, qui sont très connues (toutefois, moins générales que (**PI**)) : la condition de Mangasarian-Fromovitz, et la condition de Slater.

Condition de Mangasarian-Fromovitz. Cette condition est la suivante :

(**MF**) Les gradients $\nabla h_i(x^*)$, $i \in \{1, \dots, p\}$, sont linéairement indépendants, et il existe $d \in \mathbb{R}^n \setminus \{0\}$ tel que

$$dh_i(x^*).d = 0 \quad \forall i \in \{1, \dots, p\} \qquad \text{et} \qquad dg_j(x^*).d < 0 \qquad \forall j \in I(x^*).$$

Cette dernière condition s'écrit de manière équivalente :

$$\langle \nabla h_i(x^*), d\rangle = 0 \quad \forall i \in \{1, \dots, p\} \qquad \text{et} \qquad \langle \nabla g_j(x^*), d\rangle < 0 \qquad \forall j \in I(x^*)$$

ou bien, en considérant la fonction H définie par (4.2) (voir la première preuve du théorème 27) :

$$dH(x^*).d \ \in \ \{0\} \times (\mathbb{R}^*_-)^{\mathrm{card}(I(x^*))}.$$

Lemme 3. *On a* (**LI**) $\Rightarrow$ (**MF**) $\Rightarrow$ (**PI**).

Démonstration. Montrons que (**LI**) $\Rightarrow$ (**MF**). L'hypothèse (**LI**) implique que $dH(x^*) : \mathbb{R}^n \to \mathbb{R}^p \times \mathbb{R}^{\mathrm{card}(I(x^*))}$ est surjective (et $p + \mathrm{card}(I(x^*)) \leqslant n$), donc un élément quelconque de $\{0\} \times (\mathbb{R}^*_-)^{\mathrm{card}(I(x^*))}$ admet au moins un antécédent $d \in \mathbb{R}^n$, ce qui donne (**MF**).

Montrons que (**MF**) $\Rightarrow$ (**PI**). Soient $\lambda = (\lambda_1, \dots, \lambda_p) \in \mathbb{R}^p$ et $\mu = (\mu_j)_{j \in I(x^*)} \in \mathbb{R}_+^{\mathrm{card}(I(x^*))}$ tels que $\sum_{i=1}^p \lambda_i \nabla h_i(x^*) + \sum_{j \in I(x^*)} \mu_j \nabla g_j(x^*) = 0$. En faisant le produit scalaire de cette égalité avec le vecteur d donné par (**MF**), on obtient $\sum_{j \in I(x^*)} \mu_j \langle \nabla g_j(x^*), d\rangle = 0$ et, comme $\mu_j \geqslant 0$ et $\langle \nabla g_j(x^*), d\rangle < 0$, on en déduit que $\mu_j = 0$ pour tout $j \in I(x^*)$. Puis on déduit que $\lambda_i = 0$ pour tout $i \in \{1, \dots, p\}$ puisque les vecteurs $\nabla h_i(x^*)$ sont linéairement indépendants. □

Condition de Slater. Cette condition, qui a l'avantage d'être facile à vérifier, concerne plus spécifiquement les problèmes d'optimisation avec contraintes d'égalité affines et contraintes d'inégalité convexes :

(**S**) Les fonctions h_i, $i \in \{1, \dots, p\}$, sont affines et linéairement indépendantes, les fonctions g_j, $j \in \{1, \dots, q\}$, sont convexes, et il existe $x \in \mathbb{R}^n$ tel que $h(x) = 0$ et $g(x) < 0$.

Lemme 4. *On a* (**S**) $\Rightarrow$ (**MF**) $\Rightarrow$ (**PI**).

Démonstration. D'après (**S**), il existe $x \in \mathbb{R}^n$ tel que $g_j(x) < 0$. Comme g_j est convexe, son graphe est au-dessus de ses tangentes, donc $g_j(x^*) + \langle \nabla g_j(x^*), x - x^*\rangle \leqslant g_j(x) < 0$, et comme $g_j(x^*) = 0$ pour $j \in I(x^*)$, en posant $d = x - x^*$, on a donc $\langle \nabla g_j(x^*), d\rangle < 0$. Par ailleurs, comme les h_i sont affines, on a $\langle \nabla h_i(x^*), d\rangle = h_i(x) - h_i(x^*) = 0$. On a donc obtenu la condition (**MF**). □

Remarque 41. On peut légèrement généraliser la condition de Slater de la manière suivante, en distinguant, parmi les fonctions g_j, celles qui sont affines de celles qui sont convexes non affines. La condition est alors : les fonctions h_i, $i \in \{1, \dots, p\}$, sont affines et les fonctions g_j, $j \in \{1, \dots, q\}$, sont convexes ; les gradients des fonctions h_i et g_j qui sont affines sont linéairement indépendants ; il existe $x \in \mathbb{R}^n$ tel que $h(x) = 0$ et $g_j(x) < 0$ si g_j n'est pas affine, pour $j = 1, \dots, q$.

Exemple 9. Considérons le problème

$$\min\left\{\frac{1}{2}\|x-y_0\|^2 \;\mid\; x_1\geqslant 0,\ x_2\geqslant 0,\ x_1+x_2\leqslant 1\right\}$$

avec $y_0=(1,1/2)$. Bien sûr, la solution est évidente (faire un dessin) : la solution x^* est le projeté orthogonal de y_0 sur le triangle hachuré. On retrouve toutefois par le calcul, après résolution des conditions KKT :

$$x_1^*=\frac{3}{4},\quad x_2^*=\frac{1}{4},\qquad \lambda_0=1,\quad \mu_1=\mu_2=0,\quad \mu_3=\frac{1}{4}.$$

4.2.2.3 Application : fonctionnelle quadratique avec contraintes d'égalité affines

Cas général. Soient $A\in\mathcal{M}_n(\mathbb{R})$ est une matrice symétrique définie positive, $b\in\mathbb{R}^n$ et $c\in\mathbb{R}$ (on peut prendre $c=0$: cela ne change rien dans ce qui suit). Soient $k\leqslant n$, $C\in\mathcal{M}_{k,n}(\mathbb{R})$ (la matrice a moins de lignes que de colonnes) et $d\in\mathbb{R}^k$. On s'intéresse au problème

$$\boxed{\min_{Cx=d}\left(\frac{1}{2}x^\top Ax-b^\top x+c\right)} \tag{4.4}$$

i.e., le problème de minimiser la fonction $f(x)=\frac{1}{2}x^\top Ax-b^\top x+c$ sous les k contraintes d'égalité $Cx=d$: en notant $L_1,\dots,L_k\in\mathcal{M}_{1,n}(\mathbb{R})$ les k lignes de la matrice C, ces k contraintes affines sont

$$h_i(x)=L_ix-d_i=0,\qquad i=1,\dots,k.$$

On suppose C surjective, i.e., $\mathrm{rg}(C)=k$, ce qui est logique car on souhaite que l'ensemble des x tels que $Cx=d$ soit non trivial ! Cela veut exactement dire que les k contraintes d'égalité sont indépendantes, donc la condition de qualification est vérifiée : les vecteurs $\nabla h_i(x)=L_i^\top$, $i=1,\dots,k$, sont linéairement indépendants.

Le problème (4.4) a une unique solution x car f est strictement convexe et infinie à l'infini, et l'ensemble des contraintes est convexe. D'après la proposition 4, l'unique solution x du problème est caractérisée par la condition de multiplicateurs de Lagrange (qui est alors une condition nécessaire et suffisante) : il existe $\lambda=(\lambda_1,\dots,\lambda_k)\in\mathbb{R}^k$ tel que

$$\nabla f(x)+\sum_{i=1}^k\lambda_i\nabla h_i(x)=0$$

c'est-à-dire

$$\begin{aligned}&Ax+C^\top\lambda=b\\&Cx=d\end{aligned}$$

ce qui s'écrit sous forme matricielle

$$\boxed{\underbrace{\begin{pmatrix}A & C^\top\\ C & 0\end{pmatrix}}_{M}\begin{pmatrix}x\\ \lambda\end{pmatrix}=\begin{pmatrix}b\\ d\end{pmatrix}} \tag{4.5}$$

Notons bien que (4.5) est une condition nécessaire et suffisante pour que x soit solution du problème (4.4), avec λ comme multiplicateur de Lagrange associé (et $\lambda_0=1$ car le problème est qualifié).

Notons aussi que la matrice $M \in \mathcal{M}_{n+k}(\mathbb{R})$ apparaissant dans (4.5) est (symétrique) inversible. En effet, soit $(y, \mu) \in \mathbb{R}^n \times \mathbb{R}^k$ tel que $M \begin{pmatrix} y \\ \mu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, alors on a $Ay + C^\top \mu = 0$ et $Cy = 0$, donc

$$0 = \begin{pmatrix} y^\top & \mu^\top \end{pmatrix} M \begin{pmatrix} y \\ \mu \end{pmatrix} = y^\top Ay + 2\mu^\top Cy = y^\top Ay$$

et donc $y = 0$ puisque A est symétrique définie positive. On en déduit que $C^\top \mu = 0$, et comme $C^\top$ est injective (car C est surjective), on a aussi $\mu = 0$. Donc M est inversible.

En particulier, le multiplicateur de Lagrange normal λ associé à x est unique.

Cas particulier : moindres carrés contraints. Dans une séance précédente, nous avons vu comment résoudre le problème des moindres carrés

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|^2$$

avec $A \in \mathcal{M}_{m,n}(\mathbb{R})$, $b \in \mathbb{R}^n$, qui est un problème dont l'importance est cruciale dans un grand nombre d'applications. On a vu que ce problème admet une unique solution lorsque A est injective (ce qui impose $m \geqslant n$: la matrice A a plus de lignes que de colonnes), donnée par la pseudo-inverse

$$x = A^\# b = (A^\top A)^{-1} A^\top b$$

On s'intéresse ici à ajouter des contraintes d'égalité affines dans cet important problème :

$$\min_{\substack{x \in \mathbb{R}^n \\ Cx = d}} \|Ax - b\|^2 \tag{4.6}$$

où $C \in \mathcal{M}_{k,n}(\mathbb{R})$ est surjective et $d \in \mathbb{R}^k$, i.e., on ajoute k contraintes d'égalité affines indépendantes. Bien entendu, on peut mettre un $\frac{1}{2}$ devant la norme dans (4.6) : cela ne change rien au problème de minimisation, et alors la fonctionnelle f à minimiser est du type précédent : $f(x) = \frac{1}{2} x^\top A^\top Ax - b^\top Ax + \frac{1}{2}\|b\|^2$ (la matrice $A^\top A$ est symétrique définie positive).

Avec ce qu'on a vu ci-dessus, il existe une unique solution optimale $x \in \mathbb{R}^n$, qui admet un unique multiplicateur de Lagrange normal $\lambda \in \mathbb{R}^k$, et le couple (x, λ) est l'unique solution du système

$$\boxed{\begin{pmatrix} A^\top A & C^\top \\ C & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} A^\top b \\ d \end{pmatrix}}$$

(dont la matrice, de même, est inversible).

Application : distance d'un point à un sous-espace affine. Dans le problème (4.6), prenons $m = n$ et $A = I_n$. On a alors le problème

$$\min_{\substack{x \in \mathbb{R}^n \\ Cx = d}} \|x - b\|^2$$

qui est exactement le problème de trouver, pour un point $b \in \mathbb{R}^n$ donné, quel est le point x du sous-espace affine $F = \{x \in \mathbb{R}^n \mid Cx = d\}$ qui minimise la distance à b. Bien sûr, géométriquement, ce point x est exactement le projeté orthogonal du point b sur F : $x = P_F(b)$.

D'après ce qu'on a vu ci-dessus, cet unique point est caractérisé par les équations

$$x + C^\top \lambda = b, \qquad Cx = d$$

En multipliant la première équation par C, on a $CC^\top\lambda = Cb - Cx = Cb - d$. Comme C est surjective, la matrice $CC^\top$ est inversible, donc $\lambda = (CC^\top)^{-1}(Cb-d)$, et donc finalement, comme $x = b - C^\top\lambda$, on obtient

$$\boxed{x = P_F(b) = \left(I_n - C^\top(CC^\top)^{-1}C\right)b + C^\top(CC^\top)^{-1}d}$$

On peut noter que, comme C est surjective, sa pseudo-inverse est $C^\# = C^\top(CC^\top)^{-1}$.

Notons que la projection P_F ci-dessus est une application affine. Si $d = 0$, la formule ci-dessus donne $x = \left(I_n - C^\top(CC^\top)^{-1}C\right)b$ qui est exactement la projection orthogonale de b sur le sous-espace $F = \ker(C)$. On a donc obtenu, en prime, le résultat suivant : le projecteur orthogonal sur le sous-espace vectoriel $\ker(C)$ est

$$\boxed{P_{\ker(C)} = I_n - C^\top(CC^\top)^{-1}C}$$

et $P_F = P_{\ker(C)} + C^\top(CC^\top)^{-1}d$.

Retour sur la pseudo-inverse. Supposons vouloir résoudre l'équation $Ax = b$ où A est surjective (A a moins de lignes que de colonnes). Dans ce cas, l'équation admet plein de solutions, et cherchons alors la solution de norme minimale :

$$\min_{Ax=b} \|x\|^2.$$

d'après ce qu'on a dit précédemment, il existe une unique solution, caractérisée par les équations $x + A^\top\lambda = 0$ et $Ax = b$, ce qui conduit à

$$x = A^\top(AA^\top)^{-1}b$$

Notons que, comme A est surjective, sa pseudo-inverse est $A^\# = A^\top(AA^\top)^{-1}$.

Comme expliqué dans le chapitre sur les moindres carrés, toutefois, ce cas est moins intéressant. Le cas intéressant en pratique est de vouloir résoudre $Ax = b$ avec A injective (A ayant plus de lignes que de colonnes), et comme ce système est sans solution on résout le problème des moindres carrés pour trouver la meilleure “solution approchée" possible.

4.2.3 Conditions d'optimalité du second ordre

Cette section est un peu plus dure et peut être passée en première lecture. Les conditions d'optimalité de second ordre sont moins souvent utilisées en pratique, car plus difficiles d'utilisation. Sur le plan théorique toutefois elles ouvrent à l'analyse de sensibilité, importante en optimal design, et qui est utile pour comprendre mieux la méthode de Lagrange-Newton qui sera traitée plus loin.

4.2.3.1 Conditions générales

Comme dans le cas sans contrainte, on a d'une part une condition nécessaire d'optimalité du second ordre (en gros, une dérivée second est positive), et d'autre part une condition suffisante d'optimalité locale du second ordre (en gros, une dérivée seconde strictement positive).

Considérons le problème d'optimisation

$$\boxed{\min_{\substack{h(x)=0\\ g(x)\leqslant 0}} f(x)}$$

où $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^q$ et $h : \mathbb{R}^n \to \mathbb{R}^p$ sont C^2. On a vu précédemment qu'un tel problème peut toujours se ramener au problème (4.1) avec contraintes d'égalité seulement, en introduisant les indices actifs et l'application H définie par (4.2).

Notons aussi que, sous les conditions de qualification en x, l'ensemble $H = 0$ est une sous-variété localement autour de x, et son espace tangent est

$$\ker dH(x) = \{d \in \mathbb{R}^n \ \mid \ dH(x).d = 0\} = \bigcap_{i=1}^{p} \ker dh_i(x) \bigcap \bigcap_{j \in I(x)} \ker dg_j(x).$$

Théorème 28. — ***Condition nécessaire.*** *Si x^* est un minimiseur, si le problème est qualifié et si $(\lambda, \mu) \in \times\mathbb{R}^p \times (\mathbb{R}^+)^q$ est un multiplicateur de Lagrange normal associé (vérifiant le théorème KKT, avec $\lambda_0 = 1$), alors*

$$\frac{\partial^2 L}{\partial x^2}(x^*, \lambda, \mu).(d, d) \geqslant 0 \qquad \forall d \in \ker dH(x^*).$$

— ***Condition suffisante.*** *Si le triplet (x^*, λ, μ), supposé normal (i.e., $\lambda_0 = 1$), vérifie toutes les conclusions du théorème KKT et si*

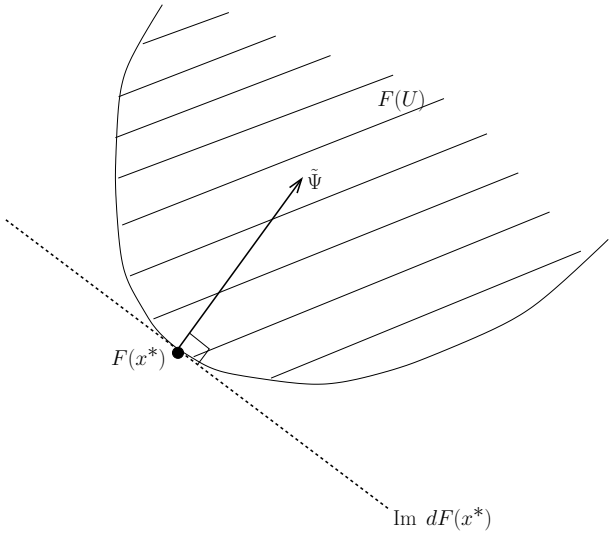
$$\frac{\partial^2 L}{\partial x^2}(x^*, \lambda, \mu).(d, d) > 0 \qquad \forall d \in \ker dH(x^*) \setminus \{0\}$$

alors x^* est un minimiseur local.

Démonstration. On peut admettre cette preuve dans un premier temps, car elle est plus difficile. La preuve fait en effet appel à un concept de géométrie différentielle un peu élaboré : le concept de *dérivée seconde intrinsèque*. Pour l'expliquer, considérons l'application augmentée $F(x) = (f(x), H(x))$. On a vu que les équations KKT sont équivalentes à la relation $\tilde{\Psi} \perp \text{Im } dF(x^*)$ (i.e., $dF(x^*)^\top \Psi = 0$).

On s'intéresse à décrire l'allure de l'ensemble image $F(U)$ où U est un voisinage ouvert du point x^*. Bien sûr, si $dF(x^*)$ est surjective, $F(U)$ est un voisinage ouvert de x^* : c'est le théorème des fonctions implicites.

Mais lorsque $F(x^*)$ est au bord de $F(\mathbb{R}^n)$ ce n'est pas le cas, et alors on va voir que $F(U)$ ressemble à l'intérieur d'une parabole, cf dessin.



En effet, Im $dF(x^*)$ décrit l'ensemble des points images par F, "à l'ordre 1" : c'est, de manière approchée, l'ensemble des points $F(x^* + d) = F(x^*) + dF(x^*).d + o(\|d\|)$ où on néglige tous les termes d'ordre > 1. Mais comme $dF(x^*)$ n'est pas surjective, cet ensemble ne donne pas une bonne approximation de l'ensemble image $F(U)$! (cf théorème des fonctions implicites) On a alors besoin de faire une approximation d'ordre 2. Pour cela, on regarde la dérivée seconde intrinsèque, qui est la forme quadratique :

$$Q = \tilde{\Psi}.\, d^2F(x^*)_{|\ker dF(x^*)}$$

C'est la Hessienne de F en x^*, qu'on restreint au noyau de $dF(x^*)$ (puisqu'on connaît déjà ce qui se passe à l'ordre 1) et qu'on co-restreint, autrement dit qu'on regarde, le long de $\tilde{\Psi}$ (car, on le voit sur le dessin, c'est la direction manquante). Du point de vue géométrie différentielle, cet objet est intrinsèque : la forme quadratique Q est bien définie et ne dépend pas des choix de coordonnées.

Maintenant, on a les faits suivants :

— Si $F(x^*)$ est au bord de $F(\mathbb{R}^n)$ alors $Q \geqslant 0$: la forme quadratique Q est positive.

— Si $Q > 0$, i.e., si la forme quadratique Q est définie positive, alors $F(x^*)$ est "strictement au bord" de $F(U)$, au sens où $F(U)$ ressemble à l'intérieur d'une parabole dont le sommet est $F(x^*)$ (cf dessin).

C'est une analyse d'ordre deux, de type "théorie de Morse" en géométrie différentielle, mais qui ne fait que généraliser ce qu'on connaît bien sur les fonctions de $\mathbb{R}$ dans $\mathbb{R}$: si x minimise f alors $f'(x) = 0$ et $f''(x) \geqslant 0$, et réciproquement, si $f'(x) = 0$ et $f''(x) > 0$ alors x minimise f localement.

Il reste alors à relier Q à la Hessienne de L restreinte à $\ker dH(x^*)$. Tout d'abord, comme $F = (f, H)$ et que les éléments non nuls de $\tilde{\Psi}$ sont $\tilde{\lambda} = (1, \lambda_1, \ldots, \lambda_p, (\mu_j)_{j\in I(x^*)})$, on a $\tilde{\Psi}.F = L$. Il reste alors seulement à montrer que $\ker dF(x^*) = \ker dH(x^*)$. Comme $F = (f, H)$, on a $\ker dF(x^*) = \ker df(x^*) \cap \ker dH(x^*)$. Par ailleurs, x^* vérifie la condition de multiplicateur de Lagrange, qui s'écrit $df(x^*) + \tilde{\lambda}.dH(x^*) = 0$, ce qui implique que $\ker dH(x^*) \subset \ker df(x^*)$, et donc $\ker dF(x^*) = \ker dH(x^*)$. □

4.2.3.2 Application à l'analyse de sensibilité

Dans cette section, on considère des problèmes d'optimisation dépendant d'un ou plusieurs paramètres, et on veut savoir comment la solution optimale dépend de ces paramètres. C'est une question très importante dans de nombreuses applications, par exemple lorsqu'on fait de la conception optimale (optimal design).

Le cadre est le suivant. On note $s \in \mathcal{S}$ le paramètre, où $\mathcal{S}$ est un espace de Banach. Pour tout $s \in \mathcal{S}$, on considère le problème d'optimisation sous contrainte d'égalité (on s'y ramène toujours comme on l'a vu)

$$\min \{f(x, s) \ \mid \ x \in \mathbb{R}^n \text{ t.q. } h(x, s) = 0\} \tag{4.7}$$

où $f : \mathbb{R}^n \times \mathcal{S} \to \mathbb{R}$ et $h : \mathbb{R}^n \times \mathcal{S} \to \mathbb{R}^p$ sont C^2. Soit $U \subset \mathbb{R}^p$ un voisinage ouvert de 0. On suppose ce problème "bien posé" au sens où, pour tout $s \in \mathcal{S}$, il existe un unique minimiseur $x(s)$. On suppose aussi que, pour tout $s \in \mathcal{S}$, le problème est qualifié, ce qui signifie que l'application linéaire $\frac{\partial h}{\partial x}(x(s), s) : \mathbb{R}^n \to \mathbb{R}^p$ est surjective.

On définit la "fonction valeur"

$$S(s) = \min_{h(x,s)=0} f(x, s) = f(x(s), s).$$

On s'intéresse à savoir comment la fonction valeur $S(s)$ et le minimiseur $x(s)$ dépendent du paramètre $s \in \mathcal{S}$: c'est ce qui s'appelle l'analyse de sensibilité ("sensitivity analysis" en anglais).

L'analyse de sensibilité se fait grâce au théorème des fonctions implicites appliqué au système d'optimalité obtenu avec la règle des multiplicateurs de Lagrange. Pour tout $s \in \mathcal{S}$, comme $x(s)$

est minimiseur, et comme le problème est supposé qualifié, il existe un multiplicateur de Lagrange $\lambda(s) \in \mathbb{R}^p$ (vecteur colonne ; donc $\lambda(s)^\top$ est un vecteur ligne) tel que

$$\frac{\partial f}{\partial x}(x(s), s) + \lambda(s)^\top \frac{\partial h}{\partial x}(x(s), s) = 0$$

En écriture Lagrangienne, on définit ici

$$L(x, \lambda, s) = f(x, s) + \langle \lambda, h(x, s) \rangle = f(x, s) + \lambda^\top h(x, s) = f(x, s) + \sum_{i=1}^{p} \lambda_i h_i(x, s)$$

et la condition ci-dessus s'écrit

$$\nabla_x L(x(s), \lambda(s), s) = 0$$

Ainsi, notre système d'optimalité est

$$\boxed{F(x(s), \lambda(s), s) = \begin{pmatrix} \nabla_x L(x(s), \lambda(s), s) \\ h(x(s), s) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}}$$

où l'application $F : \mathbb{R}^n \times \mathbb{R}^p \times \mathcal{S} \to \mathbb{R}^n \times \mathbb{R}^p$ est définie par

$$F(x, \lambda, s) = \begin{pmatrix} \nabla_x L(x, \lambda, s) \\ h(x, s) \end{pmatrix}$$

On voit maintenant qu'on est dans le cadre du théorème des fonctions implicites : on veut en effet résoudre, par rapport à (x, λ), le système de $n+p$ équations $F(x, \lambda, s) = 0$ à $n+p$ inconnues (x, λ), en fonction du paramètre s.

Pour appliquer le théorème des fonctions implicites, il faut vérifier que, pour tout $s \in \mathcal{S}$, la jacobienne $\frac{\partial F}{\partial (x,\lambda)}(x, \lambda, s)$ est inversible. Calculons cette jacobienne :

$$\frac{\partial F}{\partial (x, \lambda)}(x, \lambda, s) = \begin{pmatrix} \frac{\partial^2 L}{\partial x^2}(x, \lambda, s) & \frac{\partial^2 L}{\partial x \partial \lambda}(x, \lambda, s) \\ \frac{\partial h}{\partial x}(x, s) & \frac{\partial h}{\partial \lambda}(x, s) \end{pmatrix} = \begin{pmatrix} \frac{\partial^2 L}{\partial x^2}(x, \lambda, s) & \frac{\partial h}{\partial x}(x, s)^\top \\ \frac{\partial h}{\partial x}(x, s) & 0 \end{pmatrix} = \begin{pmatrix} A & C^\top \\ C & 0 \end{pmatrix}$$

avec $A = \frac{\partial^2 L}{\partial x^2}(x, \lambda, s)$ et $C = \frac{\partial h}{\partial x}(x, s)$.

Montrons que, sous la condition suffisante du second ordre du théorème 28, qui s'écrit ici

$$A_1 = \frac{\partial^2 L}{\partial x^2}(x, \lambda, s)_{|\ker \frac{\partial h}{\partial x}(x,s)} \quad \text{définie positive}$$

(plus généralement, il suffit de supposer que cette forme quadratique soit non dégénérée) la jacobienne $\frac{\partial F}{\partial (x,\lambda)}(x, \lambda, s)$ est inversible. La preuve est similaire à celle qui a été faite en section 4.2.2.3. Soit $(y, \mu) \in \mathbb{R}^n \times \mathbb{R}^p$ tel que $\frac{\partial F}{\partial (x,\lambda)}(x, \lambda, s).(y, \mu) = 0$, i.e.,

$$Ay + C^\top \mu = 0, \qquad Cy = 0.$$

Comme $y \in \ker C$, la première équation s'écrit donc $A_1 y + C^\top \mu = 0$. On la multiplie par $y^\top$, et comme $y^\top C^\top = 0$ on obtient $y^\top A_1 y = 0$, d'où $y = 0$ puisque A_1 est définie positive (non dégénérée suffirait). Donc $C^\top \mu = 0$, et comme C est supposée surjective, $C^\top$ est injective et donc $\mu = 0$.

On déduit du théorème des fonctions implicites que l'équation $F(x, \lambda, s) = 0$ se résout par rapport à (x, λ) et donne $s \mapsto (x(s), \lambda(s))$ de classe C^1.

On a donc obtenu le résultat suivant.

Proposition 5. *Soit $s^* \in \mathcal{S}$ un paramètre fixé. On suppose que, pour $s = s^*$, le problème (4.7) admet un unique minimiseur local $x^* = x(s^*)$, ayant un multiplicateur de Lagrange (normal) $\lambda(s^*)$ et que*

- *l'application linéaire $\frac{\partial h}{\partial x}(x(s^*), s^*) : \mathbb{R}^n \to \mathbb{R}^p$ est surjective ;*
- *la forme quadratique $\frac{\partial^2 L}{\partial x^2}(x^*(s^*), \lambda(s^*), s^*)_{|\ker \frac{\partial h}{\partial x}(x(s^*), s^*)}$ est définie positive.*

Alors il existe un voisinage ouvert V de s^* dans $\mathcal{S}$ tel que, pour tout $s \in V$, le problème (4.7) admet un unique minimiseur local $x(s)$, ayant un multiplicateur de Lagrange (normal) $\lambda(s)$, tous deux dépendant de s de manière C^1. La fonction valeur $S(s)$ dépend aussi de s de manière C^1.

En dérivant la relation $F(x(s), \lambda(s), s) = 0$ par rapport à $s \in \mathcal{S}$, on peut de plus noter que les différentielles $x'(s)$ et $\lambda'(s)$ par rapport à s vérifient le système

$$\begin{pmatrix} \frac{\partial^2 L}{\partial x^2}(x(s), \lambda(s), s) & \frac{\partial h}{\partial x}(x(s), s)^\top \\ \frac{\partial h}{\partial x}(x(s), s) & 0 \end{pmatrix} \begin{pmatrix} x'(s) \\ \lambda'(s) \end{pmatrix} = - \begin{pmatrix} \frac{\partial^2 L}{\partial s \partial x}(x(s), \lambda(s), s) \\ \frac{\partial h}{\partial s}(x(s), s) \end{pmatrix}$$

ce qui est sous-jacent à la méthode numérique de résolution de Lagrange-Newton (voir plus loin).

Par ailleurs, en dérivant par rapport à s les relations $S(s) = f(x(s), s)$ et $h(x(s), s) = 0$, on obtient (on note toujours $'$ la différentielle par rapport à $s \in \mathcal{S}$)

$$S'(s) = \frac{\partial f}{\partial x}(x(s), s).x'(s) + \frac{\partial f}{\partial s}(x(s), s) \qquad \text{et} \qquad \frac{\partial h}{\partial x}(x(s), s).x'(s) + \frac{\partial h}{\partial s}(x(s), s) = 0.$$

Or, la relation de multiplicateur de Lagrange s'écrit $\frac{\partial f}{\partial x}(x(s), s) + \lambda(s)^\top \frac{\partial h}{\partial x}(x(s), s) = 0$, et en l'appliquant à $x'(s)$ et en utilisant les relations ci-dessus, on obtient finalement

$$S'(s) = \frac{\partial f}{\partial s}(x(s), s) + \lambda(s)^\top \frac{\partial h}{\partial s}(x(s), s).$$

Remarque 42 (Cas particulier)**.** Un cas particulier intéressant est lorsque $f(x, s) = f(x)$ et $h(x, s) = h(x) - s$ avec $\mathcal{S} = \mathbb{R}^p$, autrement dit on considère le problème d'optimisation dont la fonction valeur est

$$S(s) = \min_{h(x)=s} f(x).$$

On peut le voir comme le problème d'optimisation $\min_{h=0} f$ où l'on perturbe les contraintes d'égalité avec un petit paramètre s. Dans les conditions de la proposition 5, on obtient alors $S'(s) = -\lambda(s)^\top$, c'est-à-dire

$$\lambda_i(s) = -\frac{\partial S}{\partial s_i}(s) \qquad \forall i \in \{1, \dots, p\}.$$

Ce résultat est intéressant car il donne une interprétation des multiplicateurs de Lagrange : pour $i = 1, \dots, p$, le multiplicateur $\lambda_i = \lambda_i(0)$ associé au problème d'optimisation $\min_{h=0} f$ est égal à $\lambda_i = -\frac{\partial S}{\partial s_i}(0)$, autrement dit il est lié à la façon dont la fonction valeur S est modifiée si on perturbe la contrainte d'égalité $h_i(x) = 0$ par $h_i(x) = s_i$. En ce sens, le multiplicateur de Lagrange λ_i est une mesure de la sensibilité de la fonction valeur par rapport à la contrainte $h_i(x) = 0$.

Petit aparté : théorème de Danskin. Mentionnons ici un théorème parfois bien utile, qui relève de l'analyse de sensibilité, qui permet de dériver une "fonction valeur". Le contexte est le suivant. Soient X un espace de Banach, K un espace topologique compact et $f : X \times K \to \mathbb{R}$ une fonction continue, dérivable dans toute direction par rapport à sa première variable. On définit la *fonction valeur* $S : X \to \mathbb{R}$ par

$$S(x) = \min\{f(x, y) \ \mid \ y \in K\}.$$

Pour tout $x \in X$, on note $\hat{K}(x) = \{y \in K \ \mid \ f(x, y) = S(x)\}$ l'ensemble des minimiseurs de $f(x, \cdot)$.

Théorème 29 (Danskin). *La fonction S est différentiable dans toute direction en tout point $x \in X$, et*

$$S'(x;h) = \min\left\{ \frac{\partial f}{\partial x}(x,y).h \;\mid\; y \in \hat{K}(x) \right\} \qquad \forall h \in X.$$

Ici, $\frac{\partial f}{\partial x}(x,y).h$ est la dérivée de f par rapport à x dans la direction h. Notons que, si $\hat{K}(x) = \{y_x\}$ est un singleton (i.e., si $f(x,\cdot)$ a un unique minimiseur $y_x \in K$), alors $S'(x;h) = \frac{\partial f}{\partial x}(x,y_x).h$.

4.3 Algorithmes d'optimisation avec contraintes

Dans cette section nous allons étudier différentes méthodes algorithmiques pour les problèmes d'optimisation avec contraintes. On distingue les méthodes primales (qui sont, en gros, de type gradient) des méthodes duales (qui sont basées sur l'interprétation Lagrangienne).

4.3.1 Méthodes primales

4.3.1.1 Méthodes de projection

On considère le problème d'optimisation avec contraintes

$$\boxed{\min_{x\in\mathcal{C}} f(x)}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est C^1 et $\mathcal{C} \subset \mathbb{R}^n$ est convexe.

Méthode de gradient projeté. Lorsque $\mathcal{C} = \mathbb{R}^n$, la méthode de gradient s'écrit

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

Le défaut est que, même si $x_0 \in \mathcal{C}$, les itérés suivants x_k ne restent pas forcément dans $\mathcal{C}$. L'idée de la méthode de gradient projeté est toute simple : à chaque itération, on projette le nouveau point dans le convexe $\mathcal{C}$.

Soit $P_{\mathcal{C}}$ la projection orthogonale sur $\mathcal{C}$ (voir le théorème du convexe dans le chapitre sur la convexité : on rappelle que cette projection est bien définie et que, de plus, c'est une application 1-Lipschitzienne).

La méthode du gradient projeté est alors

$$\boxed{x_{k+1} = P_{\mathcal{C}}(x_k - \rho_k \nabla f(x_k))}$$

et on obtient, comme on l'a vu précédemment, diverses variantes en prenant un pas fixe ou variable (pas optimal par exemple).

Comme pour la méthode de gradient, on peut démontrer un théorème de convergence :

Théorème 30. *On suppose que f est α-convexe (avec $\alpha > 0$) et que ∇f est M-Lipschitzienne (avec $M > 0$).*[2] *Alors le problème d'optimisation a une unique solution x^*. Soient $0 < \beta_1 < \beta_2 < \frac{2\alpha}{M^2}$. Si à toute itération on choisit le pas $\rho_k \in [\beta_1, \beta_2]$ alors, quel que soit le point initial x_0, la méthode de gradient projeté converge vers x^*.*

Remarque 43. Comme pour la méthode de gradient, si les hypothèses sur f ne sont vraies que localement autour de x^*, alors la convergence est locale.

2. On peut noter que, forcément, $\alpha \leqslant M$.

Démonstration. Tout d'abord, comme f est α-convexe, elle admet sur le convexe $\mathcal{C}$ un unique minimiseur $x^* \in \mathcal{C}$. Par la condition nécessaire d'optimalité de f sur un convexe, on a

$$\langle \nabla f(x^*), x - x^* \rangle \geqslant 0 \qquad \forall x \in \mathcal{C}$$

donc

$$\langle x^* - t\nabla f(x^*) - x^*, x - x^* \rangle \leqslant 0 \qquad \forall x \in \mathcal{C} \qquad \forall t > 0.$$

Par le théorème du convexe (caractérisation du projeté), on en déduit que

$$x^* = P_{\mathcal{C}}(x^* - t\nabla f(x^*)) \qquad \forall t > 0. \tag{4.8}$$

On a alors

$$x_{k+1} - x^* = P_{\mathcal{C}}(x_k - \rho_k \nabla f(x_k)) - P_{\mathcal{C}}(x^* - \rho_k \nabla f(x^*))$$

et comme $P_{\mathcal{C}}$ est 1-Lipschitzienne, on en déduit que

$$\begin{aligned}\|x_{k+1} - x^*\|^2 &\leqslant \|x_k - x^* - \rho_k(\nabla f(x_k) - \nabla f(x^*))\|^2 \\ &\leqslant \|x_k - x^*\|^2 - 2\rho_k \langle x_k - x^*, \nabla f(x_k) - \nabla f(x^*)\rangle + \rho_k^2 \|\nabla f(x_k) - \nabla f(x^*)\|^2\end{aligned}$$

Comme ∇f est M-Lipschitzienne, on a $\|\nabla f(x_k) - \nabla f(x^*)\| \leqslant M\|x_k - x^*\|$, et comme f est α-convexe, on a $\langle x_k - x^*, \nabla f(x_k) - \nabla f(x^*)\rangle \geqslant \alpha \|x_k - x^*\|^2$. Par conséquent,

$$\|x_{k+1} - x^*\|^2 \leqslant (1 - 2\alpha\rho_k + M^2\rho_k^2)\|x_k - x^*\|^2.$$

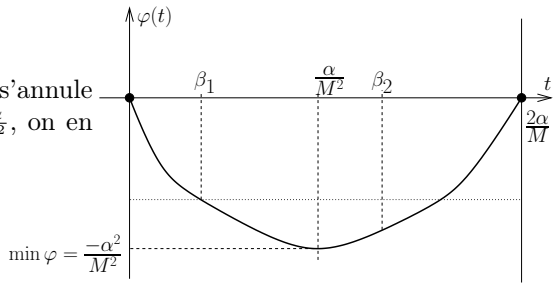
La fonction

$$\varphi(t) = M^2 t^2 - 2\alpha t = M^2 t \left(t - \frac{2\alpha}{M^2}\right)$$

est négative et convexe sur l'intervalle $[0, \frac{2\alpha}{M^2}]$, et s'annule aux deux bords. Comme $0 < \beta_1 \leqslant \rho_k \leqslant \beta_2 < \frac{2\alpha}{M^2}$, on en déduit que

$$\varphi(\rho_k) \leqslant \max(\varphi(\beta_1), \varphi(\beta_2)) < 0$$

et donc



$$1 - 2\alpha\rho_k + M^2\rho_k^2 \leqslant 1 + \max(\varphi(\beta_1), \varphi(\beta_2)) = \kappa < 1.$$

On peut noter que $\kappa \geqslant 0$ (en effet, $\kappa \geqslant 1 + \min\varphi = 1 - \frac{\alpha^2}{M^2} \geqslant 0$ car $\alpha \leqslant M$). Par conséquent,

$$\|x_{k+1} - x^*\| \leqslant \sqrt{\kappa}\|x_k - x^*\|$$

avec $0 \leqslant \kappa < 1$, donc la suite $(x_k)_{k\in\mathbf{N}}$ converge vers x^*. □

Remarque 44. Cet algorithme est très simple à mettre en oeuvre lorsqu'on connaît explicitement la projection $P_{\mathcal{C}}$. Les cas typiques sont les suivants :

- $\mathcal{C} = \{x \in \mathbb{R}^n \mid Cx = d\}$ avec C surjective : on a vu en section 4.2.2.3 que [3]

$$P_{\mathcal{C}}(x) = P_{\ker(C)}(x) + C^\top (CC^\top)^{-1} d \qquad \text{avec} \qquad P_{\ker(C)} = I_n - C^\top (CC^\top)^{-1} C$$

3. Voici une autre manière de calculer $P_{\ker(C)}$, le projecteur orthogonal sur $\ker(C)$ (avec C surjective) :
   - Par définition du projecteur orthogonal, pour tout x, on a $P_{\ker(C)}(x) \in \ker C$ et $x - P_{\ker(C)}(x) \perp \ker C$, i.e., $CP_{\ker(C)}(x) = 0$ et $\forall y,\ \langle x - P_{\ker(C)}(x), y\rangle = 0$.
   - On a $\ker C = (\operatorname{Im} C^\top)^\perp$. En effet, $x \in \ker C \Leftrightarrow Cx = 0 \Leftrightarrow \forall y,\ \langle y, Cx\rangle = 0 = \langle C^\top y, x\rangle = 0 \Leftrightarrow x \in (\operatorname{Im} C^\top)^\perp$.
   - Donc $x - P_{\ker(C)}(x) \in (\ker C)^\perp = \operatorname{Im} C^\top$, i.e., il existe y tel que $x - P_{\ker(C)}(x) = C^\top y$. Or, $CP_{\ker(C)}(x) = 0$ donc $Cx = CC^\top y$ d'où $y = (CC^\top)^{-1}Cx$ ($CC^\top$ est inversible car C est surjective), et donc $P_{\ker(C)}(x) = x - C^\top y = x - C^\top(CC^\top)^{-1}Cx$.

Comme la projection $P_{\mathcal{C}}$ est affine, on a $P_{\mathcal{C}}(x_k - \rho_k \nabla f(x_k)) = x_k - \rho_k P_{\ker(C)}(\nabla f(x_k))$ car, à l'itération k, on avait déjà $Cx_k = d$. Donc, pour cette contrainte affine, la méthode du gradient projeté s'écrit

$$\boxed{x_{k+1} = x_k - \rho_k P_{\ker(C)}(\nabla f(x_k)) = x_k - \rho_k \left(I_n - C^\top (CC^\top)^{-1} C\right) \nabla f(x_k)}$$

- $\mathcal{C} = \mathbb{R}^n_+$: on a alors

$$P_{\mathcal{C}}(x) = \max(x, 0) = (\max(x^1, 0), \ldots, \max(x^n, 0))$$

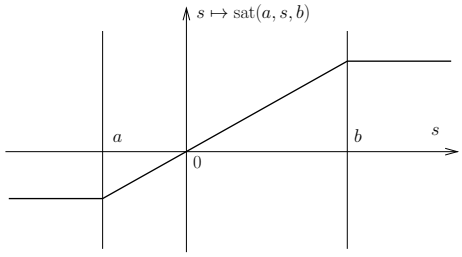
où $x = (x^1, \ldots, x^n)$ en coordonnées. Les contraintes de positivité sont très fréquentes en optimisation.

- $\mathcal{C} = \prod_{i=1}^{n} [a_i, b_i]$ (contraintes de bornes) : on a alors

$$P_{\mathcal{C}}(x) = \big(\mathrm{sat}(a_1, x^1, b_1), \ldots, \mathrm{sat}(a_n, x^n, b_n)\big)$$

où

$$\mathrm{sat}(a, s, b) = \begin{cases} a & \text{si} \quad s \leqslant a \\ s & \text{si} \quad a \leqslant s \leqslant b \\ b & \text{si} \quad b \leqslant s \end{cases}$$



(fonction de saturation)
Les contraintes de bornes sont très fréquentes en optimisation.

Remarque 45. Les contraintes de bornes ou de positivité sont d'autant plus importantes que, en fait, tout problème d'optimisation sous contraintes d'égalité et d'inégalité non linéaires se ramène à un problème d'optimisation en dimension plus grande, avec contraintes d'égalité et contraintes de négativité sur les coordonnées. En effet, considérons le problème

$$\min_{\substack{h(x)=0 \\ g(x)\leqslant 0}} f(x) \tag{4.9}$$

avec $h : \mathbb{R}^n \to \mathbb{R}^p$ et $g : \mathbb{R}^n \to \mathbb{R}^q$. Pour tout $j \in \{1, \ldots, q\}$, la contrainte $g_j(x) \leqslant 0$ est équivalente à $g_j(x) = y_j$ et $y_j \leqslant 0$. Ainsi, en posant $y = (y_1, \ldots, y_q)$, le problème d'optimisation (4.9) est équivalent à

$$\min\{f(x) \;\mid\; h(x) = 0,\ g(x) - y = 0,\ y \leqslant 0\}$$

qui est un problème d'optimisation en dimension $n+q$ puisque, maintenant, l'inconnue est $(x, y) \in \mathbb{R}^{n+q}$, comportant des contraintes d'égalité non linéaires, et des contraintes d'inégalité très simples : négativité de certaines coordonnées.

Cette technique d'ajouter des “variables molles" (*slack variables* en anglais) est très utilisée en optimisation.

Remarque 46. Au passage, décrivons une autre réduction, utilisant également des variables molles. Le problème (4.9) se ramène à un problème avec contraintes d'égalité non linéaires seulement. Pour cela, il suffit de dire que, pour tout $j \in \{1, \ldots, q\}$, la contrainte $g_j(x) \leqslant 0$ est équivalente à $g_j(x) = -y_j^2$, avec $y_j \in \mathbb{R}$. Ainsi, en posant $G_j(x, y) = g_j(x) + y_j^2$ et $G = (G_1, \ldots, G_q)$, le problème d'optimisation (4.9) est équivalent à

$$\min_{\substack{h(x)=0 \\ G(x,y)=0}} f(x)$$

qui est un problème d'optimisation comportant uniquement des contraintes d'égalité non linéaires.

Cette technique peut d'ailleurs fournir une preuve alternative de KKT (mais, pour obtenir le signe des multiplicateurs de Lagrange, il faut toutefois supposer toutes les fonctions de classe C^2, puis appliquer la condition nécessaire d'optimalité d'ordre 2 ; alors que le théorème KKT ne nécessite que des fonctions de classe C^1).

Notons que, numériquement, cette technique, avec des y_j^2 qui sont non linéaires, n'est pas forcément préférable à la technique précédente, qui donnait des contraintes de négativité très simples.

Méthode de Newton projetée. Lorsque $\mathcal{C} = \mathbb{R}^n$, la méthode de Newton s'écrit

$$x_{k+1} = x_k - H_f(x_k)^{-1} \nabla f(x_k)$$

Comme précédemment, le défaut est que, même si $x_0 \in \mathcal{C}$, les itérés suivants x_k ne restent pas forcément dans $\mathcal{C}$. Comme précédemment, on projette chaque nouvel itéré sur $\mathcal{C}$. On obtient la méthode de Newton projetée :

$$\boxed{x_{k+1} = P_{\mathcal{C}} \left(x_k - H_f(x_k)^{-1} \nabla f(x_k) \right)}$$

Naturellement, on peut procéder de même avec les diverses variantes de la méthode de Newton que nous avons vues : quasi-Newton, Barzilai-Borwein.

La remarque 44 s'applique aussi à ces différents cas.

4.3.1.2 Méthodes de pénalisation

Les méthodes de pénalisation sont très populaires en optimisation sous contraintes, par la facilité de leur mise en oeuvre.

Principe général des méthodes de pénalisation. Considérons le problème de minimisation sous contraintes

$$\boxed{\min_{x \in \mathcal{C}} f(x)} \tag{4.10}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est une fonction et $\mathcal{C}$ est un sous-ensemble de $\mathbb{R}^n$ (on ne suppose rien de particulier ici). Notons $\chi_{\mathcal{C}} : \mathbb{R}^n \to [0, +\infty]$ la fonction (parfois appelée *fonction indicatrice* de $\mathcal{C}$) définie par

$$\chi_{\mathcal{C}}(x) = \begin{cases} 0 & \text{si} \quad x \in \mathcal{C} \\ +\infty & \text{si} \quad x \in \mathbb{R}^n \setminus \mathcal{C} \end{cases}$$

Trivialement, le problème d'optimisation sous contraintes (4.10) est équivalent au problème d'optimisation sans contraintes

$$\boxed{\min_{x \in \mathbb{R}^n} \left(f(x) + \chi_{\mathcal{C}}(x) \right)} \tag{4.11}$$

En effet, $f(x) + \chi_{\mathcal{C}}(x) = +\infty$ dès que $x \notin \mathcal{C}$, donc forcément le minimum est à chercher dans l'ensemble $\mathcal{C}$; mais dès que $x \in \mathcal{C}$, on a $f(x) + \chi_{\mathcal{C}}(x) = f(x)$. D'où l'équivalence des deux problèmes d'optimisation.

On a ainsi montré qu'un problème général d'optimisation sous contraintes est équivalent à un problème d'optimisation sans contraintes ! Cependant, cela est au prix de manipuler la fonction généralisée $f + \chi_{\mathcal{C}}$, qui prend ses valeurs dans $\mathbb{R} \cup \{+\infty\}$. En pratique cela ne résout donc rien car, en général, le minimum est atteint au bord de $\mathcal{C}$ (on sature la contrainte ! si on ne la sature pas, c'est alors qu'on avait un problème d'optimisation classique sans contraintes...), et c'est justement au bord de $\mathcal{C}$ que la fonction $f + \chi_{\mathcal{C}}$ n'est pas régulière : on ne peut la dériver en de tels points.

En pratique, l'idée est d'utiliser des fonctions $\varphi_\varepsilon : \mathbb{R}^n \to \mathbb{R}$, indicées par $\varepsilon > 0$, qui "approchent" la fonction indicatrice $\chi_{\mathcal{C}}$, i.e., pour tout $x \in \mathbb{R}^n$, $\varphi_\varepsilon(x) \to \chi_{\mathcal{C}}(x)$ lorsque $\varepsilon \to 0$, et on considère la famille (indicée par $\varepsilon > 0$) de problèmes pénalisés

$$\boxed{\min_{x \in \mathbb{R}^n} \left(f(x) + \varphi_\varepsilon(x) \right)} \tag{4.12}$$

On parle de *pénalisation exacte* dans le cas (4.11), lorsque la fonction de pénalisation est la fonction indicatrice $\chi_{\mathcal{C}}$. On parle de *pénalisation inexacte* dans le cas (4.12), lorsqu'on utilise une fonction de pénalisation φ_ε qui approche la fonction indicatrice $\chi_{\mathcal{C}}$.

En supposant que le problème (4.10) a une unique solution x^* et que, pour tout ε, le problème pénalisé (4.12) a une unique solution x_ε, on s'attend à ce que $x_\varepsilon \to x^*$ lorsque $\varepsilon \to 0$. C'est bien le cas sous des conditions naturelles. On a le résultat suivant.

Théorème 31. *On suppose que f est continue et infinie à l'infini, et que $\mathcal{C}$ est fermé non vide. Ainsi, le problème d'optimisation sous contraintes (4.10) a au moins une solution.*

On suppose que, pour tout $\varepsilon > 0$, la fonction $\varphi_\varepsilon : \mathbb{R}^n \to [0, +\infty)$ est continue et à valeurs positives et que la famille de fonctions $(\varphi_\varepsilon)_{\varepsilon > 0}$ converge vers la fonction indicatrice $\chi_{\mathcal{C}}$ de $\mathcal{C}$ au sens suivant :

- *convergence simple sur $\mathbb{R}^n$: pour tout $x \in \mathbb{R}^n$, $\varphi_\varepsilon(x) \to \chi_{\mathcal{C}}(x)$ quand $\varepsilon \to 0$;*
- *convergence uniforme sur tout compact de l'ouvert $\mathbb{R}^n \setminus \mathcal{C}$:*

$$\forall K \text{ compact} \subset \mathbb{R}^n \setminus \mathcal{C} \qquad \forall M > 0 \quad \exists \varepsilon_0 > 0 \quad | \quad \forall \varepsilon \in (0, \varepsilon_0] \quad \forall x \in K \qquad \varphi_\varepsilon(x) \geqslant M.$$

Alors, pour tout $\varepsilon > 0$, le problème pénalisé (4.12) a au moins une solution $x_\varepsilon \in \mathbb{R}^n$, la famille $(x_\varepsilon)_{\varepsilon > 0}$ est bornée, et toute valeur d'adhérence de $(x_\varepsilon)_{\varepsilon > 0}$ est une solution optimale de (4.10). Si de plus le problème (4.10) a une unique solution $x^* \in \mathcal{C}$ alors $x_\varepsilon \to x^*$ lorsque $\varepsilon \to 0$.

Remarque 47. On a utilisé ici un seul paramètre de pénalisation ε, mais on aurait pu considérer une famille à plusieurs paramètres. L'adaptation est triviale.

Démonstration. Comme f est continue et infinie à l'infini et $\mathcal{C}$ est fermé non vide, le problème d'optimisation sous contraintes (4.10) a au moins une solution $x^* \in \mathcal{C}$ (pas forcément unique). Notons que $\chi_{\mathcal{C}}(x^*) = 0$ puisque $x^* \in \mathcal{C}$.

Pour tout $\varepsilon > 0$, comme $\varphi_\varepsilon \geqslant 0$, la fonction $f + \varphi_\varepsilon$ est continue et infinie à l'infini, donc elle a au moins un minimiseur $x_\varepsilon \in \mathbb{R}^n$ (pas forcément unique). En particulier, on a

$$f(x_\varepsilon) \leqslant f(x_\varepsilon) + \varphi_\varepsilon(x_\varepsilon) \leqslant f(x^*) + \varphi_\varepsilon(x^*) \leqslant f(x^*) + 1 \qquad \forall \varepsilon \in (0, \varepsilon_0) \tag{4.13}$$

où $\varepsilon_0 > 0$ est fixé assez petit, car $\varphi_\varepsilon(x^*) \to \chi_{\mathcal{C}}(x^*) = 0$ quand $\varepsilon \to 0$. Donc la famille de réels $(f(x_\varepsilon))_{\varepsilon > 0}$ est bornée, et comme f est continue et infinie à l'infini, on en déduit que la famille $(x_\varepsilon)_{\varepsilon > 0}$ est bornée dans $\mathbb{R}^n$.

Soit alors $(x_{\varepsilon_k})_{k \in \mathbb{N}}$ une sous-suite convergeant vers $\bar{x} \in \mathbb{R}^n$. Montrons que $\bar{x}$ est un minimiseur du problème (4.10). D'après (4.13), on a $f(x_\varepsilon) \leqslant f(x^*) + \varphi_\varepsilon(x^*)$, donc en passant à la limite on obtient $f(\bar{x}) \leqslant f(x^*)$.

Montrons par l'absurde que $\bar{x} \in \mathcal{C}$. Si $\bar{x} \notin \mathcal{C}$: soit K une boule compacte de centre $\bar{x}$, contenue dans l'ouvert $\mathbb{R}^n \setminus \mathcal{C}$, et soit $k_0 \in \mathbb{N}$ assez grand pour que $x_{\varepsilon_k} \in K$ pour tout $k \geqslant k_0$. D'après (4.13), on a $f(x_{\varepsilon_k}) + \varphi_{\varepsilon_k}(x_{\varepsilon_k}) \leqslant f(x^*) + 1$, et comme $f(x_{\varepsilon_k}) \to f(\bar{x})$ lorsque $k \to +\infty$, on en déduit qu'il existe $k_1 \geqslant k_0$ tel que $\varphi_{\varepsilon_k}(x_{\varepsilon_k}) \leqslant f(x^*) - f(\bar{x}) + 2$ pour tout $k \geqslant k_1$. Comme $x_{\varepsilon_k} \to \bar{x}$ lorsque $k \to +\infty$ et comme φ_{ε_k} converge uniformément vers $+\infty$ sur le compact K, on obtient que $\varphi_{\varepsilon_k}(x_{\varepsilon_k}) \to +\infty$, d'où une contradiction.

Par conséquent $\bar{x} \in \mathcal{C}$. Comme $f(\bar{x}) \leqslant f(x^*)$, on a forcément $f(\bar{x}) = f(x^*)$ (puisque x^* minimise f sur $\mathcal{C}$) donc $\bar{x}$ est solution du problème (4.10) (mais on n'a pas forcément $\bar{x} = x^*$). C'est ce qu'on voulait démontrer. □

Le théorème ci-dessus donne des conditions générales sur la fonction de pénalisation.

Une fois qu'on a choisi une fonction de pénalisation, on a un problème d'optimisation sans contraintes, que l'on peut résoudre, par exemple, par une méthode de gradient (simple, pas optimal, gradient conjugué) ou de Newton.

En pratique on utilise majoritairement les deux procédés suivants : pénalisation externe, ou interne. On considère le problème d'optimisation

$$\min_{\substack{h(x)=0\\ g(x)\leqslant 0}} f(x) \tag{4.14}$$

Pénalisation externe. On pose

$$\varphi_\varepsilon(x) = \frac{1}{\varepsilon}\sum_{i=1}^{p} h_i(x)^2 + \frac{1}{\varepsilon}\sum_{j=1}^{q} \max(g_j(x),0)^2 \tag{4.15}$$

L'idée est ici que, lorsque $\varepsilon > 0$ est petit, dès que $h_i(x) \neq 0$ ou que $g_j(x) > 0$ alors $\varphi_\varepsilon(x)$ est très grand. Donc, minimiser $f + \varphi_\varepsilon$ force $h_i(x) \simeq 0$ et $g_j(x) \lesssim 0$. On est bien dans le cadre du théorème 31. On parle ici de pénalisation externe, parce qu'un minimiseur x_ε de φ_ε ne vérifie pas forcément les contraintes. Il ne les vérifie que de manière approchée lorsque ε est petit.

Bien sûr, on peut imaginer de nombreuses variantes à la fonction φ_ε ci-dessus. Par exemple au lieu de choisir un seul paramètre de pénalisation $\varepsilon > 0$, on peut prendre différents paramètres $\alpha_i > 0$, $\beta_j > 0$, selon les indices des contraintes. On peut choisir aussi d'autres fonctions que les carrés. Par exemple, on peut prendre

$$\varphi_{\alpha,\beta}(x) = \sum_{i=1}^{p} \frac{a_i(h_i(x))}{\alpha_i} + \sum_{j=1}^{q} \frac{b_j(g_j(x))}{\beta_j}$$

où $a_i : \mathbb{R} \to \mathbb{R}$ est C^1, positive, ne s'annule qu'en 0 (ci-dessus, on a pris $a_i(s) = s^2$), et $b_j : \mathbb{R} \to \mathbb{R}$ est C^1 et vérifie $b_j(s) = 0$ si $s \leqslant 0$ et $b_j(s) > 0$ si $s > 0$ (ci-dessus, on a pris $b_j(s) = \max(s,0)^2$).

Notons que, ci-dessus, on a pris des fonctions dérivables, afin d'assurer que φ_ε est différentiable. On pourrait choisir $a_i(s) = |s|$ par exemple, mais dans ce cas φ_ε ne serait pas différentiable (elle serait toutefois sous-différentiable, ce qui nécessiterait alors ensuite de mettre en oeuvre des méthodes d'optimisation sous-différentiable).

Dans le cas (4.15), on a

$$\nabla\varphi_\varepsilon(x) = \sum_{i=1}^{p} \frac{2h_i(x)}{\varepsilon}\nabla h_i(x) + \sum_{j=1}^{q} \frac{2\max(g_j(x),0)}{\varepsilon}\nabla g_j(x) \tag{4.16}$$

En effet, la fonction $s \mapsto \max(s,0)^2$ est dérivable sur $\mathbb{R}$, et $\frac{d}{ds}\max(s,0)^2 = 2\max(s,0)$.

Pénalisation interne. Dans la pénalisation interne, il faut supposer que le point initial x_0 de la méthode itérative (par exemple gradient) qu'on met en oeuvre pour résoudre le problème pénalisé, vérifie strictement les contraintes d'inégalité, i.e., est tel que $g(x_0) < 0$.

Un exemple de fonction de pénalisation interne est

$$\varphi_\varepsilon(x) = \frac{1}{\varepsilon}\sum_{i=1}^{p} h_i(x)^2 + \varepsilon\sum_{i=1}^{q} \frac{1}{g_j(x)^2} \tag{4.17}$$

L'idée est que, au cours des itérations, si le point x_k s'approche d'une frontière $g_j(x) = 0$, alors $\frac{1}{g_j(x_k)^2}$ devient très grand, ce qui n'est pas favorisé par la minimisation de φ_ε. On est de nouveau

dans le cadre du théorème 31. On parle ici de pénalisation interne, parce que les itérés x_k, ainsi que les minimiseurs de φ_ε, restent à l'intérieur du domaine des contraintes en inégalité. Notons que

$$\nabla\varphi_\varepsilon(x) = \sum_{i=1}^{p} \frac{2h_i(x)}{\varepsilon}\nabla h_i(x) - \sum_{j=1}^{q} \frac{2\varepsilon}{g_j(x)^3}\nabla g_j(x) \tag{4.18}$$

Comme précédemment, de nombreuses variantes existent. Une variante très souvent utilisée est la *pénalisation logarithmique* :

$$\varphi_\varepsilon(x) = \frac{1}{\varepsilon}\sum_{i=1}^{p} h_i(x)^2 - \varepsilon\sum_{j=1}^{q} \ln(-g_j(x)) \tag{4.19}$$

C'est la même idée : si le point x_k s'approche d'une frontière $g_j(x) = 0$, alors $-\ln(-g_j(x))$ devient très grand. Notons que

$$\nabla\varphi_\varepsilon(x) = \sum_{i=1}^{p} \frac{2h_i(x)}{\varepsilon}\nabla h_i(x) - \sum_{j=1}^{q} \frac{\varepsilon}{g_j(x)}\nabla g_j(x) \tag{4.20}$$

Remarque 48. Que ce soit pour une pénalisation externe ou interne, lorsqu'on met en oeuvre une telle méthode de pénalisation, on doit :

1. choisir une méthode de minimisation sans contrainte (par exemple, gradient simple ou pas optimal, gradient conjugué, Newton ou quasi-Newton), pour résoudre le problème de minimiser $f + \varphi_\varepsilon$ sur $\mathbb{R}^n$;
2. choisir un paramètre de pénalisation $\varepsilon > 0$.

Or, ce choix de ε n'est en fait pas évident ! d'après le théorème de convergence 31, on a intérêt à prendre $\varepsilon > 0$ petit. Mais si on prend ε trop petit, le calcul de $\varphi_\varepsilon(x)$ et de son gradient peuvent générer des erreurs importantes (lorsqu'on divise par ε). En pratique, on utilise souvent une suite $(\varepsilon_k)_{k\in\mathbb{N}}$ de paramètres de pénalisation, qui évolue au cours des itérations. Une méthode est de diviser ε par 2, toutes les 10 itérations par exemple ; et si on observe que, à une itération donnée, le problème est mal conditionné, alors on double la valeur de ε.

Estimation des multiplicateurs de Lagrange. Considérons par exemple la pénalisation externe (4.15). En un point x_ε qui minimise la fonction $f + \varphi_\varepsilon$, vu qu'il n'y a pas de contrainte dans ce problème pénalisé on doit avoir

$$\nabla f(x_\varepsilon) + \nabla\varphi_\varepsilon(x_\varepsilon) = 0$$

ce qui donne, en utilisant (4.16),

$$\nabla f(x_\varepsilon) + \sum_{i=1}^{p} \frac{2h_i(x_\varepsilon)}{\varepsilon}\nabla h_i(x_\varepsilon) + \sum_{j=1}^{q} \frac{2\max(g_j(x_\varepsilon), 0)}{\varepsilon}\nabla g_j(x_\varepsilon) = 0.$$

Or, on rappelle que KKT en x^* s'écrit

$$\nabla f(x^*) + \sum_{i=1}^{p} \lambda_i \nabla h_i(x^*) + \sum_{j=1}^{q} \mu_i \nabla g_j(x^*) = 0.$$

En comparant ces deux expressions, et en ayant en tête le théorème de convergence 31, on s'attend donc à ce que

$$\boxed{\frac{2h_i(x_\varepsilon)}{\varepsilon} \underset{\varepsilon\to 0}{\longrightarrow} \lambda_i \qquad \text{et} \qquad \frac{2\max(g_j(x_\varepsilon), 0)}{\varepsilon} \underset{\varepsilon\to 0}{\longrightarrow} \mu_j}$$

pour tous i, j. C'est le cas en effet, pourvu que le multiplicateur (normal) (λ, μ) soit unique ! Cela résulte d'un raisonnement facile par passage à la limite, calqué sur la deuxième démonstration de KKT donnée en Section 4.2.2.2.

Ce résultat est intéressant car il montre que la pénalisation donne un moyen d'estimer (de manière approchée) les multiplicateurs de Lagrange.

Bien sûr, on a des résultats équivalents en utilisant d'autres pénalisations (cf les expressions des gradients (4.18) ou (4.20)).

4.3.1.3 Méthode du Lagrangien augmenté

Comme on l'a vu, tout problème d'optimisation sous contraintes d'égalité et d'inégalité peut se ramener à un problème d'optimisation sous contraintes d'égalité seulement (soit en sélectionnant les indices actifs, soit en appliquant une astuce comme celle de la remarque 46). Dans cette section, on considère donc le problème

$$\min_{h(x)=0} f(x) \tag{4.21}$$

avec les mêmes notations que d'habitude. Supposons que ce problème a un minimiseur x^*, qui a un multiplicateur de Lagrange normal λ^*. Le Lagrangien de ce problème est alors

$$L(x, \lambda) = f(x) + \langle \lambda, h(x) \rangle$$

et la condition nécessaire de multiplicateurs de Lagrange est $\nabla_x L(x^*, \lambda^*) = 0$.

Soit $c > 0$ une constante. On "pénalise" le problème de la façon suivante (en conservant toutefois la contrainte) :

$$\min_{h(x)=0} \left(f(x) + \frac{c}{2} \|h(x)\|^2 \right) \tag{4.22}$$

qui est un problème complètement équivalent à (4.21) (car on a la contrainte $h(x) = 0$), et on considère le *Lagrangien augmenté*

$$L_c(x, \lambda) = f(x) + \langle \lambda, h(x) \rangle + \frac{c}{2} \|h(x)\|^2 = L(x, \lambda) + \frac{c}{2} \|h(x)\|^2$$

qui est le Lagrangien associé à ce nouveau problème.

Théorème 32. *On suppose que le problème (4.21) a un minimiseur x^*, associé à un multiplicateur (normal) λ^*, vérifiant $\nabla_x L(x^*, \lambda^*) = 0$ et vérifiant la condition suffisante d'ordre 2 : la Hessienne $\frac{\partial^2 L}{\partial x^2}(x^*, \lambda^*)$ restreinte à $\ker dh(x^*)$ est définie positive.*

Alors il existe $c_0 > 0$ tel que, pour tout $c \geqslant c_0$, la fonction $L_c(\cdot, \lambda^*)$ (où L_c est le Lagrangien augmenté) a un minimum local strict en x^*.

Démonstration. Comme x^* est un minimiseur de (4.21), on a $h(x^*) = 0$ et donc x^* est aussi un minimiseur de (4.22), et on a $\nabla_x L_c(x^*, \lambda^*) = 0$: autrement dit, λ^* est aussi un multiplicateur normal associé à x^* comme solution du nouveau problème (4.22).

Toujours en notant que $h(x^*) = 0$, on calcule que la Hessienne par rapport à x du Lagrangien augmenté au point (x^*, λ^*) est

$$Q_c = \frac{\partial^2 L_c}{\partial x^2}(x^*, \lambda^*) = \frac{\partial^2 L}{\partial x^2}(x^*, \lambda^*) + c\, dh(x^*)^\top dh(x^*)$$

i.e., c'est la somme $dh(x^*)^\top dh(x^*)$, qui est identifié à une forme quadratique symétrique positive, et de $\frac{\partial^2 L}{\partial x^2}(x^*, \lambda^*)$ qui, restreinte à $\ker dh(x^*)$, est supposée symétrique définie positive. Nous avons le lemme suivant.

Lemme 5. *Soient $A \in \mathcal{M}_n(\mathbb{R})$ une matrice symétrique, et soit $B \in \mathcal{M}_{p,n}(\mathbb{R})$, telles que si $x \in \ker B \setminus \{0\}$ alors $x^\top Ax > 0$. Alors il existe $c_0 > 0$ tel que pour tout $c \geqslant c_0$, $A + cB^\top B$ est symétrique définie positive.*

Démonstration. Par l'absurde, supposons qu'il existe une suite (x_k) de $\mathbb{R}^n \setminus \{0\}$ telle que $x_k^\top (A + kB^\top B)x_k \leqslant 0$. Quitte à diviser par $\|x_k\|^2$, on peut supposer que $\|x_k\| = 1$ pour tout k. Par compacité, à sous suite-près, $x_k \to x$ avec $\|x\| = 1$, et en passant à la limite dans l'inégalité $\frac{1}{k} x_k^\top (A + kB^\top B)x_k \leqslant 0$, on obtient $\|Bx\|^2 = x^\top B^\top Bx \leqslant 0$ et donc $Bx = 0$, i.e., $x \in \ker B \setminus \{0\}$. Par ailleurs, on a $x_k^\top Ax_k \leqslant -kx_k^\top B^\top Bx_k = -k\|Bx_k\|^2 \leqslant 0$ donc en passant à la limite, $x^\top Ax \leqslant 0$. On a donc obtenu l'existence d'un $x \in \ker B \setminus \{0\}$ tel que $x^\top Ax \leqslant 0$. Cela contredit l'hypothèse. □

D'après ce lemme, il existe $c_0 > 0$ tel que, pour tout $c \geqslant c_0$, la Hessienne $\frac{\partial^2 L_c}{\partial x^2}(x^*, \lambda^*)$ est symétrique définie positive. Comme $\nabla_x L_c(x^*, \lambda^*) = 0$, il s'ensuit que x^* est un minimiseur local strict de la fonction $L_c(\cdot, \lambda^*)$. □

Remarque 49. Dans les conditions du théorème, on a la propriété $L_c(x^*, \lambda^*) \leqslant L_c(x, \lambda^*)$ pour tout $x \in V$ où V est un voisinage de x^*. De plus, comme $h(x^*) = 0$, on a aussi $L_c(x^*, \lambda^*) = L_c(x^*, \lambda)$ pour tout $\lambda \in \mathbb{R}^p$. On a donc la double inégalité

$$L_c(x^*, \lambda) \leqslant L_c(x^*, \lambda^*) \leqslant L_c(x, \lambda^*) \qquad \forall (x, \lambda) \in V \times \mathbb{R}^p$$

qui signifie que (x^*, λ^*) est un *point selle* (local) du Lagrangien augmenté L_c (voir section 4.3.2.1 plus loin). La propriété de point selle est à la base des méthodes duales étudiées en section 4.3.2.

Déduisons de cette propriété l'algorithme du Lagrangien augmenté. Supposons que, à l'itération k, on dispose d'un paramètre de pénalisation c_k et d'un multiplicateur λ_k. Le théorème (32) suggère de minimiser la fonction $x \mapsto L_{c_k}(x, \lambda_k)$, donc, on cherche

$$x_k \in \operatorname{argmin} L_{c_k}(\cdot, \lambda_k)$$

et on a alors (vu qu'il n'y a aucune contrainte) $\frac{\partial L_{c_k}}{\partial x}(x_k, \lambda_k) = 0$, c'est-à-dire,

$$0 = df(x_k) + \lambda_k^\top dh(x_k) + c_k h(x_k)^\top dh(x_k) = df(x_k) + (\lambda_k^\top + c_k h(x_k)^\top) dh(x_k)$$

ce qui suggère de choisir $\lambda_{k+1} = \lambda_k + c_k h(x_k)$, car en effet, on cherche à faire en sorte que $x_k \simeq x^*$ et $\lambda_{k+1} \simeq \lambda^*$, vérifiant la condition de multiplicateurs de Lagrange $df(x^*) + (\lambda^*)^\top dh(x^*) = 0$.

Finalement, l'algorithme du Lagrangien augmenté est :

$$\boxed{\begin{aligned} x_k &\in \operatorname{argmin} L_{c_k}(\cdot, \lambda_k) \\ \lambda_{k+1} &= \lambda_k + c_k h(x_k) \end{aligned}}$$

Cet algorithme est assez proche de l'algorithme d'Uzawa dans les méthodes duales (voir la section 4.3.2.4). Il existe de nombreuses variantes. Tout d'abord, dans l'algorithme ci-dessus, on peut choisir la suite (c_k) de diverses manières, par exemple, constante assez grande ; ou bien croissante et majorée. Ensuite, le problème de minimisation en x peut être résolu de beaucoup de manières. Si on se contente d'une approximation, on peut par exemple faire un pas de gradient (de pas donné ρ) par rapport à l'itération précédente :

$$x_k = x_{k-1} - \rho \nabla_x L_{c_k}(x_{k-1}, \lambda_k)$$

ou bien, faire 10 itérations de gradient, ou 100...

4.3.1.4 Méthode de Lagrange-Newton

Comme dans la section précédente, on note que tout problème d'optimisation sous contraintes d'égalité et d'inégalité peut se ramener à un problème d'optimisation sous contraintes d'égalité seulement (soit en sélectionnant les indices actifs, soit en appliquant une astuce comme celle de la remarque 46). On considère donc le problème

$$\min_{h(x)=0} f(x) \tag{4.23}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ et $h : \mathbb{R}^n \to \mathbb{R}^p$ sont de classe C^2. Comme précédemment, supposons que ce problème a un minimiseur x^*, qui a un multiplicateur de Lagrange normal λ^*. Le Lagrangien de ce problème est alors

$$L(x, \lambda) = f(x) + \langle \lambda, h(x) \rangle$$

et la condition nécessaire de multiplicateurs de Lagrange est $\frac{\partial L}{\partial x}(x^*, \lambda^*) = 0$. On veut donc résoudre le système

$$\boxed{F(x, \lambda) = \begin{pmatrix} \nabla_x L(x, \lambda) \\ h(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}} \tag{4.24}$$

(dont la solution cherchée est (x^*, λ^*)) qui est un système de $n + p$ équations à $n + p$ inconnues (x, λ).

La *méthode de Lagrange-Newton* consiste à résoudre le système d'équations (4.24) par la méthode de Newton :

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} - dF(x_k, \lambda_k)^{-1} F(x_k, \lambda_k)$$

Pour que ce problème de Newton soit bien posé (voir le théorème sur la méthode de Newton), il faut assurer que la Jacobienne de F en (x^*, λ^*) soit inversible, i.e., que la différentielle $dF(x^*, \lambda^*) : \mathbb{R}^{n+p} \to \mathbb{R}^{n+p}$ soit inversible. On sait alors que la méthode de Newton converge, dans un voisinage de (x^*, λ^*). Or, cette différentielle est

$$dF(x^*, \lambda^*) = \begin{pmatrix} \frac{\partial^2 L}{\partial x^2}(x^*, \lambda^*) & dh(x^*)^\top \\ dh(x^*) & 0 \end{pmatrix}$$

qui est la *matrice de sensibilité* vue en Section 4.2.3.2 : elle est inversible sous les conditions que $dh(x^*) : \mathbb{R}^n \to \mathbb{R}^p$ soit surjective et que la Hessienne $\frac{\partial^2 L}{\partial x^2}(x^*, \lambda^*)$ restreinte à $\ker dh(x^*)$ soit définie positive.

En appliquant le théorème de convergence de la méthode de Newton (théorème 22), on conclut donc :

Proposition 6. *Si $dh(x^*) : \mathbb{R}^n \to \mathbb{R}^p$ est surjective et si la Hessienne $\frac{\partial^2 L}{\partial x^2}(x^*, \lambda^*)$ restreinte à $\ker dh(x^*)$ est définie positive, alors il existe $\varepsilon > 0$ tel que, si $\|(x_0, \lambda_0)\| \leqslant \varepsilon$, alors l'algorithme de Lagrange-Newton*

$$\boxed{\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \end{pmatrix} - \begin{pmatrix} \frac{\partial^2 L}{\partial x^2}(x_k, \lambda_k) & dh(x_k)^\top \\ dh(x_k) & 0 \end{pmatrix}^{-1} \begin{pmatrix} \nabla_x L(x_k, \lambda_k) \\ h(x_k) \end{pmatrix}}$$

converge (quadratiquement) vers (x^*, λ^*).

Remarque 50. Du point de vue du code, on présente l'itération sous la forme de la résolution du système linéaire

$$\underbrace{\begin{pmatrix} \frac{\partial^2 L}{\partial x^2}(x_k, \lambda_k) & dh(x_k)^\top \\ dh(x_k) & 0 \end{pmatrix}}_{M_k} \begin{pmatrix} x_{k+1} - x_k \\ \lambda_{k+1} - \lambda_k \end{pmatrix} = - \begin{pmatrix} \nabla_x L(x_k, \lambda_k) \\ h(x_k) \end{pmatrix} \tag{4.25}$$

et on note que la matrice M_k (de sensibilité) est symétrique et inversible.

Une autre façon de présenter ce système est la suivante. Supposons x_k connu et cherchons à déterminer x_{k+1}, d'après le système linéaire (4.25). En posant

$$s_k = x_{k+1} - x_k, \qquad g_k = \nabla f(x_k), \qquad h_k = h(x_k), \qquad A_k = \frac{\partial^2 L}{\partial x^2}(x_k, \lambda_k), \qquad C_k = dh(x_k),$$

le système linéaire (4.25) s'écrit

$$\begin{aligned} A_k s_k + C_k^\top (\lambda_{k+1} - \lambda_k) &= -g_k - C_k^\top \lambda_k \\ C_k s_k &= -h_k \end{aligned}$$

i.e.,

$$\boxed{\underbrace{\begin{pmatrix} A_k & C_k^\top \\ C_k & 0 \end{pmatrix}}_{M_k} \begin{pmatrix} s_k \\ \lambda_{k+1} \end{pmatrix} = - \begin{pmatrix} g_k \\ h_k \end{pmatrix}} \tag{4.26}$$

Remarque 51. Comme il s'agit d'une méthode de Newton, la méthode de Lagrange-Newton souffre du problème d'initialisation : il est difficile, en pratique, d'initialiser une méthode de Newton. En effet si on choisit une initialisation (x_0, λ_0) qui est trop loin de la solution (x^*, λ^*) cherchée (qu'on ne connaît pas !), alors la méthode risque de diverger.

Pour obtenir une bonne approximation préliminaire de (x^*, λ^*), une très bonne méthode consiste à faire tourner, au préalable, une méthode de pénalisation, même avec un choix de ε pas trop petit : si elle converge, d'après la section 4.3.1.2, cette méthode nous fournit une approximation $(\tilde{x}, \tilde{\lambda})$ (peut-être assez grossière) de (x^*, λ^*), et on peut espérer que l'initialisation $(x_0, \lambda_0) = (\tilde{x}, \tilde{\lambda})$ va tomber dans le domaine de convergence (c'est forcément le cas si ε est assez petit !) de la méthode de Newton, et donc, faire converger la méthode.

Variante : méthode de Wilson. Cette variante est basée sur la remarque suivante : le couple (s_k, λ_{k+1}) est solution du système (4.26) si et seulement si s_k est la solution du problème d'optimisation sous contrainte d'égalité

$$\boxed{\min_{C_k s + h_k = 0} \left(\frac{1}{2} s^\top A_k s + g_k^\top s \right)} \tag{4.27}$$

associée au multiplicateur de Lagrange (normal) λ_{k+1}.

Cette équivalence découle du raisonnement fait en Section 4.2.2.3 (où on a vu que l'unique solution du problème (4.4) est caractérisée par le système d'optimalité (4.5)), qu'on peut appliquer car, pour k assez grand, A_k est définie positive et C_k est surjective.

Ainsi, au lieu de résoudre le système linéaire (4.26), on peut résoudre le problème d'optimisation quadratique (4.27) sous contrainte d'égalité affine : c'est la méthode de Wilson.

4.3.1.5 Méthode SQP

La méthode de Wilson vue ci-dessus, qui consiste à résoudre le problème quadratique (4.27), est en fait, à l'itération k, une approximation linéaire-quadratique du problème

$$\min_{h(x)=0} L(x, \lambda_k)$$

avec laquelle, à partir de l'itéré k, on déterminerait x_{k+1}. Ici, linéaire-quadratique veut dire : approximation quadratique (i.e., à l'ordre 2) du critère de minimisation, et approximation linéaire

de la contrainte. En effet, connaissant x_k, on cherche $x_{k+1} = x_k + s_k$ tel que $0 \simeq h(x_{k+1}) \simeq h(x_k) + dh(x_k).s_k$ à l'ordre 1, ce qui donne la contrainte $C_k s_k + h_k = 0$. A l'ordre 2, on a

$$L(x_{k+1}, \lambda_k) \simeq L(x_k, \lambda_k) + \frac{\partial L}{\partial x}(x_k, \lambda_k).s_k + \frac{1}{2}\frac{\partial^2 L}{\partial x^2}(x_k, \lambda_k).(s_k, s_k)$$

et donc, dire que x_{k+1} minimise L par rapport à x, avec cette approximation d'ordre 2, revient à dire que s_k minimise

$$\frac{\partial L}{\partial x}(x_k, \lambda_k).s_k + \frac{1}{2}\frac{\partial^2 L}{\partial x^2}(x_k, \lambda_k).(s_k, s_k).$$

Or

$$\frac{\partial^2 L}{\partial x^2}(x_k, \lambda_k).(s_k, s_k) = s_k^\top A_k s_k,$$

et par ailleurs

$$\frac{\partial L}{\partial x}(x_k, \lambda_k).s_k = df(x_k).s_k + \lambda_k^\top dh(x_k).s_k = g_k^\top s_k + \lambda_k^\top C_k s_k = g_k^\top s_k - \lambda_k^\top h_k.$$

Ainsi, avec cette approximation, s_k est bien le minimiseur de (4.27).

La méthode SQP (*Sequential Quadratic Programming*, en français : méthode de programmation quadratique séquentielle, ou successive) s'appuie sur cette idée d'approximation linéaire-quadratique.

Considérons le problème d'optimisation sous contraintes d'égalité et d'inégalité

$$\min_{\substack{h(x)=0 \\ g(x)\leqslant 0}} f(x)$$

et supposons que x^* est un minimiseur, associé à un multiplicateur normal (λ^*, μ^*). Le Lagrangien de ce problème est $L(x, \lambda, \mu) = f(x) + \lambda^\top h(x) + \mu^\top g(x)$. La méthode SQP imite la méthode de Wilson ci-dessus, en faisant aussi une approximation linéaire des contraintes d'inégalité. L'itération k s'écrit :

calculer s_k, minimiseur de

$$\min_{\substack{dh(x_k).s+h(x_k)=0 \\ dg(x_k).s+g(x_k)\leqslant 0}} \left(\frac{1}{2}\frac{\partial^2 L}{\partial x^2}(x_k, \lambda_k).(s, s) + df(x_k).s\right)$$

dont les multiplicateurs de Lagrange associés sont λ_{k+1}, μ_{k+1}

puis poser

$$x_{k+1} = x_k + s_k$$

4.3.2 Méthodes duales

Considérons le problème d'optimisation sous contraintes d'égalité et d'inégalité

$$\boxed{\min_{\substack{h(x)=0\\ g(x)\leqslant 0}} f(x)} \tag{4.28}$$

où $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^q$ et $h : \mathbb{R}^n \to \mathbb{R}^p$. Comme on supposera dans la suite qu'on est dans le cas normal, on définit le Lagrangien

$$L(x, \lambda, \mu) = f(x) + \lambda^\top h(x) + \mu^\top g(x).$$

4.3.2.1 Point selle du Lagrangien

Définition 9. *On dit que $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q_+$ est un point selle de L si*

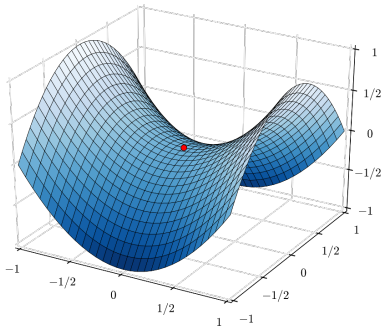
$$L(x^*, \lambda, \mu) \leqslant L(x^*, \lambda^*, \mu^*) \leqslant L(x, \lambda^*, \mu^*) \qquad \forall x \in \mathbb{R}^n \quad \forall \lambda \in \mathbb{R}^p \quad \forall \mu \in \mathbb{R}^q_+$$

autrement dit, si x^* minimise la fonction $x \mapsto L(x, \lambda^*, \mu^*)$ sur $\mathbb{R}^n$, et si (λ^*, μ^*) maximise la fonction $(\lambda, \mu) \mapsto L(x^*, \lambda, \mu)$ sur $\mathbb{R}^p \times \mathbb{R}^q_+$.

On appelle ce genre de point, un *point selle* (*saddle point* en anglais) ou un *point col*.
A droite, on voit le graphe de la fonction

$$f(x, y) = x^2 - y^2.$$

La fonction est convexe par rapport à x et concave par rapport à y : on dit qu'elle est *convexe-concave*. La surface $z = x^2 - y^2$ présente un point selle en $(0, 0)$.



Théorème 33. *Si $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q_+$ est un point selle de L alors x^* est un minimiseur du problème (4.28) (associé au multiplicateur de Lagrange normal (λ^*, μ^*) si de plus f, g et h sont différentiables).*

Démonstration. Montrons d'abord que x^* vérifie les contraintes, i.e., montrons que $h(x^*) = 0$ et $g(x^*) \leqslant 0$. Comme $L(x^*, \lambda, \mu) \leqslant L(x^*, \lambda^*, \mu^*)$ pour tout $(\lambda, \mu) \in \mathbb{R}^p \times \mathbb{R}^q_+$, on a

$$(\lambda - \lambda^*)^\top h(x^*) + (\mu - \mu^*)^\top g(x^*) \leqslant 0 \qquad \forall (\lambda, \mu) \in \mathbb{R}^p \times \mathbb{R}^q_+.$$

En prenant $\mu = \mu^*$ et $\lambda = \lambda^* + h(x^*)$, on en déduit d'abord que $\|h(x^*)\|^2 = 0$ donc $h(x^*) = 0$. Il reste donc $(\mu - \mu^*)^\top g(x^*) \leqslant 0$ pour tout $\mu \in \mathbb{R}^q_+$, et en prenant $\mu = \mu^* + e_j \in \mathbb{R}^q_+$ avec e_j le $j^{\text{ème}}$ vecteur de la base canonique de $\mathbb{R}^q$, on obtient $g_j(x^*) \leqslant 0$ pour tout $j \in \{1, \dots, q\}$.

En prenant maintenant $\mu = 0$, on obtient $(\mu^*)^\top g(x^*) \geqslant 0$, et comme $\mu^* \geqslant 0$ et $g(x^*) \leqslant 0$, on en déduit que $\mu^*_j g_j(x^*) = 0$ pour tout $j \in \{1, \dots, q\}$, ce qui est la condition de complémentarité. En particulier, $(\mu^*)^\top g(x^*) = 0$.

Montrons que x^* est un minimiseur du problème (4.28). Notons que, comme $h(x^*) = 0$ et $(\mu^*)^\top g(x^*) = 0$, on a $f(x^*) = f(x^*) + (\lambda^*)^\top h(x^*) + (\mu^*)^\top g(x^*) = L(x^*, \lambda^*, \mu^*)$. Comme $L(x^*, \lambda^*, \mu^*) \leqslant L(x, \lambda^*, \mu^*)$ pour tout $x \in \mathbb{R}^n$, on a donc

$$f(x^*) \leqslant f(x) + (\lambda^*)^\top h(x) + (\mu^*)^\top g(x) \qquad \forall x \in \mathbb{R}^n.$$

En particulier, pour $x \in \mathbb{R}^n$ vérifiant les contraintes $h(x) = 0$ et $g(x) \leqslant 0$ (donc $(\mu^*)^\top g(x) \leqslant 0$), on en déduit que $f(x^*) \leqslant f(x)$, ce qui est le résultat souhaité.

Lorsque f, g et h sont différentiables, le fait que x^* minimise $L(\cdot, \lambda^*, \mu^*)$ sur $\mathbb{R}^n$ implique que $\nabla_x L(x^*, \lambda^*, \mu^*) = 0$, ce qui donne KKT avec le multiplicateur normal (λ^*, μ^*). □

Le théorème 33 encourage à chercher des points selles du Lagrangien L. Cependant, il n'existe pas forcément de point selle : le Lagrangien n'est pas forcément convexe-concave. Par ailleurs, la réciproque du théorème 33 n'est pas vraie en général ; elle l'est toutefois lorsque f et g sont convexes et h affine.

Proposition 7. *On suppose que f, g et h sont différentiables, que f et g sont convexes et que h est affine. On suppose aussi que le problème (4.28) est qualifié. Les conditions suivantes sont équivalentes :*

- *Le triplet $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q_+$ est un point selle de L.*
- *x^* est minimiseur du problème (4.28), associé au multiplicateur normal (λ^*, μ^*).*
- *(x^*, λ^*, μ^*) vérifie KKT.*

Ce résultat complète la proposition 4.

Démonstration. Notons au préalable que, lorsque f et g sont convexes et h est affine, le Lagrangien $L(x, \lambda, \mu) = f(x) + \lambda^\top h(x) + \mu^\top g(x)$ est convexe par rapport à x (car $\mu \in \mathbb{R}^q_+$).

Si x^* est minimiseur du problème (4.28) (qui est qualifié), alors il existe $(\lambda^*, \mu^*) \in \mathbb{R}^p \times \mathbb{R}^q_+$ tel que $\nabla_x L(x^*, \lambda^*, \mu^*) = 0$ (égalité des multiplicateurs de Lagrange) et $\mu_j^* g_j(x^*) = 0$ pour tout $j \in \{1, \dots, q\}$ (conditions de complémentarité). Notons que l'égalité $h(x^*) = 0$ et les conditions de complémentarité impliquent que $L(x^*, \lambda^*, \mu^*) = f(x^*)$. Par convexité de L, la condition $\nabla_x L(x^*, \lambda^*, \mu^*) = 0$ est équivalente à dire que x^* minimise $L(\cdot, \lambda^*, \mu^*)$, i.e.,

$$L(x^*, \lambda^*, \mu^*) \leqslant L(x, \lambda^*, \mu^*) \qquad \forall x \in \mathbb{R}^n.$$

Par ailleurs, comme $h(x^*) = 0$ et $g(x^*) \leqslant 0$, on a $\mu^\top g(x^*) \leqslant 0$ pour tout $\mu \in \mathbb{R}^q_+$ et donc

$$L(x^*, \lambda, \mu) = f(x^*) + \lambda^\top h(x^*) + \mu^\top g(x^*) \leqslant f(x^*) = L(x^*, \lambda^*, \mu^*) \qquad \forall (\lambda, \mu) \in \mathbb{R}^p \times \mathbb{R}^q_+.$$

Donc (x^*, λ^*, μ^*) est un point selle de L. □

Bien que les hypothèses de la proposition 7 ne soient pas vérifiées en général, comme on l'a souvent fait précédemment, on considère qu'elles sont vraies au moins localement et approximativement, ce qui nous incite donc à chercher des points selles de L, i.e., à minimiser L par rapport à $x \in \mathbb{R}^n$ (minimisation sans contrainte) et maximiser L par rapport à $(\lambda, \mu) \in \mathbb{R}^p \times \mathbb{R}^q_+$ (maximisation sous contrainte de positivité).

C'est le principe des méthodes duales qu'on va voir plus loin.

D'autre part, bien que la réciproque du théorème 33 ne soit pas vraie en général, on a vu en section 4.3.1.3 que, en ramenant au préalable le problème comme un problème d'optimisation sous contraintes d'égalité seulement, on peut considérer le problème équivalent (4.22) qui conduit à définir le Lagrangien augmenté L_c, et on a vu en remarque 49 que, dans les conditions du théorème 32, le Lagrangien augmenté L_c admet un point selle (local) qui est exactement (x^*, λ^*). On peut donc en fait toujours se ramener à cette situation, en tout cas au moins localement.

4.3.2.2 Problème primal et problème dual

Posons

$$\mathcal{C} = \{x \in \mathbb{R}^n \ \mid \ h(x) = 0,\ g(x) \leqslant 0\}.$$

On rappelle que la fonction indicatrice $\chi_{\mathcal{C}} : \mathbb{R}^n \to [0, +\infty]$ est définie par

$$\chi_{\mathcal{C}}(x) = \begin{cases} 0 & \text{si} \quad x \in \mathcal{C} \\ +\infty & \text{si} \quad x \in \mathbb{R}^n \setminus \mathcal{C} \end{cases}$$

et que le problème d'optimisation sous contraintes (4.28) est équivalent au problème de minimiser $f + \chi_{\mathcal{C}}$ sur $\mathbb{R}^n$ (problème sans contrainte), i.e.,

$$\boxed{\min_{x \in \mathcal{C}} f(x) \quad \Longleftrightarrow \quad \min_{x \in \mathbb{R}^n} \left(f(x) + \chi_{\mathcal{C}}(x)\right)}$$

C'est la *pénalisation exacte*, vue en section 4.3.1.2.

La remarque cruciale est ici le fait que

$$\boxed{\chi_{\mathcal{C}}(x) = \sup_{\substack{\lambda \in \mathbb{R}^p \\ \mu \in \mathbb{R}^q_+}} \left(\lambda^\top h(x) + \mu^\top g(x)\right)}$$

En effet, dès que $h(x) \neq 0$ on a $\sup_{\lambda \in \mathbb{R}^p} \lambda^\top h(x) = +\infty$ (et sinon il vaut 0), et dès que $g_j(x) > 0$ pour un $j \in \{1, \ldots, q\}$ on a $\sup_{\mu \in \mathbb{R}^q_+} \mu^\top g(x) = +\infty$ (et sinon il vaut 0). On peut donc écrire le problème (4.28) sous la forme

$$\min_{x \in \mathbb{R}^n} \sup_{\substack{\lambda \in \mathbb{R}^p \\ \mu \in \mathbb{R}^q_+}} \left(f(x) + \lambda^\top h(x) + \mu^\top g(x)\right)$$

i.e., en reconnaissant le Lagrangien $L(x, \lambda, \mu) = f(x) + \lambda^\top h(x) + \mu^\top g(x)$, et en écrivant un inf plutôt qu'un min,

$$\boxed{\inf_{\substack{h(x)=0 \\ g(x) \leqslant 0}} f(x) = \inf_{x \in \mathbb{R}^n} \sup_{\substack{\lambda \in \mathbb{R}^p \\ \mu \in \mathbb{R}^q_+}} L(x, \lambda, \mu)} \tag{4.29}$$

Définition 10. *Le problème (4.29) s'appelle le problème primal. Il coïncide avec le problème d'optimisation initial (4.28).*

On observe que le problème primal s'écrit comme "$\inf \sup L$" : on maximise d'abord L par rapport à (λ, μ), puis on minimise par rapport à x.

On pourrait fort bien considérer de faire l'inverse ! Cela s'appelle alors le problème dual :

Définition 11. *On appelle problème dual le problème*

$$\boxed{\sup_{\substack{\lambda \in \mathbb{R}^p \\ \mu \in \mathbb{R}^q_+}} \inf_{x \in \mathbb{R}^n} L(x, \lambda, \mu)} \tag{4.30}$$

Dans la définition du problème dual, on minimise d'abord L par rapport à x, puis on maximise par rapport à (λ, μ). On a toujours

$$\boxed{\sup_{\substack{\lambda \in \mathbb{R}^p \\ \mu \in \mathbb{R}^q_+}} \inf_{x \in \mathbb{R}^n} L(x, \lambda, \mu) \leqslant \inf_{x \in \mathbb{R}^n} \sup_{\substack{\lambda \in \mathbb{R}^p \\ \mu \in \mathbb{R}^q_+}} L(x, \lambda, \mu)} \tag{4.31}$$

autrement dit, la valeur optimale du problème dual est toujours inférieure ou égale à la valeur optimale du problème primal. En effet, pour tout $(x', \lambda, \mu) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q_+$, on a $\inf_{x\in\mathbb{R}^n} L(x, \lambda, \mu) \leqslant L(x', \lambda, \mu)$, et en passant au sup dans cette inégalité, on obtient

$$\sup_{\substack{\lambda\in\mathbb{R}^p\\ \mu\in\mathbb{R}^q_+}} \inf_{x\in\mathbb{R}^n} L(x, \lambda, \mu) \leqslant \sup_{\substack{\lambda\in\mathbb{R}^p\\ \mu\in\mathbb{R}^q_+}} L(x', \lambda, \mu) \qquad \forall x' \in \mathbb{R}^n.$$

On prend alors l'inf sur les $x' \in \mathbb{R}^n$ dans cette inégalité, et on obtient (4.31).

Saut de dualité. On n'a pas forcément égalité dans (4.31). La différence s'appelle *saut de dualité*. On va voir ci-dessous un théorème assurant que le saut de dualité est nul dans certaines conditions.

Notons que, même lorsqu'il y a un saut de dualité non nul, résoudre le problème dual est tout de même intéressant car sa valeur optimale fournit un minorant de la valeur optimale du problème primal.

4.3.2.3 Théorème de dualité

La fonction

$$w(\lambda, \mu) = \inf_{x\in\mathbb{R}^n} L(x, \lambda, \mu)$$

s'appelle *fonction duale*. Le problème dual consiste à maximiser $w(\lambda, \mu)$ sur $\mathbb{R}^p \times \mathbb{R}^q_+$. Pour tout $x \in \mathbb{R}^n$ fixé, la fonction $(\lambda, \mu) \mapsto L(x, \lambda, \mu)$ est affine, donc w est un infimum de fonctions affines, donc w est concave (on a déjà vu ce fait).

Par contre, w n'est pas forcément différentiable. Toutefois, pour un $(\lambda, \mu) \in \mathbb{R}^p \times \mathbb{R}^q_+$ fixé, si x est l'unique minimiseur de $L(\cdot, \lambda, \mu)$ sur $\mathbb{R}^n$, alors d'après le théorème 29 de Danskin, w est différentiable en (λ, μ), et $\nabla_\lambda w(\lambda, \mu) = h(x)$ et $\nabla_\mu w(\lambda, \mu) = g(x)$.

Théorème 34. *S'il existe un point selle $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q_+$ de L, alors le saut de dualité est nul, x^* est solution de (4.28) (associé au multiplicateur de Lagrange normal (λ^*, μ^*) si de plus f, g et h sont différentiables) et*

$$\sup_{\substack{\lambda\in\mathbb{R}^p\\ \mu\in\mathbb{R}^q_+}} \inf_{x\in\mathbb{R}^n} L(x, \lambda, \mu) = \inf_{x\in\mathbb{R}^n} \sup_{\substack{\lambda\in\mathbb{R}^p\\ \mu\in\mathbb{R}^q_+}} L(x, \lambda, \mu) = L(x^*, \lambda^*, \mu^*) = f(x^*) = w(\lambda^*, \mu^*) = \sup_{\substack{\lambda\in\mathbb{R}^p\\ \mu\in\mathbb{R}^q_+}} w(\lambda, \mu)$$

Réciproquement, si (4.28) a un minimiseur x^* et s'il existe $(\lambda^*, \mu^*) \in \mathbb{R}^p \times \mathbb{R}^q_+$ tel que $f(x^*) = w(\lambda^*, \mu^*)$ alors (x^*, λ^*, μ^*) est un point selle de L.

Démonstration. Supposons que $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q_+$ est un point selle de L. D'après le théorème 33, on sait déjà que x^* est un minimiseur de (4.28), et on a vu dans la preuve de ce théorème que, de plus, $\mu^*_j g_j(x^*) = 0$ pour tout $j \in \{1, \dots, q\}$ (conditions de complémentarité), donc en particulier $(\mu^*)^\top g(x^*) = 0$, et $f(x^*) = L(x^*, \lambda^*, \mu^*)$. Comme (x^*, λ^*, μ^*) est un point selle de L, on a

$$f(x^*) = L(x^*, \lambda^*, \mu^*) = \inf_{x\in\mathbb{R}^n} L(x, \lambda^*, \mu^*) = w(\lambda^*, \mu^*).$$

De plus, l'inégalité (4.31) et l'égalité (4.29) impliquent que

$$\begin{aligned} w(\lambda^*, \mu^*) \leqslant \sup_{\substack{\lambda\in\mathbb{R}^p\\ \mu\in\mathbb{R}^q_+}} w(\lambda, \mu) &= \sup_{\substack{\lambda\in\mathbb{R}^p\\ \mu\in\mathbb{R}^q_+}} \inf_{x\in\mathbb{R}^n} L(x, \lambda, \mu) \\ &\leqslant \inf_{x\in\mathbb{R}^n} \sup_{\substack{\lambda\in\mathbb{R}^p\\ \mu\in\mathbb{R}^q_+}} L(x, \lambda, \mu) = \inf_{\substack{h(x)=0\\ g(x)\leqslant 0}} f(x) = f(x^*) = w(\lambda^*, \mu^*) \end{aligned}$$

et donc on n'a que des égalités ci-dessus, ce qui est le résultat cherché.

Réciproquement, supposons que (4.28) a un minimiseur x^* et qu'il existe $(\lambda^*, \mu^*) \in \mathbb{R}^p \times \mathbb{R}^q_+$ tel que $f(x^*) = w(\lambda^*, \mu^*)$. Par définition, $w(\lambda^*, \mu^*) \leqslant L(x^*, \lambda^*, \mu^*) = f(x^*) + (\mu^*)^\top g(x^*)$ (car $h(x^*) = 0$), donc $0 \leqslant (\mu^*)^\top g(x^*)$, mais comme $\mu^* \geqslant 0$ et $g(x^*) \leqslant 0$, on a forcément $\mu^*_j g_j(x^*) = 0$ pour tout $j \in \{1, \dots, q\}$ (conditions de complémentarité). Donc $L(x^*, \lambda^*, \mu^*) = f(x^*)$. De nouveau, par hypothèse et par définition, $f(x^*) = w(\lambda^*, \mu^*) \leqslant L(x, \lambda^*, \mu^*)$ pour tout $x \in \mathbb{R}^n$, donc $L(x^*, \lambda^*, \mu^*) \leqslant L(x, \lambda^*, \mu^*)$ pour tout $x \in \mathbb{R}^n$, ce qui est l'une des propriétés de point selle. D'autre part, comme $h(x^*) = 0$ et $g(x^*) \leqslant 0$, on a $L(x^*, \lambda, \mu) = f(x^*) + \lambda^\top h(x^*) + \mu^\top g(x^*) \leqslant f(x^*)$ pour tout $(\lambda, \mu) \in \mathbb{R}^p \times \mathbb{R}^q_+$, donc $L(x^*, \lambda, \mu) \leqslant L(x^*, \lambda^*, \mu^*)$ pour tout $(\lambda, \mu) \in \mathbb{R}^p \times \mathbb{R}^q_+$. On a démontré la propriété de point selle. □

C'est la première partie du théorème 34 qui est la plus importante. Elle montre que, lorsqu'on a un point selle de L, pour déterminer le triplet optimal (x^*, λ^*, μ^*) on peut résoudre indifféremment le problème $\inf \sup L$ ou $\sup \inf L$: on peut, dans n'importe quel ordre, minimiser L par rapport à $x \in \mathbb{R}^n$ et maximiser L par rapport à $(\lambda, \mu) \in \mathbb{R}^p \times \mathbb{R}^q_+$.

4.3.2.4 Méthodes duales

Méthode d'Uzawa. Le principe de la méthode d'Uzawa est d'utiliser un gradient projeté pour maximiser la fonction duale $w(\lambda, \mu) = \inf_{x \in \mathbb{R}^n} L(x, \lambda, \mu)$ (qui est concave) sur $\mathbb{R}^p \times \mathbb{R}^q_+$, autrement dit :

$$\begin{aligned} \lambda_{k+1} &= \lambda_k + \rho_k \nabla_\lambda w(\lambda_k, \mu_k) \\ \mu_{k+1} &= \max(\mu_k + \rho_k \nabla_\mu w(\lambda_k, \mu_k), 0) \end{aligned}$$

où le pas $\rho_k > 0$ peut être choisi fixe ou variable (pas optimal par exemple). Notons bien le signe $+\rho_k$ car on maximise w. On suppose donc ici que w est différentiable, ce qui (comme on l'a vu précédemment) est le cas lorsque la fonction $x \mapsto L(x, \lambda_k, \mu_k)$ a un minimiseur unique, noté x_k. L'algorithme d'Uzawa est alors le suivant.

A l'itération k, connaissant (λ_k, μ_k) :

1. On calcule x_k, minimiseur de $L(x, \lambda_k, \mu_k) = f(x) + \lambda_k^\top h(x) + \mu_k^\top g(x)$ sur $\mathbb{R}^n$.
2. On met à jour les multiplicateurs :

$$\begin{aligned} \lambda_{k+1} &= \lambda_k + \rho_k \nabla_\lambda L(x_k, \lambda_k, \mu_k) = \lambda_k + \rho_k h(x_k) \\ \mu_{k+1} &= \max(\mu_k + \rho_k \nabla_\mu L(x_k, \lambda_k, \mu_k), 0) = \max(\mu_k + \rho_k g(x_k), 0) \end{aligned}$$

La première étape peut avoir, naturellement, beaucoup de variantes.

Méthode d'Arrow-Hurwicz. Par rapport à l'algorithme d'Uzawa ci-dessus, dans la méthode d'Arrow-Hurwicz fait un pas de gradient seulement dans la première étape, ce qui donne l'algorithme :

$$\begin{aligned} x_{k+1} &= x_k - \rho_k \nabla_x L(x_k, \lambda_k, \mu_k) \\ \lambda_{k+1} &= \lambda_k + \rho_k \nabla_\lambda L(x_k, \lambda_k, \mu_k) = \lambda_k + \rho_k h(x_k) \\ \mu_{k+1} &= \max(\mu_k + \rho_k \nabla_\mu L(x_k, \lambda_k, \mu_k), 0) = \max(\mu_k + \rho_k g(x_k), 0) \end{aligned}$$

Chapitre 5

Conclusion et compléments

Précédemment, on a vu un ensemble de méthodes classiques en optimisation (déterministe et différentiable), et on a vu comment les implémenter. Le domaine de l'optimisation est très vaste, et selon les problèmes considérés :

- Les problèmes ne sont pas forcément déterministes, et peuvent être de nature stochastique : cela conduit à *l'optimisation stochastique* (voir l'algorithme du gradient stochastique en section 5.2, qui est très classique et peut être considéré comme une introduction à la théorie de l'optimisation stochastique).
- Les fonctions en jeu ne sont pas forcément différentiables, ce qui conduit à *l'optimisation non lisse* ou *optimisation non différentiable*, utilisant le concept de sous-différentiel, de conjuguée de Fenchel, d'opérateur proximal, etc.
  Lorsque les fonctions en jeu ne sont même pas sous-différentiables, on peut vouloir faire de l'optimisation sans dérivée (comme l'optimisation génétique par exemple).
- Jusqu'à maintenant, on a étudié ce qu'on appelle *l'optimisation continue* : les inconnues varient en effet dans des espaces continus (comme $\mathbb{R}^n$). Lorsque les inconnues varient dans un espace discret (comme $\mathbb{Z}^n$), on parle d'*optimisation discrète*, ou *optimisation combinatoire*. Souvent, les problèmes d'optimisation impliquent à la fois des inconnues continues et discrètes : on parle d'*optimisation mixte*. Cela est aussi lié à certains domaines de l'informatique où on parle aussi de *recherche opérationnelle*, en lien avec la théorie des graphes (qui donne aussi des approches à l'optimisation globale, discutée ci-dessous).
- Tous les résultats donnés précédemment sont de nature *locale* : à part dans le cas convexe, les minimiseurs sont locaux. Assurer qu'un minimiseur est global est difficile. Nos conditions sont en effet de nature différentielle : gradient nul, dérivée seconde positive. Tout cela n'assure au mieux que l'existence de minimiseur local, à moins d'avoir de la convexité. On parle donc d'optimisation locale, par rapport à la théorie de l'*optimisation globale*, qui fait appel à de tout autres considérations et méthodes :
  - des méthodes déterministes, comme la méthode des intervalles, la méthode de séparation et évaluation (en anglais “branch and bound") ;
  - des méthodes basées sur la géométrie algébrique (polynômes positifs, sommes de carrés de polynômes) ;
  - des méthodes stochastiques, faisant appel à des processus aléatoires, des considérations statistiques et des échantillonnages, comme le krigeage, le swarming, l'optimisation génétique, les méthodes de Monte-Carlo.

Les méthodes d'optimisation globale sont souvent basées sur des heuristiques (on parle de "métaheuristique"), inspirées par des systèmes naturels, par exemple :

— en physique : *méthode du recuit simulé*, s'inspirant de processus utilisés en métallurgie ;

— en biologie évolutionnaire : *algorithmes génétiques* ;

— en éthologie (étude du comportement des espèces) : *algorithmes de colonies de fourmis*, en anglais "ant colony optimization", ou bien la méthode d'*optimisation par essaims particulaires*, en anglais "particle swarm optimization".

Ces dernières méthodes connaissent actuellement un développement important, en lien avec les études de dynamiques collectives (alignement, consensus, auto-organisation). Dans le domaine de *l'Intelligence Artificielle* (IA), on parle même de "swarm intelligence" (intelligence de l'essaim), faisant référence à la capacité d'un groupe, d'un système connecté, a priori décentralisé, à s'auto-organiser.

- On a étudié des problèmes d'optimisation avec un seul critère à minimiser (ou maximiser). Dans la théorie de l'*optimisation multi-objectifs*, ou *multi-critères*, on a plusieurs fonctions à minimiser. Comme on ne peut généralement pas trouver de point qui les minimise toutes en même temps, on définit des ordres de préférence sur les objectifs, ce qui conduit à différentes stratégies possibles : équilibres de Pareto, de Nash, de Stackelberg (liés à la théorie des jeux).

Naturellement, tous les points mentionnés ci-dessus peuvent être combinés.

Par ailleurs, on souhaite appliquer l'optimisation (notamment) à des problèmes en grande dimension : science des données, data science, big data, problèmes liés à l'IA. Dans de tels problèmes, outre les difficultés et nouveautés mentionnées ci-dessus, une autre difficulté surgit rapidement : celle de calculer les différentielles et/ou Hessiennes des fonctions en jeu. Jusqu'à présent, dans les méthodes de gradient, de Newton, Uzawa, etc, on a supposé qu'on savait calculer *explicitement* les différentielles et Hessiennes des fonctions. Mais en pratique, ce calcul peut s'avérer difficile, ou être une source importante d'erreurs de codage (erreurs d'indices par exemple, lorsque les fonctions sont données de manière discrétisée et qu'on veut faire des différences finies). Pour pallier à ce problème on peut combiner des routines d'optimisation à des routines de *différentiation automatique*. C'est l'objet de la section 5.1 ci-dessous.

5.1 Utilisation de `AMPL`

Introduction. `AMPL` (*A Mathematical Programming Language*, voir https://ampl.com) est un langage / interface extrêmement simple d'utilisation, qui permet de coder très rapidement des problèmes d'optimisation pourtant difficiles. `AMPL` combine de la *différentiation automatique* avec une routine d'optimisation que l'on peut choisir.

La *différentiation automatique* permet d'éviter complètement d'avoir à calculer les différentielles et Hessiennes du problème : l'outil calcule les dérivées des fonctions qu'on lui donne. Mais contrairement à `Maple` ou `Mathematica` (ou à certaines calculatrices) qui font du *calcul formel* (autrement dit, ces logiciels "savent" que, par exemple la dérivée de x^2 est $2x$), la différentiation automatique consiste à calculer les dérivées de manière numérique, mais à la précision de l'ordinateur (autrement dit, à 10^{-14} près).

L'idée de base de la différentiation automatique (qui a été développée depuis les années 60) est basée sur l'idée d'une différentiation en passant en complexes : au lieu de calculer une dérivée approchée comme un taux d'accroissement par la formule

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \mathrm{o}(1)$$

lorsque $h \to 0$, qui repose, comme on le sait, sur le développement limité à l'ordre 1

$$f(x+h) = f(x) + hf'(x) + \mathrm{o}(h),$$

on suppose ici que f admet une extension holomorphe (au moins au voisinage de x) et on écrit le développement limité à l'ordre 2, en complexes,

$$f(x+ih) = f(x) + ihf'(x) - \frac{1}{2}f''(x) + \mathrm{o}(h^2).$$

La grosse différence maintenant est que

$$f'(x) = \mathrm{Im}\frac{f(x+ih)}{h} + \mathrm{o}(h)$$

On voit qu'on a gagné un ordre dans l'approximation ! De plus, ce n'est plus une différence finie : dans la formule ci-dessus, on prend la partie imaginaire de $f(x+ih)$ (à laquelle, pour des raisons informatiques, on peut accéder directement) qu'on divise par h ; alors que, dans la différence finie, on faisait la différence $f(x+h) - f(x)$: autrement dit, la somme de deux termes, ce qui génère d'importantes erreurs d'arrondi. Ainsi, la différentiation automatique, basée sur ce principe, permet de calculer des dérivées à l'ordre de précision de la machine, soit à 10^{-14} près.

Cette idée (remarquable) a été largement développée depuis les années 60. Des dizaines d'années de recherches ont about à des outils sophistiqués, très efficaces, pour dériver des fonctions (la difficulté principale étant de calculer des dérivées de fonctions composées, mais ce n'est pas le lieu de développer cet aspect ici). `AMPL` inclut de tels procédés de différentiation automatique. C'est très intéressant car, dans un code, cela évite d'avoir à fournir les différentielles et Hessiennes des fonctions en jeu.

Comme dit ci-dessus, `AMPL` permet de faire appel à une routine d'optimisation de notre choix, et calcule par différentiation automatique les différentes dérivées nécessaires. Il existe de nombreuses routines d'optimisation, très efficaces (et qui résultent de dizaines d'années de développements), dans des domaines divers. En optimisation non linéaire sous contraintes non linéaires, à l'heure actuelle, les deux routines les plus efficaces semblent être `IpOpt` et `Knitro`.

En gros, `IpOpt` (*Interior Point Optimizer*) est une routine d'optimisation (désormais codée en `C++`), basée sur une méthode de point intérieur (pénalisation interne ; mais c'est une routine très élaborée), voir la documentation :

https://coin-or.github.io/Ipopt/

Elle est gratuite et peut être installée sur n'importe quelle machine, voir

https://github.com/coin-or/Ipopt

`Knitro` (*Nonlinear Interior point Trust Region Optimization*) est une routine commerciale (payante) qui, majoritairement, est basée sur une méthode SQP :

https://www.artelys.com/fr/solveurs/knitro/

Du point de vue efficacité générale sur des problèmes généraux, `IpOpt` et `Knitro` ont des performances comparables. Actuellement, elles sont considérées comme étant les plus efficaces pour résoudre des problèmes généraux d'optimisation non linéaire sous contraintes non linéaires.

Utilisation pratique. On peut utiliser `AMPL` combiné avec une routine d'optimisation (on conseille `IpOpt` qui est gratuit) pour résoudre très facilement des problèmes d'optimisation non linéaire. Coder en `AMPL` est très simple, voici un exemple qui se passe de commentaire :

```
## Exemple simple de code AMPL
# Declaration des inconnues :
```

```
var x1; #, default 0;
var x2; #, default 0;

# Definition de la fonction a minimiser :
minimize mafun: x1^2+x2^2-14*x1-6*x2-7;

# Definition des contraintes :
subject to c1: x1+x2 <= 2;
subject to c2: x1+2*x2 <= 3;

# Choix du solver :
option solver ipopt;   # ligne a commenter si on lance le code sur le site AMPL
solve;

# Affichage des resultats :
display x1, x2;          # valeurs optimales de x1, x2
display c1, c2, mafun;   # ce sont les valeurs des multiplicateurs de Lagrange
```

Pour lancer un tel code, voici plusieurs solutions.

En ligne, sur le site `NEOS Solvers` : Le site web

http://www.neos-server.org/neos/solvers/index.html

met à disposition des utilisateurs un cluster de machines, sur lesquelles on peut lancer gratuitement des codes `AMPL`, combinés à quantité de routines d'optimisation (open-source ou non). L'avantage est qu'il n'y a rien à installer sur sa propre machine. Il faut juste accéder au réseau internet.

Dans le cas qui nous occupe, choisir *Nonlinearly constrained optimization*, et choisir `IpOpt` ou `Knitro` avec `AMPL` input et lancer en ligne.

Il n'y a aucune restriction en nombre de variables : on peut lancer, par ce biais, des codes très complexes et avec un très grand nombre d'inconnues. Lorsque le calcul est fini, on reçoit un mail.

Conseil : lorsqu'on lance un code sur `NEOS`, mieux vaut s'assurer à l'avance qu'on n'a pas fait une faute de frappe dans le code (par exemple, un point virgule oublié...). Il est alors conseillé de considérer la solution suivante.

Le site web `NEOS` propose beaucoup d'autres routines d'optimisation : optimisation combinatoire, globale, linéaire, en entiers, mixte, conique, stochastique, etc.

En installant `AMPL` : Le site https://ampl.com propose d'installer gratuitement `AMPL` à condition de l'utiliser avec une routine open-source comme `IpOpt` (cela nécessite alors d'avoir une connexion réseau).

On peut acheter la licence `AMPL` (une centaine de dollars seulement) pour avoir la version complète, et installer `IpOpt` sur sa propre machine, pour éviter d'avoir besoin d'une connexion réseau.

Notons qu'on peut désormais appeler `AMPL` depuis `Python` (voir en ligne les nombreux exemples). Le site web

https://colab.ampl.com/

offre la possibilité de tester des codes en ligne via un `Jupyter Notebook`.

Cette solution de type "boîte noire", `AMPL` + `IpOpt` (ou autre routine...) est à l'heure actuelle ce qu'on fait de plus efficace pour l'optimisation de problèmes généraux d'optimisation non linéaire sous contraintes non linéaires, sans devoir rentrer dans le code ou dans des complications techniques.

Apprendre à coder en AMPL n'est vraiment pas difficile. On trouve sur le web de nombreux exemples de routines AMPL, par exemple :
https://ampl.com/learn/ampl-book/example-files/
https://vanderbei.princeton.edu/ampl/nlmodels/
par exemple https://vanderbei.princeton.edu/ampl/nlmodels/robotarm/index.html
http://plato.asu.edu/pdecon.html
et beaucoup d'autres. En particulier, on peut étudier le livre d'AMPL,
https://ampl.com/learn/ampl-book/

Voici ci-dessous un exemple de code AMPL d'optimisation sur un problème d'EDP :

```
## Source d'exemples : http://plato.asu.edu/pdecon.html
##  http://plato.la.asu.edu/papers/paper91/node2.html
##
## Controle optimal de l'equation de Burgers :
##   y_t = nu y_{xx} - y y_x   sur (0,T)*(0,1)     y=y(t,x)
##   y(0,x) = 0
##   y_x(t,0) = 0,  y_x(t,1) = u(t)    (controle Neumann a droite)
##   a <= u(t) <= b
##   min  \int_0^T \int_0^1 (y(t,x)-yd(x))^2 dx dt + alpha*\int_0^T u(t)^2 dt
##   avec  yd(x)=1-x^2

param Nx = 300; param Nt = 300;
param T = 20; param nu = 0.01; param alpha = 0.1;
param yd{j in 0..Nx} = 1-(j/Nx)^2;

var y {0..Nt, 0..Nx};
var u {i in 0..Nt-1} >= -10, <= 10;

minimize cost: T/Nt*1/Nx*(sum{i in 0..Nt-1, j in 0..Nx-1}(y[i,j]-yd[j])^2)
+ alpha*T/Nt*(sum{i in 0..Nt-1}(u[i]^2));

var ydot {i in 0..Nt, j in 1..Nx-1} =
         nu*(y[i,j-1]-2*y[i,j]+y[i,j+1])*Nx^2-y[i,j]*(y[i,j+1]-y[i,j-1])*Nx/2;
subject to pde_dyn {i in 0..Nt-1, j in 1..Nx-1}:
         y[i+1,j] = y[i,j] + T/(2*Nt)*(ydot[i,j]+ydot[i+1,j]);

subject to init_state {j in 0..Nx}: y[0,j] = 0;
subject to left_bc    {i in 0..Nt}: -y[i,2]+4*y[i,1]-3*y[i,0] = 0;
subject to right_bc   {i in 1..Nt}: (y[i,Nx-2]-4*y[i,Nx-1]+3*y[i,Nx])/2*Nx = u[i-1];

option solver ipopt;
options ipopt_options "linear_solver=mumps  max_iter=2000";
solve;

printf: " # cost = %24.16e\n", cost >> out.txt;
printf: " # Nx = %d\n", Nx >> out.txt;
printf: " # Nt = %d\n", Nt >> out.txt;
printf: " # Data\n" >> out.txt;
printf {i in 0..Nt-1}: " %24.16e %24.16e\n", i*T/Nt, u[i] >> out.txt;
```

```
printf: " %24.16e %24.16e\n", T, u[Nt-1] >> out.txt;
printf{i in 0..Nt, j in 0..Nx}: " %24.16e\n", y[i,j] >> out.txt;
end;


## En t, on a utilise une discretisation de type RK2 implicite.
## En x, c'est une difference finie standard.
##
## Au bord gauche, en x, on a utilise une difference finie decentree d'ordre 2 :
##    f(x+h) = f(x) + hf'(x) + h^2/2 f''(x+h)
##    =>  f'(x) = (f(x+h)-f(x))/h - h/2 f''(x+h)
##    =>  f'(x) = (y_{i+1}-y_i)/h - (y_{i+2}-2y_{i+1}+y_i)/2h = (-y_{i+2}+4y_{i+1}-3y_i)/2h
##
## Au bord droite, en x, on a utilise une difference finie decentree d'ordre 2 :
##    f(x-h) = f(x) - hf'(x) + h^2/2 f''(x-h)
##    =>  f'(x) = (f(x)-f(x-h))/h + h/2 f''(x-h)
##    =>  f'(x) = (y_i-y_{i-1})/h + (y_i-2y_{i-1}+y_{i-2})/2h = (3y_i-4y_{i-1}+y_{i-2})/2h
```

Ci-dessus, on a choisi de renvoyer les résultats dans un fichier texte `out.txt`, qu'on lit ensuite dans `Matlab` pour l'affichage graphique, à l'aide du fichier `Matlab` ci-dessous :

```
fid = fopen('out.txt', 'r') ;

C = fscanf(fid,' # cost = %f',[1 1]);
Nx = fscanf(fid,' # Nx = %d',[1 1]);
Nt = fscanf(fid,' # Nt = %d',[1 1]);
s = fscanf(fid,' # %s',[1 1]);
tempscont = fscanf(fid,'%f',[2 Nt+1]);
mat_y = fscanf(fid,'%f',[1 (Nt+1)*(Nx+1)]);
fclose(fid);

tempscont = tempscont';
t=tempscont(:,1); u=tempscont(:,2);
y = reshape(mat_y,Nx+1,Nt+1); y=y';

x = 0:1/Nx:1;

subplot(121); plot(t,u); title('t -> u(t)') ;
subplot(122); hold on;
plot(x,y(1,:),'black');
for i=2:5:length(t)-1
plot(x,y(i,:),'b');
pause(0.2)
end
plot(x,y(end,:),'red'); title('Curves x -> y(t,x)') ; hold off;
```

Bien entendu, on peut procéder différemment. L'affichage graphique des résultats peut se faire dans `Scilab`, ou bien directement avec `Python`.

Beaucoup d'astuces et de manières de coder existent, on trouve quantité de matériel et de "templates" sur le web, dont on pourra s'inspirer.

`AMPL` est un langage d'une très grande simplicité et d'une très grande puissance, mais sa licence est payante. Il existe toutefois d'autres alternatives, gratuites, pour faire de la différentiation automatique. Les solutions existantes sont nombreuses mais en général n'atteignent pas l'efficacité ou la simplicité d'`AMPL`, à l'exception notable de `CasADi` :

https://web.casadi.org

qui est une excellente solution de différentiation automatique pour faire de l'optimisation et du contrôle optimal, en étant combinable à une routine d'optimisation comme `IpOpt`. Bien qu'étant un peu moins simple d'utilisation que `AMPL`, `CasADi` est aussi efficace et est utilisable en `Python` (ou même directement en `C++`). Le site web contient de nombreux exemples d'utilisation.

5.2 Gradient stochastique

L'algorithme du gradient stochastique est très utilisé en science des données, apprentissage, machine learning. On s'intéresse ici à minimiser sur $\mathbb{R}^n$ (sans contrainte) une fonction F qui est définie comme la moyenne de fonctions sur un grand nombre d'évènements. De manière probabiliste, une moyenne est une espérance, et ce problème s'écrit sous la forme

$$\boxed{\min_{x\in\mathbb{R}^n} F(x) \qquad \text{avec} \qquad F(x) = \mathbb{E}f(x,W) = \int_\Omega f(x,W(\omega))\,dP(\omega) = \int_X f(x,w)\,d\mu(w)} \tag{5.1}$$

autrement dit, pour tout $x \in \mathbb{R}^n$, $F(x)$ est l'espérance de la variable aléatoire $f(x,W)$, (Ω,P) est un espace probabilisé (univers d'évènements ω), $W : \Omega \to X$ est une variable aléatoire, et $\mu = W_*P$ est la loi de W (loi de probabilité sur X qui est l'image de P par W).

On suppose f différentiable par rapport à x, de sorte que

$$\nabla F(x) = \mathbb{E}\nabla_x f(x,\cdot) = \int_\Omega \nabla_x f(x,W(\omega))\,dP(\omega) = \int_X \nabla_x f(x,w)\,d\mu(w).$$

En pratique, pour estimer une espérance, on utilise une méthode de Monte-Carlo et on procède à un échantillonnage, i.e., on fait une sélection de N évènements $\omega_i \in \Omega$ qui donnent des échantillons $w_i = W(\omega_i)$, $i = 1,\dots,N$ (SAA : *Sample Average Approximation*). Lorsque μ est la mesure de probabilité uniforme sur X, on a alors (en approximation)

$$F(x) = \frac{1}{N}\sum_{i=1}^N f_i(x) \qquad \text{et} \qquad \nabla F(x) = \frac{1}{N}\sum_{i=1}^N \nabla f_i(x)$$

avec $f_i(x) = f(x,w_i)$. Autrement dit, la fonction F à minimiser est la moyenne d'un très grand nombre N de fonctions f_i. En machine learning, N est le nombre de données à partir desquelles on veut "apprendre" les paramètres $x \in \mathbb{R}^n$ du modèle (voir section 5.3).

Pour minimiser la fonction F, un algorithme de descente de gradient à pas variable $t_k > 0$ serait du type

$$x_{k+1} = x_k - t_k \nabla F(x_k)$$

mais le calcul de $\nabla F(x_k)$ est lourd, puisqu'il réclame de calculer la moyenne de tous les gradients $\nabla_x f(x,w)$, c'est-à-dire, dans le cas échantillonné, le calcul des N gradients $\nabla f_i(x_k)$, alors que N est très grand.

L'algorithme du gradient stochastique n'utilise qu'un seul gradient pour chaque itération :

$$\boxed{x_{k+1} = x_k - t_k \nabla_x f(x_k, w_{k+1})}$$

où $w_{k+1} = W(\omega_{k+1}) \in X$ est un évènement aléatoire, tiré aléatoirement dans l'espace probabilisé (X, μ), indépendamment du tirage précédent ν_k. Les tirages aléatoires $w_k = W(\omega_k)$ forment une suite $(w_k)_{k\in\mathbf{N}^*}$ qui est une réalisation d'un échantillon (de taille infinie) de la suite de variables aléatoires $(W_k)_{k\in\mathbf{N}^*}$ i.i.d. ("independently identically distributed"), c'est-à-dire des variables aléatoires indépendantes et qui ont toutes la même loi que W. Dans le cas échantillonné mentionné ci-dessus, cela s'écrit

$$x_{k+1} = x_k - t_k \nabla f_{i_{k+1}}(x_k)$$

où i_{k+1} est un indice tiré aléatoirement et uniformément dans $\{1, \dots, N\}$, chaque tirage d'indice étant indépendant du précédent.

Théorème 35. *On suppose que F est C^1 et α-convexe : il existe donc un unique minimiseur $x^* \in \mathbb{R}^n$. On suppose qu'il existe $C > 0$ telle que*

$$\mathbb{E}\|\nabla_x f(x,\cdot)\|^2 = \int_X \|\nabla_x f(x,w)\|^2 \, d\mu(w) \leqslant C + C\|x - x^*\|^2 \qquad \forall x \in \mathbb{R}^n. \tag{5.2}$$

Si la suite des pas t_k vérifie

$$\sum_{k=0}^{+\infty} t_k = +\infty \qquad \text{et} \qquad \sum_{k=0}^{+\infty} t_k^2 < +\infty$$

(par exemple, $t_k = \frac{1}{k+1}$) alors la suite $(x_k)_{k\in\mathbf{N}}$ converge presque sûrement vers x^*.

Dans le cas échantillonné, l'hypothèse (5.2) s'écrit

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla f_i(x)\|^2 \leqslant C + C\|x - x^*\|^2 \qquad \forall x \in \mathbb{R}^n$$

avec C indépendant de N. Dans la littérature, il est souvent supposé qu'il existe $C > 0$ tel que

$$\|\nabla_x f(x,w)\| \leqslant C \qquad \forall x \in \mathbb{R}^n \qquad \forall w \in X$$

qui est une hypothèse beaucoup plus forte que (5.2).

Démonstration. Nous allons d'abord démontrer que $(x_k)_{k\in\mathbf{N}}$ converge en moyenne quadratique vers x^*, i.e., $\mathbb{E}\|x_k - x^*\|^2 \to 0$ lorsque $k \to +\infty$.

On a $x_{k+1} - x^* = x_k - x^* - t_k \nabla_x f(x_k, w_{k+1})$, donc

$$\|x_{k+1} - x^*\|^2 = \|x_k - x^*\|^2 - 2t_k \langle x_k - x^*, \nabla_x f(x_k, w_{k+1})\rangle + t_k^2 \|\nabla_x f(x_k, w_{k+1})\|^2.$$

Prenons l'espérance de cette égalité par rapport à w_{k+1} : autrement dit, on applique l'intégration $\int_X (\cdot)\, d\mu(w_{k+1})$ à cette égalité. Par hypothèse d'indépendance, x_k ne dépend pas de w_{k+1}, donc

$$\mathbb{E}\|x_{k+1} - x^*\|^2 = \|x_k - x^*\|^2 - 2t_k \langle x_k - x^*, \underbrace{\mathbb{E}\nabla_x f(x_k,\cdot)}_{\nabla F(x_k)}\rangle + t_k^2 \, \mathbb{E}\|\nabla_x f(x_k,\cdot)\|^2$$

d'une part, comme F est α-convexe, on a $\langle \nabla F(x_k) - \nabla F(x^*), x_k - x^*\rangle \geqslant \alpha\|x_k - x^*\|^2$ avec $\nabla F(x^*) = 0$ puisque x^* minimise F. D'autre part, on utilise l'hypothèse (5.2), et on en déduit que

$$\mathbb{E}\|x_{k+1} - x^*\|^2 \leqslant (1 - 2\alpha t_k + Ct_k^2)\|x_k - x^*\|^2 + Ct_k^2. \tag{5.3}$$

En prenant maintenant l'espérance de cette inégalité par rapport à w_k (autrement dit, en appliquant l'intégration $\int_X (\cdot)\, d\mu(w_k)$ à cette inégalité), et en posant

$$a_k = \mathbb{E}\|x_k - x^*\|^2 \qquad \text{et} \qquad \rho_k = 1 - 2\alpha t_k + C t_k^2$$

on obtient

$$a_{k+1} \leqslant \rho_k a_k + C t_k^2.$$

On choisit alors le pas $0 < t_k < \frac{2\alpha}{C}$ de sorte que $0 < \rho_k < 1$. Par récurrence, on obtient

$$a_{k+1} \leqslant \prod_{j=0}^{k} \rho_j \|x_0 - x^*\|^2 + C \prod_{j=0}^{k} \rho_j \sum_{i=0}^{k} \frac{t_i^2}{\prod_{j=0}^{i} \rho_j}$$

Pour assurer la convergence de a_k vers 0, il faut alors assurer que

$$P_k = \prod_{j=0}^{k} \rho_j \to 0 \qquad \text{et} \qquad P_k \sum_{i=0}^{k} \frac{t_i^2}{P_i} \to 0$$

lorsque $k \to +\infty$, ce qui est vérifié sous les hypothèses sur la suite de pas. En effet, on a

$$\ln P_k = \sum_{j=0}^{k} \ln(1 - 2\alpha t_j + C t_j^2)$$

Comme $t_j \to 0$ lorsque $j \to +\infty$, on a $\ln(1 - 2\alpha t_j + C t_j^2) = -2\alpha t_j + \mathrm{O}(t_j^2)$ lorsque $j \to +\infty$, et en sommant, en utilisant le fait que $t_j > 0$, que la série des t_j diverge et que la série des t_j^2 converge, on obtient $\ln P_k = -2\alpha \sum_{j=0}^{k} t_j + \mathrm{O}(1)$ lorsque $k \to +\infty$ et donc il existe des constantes $C_1 > 0$ et $C_2 > 0$ telles que

$$C_1 \exp\Big(- 2\alpha \sum_{j=0}^{k} t_j \Big) \leqslant P_k \leqslant C_2 \exp\Big(- 2\alpha \sum_{j=0}^{k} t_j \Big) \qquad \forall k \in \mathbb{N}$$

ce qui montre que $P_k \to 0$ lorsque $k \to +\infty$. De plus,

$$P_k \sum_{i=0}^{k} \frac{t_i^2}{P_i} \leqslant \frac{C_2}{C_1} \sum_{i=0}^{k} t_i^2 \exp\Big(- 2\alpha \sum_{j=i+1}^{k} t_j \Big)$$

Pour simplifier, montrons la convergence vers 0 du membre de droite dans le cas où $t_i = \frac{1}{i+1}$ (ce n'est déjà pas si facile !). On sait que

$$\sum_{j=1}^{k} \frac{1}{j} = \ln k + \gamma + \mathrm{o}(1)$$

(résultat classique, $\gamma > 0$ est la constante d'Euler), donc

$$\exp\Big(- 2\alpha \sum_{j=i+1}^{k} t_j \Big) \sim \mathrm{Cst} \exp\Big(- 2\alpha \ln \frac{k}{i} \Big) = \mathrm{Cst} \frac{i^{2\alpha}}{k^{2\alpha}}$$

et donc

$$\sum_{i=0}^{k} t_i^2 \exp\Big(- 2\alpha \sum_{j=i+1}^{k} t_j \Big) \sim \frac{\mathrm{Cst}}{k^{2\alpha}} \sum_{i=1}^{k} i^{2\alpha-2} \sim \begin{cases} \frac{\mathrm{Cst}}{k^{2\alpha}} & \text{si} \quad \alpha < \frac{1}{2} \\ \mathrm{Cst}\, \frac{\ln k}{k} & \text{si} \quad \alpha = \frac{1}{2} \\ \frac{\mathrm{Cst}}{k} & \text{si} \quad \alpha > \frac{1}{2} \end{cases}$$

car, par comparaison entre série et intégrale, on a

$$\sum_{i=1}^{k} i^{2\alpha-2} \sim \begin{cases} \text{Cst} & \text{si} \quad \alpha < \frac{1}{2} \\ \text{Cst } \ln k & \text{si} \quad \alpha = \frac{1}{2} \\ \text{Cst } k^{2\alpha-1} & \text{si} \quad \alpha > \frac{1}{2} \end{cases}$$

Dans tous les cas, le terme converge vers 0, ce qui est la conclusion désirée.

A ce stade, on a donc montré que $(x_k)_{k\in\mathbb{N}}$ converge en moyenne quadratique vers x^*, i.e., $\mathbb{E}\|x_k - x^*\|^2 \to 0$ lorsque $k \to +\infty$. Pour montrer que $x_k \to x^*$ presque sûrement, on utilise un résultat général sur les martingales en théorie des probabilités. On pose

$$Z_k = \|x_k - x^*\|^2 + C\sum_{i=k}^{+\infty} t_i^2$$

On déduit de l'inégalité (5.3), en utilisant le fait que $1 - 2\alpha t_k + Ct_k^2 \leqslant 1$, que

$$\mathbb{E}Z_{k+1} \leqslant \|x_k - x^*\|^2 + Ct_k^2 + C\sum_{i=k+1}^{+\infty} t_i^2 = Z_k$$

donc $(Z_k)_{k\in\mathbb{N}}$ est une sur-martingale. Comme elle est minorée (car positive), elle converge presque sûrement, ce qui achève la démontration du théorème.

On rappelle ici qu'une *martingale* est une suite $(Z_k)_{k\in\mathbb{N}}$ de variables aléatoires qui vérifie $\mathbb{E}(Z_{k+1} \mid Z_0, \ldots, Z_k) = Z_k$. On parle de *sur-martingale* si l'égalité est remplacée par $\leqslant$, et de *sous-martingale* si l'égalité est remplacée par $\geqslant$. Il est connu, en théorie des probabilités, que toute sous-martingale majorée converge presque sûrement ; toute sur-martingale minorée converge presque sûrement ; toute martingale majorée ou minorée converge presque sûrement. □

On peut noter toutefois que la convergence de l'algorithme de gradient stochastique est lente. Il faut donc trouver un compromis entre cette lenteur d'exécution et le nombre N de données. Lorsque N est grand, le coût d'une itération de gradient stochastique est N fois plus petit que le coût d'une itération de gradient classique (dans lequel on calculerait N gradients).

5.3 Apprentissage, deep learning, rétropropagation

Dans l'esprit de la procédure des moindres carrés, l'objectif de la théorie de l'apprentissage est la suivante : étant donné un nuage de N points $(x^i, y^i) \in \mathbb{R}^n \times \mathbb{R}^m$, $i = 1, \ldots, N$ (échantillons de mesures), on veut trouver une "meilleure" fonction f, dans une certaine classe $\mathcal{F}$ de fonctions, telle que $f(x^i) \simeq y^i$ pour tout $i \in \{1, \ldots, N\}$, en moyenne quadratique, autrement dit on veut résoudre le problème de minimisation

$$\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \|f(x^i) - y^i\|^2$$

ou de manière plus générale, on se donne une fonction coût (souvent appelée "*loss function*" en anglais) et on considère le problème

$$\min_{f\in\mathcal{F}} C(f(X), Y)$$

où (X, Y) représente les données mesurées.

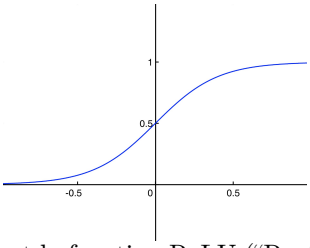
La classe de fonctions f considérées est la suivante (deep learning) : on cherche f sous la forme

$$f(X) = \sigma_k\, A_k \cdots \sigma_2\, A_2\, \sigma_1\, A_1\, X$$

où A_i est une matrice de taille $n_i \times n_{i-1}$ (autrement dit, $A_i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$ est une application linéaire), et $\sigma_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ est une application non linéaire, pour $i = 1, \ldots, k$. Ci-dessus, pour éviter la confusion de parenthèses, la notation $\sigma_i\, a$ signifie $\sigma_i(a)$.

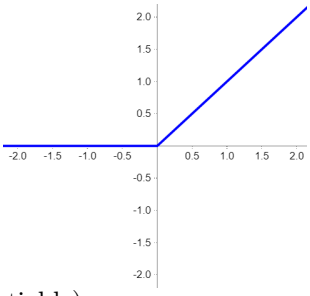
L'entier k est appelé le *nombre de couches* ("*layers*" en anglais), les coefficients des matrices A_i sont appelés *poids* (ils représentent généralement des interactions dans un réseau de neurones), et les applications non linéaires σ_i sont appelées *fonctions d'activation*. Généralement, pour $a = (a_1, \ldots, a_{n_i}) \in \mathbb{R}^{n_i}$, on prend $\sigma_i(a) = (\sigma(a_1), \ldots, \sigma(a_{n_i}))^\top$ où $\sigma : \mathbb{R} \to \mathbb{R}$ est une *fonction sigmoïde*, par exemple,

$$\sigma(s) = \frac{1}{1 + e^{-\lambda s}} \qquad \text{pour un } \lambda > 0$$



qui a l'avantage d'être dérivable partout. Un autre exemple est la fonction ReLU ("Rectified Linear Unit") :

$$\sigma(s) = \max(s, 0)$$



qui, elle, n'est pas dérivable en 0 (mais elle est sous-différentiable).

Le problème d'apprentissage est alors de trouver les meilleurs poids, donc, les meilleures matrices $A_1, \ldots, A_k$, de façon à minimiser la fonction coût

$$F(A_1, \ldots, A_k) = C(\sigma_k\, A_k \cdots \sigma_2\, A_2\, \sigma_1\, A_1\, X,\ Y).$$

Comme il s'agit d'un problème non convexe en grande dimension, on met en oeuvre une méthode simple : une méthode de gradient (ou de sous-gradient dans le cas sous-différentiable) est alors privilégiée. Lorsque C est une somme de carrés (comme écrit ci-dessus) avec N grand, on peut mettre en oeuvre une méthode de gradient stochastique (cf section 5.2).

L'objectif est alors de calculer le gradient de F. Parlons plutôt de différentielle, définie sur un produit d'espaces matriciels. Pour simplifier, ci-dessous on suppose σ_i différentiable (sinon, on remplace par un sous-différentiel dans le cas de ReLU). Il s'agit simplement d'une dérivation composée. Par la formule de dérivation composée (*chain rule*), en notant $C(a_k, Y)$ la fonction coût (et $a_k = \sigma_k\, A_k\, \sigma_{k-1}\, A_{k-1} \cdots \sigma_2\, A_2\, \sigma_1\, A_1\, X$), on a

$$\frac{\partial F}{\partial A_j}.H_j = \underbrace{\frac{\partial C}{\partial a_k}.d\sigma_k.A_k\, d\sigma_{k-1}.A_{k-1} \cdots d\sigma_{j+1}.A_{j+1}\, d\sigma_j}_{\Delta_j}.H_j\, \underbrace{\sigma_{j-1}\, A_{j-1} \cdots \sigma_1\, A_1\, X}_{a_{j-1}}$$

On note tout d'abord que

$$a_j = \sigma_j\, A_j\, a_{j-1}, \quad j = 1, 2, \ldots, k \tag{5.4}$$

autrement dit, à partir de $a_0 = X$, on calcule par indices croissants ("*forward phase*"), $a_1 = \sigma_1\, A_1\, a_0$, puis $a_2 = \sigma_2\, A_2\, a_1$, etc, jusqu'à a_k.

On note ensuite que

$$\Delta_k = \frac{\partial C}{\partial a_k}.d\sigma_k, \qquad \Delta_{j-1} = \Delta_j\, A_j\, d\sigma_{j-1}, \quad j = k, k-1, \ldots, 2 \tag{5.5}$$

autrement dit, à partir de Δ_k, on calcule par indices décroissants ("*backward phase*"), $\Delta_{k-1} = \Delta_k\, A_k\, d\sigma_{k-1}$, puis $\Delta_{k-2} = \Delta_{k-1}\, A_{k-1}\, d\sigma_{k-2}$, etc, jusqu'à Δ_1.

Ainsi, pour calculer la différentielle de F, on commence d'abord par la phase *forward*, i.e., le calcul (5.4), puis on implémente la phase *backward*, i.e., le calcul (5.5).

La phase *backward* s'appelle *rétropropagation des gradients* ("*gradient backpropagation*" en anglais) pour la raison suivante. La fonction C est à valeur réelles, donc la différentielle $\frac{\partial C}{\partial a_k}$ est identifiée à une matrice ligne. Sa transposée est le gradient :

$$\frac{\partial C}{\partial a_k}^\top = \nabla_{a_k} C$$

On peut écrire (5.5) sous la forme transposée :

$$\Delta_k^\top = (d\sigma_k)^\top\, \nabla_{a_k} C, \qquad \Delta_{j-1}^\top = (d\sigma_{j-1})^\top\, A_j^\top\, \Delta_j^\top, \quad j = 2, \ldots, k$$

On parle alors de *rétropropagation* du gradient $\nabla_{a_k} C$ car, en partant de ce gradient, on calcule successivement

$$\begin{aligned}
\Delta_k^\top &= (d\sigma_k)^\top\, \nabla_{a_k} C \\
\Delta_{k-1}^\top &= (d\sigma_{k-1})^\top\, A_k^\top\, (d\sigma_k)^\top\, \nabla_{a_k} C \\
&\vdots \\
\Delta_1^\top &= (d\sigma_1)^\top\, A_2^\top\, d\sigma_2)^\top \cdots (d\sigma_{k-1})^\top\, A_k^\top\, (d\sigma_k)^\top\, \nabla_{a_k} C
\end{aligned}$$

ce qui revient à propager en arrière le gradient. Cela est très relié aux techniques utilisées en différentiation automatique évoquées en section 5.1.

Bien entendu, les théories de machine learning, deep learning comportent de nombreuses considérations. Elles sont actuellement en pleine évolution.

Utilisation de `TensorFlow`. Un peu comme `AMPL` est une boîte noire (extrêmement efficace) pour coder facilement et rapidement des problèmes d'optimisation, même très difficiles, `TensorFlow` est un outil libre disponible sur le web pour créer des modèles d'Intelligence Artificielle (IA) et de les entraîner (Machine Learning, Deep Learning). Il faut coder en `Python`.

Pour installer `TensorFlow` : https://www.tensorflow.org/install/

On trouve sur le web quantité de bons tutoriels pour apprendre à utiliser `TensorFlow`.

Il existe beaucoup d'outils pour coder des procédures d'apprentissage. Le *deep learning*, l'IA, sont des applications de l'optimisation qui, bien qu'étant assez anciennes, ont actuellement un très fort potentiel applicatif, car la puissance computationnelle des ordinateurs permet désormais d'aborder des problèmes que l'humain a du mal à appréhender. Mais ce n'est jamais que de l'optimisation, et l'ordinateur ne fait pas autre chose que de calculer ce qu'on lui a demandé de calculer !