



DÉPARTEMENT GÉNIE MATHÉMATIQUE ET
MODÉLISATION

4ÈME ANNÉE, 2017-2018.

MÉTHODES NUMÉRIQUES POUR L'OPTIMISATION
NON LINÉAIRE DÉTERMINISTE.

Aude RONDEPIERRE

Table des matières

I	Outils théoriques, concepts algorithmiques	7
1	Eléments d'analyse convexe	9
1.1	Définitions et propriétés géométriques élémentaires	9
1.1.1	Ensembles et fonctions convexes	9
1.1.2	Semi-continuité inférieure	13
1.1.3	Opérations sur les fonctions convexes	14
1.2	Continuité et différentiabilité des fonctions convexes	14
1.3	Eléments d'analyse pour l'algorithmie	16
1.3.1	Fonctions à gradient Lipschitz	17
1.3.2	Fonctions fortement convexes	18
1.3.3	Conditionnement d'une fonction	20
2	Introduction à l'optimisation numérique	21
2.1	Qu'est-ce qu'un problème d'optimisation?	21
2.2	Résultats d'existence et d'unicité en optimisation	24
2.2.1	Cas où l'ensemble X des contraintes est borné	24
2.2.2	Cas où l'ensemble X des contraintes est non borné	24
2.2.3	Cas particulier de contraintes d'égalités et/ou d'inégalités	26
2.2.4	Convexité et optimisation	26
II	Algorithmes classiques en optimisation différentiable	27
3	Conditions d'optimalité	29
3.1	Cas général : condition d'optimalité géométrique	29
3.1.1	Le cône tangent ou comment se déplacer dans le domaine des contraintes	29
3.1.2	Condition nécessaire d'optimalité géométrique	30
3.1.3	Cas où l'ensemble des contraintes est convexe	31
3.2	Cas différentiable sans contraintes	32
3.3	Cas différentiable avec contraintes fonctionnelles	33
3.3.1	Problèmes avec contraintes d'égalité	33
3.3.2	Problèmes avec contraintes d'inégalités	38
3.3.3	Problèmes avec contraintes d'égalité et d'inégalité	41

4	Méthodes de descente en optimisation différentiable sans contrainte	43
4.1	Principe général des méthodes de descente	43
4.2	La recherche linéaire	46
4.2.1	Recherches linéaires élémentaires	46
4.2.2	Condition de Armijo ou comment éviter les pas trop grands	47
4.2.3	Conditions de Wolfe ou comment éviter les pas trop petits	49
4.2.4	Convergence globale des algorithmes avec pas de Wolfe	52
4.3	Algorithmes de type gradient	53
4.3.1	CS pour qu'un algorithme de gradient soit une méthode de descente	53
4.3.2	Résultats de convergence globale	54
4.3.3	Tests numériques : pas fixe/pas optimal vs Pas de Wolfe	54
4.4	Méthodes de type Newton	57
4.4.1	Méthode de Newton avec recherche linéaire	58
4.4.2	Méthodes de quasi-Newton	60
4.4.3	Propriétés et convergence de l'algorithme BFGS	63
4.5	Algorithmes pour la régression non-linéaire	64
4.5.1	Problèmes de moindres carrés linéaires (rappels)	64
4.5.2	Algorithme de Gauss-Newton	65
4.5.3	Algorithme de Levenberg-Marquardt	66
5	Algorithmes pour l'optimisation différentiable sous contrainte	69
5.1	Introduction à la dualité lagrangienne	69
5.1.1	Problème primal - Problème dual	70
5.1.2	Liens entre problème primal et problème dual	71
5.1.3	Cas convexe - Multiplicateurs de Kuhn-Tucker	72
5.2	Méthode du gradient projeté	73
5.3	Algorithmes newtoniens - Méthode SQP	75
5.4	Méthodes de pénalisation	78
5.4.1	Principe général	79
5.4.2	Le Lagrangien augmenté	80
III	Introduction à l'optimisation convexe non différentiable	83
6	Compléments d'analyse convexe	85
6.1	Sous-différentiel d'une fonction convexe	85
6.1.1	Définition et propriétés	85
6.1.2	Règles de calcul sous-différentiel	87
6.1.3	Exemple 1 : Sous-différentiel d'une fonction indicatrice	88
6.1.4	Exemple 2 : Sous-différentiel d'une norme	88
6.2	Condition d'optimalité	89

7	Algorithmes pour l'optimisation non différentiable	91
7.1	Un rapide tour d'horizon en optimisation non lisse	91
7.2	Méthodes de sous-gradients	93
7.2.1	Cas sans contrainte	94
7.2.2	Méthodes de sous-gradient projeté	95
7.2.3	Application aux problèmes duaux	97
7.3	Opérateur proximal	98
7.3.1	Définition et propriétés	99
7.3.2	Algorithme du point proximal	101
7.4	Algorithme Forward-Backward : un algorithme de splitting	102

L'optimisation et particulièrement l'optimisation numérique a connu un essor important ces dernières années avec l'avènement de l'ordinateur. Elle est souvent l'ultime étape de l'analyse numérique où, après avoir étudié un phénomène physique, l'avoir mis en équation, avoir étudié ces équations et avoir montré que l'on pouvait calculer les solutions avec un ordinateur, on commence à optimiser le système en changeant certains paramètres pour changer la solution dans un sens désiré.

Ce cours est un cours d'optimisation numérique : on cherche à construire des algorithmes adaptés et efficaces pour la résolution de problèmes d'optimisation de la forme :

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

où la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est supposée différentiable dans la deuxième partie de ces notes, localement Lipschitzienne et convexe mais non différentiable dans la troisième partie. Ces hypothèses requièrent un outil spécial : le sous-différentiel de l'analyse convexe, noté $\partial f(x)$.

Les fonctions convexes apparaissent abondamment dans l'ingénierie et permettent de modéliser de nombreux phénomènes non linéaires (équations de la physique, traitement du signal, théorie des jeux et de l'économie, statistiques...). Elles ont des propriétés remarquables qui permettent d'analyser plus facilement leurs propriétés et aussi de les minimiser efficacement. De nombreux problèmes non convexes irrésolvables peuvent être approchés par des problèmes convexes qui eux sont presque systématiquement minimisés en des temps rapides.

Pour conclure, nous indiquons quelques références utiles pour aller plus loin. Le livre [5] est une excellente référence en optimisation numérique différentiable, à la fois pédagogique et bien documentée. Plusieurs exemples et points de vue présentés dans ces notes sont inspirés de cet ouvrage. Les livres [12, 27] sont de très bonnes références pour découvrir les nombreux détails de l'analyse convexe en dimension finie. La référence [2] propose est une très bonne introduction à l'analyse convexe dans des espaces de Hilbert. Enfin, indiquons une référence dans les espaces de Banach [30]. Cette référence apporte des informations intéressantes même pour des questions d'analyse numérique en dimension finie. Au niveau algorithmique, ce cours est essentiellement inspiré des références [24].

Remerciements à Denis Arzelier et Pierre Weiss pour leurs notes de cours qui m'ont bien aidée à préparer ce polycopié !.

Première partie

**Outils théoriques, concepts
algorithmiques**

Chapitre 1

Eléments d'analyse convexe

Dans ce chapitre, nous présentons quelques propriétés remarquables des fonctions convexes. Elle permettront de construire des algorithmes de minimisation dans la suite du cours.

Dans toutes ces notes, on se place sur l'espace vectoriel \mathbb{R}^n , $n \in \mathbb{N}$. On le munit d'un produit scalaire $\langle \cdot, \cdot \rangle$. La norme associée au produit scalaire est notée $\| \cdot \|_2$. Elle est définie pour tout $x \in \mathbb{R}^n$ par $\|x\|_2 = \sqrt{\langle x, x \rangle}$.

1.1 Définitions et propriétés géométriques élémentaires

Dans ce chapitre, on considère des fonctions

$$f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}.$$

Autoriser les fonctions à valoir $+\infty$ présente l'intérêt de pouvoir représenter les problèmes contraints sous une forme non contrainte. On a en effet :

$$\inf_{x \in \mathbb{R}^n} f(x) = \inf_{x \in \text{dom}(f)} f(x)$$

où $\text{dom}(f)$ est défini de la façon suivante :

Définition 1.1 (Domaine d'une fonction) *Le domaine d'une fonction f est noté $\text{dom}(f)$. Il est défini par*

$$\text{dom}(f) = \{x \in \mathbb{R}^n, f(x) < +\infty\}.$$

Dans toute la suite de ce cours, on supposera (sans le préciser) que nos fonctions n'ont pas un domaine vide : $\text{dom}(f) \neq \emptyset$.

1.1.1 Ensembles et fonctions convexes

Définition 1.2 (Ensemble convexe) *Soit $X \subseteq \mathbb{R}^n$. L'ensemble X est dit convexe si :*

$$\forall (x_1, x_2) \in X \times X, \forall \alpha \in [0, 1], \alpha x_1 + (1 - \alpha)x_2 \in X.$$

Voici quelques ensembles convexes remarquables :

Définition 1.3 (Combinaison convexe) Soient x_1, \dots, x_m , m éléments de \mathbb{R}^n . On dit que x est combinaison convexe de ces points s'il existe des réels $(\alpha_1, \dots, \alpha_m)$ tels que :

- $\forall i \in \{1, \dots, m\}, \alpha_i \geq 0$,
- $\sum_{i=1}^m \alpha_i = 1$,
- $x = \sum_{i=1}^m \alpha_i x_i$.

Définition 1.4 (Enveloppe convexe) Soit $X \subseteq \mathbb{R}^n$ un ensemble. On appelle enveloppe convexe de X et on note $\text{conv}(X)$ l'ensemble convexe le plus petit contenant X . En dimension finie, c'est aussi l'ensemble des combinaisons convexes d'éléments de X :

$$\text{conv}(X) = \left\{ x \in \mathbb{R}^n \mid x = \sum_{i=1}^p \alpha_i x_i \text{ où } x_i \in X, p \in \mathbb{N} \text{ et } \sum_{i=1}^p \alpha_i = 1, \alpha_i \geq 0 \right\}.$$

Quant aux fonctions à valeurs dans $\mathbb{R} \cup \{+\infty\}$, la définition de la convexité reste identique à celle vue en 3MIC :

Définition 1.5 (Fonction convexe) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$.

- f est dite convexe si :

$$\forall (x, y) \in \text{dom}(f)^2, \forall \lambda \in [0, 1], f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y).$$

- f est dite strictement convexe si :

$$\forall (x, y) \in \text{dom}(f)^2, x \neq y, \forall \lambda \in [0, 1], f((1 - \lambda)x + \lambda y) < (1 - \lambda)f(x) + \lambda f(y).$$

Cette définition est illustrée sur la Figure 1.1. Elle implique que le domaine d'une fonction convexe est convexe (exercice). Notons qu'une fonction convexe peut ne pas être dérivable. On peut cependant montrer qu'elle est dérivable presque partout.

La définition de la convexité permet d'obtenir un lemme souvent utile (notamment en probabilités) :

Lemme 1.1 (Inégalité de Jensen) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe et x_1, x_2, \dots, x_m m points appartenant à $\text{dom}(f)$ et $\alpha_1, \dots, \alpha_m$ des coefficients réels positifs tels que $\sum_{i=1}^m \alpha_i = 1$. Alors

$$f\left(\sum_{i=1}^m \alpha_i x_i\right) \leq \sum_{i=1}^m \alpha_i f(x_i)$$

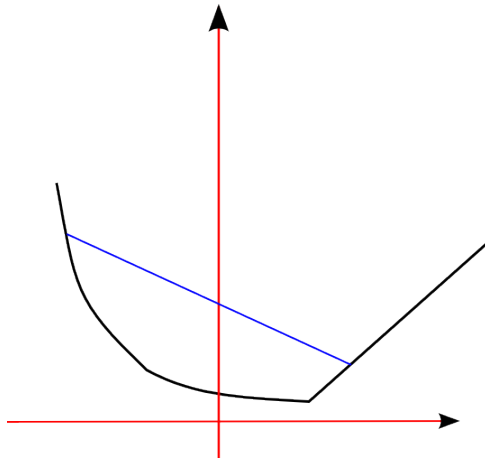


FIGURE 1.1 – Un exemple de fonction convexe

Démonstration. Par récurrence, on vérifie d'abord que pour $m = 2$, l'inégalité n'est rien d'autre que la définition de la convexité. Puis on suppose le résultat vrai au rang k et on le montre au rang $k + 1$ en réutilisant l'inégalité de convexité. \square

Corollaire 1.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe et $x \in \mathbb{R}^n$ une combinaison convexe de x_1, \dots, x_m alors :

$$f(x) \leq \max_{i \in \{1, \dots, m\}} f(x_i).$$

Démonstration. En exercice. \square

Théorème 1.1 f est convexe si et seulement si son épigraphe

$$\text{epi}(f) = \{(x, t) \in \text{dom}(f) \times \mathbb{R}, t \geq f(x)\}$$

est convexe.

Démonstration. Faite en cours. \square

Théorème 1.2 Si f est convexe alors ses ensembles de niveaux

$$\mathcal{L}_f(\beta) = \{x \in \text{dom}(f), f(x) \leq \beta\}$$

sont soit vides, soit convexes.

Démonstration. En exercice. \square

Remarque 1.1 La réciproque est fautive ! Une fonction dont les ensembles de niveaux sont convexes n'est pas convexe en général (exemple en 1D : $f(x) = \sqrt{|x|}$). Une telle fonction

est appelée *quasi-convexe*. Un exemple de fonction quasi-convexe (non convexe) est donné sur la Figure 1.2.

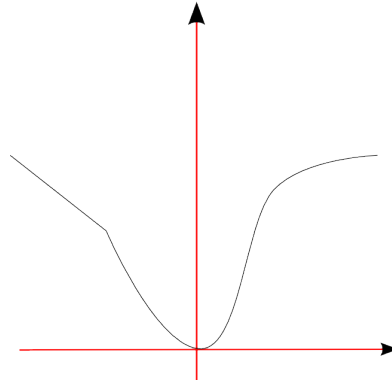


FIGURE 1.2 – Un exemple de fonction quasi-convexe. Ses lignes de niveaux sont des segments (donc des ensembles convexes), mais la fonction n'est pas convexe.

Caractérisation différentielle de la convexité

Théorème 1.3 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Les propositions suivantes sont équivalentes :

- (a) f est convexe.
- (b) Ses hyperplans tangents sont des minorants (voir Figure 1.3).

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle.$$

- (c) Le gradient de f est un opérateur monotone :

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \langle \nabla f(y) - \nabla f(x), y - x \rangle \geq 0.$$

Remarque 1.2 Un opérateur $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est dit monotone si

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \langle F(y) - F(x), y - x \rangle \geq 0.$$

Cette notion généralise la notion de gradient des fonctions convexes. Elle apparaît abondamment dans les EDP.

Les fonctions C^2 convexes peuvent être caractérisées par leur hessienne.

Proposition 1.1 (Caractérisation de second ordre de la convexité)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction de classe C^2 . f est convexe ssi :

$$\forall x \in \mathbb{R}^n, H[f](x) \succeq 0$$

(i.e. la hessienne de f est semi-définie positive en tous points).

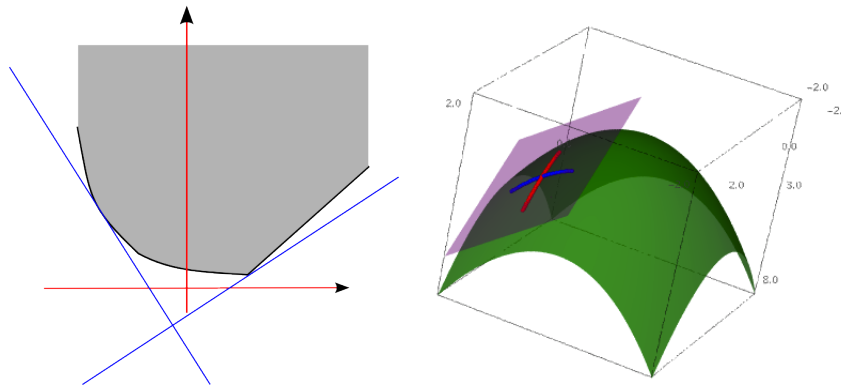


FIGURE 1.3 – Hyperplans tangents d'une fonction convexe (1D) et d'une fonction concave (2D). Notez que l'hyperplan tangent est un minorant pour la fonction convexe et un majorant pour la fonction concave.

1.1.2 Semi-continuité inférieure

De façon générale, une fonction convexe peut être très compliquée sur le bord de son domaine. Par exemple, la fonction 2D

$$f(x, y) = \begin{cases} 0 & \text{si } x^2 + y^2 < 1 \\ \phi(x, y) & \text{si } x^2 + y^2 = 1 \text{ avec } \phi(x, y) \geq 0 \end{cases} \quad (1.1)$$

est convexe. Cependant, son comportement sur le bord peut être arbitrairement complexe. Minimiser de telles fonctions est en général impossible. Cette remarque nous mène à nous restreindre à la classe de fonctions semi-continues inférieurement.

Définition 1.6 (Fonction semi-continue inférieurement) Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ est dite fermée ou semi-continue inférieurement (s.c.i.) si :

$$\forall x \in \mathbb{R}^n, \forall x_n \xrightarrow[n \rightarrow +\infty]{} x, \liminf f(x_n) \geq f(x).$$

Proposition 1.2 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$. Les propositions suivantes sont équivalentes :

- i. f est s.c.i.
- ii. Quel que soit $\alpha \in \mathbb{R}$, l'ensemble de niveau $\{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}$ est fermé.
- iii. $\text{epi}(f)$ est un ensemble fermé.

Attention! Toute fonction continue sur son domaine n'est pas nécessairement s.c.i.! Une fonction continue f est s.c.i. à condition que son domaine $\text{dom}(f)$ soit fermé.

Exercice 1.1.1 A quelle condition la fonction f définie ci-après est-elle fermée ?

$$f(x) = \begin{cases} 0 & \text{si } x \in [0, 1[\\ a \geq 0 & \text{si } x = 1 \end{cases}$$

Exemple 1.1.1 Donnons quelques exemples de fonctions convexes fermées.

1. Les fonctions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ convexes telles que $\text{dom}(f) = \mathbb{R}^n$ sont fermées.
2. Les fonctions linéaires sont convexes et fermées.
3. Les fonctions convexes, différentiables sur \mathbb{R}^n sont convexes et fermées.
4. La fonction $f(x) = \|x\|$ où $\|\cdot\|$ est une norme quelconque est convexe et fermée.
5. La fonction (1.1) n'est convexe et fermée que si $\phi(x, y) = 0, \forall (x, y) \in \mathbb{R}^2$.

Exercice 1.1.2 Montrer que la fonction plafond (qui, évaluée en x , retourne le plus petit entier supérieur ou égal à x) est s.c.i. alors que la fonction partie entière E ne l'est pas.

1.1.3 Opérations sur les fonctions convexes

Théorème 1.4 Soient f_1 et f_2 deux fonctions convexes s.c.i. et $\beta > 0$. Les fonctions suivantes sont convexes s.c.i. :

1. $f(x) = \beta f_1(x)$.
2. $f(x) = (f_1 + f_2)(x)$ et $\text{dom}(f) = \text{dom}(f_1) \cap \text{dom}(f_2)$.
3. $f(x) = \max\{f_1(x), f_2(x)\}$ et $\text{dom}(f) = \text{dom}(f_1) \cap \text{dom}(f_2)$.

Démonstration. En exercice (un peu difficile...). □

Théorème 1.5 Soit $\phi(y), y \in \mathbb{R}^m$ une fonction convexe s.c.i. Soit $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$ un opérateur linéaire et $b \in \mathbb{R}^m$. Alors la fonction $f(x) = \phi(Ax + b)$ est convexe s.c.i. et $\text{dom}(f) = \{x \in \mathbb{R}^n, Ax + b \in \text{dom}(\phi)\}$.

Démonstration. Faite en cours. □

1.2 Continuité et différentiabilité des fonctions convexes

Le premier résultat de ce paragraphe est que toute fonction convexe est continue sur l'intérieur de son domaine.

Théorème 1.6 Soit f une fonction convexe et $x_0 \in \text{int}(\text{dom}(f))$. Alors f est continue et localement Lipschitz en x_0 .

Une conséquence importante de ce théorème est que toute fonction convexe dont le domaine est \mathbb{R}^n tout entier est continue sur \mathbb{R}^n . Pour démontrer le Théorème 1.6, nous avons besoin du résultat intermédiaire suivant :

Lemme 1.2 Soit f une fonction convexe et $x_0 \in \text{int}(\text{dom}(f))$. Alors f est majorée localement en x_0 ¹.

1. Note : elle peut exploser au voisinage du bord. Il suffit par exemple de considérer la fonction $x \mapsto 1/x$ sur $]0, 1]$ pour s'en convaincre.

Démonstration. Soit $\epsilon > 0$ tel que les points $x_0 \pm \epsilon e_i \in \text{int}(\text{dom}(f))$. D'après le corollaire 1.1 on a $\forall x \in \text{conv}(\{x_0 \pm \epsilon e_i\}) = B_\infty(x_0, \epsilon)$, $f(x) \leq \max_{1 \leq i \leq n} f(x_0 \pm \epsilon e_i) < +\infty$. \square

Ce lemme implique la continuité d'une fonction convexe sur l'intérieur de son domaine.

Démonstration du théorème 1.6. D'après le lemme 1.2, il existe $M < +\infty$ et $\epsilon > 0$ tels que $f(x) \leq M, \forall x \in B_2(x_0, \epsilon)$. Soit $y \in B_2(x_0, \epsilon)$ et $z \in \partial B_2(x_0, \epsilon)$ un point de la frontière tel que $z = x_0 + \frac{1}{\alpha}(y - x_0)$ avec $\alpha = \frac{1}{\epsilon} \|y - x_0\|_2$. Par construction $\alpha \leq 1$ et $y = \alpha z + (1 - \alpha)x_0$. Par convexité de f on a donc :

$$\begin{aligned} f(y) &\leq (1 - \alpha)f(x_0) + \alpha f(z) \\ &\leq f(x_0) + \alpha(M - f(x_0)) \\ &= f(x_0) + \frac{M - f(x_0)}{\epsilon} \|y - x_0\|_2. \end{aligned}$$

Pour obtenir l'inégalité inverse, on considère un point $y \in B_2(x_0, \epsilon)$ et on pose $u = x_0 + \frac{1}{\alpha}(x_0 - y)$. On a $\|u - x_0\|_2 = \epsilon$ et $y = x_0 + \alpha(x_0 - u)$. D'après le théorème ?? on a donc :

$$\begin{aligned} f(y) &\geq f(x_0) + \alpha(f(x_0) - f(u)) \\ &\geq f(x_0) - \alpha(M - f(x_0)) \\ &= f(x_0) - \frac{M - f(x_0)}{\epsilon} \|y - x_0\|_2. \end{aligned}$$

On a donc $\forall y \in B_2(x_0, \epsilon)$:

$$|f(y) - f(x_0)| \leq \frac{M - f(x_0)}{\epsilon} \|y - x_0\|_2.$$

\square

Intéressons nous maintenant à la différentiabilité des fonctions convexes. On rappelle pour cela la définition de la dérivée directionnelle :

Définition 1.7 (Dérivée directionnelle) Soit $x \in \text{dom}(f)$. On appelle dérivée directionnelle au point x dans la direction p la limite suivante :

$$f'(x, p) = \lim_{\alpha \rightarrow 0^+} \frac{1}{\alpha} (f(x + \alpha p) - f(x)).$$

Si cette limite existe, on dit que $f'(x, p)$ est la dérivée directionnelle de f en x .

Lemme 1.3 Soit f une fonction convexe et $x \in \text{int}(\text{dom}(f))$. Alors $p \mapsto f'(x, p)$ est une fonction convexe, homogène de degré 1. Pour tout $y \in \text{dom}(f)$ on a

$$f(y) \geq f(x) + f'(x, y - x). \quad (1.2)$$

Démonstration. En exercice. \square

Lorsqu'elle existe, la dérivée directionnelle donne des informations sur la pente de la fonction dans la direction d , tout comme la dérivée donne des informations sur la pente des fonctions à une variable :

- si $f'(x; d) > 0$ alors f est croissante dans la direction d .
- si $f'(x; d) = 0$ alors on ne peut pas conclure.
- si $f'(x; d) < 0$ alors f est décroissante dans la direction d : dans ce cas, on dit que d est une direction de descente de f en x .

Dans le cas où f est différentiable, on a :

Proposition 1.3 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable et $x \in \mathbb{R}^n$. Pour tout $d \in \mathbb{R}^n$,

$$f'(x; d) = \langle \nabla f(x), d \rangle.$$

Rappelons que la dérivabilité selon toute direction en x n'implique pas nécessairement la différentiabilité de f en x . Le contre-exemple typique est la fonction $x \rightarrow |x|$. Cette fonction est bien dérivable à droite et à gauche de 0, mais elle n'est pas dérivable en 0.

Théorème 1.7 Une fonction convexe est différentiable dans toutes les directions sur tous les points de l'intérieur de son domaine.

Démonstration. Soit $x \in \text{int}(\text{dom}(f))$. On considère la fonction

$$\phi(\alpha) = \frac{1}{\alpha}(f(x + \alpha p) - f(x)), \alpha > 0.$$

Soit $\beta \in]0, 1]$ et $\alpha \in]0, \epsilon]$ avec ϵ suffisamment petit pour que $x + \epsilon p \in \text{dom}(f)$. Alors :

$$f(x + \alpha\beta p) = f((1 - \beta)x + \beta(x + \alpha p)) \leq (1 - \beta)f(x) + \beta f(x + \alpha p).$$

Ainsi

$$\phi(\alpha\beta) = \frac{1}{\alpha\beta}(f(x + \alpha\beta p) - f(x_0)) \leq \frac{1}{\alpha}(f(x + \alpha p) - f(x)) = \phi(\alpha).$$

La fonction ϕ est donc décroissante au voisinage de 0^+ . Pour $\gamma > 0$ tel que $x - \gamma p \in \text{dom}(f)$ on a par convexité :

$$\phi(\alpha) \geq \frac{1}{\gamma}(f(x) - f(x - \gamma p)).$$

La limite quand $\alpha \rightarrow 0^+$ existe donc. □

1.3 Eléments d'analyse pour l'algorithmie

On pourrait penser qu'une fonction différentiable est plus facile à minimiser qu'une fonction non-différentiable. De même, une fonction strictement convexe pourrait être plus facile à minimiser qu'une fonction simplement convexe (notamment parce qu'elle admet un minimiseur unique). Il n'en est rien. En effet, on peut approcher de façon aussi proche que l'on souhaite une fonction non différentiable par une fonction C^∞ et une fonction convexe

par une fonction strictement convexe. Il faut donc introduire des classes de fonctions plus régulières pour développer des algorithmes efficaces. Les deux classes présentées ci-après sont les plus naturelles : les fonctions différentiables à gradient Lipschitz et les fonctions fortement convexes.

1.3.1 Fonctions à gradient Lipschitz

De façon générale, les fonctions différentiables ne sont pas plus faciles à minimiser que les fonctions non différentiables. On peut en effet approcher n'importe quelle fonction, aussi précisément qu'on le souhaite en norme L^∞ , par une fonction C^∞ . La différentiabilité n'est donc pas une propriété assez forte pour permettre de faire des estimations d'erreurs sur les méthodes d'optimisation.

C'est pour cette raison qu'on introduit la classe des fonctions différentiables à gradient Lipschitz. Dans tout ce paragraphe, on considère des fonctions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiables à gradient Lipschitz :

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|. \quad (1.3)$$

Ces fonctions ont une propriété essentielle pour la preuve de convergence des méthodes de descente :

Lemme 1.4 *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable à gradient L -Lipschitz continu. Elle satisfait :*

$$f(y) \leq \underbrace{f(x) + \langle \nabla f(x), y - x \rangle}_{\text{Approximation linéaire}} + \frac{L}{2}\|y - x\|^2$$

Démonstration. Faite en cours. □

Ce résultat indique que pour les fonctions à gradient Lipschitz, l'estimation linéaire d'une fonction ne peut pas être trop mauvaise. Les fonctions à gradient Lipschitz peuvent être caractérisées par leur hessienne.

Lemme 1.5 *$f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction C^2 , convexe à gradient L -Lipschitz ssi :*

$$\begin{aligned} \forall x \in \mathbb{R}^n, \lambda_{\min}(H_f(x)) &\geq 0 && \text{(CONVEXITÉ)} \\ \forall x \in \mathbb{R}^n, \lambda_{\max}(H_f(x)) &\leq L && \text{(GRADIENT LIPSCHITZ)}. \end{aligned}$$

Terminons ce paragraphe par quelques exemples :

1. La fonction $f(x) = \frac{1}{2}\|Ax - b\|^2$ est convexe à gradient Lipschitz de constante $L = \lambda_{\max}(A^*A)$.
2. La fonction $f(x) = -\log(x)$ est convexe sur \mathbb{R}_+^* , mais son gradient n'est pas Lipschitz.

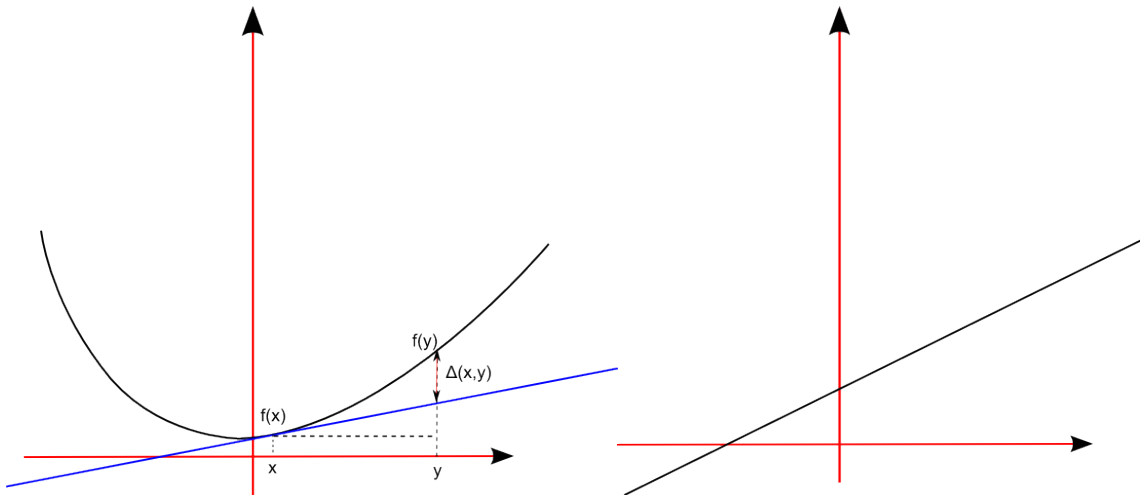


FIGURE 1.4 – Fonction à gradient Lipschitz : le graphe de la courbe ne peut pas s'éloigner trop rapidement de la tangente. Ici $\Delta(x, y) \leq \frac{L}{2} \|y - x\|_2^2$. La fonction affine de droite satisfait $L = 0$ puisque la tangente est identique à la fonction.

3. La fonction $f(x) = \exp(x)$ est convexe sur \mathbb{R} , mais son gradient n'est pas Lipschitz. Par contre sur tout intervalle borné du type $[a, b]$, on a $f''(x) \leq \exp(b)$.

1.3.2 Fonctions fortement convexes

L'hypothèse de fonction à gradient Lipschitz ne suffit pas toujours. En particulier, il est impossible de dire quelque chose sur la distance au minimiseur pour des algorithmes de premier ordre sous la simple hypothèse qu'on minimise une fonction à gradient Lipschitz. Une classe de fonctions sur laquelle on peut dire plus de choses est la classe des fonctions fortement convexes.

Définition 1.8 Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ est dite *fortement convexe* s'il existe $\mu > 0$ tel que $\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \forall \eta \in \partial f(x)$ on aie :

$$f(y) \geq f(x) + \langle \eta, y - x \rangle + \frac{\mu}{2} \|y - x\|^2. \quad (1.4)$$

Notez que le signe dans l'inégalité (1.4) est juste inversé par rapport aux fonctions différentiables à gradient Lipschitz. La forte convexité indique donc que le graphe de la courbe s'éloigne suffisamment rapidement de la tangente.

Proposition 1.4 Une fonction fortement convexe est strictement convexe, elle admet donc un minimiseur unique.

En revanche, une fonction strictement convexe n'est pas forcément fortement convexe (exemple : $f(x) = -\log(x)$). Les fonctions fortement convexes de classes C^2 sont caractérisées par leur hessienne.

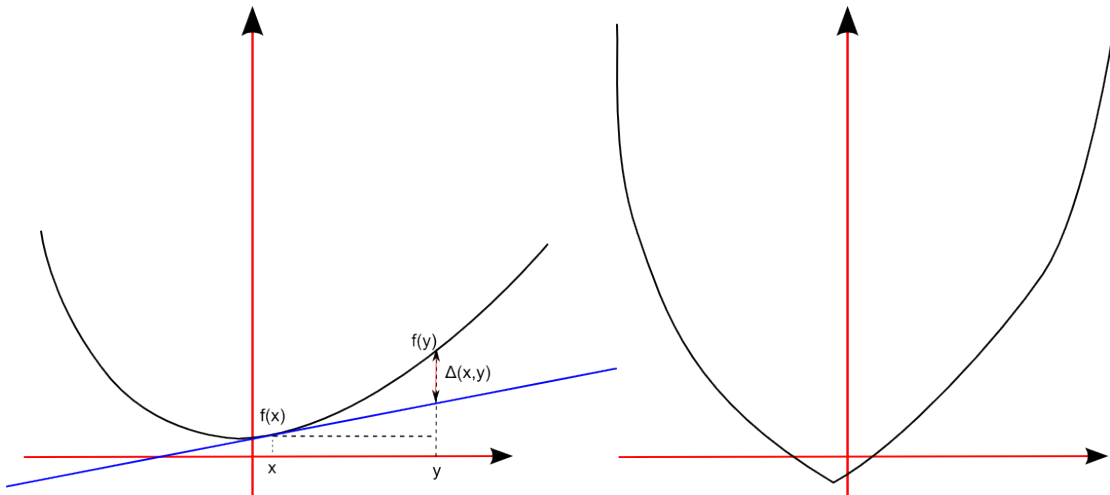


FIGURE 1.5 – Fonctions fortement convexes. Le graphe de la courbe s'éloigne rapidement de la tangente : $\Delta(x, y) \geq \frac{\mu}{2} \|y - x\|_2^2$. Notez qu'une fonction fortement convexe peut être non-différentiable et à domaine borné (voir figure de droite).

Proposition 1.5 Une fonction C^2 est μ -fortement convexe si et seulement si

$$\lambda_{\min}(H[f](x)) \geq \mu, \quad \forall x \in \mathbb{R}^n.$$

La proposition suivante est un des éléments qui permet d'obtenir des taux de convergence en norme :

Proposition 1.6 Soit f une fonction μ -fortement convexe qui admet pour minimiseur x^* . Pour tout $x \in \mathbb{R}^n$, on a :

$$f(x) \geq f(x^*) + \frac{\mu}{2} \|x - x^*\|^2.$$

Proposition 1.7 Soient f_1 et f_2 deux fonctions convexes de paramètres de forte convexité respectifs $\mu_1 \geq 0$ et $\mu_2 \geq 0$. Soient α et β deux réels positifs. Alors la fonction $f = \alpha f_1 + \beta f_2$ est μ -fortement convexe avec $\mu \geq \alpha\mu_1 + \beta\mu_2$.

Pour finir donnons quelques exemples de fonctions fortement convexes :

1. La fonction $f(x) = \frac{1}{2} \|x - x_0\|^2$ est 1 fortement convexe.
2. La fonction $f(x) = g(x) + \frac{1}{2} \|x - x_0\|^2$ où g est convexe est 1 fortement convexe.
3. La fonction $f(x) = \frac{1}{2} \|Ax - b\|^2$ est fortement convexe si A est de rang plein. Sa constante de forte convexité est alors égale à $\lambda_{\min}(A^*A)$.

4. La fonction $f(x) = -\log(x)$ n'est pas fortement convexe sur \mathbb{R}_+^* . Par contre, elle l'est sur tout intervalles $[a, b]$, $0 < a < b$. Sa constante de forte convexité est $\frac{1}{b^2}$.

1.3.3 Conditionnement d'une fonction

Les notions de constante de Lipschitz du gradient et de paramètre de forte convexité généralisent la notion de plus grande et plus petite valeur singulière d'une matrice. Elles permettent de définir une notion de conditionnement d'une fonction.

Définition 1.9 Soit f une fonction μ -fortement convexe, différentiable, telle que ∇f est L -Lipschitz. Le conditionnement de f est défini par la quantité :

$$Q_f = \frac{L}{\mu}$$

Le conditionnement apparait dans les taux de convergence de toutes les méthodes de premier ordre, de la même façon qu'en algèbre linéaire.

Pour toute fonction f , on a $\mu \leq L$ d'où :

$$Q_f \in [1, +\infty]$$

. Le conditionnement est infini dès lors que la fonction f est non différentiable ou non fortement convexe. De façon assez générale, les problèmes mal conditionnés tels que $Q_f \gg 1$ sont plus difficile à résoudre que les problèmes bien conditionnés. La Figure 1.6 montre les lignes de niveau de la fonction : $f_\alpha(x) = \frac{1}{2}\|D_\alpha x\|_2^2$ avec

$$D_\alpha = \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix}$$

et $\alpha \geq 1$. La hessienne de f_α est D_α et le conditionnement est donc $Q_{f_\alpha} = \alpha$. Plus α est grand, plus les lignes de niveaux sont resserrées. Une méthode de gradient a alors tendance à converger lentement en oscillant dans la "vallée" qui mène au minimiseur.

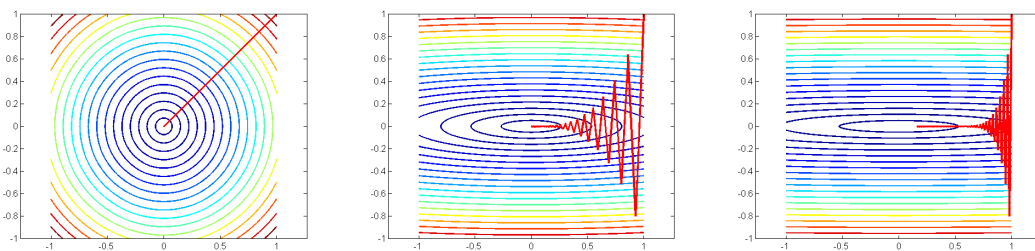


FIGURE 1.6 – Lignes de niveau de la fonction f_α pour $\alpha \in \{1, 5, 10\}$. La courbe rouge indique la trajectoire d'une descente de gradient. Plus le conditionnement α est élevé, plus les courbes de niveau sont allongées et plus la méthode de descente de gradient oscille.

Chapitre 2

Introduction à l'optimisation numérique

Optimiser : rendre optimal, donner à quelque chose les meilleures conditions d'utilisation, de fonctionnement ou de rendement au regard de certaines circonstances.

(Déf. du LAROUSSE).

2.1 Qu'est-ce qu'un problème d'optimisation ?

Soit X est un sous-ensemble non vide de \mathbb{R}^n . Considérons un problème d'optimisation de la forme :

$$\min f(x) \quad \text{s.c.} \quad x \in X, \quad (2.1)$$

La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est appelée fonction *coût*, *objectif* ou *critère*. L'ensemble X est appelé *ensemble* ou *domaine des contraintes*. Tout point $x \in \mathbb{R}^n$ vérifiant : $x \in X$, est appelé *point admissible* ou *point réalisable* du problème (2.1).

Chercher une solution du problème avec contraintes (2.1) revient à chercher un point de minimum local de f dans l'ensemble des points admissibles, au sens de la définition suivante :

Définition 2.1

- $x_0 \in \mathbb{R}^n$ est un point de minimum local de f sur $X \subset \mathbb{R}^n$ si et seulement si

$$x_0 \in X \quad \text{et} \quad \exists \mathcal{V}_{x_0} \text{ un voisinage de } x_0 \text{ tq : } \forall x \in \mathcal{V}_{x_0} \cap X, f(x) \geq f(x_0) \quad (2.2)$$

- $x_0 \in \mathbb{R}^n$ est un point de minimum global de f sur X si et seulement si

$$x_0 \in X \quad \text{et} \quad \forall x \in X, f(x) \geq f(x_0). \quad (2.3)$$

A noter que tout point de minimum global est aussi local. Les notions de maximum local et global sont définies de façon tout à fait similaire. Ces définitions sont illustrées sur la Figure 2.1

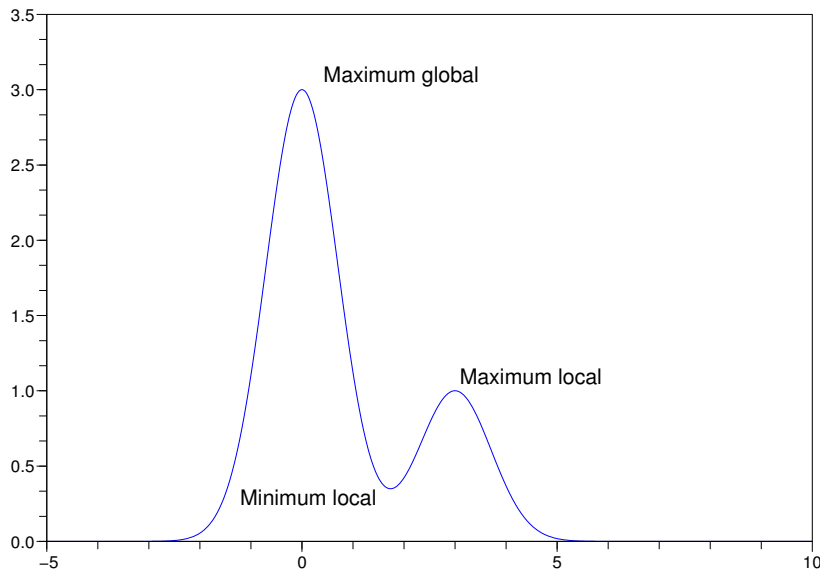


FIGURE 2.1 – Minima et maxima locaux et globaux de la fonction $x \mapsto 3e^{-x^2} + e^{-(x-3)^2}$.

On peut facilement démontrer que les problèmes (avec ou sans contraintes) $\min_x f(x)$ et $\max_x -f(x)$ sont équivalents dans le sens où ils ont même ensemble de solutions et :

$$\min_x f(x) = -\max_x -f(x) \quad \text{ou encore} \quad \max_x f(x) = -\min_x -f(x).$$

Ainsi la recherche d'un maximum pouvant se ramener à la recherche d'un minimum, nous porterons une attention plus particulière à la recherche du minimum.

Les problèmes d'optimisation peuvent être classés en plusieurs grandes familles :

- *Optimisation numérique* : $X \subset \mathbb{R}^n$.
- *Optimisation discrète (ou combinatoire)* : X fini ou dénombrable.
- *Commande optimale* : X est un ensemble de fonctions.
- *Optimisation stochastique* : données aléatoires
(à ne pas confondre avec les méthodes stochastiques d'optimisation).
- *Optimisation multicritères* : plusieurs fonctions objectifs.

Ce cours est un cours d'optimisation numérique : une étape importante en pratique consiste à identifier le type de problème auquel on a affaire afin de savoir quelle famille d'algorithme peut être pertinente.

Fonction coût	Contraintes	Domaine X	Terminologie
Linéaire $f(x) = c^\top x, c \in \mathbb{R}^n$	linéaires	Polytope / Polyèdre	Programmation linéaire (P.L.)
Linéaire	linéaires	$X \subset \mathbb{Z}$	P.L. en nombres entiers
Quadratique $f(x) = c^\top x + x^\top Qx$ $c \in \mathbb{R}^n, Q \in \mathbb{S}_n$	linéaires	Polytope / Polyèdre	Programmation quadratique
Convexe	convexes	convexe	Programmation convexe
qq	qq	qq	Programmation non linéaire (P.N.L.)

PROGRAMMATION LINÉAIRE		
		Méthode du simplexe Algorithme des points intérieurs
PROGRAMMATION NON LINÉAIRE		
SANS CONTRAINTE	AVEC DÉRIVÉES	Méthodes de type gradient Méthodes de Newton et quasi-Newton Gauss-Newton, Levenberg-Marquardt (pbs de moindres carrés)
	SANS DÉRIVÉES	(DFO, NEWUOA, MADS, NOMADS,...) Méthodes heuristiques : Nelder-Mead, surfaces de réponses (réseaux de neurones, krigeage) Méthodes stochastiques : méthodes à 2 phases algs génétiques, algs évolutionnaires, recuit simulé, recherche tabou
	NON-DIFF.	Méthodes de sous-gradient Méthodes de faisceaux Méthodes d'échantillonnage de gradient Algorithmes proximaux
	$f(x) = g(x) + h(x)$ avec g convexe diff. h convexe non diff.	Algorithmes de splitting : descente de gradient proximale (+variante accélérée) Douglas Rachford, ADMM
AVEC CONTRAINTES	AVEC DÉRIVÉES	Gradient projeté, algorithme de Uzawa Méthode SQP (Newton) Points intérieurs Méthodes de pénalisation, Lagrangien augmenté

2.2 Résultats d'existence et d'unicité en optimisation

Dans ce paragraphe, nous nous intéresserons aux théorèmes qui permettent d'assurer qu'il existe un minimum à un problème d'optimisation.

Considérons à nouveau le problème (2.1) mais d'un point de vue ensembliste : résoudre

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{sous la contrainte : } x \in X.$$

revient à chercher le minimum de l'image directe $f(X) = \{f(x) : x \in X\}$ de X par f .

Il existe principalement deux théorèmes donnant des conditions suffisantes d'existence d'un point de minimum : le premier dans le cas où l'ensemble des contraintes est fermé borné, le second pour un ensemble de contraintes fermé mais non borné.

2.2.1 Cas où l'ensemble X des contraintes est borné

Théorème 2.1 (Théorème de Weierstrass) *Soit X un ensemble fermé borné non vide de \mathbb{R}^n et $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ une application continue sur X . Alors f est bornée et atteint ses bornes. Autrement dit, f admet au moins un point $\underline{x} \in X$ de minimum global de f sur X :*

$$\forall y \in X, f(\underline{x}) \leq f(y),$$

et au moins un point $\bar{x} \in X$ de maximum global de f sur X :

$$\forall y \in X, f(\bar{x}) \geq f(y).$$

Remarque 2.1 *La condition de continuité de f n'est pas nécessaire, on peut la remplacer par la condition plus faible de semi-continuité inférieure qui dit essentiellement que $f(\lim x_n) \leq \lim f(x_n)$, alors que la continuité impose l'égalité.*

2.2.2 Cas où l'ensemble X des contraintes est non borné

Dans cette section nous nous intéressons au cas où l'ensemble des contraintes est non borné. C'est le cas tout particulièrement lorsqu'il n'y a pas de contraintes, c'est-à-dire $X = \mathbb{R}^n$. Le problème supplémentaire par rapport au cas précédent est qu'il faut empêcher l'infimum d'être à l'infini. Une fonction qui illustre ce problème est la fonction $f : x \mapsto 1/x$ qui admet 0 comme infimum sur \mathbb{R}^+ pour $x = +\infty$. Pour empêcher l'infimum d'être à l'infini, on peut utiliser l'hypothèse de "coercivité" de la fonction :

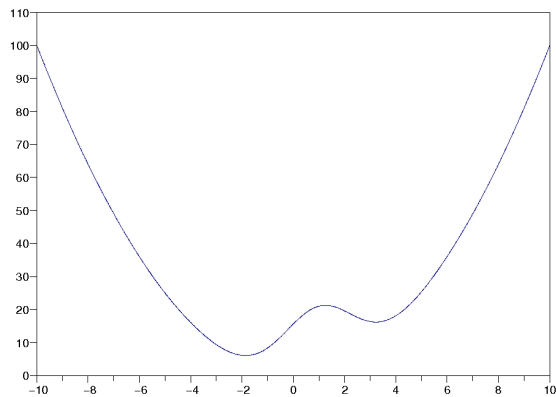
Définition 2.2 Une application $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est dite infinie à l'infini (ou coercive) sur X ssi

$$\forall A \in \mathbb{R}, \exists R > 0 \text{ tel que } \forall x \in X, \text{ si } \|x\| \geq R \text{ alors } f(x) \geq A. \quad (2.4)$$

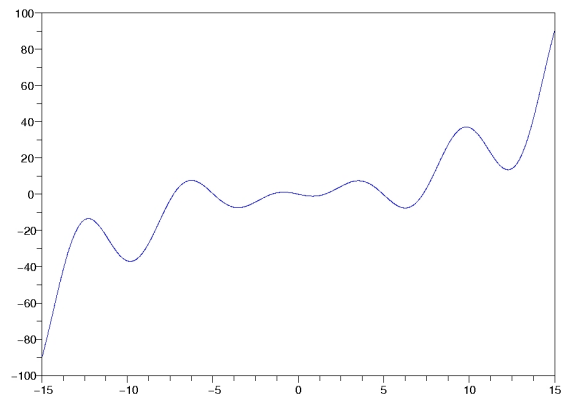
On note : $\lim_{\substack{\|x\| \rightarrow +\infty \\ x \in X}} f(x) = +\infty.$

Exemple 2.2.1 1. $f_1(x) = \|x\|_2$ est coercive.

2. $f_2(x) = x_1^2 - x_2^2$ n'est pas coercive : en effet, la suite de terme général $x_n = (0, n)$, $n \in \mathbb{N}$, est telle que : $\lim_{n \rightarrow +\infty} \|x_n\| = \lim_{n \rightarrow +\infty} n = +\infty$ mais : $\lim_{n \rightarrow +\infty} f_2(x_n) = \lim_{n \rightarrow +\infty} -n^2 = -\infty.$



Exemple de fonction coercive (f_1)



Exemple de fonction non coercive (f_2).

Comme la définition 2.2 n'est pas facile à manier en pratique, on utilise souvent la proposition suivante, qui est une hypothèse un peu plus forte, pour montrer que la fonction est infinie à l'infini.

Proposition 2.1 Soit $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ une application et $g : \mathbb{R} \rightarrow \mathbb{R}$ vérifiant

$$f(x) \geq g(\|x\|) \quad \text{avec} \quad \lim_{t \rightarrow +\infty} g(t) = +\infty.$$

Alors, f est infinie à l'infini.

Théorème 2.2 Soient F un fermé non vide de \mathbb{R}^n et $f : F \rightarrow \mathbb{R}$ une application continue infinie à l'infini sur F . Alors f admet un point de minimum global sur F , i.e. il existe $x \in F$ tel que

$$\forall y \in F, f(y) \geq f(x).$$

2.2.3 Cas particulier de contraintes d'égalités et/ou d'inégalités

L'hypothèse X fermé est assez difficile à montrer en pratique sauf dans le cas (fréquent en optimisation) où X est défini par des égalités et des inégalités :

$$X = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\}$$

où $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ et $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$. L'écriture " $h(x) = 0$ " représente en fait p contraintes d'égalité :

$$h_i(x) = 0, \quad i = 1, \dots, p,$$

et de même " $g(x) \leq 0$ " représente q contraintes d'inégalité :

$$g_i(x) \leq 0, \quad i = 1, \dots, q.$$

Dans le cas où les fonctions contraintes g et h sont continues, on a le résultat suivant :

Proposition 2.2 Soient $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$ et $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ deux fonctions continues.

- $X = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\}$ est un ensemble fermé de \mathbb{R}^n .
- $X = \{x \in \mathbb{R}^n : g(x) < 0\}$ est un ouvert de \mathbb{R}^n .

Ainsi, on peut conclure directement que si f , g et h sont continues et si :

- soit l'ensemble des contraintes $X = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\}$ est borné,
- soit f est infinie à l'infini,

alors le problème :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t. :} & g(x) \leq 0, h(x) = 0. \end{aligned}$$

admet au moins une solution globale. Autrement dit, f admet au moins un point de minimum global sur X .

2.2.4 Convexité et optimisation

Les résultats précédents ne nous donnent aucune information quant à l'unicité éventuelle d'un point de minimum global, ni sur le lien entre possibles minima locaux et minima globaux. C'est la notion de *convexité* qui permet de garantir que les minima locaux sont en fait des minima globaux. Attention la notion de convexité est suffisante pour garantir que nous avons trouvé un minimum global mais aucunement

Comme nous l'avons vu, la notion de convexité est une notion globale (elle doit être valable en toute paire de points), elle implique donc des propriétés globales sur les problèmes de minimisation. Notamment que tout point de minimum local devient global.

Théorème 2.3 Soit x^* un point de minimum local d'un problème de minimisation.

- i.* Si le problème est convexe, alors x^* est un point de minimum global.
- ii.* Si le problème est strictement convexe, alors x^* est l'unique point de minimum global.

Deuxième partie

Algorithmes classiques en optimisation différentiable

Chapitre 3

Conditions d'optimalité

Considérons un problème d'optimisation très général de la forme :

$$(P) \quad \min f(x) \quad \text{s.c.} \quad x \in X, \quad (3.1)$$

où X est un sous-ensemble non vide de \mathbb{R}^n et $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction que l'on suppose différentiable.

Les résultats d'existence de solutions que nous connaissons, ne sont d'aucune aide pour trouver une solution du problème (P). Ce qu'il nous faut, c'est une caractérisation analytique de l'optimalité, i.e. un ensemble d'équations ou d'inéquations qui pourront être résolues par les algorithmes.

Ce chapitre est consacré à l'écriture de conditions d'optimalité associées à différents problèmes d'optimisation. Tout d'abord nous donnerons une condition nécessaire d'optimalité du problème (3.1), puis nous montrerons comment s'écrit cette condition dans les cas suivants :

1. X ouvert de \mathbb{R}^n .
2. X convexe de \mathbb{R}^n .
3. X défini par des égalités et/ou des inégalités fonctionnelles :

$$X = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\}.$$

3.1 Cas général : condition d'optimalité géométrique

3.1.1 Le cône tangent ou comment se déplacer dans le domaine des contraintes

Une difficulté importante en optimisation sous contrainte consiste à savoir se déplacer dans l'ensemble des contraintes i.e. étant donnée une direction de recherche comment garantir que l'on reste dans l'ensemble X . Pour cela, on introduit la notion de direction admissible :

Définition 3.1 (Direction admissible) Soit $x \in X$. Une direction $d \in \mathbb{R}^n$ sera dite admissible en x s'il existe $\eta > 0$ tel que $x + sd$ soit admissible quel que soit $s \in]0, \eta]$. On dit également que la direction d est rentrante dans X en x .

Dans le cas particulier où le domaine des contraintes est convexe, déterminer une direction d admissible en x revient à déterminer un point admissible y , différent de x : $d = y - x$ est alors une direction admissible.

Définition 3.2 (Cône tangent) Soit $x \in X$. Le cône tangent (ou cône des directions admissibles) à X en x , noté $T_x(X)$, est l'ensemble des vecteurs $v \in \mathbb{R}^n$ tels qu'il existe une suite $(v_n)_{n \in \mathbb{N}}$ de limite v et une suite $(\varepsilon_n)_{n \in \mathbb{N}}$ de réels strictement positifs de limite nulle telles que :

$$x + \varepsilon_n v_n \in X.$$

Autrement dit, le cône admissible est l'ensemble des directions admissibles dans X au point x , ainsi que les limites de ces directions.

Le passage à la limite est essentiel, sous peine d'appauvrir radicalement le cône $T_x(X)$ et de le rendre ainsi inutilisable. On remarquera que $T_x(X)$ est un cône fermé, et que les seuls cas intéressants sont ceux où le point x est sur la frontière de X .

Proposition 3.1 Soit X un sous-ensemble d'intérieur non vide de \mathbb{R}^n et $x \in X$.

- i. $T_x(X)$ est un cône fermé.
- ii. $x \notin \bar{X} \Rightarrow T_x(X) = \emptyset$.
- iii. $x \in \text{int } X \Rightarrow T_x(X) = \mathbb{R}^n$.

Démonstration. Faite en cours. □

Exercice 3.1.1 Soit $X = \{x \in \mathbb{R}^2 | x_1^2 \leq x_2 \leq 2x_1^2\}$. Dessiner X dans \mathbb{R}^2 et calculer l'ensemble des directions admissibles à X au point $(0, 0)$ puis le cône tangent à X en $(0, 0)$.

Exercice 3.1.2 Soit $X = \{x \in \mathbb{R}^2 | x_1^2 \leq x_2^2\}$. Dessiner X dans \mathbb{R}^2 et calculer $T_{(0,0)}(X)$ et $T_{(1,1)}(X)$

3.1.2 Condition nécessaire d'optimalité géométrique

L'écriture des conditions d'optimalité en présence de contraintes est basée sur l'intuition qu'il est impossible de "descendre" à partir d'un minimum. Considérons le problème général :

$$(P) \quad \min_{x \in \mathbb{R}^n} f(x) \quad \text{sous la contrainte : } x \in X.$$

La condition nécessaire d'optimalité du premier ordre énoncée ci-après met en évidence l'utilité de la notion de cône tangent :

Théorème 3.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une application différentiable et x^* un minimum local du problème (P). Alors :

$$\forall v \in T_{x^*}(X), \langle \nabla f(x^*), v \rangle \geq 0. \quad (3.2)$$

Démonstration. Faite en cours.

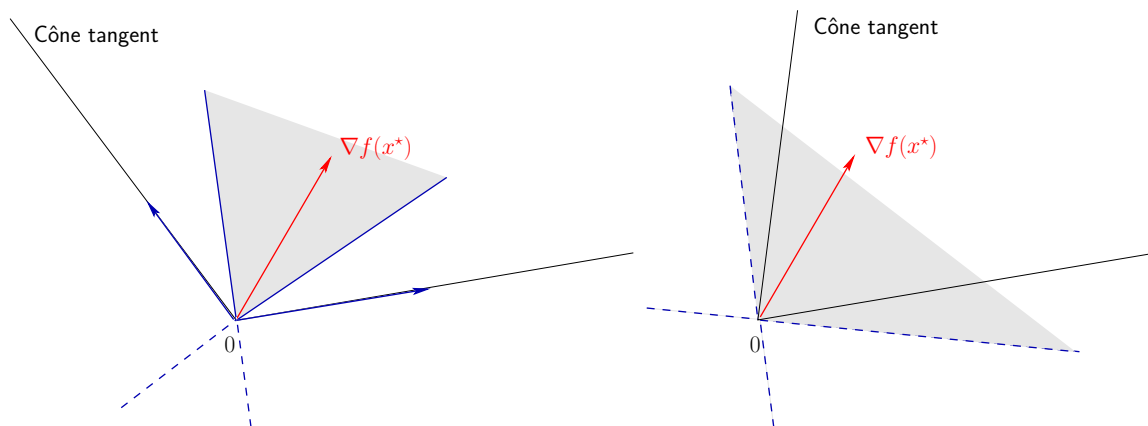


FIGURE 3.1 – Interprétation géométrique de la condition nécessaire d'optimalité (3.2)

3.1.3 Cas où l'ensemble des contraintes est convexe

Dans le cas où le domaine X des contraintes est convexe, la condition d'optimalité se simplifie dans le sens où elle ne fait plus intervenir le cône tangent mais uniquement l'ensemble des directions admissibles. Bien que ce dernier soit plus petit que le cône tangent, il n'y a pas de perte d'information lorsque f est différentiable en x^* , que son adhérence est le cône tangent à X en x^* et que $v \mapsto \langle \nabla f(x^*), v \rangle$ est continue.

Corollaire 3.1 (CN et CS en présence de convexité) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une application différentiable et X un sous-ensemble convexe de \mathbb{R}^n . Si x^* est un minimum local de (P), alors :

$$\forall x \in X, \langle \nabla f(x^*), x - x^* \rangle \geq 0. \quad (3.3)$$

Si f est convexe sur le convexe X , alors la condition (3.3) est suffisante pour que x^* soit un point de minimum global de f sur X .

Exercice 3.1.3 Démontrer le corollaire 3.1.

3.2 Cas différentiable sans contraintes

Supposons maintenant que le domaine X des contraintes est un ouvert de \mathbb{R}^n . Par conséquent : $\text{int } X = X$, d'où :

$$T_x(X) = \mathbb{R}^n.$$

Proposition 3.2 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ supposée différentiable et X un ouvert de \mathbb{R}^n . Si x^* est un point de minimum local de f sur X alors :

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

Démonstration. Faite en cours. □

Nous retrouvons ainsi la condition nécessaire d'optimalité du premier ordre pour l'optimisation sans contrainte. Cela nous rappelle qu'un problème d'optimisation sur un ouvert, doit être traité comme un problème d'optimisation sans contrainte et les contraintes vérifiées *a posteriori*.

Exercice 3.2.1 Les problèmes $\min_{x \in \mathbb{R}} x^2$ s.t. $|x| < 1$ et $\min_{x \in \mathbb{R}} x^2$ s.t. $x > 1$ ont-ils des solutions ?

Rappelons (sans démonstration) les conditions nécessaires du premier et du second ordre pour des problèmes d'optimisation différentiable sans contrainte :

Conditions nécessaires d'optimalité locale

Si $x^* \in \mathbb{R}^n$ réalise un minimum local (resp. maximum local) de f , alors :

$$\nabla f(x^*) = 0 \quad (\text{CN d'optimalité du } 1^{\text{er}} \text{ ordre})$$

$$H[f](x^*) \text{ est semidéfinie positive} \quad (\text{CN d'optimalité du } 2^{\text{nd}} \text{ ordre})$$

(resp. $H[f](x^*)$ est semidéfinie négative)

Condition suffisante d'optimalité locale

Soit X un ouvert de \mathbb{R}^n et $x^* \in X$. Si :

$$\nabla f(x^*) = 0 \quad \text{et} \quad H[f](x^*) \text{ symétrique, définie positive (resp. définie négative)}$$

Alors x^* est un point de minimum local (resp. maximum local) de f sur X .

Condition Suffisante d'optimalité globale

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

- i. Si f est convexe, alors x^* est un point de minimum global de f .
- ii. Si f est strictement convexe, alors x^* est l'unique point de minimum global de f .

3.3 Cas différentiable avec contraintes fonctionnelles

Nous nous intéressons maintenant plus particulièrement à la résolution de problèmes d'optimisation dont le domaine des contraintes X est défini par des égalités et/ou des inégalités :

$$X = \{x \in \mathbb{R}^n : h_i(x) = 0, i = 1, \dots, p, g_j(x) \leq 0, j = 1, \dots, q\}$$

noté également :

$$X = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\}$$

où les fonctions $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$ et $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ sont supposées différentiables sur \mathbb{R}^n . On rappelle que par continuité des applications g et h , l'ensemble X est un fermé de \mathbb{R}^n , non nécessairement borné (penser au cas $h \equiv 0$ et $g(x, y) = -x$).

A partir de maintenant, nous travaillerons toujours à partir d'un problème d'optimisation écrit sous *forme standard*, à savoir :

$$(P) \left| \begin{array}{l} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.c. } h_i(x) = 0, \quad i = 1, \dots, p \\ \quad \quad g_j(x) \leq 0, \quad j = 1, \dots, q. \end{array} \right.$$

Proposition 3.3 *Le problème (P) est dit convexe si h est affine, g convexe et si la fonction objectif f est convexe sur X .*

Exercice 3.3.1 Donner la forme standard du problème $\max_{(x,y) \in \mathbb{R}^2} f(x,y)$ s.t. : $x^2 + y^2 \geq 1$ et $x + y = 5$. Ce problème est-il convexe ?

L'essentiel de notre travail dans ce chapitre consistera donc à trouver une expression plus "pratique" de la condition (3.2), et donc du cône tangent associé à un ensemble d'égalités et/ou d'inégalités fonctionnelles.

3.3.1 Problèmes avec contraintes d'égalité

Considérons un problème avec contraintes uniquement d'égalité :

$$(P_E) \left| \begin{array}{l} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.c. } h_i(x) = 0, \quad i = 1, \dots, p. \end{array} \right.$$

On note : $X = \{x \in \mathbb{R}^n : h_i(x) = 0, i = 1, \dots, p\}$ le domaine admissible. On cherche les conditions d'optimalité associées au problème (P_E) . La première étape (la plus difficile) consiste à expliciter le cône tangent.

Etape 1 : Qualification des contraintes : comment calculer le cône tangent à X .

Définition 3.3 (Qualification des contraintes d'égalité) La contrainte $h(x) = 0$ du problème (P_E) est dite qualifiée en $x \in X$ si h est différentiable en x et si :

$$T_x(X) = \{v \in \mathbb{R}^n \mid \forall j = 1, \dots, p, \langle \nabla h_j(x), v \rangle = 0\} (= \text{Ker } Dh(x)).$$

Comment interpréter cette définition ? Le cône tangent ne dépend que de l'ensemble X et pas de la fonction h choisie pour le représenter. Au contraire $\text{Ker } Dh(x)$ dépend directement de h . La qualification des contraintes peut être vue comme un critère permettant de sélectionner de bonnes représentations de X par h .

En pratique, cette définition de qualification des contraintes est difficile à vérifier ; une hypothèse couramment utilisée pour garantir la qualification des contraintes est qu'elles soient régulières.

Définition 3.4 (Contraintes régulières) Soit $x \in X$. Supposons h de classe C^1 au voisinage de x . On dit que les contraintes du problème (P_E) sont régulières en $x \in X$, ou que $x \in X$ est régulier, si la jacobienne $Dh(x)$ des contraintes est surjective.

Sous l'hypothèse que h est de classe C^1 au voisinage de $x \in X$, on rappelle que la matrice Jacobienne $Dh(x)$ de h en x est définie par :

$$Dh(x)^\top = [\nabla h_1(x)^\top \mid \dots \mid \nabla h_p(x)^\top].$$

La matrice $Dh(x)$ étant de taille $p \times n$. On a alors :

Proposition 3.4 Le point $x \in X$ est régulier ssi les vecteurs $\nabla h_i(x)$, $i = 1, \dots, p$, sont linéairement indépendants.

Proposition 3.5 (CS de qualification des contraintes) Soit $x \in X$. Si x est régulier, alors les contraintes d'égalité " $h = 0$ " sont qualifiées au point x i.e. :

$$T_x(X) = \text{Ker } Dh(x) = \{v \in \mathbb{R}^n \mid \forall j = 1, \dots, p, \langle \nabla h_j(x), v \rangle = 0\}.$$

Démonstration de la proposition 3.5.

i. Commençons par démontrer que : $T_x(X) \subset \text{Ker } Dh(x) = \{v \in \mathbb{R}^n \mid \langle v, \nabla h_i(x) \rangle = 0\}$.

Soit $v \in T_x(X)$: il existe une suite $(v_n)_{n \in \mathbb{N}}$ de \mathbb{R}^n qui converge vers v et une suite $(\varepsilon_n)_{n \in \mathbb{N}}$ de réels strictement positifs telles que : $x_n = x + \varepsilon_n v_n \in X$. Remarquons qu'on a alors également : $(x_n)_n$ converge vers x .

Soit $i \in \{1, \dots, p\}$. Montrons que $\langle \nabla h_i(x), v \rangle = 0$. La contrainte h_i étant supposée C^1 , pour n assez grand, on peut écrire :

$$h_i(x + \varepsilon_n v_n) = h_i(x) + \varepsilon_n \langle \nabla h_i(x), v_n \rangle + \varepsilon_n \|v_n\| \gamma(n), \text{ avec : } \lim_{n \rightarrow +\infty} \gamma(n) = 0.$$

Or $x + \varepsilon_n v_n$ et x sont des éléments de X , d'où : $h_i(x + \varepsilon_n v_n) = h_i(x) = 0$ et :

$$\langle \nabla h_i(x), v_n \rangle + \|v_n\| \gamma(n) = 0.$$

Par passage à la limite quand n tend vers $+\infty$, on obtient alors : $\langle \nabla h_i(x), v \rangle = 0$ et ce pour tout $i \in \{1, \dots, p\}$. Donc $v \in \text{Ker } Dh(x)$, et : $T_x(X) \subset \text{Ker } Dh(x)$.

ii. Réciproquement, montrons maintenant que $\text{Ker } Dh(x) \subset T_x(X)$.

Étape 1 : Le but de cette étape est de décrire explicitement les vecteurs vérifiant les contraintes, i.e. l'équation implicite : $h(x) = 0$ à l'aide du théorème d'inversion locale.

Le point $x \in X$ étant supposé régulier, l'application linéaire $Dh(x)$ est de rang p . Quitte à renuméroter les variables, on suppose les p dernières colonnes de $Dh(x)$ linéairement indépendantes. Donc h s'écrit :

$$\begin{aligned} h : \mathbb{R}^{n-p} \times \mathbb{R}^p &\rightarrow \mathbb{R}^p \\ y = (u, v) &\mapsto h(y) = h(u, v) \end{aligned}$$

Notons : $x = (u_0, v_0)$. On a :

- h est de classe C^1 au voisinage de x ,
- $h(x) = 0$,
- la jacobienne de h par rapport à v , notée $D_v h(x)$, est la matrice carrée d'ordre p extraite de $Dh(x)$ en sélectionnant les p dernières colonnes :

$$D_v h(x) = \left[Dh(x) \right]_{1 \leq i \leq p, n-p+1 \leq j \leq n}.$$

C'est une matrice de rang p et de taille $p \times p$, donc inversible.

D'après le théorème des fonctions implicites, il existe un voisinage U_0 de u_0 , un voisinage V_0 de v_0 et une application $\varphi : U_0 \rightarrow V_0$ tels que :

$$\begin{cases} (u, v) \in U_0 \times V_0 \\ h(u, v) = 0 \end{cases} \Leftrightarrow \begin{cases} (u, v) \in U_0 \times V_0 \\ v = \varphi(u) \end{cases}$$

On pose alors : $\psi(u) = (u, \varphi(u))$ dont l'image $\text{Im } \psi$ décrit l'ensemble des points vérifiant les contraintes :

$$\begin{cases} \psi(u_0) = (u_0, \varphi(u_0)) = (u_0, v_0) = x, \\ \forall u \in U_0, \quad \psi(u) \in X, \end{cases} \quad \text{i.e. : } \begin{cases} \psi(u_0) = x, \\ \forall u \in U_0, \quad h(\psi(u)) = 0 \end{cases}$$

Étape 2 : on réécrit le noyau de $Dh(x)$ de la façon suivante : $\text{Ker } Dh(x) = \text{Im } \psi(u_0)$. En effet, différencions la relation : $h(\psi(u)) = 0$ au voisinage de u_0 :

$$Dh(\psi(u_0)) \circ D\psi(u_0) = 0, \quad \text{soit : } Dh(x) \circ D\psi(u_0) = 0.$$

D'où : $\text{Im } \psi(u_0) \subset \text{Ker } Dh(x)$. De plus,

$$D\psi(u_0) = \begin{pmatrix} I_{n-p} \\ D\varphi(u_0) \end{pmatrix}$$

est de rang $(n - p)$, ainsi que $\text{Ker } Dh(x)$. Donc : $\text{Im } \psi(u_0) = \text{Ker } Dh(x)$.

Étape 3 : soit $v \in \text{Ker } Dh(x) = \text{Im } D\psi(u_0)$: il existe $z \in \mathbb{R}^{n-p}$ tel que : $v = D\psi(u_0)(z)$. Soit $(\varepsilon_n)_{n \in \mathbb{N}}$ une suite de réels strictement positifs, de limite nulle. Au voisinage de u_0 , on écrit le développement de Taylor d'ordre 1 de ψ , soit :

$$\begin{aligned} \psi(u_0 + \varepsilon_n z) &= \psi(u_0) + \varepsilon_n D\psi(u_0)(z) + \varepsilon_n \gamma(n), \text{ avec : } \lim_{n \rightarrow 0} \gamma(n) = 0 \\ &= x + \varepsilon_n v + \varepsilon_n \gamma(n). \end{aligned}$$

On pose : $x_n = \psi(u_0 + \varepsilon_n z)$. Par construction (cf étape 1), les x_n sont des éléments de X . On définit maintenant la suite $(v_n)_{n \in \mathbb{N}}$ de la façon suivante :

$$v_n = \frac{x_n - x}{\varepsilon_n} = v + \gamma(n),$$

On a d'une part : $x + \varepsilon_n v_n = x_n \in X$, et d'autre part : $v_n = v + \gamma(n) \xrightarrow{n \rightarrow +\infty} v$. D'après la définition 3.2 du cône tangent, on a donc : $v \in T_x(X)$. \square

Étape 2 : Condition nécessaire d'optimalité

Théorème 3.2 (Théorème de Lagrange. CN d'optimalité du premier ordre)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ différentiables en x^* . Supposons la contrainte $h(x) = 0$ qualifiée en $x^* \in X$. Si x^* est un point de minimum local de f sur X , alors il existe des réels $\lambda_1^*, \dots, \lambda_p^*$ tels que :

$$\nabla f(x^*) + \sum_{i=1}^p \lambda_i^* \nabla h_i(x^*) = 0.$$

Le vecteur $\lambda^* = [\lambda_1^* \dots \lambda_p^*]^\top$ est appelé **multiplicateur de Lagrange** et est déterminé de façon unique.

Démonstration du théorème 3.2 en exercice.

TERMINOLOGIE :

- Le vecteur λ^* est aussi appelé solution *duale* du problème (P_E) , x^* solution *primale* de (P_E) et (x^*, λ^*) solution *primate-duale* de (P_E) .
- On appelle *point stationnaire* du problème (P_E) tout point \bar{x} vérifiant les conditions nécessaires d'optimalité du premier ordre :

$$\begin{cases} \nabla f(\bar{x}) + \sum_{i=1}^p \bar{\lambda}_i \nabla h_i(\bar{x}) = 0 \\ h(\bar{x}) = 0 \end{cases}$$

pour un certain multiplicateur $\bar{\lambda} \in \mathbb{R}^p$.

Réécriture des conditions d'optimalité à l'aide du Lagrangien

En pratique, on retrouve les conditions d'optimalité du problème (P_E) en introduisant le *Lagrangien* du problème (P_E) :

$$L(x; \lambda) = f(x) + \sum_{i=1}^p \lambda_i h_i(x), \quad x \in \mathbb{R}^n, \quad \lambda \in \mathbb{R}^p.$$

Sous hypothèse de qualification des contraintes, les conditions d'optimalité du problème (P_E) s'écrivent alors :

$$\begin{cases} \nabla_x L(x; \lambda) = 0 \\ h(x) = 0 \end{cases} \quad \text{ou encore : } \nabla_{(x,\lambda)} L(x; \lambda) = 0.$$

Nous admettrons sans démonstration que les conditions nécessaire et suffisante d'optimalité du second ordre s'obtiennent en étudiant le Hessien du Lagrangien calculé selon la composante x uniquement :

CN d'optimalité locale du second ordre

$$\forall v \in T_x(X), \quad \langle H_x[L](x; \lambda) v, v \rangle \geq 0.$$

CS d'optimalité locale du second ordre

$$(CS 1) \quad \forall v \in T_x(X), \quad v \neq 0, \quad \langle H_x[L](x; \lambda) v, v \rangle > 0.$$

Remarquons que si la matrice hessienne $H_x[L](x; \lambda)$ est définie positive, alors il est inutile de calculer le cône tangent au point x puisque dans ce cas :

$$\forall v \in \mathbb{R}^n, \quad v \neq 0, \quad \langle H_x[L](x) v, v \rangle > 0,$$

et donc :

$$\forall v \in T_x(X), \quad v \neq 0, \quad \langle H_x[L](x) v, v \rangle > 0.$$

La condition suffisante d'optimalité du second ordre est automatiquement satisfaite.

Une autre CS d'optimalité locale du second ordre

$$(CS 2) \quad \forall v \in \mathbb{R}^n, \quad v \neq 0, \quad \langle H_x[L](x; \lambda) v, v \rangle > 0.$$

On a :

$$(CS 2) \quad \implies \quad (CS 1).$$

Exercice 3.3.2 Trouver tous les rectangles de \mathbb{R}^2 de surface maximale pour un périmètre p fixé.

3.3.2 Problèmes avec contraintes d'inégalités

Intéressons nous maintenant aux problèmes d'optimisation avec contraintes d'inégalité :

$$(P_I) \quad \min_{x \in \mathbb{R}^n} f(x) \quad \text{sous la contrainte : } g_j(x) \leq 0, \quad j = 1, \dots, q.$$

On note : $X = \{x \in \mathbb{R}^n : g_j(x) \leq 0, \quad j = 1, \dots, q\}$.

Etape 1 : Qualification des contraintes : comment calculer le cône tangent à X .

Définition 3.5 (Qualification des contraintes d'inégalité) *La contrainte $g \leq 0$ du problème (P_I) est dite qualifiée au point $x \in X$ si g est différentiable en x et si :*

$$T_x(X) = \{v \in \mathbb{R}^n : \text{pour tout } j \text{ actif en } x, \langle \nabla g_j(x), v \rangle \leq 0\}.$$

L'interprétation est la même que dans le cas de contraintes d'égalité : la qualification des contraintes peut être vue comme un critère permettant de sélectionner de "bonnes" représentations de X par g .

Remarque 3.1 *La plupart des algorithmes échouent lorsqu'ils doivent résoudre un problème dont les contraintes ne sont pas qualifiées en la solution. Il est dans ce cas préférable de changer la description de l'ensemble X des contraintes avant de chercher à résoudre le problème.*

Une fois de plus, cette condition est difficile à vérifier en pratique et couramment remplacée par des conditions suffisantes de qualification des contraintes :

Proposition 3.6 (CS de qualification des contraintes) *Les contraintes $g_j \leq 0, \quad j = 1, \dots, q$, sont qualifiées en un point $x \in X$ s'il existe un vecteur $v \neq 0$ tel que pour toute contrainte active $g_j(x) = 0$, on ait :*

- soit g_j est affine et : $\langle \nabla g_j(x), v \rangle = 0$,
- soit : $\langle \nabla g_j(x), v \rangle < 0$.

Démonstration. Soit $v \in \mathbb{R}^n$ tel que pour toute contrainte j active en x , on ait : $\langle \nabla g_j(x), v \rangle \leq 0$.

Soit u un vecteur de \mathbb{R}^n assurant la qualification des contraintes en x . On se donne une suite $(\varepsilon_n)_n$ quelconque de réels strictement positifs, de limite nulle et un réel $\eta > 0$. On pose :

$$x_n = x + \varepsilon_n(v + \eta u).$$

La suite $(x_n)_n$ ainsi définie converge vers x et : $\lim_{n \rightarrow +\infty} \frac{x_n - x}{\varepsilon_n} = v + \eta u$. Pour que v soit une direction tangente à X en x , il suffit de montrer que : $x_n \in X$ à partir d'un certain rang.

1^{er} cas : la contrainte j n'est pas active en x . Alors : $g_j(x) < 0$. De plus la suite $(x_n)_n$ converge vers x , d'où par continuité de g_j : il existe un entier $N \in \mathbb{N}$ tel que : $\forall n \geq N, \quad g_j(x_n) \leq 0$.

2nd cas : la contrainte j est active en x . Alors : $g_j(x) = 0$ et deux sous-cas se présentent :

— soit g_j est affine et : $\langle \nabla g_j(x), u \rangle = 0$. Dans ce cas :

$$g_j(x_n) = g_j(x) + \varepsilon_n \langle \nabla g_j(x), v + \eta u \rangle = \varepsilon_n \langle \nabla g_j(x), v \rangle \leq 0$$

— soit : $\langle \nabla g_j(x), u \rangle < 0$. Sachant que : $g_j(x) = 0$, le développement de Taylor d'ordre 1 de g_j au voisinage de x , s'écrit :

$$\begin{aligned} g_j(x_n) &= \varepsilon_n \langle \nabla g_j(x), v + \eta u \rangle + o(\varepsilon_n) = \varepsilon_n \langle \nabla g_j(x), v \rangle + \eta \varepsilon_n \langle \nabla g_j(x), u \rangle + o(\varepsilon_n) \\ &< \eta \varepsilon_n \langle \nabla g_j(x), u \rangle + o(\varepsilon_n) \leq 0 \end{aligned} \quad \text{à partir d'un certain rang.}$$

Dans tous les cas : $x_n \in X$ à partir d'un certain rang, soit : $v + \eta u \in T_x(X)$ pour tout $\eta > 0$. Par passage à la limite quand η tend vers 0^+ , le cône $T_x(X)$ étant fermé, on en conclut : $v \in T_x(X)$. \square

Une autre condition suffisante (un peu plus restrictive) est souvent utilisée en pratique :

Proposition 3.7 *Les contraintes $g_j \leq 0$, $j = 1, \dots, q$ sont qualifiées au point $x \in X$ si les gradients des contraintes actives en x sont linéairement indépendants.*

A noter que, dans le cas convexe, ces conditions de qualification des contraintes s'écrivent plus simplement sous la forme :

Proposition 3.8 *Supposons les fonctions g_j , $j = 1, \dots, q$, convexes. Les contraintes $g_j \leq 0$, $j = 1, \dots, q$ sont qualifiées en tout point de X s'il existe un point $x_0 \in X$ tel que pour tout $j = 1, \dots, q$, on a :*

- soit $g_j(x_0) < 0$
- soit g_j est affine.

Etape 2 : Condition nécessaire d'optimalité

Si x est solution du problème (P_I) avec contraintes d'inégalités et si ces contraintes sont qualifiées en x au sens de la Proposition 3.6, alors la condition nécessaire d'optimalité (3.2) s'écrit :

$$\forall v \in \mathbb{R}^n, \quad \left[\forall j \text{ actif en } x, \quad \langle \nabla g_j(x), v \rangle \leq 0 \right], \quad \Rightarrow \quad \langle \nabla f(x), v \rangle \geq 0.$$

En utilisant le lemme de Farkas (ici admis) :

Lemme 3.1 (Lemme de Farkas) *Soient v_1, \dots, v_l un ensemble de vecteurs de \mathbb{R}^n et $u \in \mathbb{R}^n$. Alors les propositions suivantes sont équivalentes :*

i. $\forall x \in \mathbb{R}^n, \quad \left[\forall j = 1, \dots, l, \quad \langle v_j, x \rangle \leq 0 \right], \quad \Rightarrow \quad \langle u, x \rangle \geq 0.$

ii. *Il existe des réels $\lambda_1, \dots, \lambda_l$ positifs tels que :* $u = - \sum_{j=1}^l \lambda_j v_j.$

on obtient immédiatement la formulation suivante :

Théorème 3.3 (CN d'optimalité du 1^{er} ordre sous contraintes d'inégalités)

Soit $x^* \in X$. Supposons f et g différentiables en x^* et les contraintes qualifiées en x^* . Si x^* est un point de minimum local de f sur X , alors il existe des réels positifs ou nuls $\lambda_1^*, \dots, \lambda_q^*$ tels que :

$$\nabla f(x^*) + \sum_{j=1}^q \lambda_j^* \nabla g_j(x^*) = 0,$$

avec pour tout $j = 1, \dots, q$, $\lambda_j^* \geq 0$ et $\lambda_j^* g_j(x^*) = 0$.

TERMINOLOGIE :

- Le vecteur λ^* est aussi appelé solution *duale* du problème (P_I) , x^* solution *primale* de (P_I) et (x^*, λ^*) solution *primate-duale* de (P_I) .
- On appelle *point stationnaire* du problème (P_I) tout point \bar{x} vérifiant les conditions nécessaires d'optimalité du premier ordre :

$$\begin{cases} \nabla f(\bar{x}) + \sum_{j=1}^q \lambda_j \nabla g_j(\bar{x}) = 0 \\ \lambda_j \geq 0, & j = 1, \dots, q, \\ \lambda_j g_j(\bar{x}) = 0, & j = 1, \dots, q. \end{cases}$$

pour un certain multiplicateur $\lambda \in \mathbb{R}^q$.

- Les relations : $\lambda_j g_j(\bar{x}) = 0$ sont appelées *relations de complémentarité* et signifie :

$$\text{Soit : } \lambda_j = 0, \quad \text{soit : } g_j(\bar{x}) = 0.$$

Ces relations sont trivialement satisfaites par toute contrainte j active en \bar{x} et indiquent que pour toute contrainte inactive, le multiplicateur λ_j correspondant est nul. Cela signifie que toute contrainte inactive à l'optimum aurait pu être relaxée.

Réécriture des conditions d'optimalité à l'aide du Lagrangien

En pratique, on retrouve les conditions d'optimalité du problème (P_I) en introduisant le *Lagrangien* associé au problème (P_I) :

$$L(x; \lambda) = f(x) + \sum_{j=1}^q \lambda_j g_j(x), \quad x \in \mathbb{R}^n, \quad \lambda \in \mathbb{R}^q.$$

Sous hypothèse de qualification des contraintes, les conditions nécessaires d'optimalité du premier ordre du problème (P_I) s'écrivent alors :

$$\begin{cases} \nabla_x L(x; \lambda) = 0 \\ \lambda_j g_j(x) = 0, \quad j = 1, \dots, q \\ \lambda_j \geq 0, \quad j = 1, \dots, q. \end{cases}$$

Remarque 3.2 Concernant les conditions d'optimalité du second ordre, le cas des problèmes avec des contraintes d'inégalité est délicat et ne figure pas dans ce cours.

3.3.3 Problèmes avec contraintes d'égalité et d'inégalité

Considérons un problème d'optimisation avec des contraintes d'égalité et d'inégalité :

$$(P) \quad \min f(x)$$

$$\text{s.c.} \quad h_i(x) = 0, \quad i = 1, \dots, p, \quad g_j(x) \leq 0, \quad j = 1, \dots, q.$$

Notons X le domaine des contraintes. Les contraintes sont dites qualifiées au point $x \in X$ si chacune est qualifiée au sens des définitions vues précédemment à savoir :

Définition 3.6 *Les contraintes du problème (P) sont dites qualifiées au point $x \in X$ si g et h sont différentiables au point x et si :*

$$T_x(X) = \{v \in \mathbb{R}^n; \quad \forall i = 1, \dots, p, \langle \nabla h_i(x), v \rangle = 0$$

$$\text{et pour toute contrainte } j \text{ active, } \langle \nabla g_j(x), v \rangle \leq 0\}.$$

Cette condition est rarement utilisée en pratique car difficile à vérifier. Il existe plusieurs conditions suffisantes de qualification des contraintes utilisées dans la littérature (cf qualification de Slater, de Mangasarian-Fromovitz e.g.); la plus simple est la suivante :

Proposition 3.9 (CS de qualification des contraintes) *Les contraintes du problème (P) sont qualifiées au point $x \in X$ si les gradients des contraintes actives en x :*

$$\{\nabla h_i(x); i = 1, \dots, p\} \cup \{\nabla g_j(x); j \in \{1, \dots, q\} \text{ active}\},$$

sont linéairement indépendants.

On introduit le Lagrangien associé au problème (P) :

$$L(x; \lambda, \mu) = f(x) + \sum_{i=1}^p \lambda_i h_i(x) + \sum_{j=1}^q \mu_j g_j(x), \quad x \in \mathbb{R}^n, \quad \lambda \in \mathbb{R}^p, \quad \mu \in \mathbb{R}^q.$$

Théorème 3.4 (CN d'optimalité de Karush-Kuhn-Tucker (1951)) *Soit $x^* \in X$ un point admissible du problème (P). Supposons f , g et h différentiables en x^* et les contraintes qualifiées au point x^* . Si x^* est un point de minimum local de f sur X alors il existe $\lambda^* \in \mathbb{R}^p$ et $\mu^* \in \mathbb{R}^q$ tels que :*

$$\begin{cases} \nabla_x L(x^*; \lambda^*, \mu^*) = 0 \\ h_i(x^*) = 0, & i = 1, \dots, p \\ \mu_j^* g_j(x^*) = 0, & j = 1, \dots, q \\ \mu_j^* \geq 0, & j = 1, \dots, q. \end{cases}$$

Si de plus le problème (P) est convexe, alors les conditions de KKT sont suffisantes pour que x^ soit un point de minimum global de f sur X .*

Remarque 3.3 *Attention ! Le signe des multiplicateurs associés aux contraintes d'inégalité peut changer si le problème n'est pas sous forme standard !*

Chapitre 4

Méthodes de descente en optimisation différentiable sans contrainte

Nous nous intéressons dans ce chapitre à la conception de méthodes numériques pour la résolution de problèmes d'optimisation différentiable sans contrainte. Autrement dit, le domaine X des contraintes est un ouvert de \mathbb{R}^n . Sans perte de généralité, nous supposons dans ce chapitre que : $X = \mathbb{R}^n$. Nous cherchons donc à résoudre le problème :

$$(P) \quad \min_{x \in \mathbb{R}^n} f(x).$$

où f est une fonction définie sur \mathbb{R}^n à valeurs réelles supposée différentiable, voire même deux fois différentiable. Les conditions nécessaires d'optimalité du premier et du second ordre expriment le fait qu'il n'est pas possible de "descendre" à partir d'un point de minimum (local ou global). Cette observation va servir de point de départ à l'élaboration des méthodes dites de *descente* étudiées dans ce chapitre.

4.1 Principe général des méthodes de descente

Partant d'un point x_0 arbitrairement choisi, un algorithme de descente va chercher à générer une suite d'itérés $(x_k)_{k \in \mathbb{N}}$ définie par :

$$x_{k+1} = x_k + s_k d_k$$

et telle que :

$$\forall k \in \mathbb{N}, \quad f(x_{k+1}) \leq f(x_k).$$

Un tel algorithme est ainsi déterminé par deux éléments : le choix de la direction d_k appelée direction de descente, et le choix de la taille du pas s_k à faire dans la direction d_k . Cette étape est appelée *recherche linéaire*.

Définition d'une direction de descente

Un vecteur $d \in \mathbb{R}^n$ est une direction de descente pour f à partir d'un point $x \in \mathbb{R}^n$ si $t \mapsto f(x + td)$ est décroissante en $t = 0$, c'est-à-dire s'il existe $\eta > 0$ tel que :

$$\forall t \in]0, \eta], f(x + td) < f(x). \quad (4.1)$$

Il est donc important d'analyser le comportement de la fonction f dans certaines directions. Lorsqu'elle existe, la dérivée directionnelle donne des informations sur la pente de la fonction dans la direction d .

Proposition 4.1 *Le vecteur $d \in \mathbb{R}^n$ est une direction de descente de f au point $x \in \mathbb{R}^n$ si $f'(x; d) < 0$.*

Dans le cas où f est différentiable, on obtient une définition plus pratique d'une direction de descente :

Proposition 4.2 *Supposons $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable. Un vecteur $d \in \mathbb{R}^n$ est une direction de descente de f au point x ssi :*

$$f'(x; d) = \nabla f(x)^\top d < 0. \quad (4.2)$$

De plus pour tout $\beta < 1$, il existe $\bar{\eta} > 0$ tel que :

$$\forall t \in]0, \bar{\eta}], f(x + td) < f(x) + t\beta \nabla f(x)^\top d < f(x). \quad (4.3)$$

Cette dernière inégalité garantit une décroissance minimum de la fonction f dans la direction d . Le schéma général d'un algorithme de descente est alors le suivant :

ALGORITHME DE DESCENTE MODÈLE.

Données: $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable, x_0 point initial arbitraire.

Sortie: une approximation de la solution du problème : $\min_{x \in \mathbb{R}^n} f(x)$.

1. $k := 0$
2. Tant que "test d'arrêt" non satisfait,
 - (a) Trouver une direction de descente d_k telle que : $\nabla f(x_k)^\top d_k < 0$.
 - (b) *Recherche linéaire* : Choisir un pas $s_k > 0$ à faire dans cette direction et tel que :

$$f(x_k + s_k d_k) < f(x_k).$$

- (c) Mise à jour : $x_{k+1} = x_k + s_k d_k$; $k := k + 1$;

3. Retourner x_k .
-

Choix de la direction de descente

Une fois la théorie bien maîtrisée, calculer une direction de descente est relativement simple. Dans le cas différentiable, il existe deux grandes stratégies de choix de direction de descente :

- la stratégie de Cauchy : $d_k = -\nabla f(x_k)$, conduisant aux *algorithmes de gradient* décrits au paragraphe 4.3.
- la stratégie de Newton : $d = -H[f](x_k)^{-1}\nabla f(x_k)$, conduisant aux *algorithmes Newtoniens* décrits au paragraphe 4.4.

Remarquons que si x_k est un point stationnaire ($\nabla f(x_k) = 0$) non optimal alors toutes ces directions sont nulles et aucun de ces algorithmes ne pourra progresser. Ce problème peut être résolu en utilisant des approches de type région de confiance qui ne seront pas étudiées dans le cadre de ce cours.

Critère d'arrêt

Soit x^* un minimum local du critère f à optimiser. Supposons que l'on choisisse comme test d'arrêt dans l'algorithme de descente modèle, le critère idéal : " $x_k = x^*$ ". Dans un monde idéal (i.e. en supposant tous les calculs exacts et la capacité de calcul illimitée), soit l'algorithme s'arrête après un nombre fini d'itérations, soit il construit (théoriquement) une suite infinie $x_0, x_1, \dots, x_k, \dots$ de points de \mathbb{R}^n qui converge vers x^* .

En pratique, un test d'arrêt devra être choisi pour garantir que l'algorithme s'arrête toujours après un nombre fini d'itérations et que le dernier point calculé soit suffisamment proche de x^* .

Soit $\varepsilon > 0$ la précision demandée. Plusieurs critères sont à notre disposition : tout d'abord (et c'est le plus naturel), un critère d'optimalité basé sur les conditions nécessaires d'optimalité du premier ordre présentées dans le chapitre 3 : en optimisation différentiable sans contrainte, on testera si

$$\|\nabla f(x_k)\| < \varepsilon, \quad (4.4)$$

auquel cas l'algorithme s'arrête et fournit l'itéré courant x_k comme solution.

En pratique, le test d'optimalité n'est pas toujours satisfait et on devra faire appel à d'autres critères (fondés sur l'expérience du numérique) :

- Stagnation de la solution : $\|x_{k+1} - x_k\| < \varepsilon(1 + \|x_k\|)$.
- Stagnation de la valeur courante : $\|f(x_{k+1}) - f(x_k)\| < \varepsilon(1 + |f(x_k)|)$.
- Nombre d'itérations dépassant un seuil fixé à l'avance : $k < \text{IterMax}$.

et généralement une combinaison de ces critères :

Critère d'arrêt = Test d'optimalité satisfait
 OU (Stagnation de la valeur courante & Stagnation de la solution)
 OU Nombre d'itérations maximum autorisé dépassé.

Remarque 4.1 *En pratique, on préférera travailler avec les erreurs relatives plutôt qu'avec les erreurs absolues, trop dépendantes de l'échelle.*

4.2 La recherche linéaire

Supposons pour l'instant résolu le problème du choix de la direction de descente et intéressons nous uniquement au calcul du pas : c'est la phase de recherche linéaire.

Soit $x \in \mathbb{R}^n$ un point de \mathbb{R}^n non critique et d une direction de descente de f en x . Nous cherchons à calculer un pas $s > 0$ de sorte que :

$$f(x + sd) < f(x).$$

Le choix de ce pas répond généralement à deux objectifs souvent contradictoires : trouver le meilleur pas possible et effectuer le moins de calculs possibles. Ces deux objectifs ont donné naissance à deux grandes familles : les algorithmes à pas fixe et ceux à pas optimal.

RECHERCHE LINÉAIRE : PAS FIXE. $s_k = s_{k-1}$

RECHERCHE LINÉAIRE : PAS OPTIMAL. s_k solution du problème $\min_{s>0} f(x_k + sd_k)$

Illustrées par les méthodes de descente de gradient, aucune de ces deux stratégies ne s'est révélée réellement convaincante : si la première peut être "risquée" du point de vue de la convergence, la seconde est souvent loin d'être triviale à mettre en oeuvre (sauf dans le cas quadratique) et généralement inutilement coûteuse : en effet, à quoi bon calculer très précisément un pas optimal dans une direction qui n'est peut-être pas la bonne ? (comme c'est par exemple le cas pour la méthode de plus profonde descente). Les recherches linéaires modernes reposent sur l'idée qu'un pas de descente acceptable est un pas qui fait "suffisamment" décroître la fonction objectif. Reste alors à définir les pas qui sont acceptables et ceux qui ne le sont pas.

On notera dans la suite $\varphi : s \in \mathbb{R} \mapsto f(x + sd)$ la fonction dite de *mérite* associée à f . La fonction f étant supposée au moins différentiable, φ est dérivable sur \mathbb{R} , de dérivée : $\varphi'(s) = \nabla f(x + sd)^\top d$ et :

$$\varphi'(0) = \nabla f(x)^\top d < 0. \quad (4.5)$$

4.2.1 Recherches linéaires élémentaires

Commençons par décrire quelques améliorations des recherches linéaires à pas fixe et à pas optimal. Le problème majeur des algorithmes à pas fixe est la convergence, en particulier si le pas est trop grand. Il est possible d'assurer la convergence par des stratégies de rebroussement ("backtracking" en anglais) de la façon suivante :

RECHERCHE LINÉAIRE AVEC REBROUSSEMENT.

$$\begin{aligned} s_k &= s_{k-1} \\ \text{Tant que } f(x_k + s_k d_k) &\geq f(x_k) : \\ s_k &= s_k/2 \end{aligned}$$

On peut également réduire le coût de la recherche linéaire à pas optimal en réduisant l'ensemble sur lequel la recherche linéaire est faite :

RECHERCHE LINÉAIRE PARTIELLE.

Données: s_{k-1} , $T = [1 \ a_1 \ a_2 \ \dots \ a_k]$ un tableau de réels > 0 contenant 1.

Sortie: s_k .

$$S = s_{k-1} * T = [s_{k-1} \ s_{k-1} * a_1 \ s_{k-1} * a_2 \ \dots \ s_{k-1} * a_k].$$

$$s_k = \arg \min_{s \in S} f(x_k + sd_k).$$

4.2.2 Condition de Armijo ou comment éviter les pas trop grands

Commençons par illustrer sur un exemple simple le fait que la condition de descente :

$$f(x_k + s_k d_k) < f(x_k)$$

n'est pas suffisante pour que le pas s_k soit considéré comme acceptable.

Exemple numérique

Soit $f : x \in \mathbb{R} \mapsto \frac{1}{2}x^2$ à minimiser sur \mathbb{R} . Appliquons une méthode de descente de gradient normalisée à pas variable, à partir du point $x_0 = 2$, définie par :

- (\mathcal{A}_1)
- Direction de descente normalisée : $d_k = -f'(x_k)/|f'(x_k)| = -\text{sgn}(x_k)$.
 - Pas de descente : $s_k = 2 + \frac{3}{2^{k+1}}$.

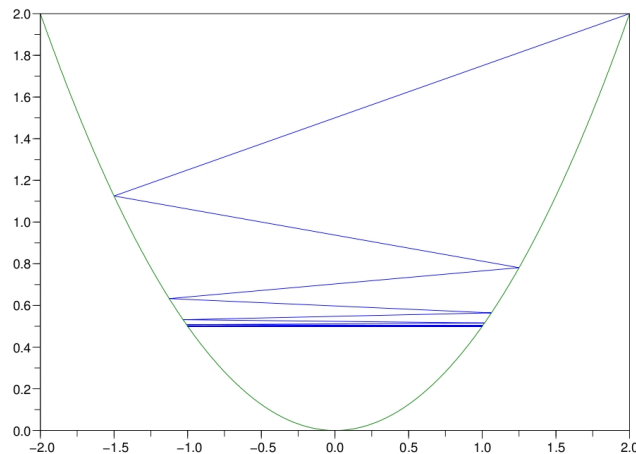


FIGURE 4.1 – 1000 premiers itérés $(f(x_k))_k$ générés par l'algorithme de gradient (\mathcal{A}_1).

Les résultats sont présentés sur la figure 4.1. On “voit” qu'il s'agit bien d'un algorithme de descente puisque la suite $(f(x_k))_k$ est décroissante, mais la convergence ne semble pas vérifiée.

Vérifions analytiquement ces observations numériques. On démontre par récurrence (en distinguant les cas k pair/ k impair) que :

$$x_k = (-1)^k \left(1 + \frac{1}{2^k}\right).$$

Pour tout $k \in \mathbb{N}$: $f(x_{k+1}) < f(x_k)$: on a donc bien un algorithme de descente mais la suite $(x_k)_{k \in \mathbb{N}}$ ne converge pas : elle possède deux points d'accumulation en $x = 1$ et $x = -1$ et aucun de ces deux points n'est un extremum de f .

L'échec de l'algorithme (\mathcal{A}_1) tient au fait que la notion de direction de descente est une notion locale, i.e. qui ne s'applique qu'au voisinage de l'itéré courant. Si jamais on choisit un pas de descente "trop grand", trouver un meilleur itéré sera uniquement le fait du hasard.

Condition de Armijo

Au lieu de chercher à minimiser φ , on préfère imposer des conditions moins restrictives (et moins coûteuses à mettre en oeuvre) : une première condition est dû à Armijo (1966) :

$$f(x + sd) \leq f(x) + \varepsilon s \left(\nabla f(x)^\top d \right), \quad 0 < \varepsilon < 1 \quad (4.6)$$

et est appelée condition de *Armijo* ou encore *condition de décroissance linéaire*.

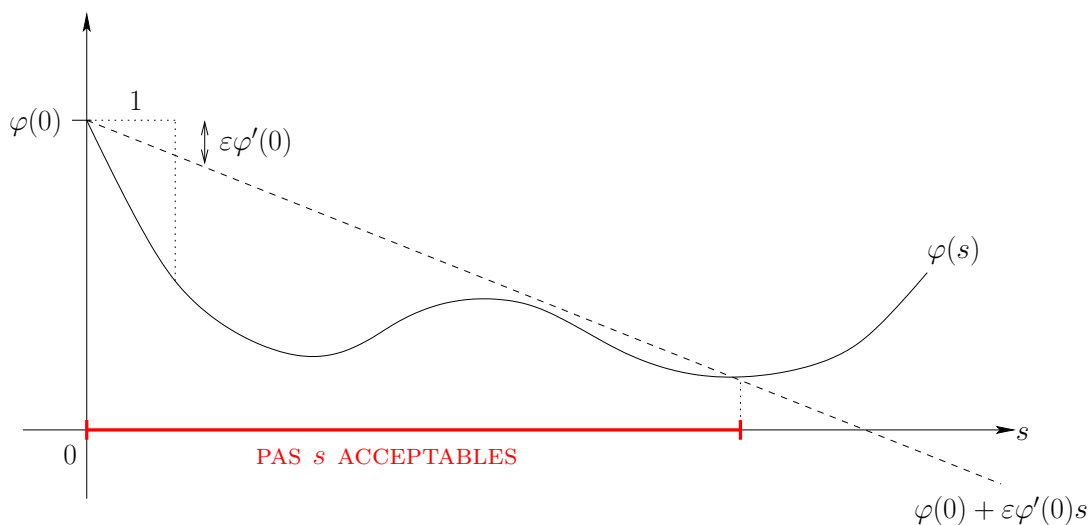


FIGURE 4.2 – Règle d'Armijo

Autrement dit, on demande à ce que f décroisse au moins autant que ce que ferait son modèle linéaire en x . En pratique la constante ε est choisie très petite de manière à satisfaire (4.6) le plus facilement possible ; typiquement : $\varepsilon = 10^{-4}$.

Remarque 4.2 D'après la définition (4.1) d'une direction de descente, il existe tout un intervalle de pas satisfaisant la condition d'Armijo (4.6).

4.2.3 Conditions de Wolfe ou comment éviter les pas trop petits

Reprenons le même exemple que précédemment et testons un autre algorithme de descente :

- (\mathcal{A}_2)
- Direction de descente normalisée : $d_k = -f'(x_k)/|f'(x_k)| = -\text{sgn}(x_k)$.
 - Pas de descente : $s_k = \frac{1}{2^{k+1}}$.

En effet, compte tenu de notre dernière tentative, on essaie cette fois-ci un pas très petit dans l'espoir de garantir la convergence de l'algorithme.

Commençons par tester numériquement l'algorithme (\mathcal{A}_2), cf figure 4.3. On "s'aperçoit" à nouveau qu'il s'agit bien d'un algorithme de descente puisque la suite $(f(x_k))_k$ est décroissante, mais les itérés semblent converger vers le point $x = 1$ qui n'est pas un extremum de f !

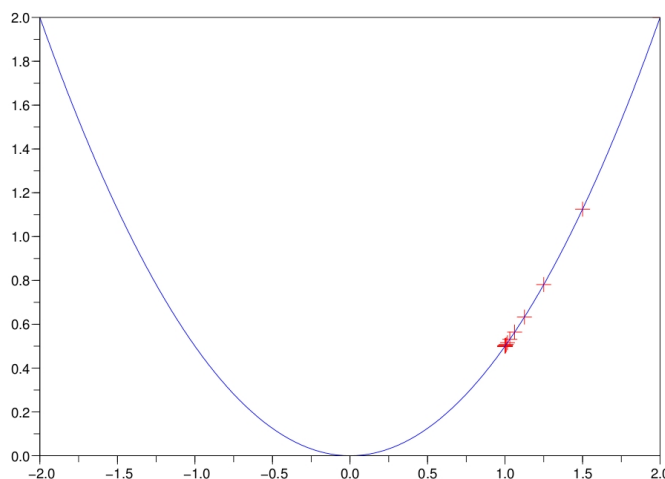


FIGURE 4.3 – 1000 premiers itérés $(f(x_k))_k$ générés par l'algorithme de gradient (\mathcal{A}_2).

Vérifions analytiquement ces observations numériques. Les itérés générés par l'algorithme (\mathcal{A}_2) à partir de $x_0 = 2$, sont tous positifs et vérifient :

$$\forall k \in \mathbb{N}, x_{k+1} = x_k - \frac{1}{2^{k+1}} \text{sgn}(x_k) = x_k - \frac{1}{2^{k+1}}.$$

On en déduit alors la suite des itérés est décroissante, d'où : $\forall k \in \mathbb{N}, f(x_{k+1}) < f(x_k)$. L'algorithme (\mathcal{A}_2) est donc bien un algorithme de descente. Calculons maintenant par récurrence l'expression explicite des x_k :

$$\begin{aligned} x_k &= x_{k-1} - \frac{1}{2^k} = x_{k-2} - \frac{1}{2^{k-1}} - \frac{1}{2^k} = \dots = x_0 - \sum_{i=1}^k \frac{1}{2^i} \\ &= 2 - \left(1 - \frac{1}{2^k}\right) = 1 + \frac{1}{2^k} \end{aligned}$$

ce qui prouve bien la convergence de la suite des itérés vers le point $x = 1$ qui n'est pas un extremum local de f . La raison de ce second échec réside dans le fait que les pas de descente deviennent arbitrairement proches de 0, ce qui fait que la méthode ne peut plus progresser.

En plus de la condition d'Armijo, il est donc important d'imposer une condition qui permettent d'éviter les pas trop petits. Remarquons que si on note s^* le pas optimal à faire en x dans la direction d , on a :

$$\nabla f(x)^\top d < 0 \quad \text{et} \quad \nabla f(x + s^*d)^\top d = 0.$$

La dérivée : $s \mapsto \nabla f(x + sd)^\top d$ augmente donc par rapport à sa valeur négative initiale. Afin d'assurer des pas suffisamment grands, l'idée va être de demander à la dérivée directionnelle d'augmenter suffisamment.

Ces deux conditions sont regroupées sous le nom de **conditions de Wolfe** et constituent les fondamentaux de la recherche linéaire moderne :

$$f(x + sd) \leq f(x) + \varepsilon_1 s (\nabla f(x)^\top d) \quad (4.7)$$

$$\nabla f(x + sd)^\top d \geq \varepsilon_2 (\nabla f(x)^\top d) \quad (4.8)$$

où les constantes ε_1 et ε_2 sont choisies telles que :

$$0 < \varepsilon_1 < \varepsilon_2 < 1.$$

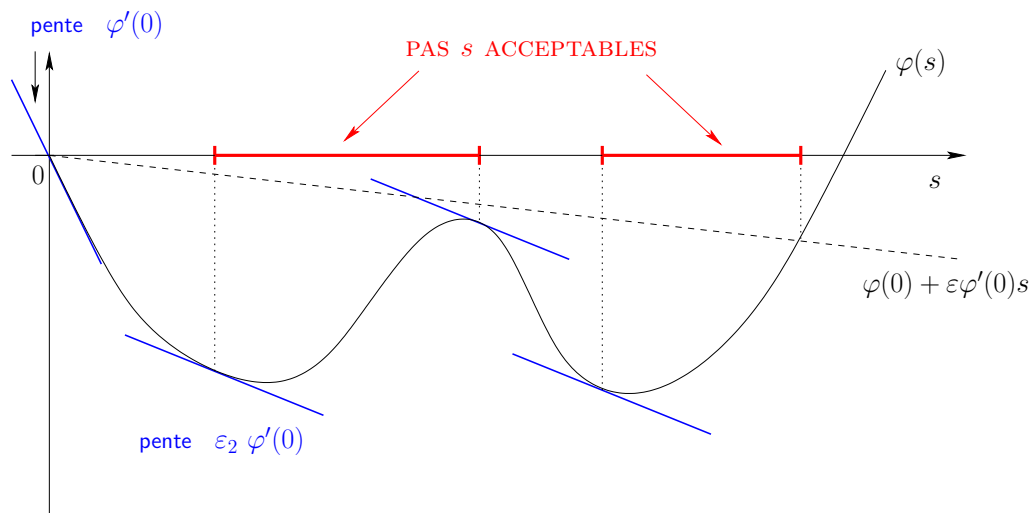


FIGURE 4.4 – Conditions de Wolfe

En pratique, on prend souvent : $\varepsilon_1 = 10^{-4}$ et $\varepsilon_2 = 0.99$. Un pas vérifiant ces deux conditions est appelé *pas de Wolfe*.

Remarque 4.3 La seconde condition de Wolfe n'est pas vérifiée pour $s = 0$, ce qui implique par la continuité supposée de la fonction mérite φ' , qu'elle n'est pas non plus vérifiée par de trop petits pas $s_k > 0$.

Proposition 4.3 (Validité des conditions de Wolfe) Soit d_k une direction de descente de f en x_k . Supposons la fonction mérite : $\varphi_k : s \in \mathbb{R} \mapsto f(x_k + sd_k)$ dérivable et bornée inférieurement. Alors il existe un pas $s_k > 0$ vérifiant les conditions de Wolfe (4.7) et (4.8).

Démonstration. Faite en cours. □

Le premier algorithme de recherche linéaire selon les conditions de Wolfe, attribué à Fletcher (1980) et Lemaréchal (1981), est décrit ci-après.

RECHERCHE LINÉAIRE DE WOLFE.

Données: f de classe C^1 , $x \in \mathbb{R}^n$ point courant, d direction de descente de f en x , s_0 première approximation de la solution, $\varepsilon_1 > 0$ et $\varepsilon_2 > 0$ tels que : $0 < \varepsilon_1 < \varepsilon_2 < 1$

Sortie: un pas s^* vérifiant les conditions de Wolfe.

1. $k := 0$; $s_- = 0$; $s_+ = +\infty$;

2. Tant que s_k ne vérifie pas les deux conditions de Wolfe,

(a) Si s_k ne vérifie pas (4.7), alors le pas est trop long et :

$$s_+ = s_k \quad \text{et} \quad s_{k+1} = \frac{s_- + s_+}{2}.$$

(b) Si s_k vérifie (4.7) mais pas (4.8), alors le pas est trop court et :

$$s_- = s_k \quad \text{et} \quad s_{k+1} = \begin{cases} \frac{s_- + s_+}{2} & \text{si } s_+ < +\infty \\ 2s_k & \text{sinon.} \end{cases}$$

(c) $k := k + 1$;

3. Retourner s_k .

On admet sans démonstration le résultat suivant :

Théorème 4.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction supposée différentiable. Soient $x_k \in \mathbb{R}^n$ et d_k une direction de descente de f en x_k . On suppose f bornée inférieurement dans la direction d_k . Alors l'algorithme de recherche linéaire de Wolfe se termine en un nombre fini d'itérations.

L'algorithme présenté ci dessus est très simple (interpolation par dichotomie) mais aussi très lent. La recherche linéaire étant exécutée à chaque itération de minimisation, il est essentiel d'accélérer le procédé. L'idée générale est de sélectionner des valeurs précédemment testées du pas s , puis d'interpoler la fonction mérite $\varphi : s \mapsto f(x + sd)$ par une fonction simple (un polynôme) passant par les points sélectionnés pour enfin calculer s^* qui minimise cette fonction. Cette recherche linéaire de Wolfe par interpolation cubique sera vue en TP.

4.2.4 Convergence globale des algorithmes avec pas de Wolfe

On s'intéresse maintenant à la convergence globale des algorithmes de descente avec recherche linéaire. Il s'agit de montrer que :

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

Ce résultat signifie que tout point d'accumulation \bar{x} de la suite $(x_k)_{k \in \mathbb{N}}$ est un point critique de f ($\nabla f(\bar{x}) = 0$) (le démontrer!).

Remarque 4.4 *La convergence globale des méthodes de descente avec recherche linéaire ne garantit que la convergence de sous-suites des itérés vers un point critique, alors que la convergence de la suite elle-même n'est pas garantie, cf [4, Proposition 1.2.1] ou [25, Theorem 3.2]. Ces résultats ont été obtenus pour les fonctions de classe $C^{1,1}$ ou C^2 . Récemment, P.A. Absil, R. Mahoney et B. Andrews [1] ont démontré la convergence des itérés générés par une méthode de descente, vers un point limite pour des fonctions analytiques.*

Principe de démonstration

Une technique classique en optimisation pour obtenir des résultats de convergence globale consiste à montrer que l'algorithme de descente considéré vérifie une inégalité du type :

$$f(x_k) - f(x_{k+1}) \geq c \|\nabla f(x_k)\|^2, \quad (4.9)$$

où $c > 0$ est une constante réelle. En sommant ces inégalités pour k variant de 0 à $N - 1$, on obtient :

$$\forall N \in \mathbb{N}, f(x_0) - f(x_N) \geq c \underbrace{\sum_{n=0}^{N-1} \|\nabla f(x_n)\|^2}_{\text{somme partielle d'une série à termes positifs}}.$$

Si f est bornée inférieurement, alors nécessairement $f(x_0) - f(x_N)$ est majorée et donc la somme partielle est majorée, et donc la série $\sum \|\nabla f(x_k)\|^2$ converge, ce qui implique :

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

Convergence globale des algorithmes de descente avec pas de Wolfe

On dispose toutefois d'un résultat général de convergence globale pour les algorithmes de descente avec pas de Wolfe. Ce résultat énoncé ci-après fait intervenir l'angle θ_k entre la direction de descente d_k et la direction du gradient défini par :

$$\cos(\theta_k) = \frac{\langle -\nabla f(x_k), d_k \rangle}{\|\nabla f(x_k)\| \|d_k\|}.$$

Théorème 4.2 (Théorème de Zoutendijk) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ supposée différentiable, de gradient Lipschitz et bornée inférieurement. Soit A un algorithme générant des itérés définis par :

$$x_{k+1} = x_k + s_k d_k,$$

où d_k est une direction de descente de f en x_k et $s_k > 0$ un pas vérifiant les conditions de Wolfe. Alors :

$$\sum \cos(\theta_k)^2 \|\nabla f(x_k)\|^2 \quad \text{converge.}$$

Démonstration. Faite en cours. □

Ce théorème implique que si la direction de descente d_k n'est pas orthogonale à la direction de $-\nabla f(x_k)$ à l'infini i.e. si :

$$\exists c > 0, \forall k \in \mathbb{N}, \cos(\theta_k) \geq c,$$

alors la série $\sum \|\nabla f(x_k)\|^2$ converge, et on obtient ainsi la convergence globale de l'algorithme de descente considéré.

4.3 Algorithmes de type gradient

Supposons la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable. Un algorithme de type gradient utilise comme direction de descente l'opposée de la direction du gradient de f au point courant :

$$x_0 \in \mathbb{R}^n \text{ quelconque, } \quad x_{k+1} = x_k - s_k \nabla f(x_k) \quad (4.10)$$

où le pas $s_k > 0$ est déterminé par l'une des stratégies de recherche linéaire présentées au paragraphe précédent. On rappelle que $d_k = -\nabla f(x_k)$ est bien une direction de descente de f en x_k , et que c'est la direction de "plus forte pente" de f en x_k .

Exercice 4.3.1 (Rappels de 3ème année) Démontrer que $d_k = -\nabla f(x_k)$ est bien une direction de descente de f en x_k , et que c'est la direction de "plus forte pente" de f en x_k .

Dans ce paragraphe nous nous intéressons aux conditions suffisantes de convergence de ce type d'algorithmes. On supposera pour cela que la fonction objectif f est bornée inférieurement, de classe C^1 à gradient Lipschitz de constante $L > 0$ sur \mathbb{R}^n .

4.3.1 CS pour qu'un algorithme de gradient soit une méthode de descente

On cherche une condition suffisante sur le pas s_k pour que l'algorithme (4.10) soit une méthode de descente, i.e. :

$$\forall k \in \mathbb{N}, f(x_{k+1}) < f(x_k).$$

Proposition 4.4 Soit f une fonction bornée inférieurement, de classe C^1 à gradient Lipschitz de constante $L > 0$. Si $s_k < \frac{2}{L}$, $\forall k$, alors l'algorithme :

$$x_0 \in \mathbb{R}^n \text{ quelconque, } \quad x_{k+1} = x_k - s_k \nabla f(x_k)$$

est bien un algorithme de descente i.e. : $\forall k \in \mathbb{N}$, $f(x_{k+1}) < f(x_k)$.

Démonstration. Faite en cours. □

4.3.2 Résultats de convergence globale

Supposons maintenant que le pas s_k vérifie bien : $\forall k \in \mathbb{N}$, $s_k < \frac{2}{L}$. On souhaite démontrer la convergence globale de l'algorithme (4.10) i.e. montrer que :

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

Théorème 4.3 Soit f une fonction bornée inférieurement, de classe C^1 à gradient Lipschitz de constante $L > 0$. Alors :

1. Si $\forall k$, $s_k := s < \frac{2}{L}$, l'algorithme de descente de gradient à pas constant converge globalement et :

$$0 < s \left(1 - \frac{L}{2}s\right) \|\nabla f(x_k)\|^2 \leq f(x_k) - f(x_{k+1}).$$

De plus, le pas constant assurant la meilleure décroissance possible est : $s_k = \frac{1}{L}$.

2. L'algorithme de gradient à pas optimal converge globalement et :

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

3. L'algorithme de gradient avec pas de Wolfe converge globalement.

Démonstration. En exercice. □

4.3.3 Tests numériques : pas fixe/pas optimal vs Pas de Wolfe

L'objet de ce paragraphe est de mettre en évidence la performance des méthodes de recherche linéaire modernes par rapport aux stratégies pas fixe et pas optimal. Pour cela, nous allons mettre en œuvre les algorithmes de gradient à pas fixe, à pas optimal et à pas de Wolfe sur l'exemple (quadratique) suivant :

$$\min_{(x,y) \in \mathbb{R}^2} f(x,y) = \frac{1}{2}x^2 + \frac{7}{2}y^2,$$

Sans trop d'effort, on vérifie que la fonction f est deux fois différentiable sur \mathbb{R}^2 , strictement convexe et qu'elle admet un unique point de minimum (global) en $(0, 0)$. De plus, le gradient ∇f est bien Lipschitzien de constante $L = 5\sqrt{2}$ (le vérifier!).

D'un point de vue numérique, soit $X_k = (x_k, y_k) \in \mathbb{R}^2$ l'itéré courant supposé non stationnaire (i.e. : $\nabla f(x_k, y_k) \neq 0$). La direction de recherche est donnée :

$$d_k = -\nabla f(X_k) = \begin{pmatrix} -x_k \\ -7y_k \end{pmatrix}.$$

Les algorithmes de gradient à pas fixe et à pas optimal sont définis de la façon suivante :

- **Descente de gradient à pas fixe.** On fixe $s_k = s$ et à chaque itération de l'algorithme on a :

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

Rappelons que, d'après la proposition 4.4, l'algorithme de gradient à pas fixe est un algorithme de descente si le pas s est choisi inférieur à $\frac{2}{L} \simeq 0.2828$. Le Tableau 4.1 illustre le comportement de cet algorithme pour différentes valeurs du pas s .

pas	0.325	0.25	0.125	0.05	0.01
Nb d'itérations	DV	49	101	263	1340

TABLE 4.1 – Nombres d'itérations de l'algorithme de gradient à pas fixe pour approcher le point de minimum global de f à 10^{-5} près, en fonction du pas - Point initial : $x_0 = (7, 1.5)$.

- **Descente de gradient à pas optimal.** A chaque itération, on calcule le pas optimal s_k solution de :

$$\min_{s>0} f(X_k + sd_k) = \min_{s>0} \frac{1}{2}x_k^2(1-s)^2 + \frac{7}{2}y_k^2(1-7s)^2.$$

dont la solution est :

$$s_k = \frac{x_k^2 + 7^2y_k^2}{x_k^2 + 7^3y_k^2}.$$

On retrouve sur la Figure 4.5 le comportement caractéristique des méthodes de gradient à pas fixe ou optimal, à savoir :

- La non-garantie de convergence pour l'algorithme de gradient à pas fixe.
- la lenteur de la méthode de plus profonde descente, caractérisée par le comportement en zigzag des itérés lorsque l'on se rapproche de la solution (deux directions de descente successives sont orthogonales).

En revanche, l'intérêt de la recherche linéaire de Wolfe apparaît clairement sur la Figure 4.6 : elle permet de forcer la convergence et d'accélérer la vitesse de convergence des itérés.

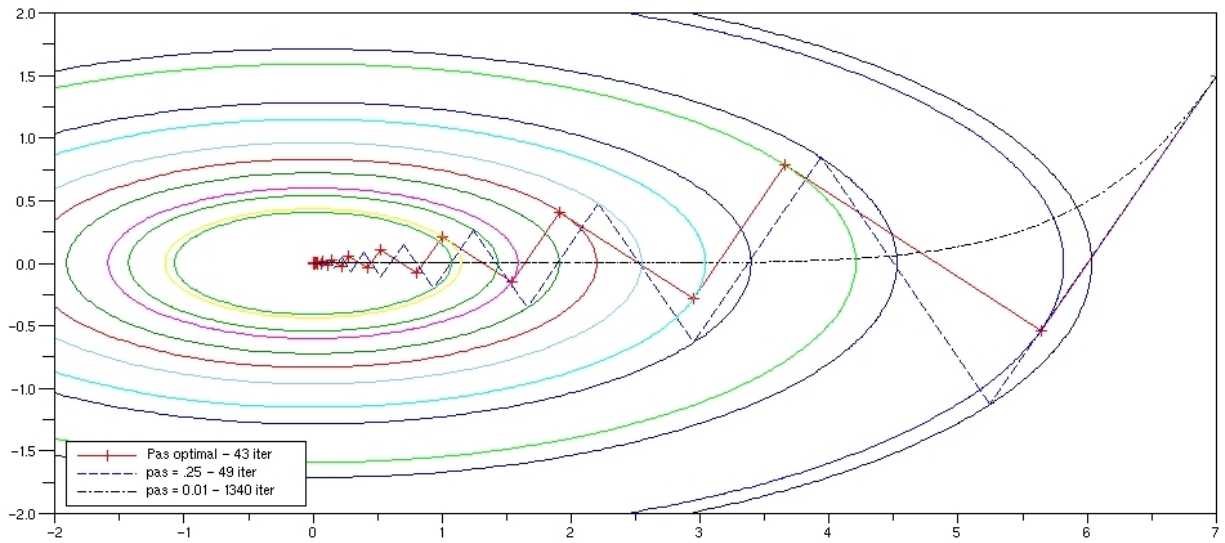


FIGURE 4.5 – Itérations des algos de gradient pas fixe et optimal, générées à partir du point $(7, 1.5)$.

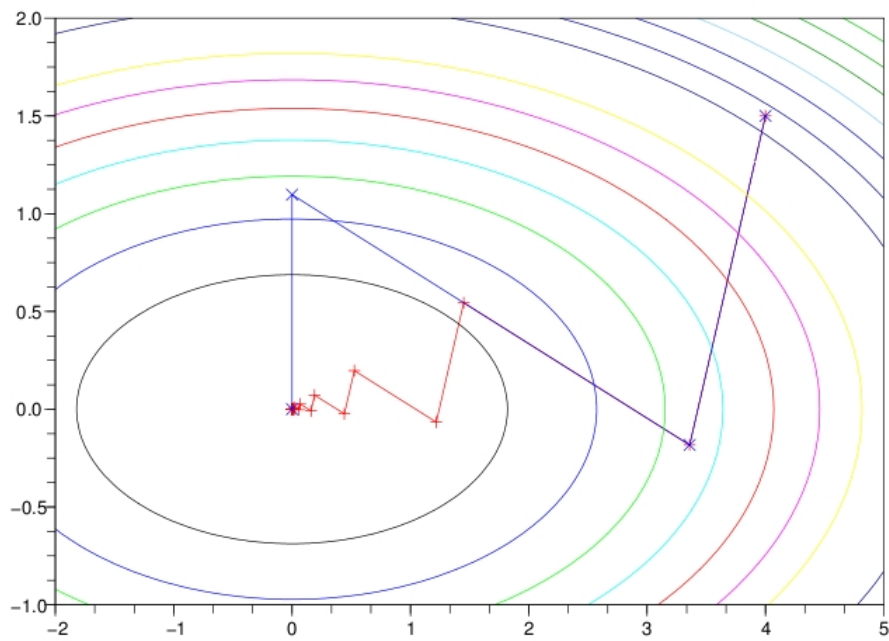


FIGURE 4.6 – 23 it. de l’algorithme de plus profonde descente (en rouge) vs 3 it. l’algorithme de gradient avec recherche linéaire de Wolfe (en bleu) à partir de $(x_0, y_0) = (4, 1.5)$.

4.4 Méthodes de type Newton

L'algorithme de Newton en optimisation est une application directe de l'algorithme de Newton pour la résolution d'équations du type : $F(x) = 0$. En optimisation sans contrainte, l'algorithme de Newton cherche les solutions de l'équation :

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

autrement dit, les points critiques de la fonction f à minimiser. En supposant f de classe C^2 et la matrice hessienne $H[f](x_k)$ inversible, une itération de l'algorithme de Newton s'écrit :

$$x_{k+1} = x_k - H[f](x_k)^{-1} \nabla f(x_k), \tag{4.11}$$

où $d_k = -H[f](x_k)^{-1} \nabla f(x_k)$ est appelée **direction de Newton**. La direction d_k est également l'unique solution du problème :

$$\arg \min_{d \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), d \rangle + \frac{1}{2} \langle H_f(x_k) d, d \rangle.$$

Autrement dit, d_k est le point de minimum global de l'approximation de second ordre de f au voisinage du point courant x_k .

A condition que la matrice $H[f](x_k)$ soit définie positive à chaque itération, la méthode de Newton est bien une méthode de descente à pas fixe égal à 1. Les propriétés remarquables de cet algorithme sont :

- ⊙ sa convergence quadratique (le nombre de décimales exactes est multiplié par 2 à chaque itération).
- ⊙ les difficultés et le coût de calcul de la hessienne $H[f](x_k)$: l'expression analytique des dérivées secondes est rarement disponible dans les applications.
- ⊙ le coût de résolution du système linéaire $H[f](x_k)(x_{k+1} - x_k) = \nabla f(x_k)$.
- ⊙ l'absence de convergence si le premier itéré est trop loin de la solution, ou si la hessienne est singulière.
- ⊙ Pas de distinction entre minima, maxima et points stationnaires.

La question que l'on se pose dans cette partie est donc : comment forcer la convergence globale de l'algorithme de Newton ? L'idée des méthodes de type Newton consiste à reprendre l'algorithme de Newton en remplaçant les itérations par :

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

où

- la matrice H_k est une approximation de la hessienne $H[f](x_k)$.
- $s_k > 0$ est le pas calculé par une recherche linéaire bien choisie.

Plusieurs questions se posent alors : comment déterminer une matrice H_k qui soit une "bonne" approximation de la hessienne à l'itération k sans utiliser les informations de second ordre et garantir que $-H_k^{-1} \nabla f(x_k)$ soit bien une direction de descente de f en x_k , sachant que la direction de Newton, si elle existe, n'en est pas nécessairement une ? Comment conserver les bonnes propriétés de l'algorithme de Newton ?

4.4.1 Méthode de Newton avec recherche linéaire

Considérons une itération de type Newton très générale :

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

On veut imposer à H_k d'être définie positive afin de garantir que $-H_k^{-1} \nabla f(x_k)$ est bien une direction de descente de f en x_k .

1. Si la matrice $H[f](x_k)$ est définie positive, alors :

$$H_k = H[f](x_k).$$

Si le pas de Newton ($s_k = 1$) est acceptable au sens des conditions de Wolfe, alors l'algorithme effectue une itération de Newton ; sinon il exécute une recherche linéaire de Wolfe initialisée par un pas égal à 1.

2. Sinon la technique la plus utilisée consiste à générer une matrice E telle que :

$$H_k = H[f](x_k) + E$$

soit définie positive. Remarquons que ceci est toujours possible si l'on choisit E de la forme αI .

Remarque 4.5 *Effectuer une itération de Newton dès que c'est possible, permet de bénéficier de la rapidité de convergence de cet algorithme.*

Pour terminer appliquons les algorithmes de Newton avec et sans recherche linéaire aux problèmes suivants :

$$(P_1) \quad \min_{(x,y) \in \mathbb{R}^2} f(x,y) = 100(y - x^2)^2 + (1 - x)^2 \quad \text{fonction de Rosenbrock.}$$

$$(P_2) \quad \min_{(x,y) \in \mathbb{R}^2} g(x,y) = \frac{1}{2}x^2 + x \cos y.$$

Le problème (P_1) admet un unique point critique au point $(1, 1)$ qui est un point de minimum global de f , tandis que le problème (P_2) admet une infinité de points critiques :

$$((-1)^{k+1}, k\pi), \quad k \in \mathbb{Z} \quad \text{points de minimum local de } g$$

$$(0, \frac{\pi}{2} + k\pi), \quad k \in \mathbb{Z} \quad \text{points selle de } g$$

Les résultats sont présentés sur les figures 4.7 et 4.8. Comme attendu, la supériorité de la méthode de Newton est évidente par rapport aux méthodes de gradient, que ce soit avec recherche linéaire ou non (voir figure 4.7). La figure 4.8 met également en évidence la globalisation de la méthode de Newton grâce à la recherche linéaire qui force la convergence vers un point de minimum local alors que la méthode de Newton seule se retrouve piégée dans un point selle de la fonction.

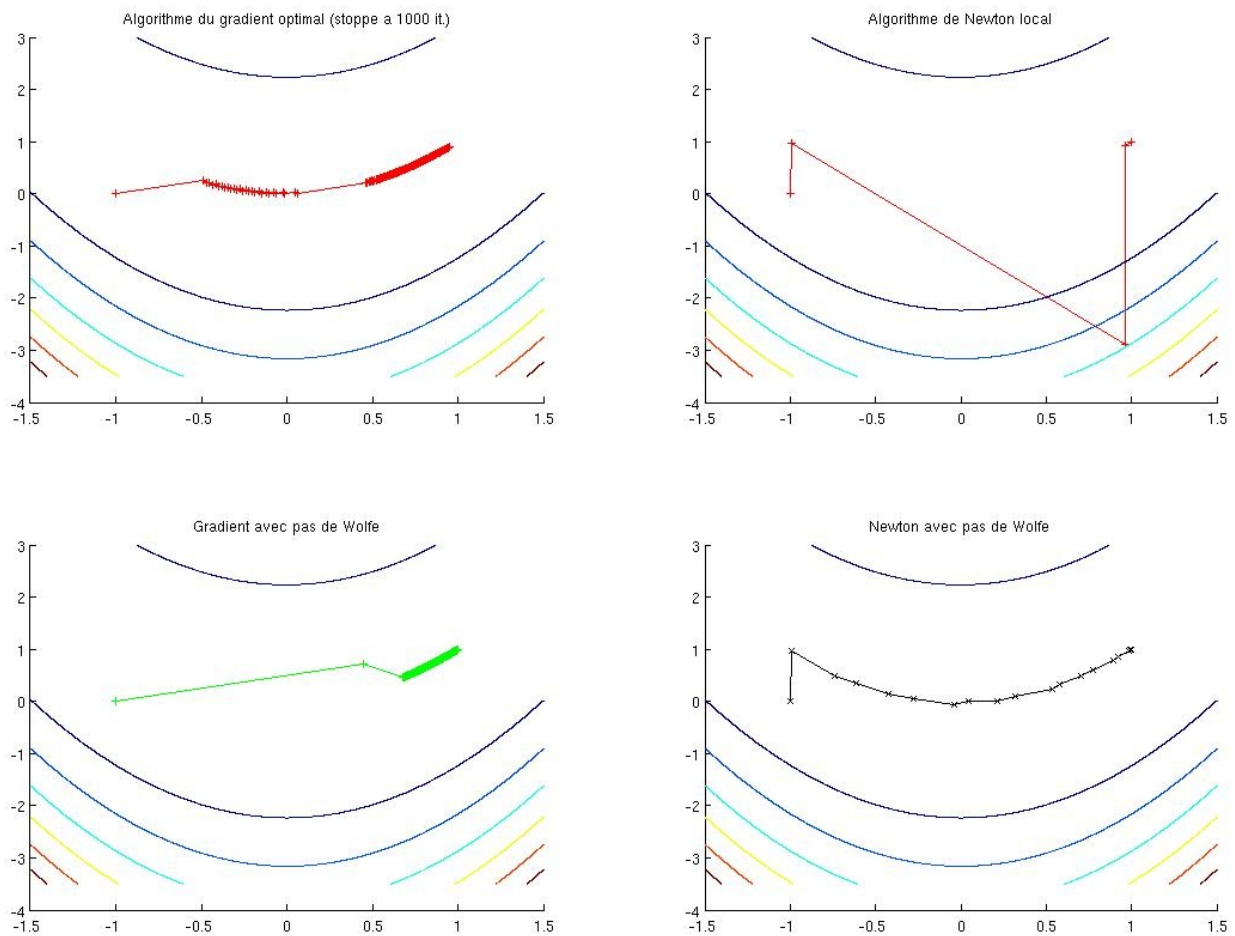


FIGURE 4.7 – Itérations des algorithmes de gradient optimal (stoppé à 10000 itérations!), de Newton (5 itérations), de gradient avec pas de Wolfe (8080 itérations) et de Newton avec pas de Wolfe (19 itérations) pour la minimisation de la fonction de Rosenbrock.

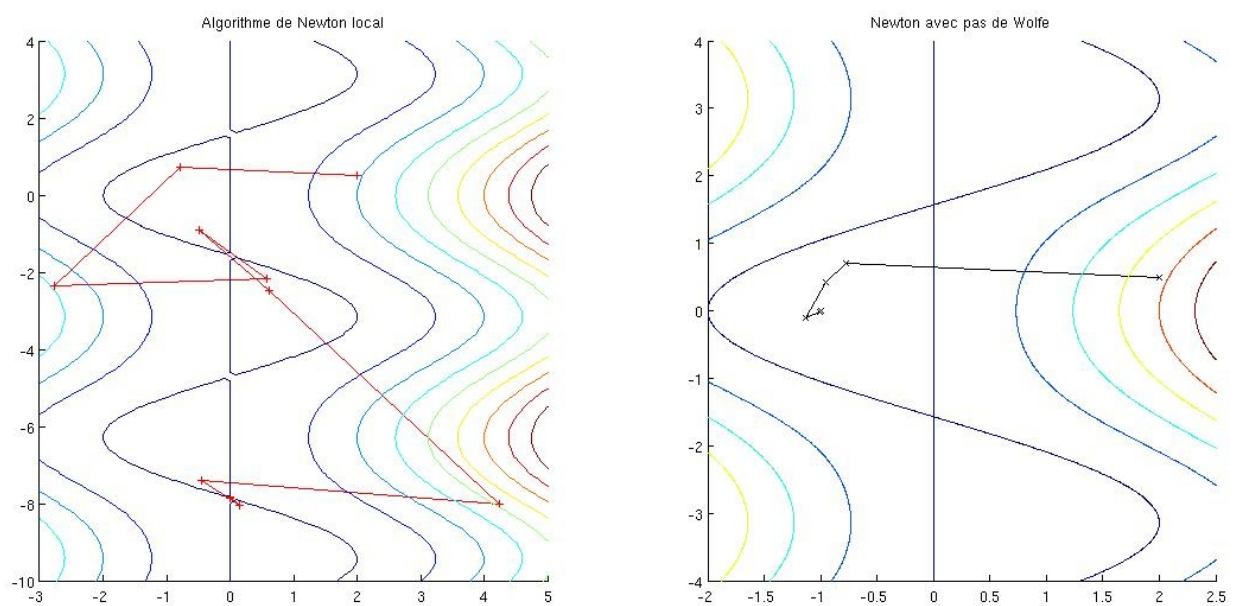


FIGURE 4.8 – Itérations des algorithmes de Newton (10 itérations) et de Newton avec recherche linéaire de Wolfe (6 itérations) pour la minimisation de la fonction $g : (x, y) \mapsto \frac{1}{2}x^2 + x \cos y$.

Remarque 4.6 *D'un point de vue algorithmique, on calcule en général une factorisation de Cholesky $L_k L_k^\top$ de $H[f](x_k) + \alpha I$ où le paramètre $\alpha > 0$ est augmenté jusqu'à ce que la factorisation soit rendue possible : on initialise α de la façon suivante :*

$$\alpha_0 = 0 \quad \text{si } H[f](x_k) \text{ est définie positive,} \quad = \frac{1}{2} \|H[f](x_k)\|_F^2 \quad \text{sinon.}$$

On résout ensuite :

$$L_k z_k = \nabla f(x_k) \quad \text{et} \quad L_k^\top d_k = -z_k.$$

L'avantage est que la hessienne au point courant devient définie positive, et on retrouve l'algorithme de Newton et la convergence quadratique de cette méthode.

Convergence globale de l'algorithme de Newton avec recherche linéaire

Supposons f de classe C^1 à gradient Lipschitz et bornée inférieurement. Le choix de la direction de descente dans cet algorithme est le suivant :

$$d_k = -H_k^{-1} \nabla f(x_k),$$

où H_k est une matrice symétrique définie positive. Couplée avec une recherche linéaire de Wolfe, nous sommes exactement dans le cadre d'application du théorème de Zoutendijk qui nous donne le résultat suivant :

Proposition 4.5 (CS de convergence globale) *Si le conditionnement de la matrice H_k reste borné au cours des itérations, i.e. si :*

$$\exists M > 0, \forall k \in \mathbb{N}, \kappa_2(H_k) = \|H_k\|_2 \|H_k^{-1}\|_2 \leq M,$$

alors l'algorithme de Newton avec pas de Wolfe converge globalement.

Démonstration. Faite en cours. □

4.4.2 Méthodes de quasi-Newton

Le couplage de la méthode de Newton avec la recherche linéaire de Wolfe a permis de construire une méthode globalement convergente. Les méthodes de quasi-Newton ont été développées pour pallier d'autres inconvénients de la méthode de Newton : en particulier le problème du calcul de la matrice hessienne qui n'est pas toujours possible ou conseillé. Ces méthodes se concentrent donc sur la construction itérative de matrices H_k approchant la hessienne, ou de matrices B_k approchant l'inverse de la hessienne.

Equation de sécante et approximation

Comment calculer une approximation H_{k+1} de la matrice hessienne $H[f](x_{k+1})$ connaissant x_k et x_{k+1} , $\nabla f(x_k)$ et $\nabla f(x_{k+1})$?

Ecrivons le développement limité de ∇f au voisinage de x_{k+1} et appliqué en x_k :

$$\nabla f(x_k) = \nabla f(x_{k+1}) + H[f](x_{k+1})(x_k - x_{k+1}) + o(x_k - x_{k+1}).$$

D'où :

$$H[f](x_{k+1})(x_k - x_{k+1}) \simeq \nabla f(x_{k+1}) - \nabla f(x_k).$$

On construit une approximation H_{k+1} de $H[f](x_{k+1})$ comme solution de l'équation :

$$\nabla f(x_{k+1}) - \nabla f(x_k) = H_{k+1}(x_{k+1} - x_k), \quad (4.12)$$

appelée **équation de sécante** ou **équation de quasi-Newton**. De façon similaire, on peut construire une approximation B_{k+1} de $H[f](x_{k+1})^{-1}$ comme solution de l'équation :

$$B_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k, \quad (4.13)$$

Dans les deux cas, les équations de quasi-Newton forment un système sous-déterminé à n équations et n^2 inconnues. Il existe donc une infinité de matrices H_{k+1} pouvant convenir.

Mises à jour DFP et BFGS

L'idée proposée par Broyden (1965) est de choisir parmi ce nombre infini de modèles linéaires celui qui est le plus proche du modèle établi à l'itération précédente, conservant ainsi le plus possible ce qui a déjà été calculé. On cherche donc à résoudre :

$$\begin{aligned} \min_H \quad & \frac{1}{2} \|H - H_k\|_F^2 \\ \text{s.t.} \quad & \nabla f(x_{k+1}) - \nabla f(x_k) = H(x_{k+1} - x_k) \end{aligned}$$

où $\|\cdot\|_F$ désigne la norme de Frobenius. Il s'agit là d'un problème quadratique strictement convexe dont l'unique solution est donnée par :

$$H_{k+1} = H_k + \frac{(y_k - H_k \sigma_k) \sigma_k^\top}{\sigma_k^\top \sigma_k}, \quad \text{où : } \begin{cases} \sigma_k &= x_{k+1} - x_k \\ y_k &= \nabla f(x_{k+1}) - \nabla f(x_k), \end{cases}$$

Le problème de cette méthode est que la matrice H_k ainsi construite, n'est en général pas symétrique, ni définie positive. La hessienne d'une fonction de classe C^2 étant symétrique, il est naturel d'imposer à son approximation de l'être également. Nous cherchons donc H_{k+1} comme solution du problème :

$$(P) \quad \begin{aligned} \min_H \quad & \frac{1}{2} \|H - H_k\|^2 \\ \text{s.t.} \quad & y_k = H \sigma_k, \quad H^\top = H. \end{aligned}$$

De nombreuses normes matricielles peuvent être utilisées, et conduisent à des méthodes de quasi-Newton différentes. Les méthodes les plus utilisées aujourd'hui sont les méthodes DFP et BFGS obtenues en choisissant comme des normes de la forme :

$$\|A\|_W = \|W^{\frac{1}{2}} A W^{\frac{1}{2}}\|_F$$

où W est une matrice symétrique inversible vérifiant : $y_k = W\sigma_k$. La résolution (longue et technique) du problème (P) conduit alors aux formules de mise à jour suivantes :

MÉTHODE DFP (DAVIDSON, FLETCHER, POWELL. 1959-63)

$$H_{k+1} = H_k + \frac{y_k y_k^\top}{y_k^\top \sigma_k} - \frac{H_k \sigma_k \sigma_k^\top H_k}{\sigma_k^\top H_k \sigma_k}.$$

MÉTHODE BFGS (BROYDEN, FLETCHER, GOLDFARB, SHANNON. 1969-70)

$$B_{k+1} = \left(I - \frac{\sigma_k y_k^\top}{y_k^\top \sigma_k} \right)^\top B_k \left(I - \frac{\sigma_k y_k^\top}{y_k^\top \sigma_k} \right) + \frac{\sigma_k \sigma_k^\top}{y_k^\top \sigma_k}.$$

Ces deux formules sont duales l'une de l'autre dans le sens où l'on obtient la formule BFGS en inversant la relation DFP avec : $H_k = B_k^{-1}$. On a alors également : $H_{k+1} = B_{k+1}^{-1}$.

Exercice 4.4.1 Vérifier que les matrices H_{k+1} et B_{k+1} sont bien symétriques et satisfont leurs équations de quasi-Newton respectives :

Il reste donc maintenant à vérifier que les directions de quasi-Newton ainsi définies sont bien des directions de descente.

Proposition 4.6 Si B_k est définie positive et $y_k^\top \sigma_k > 0$, alors B_{k+1} est définie positive.

Démonstration. Supposons B_k définie positive ; soit $x \in \mathbb{R}^n$ quelconque.

$$\begin{aligned} x^\top B_{k+1} x &= x^\top \left(I - \frac{\sigma_k y_k^\top}{y_k^\top \sigma_k} \right)^\top B_k \left(I - \frac{\sigma_k y_k^\top}{y_k^\top \sigma_k} \right) x + \frac{x^\top \sigma_k \sigma_k^\top x}{y_k^\top \sigma_k} \\ &= u^\top B_k u + \frac{(\sigma_k^\top x)^2}{y_k^\top \sigma_k} \end{aligned}$$

en posant : $u = \left(I - \frac{\sigma_k y_k^\top}{y_k^\top \sigma_k} \right) x$. A condition que :

$$y_k^\top \sigma_k > 0,$$

on a donc : $x^\top B_{k+1} x \geq 0$ pour tout $x \in \mathbb{R}^n$, soit B_{k+1} semidéfinie positive. Il reste à montrer que $x^\top B_{k+1} x$ ne s'annule que pour $x = 0$:

$$\begin{aligned} x^\top B_{k+1} x = 0 &\Rightarrow u^\top B_k u + \frac{(\sigma_k^\top x)^2}{y_k^\top \sigma_k} = 0 \\ &\Rightarrow u^\top B_k u = 0 \quad \text{et} \quad \sigma_k^\top x = 0 \Rightarrow u = 0 \quad \text{et} \quad \sigma_k^\top x = 0 \end{aligned}$$

puisque B_k est supposée définie positive. Or : $u = \left(I - \frac{\sigma_k y_k^\top}{y_k^\top \sigma_k} \right) x$. Donc :

$$x^\top u = 0 \quad \text{et} \quad x^\top u = x^\top x - \underbrace{x^\top \sigma_k}_{=0} \frac{y_k^\top}{y_k^\top \sigma_k} = x^\top x.$$

On en déduit : $x^\top x = \|x\|^2 = 0$, d'où : $x = 0$. □

4.4.3 Propriétés et convergence de l'algorithme BFGS

D'après ce qui précède, l'algorithme BFGS sera un algorithme de descente à condition de l'initialiser avec une matrice H_0 symétrique définie positive et en maintenant au fil des itérations la condition :

$$(\nabla f(x_{k+1}) - \nabla f(x_k))^\top (x_{k+1} - x_k) > 0.$$

Remarquons que cette inégalité est satisfaite dès lors qu'on utilise la recherche linéaire de Wolfe pour le calcul du pas (le démontrer!).

ALGORITHME BFGS.

Données: $f : \mathbb{R}^n \rightarrow \mathbb{R}$, x_0 point initial et H_0 matrice symétrique définie positive arbitraires.

Sortie: une approximation de la solution du problème : $\min_{x \in \mathbb{R}^n} f(x)$.

1. $k := 0$

2. Tant que "test d'arrêt" non satisfait,

(a) Choix de la direction : $d_k = -H_k^{-1} \nabla f(x_k)$.

(b) Recherche linéaire de Wolfe pour calculer un pas $s_k > 0$.

(c) Nouvel itéré :

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

(d) Mise à jour de la matrice H_k en H_{k+1} par la formule BFGS ; $k := k + 1$;

3. Retourner x_k .

Théorème 4.4 *Supposons f convexe, de gradient lipschitzien sur le domaine $\{x \in \mathbb{R}^n; f(x) \leq f(x_0)\}$. Alors l'algorithme BFGS vérifie :*

— soit f est non bornée inférieurement,

— soit $\liminf_{k \rightarrow \infty} \nabla f(x_k) = 0$.

Cet algorithme est considéré de nos jours comme le plus performant. Bien qu'il n'existe pas de résultat de convergence de cet algorithme dans le cas non convexe, rien n'interdit son utilisation : les itérations restent bien définies et les résultats numériques sont souvent remarquables.

4.5 Algorithmes pour la régression non-linéaire

Terminons ce chapitre par le cas particulier des problèmes de moindres carrés rencontrés fréquemment en pratique dans le cadre de l'identification de paramètres en fouille de données, en problèmes inverses ou encore en analyse statistique. Un problème de moindres carrés est un problème d'optimisation de la forme :

$$\min_{x \in \mathbb{R}^n} r(x) := \frac{1}{2} \|F(x)\|_2^2, \quad (4.14)$$

où F désigne une application de \mathbb{R}^n dans \mathbb{R}^m , en supposant $m \geq n$. Les variables d'optimisation x_i sont les paramètres du modèle physique non linéaire proposé. On effectue m mesures et on cherche les x_i qui permettent d'ajuster au mieux le modèle aux mesures. Remarquons que si le système $F(x) = 0$ a des solutions, alors ce sont également les solutions du problème (4.14).

4.5.1 Problèmes de moindres carrés linéaires (rappels)

On se place dans le cas où la fonction F est linéaire, i.e. :

$$F(x) = Ax - b, \quad \text{avec} \quad A \in M_{n,p}(\mathbb{R}),$$

et on résout donc le problème de *moindres carrés linéaire* suivant :

$$\min_{x \in \mathbb{R}^n} r(x) := \frac{1}{2} \|Ax - b\|_2^2 \quad (4.15)$$

On calcule la jacobienne de F en tout point : $J_F(x) = A$, puis on en déduit :

$$\nabla r(x) = A^\top (Ax - b) \quad \text{et} \quad H_r(x) = J_F(x)^\top J_F(x) = A^\top A.$$

La hessienne de r est positive en tout point, le problème de moindres carrés linéaire est donc convexe. Cherchons les points critiques du problème (4.15) i.e. vérifiant : $\nabla r(x) = 0$, d'où le système suivant à résoudre :

$$A^\top Ax = A^\top b \quad \text{EQUATIONS NORMALES}$$

appelé *système d'équations normales* du problème de moindres carrés linéaire (4.15). Ainsi :

$$x^* \text{ est solution du problème (4.15) si et seulement si : } A^\top Ax^* = A^\top b.$$

De plus si A est de rang plein, alors la fonction r est strictement convexe et x^* est l'unique solution de (4.15).

Appliquons maintenant la méthode de Newton au problème (4.15) : la direction de Newton $d_k = x_{k+1} - x_k$ est solution du système :

$$A^\top A d_k = -A^\top (Ax_k - b), \quad \text{d'où : } A^\top A x_{k+1} = A^\top b,$$

et ceci quel que soit x_k . On reconnaît ici le système d'équations normales du problème (4.15). Ceci signifie donc que la méthode de Gauss-Newton identifie la solution en une seule itération lorsque la fonction F est linéaire.

Exercice 4.5.1 Soit $F : \mathbb{R}^3 \rightarrow \mathbb{R}^n$ une application définie par :

$$F_i(x_0, x_1, x_2) = c_i - x_0 - x_1 e^{-x_2 t_i}, \quad i = 1, \dots, n.$$

1. Calculer ∇F_i et $H[F_i]$.
2. En déduire : J_r et H_r où $r(x) = \frac{1}{2}F(x)^\top F(x)$.

4.5.2 Algorithme de Gauss-Newton

Revenons au problème de moindres carrés non linéaire général (4.14) :

$$\min_{x \in \mathbb{R}^n} r(x) = \frac{1}{2} \|F(x)\|_2^2.$$

Il existe deux façons de présenter l'algorithme de Newton : l'une (vue en cours en 3MIC) consiste à appliquer la méthode de Newton au problème (4.14) en négligeant le terme de second ordre $\sum_{i=1}^m F_i(x)H_{F_i}(x)$ dans la hessienne de r . En effet, le terme $J_F(x)^\top J_F(x)$ est souvent prépondérant par rapport au second terme à cause de la presque linéarité du modèle au voisinage de la solution (i.e. $H_{F_i}(x)$ petit), et/ou du résidu F_j petit au voisinage de la solution. Nous ne reviendrons pas sur cette approche, nous nous intéressons ici à une autre : l'idée est de remplacer à chaque itération le problème de moindres carrés non linéaire par un problème approché de moindres carrés linéaires.

Soit $x_k \in \mathbb{R}^n$ le point courant. Remplaçons au voisinage de x_k le problème (4.14) par :

$$\min_{y \in \mathbb{R}^n} \tilde{r}(y) = \frac{1}{2} \|F(x_k) + J_F(x_k)(y - x_k)\|_2^2, \quad (4.16)$$

où l'on a remplacé F par une approximation du premier ordre en x_k . Le problème (4.16) est un problème de moindres carrés linéaire dont les solutions x_{k+1} vérifient les équations normales associées, à savoir :

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

La direction $d_k = x_{k+1} - x_k$ ainsi définie, est appelée direction de Gauss-Newton.

ALGORITHME DE GAUSS-NEWTON.

Données: F fonction différentiable, x_0 point initial, $\varepsilon > 0$ précision demandée.

Sortie: une approximation de la solution du problème de moindres carrés :

$$\min_{x \in \mathbb{R}^n} r(x) = \frac{1}{2} F(x)^\top F(x).$$

1. $k := 0$;
2. Tant que *critère d'arrêt à définir*,
 - (a) *Calcul d'une direction de recherche* : calculer d_k solution de :

$$J_F(x_k)^\top J_F(x_k) d = -J_F(x_k)^\top F(x_k).$$

- (b) $x_{k+1} = x_k + d_k$;
 - (c) $k := k + 1$;
 3. Retourner x_k .
-

Les avantages de cette méthode par rapport à la méthode de Newton sont les suivants :

1. L'approximation faite ne nécessite pas le calcul des dérivées secondes.
2. Dans de nombreuses applications, cette approximation est de bonne qualité.
3. Si $J_F(x_k)$ est de rang plein et si $\nabla r(x_k)$ est non nul, alors la direction de Gauss-Newton est une direction de descente de r en x_k et donc adaptée pour l'utilisation d'un algorithme de recherche linéaire.

La vitesse de convergence de l'algorithme de Gauss-Newton dépend de la qualité de l'approximation faite du problème initial. Dans le cas où le résidu est effectivement petit au voisinage de la solution, la convergence est rapide. Si la hessienne de F est nulle à la solution, alors la convergence est quadratique.

Sous des hypothèses supplémentaires (par exemple dans [25], valeurs singulières de $J_F(x_k)$ uniformément bornées hors d'un voisinage de l'origine et pas de Wolfe), on peut démontrer la convergence globale des itérés vers un point stationnaire de r . Un désavantage de la méthode de Gauss-Newton est que l'on n'a pas de résultat de convergence lorsque la jacobienne de F en x_k n'est pas de rang plein. Pour pallier cet inconvénient, on a recours à la méthode de Levenberg-Marquardt.

4.5.3 Algorithme de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt peut être vu comme une régularisation de l'algorithme de Gauss-Newton, en particulier lorsque la jacobienne de F n'est pas de rang plein. Toutefois il existe un lien étroit avec les méthodes dites de *région de confiance* et c'est par cet aspect là que nous abordons ici la description de cet algorithme.

Soit x_k l'itéré courant. Comme pour la méthode de Gauss-Newton, on remplace, au voisinage de x_k , le problème (4.14) de moindres carrés non linéaires par le problème suivant :

$$\min_{y \in \mathbb{R}^n} \frac{1}{2} \|F(x_k) + J_F(x_k)(y - x_k)\|_2^2,$$

mais en ajoutant cette fois-ci la contrainte : $\|y - x_k\|_2 \leq \Delta_k$. Le paramètre $\Delta_k > 0$ est appelé rayon de la région de confiance. A chaque itération, le problème initial est remplacé par le problème quadratique suivant :

$$(P_k) \quad \min_{y \in \mathbb{R}^n} \quad \frac{1}{2} \|F(x_k) + J_F(x_k)(y - x_k)\|_2^2 \\ \text{sous :} \quad \|y - x_k\|_2 \leq \Delta_k.$$

En introduisant le Lagrangien associé au problème (P_k) :

$$L_k(y; \lambda) = \frac{1}{2} \|F(x_k) + J_F(x_k)(y - x_k)\|_2^2 + \frac{1}{2} \lambda (\|y - x_k\|_2^2 - \Delta_k^2), \quad \lambda \in \mathbb{R}$$

on peut écrire les conditions nécessaires d'optimalité du premier ordre, à savoir :

$$(J_F(x_k)^\top J_F(x_k) + \lambda I)d = -J_F(x_k)^\top F(x_k) \\ \lambda = 0 \quad \text{ou} \quad \|y - x_k\| = \Delta_k \\ \lambda \geq 0$$

Dans le cas où la contrainte de région de confiance est inactive (i.e. $\|y - x_k\| < \Delta_k$ et $\lambda = 0$), on retrouve une itération de l'algorithme de Gauss-Newton. Dans le cas contraire ($\|y - x_k\| = \Delta_k$), la solution est sur le bord de la région de confiance. Le pas effectué peut être vu comme une régularisation du pas de Gauss-Newton dans le sens où la matrice $(J_F(x_k)^\top J_F(x_k) + \lambda I)$ est maintenant définie positive : le pas effectué est alors un pas de descente de la fonction r en x_k .

ALGORITHME DE LEVENBERG-MARQUARDT.

Données: F fonction différentiable, x_0 point initial, $\varepsilon > 0$ précision demandée.

Sortie: une approximation de la solution du problème de moindres carrés :

$$\min_{x \in \mathbb{R}^n} r(x) = \frac{1}{2} F(x)^\top F(x).$$

1. $k := 0$;
2. Tant que *critère d'arrêt à définir*,
 - (a) *Calcul d'une direction de recherche* : calculer d_k solution de :

$$(J_F(x_k)^\top J_F(x_k) + \lambda I) d = -J_F(x_k)^\top F(x_k).$$
 - (b) $x_{k+1} = x_k + d_k$;
 - (c) Mise à jour du paramètre λ .
 - (d) $k := k + 1$;
3. Retourner x_k .

Le paramètre $\lambda \geq 0$ peut être choisi fixe ou ajusté de manière heuristique : augmenté ou diminué d'un facteur suivant la qualité du pas proposé.

Chapitre 5

Algorithmes pour l'optimisation différentiable sous contrainte

Dans ce chapitre, nous nous intéressons au problème d'optimisation suivant :

$$(P) \quad \begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t. :} \quad & g_j(x) \leq 0, \quad j = 1, \dots, q \\ & h_i(x) = 0, \quad i = 1, \dots, p. \end{aligned}$$

où les fonctions f , g_j et h_i sont supposées à valeurs réelles et de classe C^1 .

Trouver une solution d'un problème d'optimisation sous contraintes fonctionnelles consiste à déterminer un point optimal x^* et des multiplicateurs associés (λ^*, μ^*) . Deux grandes familles de méthodes peuvent être définies pour la résolution des problèmes d'optimisation sous contraintes : les méthodes primales et les méthodes duales. Les approches primales se concentrent sur la détermination du point x^* , les multiplicateurs (λ, μ) ne servant souvent qu'à vérifier l'optimalité de x^* . Les méthodes duales quant à elles mettent l'accent sur la recherche d'un multiplicateur en travaillant sur un problème d'optimisation déduit du problème initial par *dualité*.

5.1 Introduction à la dualité lagrangienne

On définit le *Lagrangien* (ou fonction lagrangienne) associé à (P) et défini par :

$$\begin{aligned} L : \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q &\rightarrow \mathbb{R} \\ (x, \lambda, \mu) &\mapsto L(x; \lambda, \mu) = f(x) + \sum_{i=1}^p \lambda_i h_i(x) + \sum_{j=1}^q \mu_j g_j(x), \\ &= f(x) + \langle \lambda, h(x) \rangle_{\mathbb{R}^p} + \langle \mu, g(x) \rangle_{\mathbb{R}^q}. \end{aligned}$$

où l'on note λ et μ les vecteurs de coordonnées respectives $(\lambda_1, \dots, \lambda_p)$ et (μ_1, \dots, μ_q) .

5.1.1 Problème primal - Problème dual

Commençons par observer que le problème primal (P) peut se réécrire à l'aide du Lagrangien de la façon suivante :

$$\inf_{x \in X} f(x) = \inf_{x \in \mathbb{R}^n} \left(\sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} L(x; \lambda, \mu) \right) \quad \text{PROBLÈME PRIMAL}$$

On définit alors :

- la fonction primale $\bar{f} : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ associée au problème (P) :

$$\bar{f}(x) = \sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} L(x; \lambda, \mu) = \begin{cases} f(x) & \text{si } x \in X \\ +\infty & \text{sinon.} \end{cases}$$

- le *domaine admissible primal* :

$$\{x \in \mathbb{R}^n : \bar{f}(x) < +\infty\} = X$$

sur lequel la fonction primale est à valeurs finies.

Le problème dual du problème (P) est obtenu en échangeant l'inf et le sup :

$$\sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} \left(\inf_{x \in \mathbb{R}^n} L(x; \lambda, \mu) \right) = \sup_{(\lambda, \mu) \in X^*} f^*(\lambda, \mu) \quad \text{PROBLÈME DUAL}$$

où l'on définit :

- la fonction duale du problème (P) :

$$f^*(\lambda, \mu) = \inf_{x \in \mathbb{R}^n} L(x; \lambda, \mu).$$

- le *domaine admissible dual* :

$$X^* = \left\{ (\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q : \inf_{x \in \mathbb{R}^n} L(x; \lambda, \mu) > -\infty \right\},$$

sur lequel la fonction duale est à valeurs finies.

La question de la dualité est savoir si l'on change les solutions du problème primal en échangeant l'inf et le sup. D'un point de vue théorique, le problème dual est beaucoup plus simple à résoudre car il s'agit, sous sa forme standard, d'un problème d'optimisation convexe. En effet :

Proposition 5.1 *La fonction duale f^* est toujours concave et le domaine admissible dual X^* convexe.*

Exercice 5.1.1 *Le démontrer!*

5.1.2 Liens entre problème primal et problème dual

Principe de dualité faible.

$$\sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} \left(\inf_{x \in \mathbb{R}^n} L(x; \lambda, \mu) \right) \leq \inf_{x \in \mathbb{R}^n} \left(\sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} L(x; \lambda, \mu) \right).$$

Dans le cadre de la dualité Lagrangienne, on appelle *saut de dualité* la différence positive entre les deux quantités précédentes, à savoir :

$$\inf_{x \in \mathbb{R}^n} \left(\sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} L(x; \lambda, \mu) \right) - \sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} \left(\inf_{x \in \mathbb{R}^n} L(x; \lambda, \mu) \right).$$

Si le saut de dualité est non nul, cela signifie que les solutions des problèmes primal et dual n'ont *a priori* rien à voir entre elles. En l'absence de saut de dualité, les problèmes sont équivalents (même ensemble de solutions) et l'existence de solutions primale et duale dans ce cas est étroitement liée à l'existence de points selle du Lagrangien.

Définition 5.1 (Point-selle du Lagrangien) Soit $\bar{x} \in \mathbb{R}^n$ et $(\bar{\lambda}, \bar{\mu}) \in \mathbb{R}^p \times (\mathbb{R}_+)^q$. Le point $(\bar{x}, \bar{\lambda}, \bar{\mu})$ est un point-selle du Lagrangien si :

$$\begin{cases} \forall (\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q, & L(\bar{x}; \lambda, \mu) \leq L(\bar{x}; \bar{\lambda}, \bar{\mu}) \\ \forall x \in \mathbb{R}^n, & L(x; \bar{\lambda}, \bar{\mu}) \geq L(\bar{x}; \bar{\lambda}, \bar{\mu}) \end{cases}$$

Autrement dit, si :

$$f(\bar{x}) = \sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} L(\bar{x}; \lambda, \mu) = L(\bar{x}; \bar{\lambda}, \bar{\mu}) = \min_{x \in \mathbb{R}^n} L(x; \bar{\lambda}, \bar{\mu}) = f^*(\bar{\lambda}, \bar{\mu}).$$

Théorème 5.1 (Théorème de dualité) Le point $(\bar{x}, \bar{\lambda}, \bar{\mu})$ est un point-selle du Lagrangien si et seulement si :

- i. \bar{x} est solution du problème primal.
- ii. $(\bar{\lambda}, \bar{\mu})$ est solution du problème dual.
- iii. le saut de dualité est nul c'est-à-dire :

$$\sup_{(\lambda, \mu) \in X^*} \left(\inf_{x \in \mathbb{R}^n} L(x; \lambda, \mu) \right) = \inf_{x \in X} \left(\sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} L(x; \lambda, \mu) \right).$$

Dans ce cas, on a en plus :

$$\begin{aligned} f(\bar{x}) &= \inf_{x \in X} f(x) = \inf_{x \in \mathbb{R}^n} L(x; \bar{\lambda}, \bar{\mu}) \\ &= \sup_{(\lambda, \mu) \in X^*} \left(\inf_{x \in \mathbb{R}^n} L(x; \lambda, \mu) \right) = \inf_{x \in X} \left(\sup_{(\lambda, \mu) \in \mathbb{R}^p \times (\mathbb{R}_+)^q} L(x; \lambda, \mu) \right). \end{aligned}$$

Démonstration. Faite en cours. □

En pratique le théorème de dualité ne nous dit pas comment calculer les points selle du Lagrangien (puisque'il faudrait déjà connaître les solutions optimales des problèmes primal et dual!).

Théorème 5.2

1. Si le problème primal (P) admet une solution optimale, alors le problème dual associé admet une solution optimale de coût égal.
2. Pour que \bar{x} soit une solution optimale du problème primal et que $(\bar{\lambda}, \bar{\mu})$ soit une solution optimale du problème dual associé, il est nécessaire et suffisant d'avoir les trois conditions suivantes :
 - (a) \bar{x} soit admissible pour (P) (i.e. $\bar{x} \in X$).
 - (b) $\bar{\mu} \geq 0$.
 - (c) $f(\bar{x}) = L(\bar{x}; \bar{\lambda}, \bar{\mu}) = \inf_{x \in \mathbb{R}^n} L(x; \bar{\lambda}, \bar{\mu})$.

Démonstration. En exercice. □

L'intérêt du problème dual est double : il est concave et les contraintes sont simples à prendre en compte (contraintes de positivité dans le cas général, pas de contrainte dans le cas de contraintes d'égalité). Pour le résoudre, on peut par exemple utiliser :

- une méthode de gradient dans le cas de contraintes d'égalité : c'est l'algorithme d'Uzawa.
- une méthode de gradient projeté dans le cas de contraintes d'inégalité.
- des méthodes de type Newton.

5.1.3 Cas convexe - Multiplicateurs de Kuhn-Tucker

Plaçons nous dans le cas particulier où la fonction objectif f et les contraintes g_j , $j = 1, \dots, q$, sont convexes et les fonctions h_i , $i = 1, \dots, p$ affines. Le problème :

$$(P) \quad \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.c.} \quad \begin{aligned} g_j(x) &\leq 0, & j = 1, \dots, q, \\ h_i(x) &= 0, & i = 1, \dots, p, \end{aligned}$$

est donc convexe.

Définition 5.2 (CS de qualification convexe : condition de Slater) On dit que les contraintes associées au problème (P) sont qualifiées au sens de la qualification convexe s'il existe $x_0 \in X$ tel que :

$$\begin{aligned} \forall j = 1, \dots, q, \quad g_j(x_0) &< 0, \\ \forall i = 1, \dots, p, \quad h_i(x_0) &= 0. \end{aligned}$$

Théorème 5.3 (Théorème de dualité forte) Si \bar{x} est un point solution de (P) et si la condition de Slater est vérifiée, alors le saut de dualité est nul et il existe au moins un multiplicateur de Lagrange $\bar{\lambda} \in (\mathbb{R}_+)^q$ tel que $(\bar{x}, \bar{\lambda})$ soit un point-selle du Lagrangien et :

$$\nabla f(\bar{x}) + \sum_{j=1}^q \bar{\lambda}_j \nabla g_j(\bar{x}) = 0.$$

Les $\bar{\lambda}_j$, $j = 1, \dots, q$, sont appelés multiplicateurs de Kuhn-Tucker.

Démonstration. Soit \bar{x} une solution du problème (P). L'hypothèse de qualification convexe impliquant la qualification des contraintes énoncée pour les problèmes avec contraintes d'inégalité, il existe $\bar{\lambda} \in (\mathbb{R}_+)^q$ tel que :

$$\nabla f(\bar{x}) + \sum_{j=1}^q \bar{\lambda}_j \nabla g_j(\bar{x}) = 0 \quad \text{soit :} \quad \nabla_x L(\bar{x}; \bar{\lambda}) = 0.$$

Donc \bar{x} est un point critique de l'application : $\psi : x \in \mathbb{R}^n \mapsto L(x; \bar{\lambda})$. Or ψ est convexe comme combinaison linéaire de fonctions convexes, donc \bar{x} est un point de minimum global de $x \in \mathbb{R}^n \mapsto L(x; \bar{\lambda})$.

De plus : $\nabla_\lambda L(\bar{x}; \lambda) = g(\bar{x}) \leq 0$ pour $\lambda \in (\mathbb{R}_+)^q$. Donc l'application $\lambda \mapsto L(\bar{x}; \lambda)$ est maximum en tout point $\lambda \in (\mathbb{R}_+)^q$ vérifiant : $\lambda^\top g(\bar{x}) = 0$, et donc en particulier en $\bar{\lambda}$.

En conclusion, le point \bar{x} est bien un point selle du Lagrangien. \square

5.2 Méthode du gradient projeté

Revenons au problème général suivant :

$$(P) \quad \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. : } x \in X.$$

La méthode du gradient projeté s'inspire des méthodes de gradient décrites dans le chapitre précédent. L'idée de base consiste à suivre la direction de plus profonde descente, comme dans le cas sans contrainte :

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

où $s_k > 0$ est choisi de sorte que : $f(x_k + s_k d_k) < f(x_k)$. Toutefois, si $x_k \in X$, rien ne garantit que x_{k+1} appartienne également à X . Dès que l'on obtient un point non admissible, on projette celui-ci sur l'ensemble de contraintes X .

Projection sur un convexe. Soit X un convexe fermé, non vide de \mathbb{R}^n . La projection d'un point $x \in \mathbb{R}^n$ sur X , notée $p_X(x)$, est obtenue comme solution du problème d'optimisation suivant :

$$\text{Minimiser } \frac{1}{2} \|x - y\|_2^2 \quad \text{sous la contrainte : } y \in X \quad (5.1)$$

La fonctionnelle : $y \in \mathbb{R}^n \mapsto \frac{1}{2}\|x - y\|_2^2$ étant convexe et $\nabla f(y) = y - x$, le corollaire 3.3 nous donne une condition nécessaire et suffisante pour que $x^* = p_X(x)$ soit solution de (5.1) :

$$\forall y \in X, (x^* - x)^\top (y - x^*) \geq 0. \quad (5.2)$$

On remarque en particulier que si $x \in X$, alors nécessairement : $x^* = x$.

Principe de l'algorithme.

ALGORITHME DU GRADIENT PROJETÉ.

Données: f , p_X un opérateur de projection sur X , x_0 point initial $\varepsilon > 0$ précision demandée.

Sortie: une approximation x^* de la solution.

1. $k := 0$;
 2. Tant que critère d'arrêt non satisfait,
 - (a) $y_k = x_k - s\nabla f(x_k)$ où s est le pas calculé par la méthode de gradient choisie ;
 - (b) *Projection sur X :* $x_{k+1} = p_X(y_k)$; $k := k + 1$;
 3. Retourner x_k .
-

Remarque 5.1 *Il est important de remarquer que le calcul à l'étape 2(a) du projeté sur X , peut parfois être aussi difficile que le problème initial. En effet y_k est obtenu en résolvant le problème :*

$$\begin{aligned} \min_{y \in \mathbb{R}^n} \quad & \frac{1}{2} \|x_k - s\nabla f(x_k) - y\|_2^2 \\ \text{s.c.} \quad & x \in X. \end{aligned}$$

Il s'agit donc de résoudre un problème d'optimisation sur un convexe, avec une fonction objectif convexe. Lorsque le domaine X des contraintes est simple (contraintes de bornes en particulier), c'est faisable. Dès que les contraintes ne sont pas des contraintes de bornes, le calcul de la projection devient beaucoup plus délicat.

Vérifions que la direction $d_k = x_{k+1} - x_k$, si elle est non nulle, est bien une direction de descente de f en x_k .

Lemme 5.1 *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ supposée différentiable et $X \subset \mathbb{R}^n$ un convexe fermé, non vide. Notons x_k l'itéré courant et :*

$$d(s) = p_X(x_k - s\nabla f(x_k)) - x_k, \quad s > 0$$

Si $d(s)$ est non nulle, alors $d(s)$ est une direction de descente pour tout $s > 0$.

Démonstration. En exercice. □

Remarque 5.2 *La direction $d(s)$ possède les propriétés suivantes :*

1. Si $d(s) = 0$, alors : $p_X(x_k - s\nabla f(x_k)) = x_k$. Cela signifie que la direction choisie par l'algorithme de gradient est orthogonale à l'ensemble X des contraintes en x_k . Le point x_k est alors un point stationnaire car la condition nécessaire d'optimalité (3.3) est satisfaite.
2. Supposons $d(s) \neq 0$. Alors x_k et $p_X(x_k - s\nabla f(x_k))$ sont des points admissibles du problème (P). La convexité de X nous garantit alors : $\forall \alpha \in [0, 1], x_k + \alpha d(s) \in X$.

Convergence de l'algorithme de gradient projeté

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ convexe, différentiable de gradient supposé Lipschitz de constante $L > 0$, i.e. :

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|.$$

Soit $(x_k)_{k \in \mathbb{N}}$ une suite d'itérés générés par un algorithme de gradient projeté selon le schéma :

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

De même que pour les méthodes de descente de gradient, une condition suffisante de convergence est la suivante :

Proposition 5.2 Si $0 < s_k < \frac{2}{L}$ alors la suite $(x_k)_k$ générée par l'algorithme de gradient projeté converge vers un point de minimum de f .

5.3 Algorithmes newtoniens - Méthode SQP

Les algorithmes newtoniens sont basés sur la linéarisation d'équations caractérisant les solutions que l'on cherche, fournies par les conditions d'optimalité d'ordre 1. Ces algorithmes sont *primaux-duaux* dans le sens où ils génèrent à la fois une suite primale (x^k) convergeant vers une solution \bar{x} du problème considéré, et une suite duale (λ^k) de multiplicateurs convergeant vers un multiplicateur optimal $\bar{\lambda}$ associé à \bar{x} .

Contraintes d'égalité

Considérons un problème d'optimisation (différentiable) avec contraintes d'égalité :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t. :} \quad & h_i(x) = 0, \quad i = 1, \dots, p. \end{aligned} \tag{5.3}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$, sont supposées au moins différentiables. D'après ce qui a été vu au chapitre 3, les conditions d'optimalité de Lagrange s'écrivent :

$$\nabla L(x; \lambda) = 0 \quad \text{soit :} \quad \begin{pmatrix} \nabla f(x) + \sum_{i=1}^p \lambda_i \nabla h_i(x) \\ h(x) \end{pmatrix} = 0.$$

Pour résoudre ce système d'équations, utilisons la méthode de Newton dont une itération s'écrit ici :

$$H[L](x^k; \lambda^k) \begin{pmatrix} x^{k+1} - x^k \\ \lambda_{k+1} - \lambda^k \end{pmatrix} = -\nabla L(x^k; \lambda^k), \quad (5.4)$$

soit :

$$\begin{pmatrix} H_x[L](x^k; \lambda^k) & Dh(x^k)^\top \\ Dh(x^k) & 0 \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} = - \begin{pmatrix} \nabla_x L(x^k; \lambda^k) \\ h(x^k) \end{pmatrix}$$

où $Dh(x)$ désigne la matrice jacobienne de l'application $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ définie par :

$$Dh(x)^\top = [\nabla h_1(x) | \dots | \nabla h_p(x)].$$

Posons : $H_k = H_x[L](x^k; \lambda^k)$, $d = x^{k+1} - x^k$ et $\mu = \lambda^{k+1}$. L'itération (5.4) s'écrit donc :

$$\begin{pmatrix} H_k & Dh(x^k)^\top \\ Dh(x^k) & 0 \end{pmatrix} \begin{pmatrix} d \\ \mu - \lambda^k \end{pmatrix} = - \begin{pmatrix} \nabla_x L(x^k; \lambda^k) \\ h(x^k) \end{pmatrix}$$

et est bien définie à condition que la matrice $H[L](x^k; \lambda^k)$ soit inversible. Ce sera le cas si :

- (i) Les colonnes $\nabla h_1(x), \dots, \nabla h_p(x)$ de $Dh(x^k)^\top$ sont linéairement indépendants : c'est l'hypothèse de qualification des contraintes.
- (ii) Quel que soit $d \neq 0$ tel que $Dh(x^k)d = 0$, $d^\top H_k d > 0$: c'est la condition suffisante d'optimalité du second ordre dans le cas de contraintes d'égalité.

Revenons à l'itération (5.4). Elle s'écrit encore :

$$\begin{cases} H_k d + \sum_{i=1}^p (\mu_i - \lambda_i^k) \nabla h_i(x^k) = -\nabla_x L(x^k; \lambda^k) \\ \nabla h_i(x^k)^\top d + h_i(x^k) = 0, & i = 1, \dots, p \end{cases}$$

Or : $\nabla_x L(x^k; \lambda^k) = \nabla f(x_k) + \sum_{i=1}^p \lambda_i^k \nabla h_i(x^k)$, d'où :

$$\begin{cases} H_k d + \sum_{i=1}^p \mu_i \nabla h_i(x^k) = -\nabla f(x^k) \\ \nabla h_i(x^k)^\top d + h_i(x^k) = 0, & i = 1, \dots, p \end{cases}$$

On reconnaît dans le système ci-dessus les conditions d'optimalité de Lagrange du problème quadratique suivant :

$$(QP_k) \quad \begin{cases} \min_{d \in \mathbb{R}^n} \nabla f(x_k)^\top d + \frac{1}{2} d^\top H_k d \\ \text{s.t. } h_i(x^k) + \nabla h_i(x^k)^\top d = 0. \end{cases}$$

Le problème (QP_k) peut être vu comme la minimisation d'une approximation quadratique du Lagrangien de (P_E) avec une approximation linéaire des contraintes.

Comme son nom l'indique, la méthode SQP (Sequential Quadratic Programming) consiste à remplacer le problème initial par une suite de problèmes quadratiques sous contraintes linéaires plus faciles à résoudre. L'algorithme est le suivant :

ALGORITHME SQP AVEC CONTRAINTES D'ÉGALITÉ.

Données: $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ différentiables, x_0 point initial, $\lambda_0 \in \mathbb{R}^p$ multiplicateur initial, $\varepsilon > 0$ précision demandée.

Sortie: une approximation x^* de la solution.

1. $k := 0$;
2. Tant que $\|\nabla L(x^k; \lambda^k)\| > \varepsilon$,
 - (a) Résoudre le sous-problème quadratique :

$$(QP_k) \quad \left\{ \begin{array}{l} \min_{d \in \mathbb{R}^n} \quad \nabla f(x_k)^\top d + \frac{1}{2} d^\top H_k d \\ \text{s.t.} \quad h_i(x^k) + \nabla h_i(x^k)^\top d = 0. \end{array} \right.$$

et obtenir la solution primale d_k et le multiplicateur λ' associé à la contrainte d'égalité.

- (b) $x^{k+1} = x_k + d_k$; $\lambda^{k+1} = \lambda'$; $k = k + 1$;
 3. Retourner x_k .
-

Contraintes d'inégalité

Intéressons nous maintenant aux problèmes avec contraintes d'égalité et d'inégalité :

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t. :} & g_j(x) \leq 0, \quad j = 1, \dots, q, \\ & h_i(x) = 0, \quad i = 1, \dots, p. \end{array} \quad (5.5)$$

Selon le même principe qu'avec contraintes d'égalité seules, on linéarise les contraintes et on utilise une approximation quadratique du lagrangien :

$$L(x; \lambda, \mu) = f(x) + \lambda^\top g(x) + \mu^\top h(x), \quad \text{avec : } \lambda_j \geq 0, j = 1, \dots, q.$$

ALGORITHME SQP AVEC CONTRAINTES D'ÉGALITÉ ET D'INÉGALITÉ.

Données: $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ différentiables, x_0 point initial, $\lambda_0 \in \mathbb{R}_+^q$ et $\mu_0 \in \mathbb{R}^p$ multiplicateurs initiaux, $\varepsilon > 0$ précision demandée.

Sortie: une approximation x^* de la solution.

1. $k := 0$;
2. Tant que $\|\nabla L(x^k; \lambda^k)\| > \varepsilon$,

(a) Résoudre le sous-problème quadratique :

$$(QP_k) \quad \left\{ \begin{array}{l} \min_{d \in \mathbb{R}^n} \quad \nabla f(x_k)^\top d + \frac{1}{2} d^\top H_k d \\ \text{s.t.} \quad g_j(x^k) + \nabla g_j(x^k)^\top d = 0, \quad i = 1, \dots, q, \\ \quad \quad h_i(x^k) + \nabla h_i(x^k)^\top d = 0, \quad i = 1, \dots, p. \end{array} \right.$$

et obtenir la solution primale d_k et les multiplicateurs λ' et μ' associés aux contraintes d'inégalité et d'égalité respectivement.

(b) $x^{k+1} = x_k + d_k$; $\lambda^{k+1} = \lambda'$; $\mu^{k+1} = \mu'$; $k = k + 1$;

3. Retourner x_k .

Afin que le sous-programme quadratique (QP) admette une unique solution, la plupart des implémentations actuelles de SQP utilisent une approximation du hessien H_k du Lagrangien qui soit définie positive, en particulier celle fournie par les techniques quasi-newtonienne (BFGS) par exemple.

Etant une méthode newtonienne, l'algorithme SQP converge localement quadratiquement pourvu que les points initiaux (x_0, λ_0) (resp. (x_0, λ_0, μ_0)) soient dans un voisinage d'un point stationnaire \bar{x} et de ses multiplicateurs associés $\bar{\lambda}$ (resp. $(\bar{\lambda}, \bar{\mu})$). Bien entendu, il est possible de globaliser l'algorithme en ajoutant une étape de recherche linéaire comme cela a été fait au chapitre 4.

5.4 Méthodes de pénalisation

La pénalisation est un concept simple qui permet de transformer un problème d'optimisation avec contraintes en un problème ou en une suite de problèmes d'optimisation sans contrainte.

D'un point de vue théorique, une approche par pénalisation peut être utilisée pour étudier un problème d'optimisation dont les contraintes sont difficiles à prendre en compte, alors que le problème pénalisé a des propriétés mieux comprises ou plus simples à mettre en évidence.

D'un point de vue numérique, cette transformation en problèmes sans contrainte permet d'utiliser des algorithmes d'optimisation sans contrainte pour obtenir la solution de problèmes dont l'ensemble admissible peut avoir une structure complexe. Cette approche est très souvent utilisée : elle permet d'obtenir une solution de qualité suffisante rapidement sans avoir à entrer dans l'algorithmique sophistiquée de l'optimisation avec contraintes. Ce n'est cependant pas une technique universelle, car elle a ses propres inconvénients : non-différentiabilité ou nécessité de minimiser une suite de fonctions de plus en plus mal conditionnées.

5.4.1 Principe général

Le principe de la pénalisation consiste à remplacer le problème sous contraintes $(P) \min_{x \in X} f(x)$ initial par un problème (ou une suite de problèmes) de la forme :

$$(P_c) \quad \min_{x \in \mathbb{R}^n} f(x) + c\pi(x),$$

où $c > 0$ est appelé *paramètre de pénalisation* et $\pi : \mathbb{R}^n \rightarrow \mathbb{R}$ la fonction de pénalisation à choisir (souvent choisie à valeurs positives). Plus le paramètre c est grand, plus on accorde de l'importance à la pénalisation. L'intérêt du problème (P_c) est de pouvoir être résolu par une méthode de l'optimisation sans contrainte.

La question qui se pose alors est si résoudre le problème (P_c) permettra de résoudre le problème (P) . Ou autrement dit, si les ensembles de solutions des deux problèmes coïncident. Cela dépendra bien évidemment du choix de c et π .

Définition 5.3 Une pénalisation est dite exacte si toute solution du problème (P) initial est solution du problème pénalisé (P_c) , et inexacte dans le cas contraire.

Un exemple simple de pénalisation exacte mais peu satisfaisant

$$\pi(x) = \begin{cases} 0 & \text{si } x \in X \\ +\infty & \text{sinon.} \end{cases}$$

On a alors bien équivalence du problème initial et du problème pénalisé mais ce choix est très peu utilisé en pratique car les algorithmes classiques de l'optimisation ne peuvent pas être utilisés sur des fonctions valant $+\infty$ au cours des itérations.

Pénalisation extérieure Il s'agit de définir une fonction p qui pénalise la violation des contraintes :

$$\begin{cases} \pi \text{ continue sur } \mathbb{R}^n \\ \forall x \in \mathbb{R}^n, \pi(x) \geq 0 \\ \pi(x) = 0 \Leftrightarrow x \in X \end{cases}$$

Pénalisation intérieure Dans certains cas, il n'est pas envisageable que les itérés n'appartiennent pas au domaine X des contraintes, en particulier quand la fonction objectif n'est pas définie en dehors de X . Dans ce cas, l'idée est d'utiliser un terme de pénalisation qui tend vers $+\infty$ lorsque x se rapproche de la frontière ∂X de X .

$$\begin{cases} \pi \text{ continue sur } \text{int } X \\ \forall x \in \text{int } X, \pi(x) \geq 0 \\ \pi(x) \rightarrow +\infty \text{ lorsque } x \text{ se rapproche de } \partial X \end{cases}$$

Le Lagrangien Plaçons nous dans le cas de contraintes fonctionnelles. Le Lagrangien associé au problème (P) est alors une fonction de pénalisation exacte du problème initial à condition que le problème soit convexe et si le multiplicateur utilisé est optimal (le vérifier!). Tout le problème est justement de connaître ce multiplicateur

optimal.

La relaxation lagrangienne est une approche classique consistant à chercher une solution de (P) en minimisant $L(\cdot, \lambda, \mu)$ pour une suite de multiplicateurs (λ, μ) que l'on fait converger vers un multiplicateur optimal. Deux problèmes se pose alors : comment mettre à jour ces multiplicateurs et comment étendre cette approche au cas non convexe.

5.4.2 Le Lagrangien augmenté

Dans ce paragraphe, nous nous intéressons au Lagrangien augmenté de problèmes d'optimisation avec contraintes d'égalité uniquement. La méthode présentée ci-après se généralise aux problèmes avec contraintes d'inégalité mais est un peu plus technique (cf [25] ou [6] entre autres).

Considérons le problème suivant :

$$(P) \quad \begin{array}{l} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. : } h_i(x) = 0, \quad i = 1, \dots, p. \end{array}$$

Soit $c > 0$ une constante arbitraire. On pénalise le problème (P) de la façon suivante :

$$\begin{array}{l} \min_{x \in \mathbb{R}^n} g(x) := f(x) + \frac{c}{2} \|h(x)\|_2^2 \\ \text{s.t. : } h_i(x) = 0, \quad i = 1, \dots, p. \end{array}$$

Le Lagrangien du problème pénalisé :

$$L_c(x; \lambda) = f(x) + \langle \lambda, h(x) \rangle + \frac{c}{2} \|h(x)\|_2^2 \quad (5.6)$$

est appelé *Lagrangien augmenté* associé au problème (P) .

L'intérêt de cette approche est qu'il existe une valeur seuil \bar{c} du paramètre de pénalité telle que pour tout $c > \bar{c}$, tout minimum local du problème initial est un minimum local du Lagrangien augmenté.

Théorème 5.4 ([6], chapitre 16) Soit \bar{x} un point de minimum local de f sur X . On suppose qu'il existe $\bar{\lambda} \in \mathbb{R}^p$ tel que : $\nabla_x L_c(\bar{x}, \bar{\lambda}) = 0$ et tel que la condition suffisante de second ordre soit satisfaite, i.e. :

$$\forall v \in \text{Ker} Dh(\bar{x}) - \{0\}, \quad \langle v, H_{xx}[L](\bar{x}, \bar{\lambda})v \rangle > 0.$$

Alors il existe $\bar{c} > 0$ tel que pour tout $c \geq \bar{c}$, le Lagrangien augmenté L_c admet un minimum local strict en \bar{x} .

Exercice 5.4.1 Soit :

$$\min_{x \in \mathbb{R}^2} \frac{1}{2} (-x_1^2 + x_2^2) \quad \text{sous} \quad x_1 = 1. \quad (5.7)$$

1. Calculer la solution optimale x^* du problème (5.7) et le multiplicateur λ^* associé.
2. Ecrire le Lagrangien augmenté $L_c(x; \lambda)$ associé au problème (5.7).
3. A quelle condition sur $c > 0$, x^* est-il un point de minimum local strict de L_c ?

La fonction duale associée au problème pénalisé :

$$f_{c_k}^*(\lambda) = \min_{x \in \mathbb{R}^n} L_{c_k}(x; \lambda).$$

Supposons que le problème $\min_{x \in \mathbb{R}^n} L_{c_k}(x; \lambda)$ admette une solution $x(\lambda)$ différentiable par rapport à λ . Alors :

$$f_{c_k}^*(\lambda) = L_{c_k}(x(\lambda); \lambda).$$

Sachant que $x(\lambda)$ minimise $L_{c_k}(x; \lambda)$ par rapport à x , on a nécessairement : $\nabla_x L_{c_k}(x(\lambda); \lambda) = 0$. Calculons maintenant le gradient de la fonction duale :

$$\begin{aligned} \nabla f_{c_k}^*(\lambda) &= \nabla_x L_{c_k}(x(\lambda); \lambda) dx(\lambda) + \nabla_\lambda L_{c_k}(x(\lambda); \lambda) \\ &= h(x(\lambda)). \end{aligned}$$

La direction $\nabla f_{c_k}^*(\lambda) = h(x(\lambda))$ est la direction de plus forte pente de la fonction duale en λ , ce qui justifie une itération de la forme :

$$\lambda_{k+1} = \lambda_k + c_k h(x(\lambda_k)).$$

L'algorithme est le suivant : étant donnée une suite $(c_k)_{k \in \mathbb{N}}$ croissante de réels strictement positifs telle que : $\lim_{k \rightarrow +\infty} c_k = +\infty$:

1. Initialisation : x_0 et λ_0 arbitraires, $\varepsilon > 0$ précision.
2. Tant que $\|\nabla L_{c_k}(x_k; \lambda_k)\| > \varepsilon$,
 - (a) **Approximation de la fonction duale en λ_k .** Résoudre approximativement :

$$(Q_k) \quad \min_{x \in \mathbb{R}^n} L_{c_k}(x; \lambda_k)$$

On note x_k la solution approchée.

- (b) **Itération du gradient pour la maximisation de la fonction duale.**

$$\lambda_{k+1} = \lambda_k + c_k h(x_k);$$

Troisième partie

Introduction à l'optimisation convexe non différentiable

Chapitre 6

Compléments d'analyse convexe

Dans ce chapitre, nous nous introduisons les outils spécifiques à l'optimisation non différentiable tels que le sous-différentiel et l'opérateur proximal. Nous verrons également comment s'écrivent les conditions nécessaires d'optimalité dans le cadre non lisse.

6.1 Sous-différentiel d'une fonction convexe

6.1.1 Définition et propriétés

Définition 6.1 (Sous-gradient et sous-différentiel) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction convexe. Un vecteur $\eta \in \mathbb{R}^n$ est appelé sous-gradient de f au point $x_0 \in \text{dom}(f)$ si

$$\forall x \in \text{dom}(f), f(x) \geq f(x_0) + \langle \eta, x - x_0 \rangle. \quad (6.1)$$

L'ensemble de tous les sous-gradients en x_0 est appelé sous-différentiel de f en x_0 et noté $\partial f(x_0)$.

L'interprétation géométrique du sous-différentiel est la suivante. Il est formé par toutes les directions des hyperplans qui passent par le point $(x, f(x))$ et restent "sous" le graphe de la fonction f . Ces hyperplans sont appelés *hyperplans support* ou *hyperplans d'appui* au graphe de f en x .

Exemple 6.1.1 Soit $f : x \in \mathbb{R} \mapsto |x|$. Calculer le sous-différentiel de f en tout point $x \in \mathbb{R}$.

Lemme 6.1 Le sous-différentiel $\partial f(x_0) = \{\eta \in \mathbb{R}^n, \mid \forall x \in \mathbb{R}^n, f(x) \geq f(x_0) + \langle \eta, x - x_0 \rangle\}$ est un ensemble convexe fermé.

Démonstration. En exercice. □

Théorème 6.1 Soit f une fonction convexe s.c.i. et $x_0 \in \text{int}(\text{dom}(f))$. Alors $\partial f(x_0)$ est un ensemble non vide, convexe, borné.

Démonstration. Notons d'abord que le point $(f(x_0), x_0)$ appartient à la frontière de $\text{epi}(f)$. D'après le théorème de séparation de Hahn-Banach, il existe donc un hyperplan d'appui à $\text{epi}(f)$ au point $(f(x_0), x_0)$:

$$-\alpha\tau + \langle d, x \rangle \leq -\alpha f(x_0) + \langle d, x_0 \rangle \quad (6.2)$$

pour tout $(\tau, x) \in \text{epi}(f)$. On peut choisir

$$\|d\|_2^2 + \alpha^2 = 1. \quad (6.3)$$

Puisque pour tout $\tau \geq f(x_0)$ le point (τ, x_0) appartient à $\text{epi}(f)$, on conclut que $\alpha \geq 0$.

D'après le lemme 1.2, il existe $\epsilon > 0$ et $M > 0$ tels que $B_2(x_0, \epsilon) \subseteq \text{dom}(f)$ et

$$f(x) - f(x_0) \leq M\|x - x_0\|$$

pour tout $x \in B_2(x_0, \epsilon)$. Ainsi, d'après (6.2), pour tout x dans cette boule on a

$$\langle d, x - x_0 \rangle \leq \alpha(f(x) - f(x_0)) \leq \alpha M\|x - x_0\|_2.$$

En choisissant $x = x_0 + \epsilon d$ on obtient $\|d\|_2^2 \leq M\alpha\|d\|_2$. En utilisant la condition de normalisation (6.3), on obtient

$$\alpha \geq \frac{1}{\sqrt{1 + M^2}}.$$

Ainsi, en choisissant $g = d/\alpha$ on obtient pour tout $x \in \text{dom}(f)$

$$f(x) \geq f(x_0) + \langle g, x - x_0 \rangle.$$

Finalement, si $g \in \partial f(x_0)$ et $g \neq 0$, alors en choisissant $x = x_0 + g/\|g\|_2$ on obtient

$$\epsilon\|g\|_2 = \langle g, x - x_0 \rangle \leq f(x) - f(x_0) \leq M\|x - x_0\|_2 = M\epsilon.$$

Ainsi $\partial f(x_0)$ est borné. □

Remarque 6.1 *Le sous-différentiel peut ne pas exister sur le bord du domaine. Par exemple la fonction $f(x) = -\sqrt{x}$ sur \mathbb{R}_+ est convexe et fermée, mais le sous-différentiel n'existe pas en 0 car $\lim_{x \rightarrow 0^+} f(x) = -\infty$.*

6.1.2 Règles de calcul sous-différentiel

Lemme 6.2 Soit f une fonction convexe s.c.i.. On a :

i. Si f est différentiable sur son domaine, alors :

$$\forall x \in \text{int}(\text{dom}(f)), \partial f(x) = \{\nabla f(x)\}.$$

ii. Si $\text{dom}(f) \subseteq \mathbb{R}^n$, $A : \mathbb{R}^m \rightarrow \mathbb{R}^n$ un opérateur linéaire et $b \in \mathbb{R}^n$, alors la fonction $\phi(x) = f(Ax + b)$ est convexe s.c.i. et :

$$\forall x \in \text{int}(\text{dom}(\phi)), \partial \phi(x) = A^\top \partial f(Ax + b).$$

iii. Si $f(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x)$ avec f_1 et f_2 convexes s.c.i. sur \mathbb{R}^n et $(\alpha_1, \alpha_2) \in \mathbb{R}_+ \times \mathbb{R}_+$. Alors :

$$\begin{aligned} \partial f(x) &= \alpha_1 \partial f_1(x) + \alpha_2 \partial f_2(x) \\ &= \{\eta \in \mathbb{R}^n, \exists (x_1, x_2) \in \partial f_1(x_1) \times \partial f_2(x_2), \eta = x_1 + x_2\}. \end{aligned}$$

iv. Soit $g : \mathbb{R} \cup +\infty \rightarrow \mathbb{R} \cup +\infty$ une fonction convexe croissante. Notons : $h = g \circ f$. Alors $\forall x \in \text{int}(\text{dom}(f))$,

$$\partial h(x) = \{\eta_1 \eta_2 \mid \eta_1 \in \partial g(f(x)), \eta_2 \in \partial f(x)\}. \quad (6.4)$$

Exemple 6.1.2 Soit $f(x) = \|x\|_2$, $g(t) = \frac{1}{2}t^2$ et $h(x) = g(f(x)) = \frac{1}{2}\|x\|_2^2$. Démontrer que :

$$\partial f(x) = \begin{cases} \left\{ \frac{x}{\|x\|_2} \right\} & \text{si } x \neq 0, \\ \{x, \|x\|_2 \leq 1\} & \text{sinon.} \end{cases} \quad (6.5)$$

En déduire le résultat connu : $\partial h(x) = \{x\}$.

Lemme 6.3 Soit $(f_i)_{i=1..m}$ un ensemble de fonctions convexes s.c.i. Alors la fonction

$$f(x) = \max_{i=1..m} f_i(x)$$

est aussi convexe s.c.i. de domaine $\text{dom}(f) = \bigcap_{i=1}^m \text{dom}(f_i)$ et :

$$\forall x \in \text{int}(\text{dom}(f)), \partial f(x) = \text{conv}(\partial f_i(x), i \in I(x))$$

où $I(x) = \{i \in \{1, \dots, m\}, f_i(x) = f(x)\}$.

Le lemme suivant se révèle souvent utile pour étudier les fonctions définies comme des suprema¹.

1. En réalité, toutes les fonctions convexes peuvent être représentées comme un supremum en utilisant la transformée de Fenchel. Ce résultat est donc central. Le cours est trop court pour présenter cette théorie.

Lemme 6.4 Soit $\phi(x, y)$ une fonction telle que $\forall y \in \Delta$, la fonction $x \mapsto \phi(x, y)$ est convexe s.c.i. Alors la fonction

$$f(x) = \sup_{y \in \Delta} \phi(x, y)$$

est convexe s.c.i.. De plus, $\text{dom}(f) = \{x \in \mathbb{R}^n, \exists \gamma, \phi(x, y) \leq \gamma, \forall y \in \Delta\}$ et

$$\partial f(x) \supseteq \text{conv}(\partial_x \phi(x, y), y \in I(x), I(x) = \{y, \phi(x, y) = f(x)\}).$$

6.1.3 Exemple 1 : Sous-différentiel d'une fonction indicatrice

Définition 6.2 Soit $Q \subseteq \mathbb{R}^n$ un ensemble convexe, fermé. Soit $x_0 \in \partial Q$. On appelle cône normal à Q en x_0 et on note $N_Q(x_0)$ l'ensemble suivant :

$$N_Q(x_0) = \{\eta \in \mathbb{R}^n, \langle \eta, x - x_0 \rangle \leq 0, \forall x \in Q\}.$$

Lemme 6.5 (Sous-différentiel d'une indicatrice) Soit $X \subseteq \mathbb{R}^n$ un ensemble convexe fermé. On définit l'indicatrice de X comme :

$$\chi_X(x) = \begin{cases} 0 & \text{si } x \in X \\ +\infty & \text{sinon} \end{cases}$$

Le sous-différentiel de χ_X au point $x \in \partial X$ est le cône normal de X en x :

$$\partial \chi_X(x) = N_X(x).$$

6.1.4 Exemple 2 : Sous-différentiel d'une norme

Définition 6.3 (Norme duale) Soit $\|\cdot\|_X$ une norme sur \mathbb{R}^n . On appelle norme duale et on note $\|\cdot\|_{X^*}$ la fonction définie par :

$$\|y\|_{X^*} = \sup_{x \in \mathbb{R}^n, \|x\|_X \leq 1} \langle x, y \rangle. \quad (6.6)$$

On peut montrer que la fonction $\|\cdot\|_{X^*}$ définit bien une norme sur \mathbb{R}^n . De plus, par définition, on obtient pour tout $(x, y) \in \mathbb{R}^{n \times n}$:

$$|\langle x, y \rangle| \leq \|x\|_X \|y\|_{X^*}, \quad (6.7)$$

qui généralise les inégalités de Cauchy-Schwartz et de Hölder.

Exemple 6.1.3 Soit $\|x\|_X = \|x\|_p$ où $\|x\|_p$ est la norme ℓ^p usuelle sur \mathbb{R}^n . Montrer que :

$$\|y\|_{X^*} = \|x\|_q,$$

avec $1/p + 1/q = 1$. En particulier, la norme ℓ^2 est égale à sa norme duale.

Définition 6.4 (Sous-différentielle d'une norme) Soit $f(x) = \|x\|_X$ où $\|\cdot\|_X$ est une norme arbitraire sur \mathbb{R}^n . Alors :

$$\partial f(x) = \arg \max_{\|y\|_{X^*} \leq 1} \langle x, y \rangle. \quad (6.8)$$

6.2 Condition d'optimalité

La notion de sous-différentiel est centrale car il permet notamment de caractériser les minimiseurs d'une fonction. On considère le problème suivant :

$$\text{Trouver } x^* \text{ tel que } f(x^*) = \min_{x \in \mathbb{R}^n} f(x)$$

où f est une fonction convexe s.c.i.

Théorème 6.2 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe. Le point $x^* \in \mathbb{R}^n$ est un point de minimum global de f si et seulement si :

$$0 \in \partial f(x^*).$$

Démonstration. En exercice. □

Chapitre 7

Algorithmes pour l'optimisation non différentiable

Dans ce chapitre, nous nous intéressons aux méthodes d'optimisation adaptées au cas où la fonction f à minimiser est convexe mais non différentiable. De manière générale, les méthodes pour l'optimisation non différentiable s'inspirent largement des méthodes différentiables. Pourquoi alors étudier des méthodes spéciales ? L'optimisation différentiable présente des pièges dans lesquels un utilisateur non averti pourrait tomber :

1. **Piège du test d'arrêt.** La condition " $\|g_k\| \leq \varepsilon, g_k \in \partial f(x_k)$ ", traduite directement de la condition " $\nabla f(x_k) \leq \varepsilon$ " dans le cas différentiable, peut ne jamais être vérifiée.
2. **Piège des sous-gradients approchés.** En pratique, le gradient (et même la fonction elle-même) n'est pas toujours calculé exactement : il est souvent obtenu par différences finies. Cette approche n'est plus valable quand le sous-gradient n'est pas continu.
3. **Malédiction du sous-différentiable.** L'application $x \mapsto \partial f(x)$ n'étant pas continue une petite variation de x_k peut entraîner de grandes variations de $\partial f(x_k)$. Basé sur l'information disponible du sous-différentiel, le calcul d'une direction de recherche d_k peut varier énormément et produire des x_{k+1} très différents.

7.1 Un rapide tour d'horizon en optimisation non lisse

Les problèmes d'optimisation non lisse, même en l'absence de contraintes, sont en général très difficiles à résoudre. Il existe actuellement trois grandes familles de méthodes : les méthodes de *sous-gradients* [26, 28], les méthodes de *faisceaux* [16, 31, 18, 11, 19] et plus récemment les méthodes d'*échantillonnage de gradient* (sous-gradient) [7, 8].

Les méthodes de sous-gradients sont une généralisation naturelle des méthodes de gradient au cas non différentiable en remplaçant le gradient par *un* sous-gradient arbitraire. Ces méthodes (appelées *méthodes de Kiev*) ont été introduites dans les années 60 par N.Z. Shor pour la minimisation sans contrainte de fonctions convexes, et l'on trouve les premiers résultats de convergence dans les travaux de Y.M. Ermoliev [10] ou de B.T. Polyak [26]. Une référence classique sur le sujet est le livre de N.Z. Shor [28]. Le succès de ces

méthodes est dû à la simplicité de leur mise en œuvre, en particulier en grande dimension. Bien qu'elles soient largement utilisées en optimisation non lisse, elles souffrent de plusieurs inconvénients : contrairement aux méthodes de gradient en optimisation différentiable, ce ne sont pas des méthodes de descente. Cela peut conduire à des phénomènes oscillatoires et à de mauvaises performances numériques. On ne dispose pas de critère d'arrêt naturel et implémentable permettant de détecter l'optimalité. Enfin, leur vitesse de convergence est en général assez faible (sous-linéaire) même s'il est possible de maintenir une convergence linéaire par des techniques de dilatation en espace le long du gradient [28, Chapitre 3].

Les méthodes de faisceaux ont constitué une avancée majeure pour la résolution de problèmes d'optimisation non différentiable. Historiquement, les méthodes de faisceaux sont basées sur la méthode des plans sécants [9, 13], pour la minimisation sans contrainte de fonctionnelles convexes. Le principe est le suivant : au lieu de faire appel à l'oracle pour générer des candidats à la descente, on se sert des informations de premier ordre de la fonction objectif f pour construire un modèle (convexe) de f , plus facile à minimiser. Etant donné un faisceau d'informations : $\{(x_i, f_i = f(x_i), s_i \in \partial f(x_i)) ; i = 1, \dots, k\}$ obtenu à partir des itérations précédentes, on construit une approximation convexe linéaire par morceaux de la fonction f (cf Figure 7.1) :

$$\forall y \in \mathbb{R}^n, \phi_k(y) = \max_{i=1, \dots, k} \{f_i + \langle s_i, y - x_i \rangle\} (\leq f(y)).$$

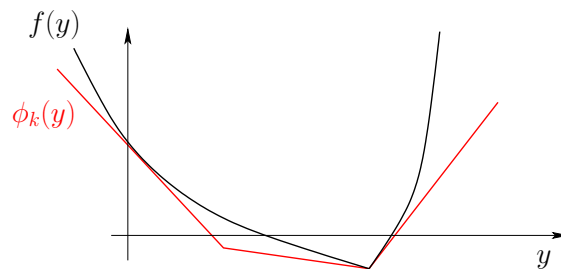


FIGURE 7.1 – Méthode des plans sécants - Construction d'un modèle convexe linéaire par morceaux d'une fonction f convexe, non différentiable.

A chaque itération de l'algorithme, la minimisation du modèle ϕ_k donne un nouvel itéré x_{k+1} qui va permettre d'enrichir le modèle courant par la donnée d'un nouveau plan sécant. L'information passée est ainsi conservée au fil des itérations.

Ces méthodes souffrent de nombreuses limitations : en effet, pour des fonctions convexes très générales, la méthode des plans sécants peut être très instable et avoir de mauvaises performances numériques, [23, Chapitre 4, Section 4.3.6] et [6, Chapitre 9, Exemple 9.7]. Comme pour les algorithmes de sous-gradient, ce ne sont pas des méthodes de descente de la fonction f . Une autre limitation provient de l'accumulation infinie de plans sécants dans le faisceau ce qui augmente à chaque itération la complexité du problème d'optimisation à résoudre, en termes de taille du problème à résoudre mais aussi de conditionnement.

Les méthodes de faisceaux sont des méthodes de stabilisation de la méthode des plans coupants. Considérées en 2001 comme "l'outil le plus efficace et le plus prometteur pour

l'optimisation non lisse" [22], elles ont été largement éprouvées et ont fait leurs preuves dans le cas convexe [16, 31, 18, 11, 22, 6]. L'hypothèse de convexité étant un élément clé de ces méthodes, leur généralisation au cas non convexe est loin d'être immédiate et fait l'objet de recherche actuelles.

Depuis 2002, une nouvelle approche basée sur des algorithmes d'échantillonnage de gradient (ou sous-gradients) et développée par J.V. Burke, A.S. Lewis et M.L. Overton [7, 8], s'est peu à peu imposée en méthode concurrente des méthodes de faisceaux, tant sur le plan théorique que sur le plan des performances numériques. Cette méthode, qui peut être vue comme une version stabilisée de l'algorithme de plus profonde descente, est dédiée à la minimisation de fonctions localement lipschitziennes, supposées de classe C^1 sur un sous-ensemble ouvert dense de \mathbb{R}^n . La fonction objectif est possiblement non lisse, non convexe. L'idée centrale est d'approcher le sous-différentiel de la fonction objectif par l'enveloppe convexe d'un échantillonnage aléatoire de gradients calculés dans un voisinage du point courant. Un premier résultat de convergence avec une probabilité 1 a été démontré dans [7, 8], puis généralisé par K.C. Kiwiel dans [14]. A noter également une extension de cet algorithme au cas sans dérivée [15].

Une implémentation de ces méthodes est en accès libre¹ sous la forme d'un package Matlab HANSO (Hybrid Algorithm for Non-Smooth Optimization). Sans entrer dans les détails, les algorithmes utilisés dans HANSO mettent en œuvre en première phase un algorithme de type BFGS (Broyden-Fletcher-Goldfarb-Shannon). Cet algorithme, appliqué à des problèmes d'optimisation non lisses, se comporte relativement bien en pratique : la convergence est rapide et conduit souvent à une approximation raisonnablement satisfaisante de l'optimum. Ce phénomène a tout d'abord été observé par plusieurs auteurs : C. Lemaréchal en 1982 [17], L. Lukšan et J. Vlček en 1999 et 2001 [21, 29] avant d'être étudié de façon plus approfondie par A. Lewis et M. Overton [20] très récemment. L'idée est que les fonctions localement lipschitziennes sont différentiables presque partout et qu'en pratique les itérés ne tombent pas sur les points de non-différentiabilité. C'est pour cela que la plupart des codes existants applique en première phase un algorithme de type BFGS, avant d'utiliser des méthodes plus sophistiquées de type faisceaux ou échantillonnage de gradients.

7.2 Méthodes de sous-gradients

Sachant que le sous-différentiel n'est souvent pas connu dans son intégralité, supposons que l'on dispose de l'oracle suivant :

Pour tout $x \in \mathbb{R}^n$, on sait calculer la valeur $f(x)$ de la fonction objectif f en x et au moins un sous-gradient $g \in \partial f(x)$.

Cet oracle fournira dans les algorithmes des candidats à la descente (et non pas nécessairement des directions de descente.

1. <http://www.cs.nyu.edu/faculty/overton/software/hanso>

Dans cette partie, on s'intéresse aux méthodes de premier ordre appliquées à la classe de problèmes suivante :

$$\min_{x \in X} f(x) \tag{7.1}$$

où $X \subset \mathbb{R}^n$ est un ensemble convexe, $f : X \rightarrow \mathbb{R}$ est convexe et f est Lipschitz de constante L sur X . On note x^* un minimiseur du problème (7.1). Notons que ce problème n'a pas de raison particulière d'admettre un minimiseur unique.

7.2.1 Cas sans contrainte

Algorithm 1: Algorithme de sous-gradient dans le cas sans contrainte.

Input:

N : un nombre d'itérations.

$x_0 \in X$: un point de départ.

Output:

x_N : une solution approchée.

begin

for k allant de 0 à N **do**

 Calculer $g_k \in \partial f(x_k)$.

 Recherche linéaire : chercher un pas $s_k > 0$ tel que :

$$f\left(x_k - s_k \frac{g_k}{\|g_k\|}\right) < f(x_k).$$

$$x_{k+1} = x_k - s_k \frac{g_k}{\|g_k\|}.$$

On remarquera que le sous-gradient est ici normalisé, contrairement à la descente de gradient. Il est facile de comprendre l'intérêt de cette normalisation de façon géométrique. Dans le cas différentiable, le gradient tend vers 0 lorsqu'on se rapproche du minimiseur. Dans le cas non différentiable, ce n'est pas le cas. Par exemple, la fonction $f(x) = |x|$ satisfait $\partial f(x) = \text{signe}(x)$ pour $x \neq 0$. Le sous-gradient est donc de norme 1 même très près du minimiseur.

On rappelle également que dans le cas non-différentiable, une direction opposée à un sous-gradient n'est pas nécessairement une direction de descente. En conséquence, la recherche linéaire peut échouer ! Bien que l'algorithme 2 ne soit pas de descente, un choix adéquat des s_k rend malgré tout la méthode convergente.

Théorème 7.1

1. Si la suite $(s_k)_k$ vérifie :

$$\lim_{k \rightarrow +\infty} s_k = 0, \quad \sum_{k=0}^{+\infty} s_k = +\infty,$$

alors : $\liminf f(x_k) = f(x^*)$. Si, de plus, l'ensemble des minimiseurs de f est borné, alors : $\lim_{k \rightarrow +\infty} f(x_k) = f(x^*)$.

2. Si la suite $(s_k)_k$ vérifie :

$$\sum_{k=0}^{+\infty} s_k = +\infty, \quad \sum_{k=0}^{+\infty} s_k^2 < +\infty,$$

alors la suite $(x_k)_{k \in \mathbb{N}}$ converge vers un minimiseur de f .

Les deux conditions $\lim_{k \rightarrow +\infty} s_k = 0$ et $\sum_{k=0}^{+\infty} s_k = +\infty$ indiquent que les pas doivent tendre vers 0, mais pas trop rapidement. Par exemple, une décroissance en $O\left(\frac{1}{k}\right)$ est trop rapide car la série $\sum \frac{1}{k^2}$ converge. Par contre une décroissance en $O\left(\frac{1}{\sqrt{k}}\right)$ satisfait les deux conditions. Si la suite $(s_k)_{k \in \mathbb{N}}$ ne tendait pas vers 0, on observerait dans beaucoup de cas des oscillations autour du minimum et pas de convergence. Par exemple, il est facile de voir que la méthode de descente de sous-gradient à pas constant ne converge pas si on l'applique à la fonction $f(x) = |x|$.

7.2.2 Méthodes de sous-gradient projeté

Revenons maintenant au problème :

$$\min_{x \in X} f(x).$$

On note $P_X(x)$ la projection orthogonale du point $x \in \mathbb{R}^n$ sur le domaine admissible X . Comme l'ensemble X est convexe, la projection est univaluée. Notons que la projection est facile à calculer dans le cas de contraintes de type "boite" mais très difficile à calculer dans la plupart des cas.

Théorème 7.2 On note $f_k^* = \min_{k \in \{0, \dots, k\}} f(x_k)$ et R le diamètre de l'ensemble X . Alors :

$$f_k^* - f^* \leq L \frac{R^2 + \sum_{i=0}^k h_i^2}{2 \sum_{i=0}^k h_i}.$$

En particulier si $h_k = \frac{R}{\sqrt{N+1}}$, on a un taux "optimal" et :

$$f_k^* - f^* \leq \frac{LR}{\sqrt{N+1}}.$$

Algorithm 2: Algorithme de sous-gradients dans le cas sous contrainte.

Input:

N : un nombre d'itérations.

$x_0 \in X$: un point de départ.

Output:

x_N : une solution approchée.

begin

for k allant de 0 à N **do**

 Calculer $g_k \in \partial f(x_k)$.

 Recherche linéaire : chercher un pas $s_k > 0$ tel que :

$$f\left(x_k - s_k \frac{g_k}{\|g_k\|}\right) < f(x_k).$$

$x_{k+1} = p_X\left(x_k - s_k \frac{g_k}{\|g_k\|}\right)$.

Démonstration. On commence par montrer que le fait que f soit Lipschitz implique que les sous-gradients de f sont bornés. On a $\forall (x, y) \in X^2$:

$$|f(x) - f(y)| \leq L\|x - y\|$$

or par définition du sous-différentiel :

$$f(y) \geq f(x) + \langle \eta, y - x \rangle$$

où $\eta \in \partial f(x)$. En combinant ces deux inégalités, on obtient :

$$|\langle \eta, y - x \rangle| \leq L\|x - y\|, \quad \forall y \in X.$$

En particulier pour $y = \eta + x$, on obtient $\|\eta\| \leq L$. Les sous-gradients ont donc une norme majorée par L .

Contrairement aux descentes de gradient, la preuve de convergence ne repose pas ici sur la décroissance monotone de la fonction coût, mais sur le fait que la distance au minimiseur diminue. On a :

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \left\|x_k - x^* - h_k \frac{\eta_k}{\|\eta_k\|}\right\|^2 \\ &= \|x_k - x^*\|^2 + h_k^2 - 2h_k \left\langle x_k - x^*, \frac{\eta_k}{\|\eta_k\|} \right\rangle. \end{aligned}$$

Or $f(x^*) \geq f(x_k) + \langle \eta_k, x^* - x_k \rangle$ (par définition du sous-gradient). Donc

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 + h_k^2 - 2 \frac{h_k}{\|\eta_k\|} (f(x_k) - f(x^*)).$$

En sommant de $k = 0$ à N et en utilisant le fait que $\|\eta\| \leq L$ on obtient :

$$\|x_{N+1} - x^*\|^2 \leq \|x_0 - x^*\|^2 + \sum_{k=0}^N h_k^2 - \frac{2h_k}{L}(f(x_k) - f(x^*))$$

D'où :

$$\begin{aligned} 2 \sum_{k=0}^N \frac{h_k}{L}(f(x_k) - f(x^*)) &\leq -\|x_{N+1} - x^*\|^2 + \|x_0 - x^*\|^2 + \sum_{k=0}^N h_k^2 \\ &\leq R^2 + \sum_{k=0}^N h_k^2. \end{aligned}$$

Ce qui permet de conclure :

$$\min_{k \in \{0, \dots, N\}} f(x_k) - f(x^*) \leq \frac{R^2 + \sum_{k=0}^N h_k^2}{2 \sum_{k=0}^N \frac{h_k}{L}}.$$

□

7.2.3 Application aux problèmes duaux

Nous nous intéressons dans cette partie à des problèmes d'optimisation avec contraintes fonctionnelles :

$$(P) \quad \min_{x \in \mathbb{R}^n} f(x) \quad \text{sous :} \quad \begin{aligned} g(x) &\leq 0 \\ h(x) &= 0 \end{aligned}$$

Exercice 7.2.1 (Méthode de sous-gradients pour le problème dual)

1. Question préliminaire : soit $X = \{(\lambda, \mu) \in \mathbb{R}^q \times \mathbb{R}^p \mid \lambda \geq 0\}$. Donner l'expression de la projection $p_X : \mathbb{R}^q \times \mathbb{R}^p \rightarrow \mathbb{R}^q \times \mathbb{R}^p$ sur X .
2. A l'aide du Lagrangien, écrire le problème dual du problème (P) sous forme standard. On note f^* la fonction duale associée.
3. Calculer le sous-différentiel de $(-f^*)$ et en déduire un sous-gradient de la fonction $(-f^*)$.
4. On note x_k la solution optimale du problème :

$$\min_{x \in \mathbb{R}^n} L(x; \lambda_k, \mu_k).$$

Décrire explicitement une itération de la méthode de sous-gradient appliquée au problème dual.

Algorithm 3: Algorithme de sous-gradient pour le problème dual.

Input:

N : un nombre d'itérations.

$x_0 \in X$: un point de départ.

Output:

x_N : une solution approchée.

begin

for k allant de 0 à N **do**

Calculer x_k solution de : $\min_{x \in \mathbb{R}^n} L(x; \lambda_k, \mu_k)$.

Direction de recherche :

$$d_k = \begin{bmatrix} g(x_k) \\ h(x_k) \end{bmatrix}, \quad \gamma_k = \frac{1}{\|d_k\|}.$$

Recherche linéaire : chercher un pas $s_k > 0$ tel que :

$$f(x_k + s_k d_k) < f(x_k).$$

$$\lambda_{k+1} = \max(0, \lambda_k + s_k \gamma_k g(x^k)).$$

$$\mu_{k+1} = \mu_k + s_k \gamma_k h(x^k).$$

L'approche duale est particulièrement intéressante pour des problèmes de grande taille (le nombre de variables duales étant le nombre de contraintes dans le primal). Le point clé de ces approches est de savoir calculer efficacement la fonction duale. En pratique, ce calcul peut poser problème, en particulier quand la solution du problème $\min_x L(x; \lambda_k, \mu_k)$ n'est pas unique. Cette situation arrive typiquement en programmation linéaire.

Théorème 7.3 *Supposons que la suite $(s_k)_{k \in \mathbb{N}}$ est décroissante et vérifie :*

$$\lim_{k \rightarrow +\infty} s_k = 0 \quad \text{et} \quad \sum_{k=1}^{+\infty} s_k = +\infty.$$

Alors tout point d'accumulation de la suite (\bar{x}_k) définie par :

$$\bar{x}_k = \frac{\sum_{i=1}^k \tau_i x_i}{\sum_{i=1}^k \tau_i}, \quad k \in \mathbb{N}^*,$$

est une solution optimale du problème (P).

7.3 Opérateur proximal

Considérons le problème

$$\min_{x \in \mathbb{R}^n} f(x) \tag{7.2}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe propre. Nous introduisons dans ce paragraphe l'opérateur proximal qui va nous permettre de caractériser les solutions optimales du problème (7.2).

7.3.1 Définition et propriétés

On introduit l'opérateur proximal défini de la façon suivante :

Définition 7.1 (Opérateur proximal) *L'opérateur proximal de f est noté prox_f et il est défini pour tout $x \in \mathbb{R}^n$ par :*

$$\text{prox}_f(x) = \arg \min_{u \in \mathbb{R}^n} f(u) + \frac{1}{2} \|u - x\|_2^2. \quad (7.3)$$

L'opérateur proximal peut être vu comme une généralisation de la notion de projection sur un ensemble convexe. Prenons par exemple le cas de la fonction indicatrice χ_X d'un ensemble X convexe fermé :

$$f(x) = \chi_X(x) = \begin{cases} 0 & \text{si } x \in X \\ +\infty & \text{sinon,} \end{cases}$$

L'opérateur proximal associé est exactement la projection euclidienne de x sur X .

Exercice 7.3.1 *Voici quelques exemples d'opérateurs proximaux :*

1. Si $f(x) = 0$, $\text{prox}_f(x) = x$.
2. Si $f(x) = \alpha \|x\|_1$, prox_f est un seuillage doux de x :

$$(\text{prox}_f(x))_i = \begin{cases} x_i - \alpha & \text{si } x_i \geq \alpha \\ 0 & \text{si } |x_i| \leq \alpha \\ x_i + \alpha & \text{si } x_i \leq -\alpha. \end{cases} \quad (7.4)$$

Proposition 7.1 *Si $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe fermée, $\text{prox}_f(x)$ existe et est unique pour tout $x \in \mathbb{R}^n$. De plus il est caractérisé par les équations suivantes :*

$$p = \text{prox}_f(x) \Leftrightarrow x - p \in \partial f(p)$$

Démonstration. En exercice. □

Ces opérateurs ont des propriétés très intéressantes qui les rend particulièrement adaptés pour la mise en place d'algorithmes de minimisation. En particulier, on peut démontrer que les points de minimum global de f sont exactement les points fixes de l'opérateur proximal associé à f :

Proposition 7.2 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe propre et $t > 0$. Alors :

$$\bar{x} \in \arg \min_{x \in \mathbb{R}^n} f(x) \iff \bar{x} = \text{prox}_{t f}(\bar{x}).$$

Démonstration. En exercice. \square

Proposition 7.3 (Contraction et expansivité) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe fermée. Alors prox_f est un opérateur fortement non expansif :

$$\langle \text{prox}_f(x_1) - \text{prox}_f(x_2), x_1 - x_2 \rangle \geq \|\text{prox}_f(x_1) - \text{prox}_f(x_2)\|_2^2. \quad (7.5)$$

Il est Lipschitz continu de constante 1 :

$$\|\text{prox}_f(x_1) - \text{prox}_f(x_2)\|_2 \leq \|x_1 - x_2\|_2. \quad (7.6)$$

Démonstration. On a :

$$x_1 - \text{prox}_f(x_1) \in \partial f(\text{prox}_f(x_1)) \quad (7.7)$$

et

$$x_2 - \text{prox}_f(x_2) \in \partial f(\text{prox}_f(x_2)) \quad (7.8)$$

or si f est convexe fermée, l'opérateur multivalué ∂f est monotone, ce qui signifie que :

$$\langle \eta_1 - \eta_2, u_1 - u_2 \rangle \geq 0, \quad \forall \eta_1 \in \partial f(\text{prox}_f(x_1)), \forall \eta_2 \in \partial f(\text{prox}_f(x_2)). \quad (7.9)$$

(Cette inégalité permet d'exprimer le fait que dans le cas C^2 , la hessienne d'une fonction convexe est semi-définie positive). L'inégalité (7.6) est obtenue en appliquant le théorème de Cauchy-Schwartz. \square

Proposition 7.4 (Identité de Moreau) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe fermée. On a alors :

$$x = \text{prox}_f(x) + \text{prox}_{f^*}(x), \quad (7.10)$$

où f^* désigne la transformée de Legendre-Fenchel de f :

$$f^*(s) = \sup_{x \in \mathbb{R}^n} \langle s, x \rangle - f(x).$$

Démonstration. On note $u = \text{prox}_f(x)$ et $v = x - u$. Le vecteur u satisfait $0 \in \partial f(u) + x - u$ soit encore $v \in \partial f(u)$. On a donc $u \in \partial f^*(v)$ soit encore $x - v \in \partial f^*(v)$ ou $v = \text{prox}_{f^*}(x)$. \square

La fonction conjuguée, aussi appelée transformée de Fenchel ou transformée de Legendre-Fenchel est utilisée pour :

- convexifier une fonction (en calculant la bi-conjuguée, i.e. la conjuguée de la conjuguée).
- calculer le sous-différentiel d'une fonction convexe.
- calculer des problèmes "duaux". Ces problèmes apportent souvent beaucoup d'informations sur les problèmes "primaux", i.e. ceux que l'on souhaite résoudre.
- passer de la mécanique lagrangienne à la mécanique hamiltonienne,...

Elle a été introduite par Mandelbrojt en 1939 puis précisée et améliorée par Fenchel en 1949. Cette transformée généralise la transformation de Legendre (1787). Cette notion est illustrée sur la figure 7.2.

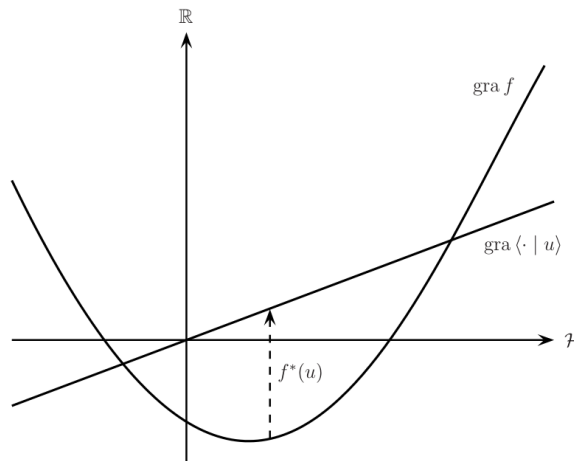


FIGURE 7.2 – $f^*(u)$ est le supremum de la différence signée verticale entre le graphe de f et celui de l'hyperplan défini par la fonction linéaire $\langle \cdot, u \rangle$.

Exemple 7.3.1 Soit L un sous-espace vectoriel de \mathbb{R}^n et L^\perp son complément orthogonal. La propriété précédente n'est alors rien d'autre que la décomposition orthogonale :

$$x = P_L(x) + P_{L^\perp}(x). \tag{7.11}$$

7.3.2 Algorithme du point proximal

Rappelons que d'après la proposition 7.2, les points de minimum global de f sont les points fixes de l'opérateur proximal associé. Le principe de l'algorithme du point proximal est alors le suivant : étant donné un point initial $x_0 \in \mathbb{R}^n$ et un pas $t > 0$, on définit récursivement :

$$x_{k+1} = \text{prox}_{tf}(x_k). \tag{7.12}$$

Exercice 7.3.2 Convergence de l'algorithme du point proximal.

Soit \bar{x} un point de minimum global de f . Le but de cet exercice est de démontrer que la suite $(x_k)_{k \in \mathbb{N}}$ des itérés générés par l'algorithme du point proximal converge vers \bar{x} .

1. Ecrire la condition d'optimalité associée à l'itération (7.12).

2. Montrer que :

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

3. En déduire que l'algorithme du point proximal est un algorithme de descente et que :

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

4. Considérons maintenant la suite $(\|x_k - \bar{x}\|)_{k \in \mathbb{N}}$.

(a) Montrer que la suite $(\|x_k - \bar{x}\|)_{k \in \mathbb{N}}$ est décroissante, et donc bornée.

(b) En déduire que la suite $(x_k)_{k \in \mathbb{N}}$ converge vers \bar{x} .

7.4 Algorithme Forward-Backward : un algorithme de splitting

Dans cette partie, on considère le problème suivant :

$$\min_{x \in \mathbb{R}^n} f(x) = g(x) + h(x) \quad (7.13)$$

où :

- $g : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction convexe, différentiable à gradient L -Lipschitz :

$$\|\nabla g(x_1) - \nabla g(x_2)\| \leq L \|x_1 - x_2\|, \quad \forall (x_1, x_2) \in \mathbb{R}^n \times \mathbb{R}^n. \quad (7.14)$$

- $h : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ est une fonction convexe, fermée.

On suppose que ce problème admet au moins une solution et on note X^* l'ensemble (convexe fermé) des minimiseurs. Par convexité, la condition nécessaire d'optimalité du problème (7.13) :

$$0 \in \partial h(x) + \nabla g(x),$$

est également suffisante et caractérise complètement les solutions du problème (7.13). Comme pour l'algorithme du point proximal, les minima globaux de la fonction f sont les points fixes d'un certain opérateur.

Proposition 7.5 Soit $t > 0$. On a :

$$0 \in \partial f(x) \iff x = \text{prox}_{th}(x - t\nabla g(x)).$$

Démonstration. En exercice. □

L'algorithme Forward-Backward (auss appelé algorithme de descente proximale) dans sa version simple est défini de la façon suivante :

Définition 7.2 (Algorithme Forward-Backward (FB))

$$x_{k+1} = \text{prox}_{th}(x_k - t\nabla g(x_k)). \quad (7.15)$$

Cet algorithme est un algorithme de descente qui consiste à effectuer alternativement un pas de descente de gradient (correspondant à un pas explicite sur la partie différentiable de f), suivi d'un pas proximal (correspondant à un pas implicite sur la partie non-différentiable de f). On peut démontrer la convergence de cet algorithme.

Théorème 7.4 *Soit $f = g + h$ avec g et h convexes, s.c.i. et bornées inférieurement. On suppose de plus que g est différentiable à gradient Lipschitz. Soit $x_0 \in \mathbb{R}^n$ un point initial arbitrairement choisi. Si $t < \frac{2}{L}$, alors la suite $(x_k)_{k \in \mathbb{N}}$ des itérés générés par l'algorithme (FB) à partir de x_0 converge vers un minimum global de f .*

Démonstration. En exercice. □

La fonction F n'est pas contractante en général, seulement non expansive. Les preuves de convergence linéaires qui reposent sur des itérés de contractions ne fonctionnent donc pas pour analyser ce schéma. De façon générale, on obtient tout de même des résultats de convergence sous-linéaire :

Théorème 7.5 (admis) *Le schéma de descente (7.15) assure que $f(x_k) - f(x^*) \leq \frac{L\|x_0 - x^*\|_2^2}{k}$ pour $t = \frac{1}{L}$.*

L'algorithme Forward-Backward a été utilisé dans de nombreux problèmes issus du traitement du signal et de l'image, et plus particulièrement pour résoudre des problèmes mettant en jeu une norme ℓ_2 au carré plus une norme ℓ_1 . Dans ce cadre particulier, l'algorithme (FB) est aussi appelé *Iterative Shrinkage Thresholding Algorithm* (ISTA). A noter qu'il existe une version accélérée de cet algorithme proposée par A. Beck et M. Teboulle [3] en 2009 (*Fast Iterative Shrinkage Thresholding Algorithm* (FISTA)), ainsi que deux autres algorithmes de splitting très utilisés et étudiés actuellement : l'algorithme de Douglas-Rachford et l'algorithme ADMM (*Alternating Descent Method of Multiplier*).

Bibliographie

- [1] P.-A. Absil, R. Mahony, and B. Andrews. *Convergence of the iterates of descent methods for analytic cost functions*. SIAM Journal on Optimization, 16(2) :531–547, 2005.
- [2] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media, 2011.
- [3] A. Beck and M. Teboulle. *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*. SIAM journal on imaging sciences, 2(1) :183–202, 2009.
- [4] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.
- [5] M. Bierlaire. *Introduction à l’optimisation différentiable*. PPUR presses polytechniques, 2006.
- [6] J. Bonnans, J. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical optimization. Universitext*. Springer-Verlag, Berlin, second edition, 2006. *Theoretical and practical aspects*.
- [7] J. Burke, A. Lewis, and M. Overton. *Approximating subdifferentials by random sampling of gradients*. Mathematics of Operations Research, 27(3) :567–584, 2002.
- [8] J. Burke, A. Lewis, and M. Overton. *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*. SIAM Journal on Optimization, 15(3) :751–779, 2005.
- [9] E. W. Cheney and A. A. Goldstein. *Newton’s method for convex programming and Tchebycheff approximation*. Numerische Mathematik, 1 :253–268, 1959.
- [10] Y. Ermoliev. *Stochastic programming methods*. Nauka, Moscow, 7(136), 1976.
- [11] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. II, volume 306 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, 1993. *Advanced theory and bundle methods*.
- [12] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I : fundamentals, volume 305*. Springer Science & Business Media, 2013.
- [13] J. E. Kelley, Jr. *The cutting-plane method for solving convex programs*. Journal of the Society for Industrial and Applied Mathematics, 8 :703–712, 1960.
- [14] K. Kiwiel. *Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization*. SIAM Journal on Optimization, 18(2) :379–388, 2007.

- [15] K. Kiwiel. *A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization*. SIAM Journal on Optimization, 20(4) :1983–1994, 2010.
- [16] K. C. Kiwiel. *Methods of descent for nondifferentiable optimization, volume 1133 of Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1985.
- [17] C. Lemaréchal. *Numerical experiments in nonsmooth optimization*. In E. e. Nurminski, editor, *Progress in Nondifferentiable Optimization*, pages 61–84. International Institute for Applied Systems Analysis, 1982.
- [18] C. Lemaréchal. *Chapter VII Nondifferentiable optimization*. In *Optimization, volume 1 of Handbooks in Operations Research and Management Science*, pages 529–572. Elsevier, 1989.
- [19] C. Lemaréchal and F. Oustry. *Nonsmooth Algorithms to Solve Semidefinite Programs*. In L. El Ghaoui and S.-I. Niculescu, editors, *Advances in Linear Matrix Inequality Methods in Control*, chapter 3. SIAM, 2000.
- [20] A. Lewis and M. Overton. *Nonsmooth optimization via quasi-newton methods*. Mathematical Programming, 141(1) :135–163, 2013.
- [21] L. Lukšan and J. Vlček. *Globally convergent variable metric method for convex nonsmooth unconstrained minimization*. Journal of Optimization Theory and Applications, 102(3) :593–613, 1999.
- [22] M. M. Mäkelä. *Survey of bundle methods for nonsmooth optimization*. Optim. Methods Softw., 17(1) :1–29, 2002.
- [23] A. Nemirovskii and D. Yudin. *Problem complexity and method efficiency in optimization*. In *Discrete Mathematics (Original Russian : Nauka 1979)*. Wiley-Intersciences Series, 1983.
- [24] Y. Nesterov. *Gradient methods for minimizing composite functions*. Mathematical Programming, 140(1) :125–161, 2013.
- [25] J. Nocedal and S. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.
- [26] B. Polyak. *Subgradient methods : a survey of Soviet research*. In *Nonsmooth optimization (Proc. IIASA Workshop, Laxenburg, 1977)*, volume 3 of IIASA Proc. Ser., pages 5–29. Pergamon, Oxford-New York, 1978.
- [27] R. T. Rockafellar. *Convex analysis*. Princeton university press, 2015.
- [28] N. Z. Shor. *Minimization methods for nondifferentiable functions, volume 3 of Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1985. *Translated from the Russian by K. C. Kiwiel and A. Ruszczyński*.
- [29] J. Vlček and L. Lukšan. *Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization*. Journal of Optimization Theory and Applications, 111(2) :407–430, 2001.
- [30] C. Zalinescu. *Convex analysis in general vector spaces*. World Scientific, 2002.
- [31] J. Zowe. *Nondifferentiable optimization*. In *Computational mathematical programming (Bad Windsheim, 1984)*, volume 15 of NATO Advanced Science Institutes Series F : Computer and Systems Sciences, pages 323–356. Springer, Berlin, 1984.