



DÉPARTEMENT GÉNIE MATHÉMATIQUE ET  
MODÉLISATION

4ÈME ANNÉE, 2020-2021.

MÉTHODES NUMÉRIQUES POUR L'OPTIMISATION  
NON LINÉAIRE DÉTERMINISTE.

Aude RONDEPIERRE



# Table des matières

<b>I</b>	<b>Outils théoriques</b>	<b>7</b>
<b>1</b>	<b>Eléments d'analyse convexe</b>	<b>9</b>
1.1	Ensembles convexes . . . . .	9
1.1.1	Définition et premiers exemples . . . . .	9
1.1.2	Cône convexe et cône dual : définitions et exemples . . . . .	11
1.2	Fonctions convexes . . . . .	14
1.2.1	Définitions et propriétés géométriques élémentaires . . . . .	15
1.2.2	Caractérisation différentielle de la convexité . . . . .	17
1.3	Eléments d'analyse pour l'algorithmique . . . . .	18
1.3.1	Fonctions à gradient Lipschitz . . . . .	19
1.3.2	Fonctions fortement convexes . . . . .	20
1.3.3	Conditionnement d'une fonction . . . . .	21
<b>2</b>	<b>Introduction à l'optimisation numérique</b>	<b>23</b>
2.1	Définitions et classification des problèmes d'optimisation . . . . .	23
2.2	Résultats d'existence et d'unicité en optimisation . . . . .	26
2.2.1	Cas où l'ensemble $X$ des contraintes est borné . . . . .	26
2.2.2	Cas où l'ensemble $X$ des contraintes est non borné . . . . .	26
2.2.3	Cas particulier de contraintes d'égalités et/ou d'inégalités . . . . .	27
2.3	Rôle de la convexité en optimisation . . . . .	28
<b>II</b>	<b>Conditions d'optimalité et dualité</b>	<b>29</b>
<b>3</b>	<b>Conditions d'optimalité</b>	<b>31</b>
3.1	Cas général : condition d'optimalité géométrique . . . . .	31
3.1.1	Le cône tangent ou comment se déplacer dans le domaine des contraintes	31
3.1.2	Condition nécessaire d'optimalité géométrique . . . . .	32
3.2	Cas où le domaine des contraintes est un ouvert . . . . .	33
3.3	Cas différentiable avec contraintes fonctionnelles . . . . .	36
<b>4</b>	<b>Dualité Lagrangienne</b>	<b>39</b>
4.1	Problème primal - Problème dual . . . . .	40
4.2	Théorèmes de dualité et points selle du Lagrangien . . . . .	41

4.3	Cas particulier de la programmation linéaire . . . . .	43
4.3.1	Généralités . . . . .	44
4.3.2	Dualité en programmation linéaire . . . . .	47
4.4	Application à la dualité conique . . . . .	48
4.4.1	Conditions de KKT en optimisation conique . . . . .	49
4.4.2	Dualité lagrangienne en optimisation conique . . . . .	50
4.4.3	Cas particulier de la dualité SDP . . . . .	51
<b>III</b>	<b>Algorithmes classiques en optimisation différentiable</b>	<b>55</b>
<b>5</b>	<b>Méthodes de descente en optimisation différentiable sans contrainte</b>	<b>57</b>
5.1	Principe général des méthodes de descente . . . . .	57
5.2	La recherche linéaire . . . . .	60
5.2.1	Recherches linéaires élémentaires . . . . .	60
5.2.2	Condition de Armijo ou comment éviter les pas trop grands . . . . .	61
5.2.3	Conditions de Wolfe ou comment éviter les pas trop petits . . . . .	63
5.2.4	Convergence globale des algorithmes avec pas de Wolfe . . . . .	66
5.3	Algorithmes de type gradient . . . . .	67
5.3.1	CS pour qu'un algorithme de gradient soit une méthode de descente . . . . .	67
5.3.2	Résultats de convergence globale . . . . .	68
5.3.3	Tests numériques : pas fixe/pas optimal vs Pas de Wolfe . . . . .	68
5.4	Méthodes de type Newton . . . . .	71
5.4.1	Méthode de Newton avec recherche linéaire . . . . .	72
5.4.2	Méthodes de quasi-Newton . . . . .	74
5.4.3	Propriétés et convergence de l'algorithme BFGS . . . . .	77
5.5	Algorithmes pour la régression non-linéaire . . . . .	78
5.5.1	Problèmes de moindres carrés linéaires (rappels) . . . . .	78
5.5.2	Algorithme de Gauss-Newton . . . . .	79
5.5.3	Algorithme de Levenberg-Marquardt . . . . .	80
<b>6</b>	<b>Algorithmes pour l'optimisation différentiable sous contrainte</b>	<b>83</b>
6.1	Méthode du gradient projeté . . . . .	83
6.2	Algorithmes newtoniens - Méthode SQP . . . . .	85
6.3	Méthodes de pénalisation . . . . .	88
6.3.1	Principe général . . . . .	89
6.3.2	Le Lagrangien augmenté . . . . .	90

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, 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 de les analyser et de les minimiser efficacement. De nombreux problèmes non convexes 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 [4] 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 [7, 10, 6] sont de très bonnes références pour découvrir les nombreux détails de l'analyse convexe en dimension finie. Au niveau algorithmique, ce cours est essentiellement inspiré de la référence [8].

**Remerciements à Denis Arzelier pour ses notes de cours et son aide à préparer ce polycopié. Certains passages de ce polycopié sont directement extraits de ses propres cours.**



# Première partie

## Outils théoriques





# Chapitre 1

## Eléments d'analyse convexe

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

### 1.1 Ensembles convexes

#### 1.1.1 Définition et premiers exemples

**Définition 1.1 (Ensemble convexe)** *Un ensemble  $X \subseteq \mathbb{R}^n$  est dit convexe si :*

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

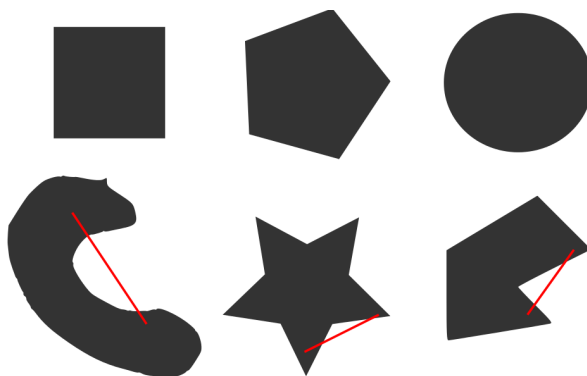


FIGURE 1.1 – En haut : quelques exemples d'ensembles convexes en 2 dimensions. En bas : quelques exemples d'ensembles non convexes (il existe des segments dont les extrémités appartiennent à l'ensemble, qui ne sont pas entièrement contenus dans les ensembles).

La propriété suivante est une conséquence immédiate de cette définition :

**Lemme 1.1** Soient  $X_1$  et  $X_2$  deux ensembles convexes dans  $\mathbb{R}^n$ . Alors :

1.  $X_1 \cap X_2$  est convexe.
2.  $X_1 \oplus X_2 = \{x_1 + x_2 \mid x_1 \in X_1, x_2 \in X_2\}$  est convexe.
3.  $X_1 \ominus X_2 = \{x_1 - x_2 \mid x_1 \in X_1, x_2 \in X_2\}$  est convexe.

**Exemple 1.1.1 (Quelques ensembles convexes)**

- $X = \{x \in \mathbb{R}^3 \mid x_1 + 2x_2 - x_3 \leq 4, 2x_1 - x_2 + x_3 \leq 6\}$  est convexe comme intersection de deux demi-espaces. Plus généralement, les ensembles de la forme  $\{x \in \mathbb{R}^n \mid Ax \preceq b, Cx = d\}$  (définis par un nombre fini d'égalités et d'inégalités linéaires) sont des ensembles convexes, appelés polyèdres convexes. On appelle polytope tout polyèdre convexe borné.
- $X = \{(x, y) \in \mathbb{R}^2 \mid y \geq |x|\}$  est un cône convexe dans  $\mathbb{R}^2$ .

**Définition 1.2 (Combinaison convexe)** Soient  $x_1, \dots, x_m \in \mathbb{R}^n$ . On dit que  $x$  est combinaison convexe de ces points s'il existe des réels  $(\alpha_1, \dots, \alpha_m)$  tels que :

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

**Définition 1.3 (Enveloppe convexe)** Soit  $X \subseteq \mathbb{R}^n$  un ensemble. On appelle enveloppe convexe de  $X$ , notée  $\text{conv}(X)$ , le plus petit ensemble convexe contenant  $X$ . En dimension finie, c'est l'ensemble des combinaisons convexes d'éléments de  $X$  :

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

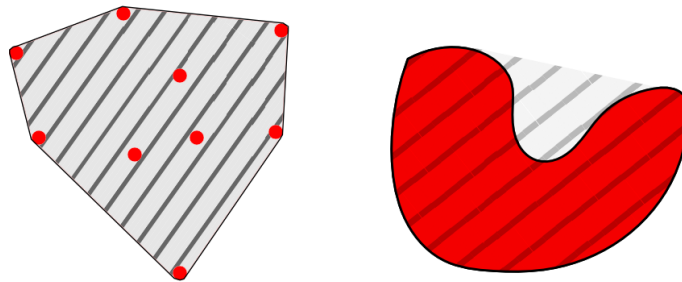


FIGURE 1.2 – Exemples d'enveloppes convexes. À gauche : enveloppe convexe d'un ensemble discret. À droite : enveloppe convexe d'un ensemble continu.

### 1.1.2 Cône convexe et cône dual : définitions et exemples

**Définition 1.4 (Cône)** L'ensemble  $K \subseteq \mathbb{R}^n$  est un cône si :

$$\forall x \in K, \forall \lambda > 0, \lambda x \in K.$$

De plus :

1.  $K$  est appelé **cône convexe** si  $K$  est un cône et si  $K$  est convexe.
2.  $K$  est appelé **cône pointé** si  $K$  est un cône et si  $K$  ne contient pas de droite vectorielle i.e. :

$$K \cap (-K) \subset \{0\}.$$

3.  $K$  est appelé **cône solide** si  $K$  est un cône et si  $K$  est d'intérieur non vide.
4.  $K$  est appelé **cône propre** si  $K$  est un cône convexe fermé solide et pointé.

Tout cône propre  $K$  dans  $\mathbb{R}^n$  induit un ordre partiel sur  $\mathbb{R}^n$  qui permet de définir des inégalités généralisées :

**Définition 1.5 (Inégalités généralisées)** Soit  $K \subseteq \mathbb{R}^n$  un cône propre. Alors  $K$  induit un ordre partiel  $\succeq_K$  sur  $\mathbb{R}^n$  :

$$a \succeq_K b \Leftrightarrow a - b \in K.$$

Voici quelques exemples classiques de cônes et leurs liens avec l'optimisation :

- **L'orthant positif**  $K = \mathbb{R}_+^n$  est un cône propre et :

$$x \succeq_{\mathbb{R}_+^n} y \Leftrightarrow \forall i = 1, \dots, n, x_i \geq y_i.$$

Ce cône intervient dans les problèmes de programmation linéaire. En effet, un programme linéaire de la forme :

$$\begin{array}{ll} \max_{x \in \mathbb{R}^n} & c_1 x_1 + \dots + c_n x_n \\ \text{sous} & a_{11} x_1 + \dots + a_{1n} x_n \geq b_1 \\ & \vdots \\ & a_{m1} x_1 + \dots + a_{mn} x_n \geq b_m \end{array}$$

peut se réécrire sous la forme :

$$\begin{array}{ll} \max_{x \in \mathbb{R}^n} & c^\top x \\ \text{sous} & Ax \succeq_{\mathbb{R}_+^m} b, \end{array}$$

en posant :  $c = (c_1, \dots, c_n)^\top$ ,  $A = (a_{ij}) \in \mathbb{R}^{m \times n}$  et  $b = (b_1, \dots, b_m)^\top$ .

- **Le cône de Lorentz** (appelé *cône en cornet de glace* ou *cône du second ordre*) défini par :

$$\mathbb{L}^n = \{(x, z) \in \mathbb{R}^n \times \mathbb{R} \mid \|x\| \leq z\},$$

et représenté en dimension 2 sur la Figure 1.3, est un cône propre. Il intervient en programmation quadratique conique.

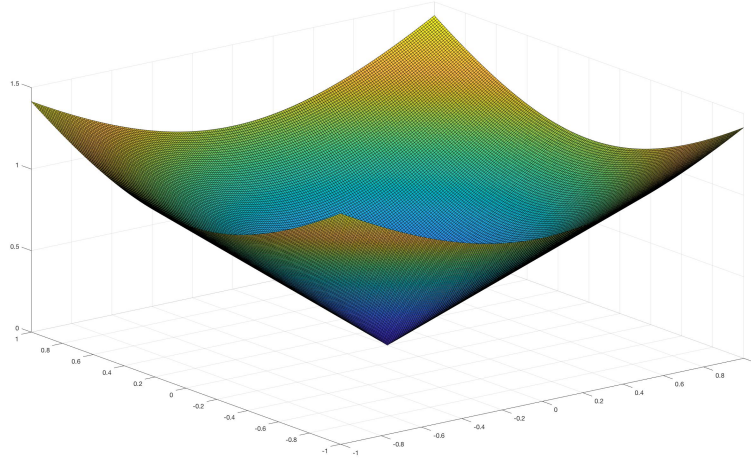


FIGURE 1.3 – Frontières du cône de Lorentz 3D :  $\mathbb{L}^2 = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 \leq z^2, z \geq 0\}$ .

- **Le cône SDP** ou cône des matrices symétriques semidéfinies positives :

$$\begin{aligned} \mathbb{S}_n^+ &= \{M \in \mathbb{R}^{n \times n} \mid M^\top = M, v^\top M v \geq 0, \forall v \in \mathbb{R}^n\}, \\ &= \{M \in \mathbb{R}^{n \times n} \mid M^\top = M, M \succeq 0\} \end{aligned}$$

est également un cône propre dont l'intérieur est l'ensemble des matrices symétriques définies positives :

$$\begin{aligned} \mathbb{S}_n^{++} &= \{M \in \mathbb{R}^{n \times n} \mid M^\top = M, v^\top M v > 0, \forall v \in \mathbb{R}^n, v \neq 0\}, \\ &= \{M \in \mathbb{R}^{n \times n} \mid M^\top = M, M \succ 0\} \end{aligned}$$

Ce cône induit un ordre partiel sur  $\mathbb{S}_n$ , appelé *ordre de Löwner*. On munit  $\mathbb{S}_n$  du produit scalaire de Frobenius :

$$\forall (A, B) \in \mathbb{S}_n \times \mathbb{S}_n, \langle A, B \rangle = \text{Tr}(AB) = \sum_{i,j=1}^n a_{ij}b_{ij}.$$

Voici quelques propriétés utiles du cône SDP :

**Lemme 1.2 (Cône SDP)**

1.  $A \succeq 0 \Leftrightarrow \forall B \succeq 0, \langle A, B \rangle \geq 0$ .
2.  $A \succ 0 \Leftrightarrow \forall B \succeq 0, B \neq 0, \langle A, B \rangle > 0$ .
3. Soient  $A$  et  $B$  deux matrices du cône  $\mathbb{S}_n^+$ . On a :

$$\langle A, B \rangle = 0 \implies AB = 0.$$

Ce cône intervient en programmation SDP i.e. pour des problèmes de la forme :

$$\min_{X \in \mathbb{S}_n} \langle C, X \rangle \quad \text{sous} \quad A(x) = b, \quad X \succeq 0,$$

où  $C \in \mathbb{S}_n$ ,  $A : \mathbb{S}_n \rightarrow \mathbb{R}^m$  est une application linéaire et  $b \in \mathbb{R}^m$ . Nous y reviendrons en détail dans la suite de ce cours.

**Exercice 1.1.1** Montrer que le cône SDP en dimension 2 est défini de la façon suivante :

$$\mathbb{S}_+^2 = \left\{ \begin{bmatrix} x & y \\ y & z \end{bmatrix} \mid x \geq 0, z \geq 0, xz \geq y^2 \right\}.$$

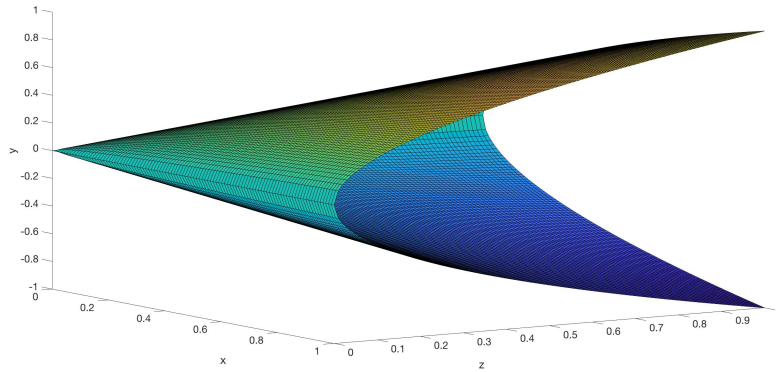


FIGURE 1.4 – Frontières du cône SDP en dimension 2 :  $\mathbb{S}_+^2$ .

**Définition 1.6 (Cône dual)** Soit  $X$  un sous-ensemble quelconque de  $\mathbb{R}^n$ . L'ensemble :

$$X^* = \{y \in \mathbb{R}^n \mid \langle x, y \rangle \geq 0, \forall x \in X\}$$

est appelé cône dual de l'ensemble  $X$ .

**Proposition 1.1 (Propriétés du cône dual)**

1. Soit  $X \subset \mathbb{R}^n$  quelconque. Alors  $X^*$  est un cône convexe.
2. Soit  $K$  un cône.
  - (a)  $K^*$  est un cône convexe fermé.
  - (b)  $K \subset K^{**}$  et  $K^{**}$  est la fermeture du plus petit cône convexe contenant  $K$ .
3. Si  $K$  est un cône convexe fermé non vide alors :  $K^{**} = K$ .
4. Soient  $K_1$  et  $K_2$  deux cônes dans  $\mathbb{R}^n$  tels que :  $K_1 \subset K_2$ . Alors :

$$K_2^* \subset K_1^*.$$

Etant donné un cône  $K$ , on peut ainsi définir une inégalité généralisée duale :

**Définition 1.7 (Inégalités généralisées duales)** Soit  $K \subseteq \mathbb{R}^n$  un cône propre. Alors :

$$x \preceq_K y \Leftrightarrow \forall \lambda \succeq_{K^*} 0, \langle \lambda, x \rangle \leq \langle \lambda, y \rangle.$$

Reprenons les cônes présentés précédemment :

- L'orthant positif et le cône S.D.P. sont auto-adjoints i.e. :

$$(\mathbb{R}_+^n)^* = \mathbb{R}_+^n, \quad (\mathbb{S}_n^+)^* = \mathbb{S}_n^+.$$

- Le cône de Lorentz admet pour dual le cône suivant :

$$(\mathbb{L}^n)^* = \{(u, v) \in \mathbb{R}^{n+1} \mid \|u\|_* \leq v\},$$

où  $\|\cdot\|_*$  désigne la norme duale associée à la norme  $\|\cdot\|$  :

$$\|u\|_* = \sup_{\|x\| \leq 1} \langle u, x \rangle.$$

## 1.2 Fonctions convexes

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. En effet, considérons un problème de la forme :  $\inf_{x \in X} f(x)$  où  $X$  est un sous-ensemble non vide de  $\mathbb{R}^n$ . On définit alors :

$$\bar{f}(x) = \begin{cases} f(x) & \text{si } x \in X, \\ +\infty & \text{sinon.} \end{cases}$$

On a alors :

$$\inf_{x \in X} f(x) = \inf_{x \in \mathbb{R}^n} \bar{f}(x)$$

Définissons la notion de domaine d'une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  :

**Définition 1.8 (Domaine d'une fonction)** *Le domaine d'une fonction  $f$  est noté  $\text{dom}(f)$  et 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 :  $\text{dom}(f) \neq \emptyset$ .

### 1.2.1 Définitions et propriétés géométriques élémentaires

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

**Définition 1.9 (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 1.5. 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.

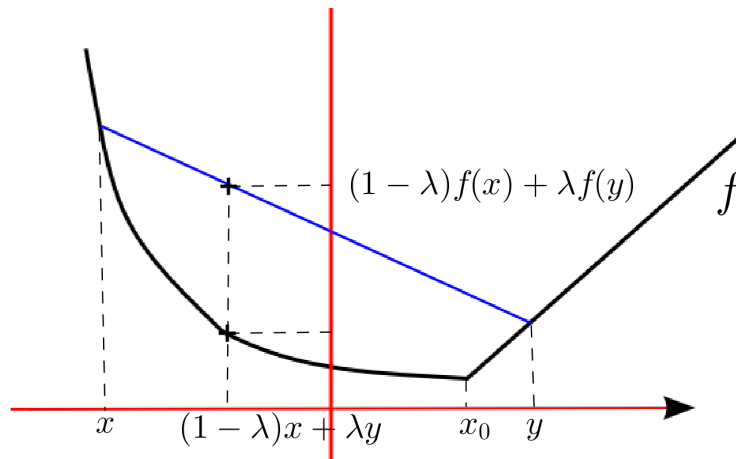


FIGURE 1.5 – Un exemple de fonction convexe non différentiable en  $x_0$

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

**Lemme 1.3 (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

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

**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 fausse ! 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 1.6.



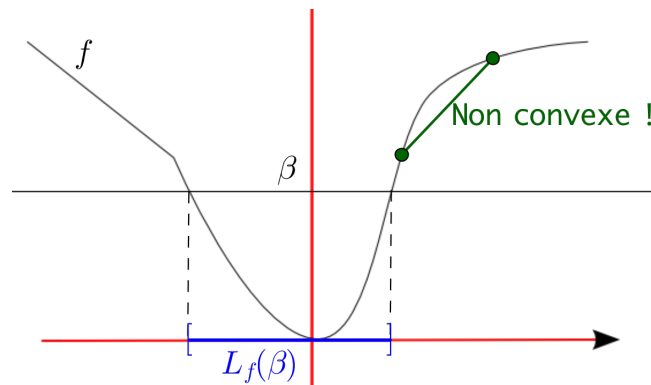


FIGURE 1.6 – Un exemple de fonction quasi-convexe. Ses lignes de niveaux sont des segments (donc des ensembles convexes), mais la fonction n'est pas convexe.

### Opérations sur les fonctions convexes

**Théorème 1.3** Soient  $f_1$  et  $f_2$  deux fonctions convexes et  $\beta > 0$ . Les fonctions suivantes sont convexes :

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  $\text{dom}(f) = \text{dom}(f_1) \cap \text{dom}(f_2)$ .

**Théorème 1.4** Soit  $\phi : \mathbb{R}^m \rightarrow \mathbb{R}$  une fonction convexe. Soit  $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$  un opérateur linéaire et  $b \in \mathbb{R}^m$ . Alors la fonction  $f : x \mapsto \phi(Ax + b)$  est convexe et  $\text{dom}(f) = \{x \in \mathbb{R}^n \mid Ax + b \in \text{dom}(\phi)\}$ .

**Démonstration.** Faite en cours. □

### 1.2.2 Caractérisation différentielle de la convexité

**Théorème 1.5** 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 1.7).

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

- (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.

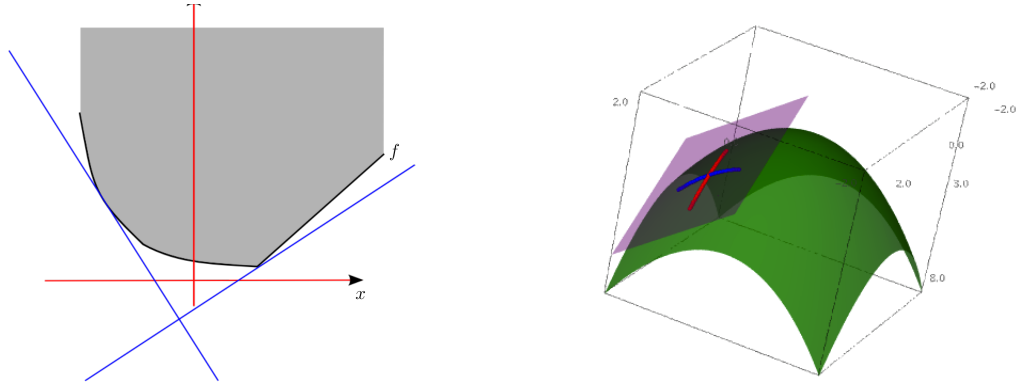


FIGURE 1.7 – Hyperplans tangents d'une fonction convexe (1D) et d'une fonction concave (2D). Notez que l'hyperplan tangent est un minorant pour la fonction convexe et un majorant pour la fonction concave.

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

**Proposition 1.2 (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 tout point).

- $f$  est strictement convexe ssi :

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

(i.e. la hessienne de  $f$  est définie positive en tout point).

## 1.3 Éléments d'analyse pour l'algorithmique

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  $C^\infty$  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  $L^\infty$ , par une fonction  $C^\infty$ . 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 :

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|. \quad (1.1)$$

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. Alors :

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

**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.
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  $f''(x) \leq \exp(b)$ .

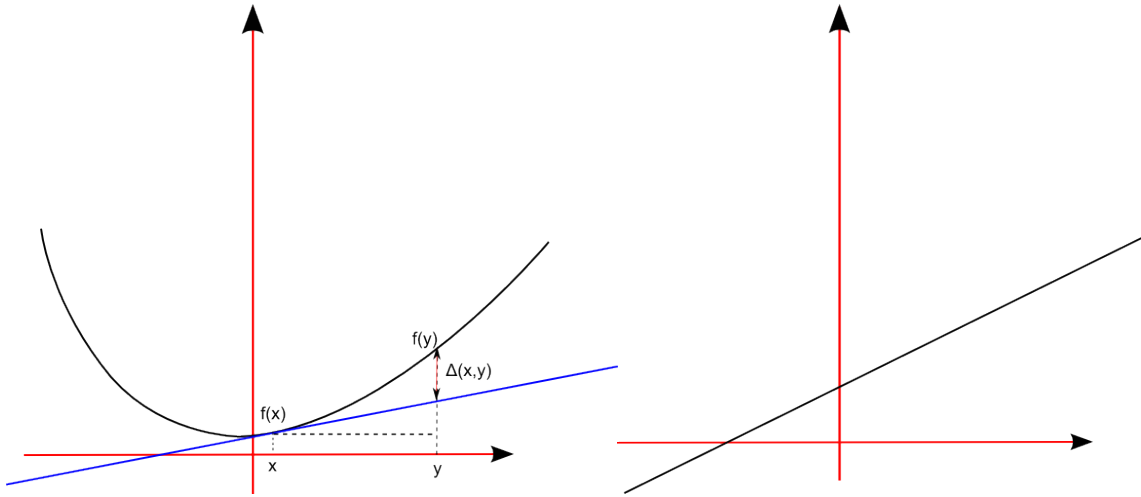


FIGURE 1.8 – Fonction à gradient Lipschitz : le graphe de la courbe ne peut pas s'éloigner trop rapidement de la tangente. Ici  $\Delta(x, y) \leq \frac{L}{2} \|y - x\|_2^2$ . La fonction affine de droite satisfait  $L = 0$  puisque la tangente est identique à la fonction.

### 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 l'hypothèse qu'on minimise une fonction à gradient Lipschitz. Une classe de fonctions sur laquelle on peut dire plus de choses est celle des fonctions fortement convexes.

**Définition 1.10** Une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  différentiable est dite *fortement convexe* s'il existe  $\mu > 0$  tel que :

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|y - x\|^2. \quad (1.2)$$

Notez que le signe dans l'inégalité (1.2) 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 1.3** Une fonction fortement convexe est strictement convexe.

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.

**Proposition 1.4** Une fonction  $C^2$  est  $\mu$ -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 permettant d'obtenir des taux de convergence en norme :

**Proposition 1.5** Soit  $f$  une fonction  $\mu$ -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 1.6** 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  $\alpha$  et  $\beta$  deux réels positifs. Alors la fonction  $f = \alpha f_1 + \beta f_2$  est  $\mu$ -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. La fonction  $f(x) = g(x) + \frac{1}{2} \|x - x_0\|^2$  où  $g$  est convexe, est 1 fortement convexe.
2. 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  $\lambda_{\min}(A^*A)$ .

### 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 1.11** Soit  $f$  une fonction  $\mu$ -fortement convexe, différentiable, telle que  $\nabla f$  est  $L$ -Lipschitz. Le conditionnement de  $f$  est défini par la quantité :

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

Le conditionnement apparaît 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 difficiles à résoudre que les problèmes bien conditionnés. La Figure 1.9 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  $f_\alpha$  est  $D_\alpha$  et le conditionnement est donc  $Q_{f_\alpha} = \alpha$ . Plus  $\alpha$  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.

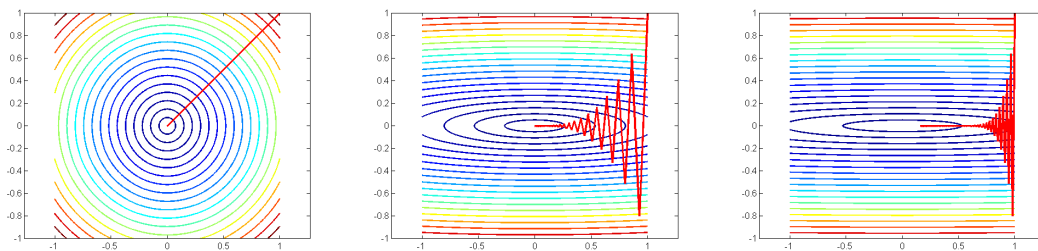


FIGURE 1.9 – Lignes de niveau de la fonction  $f_\alpha$  pour  $\alpha \in \{1, 5, 10\}$ . La courbe rouge indique la trajectoire d’une descente de gradient. Plus le conditionnement  $\alpha$  est élevé, plus les courbes de niveau sont allongées et plus la méthode de descente de gradient oscille.

# 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 Définitions et classification des problèmes d'optimisation

Soit  $X$  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

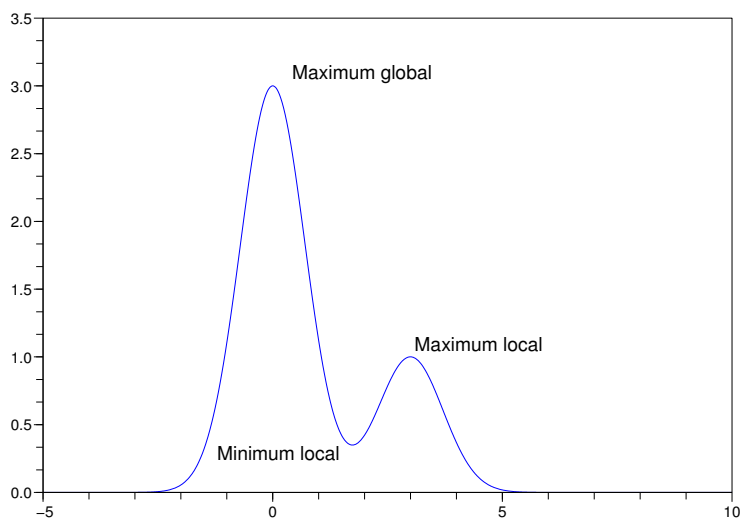


FIGURE 2.1 – Minima et maxima locaux et globaux de la fonction  $x \mapsto 3e^{-x^2} + e^{-(x-3)^2}$ .

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 un 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'algorithmes 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 (PL)
Linéaire $f(x) = c^\top x, c \in \mathbb{R}^n$	Inégalités matricielles linéaires $F_0 + \sum_{i=1}^n x_i F_i \prec 0, F_i \in \mathbb{S}_m$	Spectraèdre	Programmation SDP
Linéaire	linéaires	$X \subset \mathbb{Z}$	PL en nombres entiers
Quadratique $f(x) = c^\top x + x^\top Q x$ $c \in \mathbb{R}^n, Q \in \mathbb{S}_n$	linéaires	Polytope Polyèdre	Programmation quadratique
Convexe	convexes	convexe	Programmation convexe
qq	qq	qq	Programmation non linéaire

## 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 algos génétiques, algos évolutionnaires, recuit simulé, recherche tabou
	NON-DIFF.	Méthodes de sous-gradient, 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

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 :

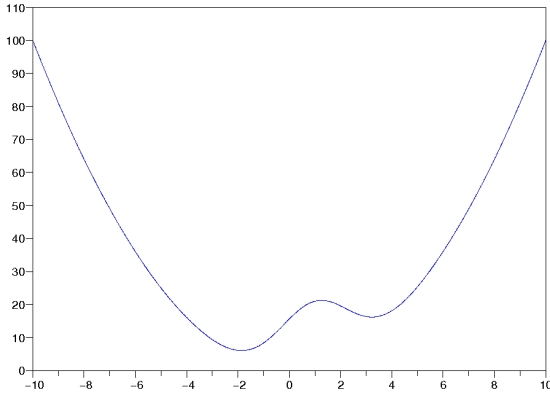
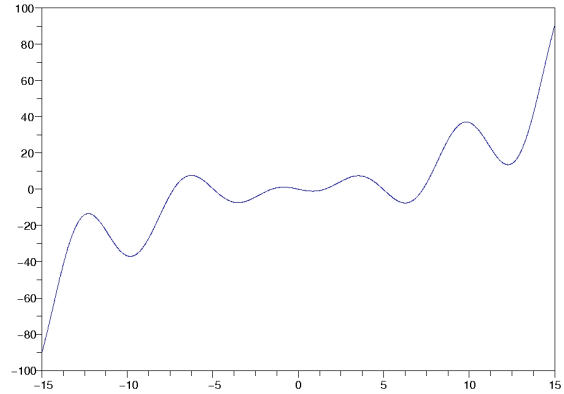
**Définition 2.2** *Une application  $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$  est dite infinie à l'infini (ou coercive) sur  $X$  si*

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

*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

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

**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 :

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

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 :

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

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

## 2.3 Rôle de la convexité en 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 nécessaire.

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 du problème :  $\min_{x \in X} f(x)$ .*

- i. *Si l’ensemble  $X$  est convexe et si la fonction  $f$  est convexe sur  $X$ , alors  $x^*$  est un point de minimum global.*
- ii. *Si l’ensemble  $X$  est convexe et si la fonction  $f$  est strictement convexe sur  $X$ , alors  $x^*$  est l’unique point de minimum global.*

# Deuxième partie

## Conditions d'optimalité et dualité



# 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 à  $X$  en  $x$ , noté  $T_x(X)$ , est l'ensemble des vecteurs  $d \in \mathbb{R}^n$  tels qu'il existe une suite  $(d_n)_{n \in \mathbb{N}}$  et une suite  $(\varepsilon_n)_{n \in \mathbb{N}}$  de réels strictement positifs telles que :

$$\lim_{n \rightarrow +\infty} d_n = d, \quad \lim_{n \rightarrow +\infty} \varepsilon_n = 0,$$

et

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

Autrement dit, le cône tangent 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 faire diminuer la valeur du critère 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 :*

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

*Autrement dit, aucune direction du cône tangent n'est une direction de descente de  $f$  en  $x$ .*

**Démonstration.** Faite en cours.

### 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.

**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 où le domaine des contraintes est un ouvert

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

Nous pouvons donc maintenant énoncer (sans démonstration) les conditions nécessaires du premier et du second ordre pour des problèmes d'optimisation différentiable sous contraintes définissant un ouvert :

On s'intéresse aux problèmes suivants :

$$\min_{x \in X} f(x) \quad \text{et} \quad \max_{x \in X} f(x),$$

où  $X$  est un **ouvert** de  $\mathbb{R}^n$ .

#### Conditions nécessaires d'optimalité locale

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

$$\begin{aligned} x^* &\in X && (x^* \text{ doit être admissible}) \\ \nabla f(x^*) &= 0 && (\text{CN d'optimalité du } 1^{er} \text{ ordre}) \\ H[f](x^*) &\succeq 0 && (\text{CN d'optimalité du } 2^{nd} \text{ ordre}) \\ &(\text{resp. } H[f](x^*) \preceq 0) \end{aligned}$$

#### Condition suffisante d'optimalité locale

Si :

- $x^* \in X$ ,
- $\nabla f(x^*) = 0$ ,
- $H[f](x^*)$  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 que  $X$  est un ouvert convexe non vide de  $\mathbb{R}^n$ .

1. Supposons que  $f$  est convexe (resp. concave) sur  $X$ . Alors  $x^*$  est un point de minimum (resp. maximum) global de  $f$  sur  $X$  si et seulement si :

$$x^* \in X \text{ et } \nabla f(x^*) = 0.$$

2. Si, de plus,  $f$  est strictement convexe (resp. concave) sur  $X$  et si  $x^*$  est un point de minimum (resp. maximum) local de  $f$  sur  $X$ , alors  $x^*$  est l'unique point de minimum (resp. maximum) global de  $f$ .

Les conditions d'optimalité pour les problèmes sans contrainte vues en 3ème année découlent naturellement de ces conditions en prenant  $X = \mathbb{R}^n$  :

Considérons les problèmes suivants :

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

### Conditions nécessaires d'optimalité locale

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

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

$$H[f](x^*) \succeq 0 \quad (\text{CN d'optimalité du } 2^{nd} \text{ ordre})$$

(resp.  $H[f](x^*) \preceq 0$ )

### Condition suffisante d'optimalité locale

Si :

- $\nabla f(x^*) = 0$ ,
- $H[f](x^*)$  symétrique et :  $H[f](x^*) \succ 0$  (resp.  $H[f](x^*) \prec 0$ ),

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

### Condition Suffisante d'optimalité globale

- i. Supposons  $f$  convexe (resp. concave) sur  $\mathbb{R}^n$ . Alors  $x^* \in \mathbb{R}^n$  est un point minimum (resp. maximum) global de  $f$  si et seulement si  $\nabla f(x^*) = 0$ .
- ii. Si, de plus,  $f$  est strictement convexe (resp. strictement concave), alors  $x^*$  est l'unique point de minimum (resp. maximum) 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 :

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

On dira que le problème  $(P)$  est convexe si  $h$  est affine, si  $g$  est convexe et si la fonction objectif  $f$  est convexe sur  $X$ . Dans la suite, nous noterons :

$$I(x) = \{j \in \{1, \dots, q\} \mid g_j(x) = 0\}$$

l'ensemble des indices des contraintes actives au point  $x$ .

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

L'essentiel de la démarche consiste 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. Dans ce cours, nous n'entrerons pas dans la théorie fine permettant d'expliciter et justifier l'expression du cône tangent. Nous nous contenterons de donner les grandes idées générales pour pouvoir ensuite mettre en oeuvre les algorithmes.

Introduisons le Lagrangien associé au problème  $(P)$  :

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

$$x \in \mathbb{R}^n, \lambda \in \mathbb{R}^p, \mu \in \mathbb{R}^q.$$

## Condition nécessaire d'optimalité du premier ordre

**Théorème 3.2 (CN d'optimalité de Karush-Kuhn-Tucker)** Soit  $x^* \in X$  un point admissible du problème (P). Supposons  $f$ ,  $g$  et  $h$  différentiables en  $x^*$ , que  $f$  admette un minimum local sur  $X$  au point  $x^*$  et que les vecteurs des contraintes actives :

$$\{\nabla h_i(x); i = 1, \dots, p\} \cup \{\nabla g_j(x); j \in I(x)\} \text{ sont linéairement indépendants.} \quad (3.4)$$

Alors il existe  $\lambda^* \in \mathbb{R}^p$  et  $\mu^* \in \mathbb{R}^q$  tels que :

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

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

## TERMINOLOGIE :

- On appelle *point stationnaire* ou *point de KKT* du problème (P) tout point  $\bar{x}$  vérifiant les conditions nécessaires d'optimalité du premier ordre du Théorème 3.2 pour un certain multiplicateur  $(\lambda, \mu)$ .
- Les relations :  $\mu_j g_j(\bar{x}) = 0$  sont appelées *relations de complémentarité* et signifient :

$$\text{Soit : } g_i(\bar{x}) < 0 \text{ et } \mu_j = 0, \quad \text{soit : } g_j(\bar{x}) = 0.$$

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

- La condition (3.4) est une condition suffisante de qualification des contraintes au point  $x^*$ . La qualification des contraintes peut être vue comme un critère permettant de sélectionner de "bonnes" représentations de  $X$  par  $g$ . 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. En pratique la qualification des contraintes est difficile à vérifier et couramment remplacée par des conditions suffisantes de qualification des contraintes. Il existe plusieurs conditions

suffisantes de qualification des contraintes utilisées dans la littérature (cf qualification de Slater, de Mangasarian-Fromovitz e.g.). Dans ce cours, nous travaillerons avec la plus simple à notre disposition, à savoir la condition (3.4) d'indépendance linéaire des contraintes actives.

### Condition nécessaire d'optimalité du second ordre

**Théorème 3.3 (Conditions d'optimalité du second ordre)** Soit  $x^* \in X$ . Supposons  $f$ ,  $g$  et  $h$  deux fois différentiables. Notons :

$$V_{x^*}(X) = \{d \in \mathbb{R}^n \mid \begin{aligned} \langle \nabla h_i(x), d \rangle &= 0, \quad \forall i = 1, \dots, p, \\ \langle \nabla g_j(x), d \rangle &= 0, \quad \forall j \text{ actif tel que } : \mu_j > 0, \\ \langle \nabla g_j(x), d \rangle &\leq 0, \quad \forall j \text{ actif tel que } : \mu_j = 0 \end{aligned}$$

le cône linéarisant des contraintes au point  $x^*$ .

#### CN d'optimalité du second ordre

Si  $x^*$  est un point de minimum local du problème (P) et si les vecteurs :

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

sont linéairement indépendants, alors :

- i.  $x^*$  est un point de KKT du problème (P) admettant des multiplicateurs de Lagrange  $\lambda^*$  et  $\mu^*$  respectivement associés aux contraintes d'égalité et d'inégalité.
- ii. Pour toute direction  $d \in V_{x^*}(X)$ , on a :

$$\langle H_x[L](x^*; \lambda^*, \mu^*)d, d \rangle \geq 0.$$

#### CS d'optimalité du second ordre

Supposons que  $x^*$  est un point de KKT du problème (P) et notons  $(\lambda^*, \mu^*)$  les multiplicateurs associés. Si pour toute direction  $d \in V_{x^*}(X)$ ,  $d \neq 0$ , on a :

$$\langle H_x[L](x^*; \lambda^*, \mu^*)d, d \rangle > 0,$$

alors  $x^*$  est un point de minimum local strict de  $f$  sur  $X$ .

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

**Exercice 3.3.3** Ecrire les conditions nécessaires et les conditions suffisantes d'optimalité du premier et du second ordre pour le problème suivant :

$$\begin{aligned} (P') \quad & \max f(x) \\ \text{s.c.} \quad & h_i(x) = 0, \quad i = 1, \dots, p, \quad g_j(x) \leq 0, \quad j = 1, \dots, q. \end{aligned}$$

en gardant les notations du Théorème 3.3.

# Chapitre 4

## Dualité Lagrangienne

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

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

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  $(\lambda^*, \mu^*)$ . 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  $(\lambda, \mu)$  ne servant souvent qu'à vérifier l'optimalité de  $x^*$  : elles engendrent une suite de solutions réalisables en assurant la décroissance de la fonction objectif.

- Avantages : quand le processus itératif s'arrête, on obtient une solution approchée satisfaisant les contraintes.
- Inconvénients : mise au point délicate, convergence globale difficile à obtenir.

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*.

- Avantages : méthodes plus robustes que les méthodes primales, convergence globale plus facile à obtenir.
- Inconvénients : une solution primale réalisable n'est obtenue qu'en fin de convergence.

## 4.1 Problème primal - Problème dual

On définit le *Lagrangien* (ou fonction lagrangienne) associé au problème  $(P)$  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  $\lambda$  et  $\mu$  les vecteurs de coordonnées respectives  $(\lambda_1, \dots, \lambda_p)$  et  $(\mu_1, \dots, \mu_q)$ .

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.



Une question importante soulevée par la définition du problème dual est de 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 4.1** *La fonction duale  $f^*$  est toujours concave et le domaine admissible dual  $X^*$  convexe.*

**Exercice 4.1.1** *Le démontrer !*

## 4.2 Théorèmes de dualité et points selle du Lagrangien

Remarquons qu'on a toujours :

**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 qu'on n'est pas capable *a priori* de quantifier l'écart entre les solutions primale et duale. 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 4.1 (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 4.1 (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.

#### Théorème 4.2

*Le point  $(\bar{x}, \bar{\lambda}, \bar{\mu})$ ,  $\mu \geq 0$ , est un point selle du Lagrangien si et seulement si :*

- 1.  $\bar{x}$  est admissible pour  $(P)$  (i.e.  $\bar{x} \in X$ ).
- 2.  $\bar{\mu}^\top g(\bar{x}) = 0$ .
- 3.  $f(\bar{x}) = L(\bar{x}; \bar{\lambda}, \bar{\mu}) = \inf_{x \in \mathbb{R}^n} L(x; \bar{\lambda}, \bar{\mu})$ .

**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.

#### Dualité dans le cas convexe

Supposons que la fonction objectif  $f$  et les contraintes  $g_j$ ,  $j = 1, \dots, q$ , sont convexes et que les fonctions  $h_i$ ,  $i = 1, \dots, p$  affines. Le problème :

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

est donc convexe. On note

**Définition 4.2 (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 4.3 (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  $\bar{\lambda} \in \mathbb{R}^p$  et  $\bar{\mu} \in (\mathbb{R}_+)^q$  tel que  $(\bar{x}, \bar{\lambda}, \bar{\mu})$  soit un point-selle du Lagrangien.

Dans le chapitre précédent, nous avons énoncé les conditions d'optimalité de KKT. Le théorème suivant explicite le lien existant entre ces conditions et le critère de point selle du Lagrangien dans le cas convexe :

**Théorème 4.4** Considérons le problème :

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

où les fonctions  $g_j$ ,  $j = 1, \dots, q$ , et  $f$  sont convexes et les fonctions  $h_i$ ,  $i = 1, \dots, p$ , sont affines. On note  $X$  le domaine admissible.

1. Supposons que  $\bar{x} \in X$  vérifie les conditions de KKT i.e. qu'il existe  $\bar{\lambda} \in \mathbb{R}^p$  et  $\bar{\mu} \in (\mathbb{R}_+)^q$  tels que :

$$\begin{aligned} \nabla f(\bar{x}) + \sum_{i=1}^p \bar{\lambda}_i \nabla h_i(\bar{x}) + \sum_{j=1}^q \bar{\mu}_j \nabla g_j(\bar{x}) &= 0, \\ \bar{\mu}^\top g(\bar{x}) &= 0. \end{aligned}$$

Alors  $(\bar{x}, \bar{\lambda}, \bar{\mu})$  est un point selle du Lagrangien associé.

2. Inversement, supposons que  $(\bar{x}, \bar{\lambda}, \bar{\mu})$  est un point selle du Lagrangien avec  $\bar{\mu} \geq 0$ . Alors le point  $\bar{x}$  associé aux multiplicateurs  $(\bar{\lambda}, \bar{\mu})$  vérifie les conditions de KKT.

**Démonstration.** En exercice. □

## 4.3 Cas particulier de la programmation linéaire

Un problème de programmation linéaire est un problème d'optimisation dans lequel l'objectif et les contraintes sont linéaires ou affines. Une formulation tout à fait générale

de ce problème est la suivante : il s'agit de trouver la solution  $x \in \mathbb{R}^n$  du problème :

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & c_1 x_1 + \cdots + c_n x_n \\ \text{sous} & a_{11} x_1 + \cdots + a_{1n} x_n \leq b_1 \\ & \vdots \\ & a_{m1} x_1 + \cdots + a_{mn} x_n \leq b_m \end{array} \quad \text{OU} \quad \begin{array}{ll} \max_{x \in \mathbb{R}^n} & c_1 x_1 + \cdots + c_n x_n \\ \text{sous} & a_{11} x_1 + \cdots + a_{1n} x_n \leq b_1 \\ & \vdots \\ & a_{m1} x_1 + \cdots + a_{mn} x_n \leq b_m \end{array}$$

ou encore sous forme matricielle :

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & c^\top x \\ \text{sous} & Ax \leq b \end{array} \quad \text{OU} \quad \begin{array}{ll} \max_{x \in \mathbb{R}^n} & c^\top x \\ \text{sous} & Ax \leq b. \end{array}$$

en posant  $A = (a_{ij})_{i,j} \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$  et  $c \in \mathbb{R}^n$ .

### 4.3.1 Généralités

Sachant que les problèmes  $\min_{Ax \leq b} c^\top x$  et  $\max_{Ax \leq b} -c^\top x$  sont équivalents, on s'intéressera ici au problème :

$$(P_L) \quad \begin{array}{ll} \min_{x \in \mathbb{R}^n} & c^\top x \\ \text{sous} & Ax \leq b \end{array}$$

On note :

$$X = \{x \in \mathbb{R}^n / AX \leq b\} = \{x \in \mathbb{R}^n / a_i x \leq b_i, i = 1, \dots, m\}$$

l'ensemble admissible du problème  $(P_L)$  avec :  $A^\top = [a_1 \mid \dots \mid a_m]$  où les  $a_i^\top \in \mathbb{R}^n$  sont les vecteurs associés aux lignes de la matrice  $A$  (de taille  $m \times n$ ).

#### Définition 4.3 (Point admissible/Problème réalisable)

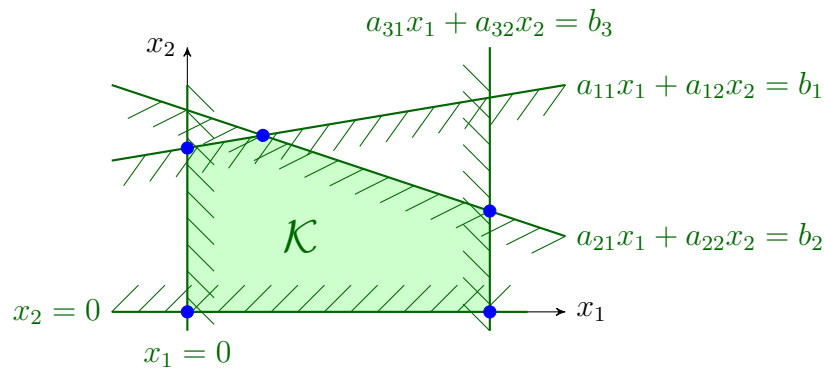
1. Un point  $\bar{x} \in \mathbb{R}^n$  est dit (strictement) admissible pour le problème  $(P_L)$  si :  $A\bar{x} \leq b$  (si  $A\bar{x} < b$ ).
2. Le problème  $(P_L)$  est dit (strictement) réalisable si son ensemble admissible  $\{x \in \mathbb{R}^n / Ax \leq b\}$  est non vide (d'intérieur non vide).
3. Le problème  $(P_L)$  est dit borné si la valeur minimale du critère dans  $(P_L)$  est bornée, i.e. si :

$$\inf\{c^\top x / Ax \leq b\} > -\infty.$$

#### Structure de l'ensemble admissible

**Proposition 4.2** *L'ensemble admissible  $X$  est polyèdre convexe fermé de  $\mathbb{R}^n$  i.e. l'intersection d'un nombre fini de demi-espaces fermés.*

**Définition 4.4 (Point extrême/sommet d'un polyèdre)** *Un point extrême (auss appelé sommet) du polyèdre  $X$ , est un point de  $X$  situé à l'intersection de  $n$  hyperplans parmi ceux définissant le polyèdre.*



**Proposition 4.3** *Tout point de  $X$  peut s'écrire comme combinaison convexe des points extrêmes du polyèdre. Autrement dit, notons  $x_1, \dots, x_p$  les sommets du polyèdre  $X$  : pour tout  $x \in X$ , il existe  $(\lambda_1, \dots, \lambda_p) \in \mathbb{R}_+^p$  tels que :*

$$x = \sum_{i=1}^p \lambda_i x_i \quad \text{et} \quad \sum_{i=1}^p \lambda_i = 1.$$

### Notion de solution

Etudions d'un peu plus près les propriétés d'un problème de programmation linéaire :

$$\begin{array}{ll} \text{Minimiser ou maximiser} & c^\top x \\ \text{sous la contrainte :} & Ax \leq b \end{array}$$

Remarquons tout d'abord que la fonction objectif  $f(x) = c^\top x$  est linéaire sur  $\mathbb{R}^n$ , donc à la fois convexe et concave sur  $\mathbb{R}^n$  (mais pas strictement !). L'ensemble  $X$  des contraintes est convexe. Toute solution du problème :

$$(P_{L,min}) \min_{x \in \mathbb{R}^n} c^\top x \quad \text{sous} \quad Ax \leq b,$$

est donc un point de minimum global de  $f$  sur  $X$ .

### Existence de solution

La fonction cout étant linéaire, donc continue sur  $\mathbb{R}^n$ , on considère deux cas :

**1er cas :  $X$  est borné.** D'après le théorème de Weierstrass, la fonction  $f$  est bornée et atteint ses bornes. Autrement dit  $f$  admet (au moins) un point de minimum global et (au moins) un point de maximum global sur  $X$ .

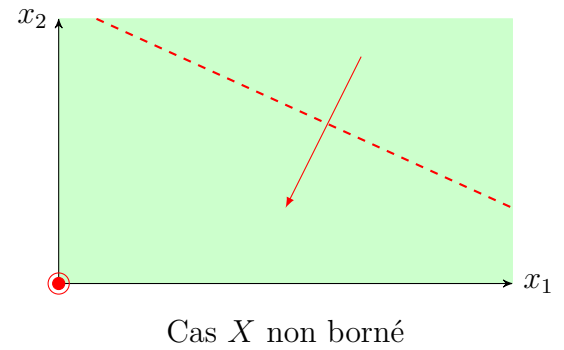
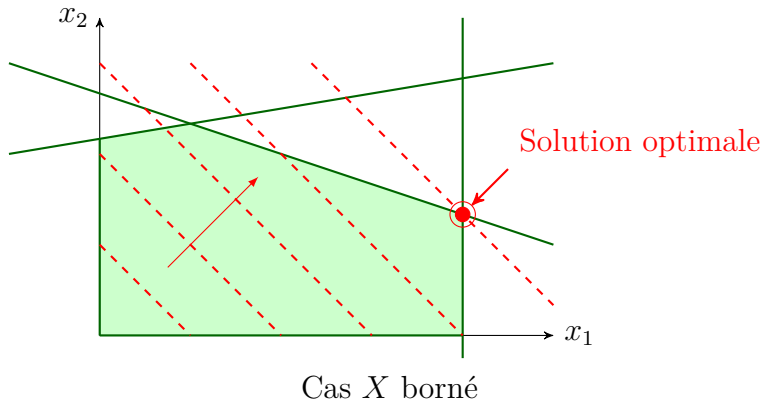
Il peut y avoir une seule solution ou une infinité de solution !

**2ème cas :  $X$  non borné.** Pas de résultat général, on peut avoir :

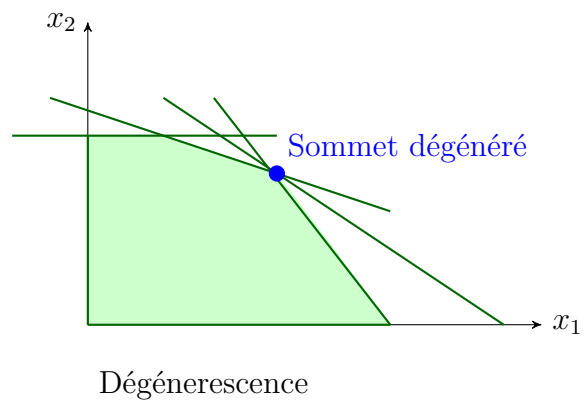
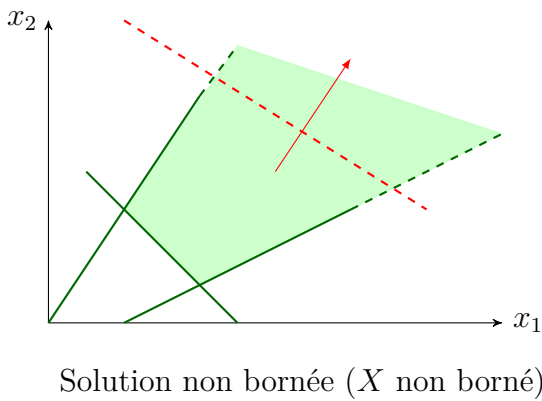
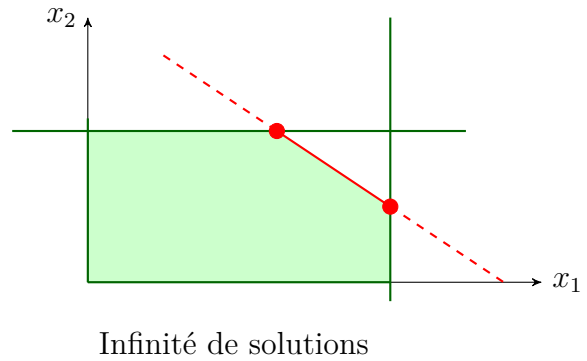
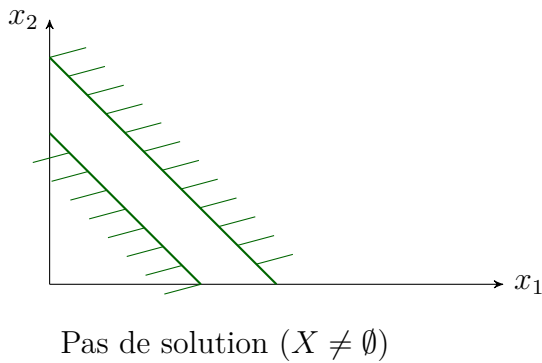
- aucune solution optimale (ensemble admissible vide ou non borné)
- une seule solution optimale
- une infinité de solutions optimales

### Résolution graphique

On trace les droites isocritères  $z = c^\top x = C^{ste}$  et on balaye l'ensemble des solutions admissibles.



### Configurations particulières



**Proposition 4.4 (Existence de solution-sommet)** *Si le problème  $(P_L)$  admet une solution, alors il admet au moins une solution en un sommet de l'ensemble admissible  $X$ .*

### Conditions d'optimalité

Les conditions nécessaires d'optimalité du premier ordre du problème  $(P_L)$  s'écrivent : il existe  $\lambda \in \mathbb{R}^m$  tel que

$$\begin{cases} c + A^\top \lambda = 0, \\ \lambda_i = 0 \text{ ou } a_i^\top x = b_i, & i = 1, \dots, m, \\ \lambda_i \geq 0, & i = 1, \dots, m. \end{cases}$$

### 4.3.2 Dualité en programmation linéaire

Considérons le programme linéaire suivant :

$$\min_{y \in \mathbb{R}^m} b^\top y \quad \text{s.c.} \quad A^\top y \geq c, \quad y \geq 0.$$

Ce problème est appelé problème dual du problème de départ appelé problème primal :

$$\begin{array}{l|l} \text{Primal} & \begin{array}{l} \max_x \quad z = c^\top x \\ \text{sous} \quad Ax \leq b, \quad x \geq 0 \end{array} \\ \text{Dual} & \begin{array}{l} \min_y \quad t = b^\top y \\ \text{sous} \quad A^\top y \geq c, \quad y \geq 0 \end{array} \end{array}$$

Notons que le problème dual du dual est le problème primal.

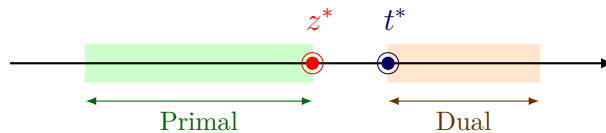
**Exemple 4.3.1** Construire le dual du problème suivant :

$$\begin{array}{rclcl} \max z = & 4x_1 & + & 5x_2 & \\ & 2x_1 & + & x_2 & \leq 800 \\ & x_1 & + & 2x_2 & \leq 700 \\ & & & x_2 & \leq 300 \\ & x_1 & , & x_2 & \geq 0. \end{array}$$

### Théorèmes de dualité

Appliquons le théorème de dualité faible vu précédemment : pour toutes solutions primale  $x$  et duale  $y$  admissibles, on a

$$\begin{array}{l} Ax \leq b, \quad x \geq 0 \\ A^\top y \geq c, \quad y \geq 0 \end{array} \Rightarrow \quad z = c^\top x \leq b^\top y = t \quad \text{et} \quad z^* \leq t^*$$



### Corollaire 4.1

- S'il existe  $x$  et  $y$  resp. primal et dual admissibles t.q.  $c^\top x = b^\top y$  alors ces solutions sont optimales

- Si le coût optimal du primal  $z^* = +\infty$  alors le dual est irréalisable
- Si le coût optimal du dual  $t^* = -\infty$  alors le primal est irréalisable

### Exemple 4.3.2

$$+\infty = z^* = \max_{\mu \succeq 0} \begin{bmatrix} 1 & -1 \end{bmatrix} \mu \quad t^* = \min_x x$$

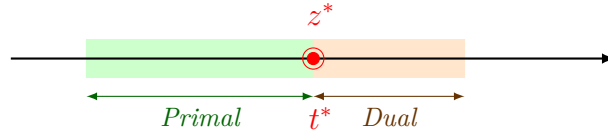
sous  $\begin{bmatrix} 0 & 1 \end{bmatrix} \mu = 1.$       sous  $\begin{bmatrix} 0 & 1 \end{bmatrix} x \preceq \begin{bmatrix} -1 & 1 \end{bmatrix}.$

### Théorème 4.5 (Théorème de la dualité (Gale, Kuhn, Tucker 1951))

Exactement une des trois alternatives est vraie :

- Le problème primal et le problème dual sont réalisables et ont une solution optimale identique.

$$z^* = c^T x^* = b^T y^* = t^*$$



- Exactement un des problèmes est irréalisable et l'autre est non borné.
- Primal et dual sont irréalisables.

Exemple : Primal et dual irréalisables

$$z^* = \max_x \begin{bmatrix} 1 & 3 \end{bmatrix} x \quad t^* = \min_y y_1 + 2y_2$$

sous  $\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$       sous  $\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} y = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$

## 4.4 Application à la dualité conique

Le cadre de travail fourni par la dualité lagrangienne peut être étendu à des problèmes d'optimisation conique construits autour de la notion d'inégalités généralisées présentée dans le Chapitre 1. Cette partie du cours est largement inspirée des références [6, 2]. Considérons un problème d'optimisation conique très général :

$$\inf_{x \in \mathbb{R}^n} f(x) \text{ s.c. } \begin{cases} g_j(x) \preceq_{K_j} 0, & j = 1, \dots, q, \\ h_i(x) = 0, & i = 1, \dots, p, \end{cases} \quad (4.1)$$

où :

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g_j : \mathbb{R}^n \rightarrow \mathbb{R}^{k_j}$ ,  $j = 1, \dots, q$ , et  $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, p$ , sont supposées différentiables.
- $K_j \subset \mathbb{R}^{k_j}$ ,  $j = 1, \dots, q$ , sont des cônes supposés propres.



Notons  $X$  l'ensemble admissible du problème (4.1). On définit alors le Lagrangien associé au Problème (4.1) de la façon suivante :

$$L(x; \lambda, \mu) = f(x) + \sum_{j=1}^q \langle \lambda_j, g_j(x) \rangle + \sum_{i=1}^p \mu_i h_i(x), \quad (4.2)$$

avec  $\lambda = (\lambda_1, \dots, \lambda_q) \in K_1^* \times \dots \times K_q^*$  et  $\mu = (\mu_1, \dots, \mu_p) \in \mathbb{R}^p$ , où les  $K_j^*$  désignent les cônes duaux des  $K_j$ ,  $j = 1, \dots, q$ .

#### 4.4.1 Conditions de KKT en optimisation conique

Intéressons nous maintenant aux conditions d'optimalité des problèmes d'optimisation conique de la forme générale (4.1). Les conditions nécessaires d'optimalité de KKT se généralisent de la façon suivante :

##### Conditions de KKT en optimisation conique

$$\begin{aligned} \nabla f(x) + \sum_{j=1}^q J_{g_j}(x)^\top \lambda_j + \sum_{i=1}^p \mu_i \nabla h_i(x) &= 0 \\ \langle \lambda_j, g_j(x) \rangle &= 0, \quad j = 1, \dots, q, \\ g_j(x) &\succeq_{K_j} 0, \quad j = 1, \dots, q, \\ \lambda_j &\succeq_{K_j^*} 0, \quad j = 1, \dots, q, \\ h_i(x) &= 0, \quad i = 1, \dots, p. \end{aligned}$$

Si le problème est convexe, alors ces conditions deviennent nécessaires et suffisantes.

On remarquera que la condition de positivité des multiplicateurs de Lagrange associés à des contraintes d'inégalités scalaires a été remplacée par la condition :

$$\lambda_j \succeq_{K_j^*} 0, \quad j = 1, \dots, q, \quad (4.3)$$

c'est-à-dire :

$$\lambda_j \in K_j^*, \quad j = 1, \dots, q.$$

Il est important de remarquer que contrairement au cas scalaire, il est possible d'avoir  $\langle \lambda_j^*, g_j(x^*) \rangle = 0$  avec  $\lambda_j^* \neq 0$  et  $g_j(x^*) \neq 0$ . Les conditions de complémentarité  $\langle \lambda_j^*, g_j(x^*) \rangle = 0$  doivent donc être comprises de la façon suivante :

$$\langle \lambda_j^*, g_j(x^*) \rangle = 0$$

signifie :

$$\begin{aligned} \lambda_j^* \succ_{K_j^*} 0 &\Rightarrow g_j(x^*) = 0, \\ g_j(x^*) \succ_{K_j} 0 &\Rightarrow \lambda_j^* = 0. \end{aligned}$$

### 4.4.2 Dualité lagrangienne en optimisation conique

Comme précédemment dans le cas scalaire, on définit la fonction duale :

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

qui est toujours concave, puis le problème dual :

$$\sup_{\lambda, \mu} f^*(\lambda, \mu) \quad \text{s.c.} \quad \lambda_j \succeq_{K_j^*} 0, \quad j = 1, \dots, q. \quad (4.4)$$

Remarquons que le cas de contraintes d'inégalité scalaires correspond au cas particulier où les  $K_j$  désignent l'orthant positif en dimension 1, soit tout simplement  $\mathbb{R}_+$ . Sachant que  $(\mathbb{R}_+)^* = \mathbb{R}_+$ , la condition  $\lambda_j \succeq_{K_j^*} 0$  s'écrit alors :

$$\lambda_j \geq 0, \quad i = 1, \dots, q.$$

#### Proposition 4.5 (Principe de dualité faible)

$$d^* = \sup_{\lambda_i \succeq_{K_j^*} 0, \mu} \left( \inf_{x \in \mathbb{R}^n} L(x; \lambda, \mu) \right) \leq p^* = \inf_{x \in \mathbb{R}^n} \left( \sup_{\lambda_j \succeq_{K_j^*} 0, \mu} L(x; \lambda, \mu) \right).$$

#### Proposition 4.6 (Principe de dualité forte)

- Si le Problème primal (4.1) est strictement réalisable et si  $p^*$  est fini, alors le saut de dualité est nul (i.e.  $d^* = p^*$ ) et il existe au moins une solution duale admissible et  $d^*$  est atteint.
- Si le Problème dual (4.1) est strictement réalisable et si  $d^*$  est fini, alors le saut de dualité est nul (i.e.  $d^* = p^*$ ) et il existe au moins une solution primale admissible et  $p^*$  est atteint.
- Si les Problèmes primal (4.1) et dual (4.4) sont tous les deux strictement réalisables alors le saut de dualité est nul et il existe des solutions admissibles pour chaque problème.

### Cas convexe : condition de Slater et dualité forte

Supposons maintenant un problème d'optimisation conique convexe i.e.  $f$  convexe et les  $g_j, j = 1, \dots, q$ , convexes. On s'intéresse au problème suivant :

$$\inf_{x \in \mathbb{R}^n} f(x) \quad \text{s.c.} : \quad \left| \begin{array}{ll} g_j(x) & \preceq_{K_j} 0, \quad j = 1, \dots, q, \\ Ax & = b. \end{array} \right.$$

Comme dans le cas scalaire, on a la dualité forte dans le cas convexe pourvu qu'une certaine condition de qualification des contraintes soit satisfaite. Par exemple, si la condition de Slater généralisée suivante :

Il existe un  $x \in \text{dom}(f)$  tel que  $Ax = b$  et  $g_j(x) \prec_{K_j} 0$ ,

est satisfaite pour le problème (4.4.2), alors le saut de dualité est nul et il existe des solutions admissibles pour chacun des problèmes primal et dual.

### 4.4.3 Cas particulier de la dualité SDP

La programmation SDP est un cas particulier de la programmation conique dans l'espace des matrices symétriques dans lequel :

- La fonction objectif et la fonction définissant les contraintes d'inégalité sont linéaires,
- Le cône définissant la contrainte d'inégalité généralisée est le cône SDP.

Considérons un problème SDP sous une forme très générique :

#### Définition 4.5 (Problème SDP primal)

$$\sup_{X \in \mathbb{S}_n} \langle C, X \rangle_F \text{ s.c. } \left| \begin{array}{l} \langle A_j, X \rangle_F = b_j, \quad j = 1, \dots, q, \\ X \succeq 0, \end{array} \right. \quad (4.5)$$

où les matrices  $A_j$ ,  $j = 1, \dots, q$ , et  $C$  sont supposées symétriques de taille  $n \times n$  et  $b \in \mathbb{R}^q$ .

On note :

$$F = \{X \in \mathbb{S}_n \mid \langle A_j, X \rangle_F = b_j, \quad j = 1, \dots, q\} \cap \mathbb{S}_n^+$$

l'ensemble admissible (on dit aussi réalisable) associé au Problème (4.5). Une solution  $X \in F$  est dite strictement réalisable si :  $X \succ 0$ . Le Problème (4.5) est dit strictement réalisable s'il existe au moins une solution strictement réalisable.

**Exercice 4.4.1** Considérons un problème SDP de la forme (4.5) avec les données suivantes :  $n = m = 2$  et

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 2a \\ 0 \end{pmatrix}.$$

1. Écrire le problème SDP associé sous la forme la plus simple possible.
2. Déterminer, en fonction de  $a \in \mathbb{R}$ , la réalisabilité des problèmes suivants et, le cas échéant, les résoudre :

$$\begin{array}{l} \sup_{X \in \mathbb{S}_2^+} \langle C, X \rangle_F \text{ s.c. } \left| \begin{array}{l} \langle A_j, X \rangle_F = b_j, \quad j = 1, \dots, 2, \\ X \succeq 0, \end{array} \right. \\ \\ \inf_{X \in \mathbb{S}_2^+} \langle C, X \rangle_F \text{ s.c. } \left| \begin{array}{l} \langle A_j, X \rangle_F = b_j, \quad j = 1, \dots, 2, \\ X \succeq 0. \end{array} \right. \end{array}$$

Même si les résultats de dualité lagrangienne appliqués à ce type de problèmes sont moins forts que ceux obtenus en programmation linéaire, il est toutefois possible d'écrire explicitement le dual lagrangien de (4.5). En effet, la fonction duale s'écrit analytiquement de la façon suivante :

$$f^*(\lambda) = \begin{cases} -\langle b, \lambda \rangle & \text{si } C + \sum_{j=1}^m \lambda_j A_j \succeq 0, \\ -\infty & \text{sinon.} \end{cases}, \quad \lambda \in \mathbb{R}^m.$$

On en déduit alors le problème SDP dual associé au Problème (4.5) :

$$\min_{\lambda \in \mathbb{R}^m} \langle b, \lambda \rangle \text{ s.c. } C + \sum_{j=1}^m \lambda_j A_j \succeq 0, \quad (4.6)$$

qui est appelé problème LMI (Inégalité Matricielle Linéaire ou *Linear Matrix Inequality* en anglais).

**Remarque 4.1** *Les Problèmes (4.6) et (4.5) sont classiquement appelés **problème LMI primal** et **problème LMI dual** en Automatique. De façon tout à fait similaire, tout problème LMI générique :*

$$\inf_{x \in \mathbb{R}^n} \langle c, x \rangle \text{ s.c. } F(x) = F_0 + \sum_{i=1}^n x_i F_i \succeq 0, \quad (4.7)$$

où les matrices  $F_i$ ,  $i = 0, \dots, n$ , sont supposées symétriques de taille  $m$  et  $c \in \mathbb{R}^n$ , admet un problème dual de la forme :

$$\sup_{Z \in \mathbb{S}_m^+} -\langle F_0, Z \rangle_F \text{ s.c. } \left| \begin{array}{l} c_i - \langle F_i, Z \rangle_F = 0, \quad i = 1, \dots, n, \\ Z \succeq 0. \end{array} \right.$$

ou de façon équivalente :

$$\inf_{Z \in \mathbb{S}_m^+} \langle F_0, Z \rangle_F \text{ s.c. } \left| \begin{array}{l} c_i - \langle F_i, Z \rangle_F = 0, \quad i = 1, \dots, n, \\ Z \succeq 0. \end{array} \right.$$

Un des intérêts majeurs<sup>1</sup> de ce type de problèmes d'optimisation est qu'il entre dans la classe des problèmes d'optimisation pour lesquels des algorithmes en temps polynomial peuvent être appliqués. Au moins théoriquement, la taille (nombre de variables, nombres de contraintes) n'est donc pas un facteur discriminant. La généralisation des méthodes de points intérieurs aux problèmes d'optimisation non linéaires convexes dont le Problème (4.7) est une sous-classe, a ouvert la voie pour le développement d'algorithmes et de solveurs (SeDuMi, LMILab, SDPT3, SDPA, CSDP, DSDP...) associés permettant en pratique d'aborder les problèmes de taille moyenne (jusqu'à 5000 variables et autant de contraintes). Un autre avantage réside dans la capacité de (4.7) à modéliser des problèmes pratiques très

---

1. cf [2] d'où est issue cette explication pour plus de détails.

variés aussi bien en théorie de la commande qu'en statistiques, finances ou conception de structures. Ceci est lié en partie à la flexibilité de (4.7) qui permet d'ajouter diverses contraintes sur les variables de décision sans altérer le degré de difficulté si l'on préserve le caractère LMI.

Une dernière raison est que la programmation SDP semble être un outil intrinsèquement adapté à la prise en compte de l'incertitude affectant les données d'un problème. Pour s'en convaincre, il suffit de constater le succès rencontré en théorie de la commande robuste mais aussi le développement de nouveaux champs d'application tels que la programmation quadratique robuste.



## Troisième partie

### Algorithmes classiques en optimisation différentiable





# Chapitre 5

## 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.

### 5.1 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 :

$$\forall t \in ]0, \eta], f(x + td) < f(x). \quad (5.1)$$

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 5.1** *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 5.2** *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 :*

$$f'(x; d) = \nabla f(x)^\top d < 0. \quad (5.2)$$

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

$$\forall t \in ]0, \bar{\eta}], f(x + td) < f(x) + t\beta \nabla f(x)^\top d < f(x). \quad (5.3)$$

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 5.3.
- la stratégie de Newton :  $d = -H[f](x_k)^{-1}\nabla f(x_k)$ , conduisant aux *algorithmes Newtoniens* décrits au paragraphe 5.4.

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

$$\|\nabla f(x_k)\| < \varepsilon, \quad (5.4)$$

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

## 5.2 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,  $\varphi$  est dérivable sur  $\mathbb{R}$ , de dérivée :  $\varphi'(s) = \nabla f(x + sd)^\top d$  et :

$$\varphi'(0) = \nabla f(x)^\top d < 0. \quad (5.5)$$

### 5.2.1 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.

---

$$s_k = s_{k-1}$$

Tant que  $f(x_k + s_k d_k) \geq f(x_k)$  :

$$s_k = s_k/2$$


---

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:*  $s_k$ .

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

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


---

### 5.2.2 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 :  $d_k = -f'(x_k)/|f'(x_k)| = -sgn(x_k)$ .
  - Pas de descente :  $s_k = 2 + \frac{3}{2^{k+1}}$ .

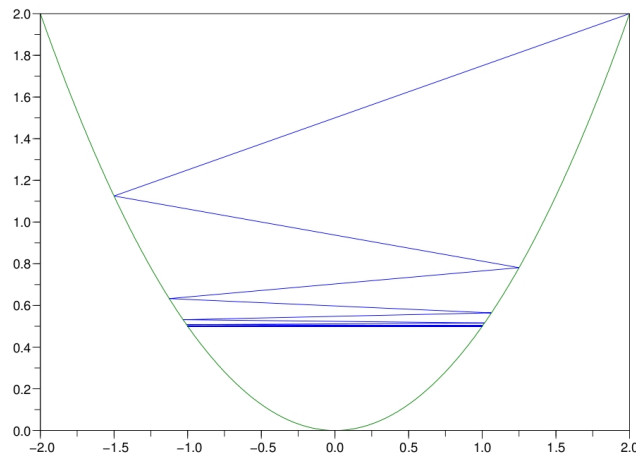


FIGURE 5.1 – 1000 premiers itérés  $(f(x_k))_k$  générés par l'algorithme de gradient ( $\mathcal{A}_1$ ).

Les résultats sont présentés sur la figure 5.1. 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  $\varphi$ , on préfère imposer des conditions moins restrictives (et moins coûteuses à mettre en oeuvre) : une première condition est dû à Armijo (1966) :

$$f(x + sd) \leq f(x) + \varepsilon s \left( \nabla f(x)^\top d \right), \quad 0 < \varepsilon < 1 \quad (5.6)$$

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

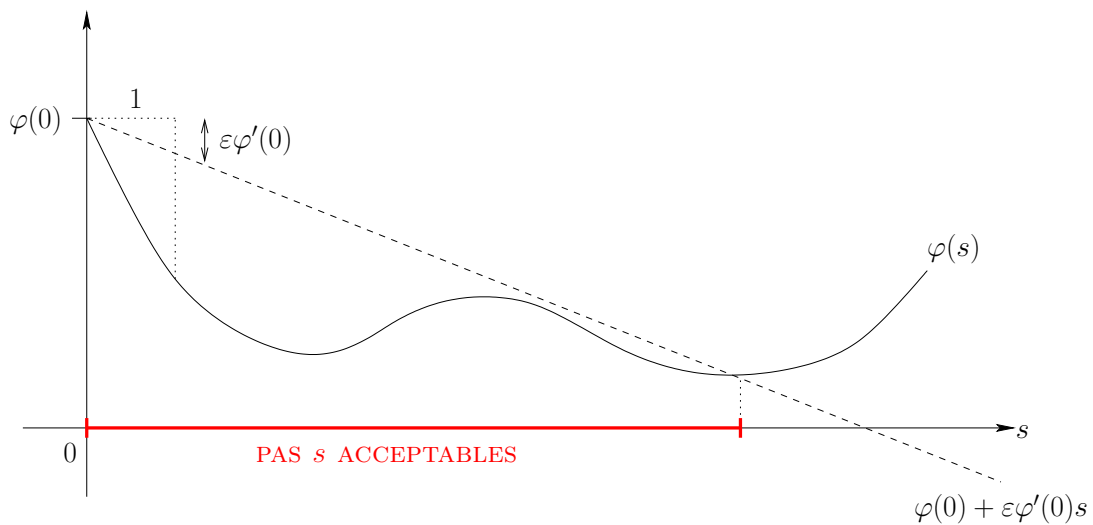


FIGURE 5.2 – Règle d'Armijo

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  $\varepsilon$  est choisie très petite de manière à satisfaire (5.6) le plus facilement possible ; typiquement :  $\varepsilon = 10^{-4}$ .

**Remarque 5.2** D'après la définition (5.1) d'une direction de descente, il existe tout un intervalle de pas satisfaisant la condition d'Armijo (5.6).

### 5.2.3 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 5.3. 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  $f$  !

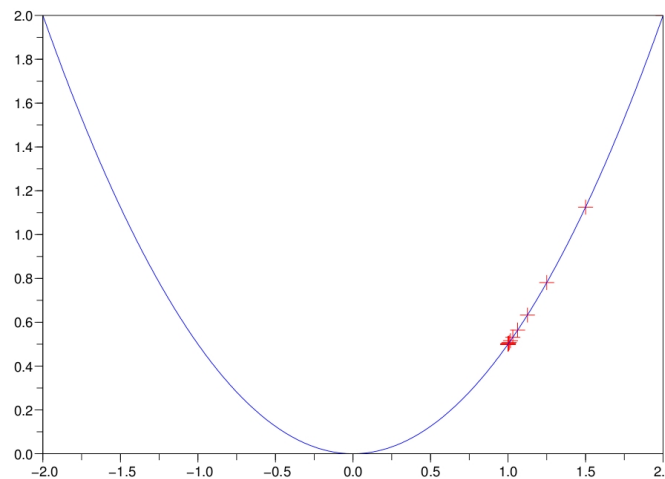


FIGURE 5.3 – 1000 premiers itérés  $(f(x_k))_k$  générés par l'algorithme de gradient ( $\mathcal{A}_2$ ).

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  $x_k$  :

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

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 :

$$f(x + sd) \leq f(x) + \varepsilon_1 s (\nabla f(x)^\top d) \quad (5.7)$$

$$\nabla f(x + sd)^\top d \geq \varepsilon_2 (\nabla f(x)^\top d) \quad (5.8)$$

où les constantes  $\varepsilon_1$  et  $\varepsilon_2$  sont choisies telles que :

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

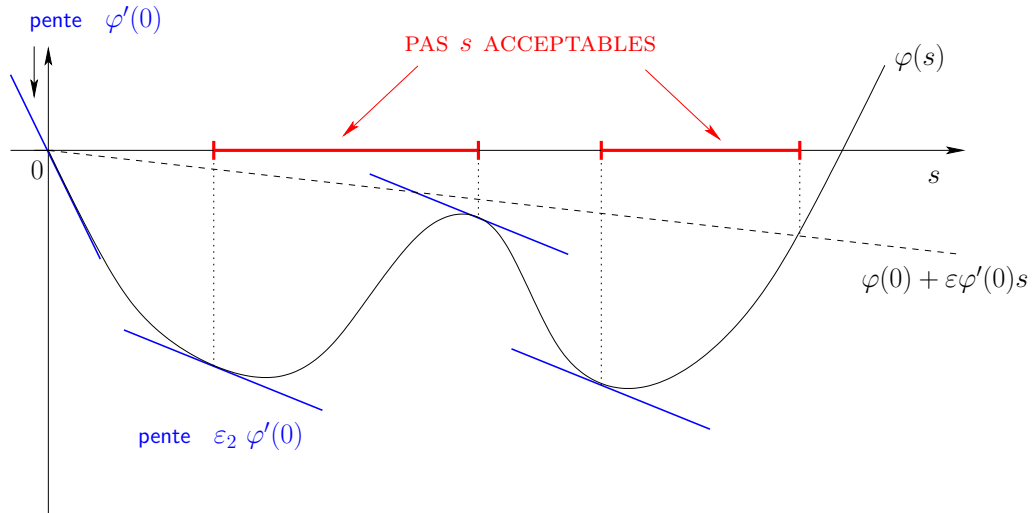


FIGURE 5.4 – Conditions de Wolfe

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

**Proposition 5.3 (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 (5.7) et (5.8).

**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 (5.7), 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 (5.7) mais pas (5.8), 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 5.1** 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.

### 5.2.4 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 5.4** *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 [3, Proposition 1.2.1] ou [9, 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 :

$$f(x_k) - f(x_{k+1}) \geq c \|\nabla f(x_k)\|^2, \quad (5.9)$$

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  $\theta_k$  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 5.2 (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.

## 5.3 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 :

$$x_0 \in \mathbb{R}^n \text{ quelconque, } \quad x_{k+1} = x_k - s_k \nabla f(x_k) \quad (5.10)$$

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 5.3.1 (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  $\mathbb{R}^n$ .

### 5.3.1 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 (5.10) soit une méthode de descente, i.e. :

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

**Proposition 5.4** *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. □

### 5.3.2 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 (5.10) i.e. montrer que :

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

**Théorème 5.3** *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. □

### 5.3.3 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  $\nabla f$  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 5.4, 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 5.1 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 5.1 – 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 5.5 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 5.6 : elle permet de forcer la convergence et d'accélérer la vitesse de convergence des itérés.

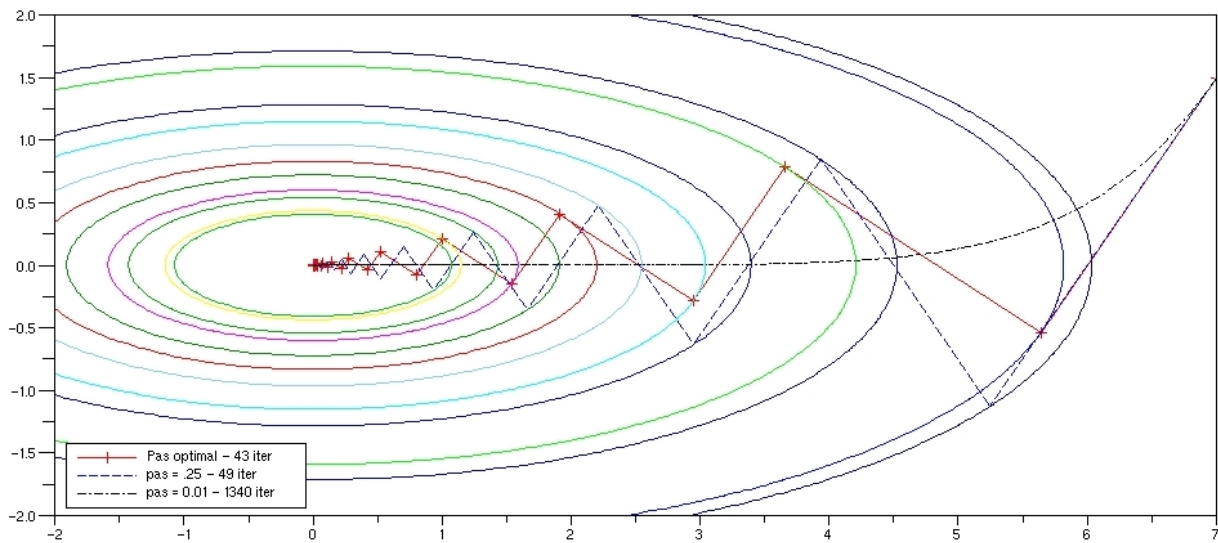


FIGURE 5.5 – Itérations des algos de gradient pas fixe et optimal, générées à partir du point  $(7, 1.5)$ .

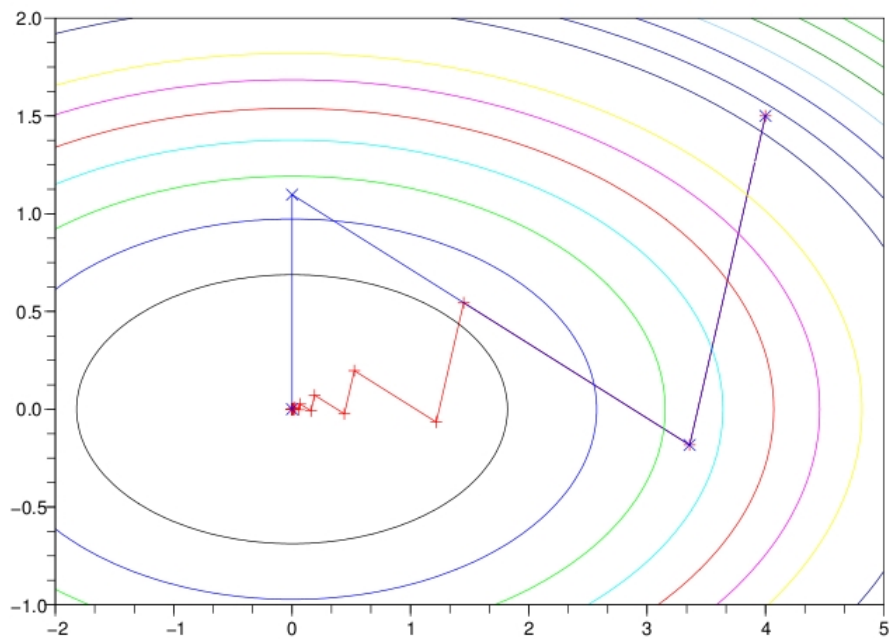


FIGURE 5.6 – 23 it. de l'algorithme de plus profonde descente (en rouge) vs 3 it. l'algorithme de gradient avec recherche linéaire de Wolfe (en bleu) à partir de  $(x_0, y_0) = (4, 1.5)$ .

## 5.4 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 :

$$x_{k+1} = x_k - H[f](x_k)^{-1} \nabla f(x_k), \quad (5.11)$$

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 ?

### 5.4.1 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  $\alpha I$ .

**Remarque 5.5** *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 :

$$\begin{aligned} (P_1) \quad \min_{(x,y) \in \mathbb{R}^2} f(x,y) &= 100(y - x^2)^2 + (1 - x)^2 && \text{fonction de Rosenbrock.} \\ (P_2) \quad \min_{(x,y) \in \mathbb{R}^2} g(x,y) &= \frac{1}{2}x^2 + x \cos y. \end{aligned}$$

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 :

$$\begin{aligned} ((-1)^{k+1}, k\pi), \quad k \in \mathbb{Z} & \quad \text{points de minimum local de } g \\ (0, \frac{\pi}{2} + k\pi), \quad k \in \mathbb{Z} & \quad \text{points selle de } g \end{aligned}$$

Les résultats sont présentés sur les figures 5.7 et 5.8. 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 5.7). La figure 5.8 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.



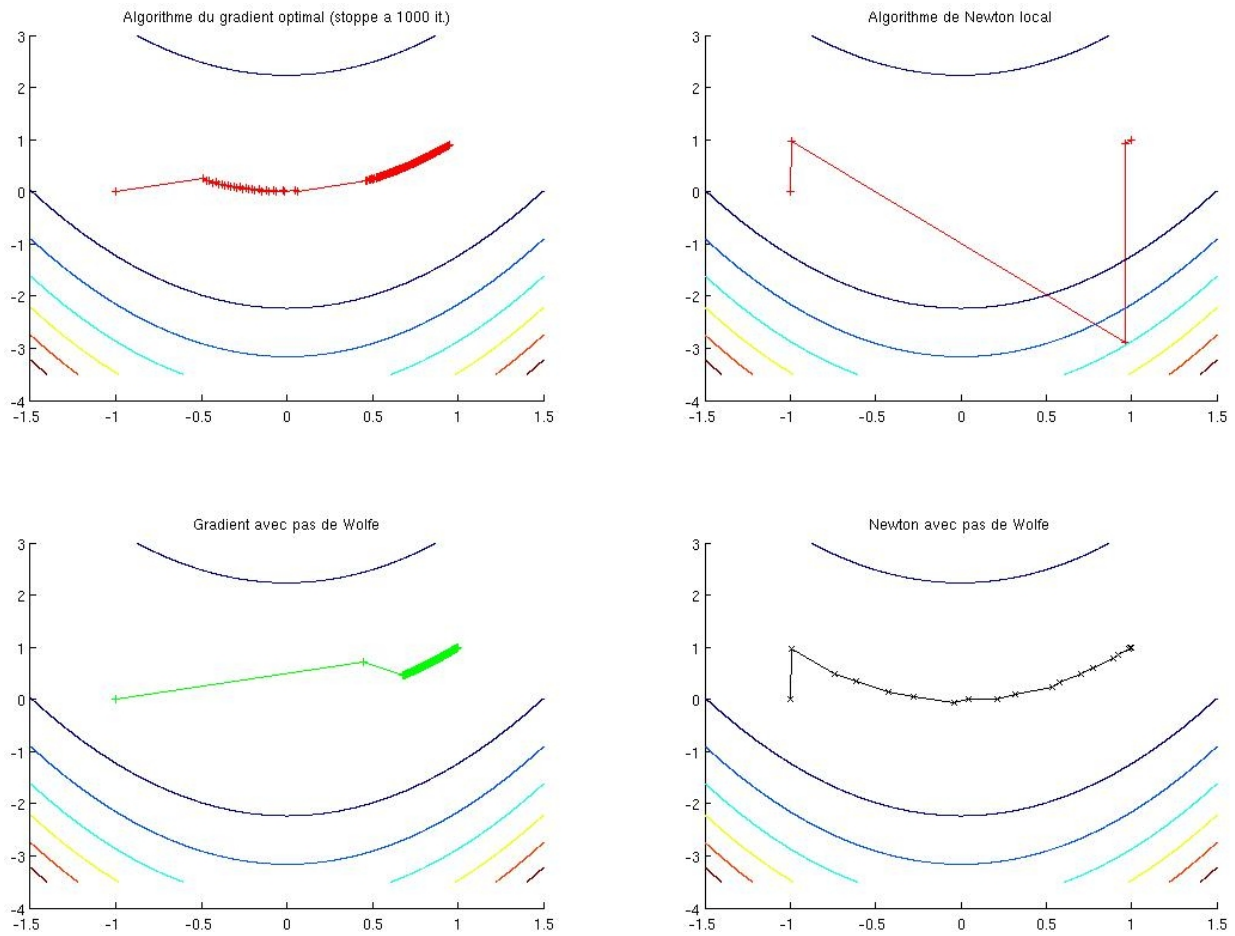


FIGURE 5.7 – Itérations des algorithmes de gradient optimal (stoppé à 10000 itérations!), de Newton (5 itérations), de gradient avec pas de Wolfe (8080 itérations) et de Newton avec pas de Wolfe (19 itérations) pour la minimisation de la fonction de Rosenbrock.

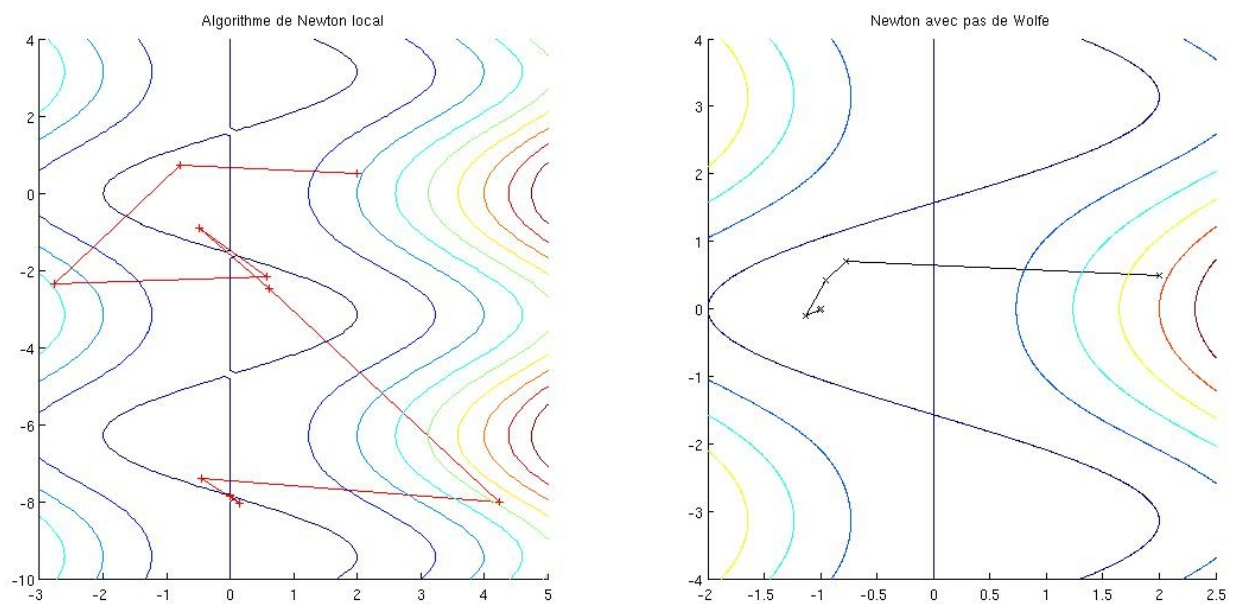


FIGURE 5.8 – Itérations des algorithmes de Newton (10 itérations) et de Newton avec recherche linéaire de Wolfe (6 itérations) pour la minimisation de la fonction  $g : (x, y) \mapsto \frac{1}{2}x^2 + x \cos y$ .

**Remarque 5.6** *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  $\alpha$  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 5.5 (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. □

### 5.4.2 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  $\nabla f$  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 :

$$\nabla f(x_{k+1}) - \nabla f(x_k) = H_{k+1}(x_{k+1} - x_k), \quad (5.12)$$

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 :

$$B_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k, \quad (5.13)$$

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 5.4.1** 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 5.6** 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 :  $x = 0$ . □

### 5.4.3 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 5.4** 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.

## 5.5 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 :

$$\min_{x \in \mathbb{R}^n} r(x) := \frac{1}{2} \|F(x)\|_2^2, \quad (5.14)$$

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 (5.14).

### 5.5.1 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 :

$$\min_{x \in \mathbb{R}^n} r(x) := \frac{1}{2} \|Ax - b\|_2^2 \quad (5.15)$$

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 (5.15) i.e. vérifiant :  $\nabla r(x) = 0$ , d'où le système suivant à résoudre :

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

EQUATIONS NORMALES

appelé *système d'équations normales* du problème de moindres carrés linéaire (5.15). Ainsi :

$$x^* \text{ est solution du problème (5.15) si et seulement si : } A^\top Ax^* = A^\top b.$$

De plus si  $A$  est de rang plein, alors la fonction  $r$  est strictement convexe et  $x^*$  est l'unique solution de (5.15).

Appliquons maintenant la méthode de Newton au problème (5.15) : la direction de Newton  $d_k = x_{k+1} - x_k$  est solution du système :

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

et ceci quel que soit  $x_k$ . On reconnaît ici le système d'équations normales du problème (5.15). Ceci signifie donc que la méthode de Gauss-Newton identifie la solution en une seule itération lorsque la fonction  $F$  est linéaire.

**Exercice 5.5.1** Soit  $F : \mathbb{R}^3 \longrightarrow \mathbb{R}^n$  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  $\nabla F_i$  et  $H[F_i]$ .
2. En déduire :  $J_r$  et  $H_r$  où  $r(x) = \frac{1}{2}F(x)^\top F(x)$ .

### 5.5.2 Algorithme de Gauss-Newton

Revenons au problème de moindres carrés non linéaire général (5.14) :

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

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 (5.14) 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 (5.14) par :

$$\min_{y \in \mathbb{R}^n} \tilde{r}(y) = \frac{1}{2} \|F(x_k) + J_F(x_k)(y - x_k)\|_2^2, \quad (5.16)$$

où l'on a remplacé  $F$  par une approximation du premier ordre en  $x_k$ . Le problème (5.16) 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 [9], 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.

### 5.5.3 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 (5.14) 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  $\lambda$ .
  - (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 6

## Algorithmes pour l'optimisation différentiable sous contrainte

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

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

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  $(\lambda^*, \mu^*)$ . 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  $(\lambda, \mu)$  ne servant souvent qu'à vérifier l'optimalité de  $x^*$  : elles engendrent une suite de solutions réalisables en assurant la décroissance de fonction objectif.

- Avantages : quand le processus itératif s'arrête, on obtient une solution approchée satisfaisant les contraintes.
- Inconvénients : mise au point délicate, convergence globale difficile à obtenir.

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*.

- Avantages : méthodes plus robustes que les méthodes primales, convergence globale plus facile à obtenir.
- Inconvénients : une solution primale réalisable n'est obtenue qu'en fin de convergence.

### 6.1 Méthode du gradient projeté

Revenons au problème général suivant :

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

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 :

$$\text{Minimiser } \frac{1}{2} \|x - y\|_2^2 \quad \text{sous la contrainte : } y \in X \quad (6.1)$$

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 (6.1) :

$$\forall y \in X, (x^* - x)^\top (y - x^*) \geq 0. \quad (6.2)$$

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 6.1** *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 6.1** 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 6.2** 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. :

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

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 6.1** 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  $f$ .

## 6.2 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  $(\lambda^k)$  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 :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t. :} \quad & h_i(x) = 0, \quad i = 1, \dots, p. \end{aligned} \quad (6.3)$$

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 :

$$H[L](x^k; \lambda^k) \begin{pmatrix} x^{k+1} - x^k \\ \lambda_{k+1} - \lambda^k \end{pmatrix} = -\nabla L(x^k; \lambda^k), \quad (6.4)$$

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 (6.4) 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 (6.4). Elle s'écrit encore :

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

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 :

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

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

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

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  $\lambda'$  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 :

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t. :} & g_j(x) \leq 0, \quad j = 1, \dots, q, \\ & h_i(x) = 0, \quad i = 1, \dots, p. \end{array} \quad (6.5)$$

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  $\lambda'$  et  $\mu'$  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  $(x_0, \lambda_0)$  (resp.  $(x_0, \lambda_0, \mu_0)$ ) 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 5.

## 6.3 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.



### 6.3.1 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  $\pi$ .

**Définition 6.1** 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  $\partial X$  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  $(\lambda, \mu)$  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.

### 6.3.2 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 [9] ou [5] entre autres).

Considérons le problème suivant :

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

Soit  $c > 0$  une constante arbitraire. On pénalise le problème  $(P)$  de la façon suivante :

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & g(x) := f(x) + \frac{c}{2} \|h(x)\|_2^2 \\ \text{s.t. :} & h_i(x) = 0, \quad i = 1, \dots, p. \end{array}$$

Le Lagrangien du problème pénalisé :

$$L_c(x; \lambda) = f(x) + \langle \lambda, h(x) \rangle + \frac{c}{2} \|h(x)\|_2^2 \quad (6.6)$$

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 6.1 ([5], 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 6.3.1** *Soit :*

$$\min_{x \in \mathbb{R}^2} \frac{1}{2}(-x_1^2 + x_2^2) \quad \text{sous} \quad x_1 = 1. \quad (6.7)$$

1. Calculer la solution optimale  $x^*$  du problème (6.7) et le multiplicateur  $\lambda^*$  associé.
2. Ecrire le Lagrangien augmenté  $L_c(x; \lambda)$  associé au problème (6.7).
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  $\lambda$ . 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  $\lambda$ , 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  $\lambda_0$  arbitraires,  $\varepsilon > 0$  précision.
2. Tant que  $\|\nabla L_{c_k}(x_k; \lambda_k)\| > \varepsilon$ ,
  - (a) **Approximation de la fonction duale en  $\lambda_k$ .** 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.**

$$\lambda_{k+1} = \lambda_k + c_k h(x_k);$$



# 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] D. Arzelier. Théorie de Lyapunov, commande robuste et optimisation. Technical report, Université Paul Sabatier - Toulouse III, <https://tel.archives-ouvertes.fr/tel-00010257>, 2004.
- [3] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.
- [4] M. Bierlaire. *Introduction à l'optimisation différentiable*. PPUR presses polytechniques, 2006.
- [5] J. Bonnans, J. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical optimization*. Universitext. Springer-Verlag, Berlin, second edition, 2006. Theoretical and practical aspects.
- [6] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [7] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I : fundamentals*, volume 305. Springer Science & Business Media, 2013.
- [8] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1) :125–161, 2013.
- [9] J. Nocedal and S. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.
- [10] R. T. Rockafellar. *Convex analysis*. Princeton university press, 2015.
- [11] C. Zalinescu. *Convex analysis in general vector spaces*. World Scientific, 2002.