

Remerciements

Au terme de ce travail, je tiens à exprimer ma profonde gratitude à Monsieur Ouanes, mon directeur de thèse, pour l'intérêt qu'il a porté à mon travail. Par ses précieux conseils. Grâce à lui, j'ai pu réaliser ce travail. Je lui en suis très reconnaissante.

Je remercie vivement l'ensemble des membres du jury pour l'honneur qu'ils me font en acceptant de juger ce travail.

Encore et avec une grande fierté et honneur que je tiens à présenter mes remerciements à toute la famille administrative du département mathématique, à tous ces enseignants qui nous ont permis d'acquérir des connaissances.

Je tiens à remercier ma chère mère qui ma aidée, encouragée et soutenue dans les moments difficiles tout au long de la préparation de ce mémoire.

Mes remerciements chaleureux s'adressent également aux camarades et amis Belkhir Rafik et Nora, Aît Ali lounis,... qui m'ont beaucoup aidée dans ce travail.

Je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste mémoire.

*A tous ceux qui de jour ou de nuit oeuvrent
pour l'avancement de la Science.*

Table des matières

Introduction	5
1 Éléments d'analyse convexe et conditions d'optimalité	9
1.1 Convexité	9
1.1.1 Ensemble convexe	9
1.1.2 Enveloppe convexe	11
1.1.3 Fonction convexe	11
1.2 Conditions d'optimalité	16
1.3 Conditions d'optimalité pour les problèmes sans contraintes . . .	16
1.3.1 Conditions nécessaires d'optimalité	16
1.3.2 Conditions suffisantes d'optimalité locale	18
1.3.3 Conditions suffisantes d'optimalité globale	19
1.3.4 Conditions d'optimalité pour les problèmes quadratiques .	20
1.4 Conditions d'optimalité pour les problèmes avec contraintes . . .	21
1.4.1 Conditions nécessaires Karush-Kuhn et Tucker	23
1.4.2 Interprétation géométrique des conditions de Karush-Kuhn et Tucker	24
1.4.3 Conditions suffisantes d'optimalité	26
2 Méthodes itératives d'optimisation sans contraintes	29
2.1 Dédution d'une classe d'algorithmes de descente	32
2.1.1 La recherche linéaire d'un pas	33
2.1.2 Méthode du gradient à pas optimal	34
2.1.3 Convergence de la méthode du gradient	35
2.2 La méthode de Newton	36
2.2.1 Convergence de la méthode de Newton	39
2.3 La méthode du gradient conjugué	41

2.3.1	Convergence de la méthode du gradient conjugué	47
2.3.2	La méthode du gradient conjugué pour une fonction quel- conque	48
2.4	La méthode de Nelder-Mead	50
2.4.1	Convergence de la méthode du Nelder-Mead	53
2.5	Etude comparative des méthodes d'optimisation sans contraintes	54
3		59
3.1	La méthode de Frank et Wolfe	59
3.1.1	Convergence de la méthode de Frank et Wolfe	63
3.2	Les méthodes de pénalités	63
3.2.1	Méthode de pénalité extérieure	65
3.2.2	Convergence de la pénalisation extérieure	67
3.2.3	Méthodes de pénalité intérieure	70
3.2.4	Convergence de la pénalisation intérieure	72
3.2.5	Etude comparative des méthodes d'optimisation avec contraintes	75
	Conclusion	81
	Bibliographie	83

Table des figures

1.1	<i>Interprétation géométrique d'un ensemble convexe</i>	10
1.2	<i>Illustration de la définition (enveloppe convexe)</i>	11
1.3	<i>Courbe représentative d'une fonction convexe</i>	12
1.4	<i>Courbe représentative d'une fonction d'appuie</i>	13
1.5	<i>Courbe représentative d'une fonction concave</i>	15
1.6	<i>Interprétation géométrique des conditions de Karush-Kuhn et Tucker</i>	25
1.7	<i>Illustration de la notion de point-col</i>	27
2.1	<i>Schéma général d'une méthode de descente</i>	33
2.2	<i>Interprétation géométrique de la méthode de Newton</i>	38
2.3	<i>Les mouvements possibles dans la méthode de Nelder-Mead . . .</i>	52

Liste des tableaux

2.1	<i>Exécution de la méthode du gradient pour (P1) depuis (1,1) . . .</i>	55
2.2	<i>Exécution de la méthode du gradient pour le problème (P1) depuis (2,27)</i>	55
2.3	<i>Exécution de la méthode de Newton pour le problème (P1) depuis (1,1)</i>	56
2.4	<i>Exécution de la méthode de Newton pour le problème (P1) depuis (2,27).</i>	56
2.5	<i>Exécution de la méthode du gradient conjuguée pour le problème (P1) depuis (1,1)</i>	57
2.6	<i>Exécution de la méthode du gradient conjuguée pour le problème (P1) depuis (2,27)</i>	57
2.7	<i>Exécution de la méthode de Nelder-Mead pour le problème (P1)</i>	58
3.1	<i>Exécution de la méthode de Frank Wolfe (Phase 1 du simplexe)</i>	76
3.2	<i>Exécution de la méthode de Frank Wolfe (Phase 2 du simplexe)</i>	77
3.3	<i>Exécution de la méthode de pénalité extérieure, pour le problème (P3) depuis un point non admissible (6,7)</i>	78
3.4	<i>Exécution de la méthode de pénalité intérieur, pour le problème (P3) depuis un point admissible (2.2)</i>	79

Introduction

<<Le désir humain de perfection trouve son expression dans la théorie de l'optimisation. Elle étudie comment décrire et atteindre ce qui est meilleur, une fois que l'on connaît comment et modifier ce qui est bon et ce qui est mauvais... la théorie de l'optimisation comprend l'étude quantitative des optimums et les méthodes pour les trouver >>[24].

A partir de la citation ci-dessus, on comprend la motivation d'un processus d'optimisation. En effet, l'optimisation cherche à améliorer une performance en se rapprochant d'un point optimal. Dès qu'on utilise un des mots minimum, minimiser, minimal, maximal, maximum, maximiser pour décrire une situation, il devient possible de formuler sous forme mathématique un problème d'optimisation.

L'optimisation est un domaine encore relativement jeune des mathématiques appliquées. Ce domaine des mathématiques, relativement récent, est appelé "programmation mathématique", et dans cette appellation il faut donner au mot "programmation" le sens de planification et non de programmation au sens de l'informatique. Les méthodes numériques de l'optimisation ont principalement été développées après la seconde guerre mondiale, en parallèle avec l'amélioration des ordinateurs, et n'ont cessé depuis de s'enrichir.

L'optimisation mathématique est utilisée dans plusieurs contextes. Dans notre vie quotidienne, nous cherchons à optimiser notre temps de travail, nos espaces de rangement, ou encore le trajet que nous aurons à parcourir pour nous rendre quelque part, etc. Nous cherchons tous une meilleure solution aux problèmes qui jalonnent notre existence. L'optimisation intervient, dans les processus de résolution des problèmes de production et de planification rencontrés dans les divers domaines de la vie tels que l'économie, l'industrie et le commerce, ou dans l'étude qualitative de certains problèmes de décision. Les gestionnaires désirent maximiser leurs profits, ou minimiser leurs coûts de production. Un

ingénieur souhaite minimiser le poids d'une structure sous contrainte que la structure soit capable de supporter X tonnes ou encore maximiser la force de la structure sous contrainte que son poids soit au plus X tonnes.

Selon la nature de la fonction objectif et de l'ensemble contraint, on distingue deux classes de problèmes: les problèmes d'optimisation linéaire dont on dispose aujourd'hui d'un arsenal théorique et numérique considérable pour les résoudre, et les problèmes non-linéaire dont l'étude est en plein essor. La programmation non linéaire regroupe un ensemble de sujets dans l'étude de problèmes d'optimisation. Pour une fonction objectif donnée, le problème consiste à trouver un point minimisant (ou maximisant) cette fonction. Parfois, tous les points de R^n sont candidats, parfois, des contraintes limitent le domaine de recherche. Les fonctions utilisées (fonction objectif, contraintes) sont continues, même différentiables. On utilise des résultats d'analyse mathématique pour caractériser les points candidats; un premier pas consiste donc à obtenir des conditions vérifiables satisfaites par les minima ou maxima recherchés. Lorsqu'un point ne satisfait pas à ces conditions d'optimalité, on en déduit une manière de calculer un point meilleur, et finalement un algorithme itératif réduisant (pour la minimisation) progressivement la fonction objectif.

Le présent projet a pour vocation de se focaliser sur les problèmes d'optimisation continus et non linéaires. Ceux-ci sont habituellement définis par un ensemble X appelé ensemble de solutions admissibles et une fonction objectif $f: X \mapsto R$ qui associe à chacun des éléments de X un nombre réel, ou un coût. Nous considérons le cas où X est un sous-ensemble de \mathbb{R}^n . Ses éléments sont donc des vecteurs de nombres réels à n dimensions de la forme (x^1, x^2, \dots, x^n) , avec $x^i \in \mathbb{R}$, $\forall i \in \{1, 2, \dots, n\}$. Le problème consiste à trouver l'élément de X dont le coût est minimal ou maximal, (mais cela ne change rien à la difficulté ou aux types de méthodes employées). Cela peut être formulé de manière plus concise comme suit :

$$\left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{sous contrainte} \\ x \in X \end{array} \right.$$

X est généralement défini par une collection de contraintes exprimées sous forme d'égalités et d'inégalités. Nous admettons que f est continue et continuellement différentiable, et parfois même nous admettrons que f est deux fois continuellement différentiable. Les dérivées seconde et première jouent un rôle important aussi bien dans la caractérisation des solutions optimales que dans les idées qui conduisent à la plupart d'algorithmes utilisés et

implémentés au cours de ce projet.

Par optimisation sans contraintes, nous désignons le cas particulier où $X = \mathbb{R}^n$. Cela peut être formulé de manière suivante :

$$\begin{cases} \text{Minimiser} & f(x) \\ x \in \mathbb{R}^n \end{cases}$$

Parfois, les algorithmes de résolution sont fondés sur les mêmes principes lorsque le problème est contraint ou lorsqu'il ne l'est pas. Ils nécessitent seulement d'être adaptés afin de produire des résultats convenables dans les deux cas. En d'autres termes, certains algorithmes pour l'optimisation non linéaire avec des contraintes sont dérivés des algorithmes d'optimisation non linéaire sans contraintes, avec quelques modifications destinées à prendre en charge ces dernières (notamment à ne pas sortir de X).

Les différents résultats auxquels nous avons aboutis sont exposés dans les chapitres constituant ce mémoire, et qui se décomposent comme suit :

Chapitre 1 : Éléments d'analyse convexe et conditions d'optimalité:

Ce chapitre se veut être, un aperçu panoramique sur les différentes notions et techniques à lesquelles nous avons eu recours dans l'implémentation des méthodes d'optimisations. Nous procédons donc, par une représentation progressive allant de la notion de convexité jusqu'à l'exposition des conditions d'optimalité.

Chapitre 2 : Méthodes itératives d'optimisation sans contraintes:

Dans ce chapitre, nous abordons la description des algorithmes pour résoudre les problèmes d'optimisation non-linéaires sans contraintes. Nous étudions d'abord les méthodes de descente, ensuite les méthodes du gradient conjugué et on termine par la méthode de Mead et Neald. Nous achevons ce chapitre par une étude comparative des résultats de l'implémentation.

Chapitre 3 : Méthodes itératives d'optimisation avec contraintes:

Ce chapitre traite l'optimisation non linéaire avec contraintes et comporte trois méthodes, à savoir la méthode de Frank Wolfe, et les méthodes de pénalités intérieur et extérieur. Dans la partie finale de ce chapitre, nous donnerons quelques conclusions liées à la mise en oeuvre de ces méthodes.

A ce stade nous allons rappeler brièvement les définitions de quelques notions importantes auxquelles nous ferons appel par la suite.

Minima locaux et globaux

Soient l'ensemble $\mathcal{S} \in \mathbb{R}^n$ et une fonction $f: \mathcal{S} \mapsto \mathbb{R}$. Les minima locaux et globaux de f sur \mathcal{S} sont définis de la manière suivante :

Minimum local

Intuitivement, un vecteur $x^* \in \mathcal{S}$ est un minimum local de f sur \mathcal{S} s'il a un coût plus faible que celui de ses voisins. Formellement, x^* est un minimum local de f sur \mathcal{S} si $\exists \epsilon > 0$ tel que

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{S} \quad |x - x^*| < \epsilon$$

où $|\cdot|$ désigne la norme.

Le minimum local est strict si

$$f(x^*) < f(x) \quad \forall x \in \mathcal{S} \quad |x - x^*| < \epsilon$$

Minimum global

Un vecteur $x^* \in \mathcal{S}$ est un minimum global de f sur \mathcal{S} s'il a un coût plus faible que celui de tous les autres vecteurs dans \mathcal{S} . Formellement, x^* est un minimum global de f sur \mathcal{S} si

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{S}$$

Le minimum global est strict si

$$f(x^*) < f(x) \quad \forall x \in \mathcal{S}$$

Chapitre 1

Eléments d'analyse convexe et conditions d'optimalité

Introduction

Pour étudier les problèmes d'optimisation, il est nécessaire de recourir à des outils spécifiques dont l'étude est basée sur l'analyse convexe. En effet, l'hypothèse de convexité va jouer un rôle très important pour la plupart des algorithmes que nous décrirons, la convergence vers un optimum global ne pourra être démontrée qu'avec ces hypothèses . Dans la première section de ce chapitre, nous introduirons rapidement quelques éléments de l'analyse convexe requis pour l'étude de l'optimisation. Nous aborderons ensuite deux conditions souvent retenues pour la résolution de ce type de problèmes : les conditions nécessaires et suffisantes d'optimalité.

1.1 Convexité

1.1.1 Ensemble convexe

Définition 1.1.1. "*Ensemble Convexe*"

Un ensemble $A \subset \mathbb{R}^n$ dit convexe si, pour tous points x et y de A , le segment $[x,y]$ est inclus dans A ,

$$\forall x \in A, \quad \forall y \in A, \quad \forall \lambda \in [0,1], \quad \lambda x + (1 - \lambda)y \in A \quad (1.1)$$

Autrement dit, un sous-ensemble convexe contient toujours le segment $[x, y]$ joignant deux de ses points x et y . Une interprétation "optique" consiste à dire que dans une pièce convexe, deux personnes peuvent toujours s'apercevoir (voir FIG 1.1).

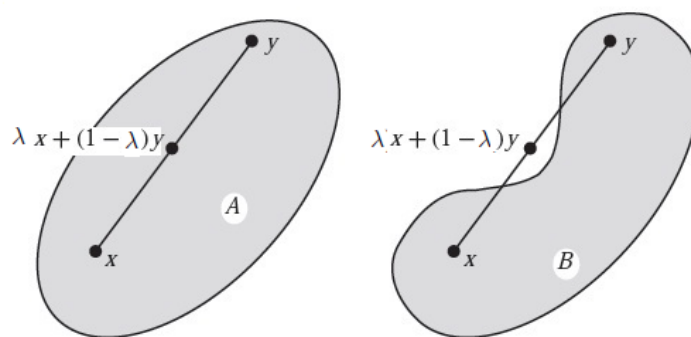


FIG. 1.1 – *Interprétation géométrique d'un ensemble convexe*

Propriétés 1.1.1.

L'intersection d'un nombre fini quelconque d'ensembles convexes est convexe.

Définition 1.1.2. "Combinaison Convexe"

On appelle combinaison convexe de n points $\{x_i\}_{i=1,\dots,n}$ tout point y obtenu par la formule

$$y = \sum_{i=1}^{i=n} \lambda_i x_i \quad \text{avec} \quad \sum_{i=1}^{i=n} \lambda_i = 1 \quad \text{avec} \quad \lambda_i \geq 0 \quad (1.2)$$

Théorème 1.1.1. [1]

Un ensemble A est convexe si et seulement si toute combinaison convexe des points de A appartient à A

Remarque 1.1.1.

L'intersection de sous-ensembles convexes est convexe.

- L'union de sous-ensembles convexes n'est pas convexe en général, mais l'union croissante de convexes (famille emboîtée) est convexe.

1.1.2 Enveloppe convexe

Étant donné un sous-ensemble A de X , l'espace X est un convexe contenant A . De plus, l'intersection de convexes contenant A étant convexes et contenant A , on peut poser la définition suivante.

Définition 1.1.3. "Enveloppe Convexe"

L'enveloppe convexe d'un sous-ensemble $A \subset X$ quelconque est le plus petit convexe (au sens de l'inclusion) qui contient A , elle est notée $\text{Conv}(A)$. Cette définition est illustrée dans la figure (1.2), en sombre c'est l'enveloppe convexe et en clair le sous-ensemble.

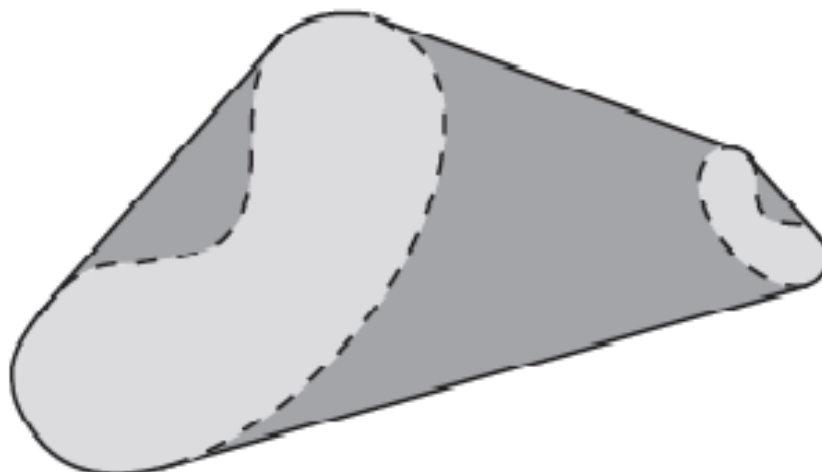


FIG. 1.2 – *Illustration de la définition (enveloppe convexe)*

La construction de l'enveloppe convexe suggérée ci-dessus (intersection des convexes contenant A) est une construction "externe". Le théorème suivant propose une construction "interne" de $\text{Conv}(A)$

1.1.3 Fonction convexe

Définition 1.1.4. "Fonction convexe"

Une fonction $f: \mathbb{R}^n \rightarrow \mathbb{R}$ est dite convexe, si elle vérifie :

$$\forall x \in \mathbb{R}^n, \quad \forall y \in \mathbb{R}^n, \quad \forall \lambda \in [0,1], \quad f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad (1.3)$$

Cette inégalité est appelée inégalité de convexité. Elle signifie que toute corde de φ_f est située au dessus de l'arc de φ_f soutenant la corde.

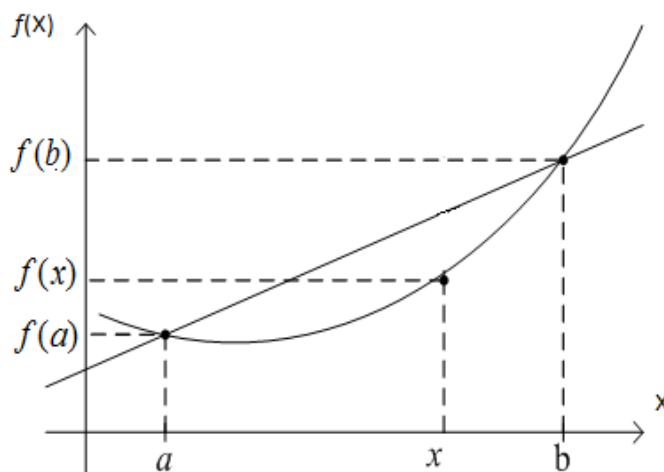


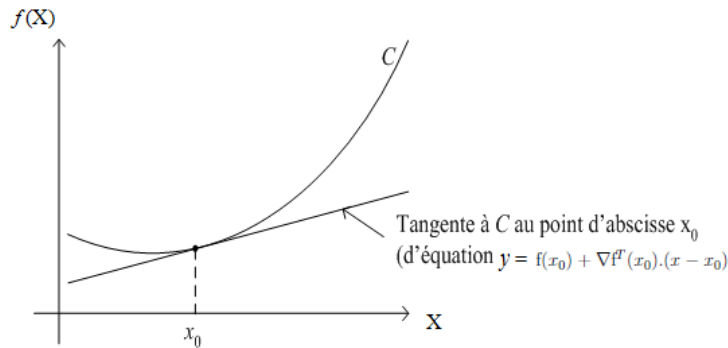
FIG. 1.3 – *Courbe représentative d'une fonction convexe*

Le segment de droite reliant les points $(a, f(a))$ et $(b, f(b))$ se trouve totalement au-dessus du graphe de f . Le point $x = \lambda a + (1 - \lambda)b$ est situé quelque part entre a et b . Le point de coordonnées $(x, \lambda f(a) + (1 - \lambda)f(b))$ se trouve sur le segment de droite entre les points $(a, f(a))$ et $(b, f(b))$. Pour que la fonction soit convexe, il faut que ce point se trouve toujours (c'est-à-dire pour tout a, b et $0 < \lambda < 1$) au-dessus du graphe de la fonction.

Théorème 1.1.2.

Si f est continûment différentiable, les conditions (1) et (2) ci-dessous sont équivalentes; Si f est deux fois continûment différentiable, les conditions (1), (2) et (3) ci-dessous sont équivalentes :

1. f est convexe;
2. $\forall x \in \mathbb{R}^n, \forall x_0 \in \mathbb{R}^n, f(x) \geq f(x_0) + \nabla f^T(x_0) \cdot (x - x_0)$; c-à-d la courbe C de f est au dessus de ses tangentes.

FIG. 1.4 – *Courbe représentative d'une fonction d'appuie*

3. $\forall x$, le hessien $\nabla^2 f(x)$ est une matrice semi-définie positive, c-à-d

$$\forall y, \quad y^T \cdot \nabla^2 f(x) \cdot y \geq 0. \quad (1.4)$$

Remarque 1.1.2.

- f est dite strictement convexe si l'inégalité stricte est toujours vérifiée pour $x \neq y$ et $\lambda \in [0,1]$

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y)$$

- f est dite fortement convexe, s'il existe $\alpha > 0$ tel que $\forall x \in \mathbb{R}^n, \quad \forall y \in \mathbb{R}^n,$
 $\forall \lambda \in [0,1],$

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{1}{2}\alpha\lambda(1 - \lambda)\|x - y\|^2 \quad (1.5)$$

On dit aussi que f est α -convexe.

- La forte convexité est le cadre agréable pour de nombreux problèmes d'optimisation, car elle donne facilement l'existence, l'unicité, et des algorithmes de calcul performants; c'est une hypothèse forte, mais elle inclut le cas des fonctions quadratiques, très important en pratique

Définition 1.1.5. "L'épigraphe"

On appelle l'épigraphe d'une fonction f noté $\text{épi}(f)$ l'ensemble défini par :

$$\text{épi}(f) = \{(x,r) \in A \times \mathbb{R}, f(x) \leq r\}$$

Proposition 1.1.1. "Critère de l'épigraphe"

f est convexe si et seulement si son épigraphe est convexe, où

$$\text{épi}(f) = \{(x,r) \in A \times \mathbb{R}, f(x) \leq r\}$$

Preuve.

f convexe \Leftrightarrow ? son épigraphe est convexe

1. f convexe \Rightarrow ? son épigraphe est convexe

Soient $(x_1, r_1), (x_2, r_2) \in \text{épi}(f)$ et $\lambda \in [0,1]$

On a $\lambda(x_1, r_1) + (1 - \lambda)(x_2, r_2) = (\lambda x_1 + (1 - \lambda)x_2, \lambda r_1 + (1 - \lambda)r_2)$

$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$ car f est convexe

$\leq \lambda r_1 + (1 - \lambda)r_2$ car $(x_1, r_1), (x_2, r_2) \in \text{épi}(f)$

D'où $\lambda(x_1, r_1) + (1 - \lambda)(x_2, r_2) \in \text{épi}(f) \Rightarrow \text{épi}(f)$ est convexe (1)

2. épigraphe de f est convexe \Rightarrow ? f convexe

Soient $(x_1, r_1), (x_2, r_2) \in \text{épi}(f)$ et $\lambda \in [0,1]$

On a $\lambda(x_1, r_1) + (1 - \lambda)(x_2, r_2) = (\lambda x_1 + (1 - \lambda)x_2, \lambda r_1 + (1 - \lambda)r_2)$

$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda r_1 + (1 - \lambda)r_2$

On pose $r_1 = f(x_1)$ et $r_2 = f(x_2)$

D'où $f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \Rightarrow f$ est convexe (2)

De (1) et (2) on a f convexe \Leftrightarrow épigraphe convexe

Définition 1.1.6. "Fonction concave"

Une fonction $f: \mathbb{R}^n \rightarrow \mathbb{R}$ est dite concave si $(-f)$ est une fonction convexe, i.e. avec les mêmes notations, si

$$\forall x \in \mathbb{R}^n, \forall y \in \mathbb{R}^n, \forall \lambda \in [0,1], f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y) \quad (1.6)$$

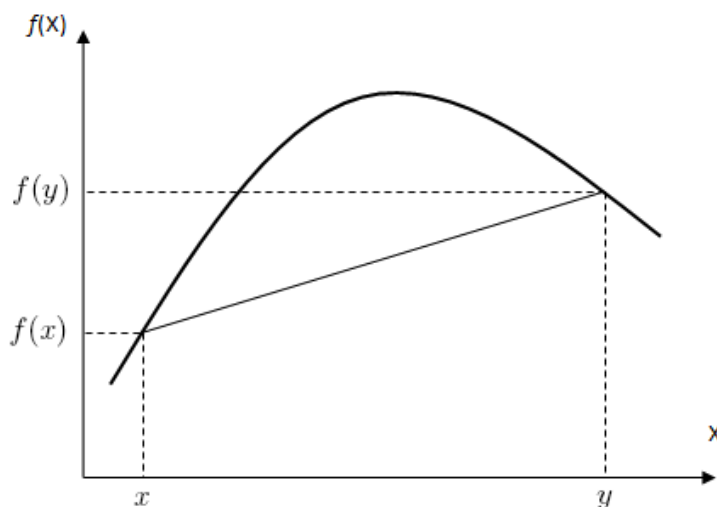


FIG. 1.5 – Courbe représentative d'une fonction concave

Remarque 1.1.3.

a) La convexité de la fonction est une caractéristique très importante en optimisation. En effet, lorsque la fonction n'est pas convexe, il est pratiquement impossible d'identifier un optimum global d'un problème d'optimisation. Notons que l'importance de la convexité est liée aux problèmes de minimisation. Lorsque l'on étudie des problèmes de maximisation, on utilise la notion de concavité.

b) Noter que convexité et concavité ne sont pas des propriétés complémentaires. Une fonction peut n'être ni convexe ni concave.

1.2 Conditions d'optimalité

Avant de développer des algorithmes permettant d'identifier des solutions d'un problème d'optimisation, il faut être capable de décider si un point donné est optimal ou non. Ces conditions d'optimalité ont trois rôles essentiels dans le développement des algorithmes

1. elles procurent une analyse théorique du problème,
2. elles inspirent directement des idées pour les algorithmes,
3. elles permettent de déterminer un critère d'arrêt pour les algorithmes itératifs.

1.3 Conditions d'optimalité pour les problèmes sans contraintes

Considérons le problème d'optimisation sans contrainte suivant

$$(P) \begin{cases} \text{Min } f(x) \\ x \in \mathbb{R}^n \end{cases}$$

On cherche à résoudre (P) , donc il s'agit de déterminer un point x^* de \mathbb{R}^n tel que :

$$\forall x \in \mathbb{R}^n, \quad f(x^*) \leq f(x) \quad (1.7)$$

c'est-à-dire un minimum global de f sur \mathbb{R}^n . Mais pour beaucoup de problèmes d'optimisation sans contrainte, les principales méthodes de résolution connues ne permettent pas la détermination d'un minimum global, il faut alors se contenter d'un minimum local, c'est-à-dire des points qui vérifient (1.7) seulement dans un voisinage de x^* . Nous allons voir maintenant comment de tels points peuvent être caractérisés.

1.3.1 Conditions nécessaires d'optimalité

On suppose que $f(x)$ est continue et a des dérivées partielles premières $\partial f/\partial x_i$ et secondes $\partial^2 f/\partial x_i \partial x_j$ continues pour tout $x \in \mathbb{R}^n$.

Théorème 1.3.1. [2]”*Conditions nécessaires d'optimalité*”

Une condition nécessaire pour que x^* soit un minimum local de f est :

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

b) si f est deux fois différentiable, alors

$$\nabla^2 f(x^*) = [\partial^2 / \partial x_i \partial x_j](x^*) \text{ est une matrice semi-définie positive.}$$

La condition (a) est appelée **condition nécessaire du premier ordre**, la condition (b) est appelée **condition nécessaire du second ordre**.

Preuve.

Soit x^* un minimum local de f .

Comme f est deux fois différentiable, le développement de Taylor au voisinage de x^* donne :

$$f(x) = f(x^*) + \nabla f(x^*)(x - x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*) + \|x - x^*\|^2 \theta(x - x^*)$$

Avec $\theta(x - x^*) \rightarrow 0$ quand $x \rightarrow x^*$

1. Si $\nabla f(x^*) \neq 0$ alors en choisissant $x = x^* - \theta \nabla f(x^*)$ on aurait, pour $\theta > 0$ suffisamment petit : $f(x) < f(x^*)$ ce qui contredirait le fait que x^* est un minimum local. Donc la condition (1) est bien nécessaire.
2. Si la matrice $\nabla^2 f(x^*)$ n'est pas semi-définie positive, c'est qu'il existe un vecteur $d \in \mathbb{R}^n (d \neq 0)$ tel que : $d^T \nabla^2 f(x^*) d < 0$.
En choisissant alors $x = x^* + \theta d$, pour $\theta > 0$ suffisamment petit on aurait $f(x) < f(x^*)$ ce qui contredirait encore l'optimalité locale de x^* .

Remarque 1.3.1.

En pratique, la condition nécessaire du second ordre est difficile à vérifier systématiquement, car elle exige de calculer les dérivées secondes et d'analyser les valeurs propres de la matrice hessienne.

La condition nécessaire d'optimalité du premier ordre joue un rôle central en optimisation. Les vecteurs x qui vérifient cette condition sont appelés des points critiques ou points stationnaires. Parmi eux, il y a des minima locaux, des maxima locaux et des points qui ne sont ni l'un ni l'autre. Ces derniers sont appelés points selle.

Définition 1.3.1. "Point critique"

Soit $f: \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Tout vecteur $x \in \mathbb{R}^n$ qui vérifie la condition (a) c'est-à-dire : $\nabla f(x) = 0$ est appelé un point stationnaire ou point critique de f .

1.3.2 Conditions suffisantes d'optimalité locale

Théorème 1.3.2 (2). "Conditions suffisantes d'optimalité local"

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Une condition suffisante pour que x^* soit un optimum local de f est:

1. $\nabla f(x^*) = 0$ (stationnarité)
2. La hessienne $\nabla^2 f(x^*)$ est une matrice définie positive

Preuve.

Considérons un point x^* satisfaisant les deux conditions (1) et (2) du théorème.

Le développement de Taylor de f au voisinage de x^* s'écrit alors :

$$f(x) = f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*) + \|x - x^*\|^2 \theta(x - x^*)$$

Avec $\theta(x - x^*) \rightarrow 0$ quand $x \rightarrow x^*$

Pour toute direction de déplacement $d \in \mathbb{R}^n$ ($\|d\| = 1$) on a :

$$f(x^* + td) = f(x^*) + \frac{t^2}{2} d^T \nabla^2 f(x^*) d + t^2 \theta(t)$$

d'où

$$f(x^* + td) - f(x^*) = \frac{t^2}{2} d^T \nabla^2 f(x^*) d + t^2 \theta(t)$$

où $\theta(t) \rightarrow 0$ quand $t \rightarrow 0$.

En vertu de la condition (2) on a :

$$\frac{t^2}{2} d^T \nabla^2 f(x^*) d > 0$$

et par suite, pour θ suffisamment petit, on a

$$f(x^* + td) > f(x^*)$$

Ce qui montre que x^* est bien un minimum local de f .

Nous présentons maintenant une condition suffisante pour qu'un minimum local soit également un minimum global.

1.3.3 Conditions suffisantes d'optimalité globale

Dans le cas d'une fonction convexe f définie sur \mathbb{R}^n , on a la condition nécessaire et suffisante pour que x^* soit un minimum global de f est donnée dans le théorème suivant:

Théorème 1.3.3. [3] "Conditions suffisantes d'optimalité globale"

Soit une fonction continue $f: \mathbb{R}^n \rightarrow \mathbb{R}$ et $x^* \in \mathbb{R}^n$ un minimum local de f .

- Si f est une fonction convexe, alors x^* est un minimum global de f .
- Si de plus f est strictement convexe, x^* est l'unique minimum global de f .

Autrement dit, dans le cas convexe, la stationnarité à elle seule constitue une condition nécessaire et suffisante d'optimalité globale.

Preuve.

Supposons par l'absurde qu'il existe un autre minimum local $x^+ \neq x^*$, tel que $f(x^+) < f(x^*)$. Par la convexité de f , nous avons

$$f(\lambda x^* + (1 - \lambda)x^+) \leq \lambda f(x^*) + (1 - \lambda)f(x^+),$$

où $0 \leq \lambda \leq 1$. Comme $f(x^+) < f(x^*)$, nous avons pour chaque $\lambda \in]0,1[$

$$f(\lambda x^* + (1 - \lambda)x^+) < \lambda f(x^*) + (1 - \lambda)f(x^*) = f(x^*), \quad (1.8)$$

Considérons $\varepsilon > 0$ arbitraire, et montrons que la définition d'un minimum local est contredite.

Si $\varepsilon \geq \|x^* - x^+\|$, la définition d'un minimum local n'est pas vérifiée pour $x = x^+$, en prenant $\lambda = 1$ dans (1.8).

Si $\varepsilon < \|x^* - x^+\|$, considérons $0 < \eta < 1$ tel que $\varepsilon < \|\eta x^* + (1 - \eta)x^+\| = \varepsilon$. Dans ce cas, la définition d'un minimum local n'est pas vérifiée pour $x = (\lambda x^* + (1 - \lambda)x^+)$ avec $\eta \leq \lambda < 1$ par (1.8). Comme $\eta < 1$, de tels λ existent toujours.

Considérons maintenant une fonction strictement convexe, et supposons que x^* et y^* soient deux minima globaux distincts, et donc $x^* \neq y^*$ et $f(x^*) = f(y^*)$. En fonction de la définition de convexité stricte, nous avons

$$f(\lambda x^* + (1 - \lambda)y^*) < \lambda f(x^*) + (1 - \lambda)f(y^*) = f(x^*) = f(y^*),$$

ce qui contredit que x^* et y^* sont des minima globaux.

Nous terminons cette section par une discussion des conditions d'optimalité pour les problèmes quadratiques.

1.3.4 Conditions d'optimalité pour les problèmes quadratiques

Parmi les fonctions non linéaires, les fonctions quadratiques joueront un rôle important dans les algorithmes d'optimisation.

Définition 1.3.2. "Fonction quadratique"

Une fonction $f: \mathbb{R}^n \rightarrow \mathbb{R}$ sera dite quadratique si elle peut s'écrire

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c \quad (1.9)$$

où A est une matrice symétrique $n \times n$, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$. Nous avons alors

$$\nabla f(x) = Ax + b \quad (1.10)$$

et

$$\nabla^2 f(x) = A \quad (1.11)$$

Théorème 1.3.4. [3] "Conditions d'optimalité pour les problèmes quadratiques"

Considérons le problème

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}x^T Ax + b^T x + c \quad (1.12)$$

où A est une matrice symétrique $n \times n$, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$.

1. Si A n'est pas semi-définie positive, alors le problème (1.12) ne possède pas de solution, c'est-à-dire qu'il n'existe aucun $x \in \mathbb{R}^n$ qui soit un minimum local de (1.12).
2. Si A est définie positive, alors

$$x^* = -A^{-1}b \quad (1.13)$$

est l'unique minimum global de (1.12)

Preuve.

Nous avons $\nabla f(x) = Ax + b$ et $\nabla^2 f(x) = A$

1. Supposons par l'absurde qu'il existe x^* minimum local de (1.12). D'après (b) du théorème (1.3.1), $\nabla^2 f(x) = A$ est semi-définie positive, ce qui contredit l'hypothèse.

2. Comme A est définie positive, le point x^* dans (1.13) est bien défini car :

$$\nabla f(x^*) = Ax^* + b$$

On remplace x^* par sa valeur, on aura

$$\nabla f(x^*) = -AA^{-1}b + b = 0$$

Les conditions suffisantes d'optimalité du théorème (1.3.2) sont vérifiées et x^* est un minimum local de f . De plus, f est convexe. D'après le théorème (1.3.3), x^* est l'unique minimum global.

1.4 Conditions d'optimalité pour les problèmes avec contraintes

Cette section est une courte introduction à l'optimisation sous contrainte. On s'intéresse aux conditions d'optimalité pour les problèmes avec contraintes de la forme :

$$\begin{cases} \min f(x), \\ \text{sous} & \text{contraintes} \\ x \in X, \end{cases} \quad (1.14)$$

L'écriture des conditions d'optimalité en présence de contraintes est basée sur la même intuition que dans le cas sans contraintes, à savoir qu'il est impossible de descendre à partir d'un minimum.

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 des contraintes.

Pour cela, on introduit la notion de direction admissible et qualification des contraintes.

Directions admissibles et qualification des contraintes

Définition 1.4.1. "Directions admissibles"

Soit le problème général d'optimisation (1.14), et soit un point $x \in \mathbb{R}^n$ admissible. Une direction d sera dite admissible en x s'il existe $\eta > 0$ tel que $x + \alpha d$ soit admissible pour tout $0 < \alpha \leq \eta$.

Définition 1.4.2. "Les indices actifs"

Soit l'ensemble des contraintes $X = \{x \in \mathbb{R}^n / g_i(x) \leq 0, \quad i \in \mathbb{I} = \{1, 2, \dots, m\}\}$. L'ensemble des indices actifs (L'ensemble des indices des contraintes saturées) au point x^* est formé des indices i qui vérifient $g_i(x) = 0$. Il est noté comme suit :

$$\mathbb{I}_a = \mathbb{I}_a(x^*) = \{i \in \mathbb{I} / g_i(x^*) = 0\} \quad (1.15)$$

Définition 1.4.3. "Qualification des contraintes (Q.c)"

Soit un problème d'optimisation (1.14), et soit un point admissible x^* . La condition de qualification des contraintes (Kuhn et Tucker 1961, Abadie 1967)^[4] est vérifiée si tout élément du cône des directions en x^* est une direction admissible à la limite en x^* .

Donc (Q.c) en x^* , revient à chercher l'existence d'une direction d pointant à l'intérieur du cône défini par :

$$K = \{d \in \mathbb{R}^n / \nabla^T g_i(x^*)d \leq 0, \quad \forall i \in \mathbb{I}_a(x^*)\}. \quad (1.16)$$

C'est à dire que :

$$(Q.c) \iff \exists d \in \mathbb{R}^n / \nabla^T g_i(x^*)d \leq 0, \quad i \in \mathbb{I}_a(x^*). \quad (1.17)$$

Evidemment, la vérification directe de (Q.c) peut être difficile en pratique, et c'est pourquoi on a recherché des conditions suffisantes pour que (Q.c) soit réalisée. Les résultats les plus importants sont rassemblés dans le lemme ci-dessous.

Lemme 1.4.1. [3]

Pour que (Q.c) soit vérifiée en tout point $x \in X$ il suffit que l'une des conditions (a) ou (b) soit réalisée :

a) toutes les fonctions g_i , sont linéaires (Karlin 1959)^[5],

b) toutes les fonctions g_i sont convexes et X a un intérieur non vide (Slater 1950)^[6].

Pour que (Q.c) soit vérifiée en un point $x^* \in X$, il suffit que l'on ait :

c) les gradients $\nabla g_i(x^*) (i \in \mathbb{I}^*)$ des contraintes saturées en x^* sont linéairement indépendants (Fiacco et McCormick 1968)^[7].

1.4.1 Conditions nécessaires Karush-Kuhn et Tucker

Les conditions nécessaires de Karush-Kuhn et Tucker (K.K.T)^[8] donnée dans le théorème suivant est fondamentale. Sous l'hypothèse de qualification des contraintes, une condition nécessaire d'optimalité locale pour un problème d'optimisation avec contraintes du type

$$\begin{cases} \min f(x) \\ \text{Sous les contraintes} \\ g_i(x) \leq 0 \\ x \in \mathbb{R}^n \end{cases} \quad i \in \mathbb{I} \quad (1.18)$$

Théorème 1.4.1. [8] "Karush-Kuhn et Tucker 1951"

On suppose que les fonctions f et $g_i (i \in \mathbb{I})$ sont continûment différentiables, et que l'hypothèse de qualification des contraintes est vérifiée en $x^* \in X$, avec :

$$X = \{x \in \mathbb{R}^n / g_i(x) \leq 0, \quad \forall i \in \mathbb{I}\}.$$

Alors, une condition nécessaire pour que x^* soit un optimum local de (1.18) est qu'il existe des nombres $\lambda_i \geq 0 (i \in \mathbb{I})$ appelés multiplicateurs de Karush-Kuhn et Tucker, tels que :

$$\begin{cases} \nabla f(x^*) + \sum_{i \in \mathbb{I}} \lambda_i \nabla g_i(x^*) = 0 \quad , \\ \text{et,} \\ \lambda_i \nabla g_i(x^*) = 0 \quad (i \in \mathbb{I}) \quad . \end{cases} \quad (1.19)$$

Pour démontrer ce théorème, nous aurons besoin des deux résultats suivants :

Lemme 1.4.2. [3] "Farkas"

Soit A une matrice $n \times m$, b un vecteur de \mathbb{R}^n , avec $m < n$.

Si pour chaque vecteur x vérifiant $Ax \leq b$, et $cx \leq 0$ alors il existe des coefficients $\lambda_i \geq 0$, tels que

$$c = \sum_{i=1}^m \lambda_i A_i$$

Lemme 1.4.3. [3]

Soit d une direction admissible en x^* . Alors nécessairement, d vérifie les relations :

$$\nabla g_i^T(x^*).d \leq 0 \quad (\forall i \in \mathbb{I}_a). \quad (1.20)$$

Preuve. "Théorème de Karush-Kuhn-Tucker"

Le point x^* étant un minimum, alors pour toute direction admissible d en x^* , on aura $\nabla f(x^*)^T.d \geq 0$, et ce, en vertu du lemme (1.4.3). En utilisant (1.20), on peut donc écrire

$$\nabla g_i^T(x^*).d \leq 0 \quad (\forall i \in \mathbb{I}_a) \implies -\nabla f(x^*)^T.d \leq 0$$

En vertu du lemme de Ferkas (1.4.2), il existe alors des coefficients λ_i^* , ($i \in \mathbb{I}_a$), tels que

$$-\nabla f(x^*) = \sum_{i \in \mathbb{I}_a} \lambda_i^* A_i$$

Posons $\lambda_i^* = 0$ pour $i \in \mathbb{I} \setminus \mathbb{I}_a(x^*)$. Donc on obtient

$$\begin{cases} -\nabla f(x^*) = \sum_{i \in \mathbb{I}} \lambda_i^* A_i \iff \nabla f(x^*) + \sum_{i \in \mathbb{I}} \lambda_i^* \nabla g_i(x^*) = 0 & , \\ \text{et,} \\ \lambda_i^* \nabla g_i(x^*) = 0 & (i \in \mathbb{I}) \end{cases} .$$

Remarque 1.4.1.

Dans le cas où les gradients des contraintes saturées sont linéairement indépendants (x^* est alors appelé un point régulier) il est important de noter que le vecteur de *Kuhn et Tucker* λ est unique.

1.4.2 Interprétation géométrique des conditions de Karush-Kuhn et Tucker

Les conditions de Karush-Kuhn et Tucker peuvent s'interpréter géométriquement de la façon suivante (fig.1.6). Sous l'hypothèse de qualification des contraintes.

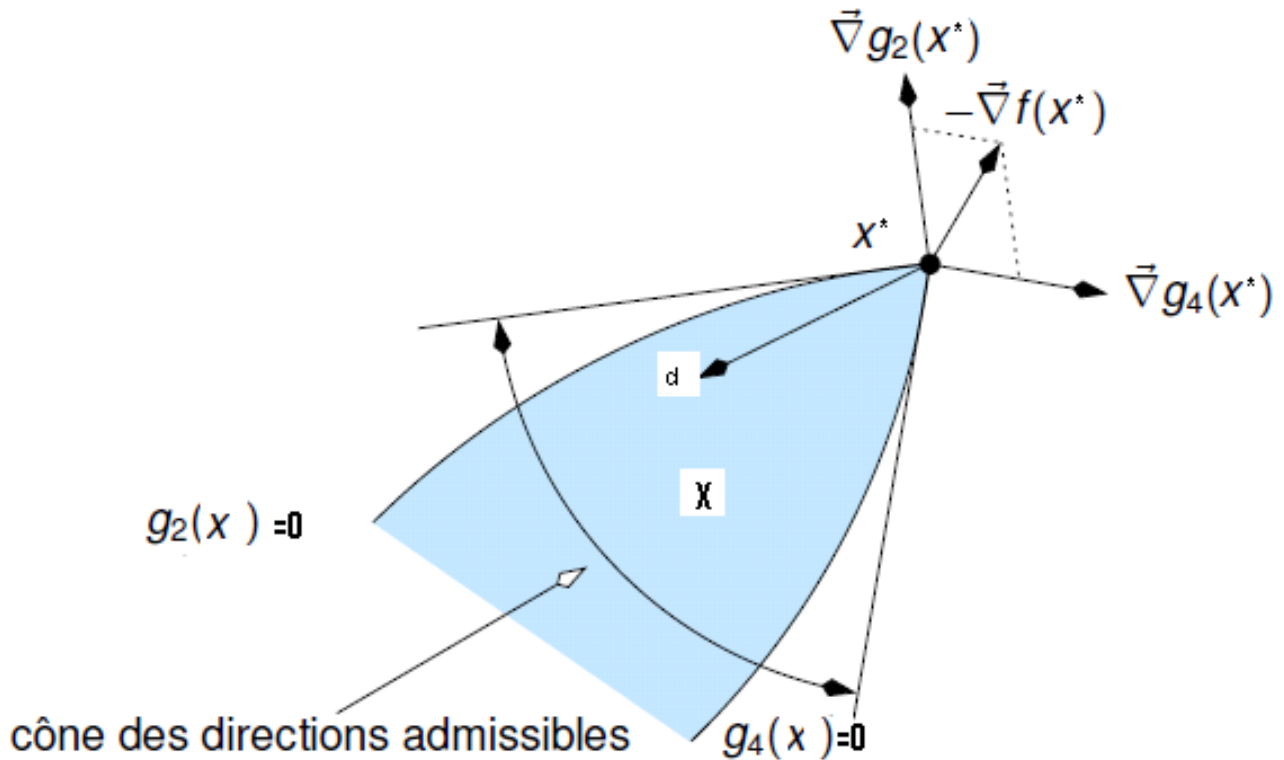


FIG. 1.6 – *Interprétation géométrique des conditions de Karush-Kuhn et Tucker*

Les conditions de Karush-Kuhn et Tucker sur un exemple à deux dimensions. Au point x^* les contraintes saturées sont les contraintes g_2 et g_4 et $\mathbb{I}_a = \{2,4\}$.

L'ensemble des directions admissibles d forme un cône intersection des $|\mathbb{I}_a|$ demi-espaces d'équation :

$$\nabla g_i^T(x^*) \cdot d \leq 0 \quad (\forall i \in \mathbb{I}_a)$$

Pour que x^* soit un optimum local il faut que le vecteur $-\nabla f(x^*)$ fasse un angle obtus avec chaque direction admissible d . Il est alors équivalent de dire que le vecteur $-\nabla f(x^*)$ doit faire un angle aigu avec le gradient $\nabla g_i(x^*)$ de chaque contrainte saturée ($i \in \mathbb{I}_a$).

Dans ces conditions, $-\nabla f(x^*)$ doit effectivement s'exprimer comme combinaison linéaire à coefficients λ_i positifs des $\nabla g_i(x^*)$ pour ($i \in \mathbb{I}_a$).

Remarque 1.4.2.

1. Les conditions de Karush-Kuhn et Tucker s'étendent sans difficulté à des problèmes comportant à la fois des contraintes d'égalité et des contraintes d'inégalité de la forme

$$(P) \begin{cases} \min f(x) \\ \text{sous les contraintes} \\ g_i(x) \leq 0 & i = \{1, 2, \dots, m\} \\ h_i(x) = 0 & i = \{1, 2, \dots, p\} \\ x \in \mathbb{R}^n. \end{cases}$$

La condition nécessaire pour que x^* soit un optimum local de (P) est qu'il existe des nombres $\lambda_i \geq 0$ et μ_i non contraints en signe tels que :

$$\begin{cases} \nabla f(x^*) + \sum_{i \in \mathbb{I}} \lambda_i \nabla g_i(x^*) + \sum_{i \in \mathbb{II}} \mu_i \nabla h_i(x^*) = 0, \\ \text{et} \\ \lambda_i \nabla g_i(x^*) = 0 \quad (i \in \mathbb{I}) \end{cases} \quad (1.21)$$

1.4.3 Conditions suffisantes d'optimalité

Nous allons étudier maintenant des conditions suffisantes d'optimalité. Associons à chaque contrainte un nombre réel $\lambda_i \geq 0$ appelé multiplicateur de Lagrange. La fonction de Lagrange est :

$$L(x, \lambda) = f(x) + \sum_{i \in \mathbb{I}} \lambda_i g_i(x) \quad (1.22)$$

Définition 1.4.4. "Point-col"

Soit : $\bar{x} \in X$ et $\bar{\lambda} \geq 0$.

On dit que $(\bar{x}, \bar{\lambda})$ est un point-col de $L(x, \lambda)$ si :

1. $L(\bar{x}, \bar{\lambda}) \leq L(x, \bar{\lambda}) \quad \forall x \in X$
2. $L(\bar{x}, \lambda) \leq L(\bar{x}, \bar{\lambda}) \quad \forall \lambda \geq 0$

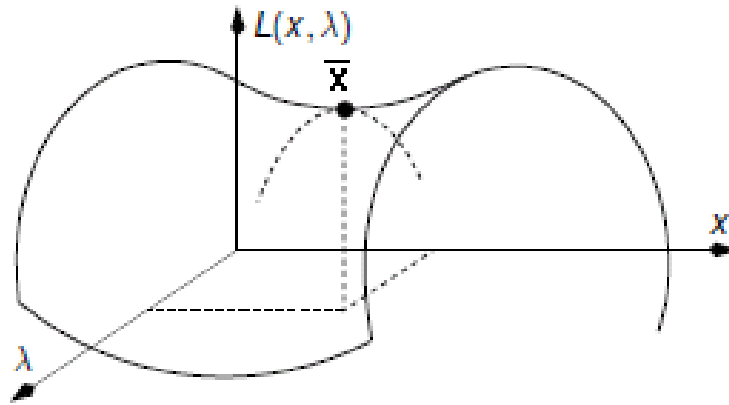


FIG. 1.7 – Illustration de la notion de point-col

Propriété 1.1. "Caractérisation d'un Point-col"

Soit $\bar{x} \in X$ et $\bar{\lambda} \geq 0$.

$(\bar{x}, \bar{\lambda})$ est un point-col pour $L(x, \lambda)$ si et seulement si :

- a) $L(\bar{x}, \bar{\lambda}) = \min_{x \in X} L(x, \lambda)$
- b) $g_i(\bar{x}) \leq 0 \quad \forall i \in \mathbb{I}$
- c) $\bar{\lambda}_i g_i(\bar{x}) \leq 0 \quad \forall i \in \mathbb{I}$

Théorème 1.4.2. [3] "Suffisance de la condition de point-col"

Si $(\bar{x}, \bar{\lambda})$ est un point-col de $L(x, \lambda)$ alors \bar{x} est un optimum global du problème (1.20).

Preuve.

La condition (a) du théorème précédent entraîne :

$$f(\bar{x}) + \sum_{\mathbb{I}} \bar{\lambda}_i g_i(\bar{x}) \leq f(x) + \sum_{\mathbb{I}} \bar{\lambda}_i g_i(x) \quad (\forall x \in X)$$

D'autre part :

$$(c) \implies \bar{\lambda}_i g_i(\bar{x}) = 0 \quad (\forall i)$$

d'où :

$$f(\bar{x}) \leq f(x) + \sum_{\mathbb{I}} \bar{\lambda}_i g_i(x) \quad \forall x \in X$$

et comme $\bar{\lambda} \geq 0$ on a : $f(\bar{x}) \leq f(x)$, $\forall x \in X$ tel que : $g_i(x) \leq 0$.

D'où \bar{x} est un optimum global du problème.

Remarque 1.4.3.

Ce résultat est très général et s'applique à n'importe quel programme mathématique convexe, non convexe, f et g_i différentiables ou non...

Cependant, pour certains problèmes, il peut ne pas exister de point-col. C'est le cas en général, pour les problèmes non convexes.

Conclusion

Nous avons vu que les conditions nécessaires d'optimalité permettent d'identifier les points critiques, qui sont les candidats pour la solution d'un problème d'optimisation. Une manière d'aborder le problème serait de résoudre le système d'équations défini par ces conditions.

Nous allons voir maintenant des méthodes numériques permettant de résoudre de tels systèmes d'équations non linéaires. Bien que celles-ci ne soient pas utilisées directement pour l'optimisation, elles servent de base aux algorithmes principaux.

Chapitre 2

Méthodes itératives d'optimisation sans contraintes

Introduction

L'optimisation sans contrainte de fonctions continues non linéaires a pour objectif, de trouver pour une fonction donnée f de \mathbb{R}^n dans \mathbb{R} continue, un vecteur x^* tel que $f(x^*)$ soit un extremum de f . Usuellement, f est appelée fonction objectif (ou critère, ou encore coût) et x variable de décision. Étant donné que la maximisation de f est équivalente à la minimisation de $-f$, tous les algorithmes présentés ici recherchent un minimiseur de la fonction objectif. Autrement dit, on cherche x^* tel que :

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) \quad (2.1)$$

On considère le problème d'optimisation sans contrainte suivant :

$$(P) \begin{cases} \text{Min } f(x) \\ x \in \mathbb{R}^n \end{cases}$$

Pour résoudre le problème (P) intéressons nous aux méthodes suivantes:

1. *Les méthodes basées sur le gradient*
2. *Les méthodes de Newton*
3. *La méthode des directions conjuguées*
4. *La méthode du simplexe (ou méthode de Nelder et Mead)*

Ces méthodes utilisent des dérivées (et donc la propriété de différentiabilité de f) à l'exception des méthodes de directions conjuguées (sauf dans le cas particulier de la méthode du gradient conjugué) et de la méthode du simplexe basée, elle, sur des propriétés plus géométriques.

Après les avoir décrites, nous discuterons plus précisément des champs d'application de chacune de ces méthodes et les analyserons du point de vue de leur convergence. Après avoir implémenté les méthodes grâce au programmes développé au cours de ce projet, nous serons en mesure d'établir une étude comparative des différentes méthodes, du point de vue de la performance et de la rapidité de convergence des algorithmes, et cela en fonction du type de problème et du point initial.

Nous allons maintenant rappeler brièvement les définitions de quelques notions importantes pour la suite.

Notion de direction de descente

Le gradient joue un rôle essentiel en optimisation. Dans le cadre des méthodes d'optimisation, il sera également important d'analyser le comportement de la fonction objectif dans certaines directions. Commençons pour cela par rappeler le concept de dérivée directionnelle.

Définition 2.1. "dérivée directionnelle"

Soit $f : \mathbb{R}^n \mapsto \mathbb{R}$ une fonction continue, soit $x \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$.

La dérivée directionnelle de f en x dans la direction d est définie par :

$$df(x,d) = \lim_{t \rightarrow 0^+} \frac{f(x + td) - f(x)}{t} \quad (2.2)$$

si cette limite existe.

Proposition 2.1.

Si f est différentiable en un point $x \in \mathbb{R}^n$, alors pour tout $d \neq 0$, f admet une dérivée dans la direction d en x et :

$$df(x,d) = d^T \nabla f(x) \quad (2.3)$$

On rappelle que la réciproque est fautive! La dérivabilité selon tout vecteur en x n'implique pas nécessairement la différentiabilité de f en x .

La dérivée directionnelle donne des informations sur la pente de la fonction dans la direction d , tout comme la dérivée donne des informations sur la pente des fonctions à une variable. En particulier,

- si $df(x,d) > 0$ alors f est croissante dans la direction d .
- si $df(x,d) < 0$ alors f est décroissante dans la direction d .

Dans ce dernier cas, on dira que d est une direction de descente de f .

Définition 2.2. "direction de descente"

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

$$\forall t \in]0, \eta], \quad f(x + td) < f(x) \quad (2.4)$$

Théorème 2.1. [9]"Direction de plus forte descente"

Soit $f: \mathbb{R}^n \mapsto \mathbb{R}$ une fonction différentiable, et x un point de \mathbb{R}^n .

Alors pour toute direction d de norme constante égale à $\|d\| = \|\nabla f(x)\|$, on a :

$$(-\nabla f(x))^T \nabla f(x) \leq d^T \nabla f(x)$$

La direction $d^ = -\nabla f(x)$ est appelée direction de plus forte descente.*

La direction opposée au gradient est donc celle dans laquelle la fonction a la plus forte descente, le gradient correspond, lui, à la plus forte pente.

Corollaire 2.1. [9]"Direction de plus forte pente"

Le vecteur $\nabla f(x)$ est appelée direction de plus forte pente de f au point $x \in \mathbb{R}^n$.

On remarquera que si x^* est un point de minimum local de f , alors il n'existe aucune direction de descente pour f au point x^* .

Proposition 2.2.

Soient $f: \mathbb{R}^n \mapsto \mathbb{R}$ différentiable et $x \in \mathbb{R}^n$ tel que : $\nabla f(x) \neq 0$.

Le vecteur $d \in \mathbb{R}^n$ est une direction de descente pour f à partir du point x ssi la dérivée directionnelle de f en x dans la direction d vérifie :

$$df(x,d) = \nabla f^T(x)d < 0 \quad (2.5)$$

Théorème 2.2. [9] " descente "

Soit $f: \mathbb{R}^n \mapsto \mathbb{R}$ une fonction différentiable. Soient $x \in \mathbb{R}^n$ tel que $f(x) \neq 0$ et $d \in \mathbb{R}^n$. Si d est une direction de descente en x alors il existe η tel que :

$$\forall \alpha \in [0, \eta[, \quad f(x + \alpha d) < f(x) \quad (2.6)$$

En utilisant ce résultat, on construit un algorithme général de minimisation, nommée algorithme de descente. Il consiste simplement à suivre une direction de descente de façon itérative jusqu'à l'obtention d'un "bon" minimiseur.

2.1 Dédution d'une classe d'algorithmes de descente

Dans cette section, nous construisons une famille d'algorithmes itératifs pour calculer un minimum local d'une fonction f . La base de notre étude est la suivante : si un point x ne satisfait pas aux conditions d'optimalité, il est possible de produire un autre point, x^1 pour lequel $f(x^1) < f(x^0)$. Cette famille d'algorithmes porte le nom de méthodes de descente. On cherche alors x^1 sous la forme $x^1 = x^0 + \alpha_1 d^1$ où d^1 est un vecteur non nul de \mathbb{R}^n et α_1 un réel strictement positif. En pratique donc, on cherche d^1 et α^1 pour que $f(x^0 + \alpha_1 d^1) < f(x^0)$. On ne peut pas toujours trouver d^1 .

Quand d^1 existe on dit que c'est une direction de descente et α_1 est le pas de descente. La direction et le pas de descente peuvent être fixes ou changer à chaque itération. Le schéma général d'une méthode de descente est le suivant :

$$\begin{cases} x^0 \in \mathbb{R}^n \\ x^{k+1} = x^k + \alpha_k d^k, \quad d^k \in \mathbb{R}^n - \{0\}, \quad \alpha_k \in \mathbb{R}^{+*} \end{cases}$$

où α_k et d^k sont choisis de telle sorte que $f(x^0 + \alpha_1 d^1) < f(x^0)$.

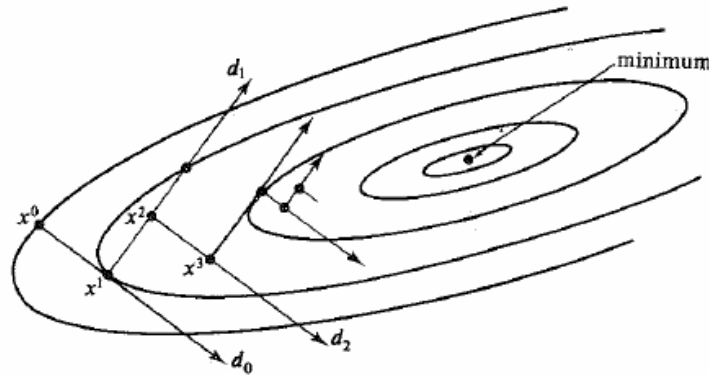


FIG. 2.1 – Schéma général d'une méthode de descente

Algorithme de descente**1. Initialisation**

$k \leftarrow 0$: On se donne une fonction f différentiable, un point initial x^0 et un seuil de tolérance $\varepsilon > 0$

2. Itération $k = 1, 2, \dots$

$$x^{k+1} \leftarrow x^k + \alpha_k d^k$$

3. Critère d'arrêt

Si $\|x^{k+1} - x^k\| < \varepsilon$ **STOP** x^k est la solution optimale

Sinon, on pose $k \leftarrow k + 1$ et on retourne à 2.

Il est important de noter que l'algorithme n'est pas complètement défini. En effet, rien n'est spécifié le pas α , et la direction d . Enfin, certaines hypothèses supplémentaires sont nécessaires afin de garantir que la méthode converge.

2.1.1 La recherche linéaire d'un pas

Soit x un point de \mathbb{R}^n tel que: $\nabla f(x) \neq 0$ et d une direction de descente de f en x . Intéressons nous maintenant d'un peu plus près à la phase de recherche linéaire i.e. au calcul d'un pas $\alpha \in \mathbb{R}$ vérifiant :

$$f(x + \alpha d) < f(x).$$

D'après la caractérisation de la descente (proposition 2.2), il s'agit donc à chaque itération k , de trouver un point x^{k+1} dans une direction d vérifiant :

$$\varphi'(0) = \nabla f(x)^T d < 0$$

Recherche linéaire d'Armijo et de Wolfe

La condition de descente: $f(x^k + \alpha d^k) < f(x^k)$ n'est pas suffisante pour que le pas α soit considéré comme acceptable car la convergence n'est pas toujours vérifiée. la notion de direction de descente est une notion locale qui ne s'applique qu'au voisinage de l'itéré courant.

De ce fait, au lieu de demander à ce que le pas cherché minimise la $f(x + \alpha d)$, on préfère imposer des conditions moins restrictives (et moins coûteuses à mettre en oeuvre): une première condition est dû à *Armijo* (1966)^[10], et la deuxième est dû à *Wolfe*(1969)^[11].

Définition 2.1.1 (12). *"Pas d'Armijo et Wolfe"*

Un pas α est dit admissible pour une direction suffisamment descendante d lorsqu'il satisfait aux deux inégalités suivantes, nommées critère d'Armijo et de Wolfe respectivement :

$$f(x + \alpha d) - f(x) \leq \varepsilon_0 \alpha \nabla f(x)^T d \quad \varepsilon_0 \in]0, \frac{1}{2}[, \quad (\textit{Armijo}) \quad (2.7)$$

$$\nabla f(x + \alpha d)d \geq \varepsilon_1 \nabla f(x)^T d \quad \varepsilon_1 \in]\varepsilon_0, 1[, \quad (\textit{Wolfe}) \quad (2.8)$$

où ε_0 et ε_1 sont des paramètres fixés pour déterminer l'admissibilité d'un pas.

La condition d'Armijo permet d'éviter les pas trop grands. En pratique la constante ε_0 est choisie très petite de manière à satisfaire (2.7) le plus facilement possible.

En plus de la condition *d'Armijo*, il est donc important d'imposer une condition qui permet d'éviter les pas trop petits. Cette condition est celle de Wolfe donnée par (2.8). Pour le choix de la direction d_k peut être:

- Soit le gradient de f en x^k : $d_k = -\nabla f(x^k)$;
- Soit calculée à partir du gradient $\nabla f(x^k)$;
- Soit choisie d'une façon plus ou moins arbitraire à condition que ce soit une direction de descente c'est-à-dire que $d_k^T \nabla f(x^k) < 0$.

Pour notre contexte intéressons nous à la méthode du gradient à pas optimal

2.1.2 Méthode du gradient à pas optimal

La méthode de plus profonde descente est la méthode de gradient la plus utilisée. Une idée naturelle consiste à suivre la direction de plus forte descente et à faire un pas qui rends la fonction à minimiser la plus petite possible dans cette direction. La première question

est de choisir cette direction. On remplace f au voisinage de x^k par son développement de Taylor de premier ordre:

$$f(x^k + d) \sim f(x^k) + \nabla f(x^k)^T d$$

On voudrait que la dérivée directionnelle $\nabla f(x^k)^T d$ soit la plus petite possible dans un voisinage de d . On cherche donc à résoudre :

$$\min_{d \in \mathbb{R}^n} \nabla f(x^k)^T d \quad s.c \quad \|d\| = 1$$

dont la solution nous ait donnée par le théorème de la direction de plus forte descente, comme la direction de plus profonde descente normalisée :

$$d^k = -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$$

On pose ensuite $q(\alpha) = f(x^k - \nabla f(x^k)\alpha)$ et on calcule α_k de façon à minimiser q pour $\alpha \geq 0$. On est alors ramené à un problème d'optimisation unidimensionnelle.

Algorithme de plus profonde descente ("Steepest descent")

1. *Initialisation*

$k \leftarrow 0$: On se donne une fonction f différentiable, un point initial x^0 et un seuil de tolérance $\varepsilon > 0$

2. *Itération* $k = 1, 2, \dots$

pose $d^k = -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$

calculer: $\alpha_k = \arg \min_{\alpha > 0} f(x^k + \alpha d^k)$

$x^{k+1} \leftarrow x^k + \alpha_k d^k$

3. *Critère d'arrêt*

Si $\|x^{k+1} - x^k\| < \varepsilon$ **STOP**) x^{k+1} est la solution optimale

Sinon, on pose $k \leftarrow k + 1$ et on retourne à 2.

2.1.3 Convergence de la méthode du gradient

On veut minimiser une fonction f . Pour cela on se donne un point de départ arbitraire x^0 . Pour construire l'itéré suivant x^1 il faut penser qu'on veut se rapprocher du minimum de f , on veut donc que:

$$f(x^1) < f(x^0)$$

Notre but est maintenant de produire une suite de points x^k dont les points d'accumulation sont stationnaires, et ce quel que soit le point de départ x^0 de la suite. Nous utiliserons pour ce faire le fait que nous tentons de minimiser une fonction f , ce qui nous aide à évaluer la qualité relative de deux points consécutifs de la suite.

Théorème 2.1.1. [13]

Soit f une fonction C^1 de \mathbb{R}^n dans \mathbb{R} , x^* un minimum de f . Supposons que

i) f est α -elliptique, c'est-à-dire,

$$\exists \alpha > 0, \quad \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \quad (\nabla f(x) - \nabla f(y)) \cdot (x - y) \geq \alpha \|x - y\|^2.$$

ii) l'application ∇f est lipschitzienne

$$\exists \sigma > 0, \quad \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \quad (\|\nabla f(x) - \nabla f(y)\|) \leq \sigma \|x - y\|.$$

S'il existe deux réels a et b tels que $\rho^{(k)}$ satisfasse $0 < a < \rho^{(k)} < b < \frac{2\alpha}{\sigma^2}$, pour tout $k \geq 0$, alors, la méthode du gradient définie par $x^{k+1} = x^k + \alpha_k d^k$ converge pour tout choix de x^0 de façon géométrique

$$\exists \beta \in]0, 1[, \quad \|x^{k+1} - x^*\| \leq \beta^k \|x^0 - x^*\|.$$

Remarque 2.1.1.

Le gradient à pas optimal qui est en principe la meilleure méthode des méthodes du gradient d'un point de vue de la rapidité de convergence, celle-ci peut être lente car elle fait des zigzags pour aller vers la solution optimale.

2.2 La méthode de Newton

La méthode de Newton n'est pas une méthode d'optimisation à proprement parler. C'est en réalité une méthode utilisée pour résoudre des équations non linéaires de la forme $f(x) = 0$ où f est une fonction de \mathbb{R}^n dans \mathbb{R}^n . Nous allons d'abord la décrire puis montrer comment on peut l'appliquer à la recherche de minimum.

La méthode de Newton dans \mathbb{R}

Présentons d'abord formellement cette méthode dans \mathbb{R} . Soit \mathbb{I} un intervalle de \mathbb{R} et $f : \mathbb{I} \rightarrow \mathbb{R}$, pour résoudre $f(t) = 0$ où f est une fonction C^1 . La méthode de Newton

part d'une solution approchée x et remplace l'équation $f(t) = 0$ par l'équation approchée $f(x) + (t - x)f'(x) = 0$, d'où la solution

$$t = x - \frac{f(x)}{f'(x)} \quad (2.9)$$

Bien entendu, cette formule n'a un sens que si $f'(x) \neq 0$. On espère que le nombre réel t donné par cette équation est un peu plus proche d'une racine que ne l'était x . En tout cas, si t appartient encore à \mathbb{I} , on peut recommencer en partant de t , etc..., d'où une suite définie (pas forcément pour tout n) par la relation de récurrence :

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)} \quad (2.10)$$

Nous préciserons ensuite les conditions d'utilisation de la méthode et justifierons l'existence des itérés successifs dans un théorème général de convergence.

Algorithme de Newton dans \mathbb{R}

1. *Initialisation*

$k = 0$: On se donne une fonction f différentiable, un point initial x^0 et un seuil de tolérance $\varepsilon > 0$

2. *Itération $k=1,2,\dots$*

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)};$$

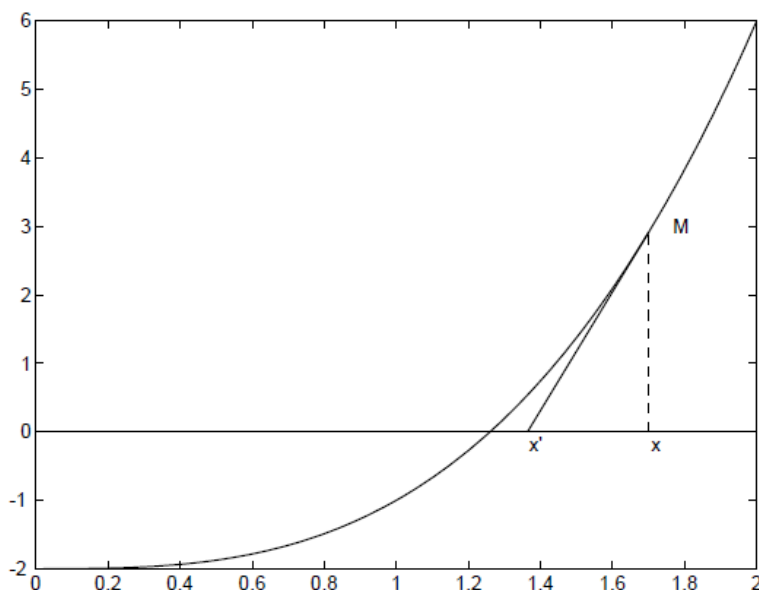
3. *Critère d'arrêt*

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP) x^{k+1} est la solution optimale

Si non, on pose $k = k + 1$ et on retourne à 2.

Remarque 2.2.1.

Remarquons qu'il faut non seulement assurer la convergence de la suite x^k vers la solution x^* , mais aussi montrer que cette suite est bien définie, c'est-à-dire montrer que $f'(x^k) \neq 0$ à l'étape 2. Cette méthode est aussi appelée *méthode de la tangente*. En effet chaque itéré x^{k+1} est obtenu à partir du précédent en traçant la tangente à la courbe de f au point $(x^k, f(x^k))$ et en prenant son intersection avec l'axe des abscisses (Fig 2.1).

FIG. 2.2 – *Interprétation géométrique de la méthode de Newton*

La méthode de Newton dans \mathbb{R}^n

Nous pouvons maintenant généraliser à \mathbb{R}^n . Soit f une fonction de classe \mathcal{C}^1 de \mathbb{R}^n à valeurs dans \mathbb{R}^n . On suppose que l'équation

$$f(x) = 0$$

On suppose ici que f est deux fois continûment dérivable et que l'on sait calculer ses dérivées secondes. Au voisinage d'un point x^k , on approche f par la fonction quadratique donnée par la formule de Taylor d'ordre 2 :

$$q(x) = f(x^k) + (x - x^k)^t \cdot \nabla f(x^k) + \frac{1}{2} (x - x^k)^t \cdot \nabla^2 f(x^k) \cdot (x - x^k) \quad (2.11)$$

On peut alors choisir pour x^k le point, s'il existe, qui minimise q ; pour que ce point minimisant q existe, il est suffisant que $\nabla^2 f(x^k)$ soit définie positive, il est alors déterminé par l'équation $\nabla q(x) = 0$ qui s'écrit :

$$\nabla f(x^k) + \nabla^2 f(x^k) \cdot (x - x^k) = 0$$

d'où :

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k) \quad (2.12)$$

On pose

$$d^k = -[\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k) \quad (2.13)$$

On a $\nabla^2 f(x^k)$ est définie positive, la direction d^k sera le minimum de (2,11).

Dans ce cas,

$$\nabla f(x^k) d^k = -(d^k)^t \nabla^2 f(x^k) d^k \leq 0 \quad (2.14)$$

on obtient bien une direction de descente.

Définition 2.2.1. "Direction de Newton"

Soit à minimiser la fonction f, \mathcal{C}^2 sur son domaine de définition, alors on désigne par direction de Newton en x le vecteur

$$d = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

Algorithme de Newton dans \mathbb{R}^n

1. Initialisation

$k = 0$: On se donne une fonction f différentiable, un point initial x^0 et un seuil de tolérance $\varepsilon > 0$

2. Itération $k = 1, 2, \dots$

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \cdot \nabla f(x^k);$$

3. Critère d'arrêt

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP) x^{k+1} est la solution optimale

Sinon, on pose $k = k + 1$ et on retourne à 2.

2.2.1 Convergence de la méthode de Newton

L'inconvénient majeur de la méthode est sa sensibilité au choix du point de départ: la convergence est *locale*. Si ce point est mal choisi ("trop loin" de la solution) la méthode peut diverger. Un autre inconvénient de la méthode de Newton réside également dans le fait que nous avons à évaluer et "à inverser" $\nabla^2 f(x^k)$ à chaque itération. Certaines méthodes proposent d'utiliser non pas $\nabla^2 f(x^k)$ comme matrice mais plutôt une approximation de celle-ci (plus précisément, dans les méthodes de Quasi-Newton)^[14]. L'avantage de cette méthode est sa grande rapidité. La convergence est quadratique, c'est-à-dire que l'erreur

$e_k = \|x^{k+1} - x^k\|$ est élevée au carré à chaque itération. Concrètement, si elle vaut 10^{-2} à l'étape k elle vaudra 10^{-4} à l'étape $k + 1$ et 10^{-8} à l'étape $k + 2$.

Nous avons le résultat de convergence suivant:

Proposition 2.2.1.

Si x^0 est choisi suffisamment proche d'un minimum local x^ où le hessien de f est défini positif, alors la suite (x^k) a une convergence quadratique vers x^* .*

Théorème 2.2.1. [2]

Soit f une fonction deux fois différentiable, possédant un unique minimum global x^ tel que $\nabla f(x^*)$ est définie positive (on note $\lambda > 0$ sa plus petite valeur propre) et tel que $\nabla^2 f(x)$ est localement Lipschitz au voisinage de x^* (on note C sa constante Lipschitz). L'algorithme de Newton converge quadratiquement vers x^* si on l'initialise en un point x^0 tel que*

$$\|x^0 - x^*\| \leq \frac{\lambda}{2C} \quad (2.15)$$

Preuve.

D'après la formule de Taylor $\exists \eta \in [x^*, x^k]$ tel que

$$\nabla f(x^*) - f(x^k) = \nabla^2 f(\eta)(x^* - x^k)$$

Ceci donne, en utilisant $\nabla f(x^*) = 0$

$$-f(x^k) = \nabla^2 f(\eta)(x^* - x^k)$$

$$-f(x^k) - \nabla^2 f(\eta)(x^* - x^k) = (\nabla^2 f(\eta) - \nabla^2 f(x^k))(x^* - x^k)$$

D'autre part, si x^k est suffisamment proche de x^* , la condition de Lipschitz donne qu'il existe $C > 0$ tel que

$$\|\nabla^2 f(\eta) - \nabla^2 f(x^k)\| \leq C\|x^* - x^k\|$$

On peut donc déduire

$$\|-\nabla f(x^k) - \nabla^2 f(x^k)(x^* - x^k)\| \leq C\|x^* - x^k\|^2 \quad (2.16)$$

L'itération de l'algorithme de Newton donne

$$x^k = x^{k+1} + \nabla^2 f(x^k)^{-1} \nabla f(x^k)$$

Après substitution et simplification dans (2.17) on a

$$\|\nabla^2 f(x^k)(x^* - x^{k+1})\| \leq C\|x^* - x^k\|^2$$

Il s'agit maintenant de faire apparaître $\nabla^2 f(x^*)$ qui est la quantité sur laquelle porte l'hypothèse. Une inégalité triangulaire donne

$$\|\nabla^2 f(x^*)(x^* - x^{k+1})\| \leq \|\nabla^2 f(x^k)(x^* - x^{k+1})\| + \|(\nabla^2 f(x^*) - \nabla^2 f(x^k))(x^* - x^{k+1})\|$$

À l'aide des majorations précédentes on a alors

$$\|\nabla^2 f(x^*)(x^* - x^{k+1})\| \leq C\|x^* - x^k\|^2 + C\|x^* - x^k\|\|x^{k+1} - x^*\| \quad (2.17)$$

Notons maintenant $\lambda > 0$ la plus petite valeur propre de $\nabla^2 f(x^*)$. L'inéquation (1.18) donne alors

$$\lambda\|(x^* - x^{k+1})\| \leq C\|x^* - x^k\|^2 + C\|x^* - x^k\|\|x^{k+1} - x^*\| \quad (2.18)$$

On a

$$\|x^k - x^*\| \leq \frac{\lambda}{2C}$$

Pour tous les indices $p > k$ suivants on encore $\|x^p - x^*\| \leq \frac{\lambda}{2C}$.

En effet on peut tirer de (1.19) que

$$(\lambda - C\|x^k - x^*\|)\|x^{k+1} - x^*\| \leq C\|x^k - x^*\|^2$$

d'où

$$\|x^k - x^*\| \leq \frac{\lambda}{2C}$$

2.3 La méthode du gradient conjugué

Il s'agit d'une variante de la méthode de plus profonde descente, introduite par *Hestenes* et *Stiefel* en 1952^[15].

L'algorithme que nous présentons dans cette section possède deux intérêts. Il permet de résoudre des systèmes linéaires à coefficients strictement positifs de grande taille et il sert d'algorithme de base pour la résolution des problèmes d'optimisation non linéaire.

Soit le problème de minimisation quadratique suivant:

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c \quad \text{avec } x \in \mathbb{R}^n \quad (2.19)$$

où A est une matrice carrée d'ordre n , symétrique, définie positive, et $b \in \mathbb{R}^n$.

On note

$$\nabla f(x) = Ax + b \quad (2.20)$$

qui est non nul, et

$$\nabla^2 f(x) = A \quad (2.21)$$

La première idée fondamentale de l'algorithme du gradient conjugué consiste à choisir chaque direction de descente conjuguée à la direction de descente précédente par rapport à A. Afin de développer le gradient conjugué, introduisons la notion de direction conjuguée.

Définition 2.3.1. "directions conjuguées"

Soit une matrice $A \in \mathbb{R}^n \times \mathbb{R}^n$ définie positive. Les vecteurs (ou directions) non nuls de \mathbb{R}^n d^1, \dots, d^k sont conjugués par rapport à A (A-conjugués) si

$$(d^i)^T A d^j = 0, \quad \forall i, j \text{ tels que } i \neq j \quad (2.22)$$

Ceci signifie que ces deux vecteurs sont orthogonaux pour le produit scalaire associé à la matrice A, défini par

$$\langle x, y \rangle_A = x^T A y, \quad \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \quad (2.23)$$

Théorème 2.3.1. [16]

Soit $\{d^1, \dots, d^k\}$ un ensemble de directions non nulles et conjugués par rapport à A. Alors les vecteurs d^1, \dots, d^k sont linéairement indépendants.

Preuve.

Soit $\{d^1, \dots, d^k\}$ est une famille de k directions A-conjuguées, et $\lambda_i \in \mathbb{R}$ avec $(i = 1, \dots, k)$.

Montrer que

$$\sum_{i=1}^{i=k} \lambda_i d^i = 0 \quad \implies \quad \lambda_i = 0$$

On multiplie dans les deux membres par $A d^j$ avec $j = 1, \dots, k$

$$\sum_{i=1}^{i=k} \lambda_i d^i A d^j = \lambda_1 (d^1)^T A d^j + \lambda_2 (d^2)^T A d^j + \dots + \lambda_k (d^k)^T A d^j = 0 \quad \text{avec } j = 1, \dots, k$$

Comme d^1, \dots, d^k sont des directions A-conjuguées alors

$$(d^1)^T A d^j = (d^2)^T A d^j = \dots = (d^i)^T A d^j = 0$$

D'où

$$\lambda_j = 0 \quad \text{avec } j = 1, \dots, k$$

Donc

d^1, \dots, d^k sont linéairement indépendants.

Remarque 2.3.1.

Noter que, si A est la matrice identité, les directions conjuguées sont orthogonales.

En dimension $n > 2$, il existe une infinité de directions conjuguées à une direction donnée. Pour trouver à l'itération k , une direction d^k conjuguée à d^{k-1} , l'idée consiste à chercher d^k sous forme d'une combinaison linéaire de d^{k-1} et de la direction de la plus forte pente au point x^k , soit :

$$x^{k+1} = x^k + \alpha_k d^k \quad k = 1, \dots, n \quad (2.24)$$

avec

$$\alpha_k = \arg \min_{\alpha} f(x^k + \alpha d^k)$$

Lemme 2.3.1. [16]

Soient d^1, d^2, \dots, d^n un ensemble de directions A -conjuguées dans \mathbb{R}^n . Soient x^1, x^2, \dots, x^{n+1} les itérés générés par une méthode de directions conjuguées. Alors,

1. pour tout $k=1, \dots, n$, le pas α_k est défini par

$$\alpha_k = -\frac{(d^k)^T \nabla f(x^k)}{(d^k)^T A d^k} \quad (2.25)$$

2. pour tout $k = 1, \dots, n$, $\nabla f(x^k)$ est orthogonal à d^1, \dots, d^{k-1} , c-à-d

$$\nabla f(x^k)^T d^i = 0, \quad i = 1, \dots, k-1; \quad (2.26)$$

3. $\nabla f(x^{n+1}) = 0$;

4. soit k tel que $\nabla f(x^k) = 0$; alors,

$$\nabla f(x^i) = 0, \quad i = k, \dots, n+1. \quad (2.27)$$

Preuve.

1. Comme α_k est le minimum de la fonction dans la direction d^k , sa valeur est calculée en annulant la dérivée directionnelle de f dans la direction d^k , c'est-à-dire

$$(d^k)^T \nabla f(x^k + \alpha_k d^k) = (d^k)^T \nabla f(x^{k+1}) = 0 \quad (2.28)$$

et, en utilisant (2.20),

$$(d^k)^T (A(x^k + \alpha_k d^k) + b) = 0$$

$$(d^k)^T Ax^k + \alpha_k (d^k)^T Ad^k + (d^k)^T b = 0$$

D'où

$$\alpha_k = -\frac{b^T d^k + (x^k)^T Ad^k}{(d^k)^T Ad^k}$$

2. Comme $x^{k+1} = x^k + \alpha_k d^k$, nous avons

$$\begin{aligned} x^k &= x^{k-1} + \alpha_{k-1} d^{k-1} \\ &= x^{k-2} + \alpha_{k-2} d^{k-2} + \alpha_{k-1} d^{k-1} \\ &\quad \vdots \\ &= x^{i+1} + \sum_{j=i+1}^{k-1} \alpha_j d^j \end{aligned} \tag{2.29}$$

Dès lors, pour $i = 1, \dots, k-1$,

d'après (2.20)

$$(d^i)^T \nabla f(x^k) = (d^i)^T (Ax^k + b)$$

d'après (2.30)

$$\begin{aligned} &= (d^i)^T \left(A(x^{i+1} + \sum_{j=i+1}^{k-1} \alpha_j d^j) + b \right) \\ &= (d^i)^T Ax^{i+1} - (d^i)^T b + \sum_{j=i+1}^{k-1} \alpha_j (d^i)^T Ad^j \end{aligned}$$

d'après (2.22)

$$= (d^i)^T (Ax^{i+1} + b)$$

d'après (2.20)

$$= \nabla f(x^{i+1})^T d^i$$

d'après (2.29)

$$\nabla f(x^{i+1})^T d^i = 0$$

3. Soit $d \neq 0$ un vecteur arbitraire de \mathbb{R}^n . Comme d^1, \dots, d^n est un ensemble de n vecteurs linéairement indépendants dans \mathbb{R}^n , il s'agit d'une base et d peut s'écrire

$$d = \sum_{i=1}^n \lambda_i d^i$$

Dès lors

$$\nabla f(x^{n+1})^T d = \sum_{i=1}^n \lambda_i \nabla f(x^{n+1})^T d^i = 0$$

D'après (2) comme d est arbitraire, nous obtenons $\nabla f(x^{n+1}) = 0$.

4. Si $\nabla f(x^k) = 0$ alors $\alpha_k = 0$, d'après (2.26). Le résultat découle d'une simple récurrence sur k .

qui permet de transformer le problème (2.37) en problème sans contrainte

Calcul des directions conjuguées

On peut construire une base de vecteurs propres orthogonaux car A est symétrique. Ces vecteurs sont par construction A -conjugués et constituent un choix possible de directions conjuguées. Mais le calculs de ces vecteurs est en général très coûteux en temps de calculs. En fait on peut calculer les directions conjuguées au fur et à mesure qu'on en a besoin par la récurrence suivante

$$d^k = -\nabla f(x^k) + \beta^k d^{k-1}$$

qui s'interprète comme l'opposé du gradient au nouveau point altéré par la direction précédente. Pour que d^k et d^{k-1} soient des directions A -conjuguées on choisit

$$\beta^k = \frac{(\nabla f(x^k))^T \nabla f(x^k)}{(\nabla f(x^{k-1}))^T \nabla f(x^{k-1})} \quad (2.30)$$

La direction d^k est également A -conjuguée avec les directions d^i pour $i = 0, \dots, k-2$. Pour démontrer ce résultat on utilise la proposition suivante

Proposition 2.3.1. [13]

On pose $r^k = \nabla f(x^k)$.

Si $x^k = x^*$ alors les propriétés suivantes sont vérifiées

1. pour tout $i = 0, \dots, k-1$

$$(r^k)^T r^i = 0$$

2. Soient les deux vecteurs (r^0, r^1, \dots, r^k) et $(r^0, Ar^0, \dots, A^k r^0)$.

$$(r^0, r^1, \dots, r^k) = (r^0, Ar^0, \dots, A^k r^0)$$

ce dernier sous espace est dénommé sous-espace de Krylov.

3. Soient les deux vecteurs (d^0, d^1, \dots, d^k) et $(r^0, Ar^0, \dots, A^k r^0)$.

$$\text{vect}(d^0, d^1, \dots, d^k) = \text{vect}(r^0, Ar^0, \dots, A^k r^0).$$

4. pour tout $i = 0, \dots, k-1$

$$(d^k)^T A d^i = 0$$

On peut calculer

$$(d^i)^T A d^k = -(d^i)^T A r^k + \beta^k (d^i)^T A d^{k-1} = -(d^i)^T A r^k$$

On sait d'après que $(r^k)^T d^i = 0$ pour tout $i = 0, \dots, k-2$. Or d'après les points 2 et 3 de la proposition (1.3.1) on sait que

$$A d^i \in A, \quad (r^0, Ar^0, \dots, A^i r^0) = (Ar^0, A^2 r^0, \dots, A^{i+1} r^0) \subset \text{vect}(d^0, d^1, \dots, d^{i+1})$$

Donc il existe un vecteur $(\gamma^0, \gamma^1, \dots, \gamma^{i+1}) \in \mathbb{R}^{i+2}$ tel que

$$(d^i)^T A r^k = \sum_{j=0}^{i+1} \gamma^j (d^j)^T r^k = 0$$

quel que soit $i = 0, \dots, k-2$. En conclusion, les directions $\{d^0, d^1, \dots, d^k\}$ sont A-conjuguées.

Algorithme du gradient conjugué**1. Initialisation**

fixer $\varepsilon > 0$, choisir $x^0 \in \mathbb{R}^n$ poser $\nabla f(x^0) = Ax^0 - b$
 et $d^0 = -\nabla f(x^0)$

2. Itération $k = 0, 1, \dots$

calculer $\alpha_k = -\frac{(d^k)^T \nabla f(x^k)}{(d^k)^T A d^k}$

calculer $x^{k+1} = x^k + \alpha_k d^k$

calculer $\nabla f(x^{k+1}) = \nabla f(x^k) + \alpha_k A d^k$

calculer $\beta^{k+1} = \frac{\|\nabla f(x^{k+1})\|^2}{\|\nabla f(x^k)\|^2}$

calculer $d^{k+1} = -\nabla f(x^{k+1}) + \beta^{k+1} d^k$

3. Critère d'arrêt

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP) x^{k+1} est la solution optimale

Sinon, on pose $k = k + 1$ et on retourne à 2.

2.3.1 Convergence de la méthode du gradient conjugué

Concernant la convergence de la méthode du gradient conjugué, on a le résultat suivant:

Théorème 2.3.2. [18]

Soit A une matrice symétrique définie positive d'ordre n . Toute méthode qui utilise des directions conjuguées conduit à la solution exacte en au plus n itérations.

Preuve.

Les directions d^0, d^1, \dots, d^{n-1} forment une base A-orthogonale de \mathbb{R}^n . De plus, puisque x^k est optimal par rapport à toutes les directions d^j , ($j = 0, \dots, k-1$), le vecteur r^k est orthogonal à l'espace $V^{k-1} = \text{vect}(d^0, d^1, \dots, d^{k-1})$.

Par conséquent,

$$r^n \perp V^{n-1} = \mathbb{R}^n$$

et donc $r^n = 0$ ce qui implique

$$x^n = x$$

Remarque 2.3.2.

Rappelons que l'algorithme présenté ici était particularisé au cas d'une fonctionnelle quadratique. On peut étendre l'algorithme pour une fonctionnelle f différentiable quelconque de manière efficace.

2.3.2 La méthode du gradient conjugué pour une fonction quelconque

La méthode de Fletcher et Reeves (1964)¹⁹ et la méthode de Polak-Ribière (1971)²⁰ sont des extensions directes de la méthode précédente au cas des fonctions quelconques. Ces méthodes a deux avantages :

1. Elle nécessite le stockage de très peu d'informations.
2. Sa vitesse de convergence est très supérieure à celle des algorithmes de gradient classiques.

Ces méthodes consiste à remplacer $r(x^k)$ par $\nabla f(x^k)$, déterminer α_k par une minimisation en dimension un, et calculer β^{k+1} de la manière suivante:

$$\beta^{k+1} = \frac{\nabla f(x^{k+1})^T (\nabla f(x^{k+1})^T - \nabla f(x^k))}{\nabla f(x^k)^T \nabla f(x^k)} \quad (\text{Fletcher-Reeves})$$

$$\beta^{k+1} = \frac{\langle \nabla f(x^{k+1}), \nabla f(x^{k+1}) - \nabla f(x^k) \rangle}{\nabla f(x^k)^T \nabla f(x^k)} \quad (\text{Polak-Ribière})$$

L'algorithme s'écrit alors :

Algorithme de Fletcher-Reeves 1964 / Polak-Ribière (1971)

1. **Initialisation**

fixer $\varepsilon > 0$, choisir $x^0 \in \mathbb{R}^n$ poser $\nabla f(x^0) = Ax^0 - b$ et $d^0 = -\nabla f(x^0)$

2. **Itération** $k = 0, 1, \dots, \dots$

$$\text{calculer } \alpha_k = -\frac{(d^k)^T \nabla f(x^k)}{(d^k)^T A d^k}$$

$$\text{calculer } x^{k+1} = x^k + \alpha_k d^k$$

$$\text{calculer } \nabla f(x^{k+1}) = \nabla f(x^k) + \alpha_k A d^k$$

calculer

$$\beta^{k+1} = -\nabla f(x^{k+1}) + \frac{\nabla f(x^{k+1})^T (\nabla f(x^{k+1})^T - \nabla f(x^k))}{\nabla f(x^k)^T \nabla f(x^k)} \quad (\text{Fletcher-Reeves})$$

$$\beta^{k+1} = -\nabla f(x^{k+1}) + \frac{\langle \nabla f(x^{k+1}), \nabla f(x^{k+1}) - \nabla f(x^k) \rangle}{\nabla f(x^k)^T \nabla f(x^k)} \quad (\text{Polak-Ribière})$$

$$\text{calculer } d^{k+1} = -\nabla f(x^{k+1}) + \beta^{k+1} d^k$$

3. **Critère d'arrêt**

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP) x^{k+1} est la solution optimale

Si non, on pose $k = k + 1$ et on retourne à 2.

Remarque 2.3.3.

Tout comme l'algorithme de gradient conjugué, l'implémentation de ces algorithmes ne requiert que peu de calculs et peu de mémoire quelque soit la taille du système. Pour assurer la décroissance entre deux itérations, on utilisera dans la recherche linéaire les conditions de Wolfe avec typiquement des paramètres $0 < \varepsilon_1 < \varepsilon_2 < 1/2$.

Il est important de remarquer que la convergence globale de la méthode de Fletcher et Reeves et la méthode de Polak-Ribière n'est assurée que si l'on procède à une réinitialisation

périodique. Par exemple, toutes les n itérations, on repartira du dernier point obtenu avec, comme direction de déplacement, le gradient en ce point. La convergence globale de cette procédure découle alors de la convergence globale des méthodes de plus forte descente.

2.4 La méthode de Nelder-Mead

Enfin, pour conclure, nous présentons la méthode du simplexe de *Nelder-Mead*(1965)^[21] applicable à des problèmes d'optimisation non linéaire sans contraintes (à ne pas confondre avec l'algorithme du simplexe pour l'optimisation linéaire). La méthode du simplexe appartient à la classe des méthodes de recherche directe, c'est une méthode d'optimisation locale. Cette méthode . Toutefois, elle présente deux avantages : d'une part, elle ne nécessite pratiquement aucune information sur la fonction objectif à optimiser (pas besoin de connaître son gradient/sa matrice hessienne ou d'être différentiable), mais simplement la possibilité de pouvoir l'évaluer en différents points ; d'autre part, elle présente assez souvent des performances satisfaisantes en pratique (surtout pour des problèmes impliquant un petit nombre de variables!).

Soit une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ de n variables à minimiser. Cette méthode utilise la notion de simplexe dans \mathbb{R}^n . Introduisons la définition du simplexe.

Définition 2.4.1. "Simplexe"

Un simplexe de dimension k est l'enveloppe convexe de $k + 1$ vecteurs x^1, x^2, \dots, x^{k+1} de \mathbb{R}^n , $k \leq n$, affinement indépendants, c'est-à-dire que les k vecteurs $x^1 - x^{k+1}, x^2 - x^{k+1}, \dots, x^k - x^{k+1}$ sont linéairement indépendants.

Par exemple, trois points non alignés dans \mathbb{R}^2 , ou quatre points non coplanaires dans \mathbb{R}^3 sont affinement indépendants et définissent des simplexes de dimension 2 et 3, respectivement.

L'idée de la méthode de *Nelder-Mead* est de définir un simplexe de dimension n dans \mathbb{R}^n à partir de $n+1$ vecteurs affinement indépendant. Nous supposons que ces points sont triés de manière que

$$f(x^1) \leq f(x^2) \leq \dots \leq f(x^{n+1})$$

Ensuite, le plus mauvais de ces points, c'est-à-dire x^{n+1} sera remplacé par un point meilleur. Afin de déterminer ce meilleur point, on calcule le centre de gravité du simplexe formé par

les autres points, c'est-à-dire

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x^i$$

Plaçons-nous à l'itération k , étant donné le simplexe actuel qu'on noté Δ^k , il s'agit de déterminer le simplexe suivant Δ^{k+1} ($\neq \Delta^k$).

On considère la direction

$$d = \bar{x} - x^{n+1}$$

Les points de test sont obtenus moyennant quatre opérations possibles, à savoir : réflexion, expansion, contraction et rétrécissement. Chaque opération est associée à un paramètre. Les coefficients associés aux opérations de réflexion, expansion, contraction et rétrécissement sont respectivement notés par ρ, χ, γ , et σ . Ces coefficients doivent remplir les conditions suivantes :

$$\rho > 0, \quad \chi > 1, \quad 0 < \gamma < 1, \quad \text{et} \quad 0 < \sigma < 1.$$

D'après *Kelley*^[22], La séquence standard associée aux coefficients $\{\rho, \chi, \gamma, \sigma\}$ est $\{1, 2, \frac{1}{2}, \frac{1}{2}\}$.

Dans chaque itération de l'algorithme de Nelder-Mead deux transformations sont possibles :

- un seul point du simplexe actuel sera modifié. Le nouveau point de test accepté par l'algorithme remplace le plus mauvais point du simplexe actuel (x^{n+1}), et prépare le nouveau simplexe pour l'itération suivante,
- ou bien un ensemble de n nouveaux points sera calculé et remplace les n actuels points $x^i (i = 2, \dots, n + 1)$ afin de former un nouveau simplexe pour l'itération suivante.

L'algorithme s'arrête lorsque la différence ($f(x^{n+1}) - f(x^1)$), qui présente la distance entre les valeurs de la fonction du meilleur et du plus mauvais point, soit inférieure à un certain seuil de précision ε .

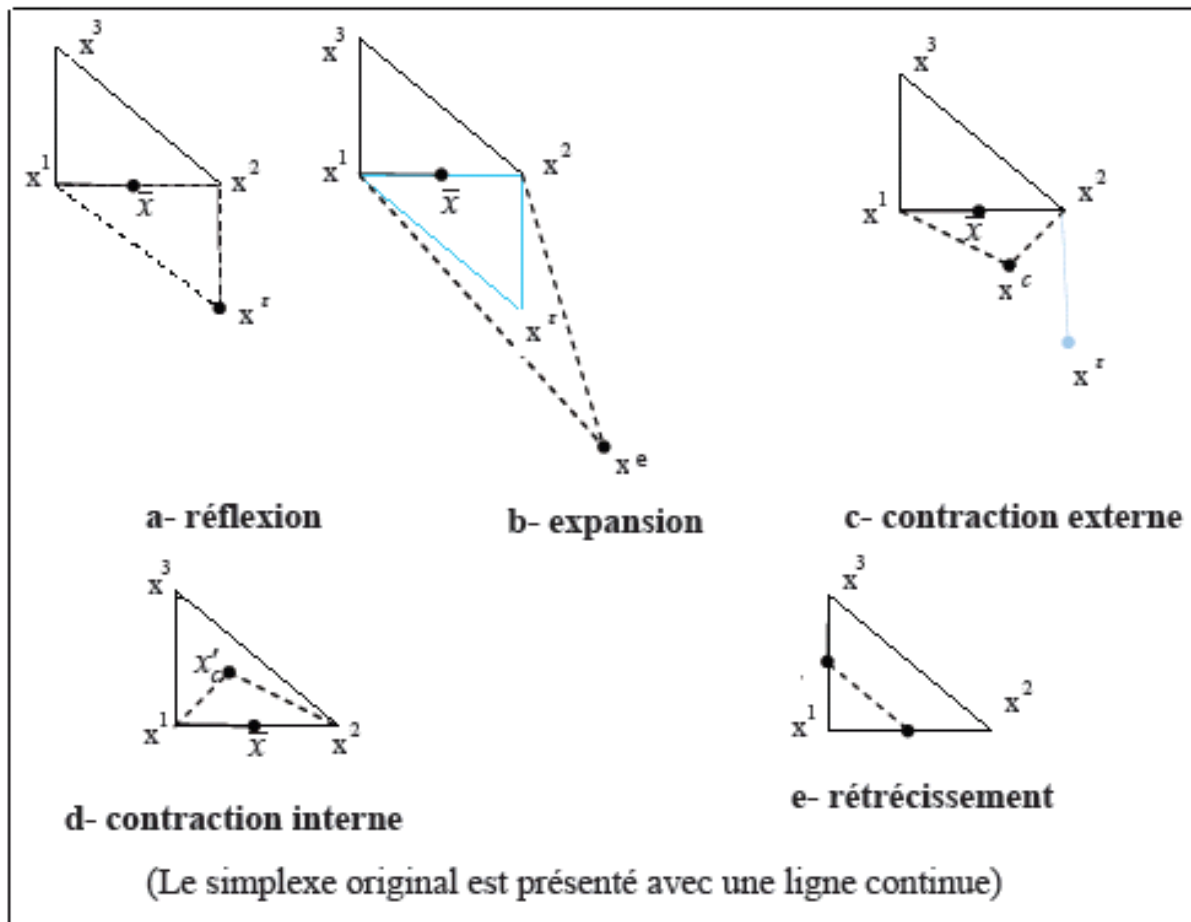


FIG. 2.3 – Les mouvements possibles dans la méthode de Nelder-Mead

La figure (2.2) montre l'effet des opérations de réflexion, de l'expansion, de la contraction (interne et externe) et du rétrécissement pour un simplexe à deux dimensions, soit un triangle (x^1, x^2, x^3) en utilisant la séquence standard des coefficients $\{\rho = 1, \chi = 2, \gamma = \frac{1}{2}, \sigma = \frac{1}{2}\}$.

Algorithme de Nelder et Mead**1. Initialisation**

Données: f une fonction et $n + 1$ points affinements indépendants x^1, x^2, \dots, x^{n+1} de \mathbb{R}^n .

Renommer les points tels que: $f(x^1) \leq f(x^2) \leq \dots \leq f(x^{n+1})$.

fixer $\varepsilon > 0$.

2. Itération: Tant que la différence entre $f(x^1)$ et $f(x^{n+1})$ est supérieure à un certain seuil de précision ε

2.1- calculer \bar{x} et $f_r = f(x(\rho))$

avec $\bar{x} = \frac{1}{n} \sum_{i=1}^n x^i$

et $x(\rho) = \bar{x} + \rho d = \bar{x} + \rho(\bar{x} - x^{n+1})$.

2.2- Réflexion :

si $f(x^1) \leq f_r < f(x^n)$,

alors $x^{n+1} = x(\rho)$, aller à 2.7.

2.3- Expansion :

si $f_r < f(x^1)$, alors calculer $f_e = f(x(\chi))$, avec $x(\chi) = \bar{x} + \chi(x^r - \bar{x})$

si $f_e < f_r$, alors $x^{n+1} = x(\chi)$ et aller à 2.7

sinon $x^{n+1} = x(\rho)$ et aller à 2.7.

2.4 Contraction externe :

si $f(x^n) \leq f_r < f(x^{n+1})$, alors calculer $f_0 = f(x(\gamma))$, avec $x(\gamma) = \bar{x} + \gamma(x^r - \bar{x})$

si $f_0 \leq f_r$, alors $x^{n+1} = x(\gamma)$ puis aller à 2.7

sinon aller à 2.6

2.5- contraction interne

si $f_r \geq f(x^{n+1})$, alors calculer $f_i = f(x(\gamma))$, avec $x(\gamma) = \bar{x} - \gamma(\bar{x} - x^{n+1})$

si $f_i < f(x^{n+1})$, alors $x^{n+1} = x(\gamma)$ puis aller à 2.7

sinon aller à 2.6.

2.6- Rétrécissement

pour $i = 2, \dots, n + 1$, calculer $x^i = \frac{(x^i + x^1)}{2}$, puis calculer $f(x^i)$

2.7- donner les sommets $(x^1, x^2, \dots, x^{n+1})$ tel que $f(x^1) \leq f(x^2) \leq \dots \leq f(x^{n+1})$.

retour à l'étape 2.

2.4.1 Convergence de la méthode du Nelder-Mead

Bien sûr, il n'existe pas de critère fiable pour déterminer quand on est proche d'un minimum, et on se contente généralement d'arrêter la méthode lorsque les sommets du

simplexe semblent se stabiliser ou converger.

Pour conclure, nous citons les rares résultats connus à propos de la convergence de cette méthode.

✓ Appliquée à une fonction strictement convexe à une variable, la méthode de *Nelder-Mead* converge vers un minimum (forcément global).

✓ Appliquée à une fonction strictement convexe à deux variables, la valeur de la fonction évaluée sur tous les sommets du simplexe converge vers la même valeur (mais ce n'est pas forcément la valeur optimale).

✓ Il existe une fonction strictement convexe à deux variables et une configuration de départ spécifique pour laquelle la méthode de *Nelder-Mead* converge vers un point qui n'est pas minimum.

2.5 Etude comparative des méthodes d'optimisation sans contraintes

Dans cette section, nous allons tâcher de résoudre des problèmes de manière réelle, et comparer les résultats obtenus pour les différentes méthodes.

Soit le problème quadratique suivant:

$$\text{minimiser } f(x) = 4x_1^2 + 4x_2^2 - 12x_2 - 4x_1x_2 \quad (P1) \quad (2.31)$$

On résout ce problème par la méthode du gradient. Le point de départ est $x^1 = (1,1)$ et la condition d'arrêt $\|\nabla f(x^k)\| < 0.01$. Le résultat est donné dans le tableau 2.1 (les valeurs ont été arrondies à deux décimales pour une lecture plus confortable) :

k	x^k	$f(x^k)$	$\nabla f(x^k)$	$d^k = -\nabla f(x^k)$	α^k	$\ \nabla f(x^k)\ $	x^{k+1}
1	(1.0, 1.0)	-8.0	(4.0,-8.0)	(-4.0, 8.0)	0.09	8.94	(0.64, 0.71)
2	(0.64, 1.71)	-11.57	(-1.71, -0.86)	(1.71,-0.86)	0.21	0.96	(1.0,1.89)
3	(1.0,1.89)	-11.95	(0.43,-0.86)	(-0.43, 0.86)	0.09	0.21	(0.96, 1.97)
4	(0.96, 1.97)	-12	(-0.18, -0.09)	(0.18, 0.09)	0.21	0.10	(1.0, 1.99)
5	(1.0, 1.99)	-12	(0.05, -0.09)	(-0.05, 0.09)	0.09	0.02	(1.0, 2.0)
6	(1.0, 2.0)	-12	(-0.02, -0.01)	(0.02,0.01)	0.21	0.01	(1.0,2.0)
7	(1.0, 2.0)	-12	(0.0, -0.01)	(0.0,0.01)	0.09	0.00	(1.0,2.0)
8	(1.0, 2.0)	-12	(0.0, 0.0)	-	-	-	-

TAB. 2.1 – *Exécution de la méthode du gradient pour (P1) depuis (1,1)*

Un premier constat est que la méthode du gradient n'a pas eu besoin de beaucoup d'itérations pour trouver la solution optimale de ce problème en partant de point initial (1,1). Cela est dû au fait que la direction menant au minimum est proche de la direction donnée par le gradient, permettant à la méthode de progresser rapidement.

Si nous résolvons le même problème en partant du point (2,27), nous obtenons :

k	x^k	$f(x^k)$	$\nabla f(x^k)$	$d^k = -\nabla f(x^k)$	α^k	$\ \nabla f(x^k)\ $	x^{k+1}
1	(2.00, 27.00)	2392.00	(-92.00 ,196.00)	(92.00 ,-196.00)	0.09	216.52	(10.31, 9.31)
2	(10.31, 9.31)	275.93	(45.22, 21.23)	(-45.22, -21.23)	49.95	0.96	(1.12,4.99)
3	(1.12,4.99)	22.49	(-11.02,23.48)	(11.02,-23.48)	0.09	25.93	(2.11, 2.88)
4	(2.11, 2.88)	-7.87	(5.42 , 2.54)	(- 5.42 , - 2.54)	0.09	5.98	(1.01, 2.36)
5	(1.01, 2.36)	-11.51	(-1.32 , 2.81)	(1.32 , -2.81)	0.20	3.11	(1.13,2.10)
6	(1.13,2.10)	-11.94	(0.65, 0.30)	(- 0.65,- 0.30)	0.21	0.72	(1.00, 2.04)
7	(1.00, 2.04)	-11.99	(-0.16 , 0.34)	(0.16 , - 0.34)	0.20	0.37	(1.02, 2.01)
8	(1.02, 2.01)	-12	(0.08 , 0.04)	(-0.08 ,- 0.04)	0.09	0.09	(1.00, 2.01)
9	(1.00, 2.01)	-12	(-0.02, 0.04)	(0.02,- 0.04)	0.20	0.04	(1.00, 2.00)
10	(1.00, 2.00)	-12	(0.01, 0.0)	(-0.01, 0.0)	0.09	0.01	(1.00, 2.00)
11	(1.00, 2.00)	-12	(0.0, 0.0)	-	-	-	-

TAB. 2.2 – *Exécution de la méthode du gradient pour le problème (P1) depuis (2,27)*

La méthode a eu besoin de plus d'itérations que lors de la première exécution, soit 3 de plus, pour trouver une approximation du minimum (1, 2) avec une précision équivalente.

Cela est dû au fait qu'à chaque itération, la direction du gradient est relativement éloignée de celle menant au minimum, conduisant à une convergence plus lente. Poussé à l'extrême, ce phénomène conduit à des exécutions au cours desquelles la direction du gradient est pratiquement orthogonale à celle menant au minimum, produisant le fameux phénomène de "zigzag", et conduisant à une convergence de la méthode considérablement lente.

Nous pouvons également observer la forte dépendance du comportement de la méthode du gradient à son point de départ. En effet, une conséquence du raisonnement précédent est que, pour un problème donné, il est possible qu'elle converge vite depuis un point initial et extrêmement lentement depuis un autre point.

Nous exécutons maintenant le même problème (P) avec l'algorithmes de Newton, en partant du même point départ $x^1 = (1,1)$ et la même condition d'arrêt $\|\nabla f(x^k)\| < 0.01$. Le résultat est donné dans le tableau 2.3 , nous obtenons

k	x^k	$f(x^k)$	$\nabla f(x^k)$	$\nabla^2 f(x^k)$	$(\nabla^2 f(x^k))^{-1}$	d^k	x^{k+1}
1	(1.0,1.0)	-8	(-4.0,-8.0)	[[8 -4;-4 8]]	[[0.17 0.08; 0.08 0.17]]	(0,1)	(1.0,2.0)
2	(1.0,2.0)	-12	(0.0, 0.0)	-	-	-	-

TAB. 2.3 – *Exécution de la méthode de Newton pour le problème (P1) depuis (1,1)*

Nous exécutons maintenant le même problème (P) en partant du point (2,27).

k	x^k	$f(x^k)$	$\nabla f(x^k)$	$\nabla^2 f(x^k)$	$(\nabla^2 f(x^k))^{-1}$	d^k	x^{k+1}
1	(2.0,27.0)	-8	(-4.0,-8.0)	[[8 -4;-4 8]]	[[0.17 0.08; 0.08 0.17]]	(-1,-25)	(1.0,2.0)
2	(1.0,2.0)	2392.00	(0.0, 0.0)	-	-	-	-

TAB. 2.4 – *Exécution de la méthode de Newton pour le problème (P1) depuis (2,27).*

La méthode de Newton est plus efficace que la descente de gradient, elle converge de manière finie en une seule itération, comme nous pouvons nous y attendre (et ceci indépendamment du point initial).

L'exécution de la méthode de gradient conjugué (β^{k+1} est calculer avec la formule de

Fletcher-Reeves) et de la méthode de Nelder et Mead donnent respectivement les résultats suivants :

k	x^k	$f(x^k)$	$\nabla f(x^k)$	direction conjugué	α^k	x^{k+1}
1	(1,1)	-8.00	(4.0, -8.0)	(4.0, -8.0)	0.09	(0.64, 1.71)
2	(0.64 , 1.71)	-11.75	(-1.71,-0.86)	(-1.71,-0.86)	0.23	(1.0,2.0)

TAB. 2.5 – *Exécution de la méthode du gradient conjuguée pour le problème (P1) depuis (1,1)*

k	x^k	$f(x^k)$	$\nabla f(x^k)$	direction conjugué	α^k	x^{k+1}
1	(2,27)	2392.0	(-92.0,196.0)	(-92.0,196.0)	0.09	(10.31,9.31)
2	(10.31,9.31)	275.93	(45.22,21.23)	(45.22,21.23)	0.23	(1.0,2.0)

TAB. 2.6 – *Exécution de la méthode du gradient conjuguée pour le problème (P1) depuis (2,27)*

La méthode de Nelder-Mead

itération	0	1	2	3	4	5
$f(x^k)$	-8	-8.39	-8.9075	-9.75688	-10.5017	-10.8923
étape	initialiation du simplexe	expansion	expansion	expansion	expansion	expansion

6	7	8	9	10	11	12
-10.8923	-10.8923	-10.9748	-11.0974	-11.312	-11.5145	-11.8241
contraction externe	contraction interne	réflexion	exansion	expansion	expansion	réflexion

13	14	15	16	17	18
-11.8241	-11.9979	-11.9979	-11.9979	-11.9979	-11.9979
réflexion	réflexion	contraction interne	contraction interne	contraction interne	réflexion

19	20	21	22	23	24	25
-11.9979	-11.9979	-11.9986	-11.9986	-11.9998	-11.9998	-11.9999
contraction interne	contraction interne	contraction interne	contraction interne	contraction interne	contraction interne	réflexion

26	27	28	29	30	31	32
	-11.9999	-12	-12	-12	-12	-12
contraction interne	contraction interne	contraction interne	contraction interne	réflexion	contraction interne	contraction interne

33	34	35	36	37	38	39
	-12	-12	-12	-12	-12	-12
contraction interne	contraction interne	contraction interne	contraction interne	réflexion	contraction interne	contraction interne

40	41	42	43
	-12	-12	-12
contraction interne	contraction interne	contraction interne	contraction interne

TAB. 2.7 – *Exécution de la méthode de Nelder-Mead pour le problème (P1)*

Ce dernier algorithme nécessite un nombre d'itérations plus important que les précédents. A première vue, nous pourrions déduire qu'il est moins efficace ou moins adapté à ce problème, néanmoins, nous devons garder à l'esprit qu'il ne requiert nullement la résolution d'un sous problème de minimisation monodimensionnel, comme c'est le cas de la méthode du gradient ni l'évaluation de l'inverse d'une matrice comme l'exige la méthode de Newton ; seules quelques opérations arithmétiques élémentaires et des comparaisons de valeurs sont nécessaires à chaque itération.

Conclusion

En conséquence, et après avoir résolu d'autres exemples quadratiques, on conclut que chacun des algorithmes décrits précédemment peut être utilisé pour résoudre ce type de problème ; cependant, la méthode de Newton et la méthode des directions conjuguées restent les mieux adaptées, de part leur convergence finie en un nombre fixe d'itérations. D'autre part la méthode de Nelder-Mead et du gradient révèlent dans ce cas les moins utiles. La méthode du gradient converge trop lentement. La méthode de Nelder-Mead permet d'obtenir une bonne solution sans trop d'effort, mais dans certaines configurations, le simplexe peut dégénérer, les points ne sont plus affinement indépendants. On peut, dans ce cas, utiliser la méthode de *Torczon*^[27] qui maintient la géométrie du simplexe.

Chapitre 3

Introduction

Nous avons traité, dans le chapitre précédent, de la résolution de problèmes de minimisation sans contraintes.

Dans ce chapitre, nous nous intéressons à la résolution de problèmes d'optimisation non linéaires sous contraintes.

Nous considérons le problème

$$\begin{cases} \min f(x), \\ \text{sous contrainte} \\ x \in X \end{cases}$$

où X est défini par une collection d'inégalités $g_i \leq (x)$ et d'égalités $h_j(x) = 0$.

Parmi les méthodes d'optimisation non linéaire, nous décrirons successivement : la méthode de Frank et Wolfe (1959)^[23] qui s'applique au cas des contraintes linéaires, en suite les méthodes de pénalisation.

3.1 La méthode de Frank et Wolfe

On s'intéresse à des problèmes avec contraintes linéaires de la forme :

$$(P) \begin{cases} \text{Minimiser } f(x) \\ \text{sous les contraintes} \\ A.x = b \\ x \geq 0 \end{cases}$$

Chapitre 3. Méthodes itératives d'optimisation non-linéaire sans contraintes 60

Beaucoup d'algorithmes ont été développés pour la résolution du problème (p), mais il serait intéressant de connaître la méthode la plus classique de Frank et Wolfe(1959)¹⁰.

Frank et Wolfe ont d'abord appliqué leur méthode au cas quadratique c-à-d:

$f(x) = \frac{1}{2}x^T D x + c^T x$, avec D symétrique, que l'on minimise sur un polyèdre convexe fermé. Ce genre de problèmes est convexe dès lors que la matrice A (la Hessienne de la fonction objectif) est semi-définie positive. Cette méthode n'est autre que la méthode du simplexe (de la programmation linéaire) légèrement modifiée. Le principe de cette méthode est la résolution du système de Kuhn-Tucker et consiste à trouver une solution réalisable pour un système linéaire avec une condition supplémentaire du type $x_j \delta_j$, où x et δ sont des vecteurs de même dimension. En d'autres termes, c'est trouver une solution réalisable basique initiale en résolvant un problème de programmation linéaire, assujetti à la nouvelle condition.

Formulons le théorème de KKT pour le problème suivant :

$$\left\{ \begin{array}{l} \text{Minimiser } f(x) = \frac{1}{2}x^T D x + c^T x \\ \text{sous les contraintes} \\ A.x = b \\ x \geq 0 \end{array} \right.$$

où $D' = D > 0, c \in \mathbb{R}^n, b \in \mathbb{R}^m, rang A = m < n$. Ce problème peut encore s'écrire sous la forme équivalente suivante :

$$\left\{ \begin{array}{l} \text{Minimiser } f(x) = \frac{1}{2}x^T D x + c^T x \\ \text{sous les contraintes} \\ A.x - b \leq 0 \\ -A.x + b \geq 0 \\ -x \leq 0 \end{array} \right.$$

La fonction Le point de minimum x^* est alors caractérisé par les équations et les inégalités suivantes :

$$\exists \lambda_1^* \geq 0, \lambda_2^* \geq 0 \quad \text{et} \quad \delta^* \geq 0 \quad \text{tels que}$$

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial x}(x^*, \lambda_1^*, \lambda_2^*, \delta^*) = 0, \\ \lambda_1^{*\prime}(Ax^* - b) = 0, \\ \lambda_2^{*\prime}(-Ax^* + b) = 0, \\ \delta^{*\prime}x^* = 0, \\ Ax^* - b = 0, \\ x^* \geq 0 \end{array} \right.$$

où

$$L(x, \lambda_1, \lambda_2, \delta) = \frac{1}{2}x'Dx + c'x + \lambda_1'(Ax - b) + \lambda_2'(-Ax + b) - \delta'x,$$

d'où

$$\frac{\partial L}{\partial x}(x, \lambda_1, \lambda_2, \delta) = Dx + c + A'\lambda_1 - A'\lambda_2 - \delta.$$

En posant $\lambda = \lambda_1 - \lambda_2, \lambda \in \mathbb{R}^m$ le point x^* est défini par le système suivant:

$$\left\{ \begin{array}{ll} Dx^* + A'\lambda^* - \delta^* = -c, & (L1) \\ Ax^* = b, & (L2) \\ \lambda^{*\prime}(Ax^* - b), & (L3) \\ \delta^{*\prime}x^* = 0, & (L4) \\ x^* \geq 0, \lambda^* \in \mathbb{R}^m, \delta^* \geq 0 \end{array} \right.$$

Comme l'équation (L4) est tout le temps vérifiée, le système se réduit ainsi :

$$\left\{ \begin{array}{ll} Dx^* + A'\lambda^* - \delta^* = -c, & (L1) \\ Ax^* = b, & (L2) \\ \lambda^{*\prime}(Ax^* - b), & (L3) \\ x^* \geq 0, \lambda^* \in \mathbb{R}^m, \delta^* \geq 0 \end{array} \right.$$

Un tel système n'est pas linéaire par rapport au multivecteur $(x^*, \lambda^*, \delta^*)$ à cause de l'équation (L3). On obtient donc un système linéaire de $(n+m)$ équations à $(n+m+n)$ inconnues, constitué des équations (L1) et (L2), avec en plus n équations non linéaires

Chapitre 3. Méthodes itératives d'optimisation non-linéaire sans contraintes62

$\delta_j x_j = 0, \quad j = \overline{1, n}$, il suffit d'obtenir une solution réalisable basique du système linéaire, avec x^j basique et δ_j non basique ou vice-versa. Pour appliquer la méthode du simplexe, il faut alors écrire le système (L1), (L2) sous forme standard, à savoir que le second membre doit être positif ou nul, ainsi que le vecteur λ^* qui doit être réécrit sous la forme : $\lambda_i^* = \alpha_i^* - \alpha_{m+1}^*, \quad \alpha_{m+1}^* \geq 0, \alpha_i^* > 0, i = \overline{1, m}$.

Algorithme de Frank et Wolfe

Début

1. *Initialisation*

Introduire les données, D,A,b,c;

Appliquer les conditions de K.K.T au problème;

Déterminer les équations de K.K.T

2. *Détermination des paramètres du programme linéaire :*

- Introduire les variables artificielles v_i ;
- Construire la matrice des contraintes A ;
- Construire le vecteur du second membre b
- Construire le vecteur des coûts c;

3. *Initialiser le vecteur solution (x, λ, δ, v) ;*
 - Déterminer l'ensemble des indices J_B et J_N ;
 - Extraire les éléments de base x_B, C_B, A_B ;
4. *Calculer le vecteur des potentiels $u' = c'_B A_B^{-1}$;*
 Calculer le vecteur des estimations $E'_N = u' A_N - c'_N$; Si $E'_N > 0$ Alors la solution actuelle est optimale ;
 Sinon aller à (5) ;
 FinSi
5. *Déterminer la variable qui entre en base tout en vérifiant la condition;*
 $\delta_j x_j = 0 \quad j = \overline{1, n}$;
 Déterminer la variable qui sort de la base;
 Mettre à jour A_B, X_B, C_B, J_g, J_N et aller en 4 ;
Fin

3.1.1 Convergence de la méthode de Frank et Wolfe

Théorème 3.1.1. [3]

On suppose que f est continûment différentiable et que l'une des deux conditions suivantes est vérifiée :

(i) *le polyèdre $X = \{x / Ax = b, x \geq 0\}$ est borné;*

(ii) *$f(x) \rightarrow +\infty$ quand $\|x\| \rightarrow +\infty$*

Alors pour tout point de départ $x^0 \in X$, la méthode de Frank et Wolfe converge vers un optimum local du problème

$$\min_{x \in \mathbb{R}^n} f(x). \tag{3.1}$$

3.2 Les méthodes de pénalités

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. C'est un concept qui a une utilité à la fois théorique et numérique. En analyse, l'approche par pénalisation est parfois 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. Si on a de la chance ou si la pénalisation est bien choisie, des passages à la limite parfois délicats permettent d'obtenir des propriétés du problème original (l'existence de solution par exemple). Les méthodes de pénalité constituent une famille d'algorithmes particulièrement intéressants du double point de vue de la simplicité de principe et de l'efficacité pratique. Parmi l'abondante littérature consacrée à ce sujet, on peut citer les travaux de Courant (1943)^[25], Ablow et Brigham (1955)^[26], Fiacco et McCormick (1968)^[7]. Nous commencerons par exposer le principe de base de toutes ces méthodes, puis nous étudierons plus spécialement deux des variantes les plus utilisées : les méthodes de pénalités extérieures et les méthodes de pénalités intérieures.

Nous considérons le problème suivant:

$$(P1) \left\{ \begin{array}{l} \text{Minimiser } f(x) \\ \text{sous les contraintes} \\ g_i(x) \leq 0 \quad (i = 1, 2, \dots, m) \\ x \in \mathbb{R}^n \end{array} \right.$$

où $f: \mathbb{R}^n \rightarrow \mathbb{R}$, et $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$.

Considérons le problème sans contrainte (problème pénalisé) :

$$(Pr) \left\{ \begin{array}{l} \text{Minimiser } \theta_r(x) \\ x \in \mathbb{R}^n \end{array} \right.$$

où $\theta_r(x)$ est obtenu en ajoutant à $f(x)$ le terme $rp(x)$:

$$\theta_r(x) = f(x) + rp(x)$$

avec

$$p(x) = \sum_{i=1}^m h(g_i(x))$$

et r est un scalaire strictement positif, appelé le facteur de pénalisation. Le but de ce terme additionnel est de pénaliser la violation des contraintes (on parle alors de pénalisation extérieure, ce que nous verrons à la section 3.3.1) ou l'abord de la frontière du domaine admissible (on parle dans ce cas de pénalisation intérieure, ce que nous étudierons à la section 3.3.3). L'intérêt de (Pr) est de pouvoir être résolu par une méthode d'optimisation

sans contrainte. La question qui se pose immédiatement est de savoir si en résolvant (Pr) on résout $(P1)$. Autrement dit, on cherche à savoir quand les ensembles de solutions de $(P1)$ et (Pr) coïncident. Cela va dépendre du choix de la fonction p et de r .

Par exemple, on pourrait choisir h comme suit :

$$h(y) \begin{cases} 0 & \text{si } y \leq 0 \\ +\infty & \text{si } y > 0 \end{cases}$$

Il est clair que dans ce cas, les problèmes $(P1)$ et (Pr) sont identiques : ils ont les mêmes ensembles de solutions. Cette fonction de pénalisation est parfois utilisée dans la théorie, car elle permet de traiter en même temps les problèmes avec ou sans contrainte . Cependant, ce choix de p n'est pas très utile en pratique car les méthodes classiques d'optimisation ne peuvent pas être utilisées sur des fonctions qui prennent la valeur $+\infty$ dans des régions visitées par les itérés (dans quelle direction se déplacer pour faire décroître θ_r si l'itéré courant n'est pas dans X ?). Nous allons donc, dans cette section, introduire divers termes de pénalisation $rp(\cdot)$ et en étudier les propriétés théoriques et algorithmiques.

3.2.1 Méthode de pénalité extérieure

La difficulté mentionnée ci-dessus peut être évitée en utilisant des fonctions de pénalisation continues et à dérivées continues. Ainsi, dans la méthode dite des «pénalités extérieures», Fiacco et McCormick (1968) ont proposé de prendre :

$$\begin{cases} h(y) = 0 & \text{si } y \leq 0 \\ h(y) = y^2 & \text{si } y > 0 \end{cases}$$

En posant :

$$p(x) = \sum_{i=1}^m h(g_i(x)) = \sum_{i=1}^m [g_i^+(x)]^2$$

où

$$\forall x, \quad g_i^+(x) = \text{Max}\{0, g_i(x)\}$$

p est appelée fonction de pénalisation extérieure.

Le problème $(P1)$ est alors remplacé par le problème d'optimisation sans contrainte (problème pénalisé) :

$$(Pr) \begin{cases} \text{Minimiser } \theta(x,r) = f(x) + r \sum_{i=1}^m \text{Max}\{0, g_i(x)\}^2 \\ x \in \mathbb{R}^n \end{cases}$$

Chapitre 3. Méthodes itératives d'optimisation non-linéaire sans contraintes 66

où $r > 0$ est appelé le coefficient de pénalité, notons $\bar{x}(r)$ un minimum de $\theta(x, r)$.

Le choix d'une valeur appropriée du coefficient de pénalité r résulte d'un compromis : d'une part, r doit être choisi suffisamment grand pour que le point $\bar{x}(r)$ obtenu soit proche de l'ensemble des solutions X (autrement dit, que p soit suffisamment faible); d'autre part, si r est choisi trop grand, la fonction θ peut être mal conditionnée d'où des difficultés numériques dans la recherche de l'optimum sans contrainte.

Ceci explique pourquoi les méthodes de pénalités sont généralement mises en oeuvre sous forme itérative de la façon suivante.

On commence par choisir un coefficient de pénalité r_1 de valeur pas trop élevée (pour éviter les difficultés numériques) puis on résout le problème sans contrainte :

$$\min_{x \in \mathbb{R}^n} \theta(x, r_1) = f(x) + r_1 p(x)$$

Soit $\bar{x}(r_1)$ le point obtenu.

Si la quantité $p(\bar{x}(r_1))$ est suffisamment faible, $\bar{x}(r_1)$ est une bonne approximation de l'optimum, et les calculs sont terminés. Dans le cas contraire, c'est que la pénalité associée à la violation des contraintes n'est pas assez élevée. On choisira donc un coefficient de pénalité $r_2 > r_1$ et on résoudra le nouveau problème sans contrainte :

$$\min_{x \in \mathbb{R}^n} \theta(x, r_2) = f(x) + r_2 p(x)$$

On obtiendra un nouveau point $\bar{x}(r_2)$, et ainsi de suite.

Remarquons qu'à chaque étape k du processus précédent, il est avantageux d'utiliser le point $\bar{x}(r_{k-1})$ obtenu à l'étape précédente comme point de départ de l'algorithme d'optimisation sans contrainte choisi (gradient conjugué, méthode de quasi Newton, ...).

Algorithme de pénalité extérieure

1. **Initialisation**

fixer $\varepsilon > 0$, choisir x^0 non admissible, , choisir $\gamma > 1$ et $(r_1 \simeq 1)$

2. **Itération** $k = 1, 2, \dots,$

$$\theta(x, r_k) = f(x) + r_k \sum_{i=1}^m \text{Max}\{0, g_i(x)\}^2$$

calculer

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

avec l'un des méthodes étudié dans le chapitre 1

3. **Critère d'arrêt**

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP x^k est optimal

Sinon, on pose $k = k + 1$ et on retourne à 2 avec $r_{k+1} = \gamma r_k$.

Remarque 3.2.1.

1. On commence toujours par un point non admissible.
2. Pour le choix du paramètre de pénalité on commence par des valeurs assez petites ($r_1 = 1$) ensuite l'augmente au fur et à mesure des itérations.

3.2.2 Convergence de la pénalisation extérieure

Le théorème ci-dessous étudie le comportement des solutions \bar{x}^r de (Pr) , lorsque r tend vers l'infini. Il donne des conditions pour que les solutions des problèmes pénalisés convergent vers une solution du problème original.

Théorème 3.2.1. [3]

Soit $p : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction de pénalisation extérieure vérifiant :

i) $p(x) \geq 0 \quad \forall x.$

ii) $p(x) = 0 \iff x \in X = \{x/g_i(x) \leq 0, i = 1, \dots, m\}$.

iii) p continue.

On suppose, d'autre part, que f est une fonction continue, que X est fermé, et que l'une des deux conditions suivantes est vérifiée :

a) $p(x) \geq 0 \quad \forall x$.

b) X est borné et $p(x) \longrightarrow +\infty$ quand $\|x\| \longrightarrow +\infty$.

Alors, lorsque le coefficient de pénalité r tend vers $+\infty$:

- la suite $\{\bar{x}(r)\}$ admet au moins un point d'accumulation, et tout point d'accumulation de cette suite est une solution optimale (globale) du problème (P1),

- X est borné et $p(x) \longrightarrow +\infty$ quand $\|x\| \longrightarrow +\infty$,

- $p(\bar{x}(r)) \longrightarrow 0$.

Preuve.

Les hypothèses du théorème entraînent l'existence de $x^* \in X$, solution optimale de (P1). On a donc :

$$\forall x \in X : \quad f(x) \geq f(x^*).$$

Considérons alors une suite infinie et croissante $\{r_k\}$ de valeurs de r telles que :

$$r_k \longrightarrow +\infty \quad (k \longrightarrow +\infty)$$

et soit $\{x^k\}$ la suite des valeurs correspondantes de x ($x^k = \bar{x}(r_k)$). Commençons par établir quelques relations utiles.

On a

$$f(x^{k+1}) + r_{k+1}p(x^{k+1}) > f(x^{k+1}) + r_k p(x^{k+1}) \geq f(x^k) + r_k p(x^k)$$

d'où :

$$\theta(x^{k+1}, r_{k+1}) > \theta(x^k, r_k) \quad \forall k. \tag{3.2}$$

Remarquons ensuite que, par définition de x^* et x^{k+1} , on a :

$$f(x^k) + r_k p(x^k) \leq f(x^{k+1}) + r_k p(x^{k+1})$$

et

$$f(x^{k+1}) + r_{k+1}p(x^{k+1}) \leq f(x^k) + r_{k+1}p(x^k)$$

d'où, en additionnant :

$$(r_{k+1} - r_k)p(x^{k+1}) \leq (r_{k+1} - r_k)p(x^k)$$

et comme $r_{k+1} > r_k$ il vient :

$$p(x^{k+1}) \leq p(x^k) \quad \forall k \tag{3.3}$$

Enfin pour tout k on peut écrire :

$$f(x^k) \leq f(x^k) + r_k p(x^k) \leq f(x^*) + r_k p(x^*)$$

où x^* est une solution optimale de $(P1)$. Comme $x^* \in X$, $p(x^*) = 0$ et on en déduit :

$$f(x^k) \leq \theta(x^k, r_k) \leq f(x^*), \quad \forall k \tag{3.4}$$

Ceci étant, on observe que les hypothèses du théorème impliquent que toute la suite $\{x^k\}$ est contenue dans un ensemble borné. En effet, si la condition (a) est vérifiée, cette propriété se déduit de la relation (3) :

$$f(x^k) \leq f(x^*), \quad \forall k$$

Si c'est la condition (b) qui est vérifiée, la propriété se déduit de la relation (2) qui montre que,

$$p(x^k) \leq p(x^1), \quad \forall k$$

où x^1 est le premier terme de la suite $\{x^k\}$.

Dans tous les cas, on peut donc extraire de $\{x^k\}$ une sous-suite $\{x^l\}_{l \in L}$ ($L \subset \mathbb{N}$) convergent vers \bar{x} .

Comme f est continue, on a :

$$\lim_{l \in L} f(x^l) = f(\bar{x})$$

et d'après (3) :

$$f(\bar{x}) \leq f(x^*)$$

La suite des valeurs $\theta(x^k, r_k)$ est monotone croissante d'après (1) et majorée par $f(x^*)$ d'après (3).

Elle a donc une limite $\theta^* \leq f(x^*)$.

On en déduit :

$$f(x^k) + r_k p(x^k) \longrightarrow \theta^*$$

et pour $l \in L$:

$$\lim_{l \in L} r_l p(x^l) = \theta^* - f(\bar{x})$$

Puisque $p(x^l) \geq 0$ et $r_l \longrightarrow +\infty$ on en déduit :

$$p(x^l) \longrightarrow 0 \quad \text{quand} \quad l \longrightarrow \infty \quad (l \in L).$$

Alors, en utilisant la continuité de p , il vient $p(\bar{x}) = 0$ ce qui prouve que $\bar{x} \in X$. Comme d'autre part $f(x) \leq f(x^*)$ on en déduit que nécessairement $f(\bar{x}) = f(x^*)$ et \bar{x} est bien une solution optimale de (P1).

Remarque 3.2.2.

- Pour que la suite $\{\bar{x}_r\}$ s'approche d'une solution de (P1), il faut faire croître le facteur de pénalisation r . Dans ces conditions, le théorème nous apprend que la suite $\{f(\bar{x}_r)\}$ croît. Si la croissance de cette suite est stricte, les points \bar{x}_r ne peuvent être qu'extérieurs à X , sinon les point d'accumelation de $\{\bar{x}_r\}$ ne pourraient pas être solutions de (P1). En effet, on aurait alors des points de X aussi proches que l'on veut d'une solution \bar{x} en lesquels f prendrait une valeur strictement inférieure à $f(\bar{x})$, ce qui contredirait l'optimalité de \bar{x} .
- Le qualificatif "extérieur" vient de la propriété (iii), qui exprime que $\theta(x, r)$ ne modifie f qu'à l'extérieur de l'ensemble admissible.

3.2.3 Méthodes de pénalité intérieure

Dans certains problèmes, le fait que les itérés \bar{x}_r générés par pénalisation extérieure ne soient pas admissibles peut être un inconvénient, par exemple, parce que f n'est pas définie à l'extérieur de X . On peut introduire des méthodes de pénalisation dans lesquelles les itérés \bar{x}_r restent dans X . C'est ce qui a conduit *Fiacco et McCormick* (1968) à proposer d'autres méthodes de pénalisation dans lesquelles l'optimum est approché par l'intérieur

Chapitre 3. Méthodes itératives d'optimisation non-linéaire sans contraintes 71

(d'où le nom de méthodes de « pénalités intérieures »).

Supposons que l'ensemble des solutions X de $(P1)$ soit tel que :

- l'intérieur de X est non vide,
- tout point de la frontière de X est limite d'une suite de points appartenant à l'intérieur de X .

Soit $B(x)$ la fonction définie par :

$$B(x) = - \sum_{i=1}^m \frac{1}{g_i(x)} \quad (i = 1, \dots, m).$$

Les termes de pénalisation B considérés dans cette section satisfont les conditions suivantes:

- $B(x) \geq 0 \quad \forall x \in \text{int}(X)$, avec $\text{int}(X)$: désigne l'intérieur de X ,
- $B(x) \rightarrow +\infty$ lorsque x tend vers la frontière de X ,
- en supposant les fonctions $g_i \quad (i = 1, \dots, m)$ continues, $B(x)$ est continue sur l'intérieur de X .

Une telle fonction est appelée une fonction de pénalisation intérieure (on dit encore, fonction « barrière »).

Considérons alors la fonction :

$$\phi(x,t) = f(x) + tB(x)$$

où t est un réel positif, est appelé le coefficient de pénalité.

Si f est continue sur X et que l'une des conditions suivantes est vérifiée :

- $f(x) \rightarrow +\infty \quad (\|x\| \rightarrow +\infty)$,
- X est borné.

alors pour toute valeur $t > 0$, $\phi(x,t)$ admet nécessairement un minimum sur X , soit $\bar{x}(t)$, et nécessairement $\bar{x} \in \text{int}(X)$.

Le principe des méthodes de pénalités intérieures est alors le suivant.

Choisissons une valeur $t_1 > 0$ et recherchons (par une méthode itérative d'optimisation sans contrainte) un minimum de $\phi(x,t)$ à partir de $x^0 \in \text{int}(X)$.

Dans le processus de minimisation de $\phi(x,t_1)$ on remarque que l'on ne peut jamais traverser

la frontière de X puisque au voisinage de celle-ci $\phi(x, t_1) \rightarrow +\infty$.

On obtient donc $x^1 = \bar{x}(t_1) \in \text{int}(X)$. Si la quantité $t_1 \cdot B(x^1)$ est suffisamment faible, alors x^1 est une bonne approximation de l'optimum de f sur X et les calculs s'arrêtent.

Dans le cas contraire on choisira une valeur $t_2 < t_1$, et on recherchera, de nouveau, $x^2 = \bar{x}(t_2)$ minimum de $\phi(x, t_2)$ en partant de $x^1 = \bar{x}(t_1)$, le point précédent. Le processus est répété jusqu'à obtention d'une approximation acceptable de l'optimum de $(P1)$.

Le résultat suivant montre que, moyennant des hypothèses très peu restrictives, la méthode des pénalités intérieures converge vers l'optimum du problème $(P1)$.

Algorithme de pénalité intérieure

1. *Initialisation*

fixer $\varepsilon > 0$, choisir x^0 admissible, , choisir $\eta > 1$ et $(t_1 > 0)$ on prend $(t_1 = 10)$

2. *Itération* $k = 1, 2, \dots$

$$\phi(x, r_k) = f(x) + (t_k \sum_{i=1}^m -\frac{1}{g_i(x)})$$

calculer

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

avec l'un des méthodes étudiées dans le chapitre 1

3. *Critère d'arrêt*

Si $\|x^{k+1} - x^k\| < \varepsilon$, STOP

Si non, on pose $k = k + 1$ et on retourne à 2 avec $t_{k+1} = \eta t_k$.

3.2.4 Convergence de la pénalisation intérieure

Le théorème suivant étudie la suite $\{\bar{x}_t\}$ des solutions des problèmes pénalisés. Contrairement à la pénalisation extérieure, il faut ici faire tendre t vers 0 (et non vers $+\infty$), ce qui a pour effet de diminuer l'influence du terme pénalisant qui repousse les points vers l'intérieur de X et donc de permettre à \bar{x}_t de se rapprocher de la frontière du domaine admissible, si cela est nécessaire.

Théorème 3.2.2.

Soit $X = \{x/g_i(x) \leq 0 \quad (i = 1, \dots, m)\}$ l'ensemble des solutions du problème (P1).

On suppose que X est fermé, qu'il a un intérieur non vide et que tout $x \in X$ est limite d'une suite de points appartenant à l'intérieur de X .

Soit $B(\mathbb{R}^n \rightarrow \mathbb{R})$ une fonction de pénalisation intérieure vérifiant :

- $B(x) \geq 0 \quad \forall x \in \text{int}(X)$,
- $B(x) \rightarrow +\infty$ lorsque x tend vers la frontière de X ,
- $B(x)$ est continue int(X).

On suppose, d'autre part, que f est une fonction continue, et que l'une des deux conditions suivantes est vérifiée:

- $f(x) \rightarrow +\infty$ quand ($\|x\| \rightarrow +\infty$),
- X est borné;

Alors, lorsque le coefficient de pénalité t tend vers 0 :

- la suite $\bar{x}(t)$ admet au moins un point d'accumulation et tout point d'accumulation de la suite $\bar{x}(t)$ est un optimum (global) du problème (P1),
- la quantité $t.B(\bar{x}(t))$ tend vers 0.

Preuve. On remarque que les hypothèses du théorème (3.2.2) impliquent l'existence de $x^* \in X$ solution optimale de (P1).

Considérons alors une suite infinie $\{t_k\}$ et décroissante de valeurs positives de ($t_k \rightarrow 0$ quand $k \rightarrow \infty$) et soit $\{x^k\}$ la suite des valeurs correspondantes de $x(x^k = \bar{x}(t_k))$.

Pour tout k , on peut écrire :

$$f(x^*) \leq f(x^k) \leq f(x^k) + t_k B(x^k) = \theta(x^k, t_k). \tag{3.5}$$

En vertu de la continuité de f , et d'après les hypothèses faites sur l'ensemble X , $\forall \varepsilon > 0$ il existe $\bar{x} \in \text{int}(X)$ tel que :

$$f(\bar{x}) \leq f(x^*) + \varepsilon,$$

Ceci entraîne, $\forall k$:

$$f(x^*) + \varepsilon + t_k B(\bar{x}) \geq f(\bar{x}) + t_k B(\bar{x}) \geq \theta(x^k, t_k).$$

On a donc :

$$\lim_{k \rightarrow \infty} \theta(x^k, t_k) \leq f(x^*) + \varepsilon.$$

Cette relation étant vraie $\forall \varepsilon > 0$, on en déduit, en utilisant (3.7) :

$$\lim_{k \rightarrow \infty} \theta(x^k, t_k) = f(x^*)$$

$$\lim_{k \rightarrow \infty} f(x^k) = f(x^*)$$

et

$$\lim_{k \rightarrow \infty} t_k B = 0.$$

Ceci étant, les hypothèses du théorème (3.2.2) impliquent que toute la suite $\{x^k\}$ est contenue dans un ensemble borné.

On peut donc extraire une sous-suite $\{x^e\}_{e \in E}$ ($E \subset \mathbb{N}$) convergeant vers \bar{x} . Par continuité de f on a :

$$f(x) = f(x^*)$$

et comme X est fermé :

$$x^e \in X \implies \bar{x} \in X.$$

On en déduit que tout point d'accumulation de la suite $\{x^k\}$ est un optimum (global) de (P1).

Proposition 3.2.1.

Soit x^k solution du problème sans contraintes suivant

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

On a les relations suivantes:

$$i) \quad \forall k : \quad \phi(x^{k+1}, t_{k+1}) < \phi(x^k, t_k).$$

$$ii) \quad \forall k : \quad B(x^k) \leq B(x^{k+1}).$$

$$iii) \quad \forall k : \quad f(x^k) \leq f(x^{k+1}).$$

La dernière inégalité montre, en particulier, que la suite des valeurs de la fonction objectif obtenues est monotone décroissante.

Les méthodes de pénalités intérieures sont très intéressantes pour traiter des problèmes d'optimisation avec contraintes fortement non linéaires. Elles supposent, en revanche, que l'on connaisse une solution de départ x^0 appartenant à l'intérieur de X , l'ensemble des solutions.

D'autre part, leur mise en oeuvre nécessite un certain nombre de précautions, en particulier au niveau des procédures de recherche unidimensionnelle, pour éviter de sortir de l'ensemble des solutions X .

3.2.5 Etude comparative des méthodes d'optimisation avec contraintes

La méthode de Frank et Wolfe

Soit le problème quadratique suivant :

$$(P2) \begin{cases} \min f(x) = x_1^2 + x_1x_2 + 6x_2^2 - 2x_1 + 8x_2 \\ x_1 + 2x_2 \leq 4 \\ 2x_1 + x_2 \leq 5 \\ x_1, x_2 \geq 0 \end{cases}$$

Le problème (P2) se formule ainsi :

$$(P2) \begin{cases} \min f(x) = x_1^2 + x_1x_2 + 6x_2^2 - 2x_1 + 8x_2 \\ x_1 + 2x_2 + x_3 = 4 \\ 2x_1 + x_2 + x_4 = 5 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

$$L(x, \lambda, \delta) = x_1^2 + x_1x_2 + 6x_2^2 - 2x_1 + 8x_2 + \lambda_1(x_1 + 2x_2 + x_3 - 4) + \lambda_2(2x_1 + x_2 + x_4 - 5) - \delta_1x_1 - \delta_2x_2 - \delta_3x_3 - \delta_4x_4$$

$$\begin{cases} \frac{\partial L}{\partial x_1}(x, \lambda, \delta) = 0 \\ \frac{\partial L}{\partial x_2}(x, \lambda, \delta) = 0 \\ \frac{\partial L}{\partial x_3}(x, \lambda, \delta) = 0 \\ \frac{\partial L}{\partial x_4}(x, \lambda, \delta) = 0 \\ \text{et} \\ x_1 + 2x_2 + x_3 = 4 \\ 2x_1 + x_2 + x_4 = 5 \\ \delta_j x_j = 0 \\ x_j \geq 0; \delta_j \geq 0, j = \overline{1,4} \end{cases}$$

Alors on obtient le système suivant :

$$\begin{cases} 2x_1 + x_2 - \delta_1 + \delta_3 + 2\delta_4 = 2 \\ -x_1 - 12x_2 + \delta_2 - 2\delta_3 - \delta_4 = 8 \\ x_1 + 2x_2 + x_4 = 5 \\ \delta_j x_j = 0 \\ x_j \geq 0; \delta_j \geq 0, j = \overline{1,4} \end{cases}$$

Nous avons obtenu un système linéaire de 4 équations avec 8 inconnues et 4 équations non linéaires. Pour trouver une solution telle que $\delta_j x_j = 0$ avec $j = \overline{1,4}$, il suffit d'obtenir une solution réalisable basique (x, δ) du système linéaire avec x_j basique et δ_j hors base ou vice-versa. Pour cela, il faut choisir l'indice j_0 entrant dans la base de telle sorte que les vecteurs-colonnes correspondant à x_{j_0} et δ_{j_0} ne se trouvent pas en même temps dans la base.

Appliquons la première phase du simplexe en considérant le problème de programmation linéaire suivant :

$$\begin{cases} Z = -v_1 \\ 2x_1 + x_2 - \delta_1 + \delta_3 + 2\delta_4 + v_1 = 2 \\ (P') \quad -x_1 - 12x_2 + \delta_2 - 2\delta_3 - \delta_4 = 8 \\ x_1 + 2x_2 + x_4 = 5 \\ x_j \geq 0; \delta_j \geq 0, j = \overline{1,4}, v_1 \geq 0 \end{cases}$$

Le vecteur $x = (x, \delta, v_1) = (0, 0, 4, 5, 0, 8, 0, 0, 2)$ est une solution réalisable initiale basique du problème (P'), avec $A_B = (a_9, a_6, a_3, a_4)$. Dressons alors les tableaux simplexes suivants :

		\bar{x}	x_1	x_2	x_3	x_4	δ_1	δ_2	δ_3	δ_4	v_1		
	c	b	0	0	0	0	0	0	0	0	-1		
c_B	basse	b	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	θ	
-1	a_9	2	2	1	0	0	-1	0	1	2	1	1	$\longrightarrow j_1 = 9$
0	a_6	8	-1	-12	0	0	0	1	-2	-1	0	∞	
0	a_3	4	1	2	1	0	0	0	0	0	0	4	
0	a_4	5	2	1	0	1	0	0	0	0	0	5/2	
$Z = -2$		E	-2	-1	0	0	1	0	-1	-2	0		
			$\uparrow j_0 = 1$										

TAB. 3.1 – Exécution de la méthode de Frank Wolfe (Phase 1 du simplexe)

Remarquons que $j_0 = 1$ est l'indice qui entre dans la base tandis que l'indice $j_1 = 9$ sort, car l'indice $j = 8$ ne peut pas être choisi à cause de la présence de a_4 dans la base, et ce,

afin que la condition $x_j \delta_j = 0$ soit assurés.

		\bar{x}	x_1	x_2	x_3	x_4	δ_1	δ_2	δ_3	δ_4	v_1	
		c	0	0	0	0	0	0	0	0	-1	
c_B	basse	b	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	θ
0	a_1	1	1	1/2	0	0	-1/2	0	1/2	1	1/2	
0	a_6	9	0	-23/12	0	0	-1/2	1	-3/2	0	1/2	
0	a_3	3	0	3/2	1	0	1/2	0	-1/2	-1	-1/2	
0	a_4	3	0	0	0	1	1	0	-1	-2	-1	
$Z = 0$		E	0	0	0	0	0	0	0	0	1	

TAB. 3.2 – *Exécution de la méthode de Frank Wolfe (Phase 2 du simplexe)*

Le critère d'optimalité étant vérifié, il s'ensuit que le vecteur $x = (1,0,3,3,0,9,0,0,0)$ est une solution optimale du problème linéaire. Par conséquent, la solution du problème (P2) est le vecteur $x^* = (1,0)$ avec $f(x^*) = -1$.

La méthodes de Frank et Wolfe se base sur le gradient de la fonction objectif pour déterminer une direction de mouvement ; on ne peut donc pas attendre de cette méthode qu'elle convergent vers des minima globaux ; elle serai attirée par tout type de minimum, local ou global. La méthode de Frank et Wolfe dans certain cas tomber dans le cas d'une convergente très lente en zigzag. En effet, le point x^k déterminé à chaque itération afin de produire une direction est typiquement un point extrême du polytope formé par les contrainte. Il est donc possible que la direction obtenue soit presque orthogonale à la direction menant au minimum.

Les méthodes de pénalité

Bien que les méthodes de pénalités sont très bien supportée par la théorie et nécessite peu de restrictions sur le problème pour assurer une convergence théorique, elle souffre de faiblesses dans le cadre de l'utilisation pratique, particulièrement lorsque la frontière de la région admissible est approchée.

Soit P3 :

$$(P3) \begin{cases} \min f(x) = (x_1 - 6)^2 + (x_2 - 7)^2 \\ -3x_1 - 2x_2 \leq -6 \\ -x_1 + x_2 \leq 3 \\ x_1 + x_2 \leq 7 \\ \frac{2}{3}x_1 - x_2 \leq \frac{4}{3} \end{cases}$$

Réolvons ce problème par la méthode de pénalité extérieure, en démarrant de point non admissible (6,7), avec un paramètre $r_1 = 0.5$, un facteur d'augmentation de $\gamma = 2$ et la condition d'arrêt $\|x^{k+1} - x^k\| < 0.01$. Le résultat est donné dans le tableau (3.2):

k	x^k	$f(x^k)$	$P(x^k, r_k)$	r_k	$g(x^k)$	x^{k+1}
0	(6.00,7.00)	0	9.00	0.50	6.00	(4.00,5.00)
1	(4.00,5.00)	8.00	12.00	1.00	3.00	(3.60,4.60)
2	(3.60,4.60)	11.52	14.40	2.00	2.00	(3.33,4.33)
3	(3.33,4.33)	14.22	16.00	4.00	1.20	(3.18,4.18)
4	(3.18,4.18)	15.94	16.94	8.00	0.67	(3.09,4.09)
5	(3.09,4.09)	16.93	17.45	16.00	0.35	(3.05,4.05)
6	(3.05,4.05)	17.45	17.72	32.00	0.18	(3.02,4.02)
7	(3.02,4.02)	17.72	17.86	64.00	0.09	(3.01,4.01)
8	(3.01,4.01)	17.86	17.93	128.00	0.05	(3.01,4.01)
9	(3.01,4.01)	17.93	17.96	256.00	-	-

TAB. 3.3 – *Exécution de la méthode de pénalité extérieure, pour le problème (P3) depuis un point non admissible (6,7)*

Un premier constat est que la méthode de pénalité extérieur n'a pas eu besoin de beaucoup d'itérations pour trouver la solution optimale de ce problème en partant de point initial non admissible (6,7). Les résultats de la table 3.2, nous apprend que les suites $\{x^k\}, \{r_k\}, \{f(x^k)\}$ et $\{P(x^k, r_k)\}$ croît, par contre la suite $\{g(x^k) = \sum_{i=1}^m (\max\{0, g_i(x^k)\})\}$ décroît. Donc les conditions pour que les solutions des problème pénalisée convergent vers une solution du problème original sont vérifiées.

Exécutons maintenant la méthode de pénalité intérieure pour ce problème, depuis le point admissible (2,2) avec un paramètre initial $t = 10$ et un facteur de réduction associé $\eta = 0.8$ et la condition d'arrêt $\|x^{k+1} - x^k\| < 0.01$. Le résultat est donné dans le tableau (3.2)

k	x^k	$f(x^k)$	$P(x^k, t_k)$	t_k	x^{k+1}
0	(2.00,2.00)	41.00	55.17	10.00	(2.72,3.24)
1	(2.72,3.24)	24.86	39.64	8.00	(2.75,3.31)
2	(2.75,3.31)	24.16	36.61	6.40	(2.78,3.38)
3	(2.78,3.38)	23.53	34.05	5.12	(2.80,3.44)
4	(2.80,3.44)	22.96	31.89	4.10	(2.81,3.50)
5	(2.81,3.50)	22.44	30.5	3.28	(2.83,3.55)
6	(2.83,3.55)	21.97	28.48	2.62	(2.84,3.60)
7	(2.84,3.60)	21.55	27.13	2.10	(2.85,3.64)
8	(2.85,3.64)	21.18	25.98	1.68	(2.86,3.69)
9	(2.86,3.69)	20.84	24.98	1.34	(2.87,3.72)
10	(2.87,3.72)	20.54	24.12	1.07	(2.88,3.76)
11	(2.88,3.76)	20.27	23.38	0.86	(2.89,3.78)
12	(2.89,3.78)	20.03	22.73	0.69	(2.89,3.81)
13	(2.89,3.81)	19.82	22.17	0.55	(2.90,3.83)
14	(2.90,3.83)	19.62	21.68	0.44	(2.91,3.85)
15	(2.91,3.85)	19.45	21.25	0.35	(2.92,3.87)
16	(2.92,3.87)	19.30	20.87	0.28	(2.92,3.89)
17	(2.92,3.89)	19.16	20.54	0.23	(2.94,4.06)
18	(2.94,4.06)	18.01	-	-	-
19	(2.94,4.06)	18.01	-	-	

TAB. 3.4 – *Exécution de la méthode de pénalité intérieur, pour le problème (P3) depuis un point admissible (2.2)*

Il n'y a pas de technique générale pour le choix du paramètre t et du facteur de réduction.

- Lorsque $\{t_k\}$ est très proche de zéro, l'arrondissement des valeurs dû au calcul en nombre de chiffres limité que l'ordinateur utilise entraîne la propagation des erreurs dues à celui-ci.
- Lorsque la méthode approche l'extrêmité du domaine, un problème est posé, dû aux algorithmes de minimisation directe utilisés. L'on se convaincra aisément du fait que le comportement d'une méthode de barrière dépend fortement du choix de cet algorithme.

Un raisonnement analogue à celui développé pour la méthode de pénalité extérieure. La méthode a eu besoin plus d'itérations, soit 10 de plus que la méthode de pénalité extérieures, pour trouver une approximation du minimum avec une précision équivalente. Les suites $\{x^k\}, \{r_k\}, \{f(x^k)\}$ et $\{P(x^k, r_k)\}, \{x^k\}, \{r_k\}$ décroît, d'où la converge de la méthode vers la solution optimale globale.

Conclusion

En conséquence, et après avoir résolu d'autres exemples, on conclut que chacun des algorithmes décrits précédemment peut être utilisé pour résoudre ce type de problème; cependant, la méthode de Frank Wolfe converge toujours, mais il existe des cas où la méthode "zigzague". Les méthodes de pénalité convergent vers la solution mais, elles souffrent de faiblesses dans le cadre de l'utilisation pratique, particulièrement lorsque la frontière de la région admissible est approchée.

Conclusion générale

A travers ce mémoire, notre objectif était l'étude théorique puis l'implémentation des méthodes d'optimisation dans le cas non linéaire. Pour ce faire, nous avons étudié et implémenté différentes approches de résolution. Pour cela, nous avons d'abord introduit quelques notions classiques importantes sur la convexité, entre autres, les conditions d'optimalité de KKT. Afin d'analyser ou de résoudre de manière efficace un problème d'optimisation, il est fondamental de pouvoir disposer de conditions d'optimalité. En effet, celles-ci nous servent non seulement à vérifier la validité des solutions obtenues, mais souvent l'étude de ces conditions aboutit au développement des algorithmes de résolution eux-mêmes.

Nous nous sommes intéressés ensuite à résoudre des problèmes d'optimisation sans contraintes pour le cas non linéaire. En premier lieu, nous avons abordé les méthodes de descente basée sur la descente du gradient, qui consiste simplement à suivre une direction de descente de façon itérative jusqu'à l'obtention d'un "bon" minimiseur. Ces méthodes utilisent une approximation linéaire pour trouver une direction de mouvement. Ensuite nous avons traité la méthode itérative de Newton. L'idée de cette méthode est de minimiser à chaque itération l'approximation quadratique de f , la convergence de la méthode est rapide car elle est quadratique. La méthode du gradient conjugué basée sur l'important concept de la conjugaison et a été développée afin de résoudre les problèmes quadratique en au plus n itérations. Nous présentons ensuite la méthode du simplexe de Nelder-Mead qui appartient à la classe des méthodes de recherche directes, c'est une méthode d'optimisation locale. Toutefois, elle présente l'avantage qu'elle ne nécessite aucune information sur la fonction objectif à optimiser.

Nous avons élaboré les programmes de ces méthodes et l'implémentation nous a permis de résoudre des exemples et réaliser une étude comparative entre ces méthodes d'optimisation. Nous avons constaté que les méthodes de Newton et du gradient conjuguées restent les

mieux adaptées, de part leur convergence finie en un nombre fixe d'itérations. D'autre part la méthode de Nelder-Mead et du gradient révèle dans ce cas les moins utiles. La méthode du gradient converge trop lentement. La méthode de Nelder-Mead permet d'obtenir une bonne solution sans trop d'efforts, mais dans certaines configurations, le simplexe peut dégénérer, les points ne sont plus affinement indépendants. Après avoir introduit les méthodes d'optimisation pour les problèmes sans contraintes, nous avons présenté en détail les méthodes d'optimisation pour les problèmes avec contraintes. Nous avons d'abord abordé la méthode de Franf Wolfe qu'est applicable si le problème est quadratique et convexe. Nous terminons par les méthodes de pénalités extérieure et intérieure. Le principe de ces méthodes réside dans la transformation d'un problème contraint en une séquence de problèmes sans contraintes, en ajoutant au coût une pénalité. L'ajout de cette fonction a pour seul but de pénaliser le coût en cas de violation d'une ou de plusieurs contraintes. La méthode de pénalité extérieure cherche à approcher le minimum depuis l'extérieur de domaine admissible et dans certains problèmes, le fait de ne pas être admissibles peut constituer un inconvénient, parce que la fonction objectif n'est pas définie à l'extérieur de domaine admissible. C'est ce qui a conduit Fiacco et McCormick (1968) à proposer d'autres méthodes de pénalisation dans lesquelles l'optimum est approché par l'intérieur c'est ce que on appelle les méthodes de pénalités intérieures. L'implémentation de ces méthode conduit a conclure que la méthode de pénalité extérieure est très similaire à celle de pénalité intérieure. En particulier, des résultats théoriques équivalents peuvent être établis et assurer sa convergence. La seule différence notable étant que la méthode de pénalité extérieure ne nécessite aucunement un point initial intérieur du domaine admissible.

Ce sujet passionnant n'a bien sûr pas été épuisé à travers ce projet et de nombreux défis restent ouverts, en particulier l'implémentation de nouvelles méthodes et d'extensions aux modèles présentés, afin de juger des gains en terme de performance qui pourraient en résulter, ou de tenter d'élargir leurs champs d'application.

Bibliographie

- [1]: Convexité et Optimisation.
Guy Cohen, École Nationale des Ponts et Chaussées et Inria, Edition (2000-2006)
- [2]: Introduction à l'Optimisation : aspects Théoriques, Numériques et Algorithmes,
Xavier Antoine, Pierre Dreyfuss et Yannick Privat, ENSMN-ENSEM 2A (2006-2007).
- [3]: Programmation Mathématique Théorie et Algorithmes .
Michel Minoux, Collection Technique et Scientifique des Télécommunications, Volume1. Année 1983
- [4]: ABADIE J. (1967), On the Kuhn-Tucker Theorem,
Nonlinear Programming, (J. Abadie éd.), North Holland, Amsterdam.
- [5]: KARLIN S. (1959), Mathematical methods and theory in games, programming and economics, Vol. 1, Addison Wesley.
- [6]: SLATER M. (1950), Lagrange Multipliers Revisited: A Contribution to Nonlinear Programming, Cowles Commission Discussion Paper, Mathematics 403.
- [7]: FIACCO A. V., McCoRMicK G. P. (1968), Nonlinear programming, John Wiley, New York.
- [8]: KUHN W. W., TUCKER A. W. (1951), Nonlinear programming, Proc, 2nd Berkeley Symp. on Mathematical Statistics and Probability, University of Cali-

- a fornia Press, Berkeley.
- [9]: Introduction à l'Optimisation Numérique,
a Aude Rondepierre et Sébastien Tordeux, Année 2009-2010.
- [10]: ARMIJO L. (1966), Minimization of functions having continuons partial a
 derivatives, Pacific J. Math., vol. 16.
- [11]: WOLFE P. (1969), Convergence conditions for ascent methods,S.I.A.M. Review
a 11.
- [12]: Éléments d'Optimisation Différentiable, Théorie et Algorithmes Jean Charles a
 Gilbert, 29 janvier 2010.
- [13]: Optimisation , Nicolas Petit
a École Nationale Supérieure des Mines de Paris, Année scolaire (2007-2008).
- [14]: GILL P. E., MURRAY W. (1972), Quasi-Newton Methods for Unconstrained a
 Optimization, J. Institute ofMath. and Its Appl. 9.
- [15]: HESTENES M. R., STIEFEL E. (1952), Methods of conjugate gradients for a
 solving linear Systems, J. Res. Nat. Bureau of Standards 49.
- [16]: Introduction à l'Optimisation Différentiable,
a Michel Bierlaire, Presses polytechniques et Universitaires Romandes. Année(2006)
- [18]: Méthodes Numériques, Algorithmes, Analyse, et Applications, Alfio quarteroni, a
 Riccardo sacco,Fausto saleri edition Springer-Verlag Italia, Milano 2007.
- [19]: FLETCHER R., REEVES C. M. (1964), Function minimization by conjugale a
 gradients, Computer Journal 7.
- [20]: POLAK E. (1971), Computational methods in optimisation, a unified approach, a
 Académie Press, New York.

- [21]: A Simplex Method for Function Minimization,
a J. A. Nelder, R. Mead, Computer Journal, 1965, Vol. 7
- [22]: C. T. Kelley, 1999. Detection and Remediation of Stagnation in the Nelder-
a Mead. Algorithm using a sufficient decrease condition.
SIAM journal on optimization. Vol. 1, <http://epubs.siam.org/sam-bin/dbq/article/31520>.
- [23]: Wolfe.P(1959) The simplex method for quadratic programming, Econometrica
a programming.
- [24]: Wild Beightler, Phillips. Foundations of optimization, 2nd ed .
a Englewood Cliffs, 1979.
- [25]: COURANT R. (1943), Variational methods for the solution of problems of a
equilibrium and vibrations, Bull. Amer. Math. Soc.
- [26]: ABLOW C. M., BRIGHAM G. (1955), An Analog Solution of Programming a
Problems, Opérations Research.
- [27]: MCCORMICK K.I., (1998), Convergence of the Nelder-Mead simplex method
a to nonstationary point, siopt

