

*République Algérienne Démocratique et Populaire*

*Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique*

*Ecole Normale Supérieure d'Enseignement Technologique -Skikda-*

Département de Mathématiques



قسم الرياضيات

*Mémoire de fin d'étude  
En vue de l'obtention du diplôme Professeur d'Enseignement Moyenne*

---

# *Les méthodes numériques d'optimisation sans contraintes*

---

*Présenté Par :*

- *Rezaiguia Douaa*
- *Bourki Amel*

*Dirigé par:*

- *Bentimama Wiam*

*Soutenu devant le jury composé de*

- |                              |                                 |           |
|------------------------------|---------------------------------|-----------|
| • <i>Derdar Najem Eddine</i> | Ecole Supérieure d'Enseignement | Président |
| • <i>Bentimama Wiam</i>      | Ecole Supérieure d'Enseignement | Encadreur |
| • <i>Merabet Farida</i>      | Ecole Supérieure d'Enseignement | Examineur |
| • <i>Ghoumrani Sarra</i>     | Ecole Supérieure d'Enseignement | Examineur |

**Année Universitaire: 2023-2024**

# Remerciements

Nous remercions Dieu le Tout-Puissant de nous avoir accordé la patience d'atteindre les sommets.

Nous tenons à adresser notre remerciements les plus chaleureux et notre profonde gratitude à notre encadreur madame bentimama wiam, Maitre-assistant à ENSET Skikda, pour nous avoir proposé le sujet de ce mémoire. C'est garce à sa grande disponibilité, ses conseils, ses orientations, et ses encouragements que nous ayons pu mener bien ce travail.

Nous remercions les membres du jury d'avoir accepté d'évaluer notre mémoire, Monsieur Dardar Nadjm Eddine, Madame Merabet Farida, Madame Ghomrani Sara...

Si nous avons réussi, c'est grâce à Dieu, et si nous avons commis des erreurs, c'est de notre propre fait et de Satan.

# TABLE DES MATIÈRES

<b>1</b>	<b>Préliminaires : optimisation sans contraintes</b>	<b>7</b>
1.1	Notations	7
1.1.1	Gradient	7
1.1.2	Dérivée directionnelle	8
1.2	Les Matrices	8
1.3	Éléments De Topologie	9
1.3.1	Boule ouverte	9
1.3.2	Boule fermée	9
1.3.3	Ensembles ouverts, Ensembles fermés	9
1.3.4	les voisinages	10
1.3.5	La continuité	10
1.4	Développement de Taylor	11
1.5	La différentiabilité	11
1.6	Ensembles Convexes	11
1.7	Fonctions Convexes	12
1.7.1	Fonctions convexes et différentiables	14
1.8	Minimisation sans contraintes	15
1.8.1	Convergence des algorithmes	16

1.8.2	Résultats d'existence et d'unicité	17
1.8.3	Conditions d'optimalités	18
<b>2</b>	<b>Optimisation Unidimensionnelle Ou Recherche Linéaire</b>	<b>21</b>
2.1	Principe de méthode de descente	21
2.2	Recherches linéaires	22
2.3	Recherches linéaires exactes	23
2.3.1	L'intervalle de sécurité	23
2.3.2	Méthode d'optimisation unidimensionnelle sans dérivées	24
2.4	Recherches linéaires inexactes ou économiques	26
2.4.1	Recherche linéaire inexacte d'Armijoo	27
2.4.2	Recherche linéaire inexacte de Goldstein et Price	29
2.4.3	Recherche linéaire inexacte de Wolfe	30
2.5	Convergence des méthodes utilisant des recherches linéaire inexactes et des di- rections de descente. Le théorème de Zoutendijk	33
2.5.1	Condition de Zoutendijk	33
<b>3</b>	<b>Méthodes Numériques d'Optimisation Sans Contraintes Et Leurs Algorithmes</b>	<b>36</b>
3.1	Méthode du Gradient Conjugué	36
3.1.1	Méthode des directions conjuguées	37
3.1.2	Convergence de la méthode des direction conjugués	39
3.1.3	Méthode du gradient conjugué	40
3.1.4	Algorithme de la méthode du gradient conjugué	41
3.1.5	Convergence de la méthode du gradient conjugué	42
3.2	Méthode de la plus forte pente	43
3.2.1	Algorithme de la méthode de la plus forte pente	45
3.2.2	Quelques remèdes	46
3.2.3	Convergence de la méthode de la plus forte pente	47
3.3	Méthodes quasi-Newtoniennes	48
3.3.1	Méthode de Newton	48
3.3.2	Méthodes de Quasi-Newton	50
3.3.3	Méthode de correction de rang un (SR1 [1960])	53
3.3.4	Méthode de Davidon-Fletcher-Powell (DFP)	57
3.3.5	Méthode de Broyden-Fletcher-Goldfard-Shanno (BFGS [1970])	64
3.3.6	Méthodes de la classe Broyden	69



# INTRODUCTION

On s'intéresse dans ce mémoire aux problèmes du type suivant : "trouver le minimum d'une fonction réelle  $f$  de  $n$  variables réelles  $x_1, x_2, \dots, x_n$  chacune de ces variables pouvant prendre n'importe quelle valeur de  $-\infty$  à  $+\infty$ ".

D'un point de vue mathématique, le problème se formule de la façon suivante :

$$\min\{f(x), x \in \mathbb{R}^n\} \quad (P)$$

L'optimisation non linéaire sans contraintes est souvent rencontrée dans des domaines variés, et l'étude des algorithmes et méthodes qui traitent ces problèmes, est importante pour plusieurs raisons :

1. Parfois pour résoudre un problème avec contraintes, on le remplace par une suite de problèmes sans contraintes, comme c'est le cas des méthodes des Pénalités.
2. Plusieurs techniques d'optimisation sans contraintes peuvent être prolongées d'une façon naturelle pour fournir et motiver des procédures pour résoudre des problèmes avec contraintes. Les méthodes que nous utilisons (gradient conjugué, la plus forte pente et quasi-Newtonienne) sont surtout utilisées pour les problèmes de grande taille.
  - La méthode du gradient conjugué a été découverte en 1952 par Hestenes et Steifel ([28]) pour la minimisation des fonctions quadratiques strictement convexes.
  - La méthode de la plus forte pente a été découverte en 1847 par Cauchy. Elle est plus connue sous le nom de la plus profonde descente.

- La méthode quasi-Newtonienne n'a pas été attribuée à une seule personne, mais s'est développée grâce aux efforts d'un groupe de scientifiques et de chercheurs au fil des ans. Cependant, il y a quelques noms notables qui ont grandement contribué au développement de cette méthode : Broyden ([27]), Davidon ([25]), Fiacco et McCormick ([28]), Murtagh et Sargent ([29]), Wolfe ([30]), Fletcher et Powell, Goldfarb, et Shanno.

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  et (P) le problème de minimisation non linéaire, sans contraintes.

On s'intéresse dans ce mémoire à résoudre le problème (P) par la méthode du gradient conjugué, la méthode de la plus forte pente, et la méthode quasi-Newtonienne avec des recherches linéaires inexactes.

Ce mémoire est composé de :

**Premier chapitre :** Préliminaires : Optimisation sans contraintes.

Ce chapitre est un rappel de quelques notions préliminaires, on insiste surtout sur les conditions d'optimalité des problèmes de minimisation sans contrainte.

**Deuxième chapitre :** Optimisation Unidimensionnelle ou recherche linéaire.

Ce chapitre est consacré aux grandes lignes des méthodes d'optimisation sans contraintes basées sur les directions de descente et les recherches linéaires inexactes célèbres (Armijo, Goldstein et Wolfe).

**Troisième chapitre :** Les méthodes numériques d'optimisation sans contraintes et leurs algorithmes.

Dans ce chapitre, on va d'abord parler de la méthode du gradient conjugué. Cette méthode permet de trouver une suite de directions de descente pour minimiser une fonction quadratique et non quadratique de  $\mathbb{R}^n$  dans  $\mathbb{R}$ . Puis on va parler de la méthode de la plus forte pente, l'une des anciennes méthodes les plus utilisées pour résoudre le problème (P). Ensuite, on va parler de la méthode quasi-newtonienne qui est basée sur la méthode de Newton, méthode de quasi-Newton, méthode de correction de rang un (SR1 [1960]), méthode de Davidon-Fletcher-Powell (DFP), méthode de Broyden-Fletcher-Goldfard-Shanno (BFGS [1970]), Les méthodes de la classe Broyden.

# CHAPITRE

## 1

# PRÉLIMINAIRES : OPTIMISATION SANS CONTRAINTES

Dans ce chapitre, on définit et on introduit les outils fonctionnels de base nécessaires pour l'optimisation sans contraintes.

## 1.1 Notations

On se place dans  $\mathbb{R}^n$ ,  $n > 1$ , considéré comme un espace vectoriel normé muni de la norme euclidienne notée  $\|\cdot\|$ .

### 1.1.1 Gradient

**Définition 1.1.1** *on note par :*

$$(\nabla f(x))^T = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)(x)$$



**Remarque 1.2.1**  $H(\hat{x})$  est dite définie positive en  $\hat{x}$

$$\forall x \in \mathbb{R}^n, x \neq 0 : x^T H(\hat{x})x > 0$$

## 1.3 Éléments De Topologie

### 1.3.1 Boule ouverte

**Définition 1.3.1** On appelle boule ouverte de centre  $x_0 \in \mathbb{R}^n$  et de rayon  $r > 0$  le sous-ensemble de  $\mathbb{R}^n$  définie par :

$$B_0(x_0, r) = \{x \in \mathbb{R}^n : \|x - x_0\| < r\}$$

### 1.3.2 Boule fermée

**Définition 1.3.2** On appelle boule fermée de centre  $x_0 \in \mathbb{R}^n$  et de rayon  $r > 0$  le sous-ensemble de  $\mathbb{R}^n$  définie par :

$$B_f(x_0, r) = \{x \in \mathbb{R}^n : \|x - x_0\| \leq r\}$$

### 1.3.3 Ensembles ouverts, Ensembles fermés

Soit  $S \subset \mathbb{R}^n$  un sous-ensemble de  $\mathbb{R}^n$

**Définition 1.3.3 (Intérieur)**  $y \in S$  est un point intérieur de  $S$  s'il existe  $\varepsilon$  tel que :

$$\|x - y\| < \varepsilon \Rightarrow x \in S$$

(autrement dit, s'il existe une boule centrée en  $y$  et contenue dans  $S$ ) L'ensemble des points intérieurs à  $S$  est appelé l'intérieur de  $S$  et est noté  $\text{int}(S)$

**Exemple 1.3.1** L'ensemble  $A = \{x / \|x\| \leq 1\}$  a pour intérieur la boule unité  $B = \{x / \|x\| < 1\}$ . L'intérieur d'un ensemble peut être vide, c'est le cas par exemple d'un hyperplan dans  $\mathbb{R}^n$ .

**Définition 1.3.4 (Sous-ensembles ouverts)** Un sous-ensemble  $S$  est dit ouvert s'il coïncide avec son intérieur, c'est-à-dire si  $S = \text{int}(S)$ .

**Exemple 1.3.2** La boule unité  $\{x / \|x\| < 1\}$  est un ouvert.

L'ensemble vide  $\emptyset$  et l'ensemble  $\mathbb{R}^n$  tout entier sont des ouverts.

**Propriété 1.3.1** L'intersection d'un nombre fini d'ensembles ouverts est un ouvert.

La réunion d'un nombre fini ou infini d'ensembles ouverts est un ouvert.

**Définition 1.3.5 (point d'adhérence, adhérence d'un sous-ensemble)** On dit que  $x \in \mathbb{R}^n$  est un point d'adhérence d'un sous-ensemble  $S$ , si tout voisinage de  $x$  rencontre  $S$ . Autrement dit,  $\forall \varepsilon > 0$ , il existe  $y \in S$  tel que  $\|x - y\| < \varepsilon$ .

L'ensemble de tous les points d'adhérence de  $S$  est appelé l'adhérence de  $S$  et est noté  $cl(S)$ .

On a évidemment  $cl(S) \supset S$

**Exemple 1.3.3** L'adhérence de la boule unité  $B = \{x / \|x\| < 1\}$  est l'ensemble  $cl(B) = \{x / \|x\| \leq 1\}$

**Définition 1.3.6 (sous-ensemble fermé)** Un sous-ensemble  $S \subset \mathbb{R}^n$  est dit fermé s'il coïncide avec son adhérence c'est-à-dire si  $cl(S) = S$ .

### 1.3.4 les voisinages

**Définition 1.3.7 (Voisinage)** Étant donné  $x \in \mathbb{R}^n$ ,  $\mathcal{V}(x)$  est un voisinage de  $x$  si c'est un sous-ensemble ouvert contenant  $x$ . De façon équivalente,  $\mathcal{V}(x)$  est un voisinage de  $x$  s'il contient une boule de centre  $x$ .

### 1.3.5 La continuité

Soient  $\mathcal{S}$  un sous-ensemble, et  $f : \mathcal{S} \rightarrow \mathbb{R}$  une fonction définie sur  $\mathcal{S}$  à valeurs réelles et  $a \in \mathcal{S}$ . La fonction  $f$  est dite continue en  $a$  si :

$$\forall \varepsilon > 0 \quad \exists \eta > 0 \quad \forall x \in \mathcal{S} \quad (|x - a| < \eta \Rightarrow |f(x) - f(a)| < \varepsilon).$$

ce qui signifie que :

$$\lim_{x \rightarrow a} f(x) = f(a)$$

**Remarque 1.3.1** • si la continuité est valable uniquement à droite (pour  $x > a$ ), on dit que  $f$  est continue à droite en  $a$ . De même à gauche pour  $a$ .

• La fonction  $f$  est dite continue (sur  $\mathcal{S}$ ) si elle est continue en tout point  $a$  de  $\mathcal{S}$

## 1.4 Développement de Taylor

Soit  $D$  de  $\mathbb{R}^n$  ouvert, et  $f : D \rightarrow \mathbb{R}^n$  alors :

1. si  $f \in C^1(D)$  alors la formule de Taylor - Young l'ordre 1 :

$$f(a+h) = f(a) + \langle \nabla f(a), h \rangle + o(\|h\|), \quad \lim_{h \rightarrow 0} o(\|h\|) = 0$$

2. Si  $f \in C^2(D)$  alors la formule de Taylor - Young l'ordre 2 :

$$f(a+h) = f(a) + \langle \nabla f(a), h \rangle + \frac{1}{2} \langle \nabla^2 f(a)h, h \rangle + o(\|h\|^2)$$

$$\lim_{h \rightarrow 0} o(\|h\|^2) = 0$$

## 1.5 La différentiabilité

**Définition 1.5.1** On dit qu'une fonction  $f$  est différentiable en un point  $x_0 \in \mathbb{R}^n$ ,  $h \in \mathbb{R}^n$  s'il existe une application linéaire continue  $\mathcal{L}$  telle que :

$$f(x_0 + h) = f(x_0) + \mathcal{L}(h) + \|h\|\varepsilon(h)$$

où :

$$\lim_{\|h\| \rightarrow 0} \varepsilon(h) = 0$$

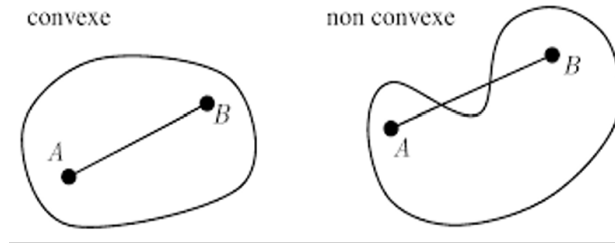
$$\mathcal{L}(h) = \nabla^t f(x_0)h$$

## 1.6 Ensembles Convexes

**Définition 1.6.1** 1) Soit  $\mathcal{S}$  un sous ensemble de  $\mathbb{R}^n$ .  $\mathcal{S}$  est dit convexe si et seulement si

$$\forall x_1, x_2 \in \mathcal{S}, \forall \lambda \in [0, 1], \lambda x_1 + (1 - \lambda)x_2 \in \mathcal{S}.$$

2) Soit  $x_1, x_2, \dots, x_k \in \mathbb{R}^n$ ,  $\lambda_j \geq 0$  et  $\sum_{j=1}^k \lambda_j = 1$ . Toute expression de la forme suivante :  $\sum_{j=1}^k \lambda_j x_j$  s'appelle combinaison convexe des points  $x_j$  ou barycentre.



## 1.7 Fonctions Convexes

**Définition 1.7.1 (fonction convexe)** Soit  $f : \mathcal{S} \rightarrow \mathbb{R}$ , où  $\mathcal{S}$  est convexe non vide de  $\mathbb{R}^n$ .  $f$  est dite convexe dans  $\mathcal{S}$  si et seulement si

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

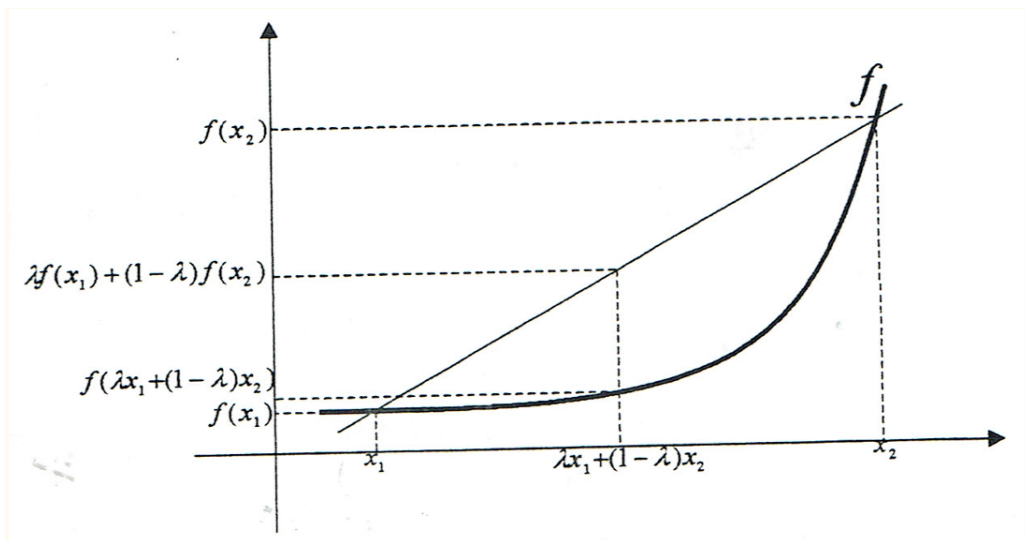
pour tout  $x, y \in \mathcal{S}$  et pour  $\lambda \in [0, 1]$

$f$  est dite strictement convexe dans  $\mathcal{S}$  si et seulement si

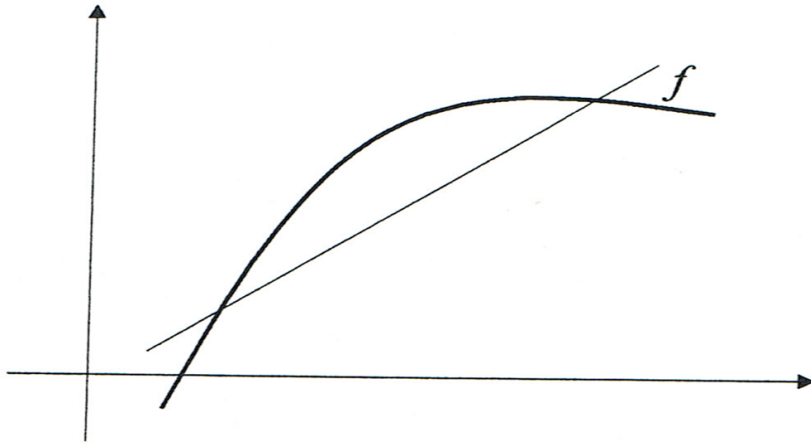
$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y)$$

pour tout  $x, y \in \mathcal{S}$  telle que  $x \neq y$  et pour  $\lambda \in ]0, 1[$

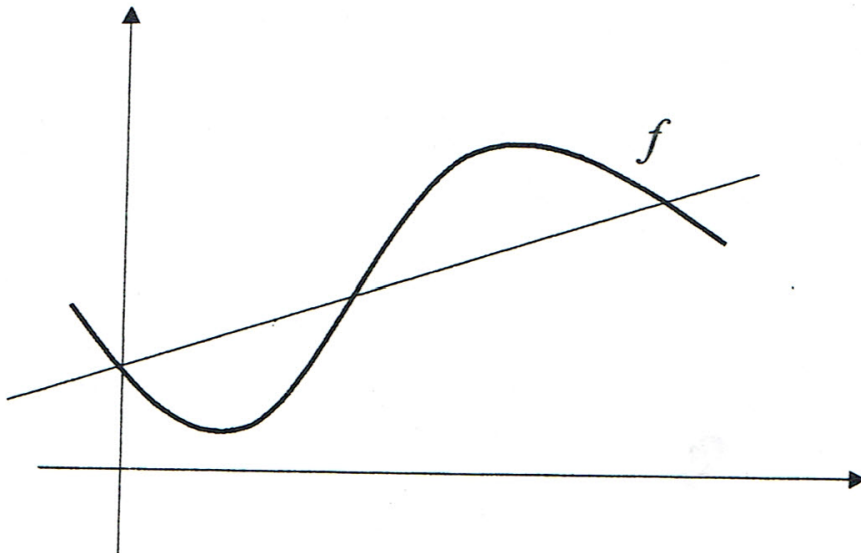
$f$  est dite concave (strictement concave) dans  $\mathcal{S}$  si et seulement si  $(-f)$  est convexe (strictement convexe) dans  $\mathcal{S}$ .



Fonction Convexe



*Fonction Concave*



*Fonction ni convexe ni concave*

**Proposition 1.7.1** soit  $S \subset \mathbb{R}^n$  convexe non vide,  $f : S \rightarrow \mathbb{R}$  convexe et  $\alpha \in \mathbb{R}$ , alors l'ensemble

$$S_\alpha = \{x \in S : f(x) \leq \alpha\}$$

est convexe.

**démonstration 1.7.1** Soient  $x, y \in S_\alpha \Leftrightarrow f(x) \leq \alpha$  et  $f(y) \leq \alpha$ . Démontrons que pour  $\lambda \in [0, 1]$ ,

$\lambda x + (1 - \lambda)y \in \mathcal{S}_\alpha$ .

En effet

$$\begin{aligned} f(\lambda x + (1 - \lambda)y) &\leq \lambda f(x) + (1 - \lambda)f(y) \\ &\leq \alpha\lambda + (1 - \lambda)\alpha = \alpha \end{aligned}$$

**Définition 1.7.2 (fonction quasi-convexe)** Une fonction  $f : \mathcal{S} \rightarrow \mathbb{R}$  définie sur une partie convexe  $\mathcal{S}$  d'un espace vectoriel réel  $E$  est dite :

- quasi convexe si pour tout réel  $a$ , l'ensemble de sous-niveau  $\{x \in \mathcal{S} \mid f(x) < a\}$  est convexe, ce qui est équivalent

$$\forall x, y \in \mathcal{S} \quad \forall z \in [x, y] \quad f(z) \leq \max(f(x), f(y))$$

- strictement quasi-convexe si l'on a même :

$$\forall x, y \in \mathcal{S} \quad \forall z \in ]x, y[ \quad f(z) < \max(f(x), f(y))$$

**Théorème 1.7.1 (continuité des fonctions convexes)** Soient  $\mathcal{S} \subset \mathbb{R}^n$  convexe non vide et  $f : \mathcal{S} \rightarrow \mathbb{R}$  convexe. Alors  $f$  est continue dans  $\text{int}(\mathcal{S})$ .

## 1.7.1 Fonctions convexes et différentiables

### Fonctions convexes une fois différentiables

**Théorème 1.7.2** Soit  $\mathcal{S} \subset \mathbb{R}^n$  convexe, non vide, ouvert et  $f : \mathcal{S} \rightarrow \mathbb{R}$  différentiable dans  $\mathcal{S}$ . Alors :

a)  $f$  est convexe dans  $\mathcal{S}$  si et seulement si pour tout  $\hat{x} \in \mathcal{S}$

$$f(x) \geq f(\hat{x}) + \nabla f(\hat{x})^T(x - \hat{x}), \quad \forall x \in \mathcal{S} \quad (1.1)$$

b) De même  $f$  est strictement convexe dans  $\mathcal{S}$  si et seulement si pour tout  $\hat{x} \in \mathcal{S}$

$$f(x) > f(\hat{x}) + \nabla f(\hat{x})^T(x - \hat{x}), \quad \forall x \in \mathcal{S}, x \neq \hat{x} \quad (1.2)$$

## Fonctions convexes deux fois différentiables

**Théorème 1.7.3** Soit  $\mathcal{S} \subset \mathbb{R}^n$  non vide, convexe et ouvert et  $f : \mathcal{S} \rightarrow \mathbb{R}$  une fonction deux fois différentiable dans  $\mathcal{S}$ . Alors  $f$  est convexe dans  $\mathcal{S}$  si et seulement si la matrice Hessienne de  $f$  est semi-définie positive en tout point  $\hat{x} \in \mathcal{S}$

## 1.8 Minimisation sans contraintes

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  on appelle problème de minimisation sans contraintes le problème suivant :

$$(p) \left\{ \min_{x \in \mathbb{R}^n} f(x) \right.$$

L'étude de ces problèmes est important pour des raisons diverses.

**Définition 1.8.1** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , une fonction continûment différentiable.

a) Soit  $\hat{x} \in \mathbb{R}^n$  est dite solution optimale globale de (P) si et seulement si

$$\forall x \in \mathbb{R}^n; f(\hat{x}) \leq f(x)$$

Lorsque l'inégalité stricte :  $f(\hat{x}) < f(x)$  est vérifiée  $\forall x \in \mathbb{R}^n, x \neq \hat{x}$ , le minimum global  $\hat{x}$  est unique.

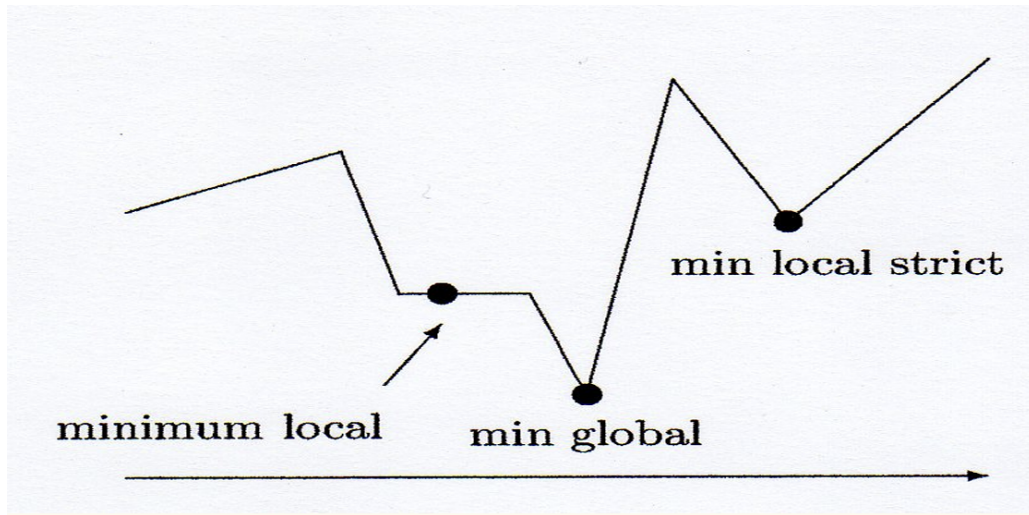
Néanmoins, il existe aussi bien des situations dans lesquelles le minimum global n'est pas unique.

b) Soit  $\hat{x} \in \mathbb{R}^n$  est dite solution optimale locale de (P) si et seulement s'il existe un voisinage  $v_\epsilon(\hat{x})$  de  $\hat{x}$  tel que :

$$f(\hat{x}) \leq f(x), \quad \forall x \in v_\epsilon(\hat{x})$$

c) Soit  $\hat{x} \in \mathbb{R}^n$  est dite solution optimale locale strict de (P) si et seulement s'il existe un voisinage  $v_\epsilon(\hat{x})$  de  $\hat{x}$  tel que :

$$f(\hat{x}) < f(x), \quad \forall x \in v_\epsilon(\hat{x}), x \neq \hat{x}$$



*solution optimale globale, locale, locale strict*

### 1.8.1 Convergence des algorithmes

#### Algorithme en optimisation

*Un algorithme de résolution est un procédé qui permet, à partir de la donnée d'un point initial  $x_0$ , d'engendrer la suite  $x_0, x_1, \dots, x_k$  un algorithme est parfaitement défini par la donnée de l'application  $A$  qui à  $x_k$  associe.  $x_{k+1} = A(x_k)$  ceci permettra de confondre un algorithme et l'application  $A$  qui lui est associée.*

#### Fonction multivoque

*Une application multivoque est une application  $A$  qui à  $x \in \mathbb{R}^n$  fait correspondre un sous ensemble  $A(x)$  de  $\mathbb{R}^n$ . Etant donné un point  $x_k$ , en point  $x_{k+1}$ . Cette procédure peut être décrite par une application multivoque  $A$  appelée application algorithmique.*

#### Convergence globale

*Nous dirons qu'un algorithme décrit par une application multivoque  $A$ , est globalement convergent si quelque soit le point de départ  $x_0$  choisi, la suite  $\{x_k\}$  engendrée par  $x_{k+1} \in A(x_k)$  (ou une sous suite) converge vers un point satisfaisant les conditions nécessaires d'optimalité (ou solution optimale).*

## Les modes de convergence

Soit  $\{x_k\}_{k \in \mathbb{N}}$  une suite dans  $\mathbb{R}^n$  convergente vers  $\hat{x}$ .

Si :

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - \hat{x}\|}{\|x_k - \hat{x}\|} = \alpha < 1$$

On dit que  $\{x_k\}_{k \in \mathbb{N}}$  converge vers  $\hat{x}$  linéairement avec le taux  $\alpha$ .

Si :

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - \hat{x}\|}{\|x_k - \hat{x}\|} = 0$$

On dit que la convergence est superlinéaire. Plus précisément si  $\exists \gamma > 1$  tel que :

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - \hat{x}\|}{\|x_k - \hat{x}\|^\gamma} < \infty$$

On dit que la convergence est superlinéaire d'ordre  $\gamma$ .

En particulier si :

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - \hat{x}\|}{\|x_k - \hat{x}\|^2} < \infty$$

On dit que la convergence est quadratique (superlinéaire d'ordre 2).

### 1.8.2 Résultats d'existence et d'unicité

Avant d'étudier les propriétés de la solution (ou des solutions) de (P) il faut s'assurer de leur existence.

**Définition 1.8.2** On dit que  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est coercive si

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

**Théorème 1.8.1 (existence)** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , continue et coercive, Alors (P) admet au moins une solution.

**démonstration 1.8.1** Soit  $x_0 \in \mathbb{R}^n$ ,  $f$  coercive donc

$$\exists M > 0; \|x\| > M \Rightarrow f(x) > f(x_0)$$

Donc :

$$\exists M > 0; f(x) \leq f(x_0) \Rightarrow \|x\| \leq M$$

Puisque  $\hat{x}$  est caractérisé par  $f(x) \leq f(\hat{x}), \forall x \in \mathbb{R}^n$ , on a donc forcément  $\|\hat{x}\| \leq M$ . Donc  $\hat{x}$  est solution du problème.

**Théorème 1.8.2 (Unicité)** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  strictement convexe. Alors le problème (P) admet au plus une solution.

**démonstration 1.8.2** Supposons que  $f$  admette au moins un minimum  $m$  et soit  $x_1 \neq x_2$  dans  $\mathbb{R}^n$  réalisant ce minimum :

$$f(x_1) = f(x_2) = m.$$

Par stricte convexité de la fonction  $f$  on a alors :

$$f\left(\frac{x_1 + x_2}{2}\right) < \frac{1}{2}(f(x_1) + f(x_2)) = m$$

ceci contredit le fait que  $m$  est le minimum, donc  $x_1 = x_2$ .

### 1.8.3 Conditions d'optimalités

La question est : comment reconnaître un optimum ? Il y a des conditions nécessaires et des conditions suffisantes bien connues

#### Conditions nécessaires d'optimalité

Étant donné un point  $\hat{x}$ , la propriété de différentiabilité continue de la fonction  $f$  fournit une première manière de caractériser une solution optimale.

#### Conditions nécessaires d'optimalité du premier ordre

**Théorème 1.8.3** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  telle que  $f$  soit différentiable au point  $\hat{x} \in \mathbb{R}^n$ .

Soit  $d \in \mathbb{R}^n$  telle que  $\nabla f(\hat{x})^t d < 0$ . Alors il existe  $\delta > 0$  tel que  $f(\hat{x} + \lambda d) < f(\hat{x})$  pour tout  $\lambda \in ]0, \delta[$

La direction  $d$  s'appelle dans ce cas direction de descente.

**démonstration 1.8.3** Comme  $f$  est différentiable en  $\hat{x} \in \mathbb{R}^n$  alors :

$$f(\hat{x} + \lambda d) = f(\hat{x}) + \lambda \nabla f(\hat{x})^t d + \lambda \|d\| \epsilon(\hat{x}; \lambda d)$$

Où

$$\lim_{\lambda \rightarrow 0} \epsilon(\hat{x}, \lambda d) = 0$$

Ceci implique :

$$\frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} = \nabla f(\hat{x})^t d + \|d\| \epsilon(\hat{x}; \lambda d); \quad \lambda \neq 0$$

et comme  $\nabla f(\hat{x})^t d < 0$  et  $\lim_{\lambda \rightarrow 0} \epsilon(\hat{x}; \lambda d) = 0$ , il existe  $\delta > 0$  :

$$\nabla f(\hat{x})^t d + \|d\| \lambda(\hat{x}; \alpha d) < 0, \quad \alpha \in ]0, \delta[$$

et par conséquent on obtient :

$$f(\hat{x} + \alpha d) < f(\hat{x}), \quad \forall \alpha \in ]0, \delta[$$

### Corollaire 1.1

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  différentiable au point  $\hat{x} \in \mathbb{R}^n$ . Si  $\hat{x}$  est un minimum local de (P) alors  $\nabla f(\hat{x}) = 0$ .

**démonstration** On démontre par l'absurde, on suppose que  $\nabla f(\hat{x}) \neq 0$ .

Si on suppose :  $d = -\nabla f(\hat{x})$

On obtient :

$$\nabla f(\hat{x})^t d = -\|\nabla f(\hat{x})\|^2 < 0$$

et par le théorème 1.8.3, il existe  $\delta > 0$  tel que :

$$f(\hat{x} + \alpha d) < f(\hat{x}), \quad \forall \alpha \in ]0, \delta[$$

ce qui donne une contradiction avec le fait que  $\hat{x}$  est un minimum local d'où :

$$\nabla f(\hat{x}) = 0$$

### Conditions nécessaires d'optimalité du second ordre

**Théorème 1.8.4** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois différentiable au point  $\hat{x} \in \mathbb{R}^n$ . Si  $\hat{x}$  est un minimum local de (P) alors  $\nabla f(\hat{x}) = 0$  et la matrice Hessienne de  $f$  au point  $\hat{x}$  est semi définie positive.

**démonstration 1.8.4** Soit  $d \in \mathbb{R}^n$  quelconque,  $f$  étant deux fois différentiable au point  $\hat{x}$ , on aura pour tout  $\alpha \neq 0$ .

$$f(\hat{x} + \alpha d) = f(\hat{x}) + \frac{1}{2} \alpha^2 d^t H(\hat{x}) d + \alpha^2 \|d\|^2 \lambda(\hat{x}; \alpha d)$$

Où :

$$\lim_{\alpha \rightarrow 0} \lambda(\hat{x}; \alpha d) = 0$$

Ceci implique :

$$\frac{f(\hat{x} + \alpha d) - f(\hat{x})}{\alpha^2} = \frac{1}{2} d^t H(\hat{x}) d + \alpha \|d\|^2 \lambda(\hat{x}; \alpha d)$$

ainsi  $\hat{x}$  est un optimum local, il existe alors  $\delta > 0$  tel que :

$$\frac{f(\hat{x} + \alpha d) - f(\hat{x})}{\alpha^2} \geq 0; \quad \alpha \in ]0, \delta[$$

Comme  $\hat{x}$  est un minimum local alors  $f(\hat{x} + \alpha d) \geq f(\hat{x})$  pour  $\alpha$  suffisamment petit.

En passant la limite quand  $\alpha \rightarrow 0$  on obtient que  $d^t H(\hat{x}) d \geq 0$  est semi définie positive.

### Conditions suffisantes d'optimalité

Les conditions données précédemment sont nécessaires (si  $f$  n'est pas convexe), c'est-à-dire quelle doivent être satisfaites pour tout minimum local, cependant, tout point vérifiant ces conditions n'est pas nécessairement un minimum local.

Le théorème suivant établit une condition suffisante pour qu'un point soit un minimum local, si  $f$  deux fois différentiable.

**Théorème 1.8.5** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois différentiable au point  $\hat{x} \in \mathbb{R}^n$ . Si  $\nabla f(\hat{x}) = 0$  et  $H(\hat{x})$  est définie positive alors  $\hat{x}$  est un minimum local strict de  $(P)$ .

### Cas convexe

**Théorème 1.8.6** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  telle que  $f$  est convexe et différentiable. Alors  $\hat{x}$  est un minimum global de  $f$  si et seulement si :  $\nabla f(\hat{x}) = 0$ .

**Remarque 1.8.1** Dans le cas où  $f$  est convexe, alors tout minimum local est aussi globale. De plus si  $f$  est strictement convexe, alors tout minimum local devient non seulement global mais aussi unique.

## CHAPITRE

### 2

# OPTIMISATION UNIDIMENSIONNELLE OU RECHERCHE LINÉAIRE

*L'optimisation unidimensionnelle (recherche linéaire) consiste à trouver  $\lambda_k$  de façon à diminuer la fonction  $f$  Suffisamment le long de cette direction.*

*Ce " Suffisamment " sera quantifié dans la suite dans la description des conditions dites d'Armijo, Wolfe, Goldstein et Price (recherches linéaires inexactes).*

*Mais d'abord on expose le principe de méthode de descente :*

### 2.1 Principe de méthode de descente

*Le principe d'une méthode de descente consiste à faire les itérations suivantes :*

$$x_{k+1} = x_k + \lambda_k d_k, \quad k > 0$$

*tout en assurant la propriété*

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

Le vecteur  $d_k$  est la direction de descente en  $x_k$ . Le scalaire  $\lambda_k$  est appelé le pas de la méthode à l'itération  $k$ .

On peut caractériser les directions de descente en  $x_k$  à l'aide du gradient :

**Proposition 2.1.1** Soit  $d \in \mathbb{R}^n$  vérifiant

$$\nabla f(x)^t d < 0$$

alors  $d$  est une direction de descente en  $x$ .

**démonstration 2.1.1** on a pour  $\lambda > 0$

$$f(x + \lambda d) = f(x) + \lambda \nabla f(x)^t d + \lambda \epsilon(\lambda)$$

donc si on écrit

$$\frac{f(x + \lambda d) - f(x)}{\lambda} = \nabla f(x)^t d + \epsilon(\lambda)$$

on voit bien que pour  $\lambda$  suffisamment petit on aura

$$f(x + \lambda d) - f(x) < 0.$$

Ou encore que  $d$  fait avec l'opposé du gradient  $-\nabla f(x)$  un angle strictement plus petit que  $90^\circ$  :

$$\theta := \arccos \frac{-\nabla f(x)^t d}{\|\nabla f(x)\| \|d\|} \in ]0, \frac{\pi}{2}[$$

L'ensemble des directions de descente de  $f$  en  $x$

$$\{d \in \mathbb{R}^n : \nabla f(x)^t d < 0\}$$

forme un demi-espace ouvert de  $\mathbb{R}^n$

## 2.2 Recherches linéaires

Considérons le problème d'optimisation sans contraintes  $(P)$  :

$$(P) : \min_{x \in \mathbb{R}^n} f(x) \text{ où } f : \mathbb{R}^n \rightarrow \mathbb{R}$$

Au point  $x_k$ , on se donne une direction  $d_k$  (en général de descente). Le successeur de  $x_k$  est donné par :

$$x_{k+1} = x_k + \lambda_k d_k$$

où  $\lambda_k$  est une solution optimale du problème suivant :

$$f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k), \quad \lambda \geq 0$$

Considérons la fonction  $\varphi(\lambda)$  :

$$\varphi(\lambda) = f(x_k + \lambda d_k), \quad \lambda \geq 0$$

Chercher  $\lambda_k$  revient à trouver le  $\min_{\lambda \geq 0} \varphi(\lambda) = \varphi(\lambda_k) = f(x_k + \lambda_k d_k)$ .

## 2.3 Recherches linéaires exactes

Dans ce cas la solution optimale  $\lambda_k$  est calculée de façon exacte (d'un point de vue théorique car pratiquement on n'obtient en général qu'une approximation). On donnera l'algorithme de la recherche linéaire de dichotomie et du nombre d'or.

### 2.3.1 L'intervalle de sécurité

**Définition 2.3.1** Considérons le problème unidimensionnel suivant :

$$\text{Minimiser}_{\lambda \in [a, b]} \varphi(\lambda)$$

L'intervalle  $[a, b]$  est dit intervalle de sécurité si le minimum  $\bar{\lambda}$  de  $\varphi(\lambda)$  appartient à  $[a, b]$ , mais sa valeur exacte n'est pas connue.

**Théorème 2.3.1** Soit  $\varphi : \mathbb{R} \rightarrow \mathbb{R}$  strictement quasi-convexe sur  $[a, b]$ .

Soient  $\lambda, \mu \in [a, b]$  tels que  $\lambda < \mu$

1- si  $\varphi(\lambda) > \varphi(\mu)$ , alors  $\varphi(z) \geq \varphi(\mu)$ ,  $\forall z \in [a, \lambda[$ .

2- si  $\varphi(\lambda) \leq \varphi(\mu)$ , alors  $\varphi(z) \geq \varphi(\lambda)$ ,  $\forall z \in ]\mu, b]$ .

Supposons que  $\varphi(\lambda) > \varphi(\mu)$  et qu'il existe  $z_0 \in [a, \lambda[$  tel que  $\varphi(z_0) < \varphi(\mu)$ .

Alors  $\lambda$  est une combinaison linéaire convexe de  $z_0$  et  $\mu$ . La stricte convexité de  $\varphi$  donne :

$$\varphi(\lambda) < \max\{\varphi(z_0), \varphi(\mu)\} = \varphi(\mu)$$

*ceci contredit l'hypothèse.*

### Conséquence importante du théorème 2.3.1

- 1- Si  $\varphi(\lambda) > \varphi(\mu)$ , alors le nouvel intervalle d'incertitude est :  $[\lambda, b]$  (On supprime  $[a, \lambda[$ )
- 2- Si  $\varphi(\lambda) \leq \varphi(\mu)$ , alors le nouvel intervalle d'incertitude est :  $[a, \mu]$  (On supprime  $]\mu, b]$  )

### 2.3.2 Méthode d'optimisation unidimensionnelle sans dérivées

#### La méthode de dichotomie

##### Initialisation :

Choisir  $2\varepsilon > 0$  et choisir la longueur finale de l'intervalle d'incertitude  $l$ .  $[a_1, b_1]$  étant l'intervalle initial.

poser  $k = 1$  et aller à l'étape 1.

##### Etape 1 :

Si  $b_k - a_k < l$  STOP. Le minimum appartient à  $[a_k, b_k]$ .

Sinon poser :

$$\lambda_k = \frac{a_k + b_k}{2} - \varepsilon$$
$$\mu_k = \frac{a_k + b_k}{2} + \varepsilon$$

et aller à l'étape 2

##### Etape 2 :

Si  $\varphi(\lambda_k) > \varphi(\mu_k)$  poser  $a_{k+1} = a_k$ ,  $b_{k+1} = \mu_k$

Sinon  $a_{k+1} = \lambda_k$ ,  $b_{k+1} = b_k$ . Remplacer  $k$  par  $k + 1$  et aller à l'étape 1.

#### La méthode du nombre d'or (golden section)

La méthode du nombre d'or améliore (en diminuant le nombre d'observations) la méthode de dichotomie.

**Initialisation :**

A l'itération  $k$  supposons que l'on a l'intervalle  $[a_{k+1}, b_{k+1}]$  alors :

Si  $\varphi(\lambda_k) > \varphi(\mu_k) : [a_{k+1}, b_{k+1}] = [\lambda_k, b_k]$ .

Si  $\varphi(\lambda_k) \leq \varphi(\mu_k) : [a_{k+1}, b_{k+1}] = [a_k, \mu_k]$ .

**Choix des points  $\lambda_k$  et  $\mu_k$** 

La longueur de  $[a_{k+1}, b_{k+1}]$  ne dépend pas de l'itération, c'est à dire que dans les deux cas d'observations  $\varphi(\lambda_k) > \varphi(\mu_k)$  ou  $\varphi(\lambda_k) \leq \varphi(\mu_k)$  on a :

$$b_k - \lambda_k = \mu_k - a_k$$

Plus précisément si  $\lambda_k$  est de la forme

$$\lambda_k = a_k + (1 - \alpha)(b_k - a_k); \alpha \in ]0, 1[$$

Alors  $\mu_k$  s'écrit :

$$\mu_k = a_k + \alpha(b_k - a_k)$$

Ces deux relations donnent :

$$b_{k+1} - a_{k+1} = \alpha(b_k - a_k)$$

Le choix de  $\lambda_{k+1}$  et  $\mu_{k+1}$  se fait de façon que ou bien  $\lambda_{k+1}$  coïncide avec  $\mu_k$  ou bien  $\mu_{k+1}$  coïncide avec  $\lambda_k$ .

**Algorithme de la méthode du nombre d'or.****Etape initiale :**

Choisir  $l > 0$  la longueur finale de l'intervalle d'incertitude et  $[a_1, b_1]$  l'intervalle initial.  
Calculer :

$$\lambda_1 = a_1 + (1 - \alpha)(b_1 - a_1)$$

$$\mu_1 = a_1 + \alpha(b_1 - a_1)$$

$$\text{avec } \alpha = 0.618$$

Poser  $k = 1$  et aller à l'étape principale.

### Etape principale

1- Si  $b_k - a_k < l$  STOP. La solution optimale appartient à  $[a_k, b_k]$ .

Sinon si  $\varphi(\lambda_k) > \varphi(\mu_k)$  aller à 2, et si  $\varphi(\lambda_k) \leq \varphi(\mu_k)$  aller à 3.

2- Poser  $a_{k+1} = \lambda_k$  et  $b_{k+1} = b_k$ ,  $\lambda_{k+1} = \mu_k$  et  $\mu_{k+1} = a_{k+1} + \alpha(b_{k+1} - a_{k+1})$ .

Evaluer  $\varphi(\mu_{k+1})$  et aller à 4.

3- Poser  $a_{k+1} = a_k$  et  $b_{k+1} = \mu_k$  et  $\mu_{k+1} = \lambda_k$  et  $\lambda_{k+1} = a_{k+1} + (1 - \alpha)(b_{k+1} - a_{k+1})$ .

Evaluer  $\varphi(\lambda_{k+1})$  et aller à 4.

4- Remplacer  $k$  par  $k + 1$  et aller à 1.

## 2.4 Recherches linéaires inexactes ou économiques

Les recherches linéaires exactes, malgré qu'elles n'aboutissent à une solution optimale qu'avec une tolérance fixée à l'avance, elles nécessitent beaucoup d'observations à chaque itération de l'algorithme principal. Des mathématiciens ont réussi (années 60,70,80) à élaborer des recherches linéaires qui demandent peu d'observations, mais respectent en même temps la descente de la fonction économique. On a même réussi à avoir une vitesse de convergence superlinéaire en appliquant à la méthode quasi-Newton B.F.G.S.

Toutes les recherches linéaires inexactes exigent au pas  $\lambda_k$  à ce que la fonction économique  $f$  décroisse au point  $x_{k+1} = x_k + \lambda_k d_k$  relativement au point  $x_k$ , c'est à dire qu'on doit avoir

$$f(x_k + \lambda_k d_k) < f(x_k) \quad (2.1)$$

La condition de décroissance (2.1) n'est pas en general suffisante pour assurer la convergence de la suite  $\{x_k\}_{k \in \mathcal{N}}$  vers le minimum  $x^*$  de la fonction  $f$ .

Décrivons maintenant en détail les trois recherches linéaires inexactes les plus importantes. Ils s'agit des recherches linéaires inexactes d'Armijoo, de Wolfe et de Goldstein et Price.

## 2.4.1 Recherche linéaire inexacte d'Armijoo

La méthode d'Armijoo(1966).

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $x_k \in \mathbb{R}^n$ ,  $d_k \in \mathbb{R}^n$  vérifiant

$$\nabla f(x_k)^t d_k < 0 \quad (2.2)$$

La règle d'Armijoo exige à ce que  $f$  décroisse de façon suffisante au point  $x_k + \lambda_k d_k$ . Cette condition est décrite par l'inégalité suivante appelée aussi condition d'Armijoo :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \varepsilon \lambda_k \nabla f(x_k)^t d_k, \quad \varepsilon \in ]0, 1[ \quad (\text{Armijoo}) \quad (2.3)$$

En d'autres termes la réduction de  $f$  doit être proportionnelle en même temps à  $\lambda_k$  et à la dérivée directionnelle  $\nabla f(x_k)^t d_k$ .

### Interprétation graphique de la condition (Armijoo)

Définissons

$$\varphi : \mathbb{R} \rightarrow \mathbb{R}$$

par

$$\varphi(\lambda) = f(x_k + \lambda d_k)$$

On a

$$\varphi(0) = f(x_k), \quad \varphi'(0) = \nabla f(x_k)^t d_k,$$

L'équation de la tangente au graphe de  $\varphi(\lambda)$  est passant par le point  $(0, \varphi(0))$ , qu'on note  $\tilde{\varphi}(\lambda)$  est la suivante :

$$\tilde{\varphi}(\lambda) = \varphi(0) + \lambda \varphi'(0), \quad \lambda \geq 0,$$

considérons enfin la droite  $\hat{\varphi}$  suivante :

$$\hat{\varphi}(\lambda) = \varphi(0) + \lambda \varepsilon \varphi'(0), \quad \varepsilon \in ]0, 1[.$$

### Condition d'Armijoo

On voit bien d'après (2.3) que  $\lambda_k$  est acceptable s'il vérifie

$$\varphi(\lambda_k) \leq \hat{\varphi}(\lambda_k) \quad (2.4)$$

Ceci veut dire qu'au point  $\lambda_k$  le graphe de  $\varphi$  doit être au dessous de la droite  $\widehat{\varphi}$ .

**Remarque 2.4.1** L'inégalité (2.4) ou (2.3) assure la descente de la fonction économique, c'est à dire

$$f(x_k + \lambda_k d_k) < f(x_k)$$

**Remarque 2.4.2** En général, on s'assure dans la règle d'Armijoo que  $\lambda_k$  ne soit pas trop petit car cela va nuire à la convergence de l'algorithme qui pourrait converger prématurément vers un point qui n'est pas stationnaire. Cela se verra explicitement dans l'algorithme suivant :

### Algorithme de la règle d'Armijoo

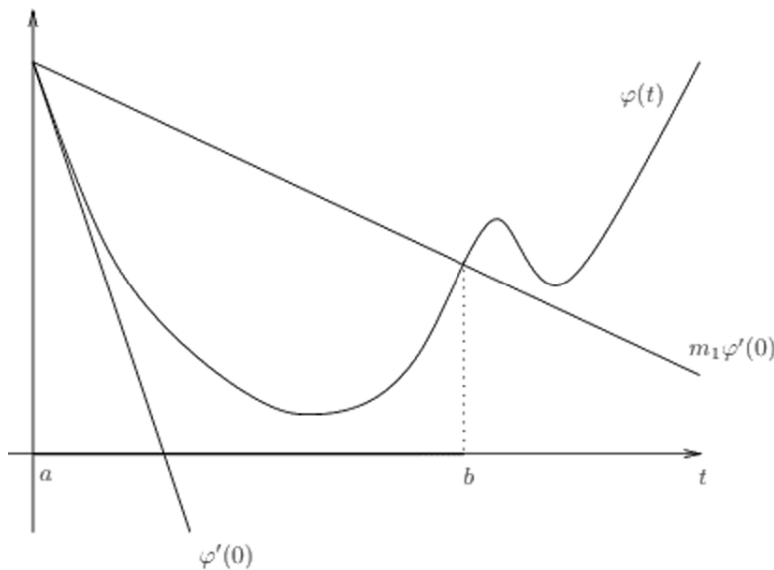
Choisir  $\lambda^*$  initial.

si  $\varphi(\lambda^*) \leq \widehat{\varphi}(\lambda^*)$ , choisir le plus grand entier  $t \geq 0$  tel que :

$$\varphi(2^t \lambda^*) \leq \widehat{\varphi}(2^t \lambda^*), \quad \text{et } \lambda_k = 2^t \lambda^*$$

Sinon choisir le plus petit entier  $t \geq 0$  tel que :

$$\varphi\left(\frac{\lambda^*}{2^t}\right) \leq \widehat{\varphi}\left(\frac{\lambda^*}{2^t}\right), \quad \text{et } \lambda_k = \frac{\lambda^*}{2^t}$$



La règle d'Armijoo

## 2.4.2 Recherche linéaire inexacte de Goldstein et Price

La règle de Goldstein s'applique lorsque le gradient de la fonction ne peut être évalué (ou est trop coûteux à obtenir), en ajoutant une deuxième inégalité à la règle d'Armijoo on obtient la règle de Goldstein et Price avec  $m_1, m_2$  sont deux constants vérifiant  $0 < m_1 < m_2 < 1$ .

Les deux inégalités de la règle de Goldstein et Price sont donc :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + m_1 \lambda \nabla^T f(x_k) d_k \quad m_1 \in ]0, 1[ \quad (2.5)$$

Et

$$f(x_k + \lambda_k d_k) > f(x_k) + m_2 \lambda \nabla^T f(x_k) d_k \quad m_2 \in ]0, 1[ \quad (2.6)$$

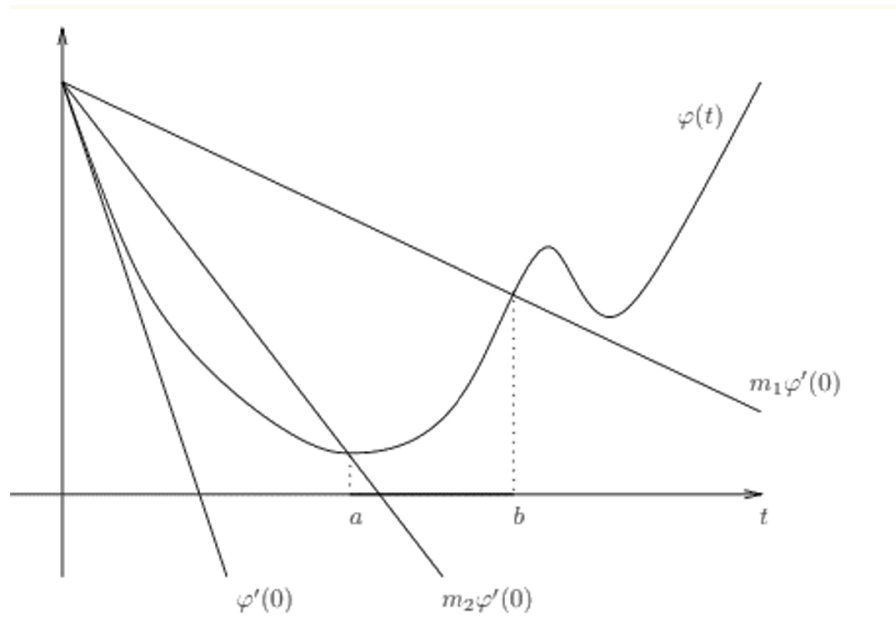
Autrement dit :

$$\varphi_k(\lambda) \leq \widehat{\varphi}_k(\lambda) \quad (2.7)$$

Et

$$\varphi_k(\lambda) \geq \varphi_k(0) + m_2 \varphi'_k(0) \quad (2.8)$$

On voit bien à la figure ci-dessous ce que signifie les conditions (2.7) et (2.8)



La règle de Goldstein

### Algorithme de la règle de Goldstein et Price :

**Étape initiale :** On dispose  $\lambda_g = 0$ ,  $\lambda_d =$  une valeur maximale quelconque, et soit  $\lambda \in ]\lambda_g, \lambda_d[$   
poser  $m_1 = 0.1$ ,  $m_2 = 0.7$

**Étape principales :**

**Étape 1 :** Calculer  $\varphi(\lambda) = f(x_k + \lambda d_k)$

Si

$$\varphi(\lambda) \leq \varphi(0) + m_1 \lambda \varphi'(0)$$

Alors aller à l'étape 2 Sinon prend  $\lambda = \lambda_d$  et aller à l'étape 4

**Étape 2 :** si

$$\varphi(\lambda) \geq \varphi(0) + m_2 \lambda \varphi'(0)$$

STOP

Sinon aller à l'étape 3

**Étape 3 :** poser  $\lambda_g = \lambda$

**Étape 4 :** rechercher un nouveau  $\lambda \in ]\lambda_g, \lambda_d[$  et retourner à l'étape 2.

### 2.4.3 Recherche linéaire inexacte de Wolfe

La règle de Wolfe fait appel au calcul de  $\varphi'_k(\lambda)$ , elle est donc en théorie plus coûteuse que la règle de Goldstein et Price. Cependant, dans de nombreuses applications, le calcul du  $\nabla f(x)$  représente un faible coût additionnel en comparaison du coût d'évaluations de  $f(x)$  c'est pourquoi cette règle est très utilisée étant donné deux réels  $m_1, m_2$  tel que  $0 < m_1 < m_2 < 1$

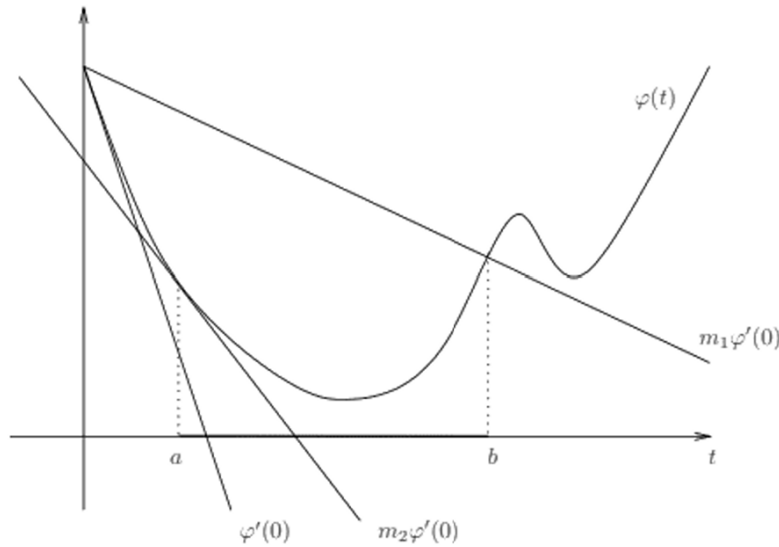
#### Recherche linéaire inexacte de Wolfe faible

Le pas  $\lambda_k$  est acceptable par la recherche linéaire inexacte de Wolfe faible ou de Wolfe tout simplement, s'il satisfait les deux conditions Wolfe1 et Wolfe2 suivante :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + m_1 \lambda_k \nabla f(x_k)^t d_k, \quad m_1 \in ]0, 1[ \quad (\text{Wolfe1})$$

$$\nabla f(x_k + \lambda_k d_k)^t d_k \geq m_2 \nabla f(x_k)^t d_k, \quad m_2 \in ]m_1, 1[ \quad (\text{Wolfe2})$$

La condition Wolfe2 s'appelle condition de courbure.



La règle de Wolfe

### Interprétation de la relation Wolfe1

La condition Wolfe1 est exactement la condition Armijoo étudiée précédemment. Cette condition assure une décroissance suffisante de la fonction  $f$ .

### Interprétation de la relation Wolfe2

Les  $\lambda_k$  sélectionnés par la condition Wolfe1 peuvent être très petits. Ceci peut avoir des conséquences facheuses sur la convergence de l'algorithme. La condition Wolfe2 évite cet inconvénient et supprime les très petites valeurs de  $\lambda_k$ . En effet

$$\nabla f(x_k + \lambda_k d_k)^t d_k = \varphi'(\lambda_k) \quad (2.9)$$

et

$$\nabla f(x_k)^t d_k = \varphi'(0) \quad (2.10)$$

Wolfe2, (2.9) et impliquent

$$\varphi'(\lambda_k) \geq m_2 \varphi'(0) \quad (2.11)$$

$\varphi'(0) = \nabla f(x_k)^t d_k < 0$ . La droite de pente  $m_2 \varphi'(0)$  va basculer vers le haut. Par conséquent la condition (2.11) va éliminer les petites valeurs de  $\lambda_k$ .

## Recherche linéaire inexacte de Wolfe forte

Le pas  $\lambda_k$  est acceptable par la recherche linéaire inexacte de Wolfe forte, s'il satisfait les deux conditions Wolfe-forte1 et Wolfe-forte2 suivantes :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + m_1 \lambda_k \nabla f(x_k)^t \cdot d_k, \quad m_1 \in ]0, 1[ \quad (\text{Wolfe - forte1})$$

$$|\nabla f(x_k + \lambda_k d_k)^t d_k| \leq m_2 \cdot |\nabla f(x_k)^t \cdot d_k|, \quad m_2 \in ]m_1, 1[ \quad (\text{Wolfe - forte2})$$

### Interprétation de la relation Wolfe-forte1

La condition Wolfe-forte1 est exactement la condition Wolfe1 ou Armijoo étudiée précédemment. Cette condition assure une décroissance suffisante de la fonction  $f$ .

### Interprétation de la relation Wolfe-forte2

La condition Wolfe forte2 implique Wolfe2. Le pas  $\lambda_k$  sélectionné par les conditions Wolfe1 et Wolfe2 peut être très loin d'un point optimal ou stationnaire de la fonction  $\varphi$ . La condition Wolfe forte2 assure que le pas  $\lambda_k$  se trouve dans le voisinage d'un point stationnaire ou un point optimal de  $\varphi$ .

## Algorithme de la règle de Wolfe

### Étape 1 (Initialisation)

Prendre  $\lambda_0 \in [0, 10^{99}]$ , calculer  $\varphi(0)$ ,  $\varphi'_0(0)$ . Prendre  $\rho = 0.1$  (ou  $\rho = 0.01$  ou  $\rho = 0.001$  ou  $\rho = 10^{-4}$ )  $\theta = 0.9$  (ou plus petit encore)

Poser  $a_0 = 0$ ,  $b_0 = 10^{99}$ ,  $k = 0$  et allez à Étape 2

### Étape 2 (test de (Wolfe1))

Calculer  $\varphi(\lambda_k)$ . Si  $\varphi(\lambda_k) \leq \varphi(0) + \rho \lambda_k \varphi'(0)$ , allez à Étape 3. Sinon

Poser  $a_{k+1} = a_k$ ,  $b_{k+1} = \lambda_k$  et allez à Étape 4

### Étape 3 (test (Wolfe2) ou (Wolfe-forte2))

Calculer  $\varphi'(\lambda_k)$ . Si  $\varphi'(\lambda_k) \geq \theta \varphi'(0)$  ( $|\varphi'(\lambda_k)| \leq -\theta \varphi'(0)$ ). STOP

Prendre  $\bar{\lambda} = \lambda_k$ . Sinon Poser  $a_{k+1} = \lambda_k$ ,  $b_{k+1} = b_k$  et allez à Étape 4

### Étape 4 (calcul de $\lambda_{k+1}$ )

$$\lambda_{k+1} = \frac{a_{k+1} + b_{k+1}}{2}$$

Poser  $k = k + 1$  et allez à Étape 2.

## 2.5 Convergence des méthodes utilisant des recherches linéaire inexactes et des directions de descente. Le théorème de Zoutendijk

### 2.5.1 Condition de Zoutendijk

Dans cette section, on va étudier la contribution de la recherche linéaire inexacte dans la convergence des algorithmes à directions de descente. Ce n'est qu'une contribution, parce que la recherche linéaire ne peut à elle seule assurer la convergence des itérés. On comprend bien que le choix de la direction de descente joue aussi un rôle. Cela se traduit par une condition, dite de Zoutendijk, dont on peut tirer quelques informations qualitatives intéressantes.

On dit qu'une règle de recherche linéaire inexacte satisfait la condition de Zoutendijk s'il existe une constante  $C > 0$  telle que pour tout indice  $k \geq 1$  on ait

$$f(x_{k+1}) \leq f(x_k) - C \|\nabla f(x_k)\|^2 \cos^2 \theta_k \quad (2.12)$$

où  $\theta_k$  est l'angle que fait  $d_k$  avec  $-\nabla f(x_k)$ , défini par

$$\cos \theta_k = \frac{-\nabla^T f(x_k) d_k}{\|d_k\| \|\nabla f(x_k)\|}. \quad (2.13)$$

Voici comment on se sert de la condition de Zoutendijk.

#### Théorème 2.2 (de Zoutendijk)

Si la suite  $\{x_k\}$  générée par un algorithme d'optimisation vérifie la condition de Zoutendijk (2.12) et si la suite  $\{f(x_k)\}$  est minorée, alors

$$\sum_{k \geq 1} \|\nabla f(x_k)\|^2 \cos^2 \theta_k < \infty \quad (2.14)$$

### Démonstration

En sommant les quantités inégalités  $\| \nabla f(x_k) \|^2 \cos^2 \theta_k$  tout en prenant en considération (2.13), on a

$$\sum_{k \geq 1}^l \| \nabla f(x_k) \|^2 \cos^2 \theta_k \leq \frac{1}{c} (f(x_1) - f(x_{l+1}))$$

La série est donc convergente puisqu'il existe une constante  $C'$  telle que pour tout  $k$ ,  $f(x_k) \geq C'$ .

### Conséquence importante du théorème de Zoutendijk

La condition (2.14) implique

$$\| \nabla f(x_k) \|^2 \cos^2 \theta_k \rightarrow 0 \quad (k \rightarrow \infty) \quad (2.15)$$

Cette limite peut être utilisée pour en déduire la convergence de l'algorithme. En effet si notre algorithme génère une suite  $\{x_k\}$  de la forme :

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

Si le choix de  $d_k$  est tel que

$$\cos \theta_k \geq \delta > 0; \quad \forall k$$

alors il découle de (2.15) que

$$\lim_{k \rightarrow \infty} \| \nabla f(x_k) \| = 0 \quad (2.16)$$

Les deux propositions suivantes précisent les circonstances dans lesquelles la condition de Zoutendijk (2.12) est vérifiée avec les règles d'Armijoo et de Wolfe.

**Proposition 2.5.1** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction continûment différentiable dans un voisinage de  $\Gamma = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ .

On considère un algorithme à directions de descente  $d_k$ , qui génère une suite  $\{x_k\}$  en utilisant la recherche linéaire d'Armijoo, avec  $\alpha_1 > 0$ .

Alors il existe une constante  $C > 0$  telle que, pour tout  $k \geq 1$ , l'une des conditions

$$f(x_{k+1}) \leq f(x_k) - C \nabla^T f(x_k) d_k \quad (2.17)$$

ou

$$f(x_{k+1}) \leq f(x_k) - C \|\nabla f(x_k)\|^2 \cos^2 \theta_k \quad (2.18)$$

est vérifiée.

*Proposition 2.5.2* Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction continûment différentiable dans un voisinage de  $\Gamma = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ . On considère un algorithme à directions de descente  $d_k$ , qui génère une suite  $\{x_k\}$  en utilisant la recherche linéaire de Wolfe (Wolfe1) et (Wolfe2).

Alors il existe une constante  $C > 0$  telle que, pour tout  $k \geq 1$ , la condition de Zoutendijk (2.12) est vérifiée.

## CHAPITRE

3

# MÉTHODES NUMÉRIQUES D'OPTIMISATION SANS CONTRAINTES ET LEURS ALGORITHMES

### 3.1 Méthode du Gradient Conjugué

*La méthode du Gradient Conjugué est une méthode très puissante et très utilisée pour résoudre les problèmes d'optimisation sans contraintes de grande taille. On l'utilise aussi pour résoudre les grands systèmes linéaires.*

*Elle repose sur le concept des directions conjuguées parce que les gradients successifs sont orthogonaux entre eux et aux directions précédentes.*

*L'idée initiale était de trouver une suite de directions de descente permettant de résoudre le problème :*

$$\min\{f(x); x \in \mathbb{R}^n\}$$

*Où  $f$  est régulière (continûment différentiable).*

*La méthode a été découverte indépendamment par Stiefel (1952) et Hestenes (1951) pour la minimisation de fonctions quadratiques strictement convexes (cas linéaire), qui l'on finalisée et publiée ensemble (Hestenes et Stiefel, 1952).*

*Plusieurs mathématiciens ont étendu cette méthode pour le cas non linéaire, pour la première fois, en 1964 par Fletcher et Reeves (méthode de Fletcher-Reeves), en 1969 par Polak, Ribière et Ployak (méthode de Polak Ribière- Ployak), en 1987 par Fletcher (méthode de la descente), en 1995 par Dai et Yuan (méthode de Dai-Yuan).*

### 3.1.1 Méthode des directions conjuguées

#### Description de la méthode

*Ces méthodes sont basées sur l'important concept de la conjugaison et ont été développées afin de résoudre le problème quadratique :*

$$\min f(x) = \frac{1}{2}x^T Hx + b^T x + c, \text{ avec } x \in \mathbb{R}^n \quad (3.1)$$

*Où  $H \in M_{n \times n}$  est symétrique et définie positive,  $c \in \mathbb{R}$  et  $b \in \mathbb{R}^n$ .*

*Les méthodes des directions conjuguées peuvent résoudre les problèmes de cette forme en au plus  $n$  itérations et elles n'utilisent pas des dérivées, sauf dans le cas particulier de la méthode du Gradient Conjugué.*

#### Principe général

*Il s'agit de méthodes itératives qui sont appliquées à une fonction quadratique strictement convexe de  $n$  variables, conduisent à l'optimum en  $n$  étapes au plus. Considérons une fonction quadratique quelconque :*

$$f(x) = \frac{1}{2}x^T Hx + b^T x + c; \quad x \in \mathbb{R}^n, b \in \mathbb{R}^n, c \in \mathbb{R},$$

*$H \in M_{n \times n}$  est symétrique et définie positive*

*Et cette méthode de la forme :*

$$\begin{cases} x_0 \text{ donné} \\ x_{k+1} = x_k + \lambda_k d_k \end{cases} \quad (3.2)$$

Où  $\lambda_k$  est optimal et  $d_1, d_2, \dots, d_n$  possédant la propriété d'être mutuellement conjuguées par rapport à la fonction quadratique.

Donnons la définition de "conjugaison"

**Définition 3.1.1** Soit  $H$  une matrice symétrique  $n \times n$ , définie positive. Les vecteurs non nuls de  $\mathbb{R}^n$  :  $d_1, d_2, \dots, d_k$  sont dits  $H$ -conjugué (ou conjugués par rapport à  $H$ ) si :

$$d_i^T H d_j = 0 \quad \forall i, j; \quad i \neq j \quad (3.3)$$

**Théorème 3.1.1** Soit  $\{d_0, d_1, \dots, d_n\}$  une famille de vecteurs  $H$ -conjugués alors elles sont linéairement indépendants.

Et soit  $\{x_0, x_1, \dots, x_n\}$  dans  $\mathbb{R}^n$  les itérés générés par une méthode de directions conjuguées, alors :

1- pour tout  $k = 0, \dots, n$  le pas  $\lambda_k$  est défini par :

$$\lambda_k = \frac{-d_k^T (Hx_k + b)}{d_k^T H d_k} = \frac{-d_k^T \nabla f(x_k)}{d_k^T H d_k} \quad (3.4)$$

2- Pour tout  $k = 0, \dots, n$ ,  $\nabla f(x_k)$  est orthogonal à  $d_0, d_1, \dots, d_n$ , c'est-à-dire

$$\nabla f(x_k)^T d_i = 0, \quad \forall i = 0, \dots, n \quad (3.5)$$

3- Soit  $k$  tel que  $\nabla f(x_k) = 0$  alors,

$$\nabla f(x_i) = 0, \quad \forall i = k, \dots, n \quad (3.6)$$

Calcul de  $\lambda_k$  : Comme  $\lambda_k$  minimise  $f$  dans la direction  $d_k$ , on a,  $\forall k$  :

$$f'(\lambda_k) = d_k^T \nabla f(x_{k+1}) = 0$$

$$d_k^T \nabla f(x_{k+1}) = d_k^T (Hx_{k+1} + b) = 0$$

Soit :

$$d_k^T H(x_k + \lambda_k d_k) + d_k^T b = 0$$

D'où l'on tire :

$$d_k^T H x_k + \lambda_k d_k^T H d_k + d_k^T b = 0$$

$$\lambda_k = \frac{-d_k^T (Hx_k + b)}{d_k^T H d_k} \quad (3.7)$$

### Comment construire les directions H-conjugués ?

*Des directions H-conjuguées  $\{d_0, d_1, \dots, d_n\}$  peuvent être générées à partir d'un ensemble de vecteurs linéairement indépendants  $\{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_n\}$  en utilisant la procédure dite de Gram-Schmidt, de telle sorte que pour tout  $i$  entre 0 et  $n$ , le sous-espace généré par  $\{d_0, d_1, \dots, d_i\}$  soit égale au sous-espace généré par  $\{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_i\}$ . Alors  $d_{i+1}$  est construite comme suit :*

$$d_{i+1} = \varepsilon_{i+1} + \sum_{m=0}^i \varphi_{(i+1)m} d_m \quad (3.8)$$

*Nous pouvons noter que si  $d_{i+1}$  est construite d'une telle manière, elle est effectivement linéairement indépendante avec  $\{d_0, d_1, \dots, d_i\}$ .*

*En effet, le sous-espace généré par les directions  $\{d_0, d_1, \dots, d_i\}$  est le même que le sous-espace généré par les directions  $\{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_i\}$ , et  $\varepsilon_{i+1}$  est linéairement indépendant de  $\{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_i\}$ .*

*$\varepsilon_{i+1}$  ne fait donc pas partie du sous-espace généré par les combinaisons linéaires de la forme  $\sum_{m=0}^i \varphi_{(i+1)m} d_m$ , de sorte que  $d_{i+1}$  n'en fait pas partie non plus et est donc linéairement indépendante des  $\{d_0, d_1, \dots, d_i\}$ .*

### 3.1.2 Convergence de la méthode des direction conjugués

*Théorème 3.1.2 pour tout  $x_0 \in \mathbb{R}^n$  la suite  $\{x_k\}$  généré par l'algorithme (3.2) (3.7) converge vers la solution  $x^*$  du système linéaire  $Hx = b$  en  $n$  étapes au plus.*

*démonstration 3.1.1 Puisque les directions  $d_k$  pour  $0 \leq k \leq n-1$  sont linéairement indépendant, alors forment une base de  $\mathbb{R}^n$ . Donc on peut écrire la différence entre  $x_0$  et la solution  $x^*$  par l'expression suivante :*

$$x^* - x_0 = \beta_0 d_0 + \beta_1 d_1 + \dots + \beta_{n-1} d_{n-1} \quad (*)$$

*On multiple (\*) par  $d_k^T H$  et utilisant (3.1), on obtient :*

$$\beta_k = \frac{d_k^T H(x^* - x_0)}{d_k^T H d_k} \quad \text{pour } 0 \leq k \leq n-1 \quad (**)$$

*démontrons que  $\beta_k$  coïncide avec  $\lambda_k$  donnée par (3.3)*

Si g n r  par l'algorithme (3.2) (3.3), Alors on obtient :

$$\begin{aligned}x_k &= x_0 + \lambda_0 d_0 + \lambda_1 d_1 + \dots + \lambda_{n-1} d_{n-1} \\x_k - x_0 &= \lambda_0 d_0 + \lambda_1 d_1 + \dots + \lambda_{n-1} d_{n-1}\end{aligned}$$

Et donc :

$$\begin{aligned}x^* - x_0 &= (x^* - x_k) + (x_k - x_0) \\d_k^T H(x^* - x_0) &= d_k^T H(x^* - x_k) + d_k^T H(x_k - x_0) \\&= d_k^T H(x^* - x_k) \\&= d_k^T H(b - Hx_k) \\&= d_k^T g_k; \quad g_k = \nabla f(x_k)\end{aligned}$$

On compare cette relation avec (3.7) et (\*\*\*) on trouve :

$$\beta_k = \frac{d_k^T H(x^* - x_0)}{d_k^T H d_k} = \frac{-d_k^T g_k}{d_k^T H d_k} = \lambda_k \quad (3.9)$$

Donc  $\beta_k = \lambda_k$  ce qui ach ve la d monstration.

### 3.1.3 M thode du gradient conjugu 

On s'int resse dans cette section   la minimisation d'une fonction  $f$  de  $\mathbb{R}^n$  dans  $\mathbb{R}$ , non n cessairement quadratique :

$$\min\{f(x); x \in \mathbb{R}^n\} \quad (3.10)$$

Les m thodes du gradient conjugu  pour r seaux cette probl me sont des m thodes it ratives de la forme :

$$x_{k+1} = x_k + \lambda_k d_k \quad (3.11)$$

Le pas  $\lambda_k \in \mathbb{R}$   tant d termin  par une recherche lin aire. La direction  $d_k$  est d finie par la formule de r currence suivante ( $\beta_k \in \mathbb{R}$ ) :

$$d_k = \begin{cases} -g_1 & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases} \quad (3.12)$$

telle que  $g_k = \nabla f(x_k)$

Ces méthodes sont des extensions de la méthode du gradient conjugué linéaire du cas quadratique, si  $\beta_{k+1}$  prend l'une des valeurs :

D'où

$$\beta_{k+1}^{PRP} = \frac{g_{k+1}^T y_{k+1}}{\|g_k\|^2} \quad (3.13)$$

$$\beta_{k+1}^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T g_k} \quad (3.14)$$

$$\beta_{k+1}^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k} \quad (3.15)$$

$$\beta_{k+1}^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \quad (3.16)$$

$$\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \quad (3.17)$$

où  $y_k = g_{k+1} - g_k$

### 3.1.4 Algorithme de la méthode du gradient conjugué

Algorithme

*Etape 0 : (initialisation) Soit  $x_0$  le point de départ,  $g_0 = \nabla f(x_0)$ , poser  $d_0 = -g_0$ , poser  $k = 0$  et aller à l'étape 1.*

*Etape 1 : Si  $g_k = 0$  STOP ( $x_k = \hat{x}$ ). " Test d'arrêt "*

*Sinon aller à l'étape 2.*

*Etape 2 : Définir  $x_{k+1} = x_k + \lambda_k d_k$  avec :*

$\lambda_k$  : calculé par la recherche linéaire

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$$

$\beta_k$  : définir selon la méthode

*Poser  $k = k + 1$  et aller à l'étape 1.*

### 3.1.5 Convergence de la méthode du gradient conjugué

On s'intéresse dans cette section à la minimisation d'une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , non nécessairement quadratique :

$$\min\{f(x); x \in \mathbb{R}^n\}$$

Les méthodes du gradient conjugué pour résoudre ce problème sont des méthodes itératives de la forme :

$$x_{k+1} = x_k + \lambda_k d_k$$

Avec

$\lambda_k$  : calculé par la recherche linéaire

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$$

$\beta_k$  : définir selon la méthode

**Supposition 3.1 :**

- (i) L'ensemble  $E := \{x \in \mathbb{R}^n, f(x) < f(x_1)\}$  est borné, ou  $x_1 \in \mathbb{R}^n$  est le point initial.
- (ii) Sur un voisinage  $N$  de  $E$ , la fonction objectif  $f$  est continûment différentiable et son gradient est lipchitzien i.e.

$$\exists L > 0; \|g(x) - g(\hat{x})\| \leq L \|x - \hat{x}\| \quad \forall x, \hat{x} \in N \quad (3.18)$$

**Remarque 3.1.1** Ce supposition implique qu'il existe  $\gamma > 0$  tel que :

$$\|g(x)\| < \gamma; \quad \forall x \in E$$

**Définition 3.1.2** On dit que  $d_k$  est une direction de descente suffisante si :

$$g_k^T d_k < -c \|g_k\|^2$$

Rappelons les conditions de Wolfe faibles :

$$f(x_k + \lambda d_k) \leq f(x_k) + m_1 \lambda \nabla^T f(x_k) d_k \quad (3.19)$$

$$\nabla^T f(x_k + \lambda d_k) d_k \geq m_2 \nabla^T f(x_k) d_k \quad (3.20)$$

$$0 < m_1 < m_2 < 1$$

*Théorème 3.1.3* Considérons une méthode du type (3.11) et (3.12) dans laquelle  $d_k$  est une direction de descente et le pas  $\lambda_k$  est déterminé par la règle de Wolfe faible avec :  $0 < \omega_1 < \frac{1}{2}$ , Considérons aussi que la supposition (3.1) soit satisfaite. Alors pour une telle méthode la condition de Zoutendijk suivante :

$$\sum_{k \geq 1} \cos^2 \theta_k \|g_k\|^2 < \infty \quad (3.21)$$

est vérifiée.

## 3.2 Méthode de la plus forte pente

Cette méthode a été découverte par Cauchy en 1847 est l'une des anciennes méthodes les plus utilisées pour résoudre le problème (P), elle est également connu sous le nom de méthode de la plus forte pente ou de plus de la plus profonde descente (steepest descente, en anglais) car si  $x_k \in \mathbb{R}^n$  et si  $\nabla f(x_k) \neq 0$  alors la direction  $d_k = -\nabla f(x_k)$  est la meilleure direction de descente.

Le théorème suivant va montrer que la décroissance de la fonction sera la plus forte en suivant la direction  $-\nabla f(x_k)$ .

*Théorème 3.2.1* Supposons que  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  soit différentiable au point  $x$ , et supposons que

$$\nabla f(x) \neq 0$$

Considérons le problème optimal

$$\text{Minimiser}_{\|d\| \leq 1} f'(x, d)$$

où  $f'(x, d)$  est la dérivée directionnelle de  $f$  au point  $x$  et dans la direction  $d$ . Alors la solution optimale de ce problème est donnée par

$$\bar{d} = -\frac{\nabla f(x)}{\|\nabla f(x)\|}.$$

*démonstration 3.2.1 On a*

$$f'(x, d) = \lim_{\lambda \rightarrow 0^+} \frac{f(x + \lambda d) - f(x)}{\lambda} = \nabla f(x)^t d$$

*donc on va minimiser  $\nabla f(x)^t d$  dans  $\{d : \|d\| \leq 1\}$ .*

*En appliquant l'inégalité de Schwartz, on obtient*

$$|\nabla f(x)^t d| \leq \|\nabla f(x)\| \|d\| \quad (3.22)$$

*Si*

$$\nabla f(x)^t d \geq 0$$

*alors*

$$\nabla f(x)^t d \geq -\|\nabla f(x)\| \|d\|$$

*Si*

$$\nabla f(x)^t d \leq 0$$

(3.22) *implique*

$$-\nabla f(x)^t d \leq \|\nabla f(x)\| \|d\|$$

*par conséquent on a toujours*

$$\nabla f(x)^t d \geq -\|\nabla f(x)\| \|d\|$$

*Pour  $\|d\| \leq 1$  on a*

$$\|\nabla f(x)\| \|d\| \leq \|\nabla f(x)\| \Rightarrow -\|\nabla f(x)\| \|d\| \geq -\|\nabla f(x)\|$$

*alors  $\forall d : \|d\| \leq 1$  on a*

$$\nabla f(x)^t d \geq -\|\nabla f(x)\|$$

*D'autre part on a  $\|\bar{d}\| = 1$  et  $\bar{d}$  vérifie*

$$\nabla f(x)^t \bar{d} = \nabla f(x)^t \left( -\frac{\nabla f(x)}{\|\nabla f(x)\|} \right) = -\|\nabla f(x)\|$$

.

Interprétation du théorème 3.2.1 :

*Nous allons à partir du théorème 3.2.1 donner une idée intuitive sur l'appellation méthode de la plus forte pente. En effet d'après le théorème 3.2.1 on a :*

$$f'(x, d) \geq f'(x, \bar{d}); \quad \forall d, \|d\| \leq 1.$$

*En utilisant la définition de la dérivée directionnelle*

$$\lim_{\lambda \rightarrow 0^+} \frac{f(x + \lambda d) - f(x)}{\lambda} \geq \lim_{\lambda \rightarrow 0^+} \frac{f(x + \lambda \bar{d}) - f(x)}{\lambda}$$

*Cette dernière inégalité implique qu'il existe  $\delta > 0$  tel que :*

$$[f(x + \lambda d) - f(x)] - [f(x + \lambda \bar{d}) - f(x)] \geq 0, \quad \forall \lambda \in ]-\delta, +\delta[$$

*alors*

$$f(x + \lambda d) \geq f(x + \lambda \bar{d}), \quad \forall \lambda \in ]-\delta, +\delta[ \text{ et } \forall d, \|d\| \leq 1.$$

### 3.2.1 Algorithme de la méthode de la plus forte pente

*Étape initiale : Choisir un  $\varepsilon > 0$ . Choisir un point initial  $x_1$ . Poser  $k = 1$  et aller à l'étape principale.*

*Étape principale : Si  $\|\nabla f(x)\| < \varepsilon$  **STOP**.*

*Sinon poser  $d_k = -\nabla f(x)$  et soit  $\lambda_k$  la solution optimale de la recherche linéaire*

$$\min\{f(x_k + \lambda d_k); \lambda \geq 0\}$$

*Poser*

$$x_{k+1} = x_k + \lambda_k d_k$$

*Remplacer  $k$  par  $k + 1$  et répéter l'étape principale.*

Inconvénients de la méthode de la plus forte pente :

Lenteur de la méthode au voisinage des points stationnaires :

*Cette méthode travaille de façon performante dans les premiers étapes de l'algorithme mais au voisinage des points stationnaires elle devient très lente.*

*On peut expliquer ce problème comme suit :*

*On a :*

$$f(x_k + \lambda d) = f(x_k) + \lambda \nabla f(x_k)^t d + \lambda \|d\| \alpha(x_k, \lambda d)$$

*où  $\alpha(x_k, \lambda d) \rightarrow 0$  quand  $\lambda d \rightarrow 0$ .*

*Si  $d = -\nabla f(x_k)$  alors  $x_{k+1} = x_k - \lambda \nabla f(x_k)$  et donc*

$$f(x_k + \lambda d) - f(x_k) = \lambda [-\|\nabla f(x_k)\|^2 + \|\nabla f(x_k)\| \alpha(x_k, -\lambda \nabla f(x_k))]$$

*D'après l'expression précédente, on voit que lorsque  $x_k$  s'approche d'un point stationnaire, et si  $f$  est continûment différentiable, alors  $\|\nabla f(x_k)\|$  est proche de zéro. Donc le terme à droite s'approche de zéro, indépendamment de  $\lambda$ , et par conséquent  $f(x_k + \lambda d)$  ne s'éloigne pas beaucoup de  $f(x_k)$  quand on passe du point  $x_k$  au point  $x_{k+1}$ .*

**Le phénomène de Zigzaguing :**

*Il n'est pas facile de vérifier que pour la méthode du gradient on a toujours*

$$d_k^t d_{k+1} = 0$$

*c'est à dire que la suite  $\{x_k\}$  engendrée par l'algorithme de la méthode du gradient, zigzague. Ceci crée un phénomène de ralentissement dans l'acheminement des points  $x_k$  vers la solution optimale.*

### 3.2.2 Quelques remèdes

**Changement de direction**

*Au lieu de prendre comme direction de descente, la direction :*

$$d_k = -\nabla f(x_k)$$

*on prend des directions de la forme :*

$$d_k = -D\nabla f(x_k)$$

où  $D$  est une matrice choisie convenablement ( $D$  pourrait être par exemple l'inverse de la matrice Hessienne au point  $x_k$  c'est à dire  $(H(x_k))^{-1}$ ). Un autre choix pourrait s'opérer de la façon suivante :

$$d_k = -\nabla f(x_k) + g_k,$$

où  $g_k$  est un vecteur approprié.

Accélération de la convergence :

On peut aussi accélérer la convergence de la méthode du gradient. Pour cela on transforme grâce à un algorithme d'accélération de la convergence la suite  $\{x_k\}$  en une suite  $\{y_k\}$  qui convergerait vers la même limite que la suite  $\{x_k\}$ , mais convergerait plus rapidement. Si on note par  $x^*$  cette limite commune on exprime cette rapidité par la limite suivante :

$$\lim_{k \rightarrow \infty} \frac{y_k - x^*}{x_k - x^*} = 0.$$

### 3.2.3 Convergence de la méthode de la plus forte pente

*Théorème 3.2.2 ([Bazara]) (convergence de la méthode de la plus forte pente avec la recherche linéaire d'Armijo). Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , telle que  $\nabla f(x)$  est continûment Lipschitzien de constante  $G$  dans l'ensemble*

$$\delta(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}, \quad x_0 \in \mathbb{R}^n \text{ quelconque.}$$

*Considérons l'algorithme de la méthode de la plus forte pente avec la recherche linéaire d'Armijo, c'est à dire l'algorithme défini comme suit :*

*Initialisation : Fixons  $\bar{\lambda} > 0$ ,  $0 < \varepsilon < 1$  et  $x_0 \in \mathbb{R}^n$  point initial, poser  $k = 0$  et aller à Étape principale.*

*Étape principale : A l'itération  $k$  définissons la direction  $d_k = -\nabla f(x_k)$  et considérons la fonction d'Armijo*

$$\widehat{\theta}(\lambda) = \theta(0) + \lambda \varepsilon \theta'(0) \tag{3.23}$$

où  $\theta(\lambda)$  est la fonction suivante :

$$\theta(\lambda) = f(x_k + \lambda d_k) = f[x_k - \lambda \nabla f(x_k)]; \quad \lambda \geq 0. \tag{3.24}$$

Si  $\nabla f(x_k) = 0$  *STOP*.

*Sinon*

Trouver le plus petit entier  $t \geq 0$  tel que

$$\theta\left(\frac{\bar{\lambda}}{2^t}\right) \leq \hat{\theta}\left(\frac{\bar{\lambda}}{2^t}\right) \quad (3.25)$$

et définir le successeur  $x_{k+1}$  de  $x_k$  comme suit

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

avec

$$\lambda_k = \frac{\bar{\lambda}}{2^t}.$$

Poser  $k = k + 1$  et aller à *Étape principale*.

*Fin*

Soit  $\{x_k\}$  une suite générée par cette algorithm. Alors ou bien il s'arrête après un nombre fini d'itérations en un point  $x_{k_0}$  tel que  $\nabla f(x_{k_0}) = 0$ , ou bien il génère une suite infinie  $\{x_k\}_{k \in \mathbb{N}}$  telle que :

$$\nabla f(x_k) \rightarrow_{k \rightarrow \infty} 0$$

### 3.3 Méthodes quasi-Newtoniennes

#### 3.3.1 Méthode de Newton

Considérons le problème d'optimisation sans contraintes (P)

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

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}$

Le principe de la méthode de Newton consiste à minimiser successivement les approximations du second ordre de  $f$ , plus précisément si

$$f(x) = f(x_k) + \nabla f(x_k)^t (x - x_k) + (x - x_k)^t H(x_k) (x - x_k) + o\|x - x_k\|^2,$$

posons

$$q(x) = f(x_k) + \nabla f(x_k)^t(x - x_k) + (x - x_k)^t H(x_k)(x - x_k).$$

Soit  $x_{k+1}$  l'optimum de  $q$ , alors il vérifie  $\nabla q(x_{k+1}) = 0$ , soit en remplaçant :

$$\nabla f(x_k) + H(x_k)(x_{k+1} - x_k) = 0$$

ou encore

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

donc

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

Donc pour obtenir une méthode qui converge superlinéairement, il est nécessaire d'approximer l'étape de Newton asymptotiquement. C'est le principe de Dennis et Moré. Comment peut-on y aboutir sans évaluer la matrice Hessienne dans chaque itération ?

La réponse a été découverte par Davidon en 1959 et a été développée et popularisée par Fletcher et Powell en 1963. Elle consiste à commencer par n'importe quelle approximation de la matrice Hessienne et à chaque itération, on améliore la matrice en introduisant la courbure du problème mesuré tous au long de l'étape. Si cette amélioration est faite correctement, on obtient quelques méthodes remarquablement robustes et efficaces, qu'on appelle les méthodes de la variable métrique ou quasi Newton. Ils ont libéré l'optimisation non linéaire en procurant une alternative à la méthode de Newton, qui est très coûteuse pour plusieurs applications.

Il y a plusieurs méthodes de variable métrique, on s'étalera particulièrement sur les trois plus importantes, la méthode de correction de rang un, la méthode DFP (Davidon, Fletcher, Powell), et la méthode BFGS (Broyden, Fletcher, Goldfarb, Shanno).

Algorithme de la méthode de Newton

*Étape initiale* : Soit  $\varepsilon > 0$ , critère d'arrêt. Choisir  $x_1$  point initial, poser  $k = 1$  et aller à l'étape principale.

*Étape principale* : Si  $\|\nabla f(x_k)\| \leq \varepsilon$  STOP, sinon poser  $x_{k+1} = x_k - [H(x_k)]^{-1} \nabla f(x_k)$  remplacer  $k$  par  $k + 1$  et aller à l'étape principale.

## Avantages et inconvénients

### ▼ Avantages

- Si le point  $x_0$  est assez proche de la solution optimale locale  $x^*$  telle que  $H(x^*)$  soit définie positive, alors l'algorithme de Newton converge de façon quadratique vers la solution  $x^*$   
c'est-à-dire que l'on a

$$\|x_{k+1} - x^*\| \leq \gamma \|x_k - x^*\|^2 \quad \gamma \geq 0$$

- La méthode de Newton fonctionne bien si  $1 < n < 10$  car il est facile de déterminer  $\nabla^2 f(x_k)$ .
- La méthode de Newton converge en une seule itération lorsqu'elle est appliquée à une fonction quadratique strictement convexe.

### ▼ Inconvénients

- Il faut calculer les dérivées secondes de  $f$ , ce qui peut être très coûteux en temps de calcul.
- L'algorithme n'est pas globalement convergent si le premier itéré  $x_0$  est éloigné de  $x^*$ .
- Le successeur  $x_{k+1}$  de  $x_k$  n'est pas toujours bien défini, si  $\nabla^2 f(x) = H(x)$  est singulière.
- Si  $f$  n'est pas strictement convexe, l'algorithme ne génère par nécessairement des directions de descente de  $f$ .
- Un système linéaire d'ordre  $n$  doit être résolu à chaque itération, et on dispose pas de place mémoire pour stocker les  $G(n^2)$  éléments d'une matrice.

## 3.3.2 Méthodes de Quasi-Newton

La classe des méthodes quasi-newtoniennes n'exigent pas des expressions explicites pour les deuxièmes dérivés. Celles-ci désignés parfois sous le nom des méthodes métriques variables. Car le nom implique que la base de ces méthodes est la méthode classique de Newton. Le principe de base dans des méthodes quasi-newtoniennes est que la direction de recherche est basée sur une matrice  $M_k$ , qui atteint le même but de l'inverse du Hessien  $H_k$ , dans la méthode de Newton.

Les algorithmes de quasi-Newton prennent donc en charge la génération de la suite des itérés  $\{x_k\}_{k \geq 0}$ , mais aussi celle de la suite des matrices  $B_k$  approchant  $F'(x_k)$  ou  $H_k$ . La première suite générée par la récurrence

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

où  $s_k = \lambda_k d_k$  dans les algorithmes avec recherche linéaire ( $\lambda_k > 0$  est le pas et  $d_k$  vérifie  $B_k d_k = -F(x)$  ou  $B_k d_k = -g_k$  selon le problème) et  $s_k$  est la solution d'un problème d'optimisation quadratique avec une borne sur la norme du déplacement dans les algorithmes avec régions de confiance.

Donc une méthode de quasi-Newton est une méthode de type :

$$\begin{cases} x_{k+1} = x_k + \lambda_k d_k \\ d_k = -M_k g_k \end{cases} \quad (3.26)$$

ou bien

$$\begin{cases} x_{k+1} = x_k + \lambda_k d_k \\ d_k = -B_k^{-1} g_k \end{cases} \quad (3.27)$$

où  $M_k$  (respectivement  $B_k$ ) est une matrice destinée à approcher l'inverse du Hessian de  $f$  (respectivement le hessien de  $f$ ) en  $x_k$ .

Si l'on veut que  $M_{k+1}$  approche le hessien  $H_{k+1} = \nabla^2 f(x_{k+1})$ , sans calculer ce dernier, la variation du gradient d'un itéré au suivant qu'il faut tirer de l'information. Puisque l'on a

$$y_k = g_{k+1} - g_k = \left( \int_0^1 \nabla^2 f(x_k + t s_k) dt \right) s_k$$

il est naturel d'imposer à la matrice  $M_{k+1}$  de vérifier l'équation

$$s_k = M_{k+1} y_k$$

Cette équation porte le nom d'équation de quasi-Newton.

Il est également normal d'imposer à  $M_{k+1}$  d'être symétrique (puisque  $[H(x_k)]^{-1}$  est symétrique).

### Algorithme de la méthode de Quasi-Newton

1) On choisit  $x_1, B_1$  défini positif et  $\varepsilon > 0$ .

2) Si  $\|\nabla f(x_k)\| < \varepsilon$  **STOP**.

*Sinon, on pose :  $d_k = -B_k \nabla f(x_k)$  telle que :  $x_{k+1} = x_k + \lambda_k d_k$*

3) On calcule

$$B_{k+1} = B_k + \frac{(s_k - B_k y_k)(s_k - B_k y_k)^t}{(s_k - B_k y_k)^t y_k}$$

- Où :  $\begin{cases} s_k = x_{k+1} - x_k \\ y_k = \nabla f(x_{k+1}) - \nabla f(x_k) \end{cases}$
- Mettre  $k + 1 \rightarrow k$  et retourner à 2.

### Quelques propriétés des algorithmes de quasi-Newton

• Des méthodes quasi-newtoniennes, comme la plupart des autres méthodes, sont développées pour le problème quadratique convexe et puis sont prolongées au problème général. Elles sont considérées comme étant parmi les méthodes les plus efficaces et sont employées très intensivement dans de nombreuses applications. Plusieurs méthodes quasi-newtoniennes distinctes ont évolué ces dernières années. Dans ce chapitre, nous discuterons en détail les quatre méthodes les plus importantes de cette classe qui sont :

1- Méthode de correction de rang un (SR1 [1960])

2- Méthode de Davidon-Fletcher-Powell (DFP [D – 1959 et F.P – 1963])

3- Méthode de Broyden-Fletcher-Goldfarb-Shanno (BFGS [1970])

- les algorithmes quasi-newtoniens convergent moins rapidement que l'algorithme de Newton. Dans les implémentations correctes, les itérés convergent toutefois  $q$ -superlinéairement.
- Chaque itération demande moins de calcul au simulateur que dans l'algorithme de Newton : il n'est pas nécessaire d'évaluer les dérivées secondes.
- Dans leur version standard, ces algorithmes peuvent être utilisés pour un nombre de variables qui n'est pas trop grand, disons  $n$  500 pour fixer les idées, cependant on a développé ces dernières années (à partir de 1990) des algorithmes qui sont des variantes de la méthode BFGS et qui prennent en charge les problèmes de grande taille (citons particulièrement la méthode BFGS à mémoire limitée, La lBFGS).

## Formules de mise à jour de l'approximation du Hessien

*Le principe de la mise à jour consiste à une itération donnée de l'algorithme*

$$\begin{cases} x_{k+1} = x_k + \lambda_k d_k \\ d_k = -M_k g_k \end{cases} \quad (3.28)$$

*Nous cherchons  $M_k$  sous la forme suivante :*

$$M_{k+1} = M_k + \Delta_k \quad (3.29)$$

*avec  $\Delta_k$  matrice symétrique appelée la matrice d'erreur, assurant la relation de quasi-Newton. On exige aussi que  $M_{k+1}$  soit définie positive, sous l'hypothèse que  $M_k$  est définie positive.*

*La formule (3.29) permet d'utiliser les nouvelles informations obtenues lors de l'étape  $k$  de l'algorithme, c'est-à-dire essentiellement le gradient  $g_{k+1} = \nabla f(x_{k+1})$  au point  $x_{k+1}$ , obtenu par une recherche linéaire (exacte ou approchée) dans la direction  $d_k$ . Il existe différentes formules de type (3.26). Suivant que  $\Delta_k$  est de rang un ou deux, on parlera de correction de rang un (SR1) ou de rang deux (DFP, BFGS).*

### 3.3.3 Méthode de correction de rang un (SR1 [1960])

*Cette méthode a été proposée par Broyden ([27]), Davidon ([25]), Fiacco et McCormick ([28]), Murtagh et Sargent ([29]), et Wolfe ([30]).*

*Étant donné que  $[H(x_k)]^{-1}$  est symétrique, la formule de mise à jour de l'approximation du Hessien est la suivante :*

$$M_{k+1} = M_k + a_k u_k u_k^T \quad (3.30)$$

*où  $u_k$  est un vecteur de  $\mathbb{R}^n$  et  $a_k$  constante. La matrice de correction  $\Delta_k = a_k u_k u_k^T$  est symétrique et de rang un.*

*Des équations (3.28)-(3.30) on obtient :*

$$s_k = M_k y_k + a_k u_k u_k^T y_k \quad (3.31)$$

et par conséquent

$$\begin{aligned} y_k^T (s_k - M_k y_k) &= a_k y_k^T u_k u_k^T y_k \\ &= a_k (u_k^T y_k)^2 \end{aligned} \quad (3.32)$$

Alternativement, de l'équation (3.31) on a :

$$\begin{aligned} (s_k - M_k y_k) &= a_k u_k u_k^T y_k = a_k (u_k^T y_k) u_k \\ (s_k - M_k y_k)^T &= a_k y_k^T u_k u_k^T = a_k (u_k^T y_k) u_k^T \end{aligned}$$

Si on prend

$$u_k = (s_k - M_k y_k)$$

alors

$$a_k (u_k^T y_k) = 1.$$

Soit en remplaçant :

$$a_k (s_k - M_k y_k)^T y_k = 1$$

et (3.30) devient :

$$M_{k+1} = M_k + \frac{(s_k - M_k y_k)(s_k - M_k y_k)^T}{(s_k - M_k y_k)^T y_k}$$

Cette formule porte le nom de formule SR1 (système de rang 1).

Algorithme de la méthode de correction de rang un

*Étape initiale :*

Soit  $\varepsilon > 0$  (critère d'arrêt). Choisir un point initial  $x_1$  et une matrice symétrique définie positive  $M_1$  quelconque (par exemple  $M_1 = I$ ) poser  $k = 1$ , et aller aux étapes principales.

*Étapes principales :*

*Étape 1 :*

Si  $\|\nabla f(x_k)\| < \varepsilon$  STOP ; sinon, poser  $d_k = -M_k g_k$  et déterminer le pas optimal  $\lambda_k$  solution optimale du problème  $\min f(x_k + \lambda d_k)$ ;  $\lambda \geq 0$ , et poser  $x_{k+1} = x_k + \lambda_k d_k$

*Étape 2 :*

Construire  $M_{k+1}$  comme suit :

$$M_{k+1} = M_k + \frac{(s_k - M_k y_k)(s_k - M_k y_k)^T}{(s_k - M_k y_k)^T y_k}$$

avec

$$\begin{aligned} s_k &= x_{k+1} - x_k \\ y_k &= \nabla f(x_{k+1}) - \nabla f(x_k) \end{aligned}$$

Remplacer  $k$  par  $k + 1$  et aller à l'étape 1.

Cet algorithme a un comportement remarquable dans le cas où  $f$  est une fonction quadratique.

convergence de la méthode de correction de rang un :

Pour un problème quadratique convexe, cette méthode converge au plus dans  $(n + 1)$  itérations et  $(M_{k+1})^{-1} = H$  à condition que  $s_i = M_{k+1}y_i$  pour  $0 \leq i \leq k$ . C'est en effet le cas, comme sera démontré par le théorème suivant.

**Théorème 3.3.1** Soit  $f$  une fonction quadratique, la matrice Hessienne  $H$  définie positive. Considérons la méthode itérative qui, partant d'un point  $x_0$  arbitraire engendre les points

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

où les  $s_k$  sont des vecteurs linéairement indépendants. Pour toute matrice symétrique initiale  $M_0$ , considérons la suite des matrices  $\{M_k\}$  définis par :

$$M_{k+1} = M_k + \frac{(s_k - M_k y_k)(s_k - M_k y_k)^T}{s_k^T (s_k - M_k y_k)^T} \quad (3.33)$$

avec

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

Alors on a :

$$s_i = M_{k+1}y_i \quad \text{pour} \quad 0 \leq i \leq k$$

et dans au plus  $n$  étapes, la suite  $\{x_k\}$  converge vers la solution optimale, et  $M_k$  converge vers l'inverse du Hessien  $H^{-1}$  ( $M_{k+1} = H^{-1}$ ).

**démonstration 3.3.1** Supposons que

$$s_i = M_k y_i \quad \text{pour} \quad 0 \leq i \leq k - 1 \quad (3.34)$$

est vraie et montrons qu'elle est vraie pour  $k$ , si  $0 \leq i \leq k-1$  de l'équation (3.33)

$$M_{k+1}y_i = M_k y_i + u_k (s_k - M_k y_k)^T y_i$$

où

$$u_k = \frac{s_k - M_k y_k}{y_k^T (s_k - M_k y_k)}$$

Puisque  $M_k$  est symétrique, nous pouvons écrire

$$M_{k+1}y_i = M_k y_i + u_k (s_k^T y_i - y_k^T M_k y_i)$$

et de (3.34)

$$M_{k+1}y_i = s_i + u_k (s_k^T y_i - y_k^T s_i) \quad (3.35)$$

pour  $0 \leq i \leq k$  on a

$$y_i = H s_i$$

et

$$y_k^T = s_k^T H$$

Par conséquent pour  $0 \leq i \leq k-1$ , nous prenons

$$s_k^T y_i - y_k^T s_i = s_k^T H s_i - s_k^T H s_i = 0$$

et de l'équation (3.35) on obtient

$$s_i = M_{k+1}y_i \quad \text{pour } 0 \leq i \leq k-1 \quad (3.36)$$

on a

$$s_k = M_{k+1}y_k$$

et avec (3.36), nous pouvons écrire

$$s_i = M_{k+1}y_i \quad \text{pour } 0 \leq i \leq k \quad (3.37)$$

Pour accomplir l'induction, nous notons

$$s_i = M_1 y_i = M_1 H s_i \quad \text{pour } 0 \leq i \leq 0$$

et d'après (3.37)-(3.34) nous pouvons écrire

$$\begin{aligned}
s_i &= M_2 y_i = M_2 H s_i, & \text{pour } 0 \leq i \leq 1 \\
s_i &= M_3 y_i = M_3 H s_i, & \text{pour } 0 \leq i \leq 2 \\
&\vdots & \vdots \\
s_i &= M_{k+1} y_i = M_{k+1} H s_i, & \text{pour } 0 \leq i \leq k
\end{aligned}$$

et comme les vecteurs  $s_i$  sont indépendants, c'est-à-dire l'ensemble  $\{s_0, s_1, \dots, s_K\}$  forme une base de  $\mathbb{R}^{k+1}$  alors,

$$M_{k+1} H = I$$

ou encore,

$$M_{k+1} = H^{-1}.$$

Avantages et inconvénients de la méthode de SR1 :

▼ *Avantages*

- Cette méthode présente l'avantage, que le point  $x_{k+1}$  n'a pas besoin d'être choisi comme le minimum exacte, c'est-à-dire qu'on n'a pas besoin d'effectuer des recherches linéaire exactes.

▼ *Inconvénients*

- Même si la fonction est quadratique, et même si sa Hessienne est définie positive, la matrice  $M_k$  n'est pas forcément définie positive.
- Le dénominateur  $(s_k - M_k y_k)^T y_k$  peut devenir nul ou très petit, ce qui rend le procédé instable c'est-à-dire, la méthode n'est pas bien définie.

*Remarque 3.3.1* la méthode de SR1 est souvent utilisée lorsqu'il n'est pas possible ou qu'il n'est pas nécessaire que  $M_k$  soit définie positive.

### 3.3.4 Méthode de Davidon-Fletcher-Powell (DFP)

Cette méthode a été proposée par Davidon en 1959 et développé plus tard en 1963 par Fletcher et Powell. La formule de mise à jour de DFP est une formule de correction de rang deux. De façon plus précise construisons  $M_{k+1}$  en fonction de  $M_k$  de la forme :

$$M_{k+1} = M_k + A_k + \Delta_k \tag{3.38}$$

Bien sur le choix de  $A_k$  et  $\Delta_k$  doit prendre en considération l'équation de quasi-Newton

$$M_{k+1}y_k = s_k \quad (3.39)$$

Si on écrit  $A_k$  sous la forme  $A_k = a_k u_k u_k^T$  et  $\Delta_k = b_k v_k v_k^T$ , alors (3.39) prend la forme :

$$(a_k u_k u_k^T)y_k + (b_k v_k v_k^T)y_k = s_k - M_k y_k.$$

Comme dans le cas de la formule SR1, un bon choix de  $a_k$ ,  $b_k$ ,  $u_k$  et  $v_k$  conduit à la formule de correction de rang 2 suivante :

$$M_{k+1} = M_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{M_k y_k y_k^T M_k}{y_k^T M_k y_k} \quad (3.40)$$

Algorithme de la méthode de DFP :

**Étape initiale :** Soit  $\varepsilon > 0$ , déterminer le critère d'arrêt. Choisir un point initial  $x_1$  et une matrice symétrique définie positive  $M_1$  quelconque (par exemple  $M_1 = I$ ) poser  $k = 1$ , et aller aux étapes principales

**Étape principales :**

**Étape 1 :** Si  $\|\nabla f(x_k)\| < \varepsilon$  **STOP** ; sinon, poser  $d_k = -M_k g_k$  et déterminer le pas optimal  $\lambda_k$  solution optimale du problème

$\min f(x_k + \lambda d_k)$ ,  $\lambda \geq 0$ . et poser  $x_{k+1} = x_k + \lambda_k d_k$

**Étape 2 :** Construire  $M_{k+1}$  comme suit :

$$M_{k+1} = M_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{M_k y_k y_k^T M_k}{y_k^T M_k y_k}$$

avec

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

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

Remplacer  $k$  par  $k + 1$  et aller à l'étape 1.

Cet algorithme a un comportement remarquable dans le cas où  $f$  est une fonction quadratique.

**Théorème 3.3.2** La formule (3.40) conserve la définie positivité des matrices  $M_k$ , c'est-à-dire, si  $M_k$  est définie positive alors  $M_{k+1}$  est également définie positive.

*démonstration 3.3.2 Soit  $x$  un vecteur de  $\mathbb{R}^n$  on a*

$$\begin{aligned} x^T M_{k+1} x &= x^T M_k x + \frac{(s_k^T x)^2}{s_k^T y_k} - \frac{(y_k^T M_k x)^2}{y_k^T M_k y_k} \\ &= \frac{y_k^T M_k y_k^T x M_k x - (y_k^T M_k x)^2}{y_k^T M_k y_k} + \frac{(s_k^T x)^2}{s_k^T y_k} \end{aligned}$$

*Si on définit le produit scalaire  $\langle x, y \rangle = x^T M_k y_k$ , alors on a*

$$x^T M_{k+1} x = \frac{\langle y_k, y_k \rangle \langle x, x \rangle - \langle y_k, x \rangle^2}{\langle y_k, y_k \rangle} + \frac{(s_k^T x)^2}{s_k^T y_k} \quad (3.41)$$

*Le premier terme du second membre est positive ou nul d'après l'inégalité de Cauchy Schwartz. et pour le deuxième terme on peut faire l'analyse suivante : Puisque le pas est optimal on a la relation d'après l'orthogonalité des directions successives*

$$g_{k+1}^T d_k = 0$$

*et donc*

$$\begin{aligned} s_k^T &= \lambda_k (g_{k+1} - g_k)^T d_k \\ &= \lambda_k g_k^T M_k g_k > 0 \end{aligned}$$

*On a donc*

$$x^T M_{k+1} x \geq 0$$

*Les deux termes dans (3.41) étant positifs. Cette quantité ne peut s'annuler que si les deux termes sont simultanément nuls, le premier terme ne peut s'annuler que si  $x = \lambda y_k$  pour un scalaire  $\lambda \neq 0$ , dans ce cas le deuxième terme est non nul car  $s_k^T x = \lambda s_k^T y_k$ . on a donc bien*

$$x^T M_{k+1} x > 0 \quad \text{pour} \quad x \neq 0.$$

*• Il est important de noter que le résultat ci-dessus est vérifié pour n'importe quel  $k > 0$  pour lequel*

$$s_k^T y_k = s_k^T g_{k+1} - s_k^T g_k > 0 \quad (3.42)$$

*c'est-à-dire si  $s_k^T y_k > 0$ , alors la définie positivité de  $M_{k+1}$  est préservée même dans le cas où la minimisation de  $f(x_k + \lambda d_k)$  est inexacte et donc l'équation de quasi-newton a toujours une solution  $M_{k+1}$ .*

- Si  $f$  est strictement convexe alors l'inégalité (3.42) est vérifiée pour tous  $x_k$  et  $x_{k+1}$

$$\begin{aligned} (\text{car : } f \text{ est strictement convexe} &\Leftrightarrow \forall x_k, x_{k+1} \in \mathbb{R}^n; x_k^T H(x_{k+1}) x_k > 0 \\ &\Rightarrow s_k^T M_{k+1} s_k = s_k^T y_k > 0) \end{aligned}$$

Si  $f$  est non convexe alors l'inégalité (3.42) n'est pas forcément vérifiée.

Dans ce cas on utilise la recherche linéaire de Wolfe ou de Wolfe forte pour vérifier la condition (3.42). En effet

de (wolfe2) on a

$$g_{k+1}^T s_k \geq \sigma g_k^T s_k \Rightarrow y_k^T s_k \geq (\sigma - 1) \lambda_k g_k^T d_k$$

Puisque  $\sigma < 1$  et  $g_k^T d_k < 0$ , on peut écrire

$$y_k^T s_k \geq (\sigma - 1) \lambda_k g_k^T d_k > 0$$

la convergence de la méthode de DFP :

**Théorème 3.3.3** Soit  $f$  une fonction quadratique et  $H_f$  définie positive et pour  $\{s_1, s_2, \dots, s_k\}$  rayons linéairement indépendants, la méthode de DFP converge après  $n$  itérations et  $B_n^{-1} = H$ .

**Théorème 3.3.4** (Directions conjuguées engendrées par méthode de DFP)

a) Si la recherche linéaire utilisée dans la méthode de DFP est exacte et si la fonction objective  $f(x)$  est une fonction convexe et quadratique, alors l'algorithme DFP décrit par la relation

$$M_{k+1} = M_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{M_k y_k y_k^T M_k}{y_k^T M_k y_k}$$

engendre un ensemble des directions conjuguées  $s_0, s_1, \dots, s_k$  c'est-à-dire,

$$s_i^T H s_j = 0 \quad \text{pour } 0 \leq i < j \leq k \quad (3.43)$$

b) Si

$$y_i = H s_i \quad \text{pour } 0 \leq i \leq k \quad (3.44)$$

alors

$$s_i = M_{k+1}y_i \quad (3.45)$$

démonstration 3.3.3 Supposons que

$$s_i^T H s_j = 0 \quad \text{pour } 0 \leq i < j \leq k-1 \quad (3.46)$$

$$s_i = M_k y_i \quad \text{pour } 0 \leq i \leq k-1 \quad (3.47)$$

puis nous montrons l'équations (3.43) et (3.45)

a) on a

$$y_k = H s_k$$

où

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

$$y_k = g_{k+1} - g_k$$

donc nous pouvons écrire

$$\begin{aligned} g_k &= g_{k-1} + H s_{k-1} \\ &= g_{k-2} + H s_{k-2} + H s_{k-1} \\ &= g_{k-3} + H s_{k-3} + H s_{k-2} + H s_{k-1} \\ &\vdots \\ &= g_{i+1} + H(s_{i+1} + s_{i+2} + \dots + s_{k-1}) \end{aligned}$$

Ainsi pour  $0 \leq i \leq k-1$ , nous prenons

$$s_i^T g_k = s_i^T g_{k+1} + s_i^T H(s_{i+1} + s_{i+2} + \dots + s_{k-1}) \quad (3.48)$$

et quand la recherche linéaire utilisée dans la méthode de DFP est exacte, alors  $f(x)$  est minimisée exactement au point  $x_{i+1}$  et ainsi

$$g_{i+1}^T d_i = 0 \quad (3.49)$$

implique

$$\lambda_i g_{i+1}^T d_i = g_{i+1}^T \lambda_i d_i = s_i^T g_{i+1} = 0$$

Maintenant pour  $0 \leq i \leq k-1$ , l'équation (3.46) donne

$$s_i^T H(s_{i+1} + s_{i+2} + \dots + s_{k-1}) = 0 \quad (3.50)$$

et de (3.48)-(3.50), nous obtenons

$$s_i^T g_k = 0$$

Alternativement, d'équations. (3.47)-(3.44) nous pouvons écrire

$$s_i^T g_k = (M_k y_i)^T g_k = (M_k H s_i)^T g_k = s_i^T H M_k g_k = 0$$

De plus, pour éliminer  $M_k g_k$  on utilise l'équation

$$s_k = \lambda_k d_k = -\lambda_k M_k g_k$$

ainsi

$$s_i^T g_k = -\frac{1}{\lambda_k} s_i^T H s_k = 0$$

et depuis  $\lambda_k > 0$ , nous avons

$$s_i^T H s_k = 0 \quad \text{pour } 0 \leq i \leq k-1 \quad (3.51)$$

Maintenant, par la combinaison de (3.46) et (3.51)

$$s_i^T H s_j = 0 \quad \text{pour } 0 \leq i < j \leq k \quad (3.52)$$

Pour accomplir l'induction, nous pouvons écrire

$$s_0^T g_1 = (M_1 y_0)^T g_1 = (M_1 H s_0)^T g_1 = s_0^T H M_1 g_1 = -\frac{1}{\lambda_1} s_0^T H s_1$$

et depuis  $f(x)$  est minimisée exactement au point  $x_1$ , nous avons  $s_0^T g_1 = 0$  et

$$s_i^T H s_j = 0 \quad \text{pour } 0 \leq i < j \leq 1$$

Depuis (3.52) est vérifiée si (3.46) est vérifiée, nous pouvons écrire

$$\begin{array}{ll}
s_i^T H s_j = 0 & \text{pour } 0 \leq i < j \leq 2 \\
s_i^T H s_j = 0 & \text{pour } 0 \leq i < j \leq 3 \\
& \vdots \\
s_i^T H s_j = 0 & \text{pour } 0 \leq i < j \leq k
\end{array}$$

*c'est-à-dire, les directions  $s_0, s_1, \dots, s_k$  forment un ensemble des directions conjuguées.*

b) de (3.47)

$$y_k^T s_i = y_k^T M_k y_i \quad \text{pour } 0 \leq i \leq k-1 \quad (3.53)$$

et d'autre part de (3.44) on a

$$y_k^T s_i = s_k^T H s_i \quad \text{pour } 0 \leq i \leq k-1 \quad (3.54)$$

Depuis  $s_0, s_1, \dots, s_k$ , forme un ensemble des directions conjuguées (de la partie (a)), et de (3.53)-(3.54)

$$y_k^T s_i = y_k^T M_k y_i = s_k^T H s_i \quad \text{pour } 0 \leq i \leq k-1 \quad (3.55)$$

En notant

$$s_k^T = y_k^T M_{k+1} \quad \text{et} \quad H s_i = y_i$$

(3.55) peut être exprimé

$$y_k^T s_i = y_k^T M_k y_i = y_k^T M_{k+1} y_i = 0 \quad \text{pour } 0 \leq i \leq k-1$$

et, en conséquence,

$$s_i = M_k y_i = M_{k+1} y_i \quad \text{pour } 0 \leq i \leq k-1 \quad (3.56)$$

Maintenant, de l'équation

$$s_k = M_{k+1} y_k$$

et par la combinaison de (3.56), nous obtenons

$$s_i = M_{k+1} y_i \quad \text{pour } 0 \leq i \leq k.$$

**Remarque 3.3.2** Pour  $k = n-1$ , des équations (3.44) et (3.45) on peut écrire

$$[M_n H - \alpha I]s_i = 0 \quad \text{pour } 0 \leq i \leq n-1$$

avec  $\alpha = 1$ . En effet, les vecteurs  $s_i$  sont des vecteurs propres qui correspondent à la valeur propre d'unité pour la matrice  $M_n H$ , et puisqu'ils sont linéairement indépendants, nous avons

$$M_n = H^{-1}$$

c'est-à-dire, dans un problème quadratique  $M_{k+1}$  devient le Hessien à l'itération  $n-1$  :

Avantages de la méthode de DFP :

1- Pour des fonctions quadratiques (avec une recherche linéaire exacte) :

- L'algorithme converge dans au plus  $n$  étapes avec  $M_{n+1} = H^{-1}$ .
- Elles engendrent des directions conjuguées.

2- Pour les fonctions quelconques :

- la matrice  $M_k$  reste définie positive, ce qui est nécessaire pour que la direction soit une direction de descente.

### 3.3.5 Méthode de Broyden-Fletcher-Goldfard-Shanno (BFGS [1970])

La méthode de BFGS est une méthode quasi-newtonienne bien connue pour résoudre des problèmes d'optimisation sans contraintes, elle a été proposée par : Broyden, Fletcher, Goldfarb, et Shanno. Elle est devenue une méthode de choix pour les ingénieurs et les mathématiciens qui sont intéressés à résoudre des problèmes d'optimisation.

Une autre manière d'imposer à  $M_{k+1}$  d'être proche de  $M_k$  est de minimiser l'"écart" entre  $M_{k+1}$  et  $M_k$ , toujours en requérant que  $M_{k+1}$  soit symétrique et vérifie l'équation de quasi-Newton. On est donc conduit à considérer le problème en la variable matricielle  $M \in IR^{n \times n}$  suivant :

$$\begin{cases} \min \text{"écart"}(M, M_k) \\ y_k = M s_k \\ M = M^T \end{cases} ; M \in IR^{n \times n} \quad (3.57)$$

On dit alors que la matrice est obtenue par une approche variationnelle.

Notations :

Pour indiquer qu'une matrice carrée  $M$  est symétrique semi-définie positive (resp. définie positive), on notera  $M \geq 0$  (resp  $M > 0$ ).

L'ensemble des matrices symétriques d'ordre  $n$  est noté  $S^n$ ,

$$S_+^n := \{M \in S^n : M \geq 0\}$$

et

$$S_{++}^n := \{M \in S^n : M > 0\}.$$

Il est souvent intéressant d'imposer également la définie positivité des matrices  $M_k$ , car pour que  $d_k = -M_k^{-1}g_k$  soit une direction de descente, on a besoin que  $M_k$  soit une matrice symétrique, définie positive (en effet : on sait que, toute matrice symétrique réelle et définie positive est inversible, et son inverse est elle aussi définie positive, c'est-à-dire on peut écrire :  $g_k^T d_k = -g_k^T M_k^{-1} g_k < 0$  donc,  $d_k$  est une direction de descente).

Cette condition définissant un ensemble ouvert ne peut être utilisée directement comme contrainte dans le problème définissant  $M_{k+1}$  ; et dans ce but, on commence par introduire une fonction :  $\psi : S^n \rightarrow \mathbb{R}$  dont le domaine est  $S_{++}^n$  et qui forme une "barrière" au bord du cône  $S_{++}^n$  (elle tend vers l'infini lorsque son argument se rapproche du bord de  $S_{++}^n$ ) ainsi qu'à l'infini :

$$\psi(\Upsilon) = \text{tr}\Upsilon + \text{ld}\Upsilon \tag{3.58}$$

où la fonction log-déterminant  $\text{ld} : S^n \rightarrow \mathbb{R} \cup \{+\infty\}$  est définie en  $\Upsilon \in S^n$  par

$$\text{ld}(\Upsilon) = \begin{cases} -\log \det \Upsilon & \text{si } \Upsilon \in S_{++}^n \\ +\infty & \text{sinon} \end{cases}$$

Les propriétés annoncées de  $\psi$  peuvent se voir sur son expression suivante.

Si on note  $\{\alpha_i\}$  les valeurs propres de  $\Upsilon$ , on a  $\text{tr}\Upsilon = \sum_{i=1}^n \alpha_i$ ,  $\det \Upsilon = \prod_{i=1}^n \alpha_i$  et donc

$$\psi(\Upsilon) = \sum_{i=1}^n (\alpha_i - \log \alpha_i), \quad \text{si } \Upsilon \in S_{++}^n \tag{3.59}$$

Étant donné l'allure de la fonction  $t \in \mathbb{R}_{++} \mapsto t - \log t$ ,  $\psi(\Upsilon)$  tend vers l'infini si l'une des valeurs propres de tend vers zéro ou vers l'infini.

$(\exists j \in \{1, \dots, n\}; \lim_{\alpha_j \rightarrow 0 \text{ ou } \infty} \psi(\Upsilon) = \infty)$ .

La formule (3.59) montre aussi que l'unique minimiseur de  $\psi$  est  $\Upsilon = 1$  la matrice identité ( $\alpha_i = 1$  pour tout  $i$ ).

Si  $M_k$  est une matrice réelle symétrique alors :

la matrice  $M_k$  est définie positive si et seulement s'il existe une matrice définie positive  $A_k$  telle que  $A_k^2 = M_k$ , dans ce cas, la matrice définie positive  $A_k$  et on peut la noter  $A_k = M_k^{\frac{1}{2}}$  est unique.

en effet :

Si  $M_k$  est une matrice réelle symétrique alors on peut écrire

$$U^T M_k U = A$$

où  $U$  est vérifiée

$$U^T U = U U^T = I$$

et est une matrice diagonalisée où les éléments du diagonale sont les valeurs propres de  $M_k$  qui sont strictement positives car  $M_k$  est définie positive. Donc on peut écrire

$$M_k = U \Lambda U^T = U \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} U^T = (U \Lambda^{\frac{1}{2}} U^T) (U \Lambda^{\frac{1}{2}} U^T) = M_k^{\frac{1}{2}} M_k^{\frac{1}{2}}.$$

On veut dire, qu'il faut trouver une matrice  $M$  symétrique et définie positive être proche de  $M_k$  c'est-à-dire

$$M \cong M_k^{\frac{1}{2}} M_k^{\frac{1}{2}}$$

implique

$$M_k^{-\frac{1}{2}} M M_k^{-\frac{1}{2}} \cong I$$

Donc afin de minimisée l'écart entre  $M$  et  $M_k$ , on va chercher à ce que  $M_k^{-\frac{1}{2}} M M_k^{-\frac{1}{2}}$  soit proche de  $I$ , et ceci peut être obtenu en minimisant  $\psi(M_k^{-\frac{1}{2}} M M_k^{-\frac{1}{2}})$ ,

si bien que l'on aura  $M_{k+1}$  proche de  $M_k$  en résolvant :

$$\begin{cases} \min \psi(M_k^{-\frac{1}{2}} M M_k^{-\frac{1}{2}}) \\ y_k = M s_k \\ M \in S_{++}^n \end{cases} \quad (\text{contrainte implicite}). \quad (3.60)$$

$$\text{Si } s_k = 0 \text{ alors } \begin{cases} \text{si } y_k \neq 0 \text{ alors (3.60) n'a pas de solutions} \\ \text{ou bien} \\ \text{si } y_k = 0 \text{ alors la solution du (3.60) est } M = M_k \end{cases}$$

Sinon, le cas non trivial où  $s_k \neq 0$  est examiné dans la proposition suivante.

**Proposition 3.3.1** *Supposons que  $M_k$  soit symétrique définie positive et que  $s_k \neq 0$ . Alors, le problème (3.60) a une solution si et seulement si  $y_k^T s_k > 0$ .*

*Sous cette condition la solution  $M_{k+1}$  de (3.60) est unique et est donnée par l'une des formules suivantes :*

$$M_{k+1} = M_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{M_k s_k s_k^T M_k}{s_k^T M_k s_k} \quad (3.61)$$

$$B_{k+1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) B_k \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) + \left(I - \frac{s_k s_k^T}{y_k^T s_k}\right) \quad (3.62)$$

où on a noté  $B_k := M_k^{-1}$  et  $B_{k+1} := M_{k+1}^{-1}$ .

**Algorithme de la méthode de BFGS :**

**Étape initiale :**

*Soit  $\varepsilon > 0$ , déterminer le critère d'arrêt. Choisir  $x_1$  point initial et  $M_1$  définie positive quelconque (par exemple  $M_1 = I$ ).*

*Poser  $k = 1$  et aller aux étapes principales*

**Étapes principales :**

**Étape 1 :**

*Si  $\|\nabla f(x_k)\| < \varepsilon$  STOP ; sinon, poser  $d_k = -M_k g_k$  et déterminer le pas optimal  $\lambda_k$  solution optimale du problème  $\min f(x_k + \lambda d_k)$ ,  $\lambda \geq 0$  et poser  $x_{k+1} =$*

$$x_k + \lambda_k d_k$$

Étape 2 :

Construire  $M_{k+1}$  comme suit :

$$M_{k+1} = M_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{M_k s_k s_k^T M_k}{s_k^T M_k s_k}$$

avec

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

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

Remplacer  $k$  par  $k + 1$  et aller à l'étape 1.

**Proposition 3.3.2 (Propriété algébrique)** Soit  $H$  une matrice symétrique et  $s_1, s_2, \dots, s_p$  des directions non nulles, conjuguées par rapport à cette matrice (c'est-à-dire :  $s_i^T H s_j = 0$ , pour  $i \neq j$  et  $s_i^T H s_i > 0$ ). On se donne une matrice  $M_1$  symétrique définie positive et on définit

$$M_{i+1} = M_i + \frac{y_i y_i^T}{y_i^T s_i} - \frac{M_i s_i s_i^T M_i}{s_i^T M_i s_i}, \quad \text{pour } i = 1, \dots, p$$

Alors

(i)  $M_{k+1} s_i = H s_i$ , pour  $k = 1, \dots, p$  et  $i = 1, \dots, k$ ,

(ii) si  $p = n$ , on a  $M_{n+1} = H$ .

**démonstration 3.3.4 (i)** La première propriété se démontre par récurrence.

Elle est vérifiée pour  $k = 1$  :  $M_2 s_1 = H s_1$  (c'est l'équation de quasi-Newton vérifiée par  $M_2$  où  $y_1 = H s_1$ ). Supposons qu'elle le soit pour un  $k = 1, \dots, l-1$ , avec  $2 \leq l \leq p$ , et démontrons la pour  $k = l$ .

Par conjugaison et récurrence,  $s_i^T H s_i = 0$  et  $s_i^T M_i s_i = s_i^T H s_i = 0$ .

Dès lors, la formule de BFGS (3.61) donne  $M_{i+1} s_i = M_i s_i = H s_i$ . Si  $i = l$ , on a  $M_{i+1} s_i = H s_i$  (c'est l'équation de quasi-Newton vérifiée par  $M_{i+1}$  où  $y_i = H s_i$ ).

Donc on a  $M_{k+1} s_i = H s_i$ , pour  $k = 1, \dots, p$  et  $i = 1, \dots, k$ .

(ii) Si  $p = n$ , alors  $M_{n+1}$  prend la même valeur que  $H$  sur les  $n$  vecteurs linéairement indépendants  $s_1, s_2, \dots, s_n$ , c'est-à-dire de (i) on peut écrire

$$\begin{aligned}
M_{n+1}s_1 &= Hs_1 \\
M_{n+2}s_2 &= Hs_2 \\
&\vdots \\
M_{n+1}s_n &= Hs_n
\end{aligned}$$

et puisque  $s_1, s_2, \dots, s_n$  sont linéairement indépendant, nous avons

$$M_{n+1} = H$$

### Propriétés

- La condition  $y_k^T s_k > 0$  assure la défini-positivité de  $M_k$ .
- La condition  $y_k^T s_k > 0$  est satisfaite lorsqu'on utilise la recherche linéaire de Wolfe.
- Les directions  $s_1, s_2, \dots, s_n$  engendrées par la formule de BFGS sont conjuguées.
- Dans le cas quadratique la méthode de BFGS possède les mêmes propriétés que DFP.
- Dans le cas non quadratique, il faut procéder à des réinitialisations périodiques pour assurer la convergence globale.

### 3.3.6 Méthodes de la classe Broyden

Une méthode de classe Broyden est une méthode de quasi-Newtonienne où l'approximation de l'inverse du Hessien prendre la formule suivant :

$$M_{k+1} = M_k - \frac{M_k s_k s_k^T M_k}{s_k^T M_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi (s_k^T M_k s_k) v_k v_k^T \quad (3.63)$$

$$\text{avec } \phi \in [0, 1], \quad v_k^T = \left[ \frac{y_k}{y_k^T s_k} - \frac{M_k s_k}{s_k^T M_k s_k} \right]$$

- ▼ Si  $\phi = 0$  on obtient la méthode BFGS.
- ▼ Si  $\phi = 1$  on obtient la méthode DFP.
- ▼ Si  $\phi = \frac{s_k^T y_k}{s_k^T y_k - y_k M_k y_k}$  on obtient la méthode de SR1.

### Propriétés des méthodes de la classe de broyden

- Nous pouvons donc récrire (3.63) comme "combinaison linéaire" de deux méthodes, c'est-à-dire,

$$M_{k+1} = (1 - \phi) M_{k+1}^{BFGS} + \phi M_{k+1}^{DFP},$$

*Cette relation indique que tous les membres de la classe de Broyden satisfont l'équation de quasi-newton, puisque les matrices de BGFS et de DFP elles-mêmes satisfont cette équation et aussi elle conserve la définie positivité des matrices  $M_k$ .*

## Conclusion

*En conclusion, ce mémoire a abordé les principales méthodes numériques utilisées pour résoudre des problèmes d'optimisation sans contraintes. Nous avons présenté les concepts fondamentaux, les conditions d'optimalité ainsi que les algorithmes les plus courants tels que les méthodes de descente, de la plus forte pente, de Newton et la méthode quasi-Newtonienne.*

*Il convient de noter que les théories liées à l'optimisation sans contraintes constituent une pierre angulaire pour des domaines plus avancés et complexes tels que l'optimisation sans contraintes, l'optimisation non linéaire et l'optimisation multi-objectifs. Par conséquent, une compréhension approfondie des fondements théoriques de ce domaine est essentielle pour tout chercheur aspirant à s'engager dans ces domaines spécialisés.*

*Les méthodes d'optimisation sans contraintes constituent un outil puissant avec de nombreuses applications dans divers domaines tels que l'ingénierie, l'économie, la finance ou encore le machine learning. Avec l'avènement du Big Data et de l'intelligence artificielle, leur rôle est amené à prendre encore plus d'importance dans la résolution de problèmes complexes de grande dimension.*

# BIBLIOGRAPHIE

- [1] *Antoine, X., Dreyfuss, P., Privat, Y. (2007). Introduction à l, optimisation : aspects théoriques, numériques et algorithmes. Institut National Polytechnique de Lorraine (INPL), Ecole Nationale Supérieure d'Electricité et de Mécanique, Ecole Nationale Supérieure des Mines de Nancy, Département de Génie Industriel, et Institut Elie Cartan Nancy (IECN), Université Henri Poincaré Nancy, 1, 2006-2007.*
- [2] *Bazaraa, M. S., Sherali, H. D., Shetty, C. M. (2006). Nonlinear programming : theory and algorithms. John wiley sons.*
- [3] *Bergounioux, M. (2003). Mathématique et automatique : de la boucle ouverte à la boucle fermée. J3eA, 2, 002.*
- [4] *Dai, Y. H., Yuan, Y. (1995). A nonlinear conjugate gradient method with nice global convergence properties. Research report ICM-95-038, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences.*
- [5] *Dai, Y. H., Yuan, Y. (1996). Convergence properties of the conjugate descent method. Advances in Mathematics, 25(6), 552-562.*
- [6] *Dai, YH et Yuan, Y. (1998). Quelques propriétés d'une nouvelle méthode de gradient conjugué. Dans Advances in Nonlinear Programming : Actes de*

- la 96e Conférence internationale sur la programmation non linéaire (pp. 251-262). Springer États-Unis.
- [7] Dai, Y. H., Yuan, Y. (1999). A nonlinear conjugate gradient with a strong global convergence property. *SIAM Journal of Optimization*, 10.
- [8] Fletcher, R., Reeves, C. M. (1964). Function minimization by conjugate gradients. *The computer journal*, 7(2), 149-154.
- [9] Gilbert, J. C., Nocedal, J. (1992). Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on optimization*, 2(1), 21-42.
- [10] Gilbert, J. C. (2007). *Éléments d'Optimisation Différentiable : Théorie et Algorithmes, notes de cours*. École Nationale Supérieure de Techniques Avancées, Paris.
- [11] Goldstein, A. A., Price, J. F. (1967). An effective algorithm for minimization. *Numerische Mathematik*, 10, 184-189.
- [12] Hager, W. W., Zhang, H. (2006). A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1), 35-58.
- [13] Hu, Y. F., Storey, C. (1991). Global convergence result for conjugate gradient methods. *Journal of Optimization Theory and Applications*, 71(2), 399-405.
- [14] Mendelson, B. (1990). *Introduction to topology*. Courier Corporation.
- [15] Minoux, M. (2008). *Programmation mathématique. Théorie et algorithmes*. Lavoisier.
- [16] Mottelet, S. (2003). *RO04/TI07-Optimisation non-linéaire*. Université de technologie de Compiègne.
- [17] Nocedal, J., Wright, S. J. (Eds.). (1999). *Numerical optimization*. New York, NY : Springer New York.
- [18] Powell, M. J. (1986). Convergence properties of algorithms for nonlinear optimization. *Siam Review*, 28(4), 487-500.
- [19] Veinott, A. F. (1968). *Nonlinear Programming*. J. Abadie (ed.). New York : John Wiley, 1967. Pp. xiii, 316.

## *Le Résumé*

*Dans ce travail, on s'intéresse à résoudre un problème d'optimisation sans contraintes avec des recherches linéaires inexactes.*

*Nous avons résolu ce problème par plusieurs méthodes numériques (méthode du gradient conjugué, méthode de la plus forte pente, et les méthodes quasi-Newtonienne)*

## *Abstract*

*In this work, we are interested in solving an unconstrained optimization problem with inexact line searches. We have solved this problem using several numerical methods (conjugate gradient method, steepest descent method, and quasi-Newtonian methods).*

## *Les Mots clés*

*Optimisation sans contraintes, algorithmes, Convexité, différentiabilité, matrice Hessienne, recherche linéaire inexacte, règle d'Armijoo, règle de Wolfe, règle de Goldstein et Price, méthode de Gradient conjugué, méthode de la plus forte pente, méthode Quasi-Newtonienne.*