RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de L'enseignement Supérieur et de la Recherche Scientifique
UNIVERSITÉ IBN KHALDOUN TIARET
FACULTÉ DE MATHÉMATIQUES ET DE L'INFORMATIQUES
Département de Mathématiques

MÉMOIRE DE MASTER

Présenter en vue de l'obtention du diplôme de master

Spécialité :
« Mathématiques»

Option :
« Analyse fonctionnelle et Application»

Présenté Par :
FEKIH Sabrine
KESSAR Djamila

Sous L'intitulé :

Utilisation des recherches linéaires pour résoudre un problème d'optimisation

Soutenu publiquement le 03/07/ 2023
à Tiaret devant le jury composé de :

Mr BENALLOU Mohamed	MCB	Université de Tiaret	Président
Mr REZZOUG Nadir	MCB	Université de Tiaret	Encadreur
Mr MAAZOUZ Kada	MCA	Université de Tiaret	Examinateur

Année universitaire :2022/2023

★ Remerciements ★

Avant tout, on remercie **ALLAH** tout-Puissant de nous avoir donné le courage et la volonté d'accomplir ce travail.
Nous tenons à adresser nos remerciements les plus chaleureux et notre profonde gratitude à notre encadreur Monsieur **N.Rezzoug**. C'est grâce à sa disponibilité, ses conseils, ses orientations et ses encouragements que nous pouvons mener à bien ce travail.
Un remerciement spécial et sincère au Monsieur **M.Benallou** qui nous a fait l'honneur d'accepter la présidence du jury de ce mémoire.
Sincères remerciements vont également à Monsieur **K.Maazouz**, qui en l'honneur de participer au jury de soutenance en tant qu'examinateur. Nous ne pouvons oublier de remercier nos professurs sans exception pour leur aide et patience, tout au long de nos études.

★ *À vous tous un grand merci* ★

★ Dédicace ★

Je dédie ce travail

À ma mére **Naima** qui a élevé, fatiqué et pratiqué la maternité et la paternité en même temps.

À l'âme de ma grand pére **Alhabib** le pére de ma mére et l'âme de mon frère **Ahmida, Bakhalad**, que le **ALLAH** ait pitié d'eux.

À mon petit ange ma soeur **Wiam**.

À mon frère **Morsli**.

À ma grande mére **Fatma**, la maman de ma mére.

À ma tante **Fatiha**, merci.

À mes tantes **Zahia, Aicha et Fatima**.

À les petits enfants **Bouchra, Hiba, Abdelrahman, Aya, Ali, Youcef , Ritej, Merieme, Wissam et Ikhlas.**

À mon père **Larbi**.

À mes cousins **Ahmed et M'hamed**.

À toute la famille **FEKIH**.

À mon binôme **Djamila**.

À mes enseignants **Sahli Abdelsamed et Bourace Mbarek.**

"Sabrine"

Je dédie ce modeste travail

À **mes chers parents**, pour leur sacrifices, afin de continue mes études.

À **mon grand père** le père de ma mère, je t'aime que ALLAH preserver votre santé et prologer votre vie.

À ma soeur **Fatma** et ses enfants **Amine et Ayman**.

À mes frères **Mohamed, Houari et AEK**.

À mes oncles **AbdElhadi, AEK et Houari**.

À mes tantes **Abdiya, Mesouda, Djamila, Yamina et Aicha** et leurs enfants **Anes, Adem, fatima, SidAhmed, Dayaa, Ritej, Wassim, Moustapha, Khaled et Yacine**.

À mes cousines **Fatiha, Naima, Sabahe** et ses enfants **Assile, Amir et Soulef**.

À mes cousins **Mohamed, Abed, Abdou, Walide, Younes et Boudali**.

À l'âmes de ma grand père et mes grandes mères que ALLAH pitié d'eux.

À tous la famille **KESSAR**.

À mon binôme **Sabrine**.

Un grand remerciment à tous mes professeurs du primaire, secondaire et universitaire.

◀ ***"K.Djamila"*** ▶

★ Résumés ★

Le but de ce mémoire est la résolution d'un problème d'optimisation sans contraintes en utilisant la méthode du gradient conjugué, et comme cette méthode est basée sur les recherches linéaires, nous avons présenter trois types : recherche linéaire inexacte d'Armijo, de Goldstein et de Wolf, en particulier nous étudions le cas non monotone qui donne de bons résultats pour la convergence.

★ Abstract ★

The purpose of this memoire is to solve a problem of optimization without constraints using the conjugate gradient method, and as this method is based on linear searches, we introduce three types : inaccurate linear search of Armijo Goldstein and Wolf, in particular we study the nonmonotonous case that gives good results for convergence

ملخص

الغرض من هده المذكرة هو حل مشكل أمثلة دون قيود باستخدام طريقة التدرج المقترن، وبما أن هذه الطريقة تستند إلى البحث الخطي، نقدم ثلاثة أنواع: البحث الخطي غير الدقيق عن Armigo,Goldsteinوwolfe ، على وجه الخصوص ندرس الحالة غير رتيبة التي تعطي نتائج جيدة للتقارب.

Table des matières

Notation

$\mathbb{R}$	Ensemble des nombres réels.
$int(S)$	interieur de S.
$H(x_1...x_{k+1})$	simplexe.
$\mathbb{R}^n$	$\overbrace{\mathbb{R} \times \mathbb{R} \times ... \times \mathbb{R}}^{nfois}$.
$\mathbb{R}_+$	L'ensemble des nombres réels positifs.
$\mathbb{C}$	L'ensemble des nombres complexes.
$H(x)$	Matrice hessienne.
d_k	Direction de descente.
λ_k	Pas de minimisation.
$f(.)$	Fonction objective.
$\lVert x \rVert$	Norme d'un vecteur x.
$\langle , \rangle$	Produit scalaire.
x^T	Transposé d'un vecteur x.
$f(x) = o(g(x))$	$\lim \dfrac{f(x)}{g(x)} = 0$.
∇f	Gradient de f.
$M_{m,n}$	Matrice de m lignes et n colonnes.
$C(\mathbb{R}^n, \mathbb{R})$	L'espace des fonctions continues de $\mathbb{R}^n$ dans $\mathbb{R}$.
H^{-1}	Matrice inverse.
$\dfrac{\partial f}{\partial x}$	Dérivée partielle de f suivant x.
$\dfrac{\partial^2 f}{\partial x_i \partial x_j}$	Dérivée partielle seconde de f.
$(PQSC)$	Problème quadratique sans contraintes.
(PNQSC)	Probléme non quadratique sans contraintes .
(FR)	Fletcher Reeves.
(PRP)	Polak-Ribière-Polyak.
(DY)	Dai-Yuan.
(HZ)	Hestenes-Stiefel.

★ Introduction ★

L'optimisation est un outil important utilisé dans les sciences, l'ingénierie, l'économie, la gestion, l'industrie et d'autres domaines. Pour utiliser cet outil, nous devons d'abord identifier un objectif, une mesure quantitative de la perfomance de l'énergie potentielle, ou toute quantité ou combinaison de quantité pouvant être présentées par un seul nombre. L'objectif dépent de certaines caractéristiques du système, appelées variables ou incconues. Notre but est de trouver les valeurs des variables qui optimisent l'objectif. Souvent, les variables sont restreintes ou contraintes d'une manière ou d'une autre. Comme l'a souligné le professeur Yuqi He de l'Université de Harvard, mombre de la National Academy of Engineering des États-Unis, l'optimisation est la pierre angulaire du développemant de la civilisation.

En termes mathématiques, l'optimisation est la minimisation ou la maximisation d'une fonction soumise à des contraintes sur ces variables. Nous utilisons la notation suivante :
$x \in X \subset \mathbb{R}^n$ est le vecteur de variables, aussi appelées inconnues ou paramètres,
f est la fonction objectif, une fonction de x que l'on veut maximiser ou minimiser.
En utilisant cette notation, le problème d'optimisation peut être écrit comme suit

$$\min_{x \in X} f(x) \qquad\qquad (P)$$

En particulier, ce qui nous intéresse dans notre mémoire, si $X = \mathbb{R}^n$, le problème (P) est appelé un problème d'optimisation sans contraintes. Pour maximiser f ou lieu de le minimiser, il suffit de minimiser $-f$ dans la formule de (P).

L'objectif de ce travail est l'utilisation des recherches linéaires pour résoudre un problème d'optimisation. Ce travail est réparti en 3 chapitres.
Dans le premier chapitre, nous rappelons des définitions, propriétés et quelques théorèmes sur la dérivée partielle, la dérivée directionnelle, la différentiabilité, le gradient, la matrice hessiene, la convexité et aussi les formules de taylor.
Dans le deuxième chapitre nous présentons les recherches linéaires inexactes, en particulier les recherches d'Armijo, de Goldstein et de Wolfe, le cas non monotone est aussi introduit dans ce chapitre.
Le troisième chapitre est consacré à la méthode de gradient conjuguéé, commençons par une déscription de la méthode puis passons à l'algorithme et les avantages de cette méthode. À la fin de cette chapitre, nous étudions la convergence dans les deux cas monotone et non monotone.

Chapitre 1

Préliminaires

1.1 Dérivée partielle

Définition 1.1.1. *Soient $n \in \mathbb{N}^*, U$ un ouvert de $\mathbb{R}^n$ et f une fonction de $\mathbb{R}^n$ dans $\mathbb{R}$ définie sur U.*
La fonction f est dite d'avoir une dérivée partielle à $a = (a_1, ..., a_n)$ par rapport à la ième variable notée $\dfrac{\partial f}{\partial x_i}(a)$ si la limite

$$\lim_{t \to a_i} \frac{f(a_1, ..., a_{i-1}, t, a_{i+1}, ..., a_n) - f(a)}{t - a_i}$$

existe.
On note également cette limite par

$$\partial_i f(a) \quad ou \quad D_i f(a).$$

1.2 Différentiabilité

Définition 1.2.1. *1) Soit U un ouvert de $\mathbb{R}^n$ et soit la fonction*

$$f : U \subset \mathbb{R}^n \to \mathbb{R}.$$

On dit que f est différentiable en $a \in U$ s'il existe une application L linéaire continue de $\mathbb{R}^n$ dans $\mathbb{R}$ telle que

$$\lim_{h \to 0} \frac{f(a+h) - f(a) - L(h)}{\|h\|} = 0 \tag{1.1}$$

où $h = (h_1, ..., h_n)$.
2) Lorsque l'application L existe, elle s'appelle la différentielle de f en a, on la note $Df(a)$, ainsi (1.1) devient

$$\lim_{h \to 0} \frac{f(a+h) - f(a) - Df(a)(h)}{\|h\|} = 0.$$

3) Si f est différentiable en tout point de U on dit que f est différentiable sur U.

Proposition 1.2.1. *Soient U un ouvert et $f : U \subset \mathbb{R}^n \to \mathbb{R}$, une fonction. Si $f \in C^1(U)$ alors f est différentiable sur U.*

1.3 Gradient

Si les dérivées partielles $\dfrac{\partial f}{\partial x_i}$ existent pour tout i, le gradient de f est défini de la façon suivante.

Définition 1.3.1. *On note par*

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

le gradient de f au point $x = (x_1, ..., x_n)$.
Le gradient joue un role essentiel dans le développement et l'analyse des algorithmes d'optimisation.

Proposition 1.3.1. *([20]) (Gradient de la composée)*
Soit deux ouverts $\Omega \subset \mathbb{R}^n$ et $U \subset \mathbb{R}$ et deux fonctions $f : \Omega \to \mathbb{R}$ et $g : U \to \mathbb{R}$ avec en plus $f(\Omega) \subset U$ (on peut alors définir $g \circ f : \Omega \to \mathbb{R}$). Supposons que f et g sont de classe C^1. Alors $g \circ f$ est aussi de classe C^1 avec

$$\nabla(g \circ f)(x) = g'(f(x))\nabla f(x), \quad \forall x \in \Omega.$$

1.4 Dérivée directionnelle

Définition 1.4.1. *Soient $n \in \mathbb{N}^*, U$ un ouvert de $\mathbb{R}^n, f : U \to \mathbb{R}$ et $v \in \mathbb{R}^n$. La fonction f est dite d'avoir une dérivée directionnelle au point $a \in U$ dans la direction v et on le note $D_v f(a)$ si la limite*

$$\lim_{h \to 0} \frac{1}{h}[f(a+hv) - f(a)]$$

existe.

Proposition 1.4.1. *Soient $n \in \mathbb{N}^*, U \subset \mathbb{R}^n$ un ouvert de $\mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R}$ différentiable sur U. Alors*

$$D_v f(a) = \nabla f(a)v.$$

1.5 Matrice Hessienne

Si les dérivées partielles $\left(\dfrac{\partial^2 f}{\partial x_i \partial x_j}\right)$ existent pour tout $i, j \in N^*$, la matrice hessienne définie comme suit.

Définition 1.5.1. *Soit U un ouvert de $\mathbb{R}^n$ et $f : U \to \mathbb{R}$ une fonction dont toutes les dérivées partielles d'ordre deux.*
Alors la matrice

$$H(x) = \begin{pmatrix} \dfrac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \dfrac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f(x)}{\partial x_2 \partial x_2} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f(x)}{\partial x_n \partial x_1} & \dfrac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{pmatrix}.$$

est appelée matrice hessienne.

Théorème 1.5.1. ***Schwartz*** *Soit $\Omega \subset \mathbb{R}^n$ un ouvert. Si $f \in C^2(\Omega)$ alors $\nabla^2 f(x) = H(x)$ est une matrice symétrique $\forall x \in \Omega$.*

Exemple 1.5.1. *Soit $f(x_1, x_2) = x_1^2(x_1 + x_2)$.*
On a

$$\nabla f(x) = \begin{pmatrix} 3x_1^2 + 2x_1x_2 \\ x_1^2 \end{pmatrix},$$

alors

$$H(x) = \begin{pmatrix} 6x_1^2 + 2x_2 & 2x_1 \\ 2x_1 & 0 \end{pmatrix}.$$

Définition 1.5.2. *Soit M une matrice symétrique réelle d'ordre n. M est dite semi-définie positive si elle vérifie l'une des deux propriétés équivalentes suivantes*
1. Pour tout vectur non nul x à n éléments réels , nous avons

$$x^T Mx \geq 0.$$

2. Toutes les valeurs propres de M sont positives ou nulles.

Définition 1.5.3. *Soit M une matrice symétrique dont les éléments sont des nombres réels. M est dite définie positive si pour tout vecteur $x \in \mathbb{R}^n$ non nul on a $x^T Mx > 0$.*

1.6 Formules de Taylor

1.6.1 Formule de Taylor avec reste intégrale

Théorème 1.6.1. *Soit U un ouvert de $\mathbb{R}^n$ et $a, h \in \mathbb{R}^n$ tel que $[a, a+h] \subset U$. Soit $f : \mathbb{R}^n \to \mathbb{R}$.une fonction deux fois différentiable sur U. La formule de Taylor-Lagrange alors on a*

$$f(a+h) = f(a) + \sum_{i=1}^{n} h_i \frac{\partial f}{\partial x_i}(a) + \sum_{i,j=1}^{n} h_i h_j \int_0^1 (1-t) \frac{\partial^2 f}{\partial x_i \partial x_j}(a+th)dt.$$

1.6.2 Formule de Taylor-Lagrange

Théorème 1.6.2. *Soit U un ouvert de $\mathbb{R}^n$ et $a, h \in \mathbb{R}^n$ tels que $[a, a+h] \subset U$. Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction deux fois différentiable sur U. La formule de Taylor-Lagrange d'ordre 2 est*

$$f(a+h) = f(a) + \sum_{i=1}^{n} h_i \frac{\partial f}{\partial x_i}(a) + \frac{1}{2} \sum_{i,j=1}^{n} h_i h_j \frac{\partial^2 f}{\partial x_i \partial x_j}(a+\theta h) \quad avec \quad \theta \in]0, 1[.$$

1.6.3 Formule de Taylor-Young

Théorème 1.6.3. *Soit U un ouvert de $\mathbb{R}^n$ et $a \in U$. Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction de classe $C^2(U)$, la formule de Taylor-young d'ordre 2 est*

$$f(a+h) = f(a) + \sum_{i=1}^{n} h_i \frac{\partial f}{\partial x_i}(a) + \frac{1}{2} \sum_{i,j=1}^{n} h_i h_j \frac{\partial^2 f}{\partial x_i \partial x_j}(a) + \varepsilon(h)\|h\|^2.$$

1.7 Inégalité des accroissements finis

Théorème 1.7.1. *Soit f une application continue sur un segment $[a,b] \subset \mathbb{R}$ et dérivable sur $]a,b[$. S'il existe $M \in \mathbb{R}$ tel que $f'(x) \leq M$ pour tout $x \in]a,b[$, alors*

$$f(b) - f(a) \leq M(b-a).$$

1.8 Convexité

Définition 1.8.1. *1) Soit S un sous ensemble de $\mathbb{R}^n$, on dit que S est convexe si et seulement si*

$$\forall x_1, x_2 \in S, \forall \lambda \in [0,1] \, on \, a \;\; \lambda x_1 + (1-\lambda) x_2 \in S.$$

2) Soient $x_1, x_2, ..., x_k \in \mathbb{R}^n$ et $\lambda_j \in \mathbb{R}$ tels que

$$\lambda_i \geq 0 \;\; et \;\; \sum_{j=1}^{k} \lambda_j = 1.$$

Alors toute expression de la forme $\sum_{j=1}^{k} \lambda_j x_j$ s'appelle combinaison convexe des points x_j ou barycentre.

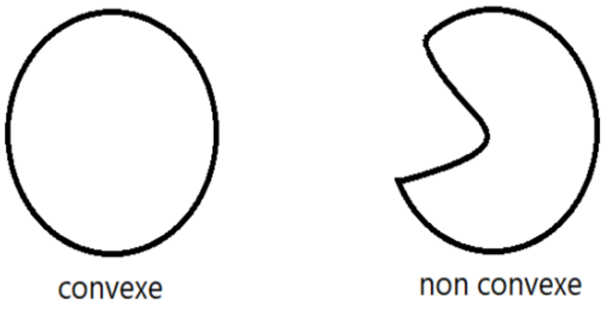


FIGURE 1.1 – ensemble convexe et non convexe

Définition 1.8.2. *Soit $S \subset \mathbb{R}^n$, on appelle envelloppe convexe de S et on le note $H(S)$, l'ensemble de toutes les combinaisons convexes de S, en d'autre termes*

$$x \in H(S) \Leftrightarrow \exists x_1, x_2, ... x_k \in S \;\; et \;\; \exists \lambda_1, \lambda_2, ..., \lambda_k \in \mathbb{R}_+$$

tels que

$$\sum_{j=1}^{k} \lambda_j = 1 \qquad et \qquad x = \sum_{j=1}^{k} \lambda_j x_j. \tag{1.2}$$

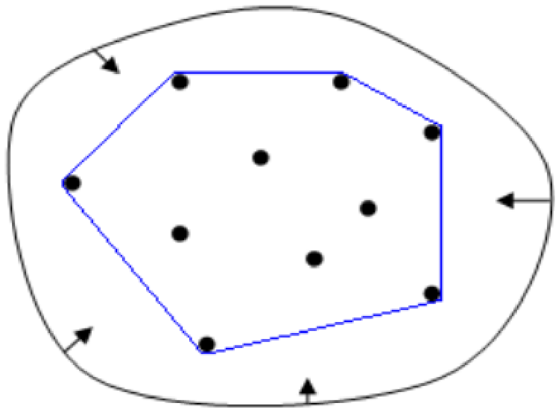


FIGURE 1.2 – envelloppe convexe

1.8.1 Ensemble convexe

Proposition 1.8.1. *Soit $S \subset \mathbb{R}^n$, alors $H(S)$ est un convexe, c'est aussi le plus petit convexe qui contient S.*

Proposition 1.8.2. *Soit $S \subset \mathbb{R}^n$. L'intersection de tous les convexes contenant S est le plus petit convexe qui contient S.*

Proposition 1.8.3. *Soit $S \subset \mathbb{R}^n$ un convexe non vide et $x_1 \in \bar{S}$ et $x_2 \in int(S)$, alors*

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

Corollaire 1.8.1. *Soit $S \subset \mathbb{R}^n$ tel que S est convexe et $int(S) \neq \emptyset$. Alors $int(S)$ est convexe.*

Corollaire 1.8.2. *Soit $S \subset \mathbb{R}^n$,tel que S convexe et $int(S) \neq \emptyset$. Alors $\bar{S}$ est convexe.*

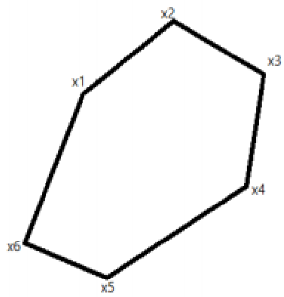
Polytope et simplexe

Définition 1.8.3. *L'envelloppe convexe d'un nombre fini de points* $x_1, x_2, ..., x_{k+1}$ *dans* $\mathbb{R}^n$ *s'appelle polytope.*
Si

$$x_2 - x_1, x_3 - x_1, ..., x_k - x_1, x_{k+1} - x_1$$

sont linéairement independants, alors $H(x_1, x_2, ..., x_{k-1}, x_k, x_{k+1})$ *s'appelle simplexe avec les points* $x_1, ..., x_k, x_{k+1}$.

Remarque 1.8.1. *Dans* $\mathbb{R}^n$*, il n'existe pas de simplexes avec plus de* $(n+1)$ *points. Dans* $\mathbb{R}^n$*, tout systéme forme de* $(n+1)$ *points ou plus est nécessairement lié.*



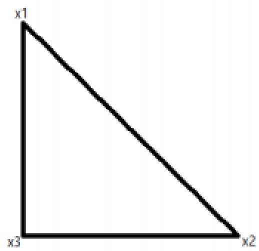


FIGURE 1.3 – polyope et simplexe dans $\mathbb{R}^2$

1.8.2 Fonction convexe

Définition 1.8.4. *Soit la fonction* $f : S \to \mathbb{R}$*, où* S *est un ensemble convexe non vide de* $\mathbb{R}^n$*.*
a) f *est dite convexe sur* S *si et seulement si*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2) \tag{1.3}$$

pour tous $x_1, x_2 \in S$ *et* $\lambda \in [0, 1]$.
b) f *est dite convexe sur* S *si et seulement si*

$$f(\lambda x_1 + (1-\lambda)x_2) < \lambda f(x_1) + (1-\lambda)f(x_2) \tag{1.4}$$

pour tous $x_1, x_2 \in S, x_1 \neq x_2$ *et* $\lambda \in]0, 1[$.
c) f *est dite fortement convexe ou uniformement convexe sur* S *si et seulement s'il existe une constante* $c > 0$ *tel que*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) - \frac{1}{2} C\lambda(1-\lambda)\|x_1 - x_2\|^2 \quad (1.5)$$

pour tous $x_1, x_2 \in S$ *et pour tout* $\lambda \in [0, 1]$.

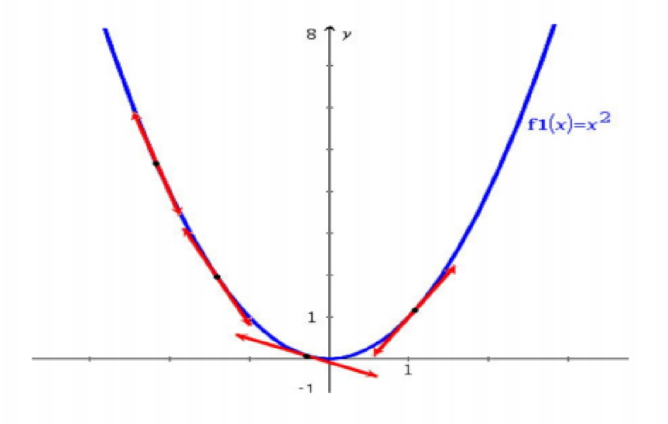


FIGURE 1.4 – fonction convexe

Définition 1.8.5. *Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction f est dite concave si $-f$ est convexe.*

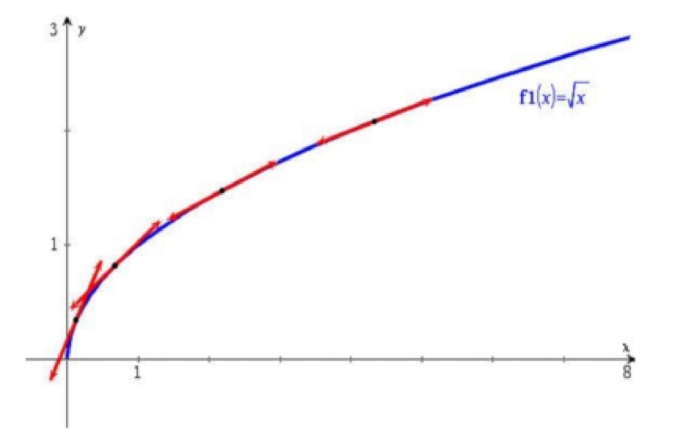


FIGURE 1.5 – fonction concave

.

Proposition 1.8.4. *([4]) Soit $S \subset \mathbb{R}^n$ convexe non vide, $f : \mathbb{R}^n \to \mathbb{R}$ convexe, alors pour tout $\alpha \in \mathbb{R}$, l'ensemble*

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

est convexe.

Définition 1.8.6. *Soit $S \subset \mathbb{R}^n$ convexe non vide, $f : \mathbb{R}^n \to \mathbb{R}$ est dite quasi convexe sur S si*

$$f(\lambda x_1 + (1-\lambda)x_2 \leq Maximum[f(x_1), f(x_2)] \tag{1.6}$$

pour tout $x_1, x_2 \in S$ et pour tout $\lambda \in [0,1]$.

1.9 Classification des problèmes d'optimisation

Forme générale d'un problème d'optimisation

Etant donné un ensemble $X \subset \mathbb{R}^n$ et une fonction $f : X \to \mathbb{R}$, la forme générale d'un problème d'optimisation est

$$\min\{f(x), x \in X\}, \tag{1.7}$$

où $X \subset \mathbb{R}^n$ est appelée ensemble des solutions admissibles ou réalisables, $x \in \mathbb{R}^n$ s'appelle variable de décision, $f(x)$ s'appelle fonction économique ou fonction objective.

Minimume local, global et strict

Définition 1.9.1. *Soit $f : \mathbb{R}^n \to \mathbb{R}$, on appelle problème de minimisation sans contraintes le problème $(\mathcal{P})$ suivant*

$$\min\{f(x)x \in \mathbb{R}^n\} \qquad (\mathcal{P})$$

1) $\hat{x} \in \mathbb{R}^n$ est un minimum global de $(\mathcal{P})$ si et seulement si $f(\hat{x}) \leq f(x) \quad \forall x \in \mathbb{R}^n$.
2) $\hat{x} \in \mathbb{R}^n$ est un minimum local de $(\mathcal{P})$ si et seulement si il existe un voisinage $V_\varepsilon(\hat{x})$ tel que

$$f(\hat{x}) \leq f(x) \quad \forall x \in V_\varepsilon(\hat{x}).$$

3) $\hat{x} \in \mathbb{R}^n$ est un minimum local strict de $(\mathcal{P})$ si et seulement s'il existe un voisinage $V_\varepsilon(\hat{x})$ tel que

$$f(\hat{x}) < f(x) : \quad \forall x \in V_\varepsilon(\hat{x}), x \neq \hat{x}.$$

4)un problème de maximisation est équivalent au problème de minimisation

$$\sup_{x \in A} f(x) = -\inf_{x \in A} -(f(x))$$

Problème d'optimisation sans contraintes

Si $X = \mathbb{R}^n$, le problème d'optimisation (1.7) s'appelle problème d'optimisation sans contraintes et aura donc la forme suivante

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

Problème d'optimisation non linéaire sans contraintes

Le problème (P) s'appelle problème d'optimisation sans contraintes non linéaire si f n'est pas affine.

Problème d'optimisation quadratique sans contraintes

Le problème (P) s'appelle problème d'optimisation sans contraintes non linéaire quadratique si

$$q(x) = \frac{1}{2}x^T Ax + B^T x + c$$

où A est une matrice symétrique d'ordre n, x^T est le transposé du vecteur x.

Problème d'optimisation avec contraintes

Soit $\mathcal{U}$ un ouvert non vide de $\mathbb{R}^n$ (le plus souvent $\mathcal{U} = \mathbb{R}^n$) et soit $f : \mathcal{U} \to \mathbb{R}$ une application différentiable sur $\mathcal{U}$ et $\varphi_1, \varphi_2, ..., \varphi_p : \mathcal{U} \to \mathbb{R}$ des applications de classe C^1 sur $\mathcal{U}$. Soit le domaine $\mathcal{D}$

$$\mathcal{D} = \{x \in \mathcal{U} | \varphi_i(x) = 0, \forall i = 1, 2, ..., p\}.$$

Le problème

$$\min_{x \in \mathcal{D}} f(x)$$

est un problème de minimisation sous contraintes.

1.10 Conditions d'optimalité

1.10.1 Conditions nécessaires d'optimalité

Théorème 1.10.1. *([20]) Soit $f : \mathbb{R}^n \to \mathbb{R}$ differentiable au point $\widehat{x} \in \mathbb{R}^n$. Si $\widehat{x}$ est un minimum local de (P) alors $\nabla f(\widehat{x}) = 0$.*

Théorème 1.10.2. *([20]) Soit $f : \mathbb{R}^n \to \mathbb{R}$ deux fois differentiable au point $\widehat{x} \in \mathbb{R}^n$ Si $\widehat{x}$ est un minimum local de (P) alors $\nabla(\widehat{x}) = 0$ et la matrice hessienne de f au point $\widehat{x}$, on note $H(\widehat{x})$, est semi définie positive.*

1.10.2 Conditions suffisantes d'optimalité

Théorème 1.10.3. *([20]) Soit $f : \mathbb{R}^n \to \mathbb{R}$ deux fois differentiable au point $\widehat{x} \in \mathbb{R}^n$. Si $\nabla f(\widehat{x}) = 0$ et $H(\widehat{x})$ est définie positive alors $\widehat{x}$ est un minimum local strict de (P)*

Exemple 1.10.1. $f(x_1, x_2) = \frac{1}{2}x_1^2 + x_1 \cos x_2$.
On a

$$\nabla f(x_1, x_2) = \begin{pmatrix} x_1 + \cos x_2 \\ -x_1 \sin x_2 \end{pmatrix},$$

le gradient s'annule pour
$\hat{x}_k = ((-1)^{k+1}), k\pi)^T, k \in \mathbb{Z}$,
$\bar{x_k} = (0, \frac{\pi}{2} + k\pi)^T$, $k \in \mathbb{Z}$.
On a

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} 1 & -\sin x_2 \\ -\sin x_2 & -x_1 \cos x_2 \end{pmatrix}$$

$$\nabla^2 f(\hat{x}_k) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

donc
$$\nabla^2 f(\bar{x_k}) = \begin{pmatrix} 1 & (-1)^{k+1} \\ (-1)^{k+1} & 0 \end{pmatrix}$$
$\hat{x}_k$ *vérifie les conditions suffisantes d'optimalité.*
$\bar{x_k}$ *ne vérifie pas les conditions nécessaires d'optimalité pour aucun* k*.*

Chapitre 2

Recherche linéaire

2.1 Direction de descente

Définition 2.1.1. *Soit $f : \mathbb{R}^n \to \mathbb{R}, \hat{x} \in \mathbb{R}^n$.le vecteur $d \in \mathbb{R}^n$ est dite direction de descente au point $\hat{x}$ si et seulement s'il existe $\delta > 0$ tel que*

$$f(\hat{x} + \lambda d) < f(\hat{x}) \qquad \forall \lambda \in]0, \delta[.$$

Donnons maintenant une condition suffisante pour que d soit une direction de descente.

Théorème 2.1.1. *Soit $f : \mathbb{R}^n \to \mathbb{R}$ différentiable au point $\hat{x} \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$ une direction verifiant la condition*

$$f'(\hat{x}, d) = \nabla f(\hat{x})^T d < 0.$$

Alors d est une direction de descente au point $\hat{x}$.

Preuve. Soit f est différentiable au point $\hat{x}$. Donc

$$f(\hat{x} + \lambda d) = f(\hat{x}) + \lambda \nabla f(\hat{x})^T d + \lambda \|d\| \alpha(\hat{x}, \lambda d)$$

avec

$$\alpha(\hat{x}, \lambda d) \to 0 \qquad si \;\; \lambda \to 0,$$

ceci implique que

$$f'(\hat{x}, d) = \lim_{\lambda \to 0} \frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} = \nabla f(\hat{x})^T d < 0.$$

La limite étant strictement négative, alors il existe un voisinage de zéro $V(0) =] - \delta, +\delta[$ tel que

$$\frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} < 0, \quad \forall \lambda \in] - \delta, +\delta[\qquad\qquad (\star)$$

la relation $(\star)$ est particulièrement vraie pour tout $\lambda \in]0, +\delta[$. On obtient le résultat cherché en multipliant la relation $(\star)$ par $\lambda > 0$. $\square$

Schémas générals des algorithmes d'optimisation sans contraintes

Supposons que d_k soit une direction de descente au point x_k. Ceci nous permet de considérer le point x_{k+1}, successeur de x_k, de la manière suivante

$$x_{k+1} = x_k + \lambda_k d_k, \quad \lambda_k \in]\, 0, +\delta[.$$

Vu la définition de direction de descente, on est assuré que

$$f\left(x_{k+1}\right) = f\left(x_k + \lambda_k d_k\right) < f\left(x_k\right).$$

Un bon choix de d_k et de λ_k permet ainsi de construire une multitude d'algorithmes d'optimisation.

Exemple de choix de direction de descente

Par exemple si on choisit $d_k = -\nabla f(x_k)$ et si $\nabla f(x_k) \neq 0$, on obtient la méthode du gradient. La méthode de Newton correspond à

$$d_k = -(H(x_k))^{-1}\nabla f(x_k).$$

Bien sur $-\nabla f(x_k)$ est une direction de descente

$$\left(\nabla f(x_k)^T d_k = -\nabla f(x_k)^T \nabla f(x_k) = -\|\nabla f(x_k)\|^2 < 0\right)$$

Pour la deuxième direction, si la matrice hessienne $H(x_k)$ est définie positive alors $d_k = -(H(x_k))^{-1}\nabla f(x_k)$ est aussi une direction de descente.

Exemple de choix du pas λ_k

On choisit en générale λ_k de façon optimale, c'est à dire que λ_k doit vérifier

$$f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k), \quad \lambda \in [0, +\infty[$$

En d'autre termes on est ramené à étudier à chaque itération un problème de minimisation d'une variable réelle. C'est ce qu'on appelle recherche linéaire.

Recherche linéaire

Définition 2.1.2. *Soit $f : \mathbb{R}^n \to \mathbb{R}$, considère le problème*

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

Les algorithmes qu'on étudie par la suite suivent les chémas généraux suivants

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

où λ_k est solution optimal du problème d'optimisation suivant

$$\min_{\lambda>0} f(x_k + \lambda d_k)$$

c'est à dire que λ_k vérifie

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

x_k, d_k sont fixes et la fonction a minimiser est une fonction d'une variable réelle définie comme suit

$$\lambda \mapsto \varphi_k(\lambda) = f(x_k + \lambda d_k)$$

Il existe deux grandes classes de méthodes qui s'intéressent à l'optimisation
a.les recherches linéaires exactes
b.les recherche linéaires inexactes

Choix optimale du pas λ_k

On peut choisir λ_k de façon optimale, c'est à dire qu'en partant du point x_k et la direction d_k, f décroit le mieux possible au point $x_{k+1} = x_k + \lambda_k d_k$. Ceci est possible si on impose à λ_k de vérifier la condition suivante

$$\forall \lambda \in]0, +\infty[\;\; f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k). \tag{2.1}$$

Notons $\varphi_k(\lambda)$ la fonction d'une variable réelle suivante

$$\varphi_k(\lambda) = f(x_k + \lambda d_k), \lambda \in]0, +\infty[\tag{2.2}$$

(2.1) et (2.2) donnent

$$\varphi_k(\lambda_k) \leq \varphi_k(\lambda), \qquad \lambda \in]0, +\infty[\tag{2.3}$$

ou encore λ_k est solution optimale de

$$\varphi_k(\lambda_k) = \min \{\varphi_k(\lambda) : \lambda \in]0, +\infty[\} \tag{2.4}$$

Trouvez λ_k vérifiant (2.4) s'appelle recherche linéaire exacte. Ce travail à effectue à chaque itération k. Donc à chaque itération k, on doit résoudre problème d'optimisation dans $\mathbb{R}$.

Intervalle de sécurité

Dans la plupart des algorithmes d'optimisation moderns, on ne fait jamais de recherche linéaire exacte, car trouverλ_k signifie qu'il va falloir calculer un grand nombre de fois la fonction φ_k et cela peut être dissuasif du point de vue du temps de calcul.

En pratique, on recherche plutôt un valeur λ^* qui assure une décroissance suffisante de f. cela conduit à la notion d'intervalle de sécurité.

Définition 2.1.3. *on dit que $[\lambda_g, \lambda_d]$ est un intervalle de sécurité s'il permet de classer les valeurs de λ de la façon suivante*
si $\lambda < \lambda_g$ alors λ est considéré trop petit,
si $\lambda_d \geq \lambda \geq \lambda_g$ alors λ est satisfaisant,
si $\lambda > \lambda_d$ alors λ est considéré trop grand .
Le problème est traduire de façon numérique sur φ_k les trois condition précédentes, ainsi que de trouver un algorithme permettent de déterminer λ_g et λ_d.

Algorithme de base
Etape 0(initialisation)
$\lambda_g = \lambda_d = 0$, choisir $\lambda_1 > 0$, pose $k = 1$ et aller à l'étape1.
Etape 1
si λ_k convient, poser $\lambda^* = \lambda_k$ et on s'arrête.
si λ_k est trop petit on prend $\lambda_{g,k+1} = \lambda_k, \lambda_d = \lambda_d$ et on va à l'étape2.
si λ_k est trop grand on prend $\lambda_{d,k+1} = \lambda_k, \lambda_g = \lambda_g$ et on va à l'étape2.
Etape 2
si $\lambda_{d,k+1} = 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, +\infty[$.
si $\lambda_{d,k+1} \neq 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, \lambda_{d,k+1}[$.
remplacer k par $k + 1$ et aller à l'étape1.

Remarque 2.1.1. *Effectuer une recherche linéaire exacte à itération est difficilement réalisable en pratique et assez couteux en temps et mémoire , c'est pour celà qu'on cherche λ_k mois couteux et facile à calcule, c'est ce qu'on appelle recherche linéaire inéxacte.*

Recherche linéaire inexacte

Si λ_k est obtenue par une rechreche linéaire exacte alors f décroit le mieux en pasant du point x_k au point $x_k + \lambda_k d_k$, c'est à dire

$$f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k) \qquad \forall \lambda \in]\,0, +\infty[$$

ou encore que

$$\forall \lambda \in]\,0,+\infty\,[\quad f\left(x_k\right)-f\left(x_k+\lambda_k d_k\right) \geq f\left(x_k\right)-f\left(x_k+\lambda d_k\right).$$

Notons par $\tau(\lambda)$ la quantité suivante

$$\tau(\lambda)=f\left(x_k\right)-f\left(x_k+\lambda d_k\right)=f\left(x_k\right)-f\left(x_k+\lambda d_k\right)$$

$\tau(\lambda)$ représente la décroissance de f, lorsqu'on pasae du point x_k au point $x_k+\lambda d_k$. On a donc

$$\tau\left(\lambda_k\right) \geq \tau(\lambda) \quad \forall \in]\,0,+\infty[.$$

2.2 Recherche linéaire d'Armijo

2.2.1 Principe de la méthode

Le but à réduire "de façon importante" la valeur de l'objectif par un pas $x_k \rightarrow x_{k+1}=x_k+\lambda_k d_k$ de x_k dans la direction d_k, tel que

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

Cette condition de décroissance stricte n'est pas suffissante pour minimiser φ_k au mois localement, par example, avec la fonction $\varphi: \mathbb{R} \rightarrow \mathbb{R}$,
$x \rightarrow \varphi(x)=x^2$ et $x_1=2$, les choix $d_k=(-1)^{k+1}$, $\alpha_k=2+3 \times 2^{-(k+1)}$ donnent

$$x_k=(-1)^k(1+2^{-k}).$$

x_k est bien strictement décroissante mais $\{x_k\}_{k>0}$ ne converge pas(diverge) vers le minimum zéro mais vers 1.
La régle d'Armijo impose une contrainte sur le choix de λ_k suffisante pour minimiser localement φ.
Une condition naturelle est de demander que f décroisse autant qu'une portion $\gamma \in]0,1[$ de ce que ferait le modéle linéaire de f en x_k.
L'idée est que la diminution jugée suffisante soit proportionnelle à la taille du pas λ_k. Ainsi plus le pas sera grand, plus la diminution de la fonction f sera grand.
Donc si $\gamma>0$, on désire que

$$f(x_k)-f(x_k+\lambda_k d_k) \geq \lambda_k \gamma$$

Ou encore

$$f(x_k+\lambda_k d_k) \leq f(x_k)-\lambda_k \gamma$$

Le facteur γ ne peut pas être choisi de façon arbitraire. Surtout il doit varier d'une itération à l'autre. Il est plus approprié de définir γ en fonction de la pente de la fonction au point x_k dans la direction d_k. En l'occurence, nous choisirons

$$\gamma = -\beta \nabla f(x_k)^T d_k,\ 0 < \beta < 1$$

telle que

$$\nabla f(x_k)^T d_k < 0.$$

Cela conduit à l'inégalité suivante, par fois appelée condition d'Armijo ou condition de décroissance linéaire

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \beta \lambda_k \nabla f(x_k)^T d_k. \tag{2.5}$$

Test d'Armijo[25]
Si

$$\varphi_k(\lambda) \leq \varphi_k(0) + \varphi_k'(0)\lambda, \tag{2.6}$$

autrement dit

$$f(x_k + \lambda d_k) \leq f(x_k) + \rho\lambda \nabla^T f(x_k) d_k,$$

alors λ convient.
Si

$$\varphi_k(\lambda) > \varphi_k(0) + \rho\varphi_k'(0)\lambda, \tag{2.7}$$

autrement dit

$$f(x_k + \lambda d_k) > f(x_k) + \rho\lambda \nabla^T f(x_k) d_k,$$

alors λ est trop grand.

2.2.2 Algorithme d'Armijo

algorithme(Règle d'Armijo)

Etape 0 (initialisation)
$\lambda_{g,1} = \lambda_{d,1}$, choisir $\lambda_1 > 0, \rho \in]0,1[$ poser k=1 et aller à l'étape 1.
Etape 1
Si $\varphi_k(\lambda_k) \leq \varphi_k(0) + \rho\varphi_k'(0)\lambda_k$: STOP ($\lambda^* = \lambda_k$).
Si $\varphi_k(\lambda_k) > \varphi_k(0) + \rho\varphi_k'(0)\lambda_k,\ alors\ \lambda_{d,k+1} = \lambda_d, \lambda_{g,k+1} = \lambda_k$ et aller à l'étape 2.
Etape 2
Si $\lambda_{d,k+1} = 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, +\infty[$
Si $\lambda_{d,k+1} \neq 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, \lambda_{d,k+1}[$
remplacer k par k+1 et aller à l'étape 1.

Théorème 2.2.1. *si* $\varphi_k : \mathbb{R}_+ \to \mathbb{R}$, *définie par* $\varphi_k(\lambda) = f(x_k + \lambda d_k)$ *est continue et bornée inférieurement, si* d_k *est une direction de descente en* x_k $(\varphi'_k(0) < 0)$ *et si* $\rho \in]0, 1[$, *alors l'ensemble des pas vérifiant la règle d'Armijo est non vide.*

Preuve. On a

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

$$\psi_\rho(\lambda) = f(x_k) + \rho\lambda_k \nabla^T f(x_k) d_k$$

le dévloppement de Taylor-Yong en $\lambda = 0$ de φ_k est

$\varphi_k(\lambda) = f(x_k+\lambda d_k) = f(x_k)+\rho\lambda\nabla^T f(x_k)d_k+\lambda\xi(\lambda)$ où $\xi(\lambda) \to 0, \lambda \to 0$ et comme $\rho \in]0, 1[$ et $\varphi'_k(0) = \nabla^T f(x_k)d_k$, pour $\lambda > 0$ on réduit

$$f(x_k) + \lambda_k \nabla^T f(x_k)d_k < f(x_k) + \rho\lambda_k \nabla^T f(x_k)d_k, \ \textit{pour}\ \lambda > 0$$

on voit que pour $\lambda > 0$ assez petit, on a

$$\varphi_k(\lambda) < \psi_\rho(\lambda).$$

De ce qui précède et du fait que φ_k est bornée inférieurement, et

$$\psi_\rho(\lambda) \to -\infty, \quad \lambda \to +\infty,$$

on déduit que la fonction $\psi_\rho(\lambda) - \varphi_k(\lambda)$ a la propriété

$$\begin{cases} \psi_\rho(\lambda) - \varphi_k(\lambda) > 0 \ \textit{pour}\ \lambda\ \textit{assez petit} \\ \psi_\rho(\lambda) - \varphi_k(\lambda) < 0 \ \textit{pour}\ \lambda\ \textit{assez grand} \end{cases}$$

donc s'annule au mois une fois pour $\lambda > 0$.
En choisissant le plus petit de ces zéros on voit qu'il existe $\bar{\lambda} > 0$ tel que

$$\varphi_k(\bar{\lambda}) = \psi_\rho(\bar{\lambda})\ \textit{et}\ \varphi_k(\lambda) < \psi_\rho(\lambda), \ \textit{pour}\ \ 0 < \lambda < \bar{\lambda}.$$

□

Contrainte des petits pas dans la régle d'Armijo

La condition (Armijo) nous permet et rejeter des pas qui, étant trop grands, ne fournissent pas une diminution suffisante de la fonction. Ayant réglé le problème des pas trop grands, nous allons maintenant nous interesser aux pas trop petits.

2.3 Recherche linéaire de Goldstein

2.3.1 Principe de la méthode

Dans la régla d'Armijo on assure la décroissance de la fonction objectif à chaque pas, mais c'est ne pas suffisant, reprenant l'example de la fonction $\varphi : \mathbb{R} \to \mathbb{R}, x \to \varphi(x) = x^2$ et $x_1 = 2$, et cette fois $d_k = -1$ le chois $\lambda_k = 2^{-(k+1)}$ donnent

$$x_k = (1 + 2^{-k})$$

φ_k est bien strictement décroissante mais$\{x_k\}_{k>0}$ ne converge pas vers le minimum zéro mais vers 1.
alors que la condition d'Armijo est satisfaite.
On va donc rajouter une deuxième condition à la condition d'Armijo qui éliminerait les pas trop petits.
Soit
le point $x_k \in \mathbb{R}^n$,
le pas de descente $\lambda_k > 0$,
le vecteur de descente $d_k \in \mathbb{R}^n$, tel que $\nabla f(x_k)^T d_k < 0$
Goldstein vérifant les deux conditions suivantes

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \rho\lambda_k \nabla f(x_k)^T d_k, \rho \in]0, \frac{1}{2}[\qquad (Goldstein1) \tag{2.8}$$

et

$$f(x_k + \lambda_k d_k) \geq f(x_k) + (1-\rho)\lambda_k \nabla f(x_k)^T d_k, \rho \in]0, \frac{1}{2}[\qquad (Goldstein2) \tag{2.9}$$

si on note

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

alors (Goldstein 1) et (Goldstein 2) deviennent

$$\varphi_k(\lambda) \leq \varphi_k(0) + \rho\lambda\varphi_k'(0) \tag{2.10}$$

et

$$\varphi_k(\lambda) \geq \varphi_k(0) + (1-\rho)\lambda\varphi_k'(0) \tag{2.11}$$

(Goldstein 1) assure une décroissance suffisante de f.
le pas λ_k trop petits sont nocif pour la convergence. La suit $\{x_k\}$ pourrait converge vers un point $\tilde{x}$ qui n'a rien à voir avec le problème d'optimisation.
(Goldstein2) élimine justement les pas λ_k trop petits.
Test de Goldstein
Si

$$\varphi_k(0) + (1-\rho)\varphi_k'(0)\lambda \leq \varphi_k(\lambda) \leq \varphi_k(0) + \rho\varphi_k'(0)\lambda$$

autrement dit

$$f(x_k) + (1-\rho)\lambda\nabla^T f(x_k)d_k \le f(x_k+\lambda d_k) \le f(x_k) + \rho\lambda\nabla^T f(x_k)d_k$$

alors λ convient.
Si

$$\varphi_k(\lambda) > \varphi_k(0) + \rho\varphi_k'(0)\lambda$$

autrement dit

$$f(x_k+\alpha d_k) > f(x_k) + \rho\lambda\nabla^T f(x_k)d_k$$

alors λ est trop grand .
Si

$$\varphi_k(\lambda) < \varphi_k(0) + (1-\rho)\varphi_k'(0)\lambda$$

autrement dit

$$f(x_k+\lambda d_k) < f(x_k) + (1-\rho)\lambda\nabla^T f(x_k)d_k$$

alors λ est trop petit.

2.3.2 Algorithme de Goldstein

algorithme(Règle de Goldstein)

Etape 0(initialisation)
$\lambda_{g,1} = \lambda_{d,1} = 0$, choisir $\lambda_1 > 0, \rho \in]0,1[, \delta \in]\rho, 1[$, pose $k = 1$ et aller à l'étape 1 .
Etape 1
Si $\varphi_k(0) + \delta\varphi_k'(0)\lambda \le \varphi_k(\lambda_k) \le \varphi_k(0) + \rho\varphi_k'(0)\lambda_k$: STOP ($\lambda^* = \lambda_k$)
Si $\varphi_k(\lambda_k) > \varphi_k(0) + \rho\varphi_k'(0)\lambda_k$, alors
$\lambda_{d,k+1} = \lambda_k, \lambda_{g,k+1} = \lambda_{g,k}$ et aller à l'étape 2.
Si $\varphi_k(\lambda_k) < \varphi_k(0) + \delta\varphi_k'(0)\lambda_k$ alors
$\lambda_{d,k+1} = \lambda_{d,k}, \lambda_{g,k+1} = \lambda_k$ et aller à l'étape2.
Etape 2
Si $\lambda_{d,k+1} = 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, +\infty[$
Si $\lambda_{d,k+1} \neq 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, \lambda_{d,k+1}[$

Théorème 2.3.1. *Soit $\varphi_k : \mathbb{R}_+ \to \mathbb{R}$; définie par $\varphi_k(\lambda) = f(x_k+\lambda d_k)$ est continue et bornée inférieurement, si d_k est une direction de descente en x_k et si $\rho \in [0,1], \delta \in]\rho, 1[$, alors l'ensemble des pas vérifiant la règle de Goldstein est non vide.*

Preuve. On a

$$\varphi_k(\lambda) = f(x_k + \lambda d_k),$$
$$\psi_\rho(\lambda) = f(x_k) + \rho\lambda\nabla^T f(x_k)d_k,$$
$$\psi_\delta(\lambda) = f(x_k) + \delta\lambda\nabla^T f(x_k)d_k,$$

le développement de Taylor-Yong en $\lambda = 0$ de φ_k est :
$\varphi_k(\lambda) = f(x_k + \lambda d_k) = f(x_k) + \rho\lambda\nabla^T f(x_k)d_k + \lambda\xi(\lambda)$, où $\xi(\lambda) \to 0, \lambda \to 0$ et comme $\rho \in]0,1[$ et $\varphi_k'(0) = \nabla^T f(x_k)d_k < 0$ on déduit
$f(x_k) + \lambda\nabla^T f(x_k)d_k < f(x_k) + \delta\lambda\nabla^T f(x_k)d_k < f(x_k) + \rho\lambda\nabla^T f(x_k)d_k$
pour $\lambda > 0$ on voit que pour $\lambda > 0$ assez petit, on a

$$\varphi_k(\lambda) < \psi_\delta(\lambda) < \psi_\rho(\lambda).$$

De ce qui précède et du fait que φ_k est bornée inférieurement,
et $\psi_\rho(\lambda) \to -\infty, \lambda \to +\infty$, on déduit que la fonction $\psi_\rho(\lambda) - \varphi_k(\lambda)$ à la propriété

$$\begin{cases} \psi_\rho(\lambda) - \varphi_k(\lambda) > 0 \ \textit{pour} \ \lambda \ \textit{assez petit} \\ \psi_\rho(\lambda) - \varphi_k(\lambda) < 0 \ \textit{pour} \ \lambda \ \textit{assez grand}, \end{cases}$$

donc s'annule au moins une fois pour $\lambda > 0$.
En choisissant le plus petit de ces zéro on voit qu'il existe $\bar{\lambda} > 0$ tel que

$$\varphi_k(\bar{\lambda}) = \psi_\rho(\bar{\lambda}), \ \textit{et} \ \varphi_k(\lambda) < \psi_\rho(\lambda), \ \textit{pour} \ 0 < \lambda < \bar{\lambda}$$

De la meme manière, il existe $\tilde{\lambda} > 0$ tel que

$$\varphi_k(\tilde{\lambda}) = \psi_\delta(\tilde{\lambda}), \ \textit{et} \ \varphi_k(\lambda) < \psi_\delta(\lambda), \ \textit{pour} \ 0 < \alpha < \tilde{\alpha}$$

et comme $\psi_\delta(\lambda) < \psi_\rho(\lambda)$ pour $\lambda > 0$, forcément $\tilde{\lambda} < \bar{\lambda}$ et $\lambda = \bar{\lambda}$

$$\begin{pmatrix} \psi_\delta(\tilde{\lambda}) = \varphi_k(\tilde{\lambda}) < \psi_\rho(\lambda) \qquad n'est \ \ autre\,que \\ f(x_k) + \delta\tilde{\lambda}\nabla^T f(x_k)d_k = f(x_k + \tilde{\lambda}d_k) < f(x_k) + \rho\tilde{\lambda}\nabla^T f(x_k)d_k \end{pmatrix}$$

Ce qu'il fallait démontrer. □

2.4 Recherche linéaire de Wolfe

2.4.1 Recherche lineaire de Wolfe faibles

Etant donnér deux réels ρ et σ tel que $0 < \rho < \sigma < 1$, λ_k est acceptable par la recherche linéaire inexacte de Wolfe si λ_k vérifie les conditions suivantes

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \rho\lambda_k\nabla^T f(x_k)d_k, \ \textit{tel que} \ 0 < \rho < \frac{1}{2} \qquad (Wolfe1) \tag{2.12}$$

$$\nabla^T f(x_k+\lambda_k d_k)d_k \geq \sigma \nabla^T f(x_k)d_k,\ \textit{tel que}\ 0<\sigma<1,\ \sigma>\rho \qquad (Wolfe2) \tag{2.13}$$

autrement dit

$$\varphi_k(\lambda_k) \leq \varphi_k(0) + \rho\lambda_k\varphi'_k(0),\ 0<\rho<\frac{1}{2} \tag{2.14}$$

$$\varphi'_k(\lambda_k) \geq \sigma\varphi'_k(0),\ 0<\sigma<1,\ \sigma>\rho. \tag{2.15}$$

algorithme(Règle de Wolfe)

Etape 0 (initialisation)
$\lambda_{g,1} = \lambda_{d,1} = 0$, choisir $\lambda_1 > 0, \rho \in [0,1], \sigma \in]\rho, 1[$, pose $k = 1$ et aller à l'étape 1.
Etape 1
Si $\varphi_k(\lambda_k) \leq \varphi_k(0) + \rho\varphi'_k(0)\lambda_k$ et $\varphi'_k(\lambda) \geq \varphi'_k(0)$: STOP($\lambda^* = \lambda_k$)
Si $\varphi_k(\lambda_k) > \varphi_k(0) + \rho\varphi'_k(0)\lambda_k$, alors $\lambda_{d,k+1} = \lambda_k, \lambda_{g,k+1} = \lambda_{g,k}$ et aller à l'étape 2. Si $\varphi'_k(\lambda) < \sigma\varphi'_k(0)$, alors $\lambda_{d,k+1} = \lambda_{d,k}, \lambda_{g,k+1} = \lambda_k$ et aller à l'étape 2.
Etape 2
Si $\lambda_{d,k+1} = 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, +\infty[$ Si $\lambda_{d,k+1} \neq 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, \lambda_{d,k+1}[$

2.4.2 Recherche linéaire de Wolfe forte

Pour certains algorithmes (par example le gradient conjugé non linéaire) il est parfois nécéssaire d'avoir une condition plus restrictive .
pour cela la deuxième condition (2.7) est remplacée par

$$|\nabla^T f(x_k+\lambda_k d_k)| \leq \sigma|\nabla^T f(x_k)d_k| = -\nabla^T f(x_k)d_k,$$

on aurra donc les conditions de Wolfe forte

$$f(x_k+\lambda_k d_k) \leq f(x_k) + \rho\lambda_k\nabla^T f(x_k)d_k \tag{2.16}$$

$$|\nabla f(x_k+\lambda_k d_k)^T d_k| \leq -\sigma\nabla^T f(x_k)d_k \tag{2.17}$$

autrement dit

$$\varphi_k(\lambda_k) \leq \varphi_k(0) + \rho\lambda_k\varphi'_k(0), 0<\rho<\frac{1}{2} \quad (Wolfeforte1) \tag{2.18}$$

$$|\varphi'_k(\lambda_k)| \leq -\sigma\varphi'_k(0), 0<\sigma<1, \sigma>\rho \quad (Wolfeforte2) \tag{2.19}$$

Théorème 2.4.1. *Si λ_k vérifie (Wolfe forte1) et (Wolfe forte2), alors λ_k vérifie (Wolfe 1)et (Wolfe 2).*

Preuve. (Wolfe fore1) et (Wolfe1) sont les meme proposition. Reste à montrer que (Wolfe forte2)$\Rightarrow$ (Wolfe2)
En effet, Supposons que λ_k vérifie (Wolfe forte2).
Alors on a

$$\sigma\varphi'_k(0) \leq \varphi'_k(\lambda_k) \leq -\sigma\varphi'_k(0)$$

par conséquent en a

$$\varphi'_k(\lambda_k) \geq \sigma\varphi'_k(0).$$

Ceci implique que λ_k vérifie (Wolfe2). □

Recherches linéaires non monotones

La technique de la recherche linéaire non monotone à été proposée en premier par Grippo, Lampariello, et Luicidi[8] et a connu de nombreuses application aussi bien en optimisation sans contraintes qu'en optimisation avec contraintes[8], [36]. Bien que prometteuse la recherche sur le théme de la recherche linéaire non monotone est toujours dans ces débuts, comme est souligné dans [27].
Soit le problème d'optimisation sans contraintes :

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

où f est une fonction différentiable et son gradient est noté par

$$g(x) = \nabla f(x)$$

supposons que le rapprochement actuel à la solution de (P) est x_k.
Si $g_k = \nabla f(x_k) \neq 0$, pour un direction de descente donnée d_k par une certine méthode, on trouve le pas λ_k par la réalisation de certines recherches linéaires tout au long de d_k, et en calcule le prochain rapprochement x_{k+1} comme suit

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

La recherche du pas λ_k doit assurer la décroissance suffisante de la fonction à chaque itération .
c'est à dire que

$$f(x_k + \lambda_k d_k) \leq \max_{0\leq j\leq m(k)} f(x_{k-j}).$$

La recherche linéaire non monotone comme en va le voir n'impose pas la condition de décroissance suffisante de la fonction objectif comme est le cas des recherches linéaires présédentes.
Nous allons expliciter dans ce qui suit deux recherches linéaires non monotones : la recherche linéaire non monotone d'Armijo et la recherche linéaire non monotone de wolfe.[25]

Recherche linéaire non monotone d'Armijo

Soit $0 < n_1 < n_2, \sigma \in]0,1[, \delta \in]0,1[$, et soit M est un entier positif, supposons lors de la k ième itération, on a $\bar{\lambda}_k \in]n_1, n_2[$ donné.
La recherche linéaire non monotone consiste à choisir dès la premiére itération un pas λ_k tel que

$$\lambda_k = \bar{\lambda}_k \sigma^{\eta_k} \tag{2.20}$$

satisfisant

$$f(x_k + \lambda_k d_k) \leq \min_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta \lambda_k g_k^T d_k \tag{2.21}$$

où

$$m(0) = 0, et\ 0 \leq m(k) \leq \min[m(k-1)+1, M-1], \qquad k \geq 1 \tag{2.22}$$

algorithme :(La Règle non monotone d'Armijo)

Etape0 (initialisation)
$\lambda_{g,1} = \lambda_{d,1} = 0$ choisir $M > 0, \rho \in]0,1[$ poser $k = 1$ et aller à l'étape1.
Etape1
Si $f(x_k + \lambda_k d_k) \leq \max\limits_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta \lambda_k g_k^T d_k$: STOP $(\lambda^* = \lambda_k)$
Si $f(x_k + \lambda_k d_k) > \max\limits_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta \lambda_k g_k^T d_k$, alors
$\lambda_{d,k+1} = \lambda_d, \lambda_{g,k+1} = \lambda_k$ et aller à l'étape2.
Etape2
Si $\lambda_{d,k+1} = 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, +\infty[$
Si $\lambda_{d,k+1} \neq 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, \lambda_{d,k+1}[$
remplacer k par $k+1$ et aller à l'étape1.

Remarque 2.4.1. *Si $m(k) = 0$, la recherche linéaire ci-dessus non monotone est réduite à la recherche linéaire inexacte d'Armijo classique.*

Recherche linéaire non monotone de Wolfe

Déterminer λ_k tel que

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \rho \lambda_k g_k^T d_k \tag{2.23}$$

$$|\langle g(x_k + \lambda_k d_k), d_k \rangle| \leq -\delta g_k^T d_k \tag{2.24}$$

où $0 < \rho < \delta < \dfrac{1}{2}$ et où l'ordre du nombre entier {m(k)} est produit par la formule

$$m(0) = 0, m(k) \leq \min\{m(k-1)+1, M_0\}, \tag{2.25}$$

avec M_0 un nombre entier.

$\boxed{\text{algorithme :(Règle non monotone de Wolfe)}}$

Etape 0(initialisation)
$\lambda_{g,1} = \lambda_{d,1}$, choisir $M > 0, \rho \in]0,1[, \delta \in]\rho, 1[$ poser $k = 1$ et aller à l'étape 1.
Etape 1
Si $f(x_k + \lambda_k d_k) \leq \max\limits_{0 \leq j \leq m(k)} f(x_{k-j}) + \rho \lambda_k g_k^T d_k$ et
$|\langle g(x_k + \lambda_k d_k), d_k \rangle| \leq -\delta g_k^T d_k$: STOP ($\lambda^* = \lambda_k$)
Si $f(x_k + \lambda_k d_k) > \max\limits_{0 \leq j \leq m(k)} f(x_{k-j}) + \rho \lambda_k g_k^T d_k$ alors $\lambda_{d,k+1} = \lambda_d$,
$\lambda_{g,k+1} = \lambda_k$ et aller à l'étape 2.
Etape 2 Si $\lambda_{d,k+1} = 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, +\infty[$.
Si $\lambda_{d,k+1} \neq 0$ déterminer $\lambda_{k+1} \in]\lambda_{g,k+1}, \lambda_{d,k+1}[$.
remplacer k par $k+1$ et aller à l'étape 2.cite3

Chapitre 3

Mèthode du gradient conjugué

3.1 Condition de Zoutendijk

Soit $f : \mathbb{R}^n \to \mathbb{R}$ on considère le probleme $(P) : minf(x)x \in \mathbb{R}^n$.

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

$$f(x_{k+1}) \leq f(x_k) - C\|\nabla f(x_k)\|^2 \cos^2 \theta_k. \tag{3.1}$$

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

$$\cos \theta_k = \frac{-\nabla^T f(x_k) d_k}{\|d_k\| \|d_k\|}. \tag{3.2}$$

Voici comment on se sert de la condition de Zoutendijk.

Proposition 3.1.1. *Si la suite $\{x_k\}$ générée par un algorithme d'optimisation vérifie la condition de Zoutendijk (3.1) et si la suite $\{f(x_k)\}$ est minorée, alors*

$$\sum_{k \geq 1} \|\nabla f(x_k)\|^2 \cos^2 \theta_k < \infty. \tag{3.3}$$

Preuve. En sommant les inégalités (3.1), on a

$$\sum_{k \geq 1}^{l} \|\nabla f(x_k)\|^2 cos^2 \theta_k \leq \frac{1}{C}(f(x_1) - f(x_{l+1})). \tag{3.4}$$

La série est donc convergente puisqu'il existe une constant C' telle que pour tout $k, f(x_k) \geq C'$
Les deux propositions suivantes précisent les circonstances dans les quelles la condition de Zoutendijk (3.1) est vérifiée avec les règles d'Armijo et de Wolfe. $\square$

Proposition 3.1.2. *[25] Soit $f : \mathbb{R}^n \to \mathbb{R}$ 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 génére une suite $\{x_k\}$ en utilisant la recherche linéaire d'Armijo avec $\lambda_1 > 0$. Alors il existe une constante $C > 0$ telle que, pour tout $k \geq 1$, l'une des conditions

$$f(x_{k+1}) \leq f(x_x) - C\nabla^T f(x_k) d_k \tag{3.5}$$

où

$$f(x_{k+1}) \leq f(x_k) - C\|\nabla f(x_k)\|^2 \cos^2 \theta_k \tag{3.6}$$

est vérifiée.

3.2 Méthode du gradient conjugué

Les méthodes du gradient conjugué sont utilisées pour résoudre les problèmes d'optimisation non linéaires sans contraintes spécialement les problèmes de grandes tailles. On l'utilise aussi pour rèsoudre les grands systèmes linéaires.

Elles reposent 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 (P).

3.2.1 Principe général d'une méthode à directions conjuguées

Définition 3.2.1. *Soit A une matrice symétrique $n \times n$, définie positive. On dit que deux vecteurs x et y de $\mathbb{R}^n$ sont A-conjugués (ou conjugués par rapport à A) s'ils vérifient :*

$$x^T A y = 0$$

3.2.2 Description de la méthode

Soit$\{d_1, d_2, \cdots, d_n\}$ une famille de vecteurs A-conjugués. On appelle alors méthode de directions conjuguées toute méthode itérative appliquée à une fonction quadratique strictement convexe de n variables définie par

$$q(x) = \frac{1}{2}x^T Ax + b^T x + c \; avec \, x \in \mathbb{R}^n$$

et A une matrice de $Mn \times n$ symétrique et définie positive, $b \in \mathbb{R}^n$ et $x \in \mathbb{R}$, conduisant à l'optimum en n étapes au plus. Cette méthode est de la forme suivante :

$$\begin{cases} x_0 & donner \\ x_{k+1} = x_k + \lambda_k d_k \end{cases}$$

où k est est optimal et $d_1, d_2, \cdots, d_n$ possédant la propriété d'être mutuellement conjuguées par rapport à la fonction quadratique.

Si l'on note $g_k = \nabla q(x)$,la méthode se construit comme suit : Calcul de λ_k Comme λ_k minimise q dans la direction d_k on a $\forall k$:

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

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

soit

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

d'où l'on tire :

$$\lambda_k = \frac{-d_k^T (Ax_k + b}{d_k^T A d_k}$$

Comment construire les directions A-conjuguées ?

Des directions A-conjuguées $d_0, \cdots, d_k$ peuvent être générées à partir d'un ensemble de vecteurs linéairement indépendants $\xi_0, \cdots, \xi_k$ en utilisant la procédure dite de Gram-Schmidt, de telle sorte que pour tout i entre 0 et k, le sous-espace généré par $d_0, \cdots, d_k$ soit égal au sous espace généré par $\xi_0, \cdots, \xi_k$

Alors d_{i+1} est construite comme suit :

$$d_{i+1} = \xi_{i+1} + \sum_{m=0}^{i} \varphi(i+1)m d_m$$

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

En effet, le sous-espace généré par les directions $d_0, \cdots, d_i$ est le même que le sous-espace généré par les directions $\xi_0, \cdots, \xi_k$ et ξ_{i+1} est linéairement indépendant de $\xi_0, \cdots, \xi_k$.

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

Les coefficients $\varphi_{(i+1)}m$, eux sont choisis de manière à assurer la A-conjugaison des $d_0, \cdots, d_i$.

3.2.3 Méthode du gradient conjugué dans le cas quadratique

La méthode du gradient conjugué est obtenue en appliquant la procédure de Gram-Schmidt aux gradients $\nabla q(x_0), \cdots, \nabla q(x_{n-1})$ c'est-à-dire en posant $\xi_0 = -\nabla q(x_0), ..., \xi_{n-1} = -\nabla q(x_{n-1})$.

En outre, nous avons que

$$\nabla q(x) = Ax + b \tag{3.7}$$

$$et \quad \nabla^2 q(x) = A \tag{3.8}$$

Notons que la méthode se termine si $\nabla q(x_k) = 0$.

Algorithme de la méthode du gradient conjugué pour les fonctions quadratiques

On suppose ici que la fonction à minimiser est quadratique sous la forme :

$$q(x) = \frac{1}{2} x^T Ax + b^T x + c.$$

Si l'on note $g_k = \nabla f(x_k)$, l'algorithme prend la forme suivante :

Algorithme

1. choisir $x_0 \in \mathbb{R}^n$.

2. Calculer $g_0 = Qx_0 - b$. Si $_0 = 0$ stop. Sinon poser $d_0 = -g_0$. Poser $k = 0$

3. Calculer $\alpha_k = -\dfrac{g_k^T d_k}{d_k^T Q d_k}$.

4. Calculer $x_{k+1} = x_k + \alpha_k d_k$.

5. Calculer $g_{k+1} = Qx_{k+1} - b$. Si $g_{k+1} = 0$ stop.

6. Calculer $\beta_k = \dfrac{g_{k+1}^T Q d_k}{d_k^T Q d_k}$

7. Calculer $d_{k+1} = -g_{k+1} + \beta_k d_k$.

8.Poser $k = k + 1$ et allez à 3.

Cet algorithme consiste à générer une suite d'itérés $\{x_k\}$ sous la forme :

$$x_{k+1} = x_k + \lambda_k d_k. \tag{3.9}$$

1-Construire itérativement des directions $d_0, ..., d_k$ mutuellement conjuguées

A chaque étape k la direction d_k est obtenue comme combinaison linéaire du gradient en x_k et de la direction précédente d_{k1} c'est-à-dire les coeefficients β_{k+1} étant choisis de telle manière que d_k soit conjuguée avec toutes les directions précédentes. Autrement dit

$$d_{k+1}^T A d_k = 0,$$

on en déduit que

$$\begin{aligned} d_{k+1}^T A d_k = 0 &\Rightarrow (-\nabla q(x_{k+1}) + \beta_{k+1} d_k)^T A d_k = 0 \\ &\Rightarrow -\nabla^T q(x_{k+1}) A d_k + \beta_{k+1} d_k^T A d_k = 0 \\ \beta_{k+1} &= \frac{\nabla^T q(x_{k+1}) A d_k}{d_k^T A d_k} = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}. \end{aligned} \tag{3.10}$$

Déterminer le pas λ_k

En particulier, une façon de choisir λ_k est de résoudre le problème d'optimisation (à une seule variable) suivant

$$\lambda_k = min f(x_k + \lambda d_k), \lambda > 0. \tag{3.11}$$

On en déduit que

$$\lambda_k = \frac{-d_k^T g_k}{d_k^T A d_k} = -\frac{1}{A d_k} g_k \frac{d_k^T}{d_k^T} = \frac{-d_k^T g_k}{d_k^T A d_k}, \tag{3.12}$$

le pas λ_k obtenu ainsi s'appelle le pas optimal.

Algorithme 3.1(Algorithme du gradient conjugué "quadratique"

Etape 0 : (initialisation)
Soit x_0 le point de depart, $g_0 = \nabla(x_0) = Ax_0 + b$, poser $d_0 = -g_0$.
poser $k = 0$ et aller à l'étape 1
Etape 1 :
si $g_k = 0$: STOP ($x^* = x_k$)."Test d'arrêt"
si non aller à l'étape 2.
Etape 2 :
Définir $x_{k+1} = x_k + \lambda_k d_k$ avec

$$\lambda_k = \frac{-d_k^T g_k}{d_k^T A d_k} \tag{3.13}$$

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k \tag{3.14}$$

$$\beta_{k+1} = \frac{g_{k+1}^T A d_k}{d_k^T A d_k} \tag{3.15}$$

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

Les avantages de la méthode du gradient conjugué

1- La consommation mémoire de l'algorithme est minime : on doit stocker les quatre vecteurs x_k, g_k, d_k, Ad_k (bien sur x_{k+1} prend la place de x_k au niveau de son calcul avec des remarques analogues pour $g_{k+1}d_{k+1}, Ad_{k+1}$.) et les scalaires λ_k,β_{k+1}.

2- L'algorithme du gradient conjugué linéaire est surtout utile pour résoudre des grands systémes creux, en effet il suffit de savoir appliquer la matrice A à un vecteur.

3- La convergence peut être assez rapide : si A admet seulement $r(r < n)$ valeurs propres distincts la convergence a lieu en au plus r itération.

3.3 Synthése des résultats de convergence de méthode du gradient conjugué

On présente dans cette section une synthése concernant la convergence des diifférentes variantes du gradient conjugué avec la recherche linéaire inexacte de Wolfe forte et de Wolfe faible.
Notre problème consiste à résoudre le problème (P) citè ci-dessus

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

où f est régulière (continûment différentiable) et g est son gradient. Notons par g_k le gradient de f au point x_k.
Rappelons que les différentes variantes du gradient conjugué genèrent une suite $\{x_k\}$ de la forme suivante

$$x_{k+1} = x_k + \lambda_k d_k \tag{3.16}$$

la direction d_k est définie par la formule de récurrence suivante ($\beta_k \in \mathbb{R}$)

$$d_k = \left\{ \begin{array}{ll} -g_1 & si \;\; k = 1 \\ -g_k + \beta_k d_{k-1} & si \;\; k \geq 2 \end{array} \right\} \tag{3.17}$$

le pas $\lambda_k \in \mathbb{R}$ étant déterminé par une recherche linéaire. Rappelons les différentes variantes du gradient conjugué qui diffèrent selon les valeurs que prennent les constantes β_k Particulièrement, on cite les variantes suivantes

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \qquad (Fletcher - Reeves) \tag{3.18}$$

$$\beta_k^{PRP} = \frac{g_k^T(g_k - g_{k-1})}{\|g_{k-1}\|^2} \qquad (Polak - Ribiere) \tag{3.19}$$

$$\beta_k^{HS} = \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T(g_k - g_{k-1})} \qquad (Hestenes - Stiefel) \tag{3.20}$$

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}} \qquad (ConjugateDescentMethod) \tag{3.21}$$

où $\|.\|$ est la norme Euclidienne. Récemment Dai et yuan ont aussi introduit la forme suivante

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T(g_k - g_{k-1})}. \tag{3.22}$$

La convergence globale des méthodes (3.18)-(3.19)-(3.20)-(3.21)-(3.22) a été étudiée par beaucoup d'auteurs. Citons en particulier Al-Baali [15], Gilbert et Nocedal [10], Hestenes et Stiefe [19], Hu et Story [30], Liu, Han et Yin [7], Powell [16] Touati-Ahmed et Story [4], et Zoutendijk [8]. Un facteur clé dans l'étude de la convergence globale est comment sélectionner le paramétre λ_k.

Le choix le plus naturel de λ_k est de faire une recherche linéaire exacte, c.-à-d poser $\lambda_k = \arg min_{\lambda \geq 0} f(x_k + \lambda d_k)$.
Cependant, ce choix naturel pourrait ne pas donner de convergence comme est le cas de la méthode (PRP) et (HS), comme l'a montré Powell dans [16] Inspiré par le travail de Powell, Gilbert et Nocedal dans [10] ont montré que

la méthode (PRP) est globalement convergente si β_k^{PR} est non-négatif et λ_k est déterminé avec la recherche linéaire satisfaisant la condition suffisante de descente suivante

$$g_k^T d_k < -c\|g_k\|^2, \qquad o \;\; c > 0 \tag{3.23}$$

en plus des conditions standard de Wolfe

$$\begin{cases} f(x_k + \lambda_k d_k) \leq f(x_k) + \omega_1 \lambda_k d_k^T g_k \\ \qquad\qquad d_k^T g_{k+1} \geq \omega_2 d_k^T g_k. \end{cases} \tag{3.24}$$

ou $0 < \omega_1 < \omega_2 < 1$
Récemment, Dai et Yuan [29][34] ont montré que la mééthode (CD) et la méthode (FR) sont globalement convergentes si les conditions de la recherche linéaire suivantes pour λ_k sont satisfaites

$$\begin{cases} f(x_k + \lambda_k d_k) \leq f(x_k) + \omega_1 \lambda_k d_k^T g_k \\ \quad \omega_2' d_k^T g_k \leq d_k^T g_{k+1} \leq -\omega_2'' d_k^T g_k. \end{cases} \tag{3.25}$$

ou $0 < \omega_1 < \omega_2' < 1 \;\; et \;\; \omega_2'' > 0$
Nous adoptons la supposition suivante sur fonction f.

Supposition 3.3.1. *Soit* $f : \mathbb{R}^n \to \mathbb{R}$.
L'ensemble $\pounds = \{x \in \mathbb{R}^n, f(x) \leq f(x_1)\}$ *est borné, où* $x_1 \in \mathbb{R}^n$*est le point initial.*
Sur un voisinage $\mathcal{N}$ *de* $\pounds$*, la fonction objectif* f *est continûment différentiable et son gradient est lipschitzien (i.e)*

$$\exists L > 0 \;\; telque \;\; \|g(x) - g(\widetilde{x})\| \leq L\|x - \widetilde{x}\|, \;\; \forall x, \widetilde{x} \in \mathcal{N}. \tag{3.26}$$

Remarque 3.3.1. *Ces suppositions impliquent qu'il existe* $\gamma > 0$ *tel que*

$$\|g(x)\| \leq \gamma, \forall x \in \pounds. \tag{3.27}$$

Rappelons maintenant le théorème suivant obtenu essentiellement par Zoutendijk [9] et Wolfe [22],[21]. Ce théorème assure la satisfaction de la condition de Zoutendijk [9], pour toute méthode du type, dans laquelle le pas λ_k *est déterminé par la règle de Wolfe faible.*

Théorème de Zoutendijk

Considérons la suite $(x_k)_k$, définie par (3.16) où d_k est une direction de descente et λ_k vérifie les conditions (3.25). Considérons aussi que la supposition (3.3.1) soit satisfaite, alors

$$\sum_{k \geq 1} \cos^2 \theta \|g_k\|^2 < \infty \tag{3.28}$$

est vérifiée, avec

$$\cos\theta_k = -\frac{g_k^T d_k}{\|g_k\| \|d_k\|}.$$

Remarque 3.3.2. *(3.28)équivalent*

$$\sum_{k\geq 1 \|d_k\|\neq 0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \tag{3.29}$$

où θ_k est l'angle que fait d_k avec $-g_k$.

3.3.1 Résultats de convergence (version Fletcher-Reves)

Avant d'exposer sans démonstration les différents résultats de convergence, donnons d'abord l'algorithme de la méthode de Fletcher Reeves

$$\boxed{\text{Algorithme de la méthode de Fletcher-Reeve}}$$

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 ($x^* = x_k$)."Test d'arrêt"
Sinon aller à l'étape 2.
Etape 2
Définir $x_{k+1} = x_k + \lambda_k d_k$ avec : $\qquad \lambda_k = \min_{\lambda>0} f(x_k + \lambda d_k)$.

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

où

$$\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}.$$

Les avantages de la méthode de Fletcher-Reeves
Cette méthode est très intéressante, d'une part parce qu'elle nécessite le stockage de très peu d'informations (essentiellement trois vecteurs de dimension n).

Convergence de la méthode de Fletcher-Reeves
Le premier résultat de convergence de la méthode du gradient conjugué non linéaire (version Flecher Reeves) avec la recherche linéaire inexacte de Wolfe forte suivante

$$\begin{cases} f(x_k + \lambda_k d_k) \leq f(x_k) + \omega_1 \lambda_k d_k^T g_k \\ \qquad |d_k^T g_{k+1}| \leq -\omega_2 d_k^T g_k. \end{cases} \tag{3.30}$$

où ($\omega < \frac{1}{2}$) a été démontré par Al-Baali [15].
Touati Ahmed et Story [4] ont généralisé ce résultat pour

$$0 \leq \beta_k \leq \beta_k^{FR}. \tag{3.31}$$

Gilbert et Nocedal [10] ont généralisé ce résultat pour

$$|\beta_k| \leq \beta_k^{FR}. \tag{3.32}$$

Récemment, Dai et Yuan [32] a montré que la méthode (FR) es globalement convergente avec la recherche linéaire inexacte de Wolfe relaxée.
On donne dans ce paragraphe les principaux résultats de convergence de la méthode du gradient conjugué non linéaire avec la recherche linéaire inexacte de Wolfe forte [Gilbertet et Nocedal], aussi avec la recherche linéaire inexacte de Wolfe relaxée [Dai et Yuan].

La propriété de descente de la méthode de Fletcher-Reeves

Théorème 3.3.1. *[10]*

On considère que la supposition (3.3.1) est satisfaite. Toute méthode du type (3.16) et (3.17)dans laquelle β_k vériffie

$$|\beta_k| \leq \beta_k^{FR}, \forall k \geq 1$$

et le pas λ_k est déterminé par la régle de Wolfe forte (3.30) avec $0 < \omega_1 < \omega_2 < \frac{1}{2}$ est une méthode de descente ($g_k^t < 0, \forall k \geq 1$) convergente, dans le sens où

$$\lim_{k\to\infty} \inf \|g_k\| = 0. \tag{3.33}$$

Théorème 3.3.2. *[10]*

Soit x_1 un point de départ pour lequel la supposition (3.3.1) soit satisfaite. Considérons une méthode du type (3.16) et (3.17) avec β_k satisfaisant à (3.18). On suppose aussi que
•Chaque direction d_k vérifie

$$d_k^T g_k \leq 0. \tag{3.34}$$

•Le pas λ_k est déterminé par la régle de Wolfe relaxée(3.25) avec $\omega_2'' < \infty$ alors

$$\lim_{k\to\infty} \inf \|g_k\| = 0.$$

3.3.2 Résultats de convergence (version Descente conjuguée)

Cette méthode fut proposée en 1987 par Fletcher et Reeves[24], Rappelons que pour la méthode de la descente conjuguée la variante β_k est

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}}.$$

La propriété de la méthode de la descente conjuguée

Fletcher [24] a démontré que la méthode de la descente conjuguée est une méthode pour laquelle la condition de descente suffisante est assurée si le pas λ_k est déterminé par la règle forte de (3.30)Wolfe avec $\sigma \leq \frac{1}{2}$.

Dai et Yuan [32] ont démontré le théorème suivant

Théorème 3.3.3. *Supposons que Supposition (3.3.1) est satisfaite. Pour toute méthode du type (3.16) et (3.17) dont β_k satisfait à (3.21) et le pas λ_k satisfait aux conditions de Wolfe relaxée*

$$f(x_k + \lambda_k d_k) \leq f_{x_k} + \rho d_k^T g_k$$

$$et$$

$$\sigma_1 d_k^T g_k \leq d_k^T g_{k+1} \leq -\sigma_2 d_k^T g_k$$

où $0 < \rho < \sigma_1 < 1 \quad et \leq \sigma_2 \leq 1.$

Alors la méthode génère des directions de descente suffisante à chaque itération $k \geq 1$.

Preuve. On a

$$\begin{aligned} -d_k^T g_k &= -(-g_k + \beta_k^{CD} d_{k-1})^T g_k \\ &= \|g_k\|^2 \left[1 + \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} \right] \\ \Rightarrow \frac{-d_k^T g_k}{\|g_k\|^2} &= 1 + \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} \end{aligned}$$

D'autre part

$$\sigma_1 d_k^T g_k \leq d_k^T g_{k+1} \leq -\sigma_2 d_k^T g_k$$

$$\Rightarrow 1 - \sigma_2 \leq 1 + \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k+1}} \leq 1 + \sigma_1,$$

d' où

$$1 - \sigma_2 \leq \frac{-d_k^T g_k}{\|d_k\|^2} \leq 1 + \sigma_1.$$

Donc si $\|g_k\| \neq 0$, on a

$$d_k^T g_k \leq -C\|d_k\|^2 \qquad ou \quad C = 1 - \sigma_2 > 0$$

et donc d_k est une direction de descente suffisante. $\square$

Convergence de la méthode de la descente conjuguée

Yuan [14] a démontré la convergence au sens (3.31) de cette méthode avec un pas satisfaisant aux conditions de Wolfe si $\sigma_1 < \frac{1}{2}$ et $\sigma_2 = 0$.
Dai et Yuan [29] ont démontré ce résultat pour $\sigma_1 < \frac{1}{2}$ et $\sigma_2 = 0$.

Théorème 3.3.4. *Supposons que Supposition 3.3.1. est satisfaite. Laquelle β_k vérifie (3.22) et le pas λ_k est déterminé par la règle de Wolfe relaxée où $0 < \rho < \sigma_1 < 1$ et $\sigma_2 = 0$, la méthode converge dans le sens où*

$$\lim_{k\to\infty} \inf \|g_k\| = 0.$$

Preuve. On a

$$1 - \sigma_2 \leq \frac{-d_k^T g_k}{\|d_k\|^2} \leq 1 + \sigma_1$$

$$\Rightarrow 1 \leq \frac{-d_k^T g_k}{\|d_k\|^2} \leq 1 + \sigma_1$$

$$\Rightarrow (1 + \sigma_1)^{-1} \leq \frac{\|d_k\|^2}{-d_k^T g_k} \leq 1$$

$$\Rightarrow (1 + \sigma_1)^{-1} \leq \frac{\|d_{k+1}\|^2 \|d_k\|^2}{-d_k^T g_k \|d_{k+1}\|^2} \leq 1$$

$$\Rightarrow (1 + \sigma_1)^{-1} \leq \frac{\beta_{k+1}^{CD}}{\beta_{k+1}^{FR}} \leq 1$$

$$\Rightarrow \beta_{k+1}^{CD} \leq \beta_{k+1}^{FR}$$

Donc β_{k+1}^{CD}vérifie l'inégalité (3.31).
On a

$$\lim_{k\Rightarrow\infty} \inf \|g_k\| = 0.$$

□

3.3.3 Résultats de convergence (version DaiYuan)

Récemment Dai et yuan [31] ont introduit une formule pour β_k sous la forme suivante

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \tag{3.35}$$

où $y_{k-1} = g_k - g_{k-1}$.
Cette méthode possède plusieurs propriétés, par exemple elle possède la propriété de descente à chaque itération et elle converge si le pas est déterminé par la règle de (Wolfe faible, Armijo et Goldstein) lorsque f est strictement convexe.

Descente de la méthode de Dai-Yuan

Théorème 3.3.5. *[33]*

Soit x_1 un point de départ pour lequel la supposition (3.3.1)est satisfaite.Considérons une méthode du type (3.16) et (3.17) avec β_k satisfaisant à (3.35).
•Si f est strictement convexe sur l'ensemble convexe £ c- à- d

$$(g(x) - g(y))^T (x - y) > 0 \qquad \textit{pour tout } x, y \in \textit{pounds} \tag{3.36}$$

Alors pour tout $k \geq 1$

$$g_k^T d_k < 0. \tag{3.37}$$

En 1999 les auteurs ont généralisé ce résultat pour toute fonction régulière avec la recherche de Wolfe faible(3.19) .

Théorème 3.3.6. *[33]*

Supposons que (3.16) soit satisfaite. Pour toute méthode du type (3.16) et (3.17) dont β_k satisfait à (3.35) et le pas $\lambda_k > 0$ satisfaisant aux conditions de Wolfe faible(3.24) , alors toutes les directions générées sont de descente, autrement dit

$$d_k^T g_k < 0, \;\; \forall k \geq 1.$$

Convergence de la méthode de Dai-Yuan

Les résultats suivant sont dûs à Dai et Yuan [28]. Ils assurent la convergence de cette méthode pour une fonction strictement convexe avec une recherche linéaire inexacte d'Armijo et Goldstein.

Théorème 3.3.7. *[3]*

Soit x_1 un point de départ pour lequel la supposition (3.3.1) soit satisfaite. Considérons une méthode du type (3.16) et (3.17)avec β_k satisfaisantt à (3.35).

$$\omega_2 \lambda_k \nabla^T f(x_k) d_k \leq f(x_k + \lambda_k d_k) - f(x_k) \leq \omega_1 \lambda_k \nabla^T f(x_k) d_k \tag{3.38}$$

où $g_k = \nabla f(x_k)$ *et* ω_1 *et* ω_2 *sont deux constantes vérifiant*
$0 < \omega_1 < \frac{1}{2} < \omega_2 < 1.$
Alors

$$\lim_{k\to\infty} \inf\|g_k\| = 0.$$

Théorème 3.3.8. *[3]*

Soit x_1 *un point de départ pour lequel la supposition (3.3.1.) soit satisfaite. Considérons une méthode du type (3.16) et (3.17)avec* β_k *satisfaisantt à (3.35).*
Si f est uniformément convexe sur l'ensemble convexe £ c-à- d s'il existe une constante $\eta > 0$ *tel que*

$$(g(x) - g(y))^T(x - y) \geq \eta\|x - y\|^2 \qquad \textit{pour tout } x, y \in £ \tag{3.39}$$

et le pas λ_k *satisfait aux conditions d'Armijo suivantes*

$$f(x_k + \lambda_k d_k) - f(x_k) \leq \omega\lambda_k\nabla^T f(x_k)d_k, \tag{3.40}$$

où $g_k = \nabla f(x_k)$ *et* $0 < \omega < 1$*, pour lequel la condition suffisante de descente (3.23) est assurée.*
Alors

$$\lim_{k\to\infty} \|g_k\| = 0$$

Algorithme de la Méthode de Dai-Yuan avec la règle de Wolfe faible

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 ($x^* = x_k$)."Test d'arrêt"
Sinon aller à l'étape 2.
Etape 2
Définir $x_{k+1} = x_k + \lambda_k d_k$ avec

$$\lambda_k \quad verifie\ les\ conditions.$$

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

où

$$\beta_{k+1}^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T\,[g_{k+1} - g_k]} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}.$$

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

3.4 Convergence de méthode du gradient conjugué avec la recherche linéaire non monotone

Nous avons constaté que les resultats de convergence pour les différentes variantes du gradient conjugué ont été obtenue en utilisant des recherches linéaires inexactes. La question naturele qui se pose est la suivante :

Peut-on obtenir des résultats de convergence avec ces variantes du gradient conjugué ou d'autres en utilisant la recherche linéaire non-monotone ?

La reponse a été donnée positivement dans les traveaux de B.Tahar, L.Yamina, and G.Rafik [26], Y.H.Dai[34].

3.4.1 Résultats de convergence avec la recherche linéaire non monotone d'Armijo

Lemme 3.4.1. *[34] Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction différentiable. on pose que*
(i)La fonction f est bornée inférieurement dans $\mathbb{R}^n$
(ii)Son gradient est lipchitzien i.e

$$\exists L > 0 \, telque \, \|g(x) - g(\tilde{x})\| \leq L\|x - \tilde{x}\| \qquad \forall x, \tilde{x} \in \mathcal{N}. \tag{3.41}$$

Considérons une méthode de type (3.16) dans laquelle d_k est une derection de descente et le pas λ_k est déterminé par la recherche linéaire non monotone d'Armijo
alors pour tout $l \geq 1$, on a

$$\max_{1\leq i\leq M} f(x_{Ml+i}) \leq \max_{1\leq i\leq M} f(x_{Ml(l-1)+i}) + \delta \max_{0\leq i\leq M-1} [\lambda_{Ml+i} g_{Ml+i}^T d_{Ml+i}], \tag{3.42}$$

$$\sum_{l\geq 1} \min_{0\leq i\leq M-1} \left\{ |g_{Ml+i}^T d_{Ml+i}|, (g_{Ml+i}^T d_{Ml+i})^2 / \|d_{Ml+i}\|^2 \right\} < +\infty. \tag{3.43}$$

Remarque 3.4.1. *de la relation (3.42), on voit que pour n'importe quelle méthode de recherche linéaire non monotone, l'ordre $\left\{ \max_{1\leq i\leq M} f(x_{Ml(l-1)+1}) \right\}$ est stirictement décroissant. En conséquance, cette relation joue un rôle important dans les analyses de convergence R-linéaire.*
Supposons qu'il existe deux constantes C_1 et C_2 telles que, pour tout k, on ait

$$g_k^T d_k \leq -c_1 \|g_k\|^2 \tag{3.44}$$

$$\|d_k\| \leq c_2 \|g_k\| \tag{3.45}$$

Nous pouvons montrons facilement le résultat de convergance suivant en utilisant le lemme 1.

Théorème 3.4.1. *Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction différentiable. On suppose que*
(i) La fonction f est bornée inférieurement dans $\mathbb{R}$.
(ii) Le gradient est lipschitzien i.e

$$\exists L > 0 \ telque \ \|g(x) - g(\tilde{x})\| \leq L\|x - \tilde{x}\| \qquad \forall x, \tilde{x} \in \mathcal{N}. \tag{3.46}$$

Considérons une méthode de type (3.16) dans laquelle d_k satisfait les deux condition (3.44)-(3.45)et le pas λ_k déterminé par la recherche linéaire non monotone d'Armijo
Alors il existe une constante C_3 tel que

$$\|g_{k+1}\| \leq c_3\|g_k\| \ pour \ tout \ k \tag{3.47}$$

en plus nous avons que

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.48}$$

Preuve. Notons que $\lambda_k \leq n_2$ donc (3.45) nous donne

$$\|x_{k+1} - x_k\| \leq \lambda_k\|d_k\| \leq c_2 n_2\|g_k\|. \tag{3.49}$$

D'aprés le (ii)on a que

$$\|g_{k+1} - g_k\| \leq c_2 n_2 L\|g_k\|. \tag{3.50}$$

Ainsi (3.47) est satisfaite pour

$$c_2 = 1 + c_2 n_2 L.$$

En autre, d'aprés (3.42),(3.44),(3.45) il s'ensuit que

$$\lim_{k\to\infty} \|g_{MI+\Phi(I)}\| = 0. \tag{3.51}$$

Où

$$0 \leq \Phi(I) \leq M - 1$$

D'aprés (3.47) on a que

$$\|g_{M(I+1)+i}\| \leq c_3^{2M}\|g_{MI+\Phi(I)}\| \ for \ i = 0, ..., M-1 \tag{3.52}$$

Par conséquent, il s'ensuit de (3.51),(3.52) que (3.48) est satisfaite.
Dans le cas où l'ensemble

$$\Psi = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$$

est bornée, la relation (3.48) implique que chaque point de $\{x_k\}$ est un point stationnaire de $f(x)$. Il s'en suit de (3.48)-(3.49) que $x_{k+1} - x_k$ tend vers zéro quand $k \to \infty$. Ceci montre que, si le nombre des points stationnaires de f est fini, la suite$\{x_k\}$ est convergente. Ainsi nous prouvaons encore la convergence globale de ces méthode avec la recherche linéaire non monotone pour des fonctions générale sous les conditions (3.44)-(3.45). Cependent, la relation utile (3.43) peut être établie, et par conséquent elle rend possible d'affaiblir les condition (3.44)-(3.45) en utilisant la direction de recherche d_k . Par example, si pour $\|d_k\|^2$ il existe deux constantes positives β et γ tel que

$$\|d_k\|^2 \leq \beta + \gamma \, pour \, tout \, k \tag{3.53}$$

et si la condition suffisante(3.44) est satisfaite, alors on peut montrer que la méthode converge dans le sens

$$\lim_{k\to\infty} \inf \|g_k\| = 0 \tag{3.54}$$

$\square$

Théorème 3.4.2. *[34] supposons que $f(x)$ est bornée inférieurement sur $\mathbb{R}^n$ et que son gradient $\nabla f(x)$ est continu Lipschitzien. Conséderons n'importe quelle méthode itérative(3.16), où d_k satisfaite (3.44)(3.53), et où λ_k est obtenu par la recherche linéaire non monotone d'Armijo. Alors, la méthode converge dans le sens de (3.54).*

3.4.2 Résultats de convergence avec la recherche linéaire non monotone de Wolfe

Dans cette partie, on étudie la convergence globale de la méthode de gradient conjugué en utilisant une nouvelle classe de recherche linéaire non monotone de Wolfe. L'idée vient des technique de recherche de F-rule.

Dans l'analyse de convergence et implémentation de la méthode de gradient conjugué. La régle de Wolfe pour la recherche linéaire non monotone classique noté (RLN)élargie est

$$f(x_k + \lambda_k d_k) \leq \max_{0\leq j\leq m(k)} f(x_{k-j}) + \gamma_1 \lambda_k g_k^T d_k \tag{3.55}$$

$$\gamma_2 g_k^T d_k \leq g(x_k + \lambda_k d_k)^T d_k \tag{3.56}$$

où $0 < \gamma_1 \leq \gamma_2 \leq 1, \lambda_1 > 0, \; et \; M\epsilon N$

$$0 \leq m(k) \leq \min\{m(k-1)+1, M\}, \quad m(0) = 0 \tag{3.57}$$

Nous allons voir dans cette partie comment on introduit la nouvelle classe de recherche linéaire non monotone de Wolfe que l'on notera par (RLN1).

Technique de la recherche linéaire non monotone de Wolfe
Etant donné d'abord, les suppositions générales de cette partir.

Supposition 3.4.1. *soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction différentiable. On suppose que*
(A_1) L'ensemble de niveau $L_0 = \{x \ f(x) \leq f(x_0), x \in \mathbb{R}^n\}$ est borné, où le point x_0 est le point de départ.
$(A_2) f$ est fortement convexe et différentiable dans l'ensemble de niveau L_0 et son gradient $g_k = \nabla f(x)$ est continue de lipchitz. ie, il existe une constante $L > 0$ tel que

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{N}. \tag{3.58}$$

Définition 3.4.1. *la fonction $\sigma : [0, \infty[\to [0, \infty[$ est une F-fonction, si pour toute suite $\{t_i\} \subset [0, \infty[$*

$$\lim_{k \Rightarrow \infty} \sigma(t_i) = 0 \Rightarrow \lim_{k \Rightarrow \infty} t_i = 0 \tag{3.59}$$

Définition 3.4.2. *soit $\eta = \sup \{\|g(x) - g(y)\|/x, y \in L_0\} > 0$*
Alors l'application $\delta : [0, \infty[\to [0, \infty[$ définie par

$$\delta(t) = \begin{cases} \inf \left\{ \dfrac{\|x - y\|}{\|g(x) - g(y)\|} \geq t \right\} & t \in [0, \eta) \\ \lim\limits_{s \to \eta^-} \delta(s), & t \in [0, \eta) \end{cases}$$

est appelé la fonction reverse modulus du gradient $g(x)$.
Maintenant on donne la F-rule de la recherche linéaire non monotone (RLN1) comme suit

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) - \sigma(t_k) \tag{3.60}$$

où σ est une fonction différentiable et $t_k = -\dfrac{g_k^T d_k}{\|d_k\|}$. Posons $x_{k+1} = x_k + \lambda_k d_k$. Evidemment, si $M = 0$, la F-rule non monotone est justement la régle de décroissance suffisante dans [12].
Notons aussi que tout fonction non décroissante $\sigma : [0, \infty) \to [0, \infty)$ tel que $\sigma(0) = 0$ et $\sigma(t_i) > 0$ pour $t > 0$ est nécessairement un F-fonction. Par convention dans la suit on pose

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j})$$

où
$$k - m(k) \leq l(k) \leq k.$$
En utilisant la définition (3.4.2) on a
$$\lambda_k \|d_k\| \geq \delta \left[(\gamma_2 - 1) \frac{g_k^T d_k}{\|d_k\|} \right]$$
qui désigne
$$\lambda_k \|d_k\| \geq \delta \left[(1 - \gamma_2)(-\frac{g_k^T d_k}{\|d_k\|}) \right] \tag{3.61}$$
où $\delta(.)$ est la fonction reverse modulus du gradient. Donc,il s'ensuit de (3.55)et (3.61) que
$$f(x_k + \lambda_k d_k) \leq f(x_{l(k)}) + \gamma_1 \lambda_k g_k^T d_k$$
$$f(x_k + \lambda_k d_k) = f(x_{l(k)}) - \gamma_1 \lambda_k \|d_k\| (-\frac{g_k^T d_k}{\|d_k\|})$$
$$f(x_k + \lambda_k d_k) \leq f(x_{l(k)}) - \gamma_1 (-\frac{g_k^T d_k}{\|d_k\|}) \delta \left[(1 - \gamma_2)(-\frac{g_k^T d_k}{\|d_k\|}) \right]$$
$$f(x_k + \lambda_k d_k) \leq f(x_{l(k)}) - \sigma(-\frac{g_k^T d_k}{\|d_k\|}) \tag{3.62}$$
où $\sigma(t) = \gamma_1 t \delta[(1 - \gamma_2)t], t \geq 0$. clairement , $\sigma(t)$ est une fonction différentiable . Ceci indique que la régle (3.55)-(3.56) satisfait la F-rule non monotone (3.60)

Algorithme

Etant donne $\rho > 0$, $\beta \in [0,1], \gamma_2 \in [0,1]$, M un entier non négatif faire l'étape initiale de test
$$r_k = \frac{\rho g_k^T d_k}{\|d_k\|^2}$$
Prendre $\lambda_k = \beta^{h(k)} r_k, h(k) = 0, 1, 2, ..., h(k)$ est faire
$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) - \sigma(t_k)$$
où $m(0) = 0$, $0 \leq m(k) \leq \min\{m(k-1) + 1, M\}$, $k \geq 1$, évidement si la direction de la descente d_k est supposée être $d_k^T g_k$, alors si $m(k)$ suffisament grand, l'inégalité (3.60) est toujour satisfaite, donc elle satisfait l'existence de λ_k .

Dans la recherche linéaire ci-dessus à chaque itération il est recommandé que l'étape initiale du test r_k reste la même mais peut être ajusté automatiquement.

Remarque 3.4.2.
1-Afin d'utiliser une recherche linéaire non monotone le calcule de pas λ_k *doitse faire suivant la direction de descente , on prouve que cette étude de gradient conjugué préserve la direction de recherche* d_k*.*
2-Afin de calcule un grand pas λ_k *on doit avoir*

$$\sum_{k=0}^{\infty} \sigma(t_k) < \infty. \tag{3.63}$$

Analyse de convergence

Dans cette section, on établit les propriétés de la convergence global des méthode de gradient conjugué aves F-rule non monotone(RLN1). Notons que pour établir notre résultat, on a besoin quelques conditions.

Lemme 3.4.2. *Soit* d_k *un direction de descente et* λ_k *le pas trouvé par la recherche linéaire non monotone alors* $\{x_k\} \subset L_0$*.*

Preuve. La recherche linéaire non monotone (RLN1)(3.60)montre que

$$f(x_1) \leq f(x_0) - \sigma(t_0) < f(x_0)$$

$$f(x_2) \leq \max_{0 \leq j \leq m(1)} f(x_{1-j}) - \sigma(t_1) < f(x_0)$$

$$f(x_3) \leq \max_{0 \leq j \leq m(2)} f(x_{2-j}) - \sigma(t_2) < f(x_0)$$

$$f(x_{k+1}) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) - \sigma(t_k) < f(x_0)$$

On alors $\{x_k\} \subset L_0$. □

Lemme 3.4.3. *Soit* λ_k *le pas obtenue par la recherche linéaire non monotone (RLN) et* d_k *une direction de descente. Soit* g_k *le gradient de* f*. Alors*

$$g_k^T d_k \leq -\frac{7}{8}\|g_k\|^2 \qquad k = 1, 2, 3, ... \tag{3.64}$$

Lemme 3.4.4. *Sous les conditions* (A_1)*, la suite* $f(x_{l(k)})$ *est croissante.*

Preuve. D'aprés (3.60), pour tout k on a

$$f(x_{k+1}) \leq f(x_{l(k)}),$$

d'aprés la recherche linéaire non monotone(RLN), $0 \leq m(k) \leq m(k-1) + 1$ on a

$$f(x_{l(k)}) = \max_{0 \leq j \leq m(k)} f(x_{k-j}) \leq \max_{0 \leq j \leq m(k-1)+1} f(x_{k-j})$$

$$= \max\left\{\max_{0\le j\le m(k-1)} f(x_{k-1-j}), f(x_k)\right\}$$
$$= \max\left\{f(x_{l(k-1)}), f(x_k)\right\}$$
$$= f(x_{l(k-1)}).$$

$\square$

Lemme 3.4.5. *En supposant que* (A_1) *soit vérifiée, alors* $f(x_{l(k)})$ *converge i.e* $\lim_{k\to\infty} f(x_{l(k)})$ *existe et*

$$\lim_{k\to\infty} \sigma(t_{l(k)-1}) = 0. \tag{3.65}$$

Théorème 3.4.3. *Soit la fonction* $f : \mathbb{R}^n \to \mathbb{R}$ *satisfaisant la supposition (3.4.1) soit la suite* $\{x_k\}$ *définie par (3.16)où le pas* λ_k *définie par la F-rule non monotone (RLN1). Si la direction* d_k *satisfait*

$$-\frac{g_k^T d_k}{\|d_k\|} \ge \sigma(\|g_k\|), \quad k = 0, 1, ... \tag{3.66}$$

et

$$\|d_k\| \le m_1 \|g_k\| \tag{3.67}$$

$\sigma(.)$ *est une fonction différentiable et* $m_1 > 0$. *Alors la suite* $\{x_k\} \subset L_0$ *et*

$$\lim_{k\to\infty} \|g_k\| = 0$$

Bibliographie

[1] Bosora S ,Khdir T *Méthode de la plus forte pente mimoir master2* ,2021,2022.

[2] Bohani A, Bouchra. *la méthode du gradient conjugué utilisant des recherches linéaires pour résoudre un problème d'optimisation,mimoir master* ,2020,2021.

[3] B.T. Polyak *The Conjugate Gradient Method In Extremem Problems, Comput.Math. Math.*1969

[4] D. Touati-Ahmed and C. Storey *Efficient Hybrid Conjugate Gradient Techniques, JOTA,*1990

[5] E. Polak and G. Ribière *note sur la convergence de directions conjuguées, Rev. française informat. recherche operationelle, 3e année,*1969

[6] G.H. Liu and J.Y. Han and H.X. Yin *global convergence of the Fletcher-Reeves algorithme With an inexact line search, appl. math. J. Chinese Univ. Ser. B,*1995.

[7] G. H. Liu, L. L. Jing, L. X. Han, And D. Han, *a class of nonmonotone conjugate gradient methods for unconstrained optimization journal of optimization theory and applications*1999

[8] Grippo, L. Lampariello, F, and Lucidi, S.,. *a non monotone line search technique for New- tonŠs method, SIAM journal on numerical analysis,*1986.

[9] G. Zoutendijk *non linear programming computational methods, integer and non linear programming, north holland, amsterdam,*1970

[10] J.C. Gilbert and J. Nocedal. *global convergence properties of conjugate gradient methods for optimization, SIAM J. optimization.*1992.

[11] J. C. Gilbert. *eléments d'optimisation différentiable : théorie et algorithmes, notes de cours, école nationale supérieure de techniques avancées, Paris.,*2007.

[12] J.M. Ortega, W.C. Rheinboldt, *iterative solution of non linear equations in several variables, academic press, NewYork,*1970

[13] J. Nocedal . *hteory of algorithm for unconstrained optimization, acta numerica,*1991

[14] L. Armijo , Minimzation of function having lipschitz continous Ěrst partial derivatives, *PacifiĚc journal of mathematics,*1966.

[15] M. Al-Baali. *descent property and global convergence of the Fletcher-Reeves method with inexacte line search. IMA J. Num. Anal.*1985.

[16] M.J.D. Powell *convergence properties of algorithms for non linear optimization, SIAM*1988

[17] M.J.D. Powell *non convex minimzation calculation and the conjugate gradient method, in : lecture notes in mathematics 1066 (Springer, Berlin),*1984

[18] M.J.D. Powell *on the convergence of the variable metric algorithme J. Inst. Math. Appl,*1971

[19] M. R. Hestenes and E.L. Stiefel *methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Standars Sect,*1952

[20] Pr Rachid Benzine, *OPTIMISATION CONVEXE. OPTIMISATION SANS CONTRAINTES,* 2007

[21] P.Wolfe *conditions for ascent methods some corrections 2, SIAM Review,*1971

[22] P. Wolfe *convergence conditions for ascent methods, SIAM Review,*1969

[23] R. Fletcher and C. Reeves. *function minimization by conjugate gradients. Comput.,*1964.

[24] R. Fletcher. *practical methods of optimization, John Wiley&Sons, Chichster,*1987.

[25] TAHAR Bouali, *doctorat en Mathématiques convergence des méthodes du gradient conjugué avec la recherche linéaire non monotone,*2014.2015.

[26] T. Bouali , Y. Laskri, and R. *GuefaiĚa, New Global Convergence Of Non monotone Line Search Algorithm, International Journal Of Pure And Applied Mathematics,*

[27] Toint, P. L. *A Nonmonotone Trust-Region Algorithm for Nonlinear Optimization Subject to Conûex Constraints, Mathematical Programming,*1997

[28] W.sun and Y.X.YUAN optimisation theay and methods

[29] Y. Dai. *A nonmonotone conjugate gradient algorithm for unconstrained optimization. J. Syst. Sci. Complex. 15,*2002.

[30] Y.F. Hu and C. Story *Global Convergence Result For Conjugate Gradient Methods, JOTA,*1991

[31] Y.H. Dai and Y. Yuan. *A non linear conjugate gradient with a strong global convergence property, SIAM J. Optimization,,*1999.

[32] Y. H. Dai and Y. Yuan. *Convergence properties of the Fletche-Reeves method, IMA J Numer. Anal.*1996.

[33] Y. H. Dai and Y. Yuan. *Some properties of a new conjugate gradient method, in : Advances in Nonlinear Programming, ed . Kluwer Academic, Boston,,*1998.

[34] Y. H. Dai,. *On the Nonmonotone Line Search, Journal of Optimization theory and Applica- tions,*2002.

[35] Zhang, Li. Zhou, Weijun, Li, Donghui,. *master2Global Convergence Of a ModiĚed Fletcher-Reeves Conjugate Gradient Method With Armijo-type Line Search. Numer. Math.,*2006

[36] Zhou, J. L, and Titas, A,. *Nonmonotone Line Search for Minimax Problems, Journal of Optimization Theory and Applications,,*1993.

[37] Z. Li, J. Chen and N. Deng *A New Conjugate Gradient Method and its Global Convergence Properties, Systems Science and Mathematical Sciences,*1998.