الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعليم العالي و البحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جامعةعباسلغرورخنشلة
ABBES LAGHROUR- KHENCHELA UNIVERSITY

Faculty of Sciences and Technology

Department of Mathematics and Computer Science

N° de série :……

Mémoire de fin d'études

Pour l'obtention du diplôme de**Master**

Filière:**Mathématiques**

Spécialité:**Mathématiques Appliquées**

Intitulé par :

Comparaison des performances d'une nouvelle méthode de gradient conjugué

Réalisé par : **Boumaraf Meryem** Dirigé par : **M. Chergui Ahmed**

Saadi Hafidha Présenté le **13/09/2020**

Membres de jury :

Dr. R.Mechraoui Président

Dr.N.Hamdi Examinateur

2019-2020

Table des matières

Remerciements

Je tiens à adresser mes remerciements les plus chaleureux et ma profonde gratitude à mon encadreur Monsieur Ahmed Chergui , professeur à l'Université Abbes Laghrour-khenchela, pour m'avoir proposer le sujet de ce mémoire. C'est grâce à sa grande disponibilité, ses conseils, ses orientations, et ses encouragements que j'ai pu mener à bien ce travail.

Mes remerciements vont également à Monsieur R_Machraoui professeur à l'Université Abbes Laghrour-khenchela, pour avoir bien voulu me faire l'honneur d'accepter de présider le jury.

De même je remercie madame N_Hamdi ,maître de conférence classe A, à l'Université Abbes Laghrour-khenchela .

Je remercie également tout ceux qui ont contribué de prés ou de loin à l'élaboration de ce mémoire, en particulier, mon pére pour ses encouragements.

Enfin, Merci à toute ma famille, en particulier, ma mère, mon père, mes frères , et à tous mes amis et tous mes collègues sans exception.

ملخص

في هذا العمل ندرس طريقة جديدة معدلة لطريقة التدرج المترافق غير الخطية لحل مسائل الاستمثال غير المقيد وهذه الطرقة تحقق شرط النزول والتقارب الكلي باستعمال البحث الخطي غير الدقيق وبعد اجراء التجارب العددية اعطت نتائج جيدة عند مقارنتها ببعض الطرق التقليدية ا

الكلمات المفتاحية

التدرج المترافق التقارب الكلي البحث الخطي غير الدقيق

Résumé

Dans ce mémoire, nous étudions une nouvelle méthode modifiée du gradient conjuguée non linéaire pour résoudre les problèmes d'optimisation sans contraintes. Cette méthode est proposée pour garantir la convergence globale avec la condition suffisante de descente sous la recherche linéaire inéxacte, cette nouvelle formule donne d'excellents résultat numériques en les comparant à certaines méthodes traditionnelles .

Mots clés :
Gradient conjugué, Convergence globale, Recheche linéaire inéxacte , condition suffisante de descente

Abstract

In this paper, we study a new modified nonlinear conjugate gradient Method for solving largescale Unconstrained problems. This method is proposed to guarantee the global convergence with the sufficient descent condition under the inexact linear search this new formula gives excellent numerical result by comparing them to some traditional methods

Mots clés :
Conjugate gradient, Global convergence , inexact linear search, suffisante descente.condition

.

Introduction

Problématique :

Cadre Un probléme d'optimisation consiste, étant donnée une fonction $f : S \to R$, à trouver :

1) son minimum v (resp. son maximum) dans S

2) un point $x_0 \in S$ qui réalise ce minimum (resp. maximum) i.e. $f(x_0) = v$.

Vocabulaire :

– f est la fonction objectif

– v est la valeur optimale

– x_0 est la solution optimale

– écriture du probléme : $\min_{x \in S} f(x)$ resp. $\max_{x \in S} f(x)$

Remarque 1) L'optimisation est une branche des mathématiques. Dans la pratique, on part d'un probléme concret, on le modélise et on le résoud mathématiquement (analytiquement : probléme d'optimisation, numériquement : programme mathématique).

2) Lien minimum/maximum : soit f une fonction dont on veut trouver le maximum. Le probléme $\max_{x \in S} f(x)$ renvoit(x_0, v) alors que le probléme $\min_{x \in S} f(x)$ renvoit $(x_0, -v)$. D'ou ce lien. Ainsi la recherche d'un maximum peut toujours se ramener à la recherche d'un minimum.

Applications

L'optimisation intervient dans de nombreux domaines :

• en recherche opérationnelle (probléme de transport, ´economie, gestion de stocks...)

• en analyse num´erique (approximation/r´esolution de systémes linéaires, non linéaires...)

• en automatique (modélisation de systémes, filtrage...)

• en ingénierie (dimensionnement de structures, conception optimale de systémes (réseaux, ordinateurs...))

On s'interesse dans ce mémoire au probléme (P) suivant :

Soit $f : \mathbb{R}^n \to \mathbb{R}$. On cherche à résoudre le problème de minimisation sans contraintes suivant :

$$(P) : \ \min \{f(x) : x \in \mathbb{R}^n\} \tag{0.1}$$

Parmi les plus anciennes méthodes utilisées pour résoudre les problèmes du type (P), on peut citer la méthode du Gradient conjugué. Cette méthode est surtout utilisée pour les problèmes de grande taille.Cette méthode a été découverte en 1952 par Hestenes et Steifel , pour la minimisation de fonctionsquadratiques strictement convexes.Plusieurs mathématiciens ont étendu

cette méthode pour le cas non linéaire. Ceci a été réalisé pour la première fois, en 1964 par Fletcher et Reevese (*méthode de Fletcher-Reeves*) puis en 1969 par Polak, Ribière et Ployak (*méthode de Polak-Ribière-Ployak*). Une autre variante a été étudiée en 1987 par Fletcher (*Méthode de la descente conjuguée*). Une nouvelle variante a été proposée en 1991 par Liu et Storey (*Méthode de* Liu et Storey). Et enfin une dernière variante qui a été étudiée en 1999 par Dai et Yuan (*Méthode de* Dai et Yuan) Toutes ces méthodes génèrent une suite $\{x_k\}_{k\in\mathbb{N}}$ de la façon suivante :

$$x_{k+1} = x_k + \alpha_k d_k \tag{0.2}$$

Le pas $\alpha_k \in \mathbb{R}$ est déterminé par une optimisation unidimensionnelle ou recherche linéaire exacte ou inexacte.

Les directions d_k sont calculées de façon récurrente par les formules suivantes :

$$d_k = \begin{cases} -g_1 \quad \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} \quad \text{si } k \geq 2 \end{cases} \tag{0.3}$$

$g_k = \nabla f(x_k)$ et $\beta_k \in \mathbb{R}$.

Les différentes valeurs attribuées à β_k définissent les différentes formes du gradient conjugué.

$$\beta_k^{HS} = \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T(g_k - g_{k-1})}, \qquad \text{Hestenes-Stiefel [1], 1952.} \tag{0.4}$$

$$\beta_k^{FR} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}, \qquad \text{Fletcher Reeves [2], 1964} \tag{0.5}$$

$$\beta_k^{PRP} = \frac{g_k^T(g_k - g_{k-1})}{\|g_{k-1}\|^2} \quad \text{Polak-Ribière-Polyak [3 , 4], 1969} \tag{0.6}$$

$$\beta_k^{CD} = -\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}}, \qquad \text{descente conjugué. [5], 1987.} \tag{0.7}$$

$$\beta_k^{LS} = -\frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T g_{k-1}}, \qquad \textit{Liu-Storey } [6\,], 1991 \tag{0.8}$$

$$\beta_k^{DY} = \frac{g_k^T g_k}{(g_k - g_{k-1})^T d_{k-1}}, \qquad \textit{Dai-Yuan} \quad [7], 1999 \tag{0.9}$$

Dans ce travail, on s'nteresse à la méthode de GC de wyl qui'a été proposée par wyl en 2006.

$$B_k^{wyl} = \frac{g_k^T \left(g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1} \right)}{g_{k-1}^T g_{k-1}} \qquad [\ 8\ \ ] \qquad (0.10)$$

Le but de ce mémoire est la Recherche linèaire non monotone est proposé pour garante la convergence globale de la méthode du gradient conjugué sous la condition suffisante de décent de la méthode W Y L gradient conjugué sous la forte Wolfe de recherche linèaire

Ce mémoire comporte quatre chapitres.

Chapitre 1

On introduit dans ce chapitre les notions préléminaires de base concernant l'optimisation sans contraintes.et.quelques notions sur les problèmes de minimisation sans contraintes et leurs algorithmes

Chapitre 2

On expose dans ce chapitre les grandes lignes des méthodes d'optimisation sans contraintes basées sur les directions de descente et les recherches linéaires. Ces méthodes génèrent des suites $\{x_k\}_{k\in\mathbb{N}}$ de la façon suivante :

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

On suppose connaitre la direction de descente d_k au point x_k. La recherche linéaire consiste à trouver α_k de façon à diminuer la fonction f *suffisamment* le long de cette direction

.

Chapitre 3

Dans ce chapitre on étudie la méthode de gradient conjugué . cas des fonctions quadratiques strictement convexes. Et le cas des fonctions non quadratiques

Chapitre 4

Ce chapitre est la partie la plus importante de ce travail, car il contient La méthode de gradient conjugué de wei-yao-liu(**wyl**) sous la recherche linéaire inéxacte de wolfe-powell fort et la comparaison des perfermance de cette méthode avec certaines méthodes traditionnelles .

Chapitre 1

optimisation sans contrainte

Nous allons exposer dans ce premier chapitre quelques concepts clés concernant l'optimisation l'effort définitionnel portera surtout sur la méthode du gradient conjugué, où les questions suivantes seront prises en considération les problèmes proposés comportent-ils des contraintes ?

- les fonctions en jeu sont-elles quadratiques ? sont-elles convexes ?
- les domaines de définition des fonctions sont-ils continus où discrets ?

Chaque problème possède sa structure spécifique et demande donc d'être traiter de manière particulière .Le présent projet ne se focalisera que sur les problèmes d'optimisation sans contraintes et non linéaires.Ce problème sera formulé comme suit :

$$(P) \qquad \min f(x) \qquad x \in \mathbb{R}^n \tag{1.1}$$

$$\text{oú } f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

La majorité des méthodes de résolution d'optimisation qu'on étudie par la suite sont de nature itérative, c'est à dire qu'à partir d'un point initial x_1, elles engendrent une suite infinie $x_1, x_2, ..., x_k, ..$dont on espère qu'elle converge vers la solution optimale.

Pour construire des algorithmes de minimisation sans contraintes on fait appel à des processus itératifs du type :

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.2}$$

où d_k détermine la direction de déplacement à partir du point x_k et α_k est solution optimale du problème d'optimisation unidimensionnel suivant :

$$\underset{\alpha>0}{min} f\left(x_k + \alpha d_k\right),$$

c'est a' dire que α_k vérifié :

$$f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k), \forall \alpha > 0 \tag{1.3}$$

Le type d'algorithme permettant de résoudre le problème (1.1) sera déterminé dés qu'on définit les procèdés de construction du imisation vecteur d_k et de calcul de α_k à chaque itération.La façon avec laquelle on construit les vecteurs d_k et les scalaires α_k détermine directement les propriétés du processus et spécialement en ce qui concerne la convergence de la suite $\{x_k\}$, la vitesse de la convergence. Pour s'approcher de la solution optimale du problème (1.1) (dans le cas général, c'est un point en lequel ont lieu peut être avec une certaine précision les conditions nécessaires d'optimalité de f)

on se déplace naturellement à partir du point x_k dans la direction de la décroissance de la fonction f.

1.1 Rappel de quelques définitions

Nous introduisons ici les principales définitions de base qui seront utilisées par la suite.

Définition 1.1 *(Convergence des algorithmes)*

Un algorithme de résolution est un procédé qui permet, à partir de la donnée du point initial x_1, d'engendrer la suite $x_1, x_2 \ldots, x_k, \ldots$.Un algorithme est parfaitement défini par la donnée de l'application A qui associe à x_k le point $x_{k+1} \in A(x_k)$. Ceci permettra de confondre un algorithme et l'application A qui lui est associée.

Définition 1.2 *(Convergence globale)*

Nous dirons qu'un algorithme décrit par une application multivoque A est globalement convergent (ou encore : possède la propriété de convergence globale) si, quelque soit le point de départ x_1 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).

Définition 1.3 *(Modes de convergence)*

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

♦ *Si*

$$\limsup \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} = \tau < 1,$$

On dit que $\{x_k\}_{k\in N}$ *converge vers* x^* *linéairement avec le taux* τ.
⧫ *Si*

$$\frac{\|x_{k+1-}x_*\|}{\|x_k - x_*\|} \longrightarrow 0 \text{ quand } k \longrightarrow \infty,$$

on dit que la convergence est superlinéaire.
Plus précisément si $\exists p > 1$ *tel que :*

$$\limsup_{k\longrightarrow\infty} \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|^p} < +\infty,$$

on dit que la convergence est superlinéaire d'ordre p.
En particulier si

$$\limsup_{k\longrightarrow\infty} \frac{\left\|x^{k+1} - x^*\right\|}{\left\|x^k - x^*\right\|^2} < +\infty,$$

on dit que la convergence est quadratique (super linéaire d'ordre 2)

Définition 1.4 *(Ensemble convexe).*

Soit L'ensemble $C \subseteq \mathbb{R}^n$.C *est convexe si seulement si* $\forall x, y \in C^2, \forall t \in [0,1], tx + (1-t)y \in C$

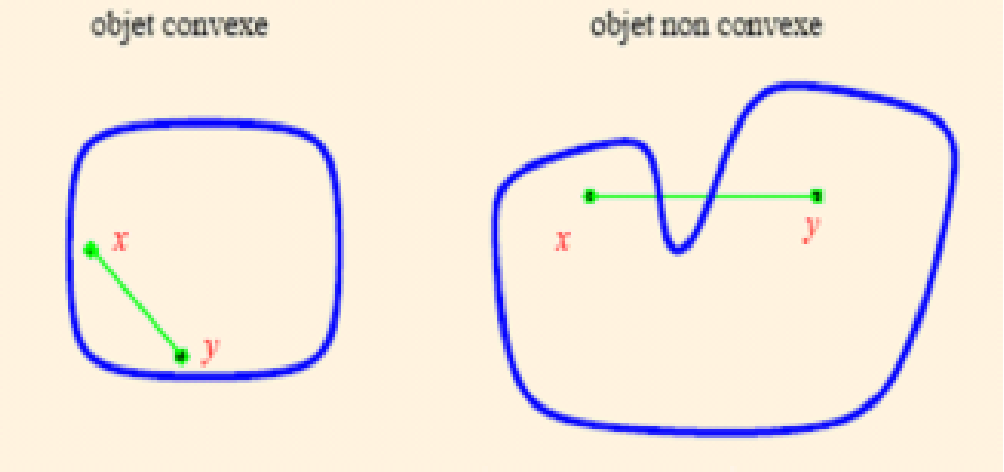


FIG. 1.1 – Objet convexe et non convexe

Objet convexe et non convexe

.

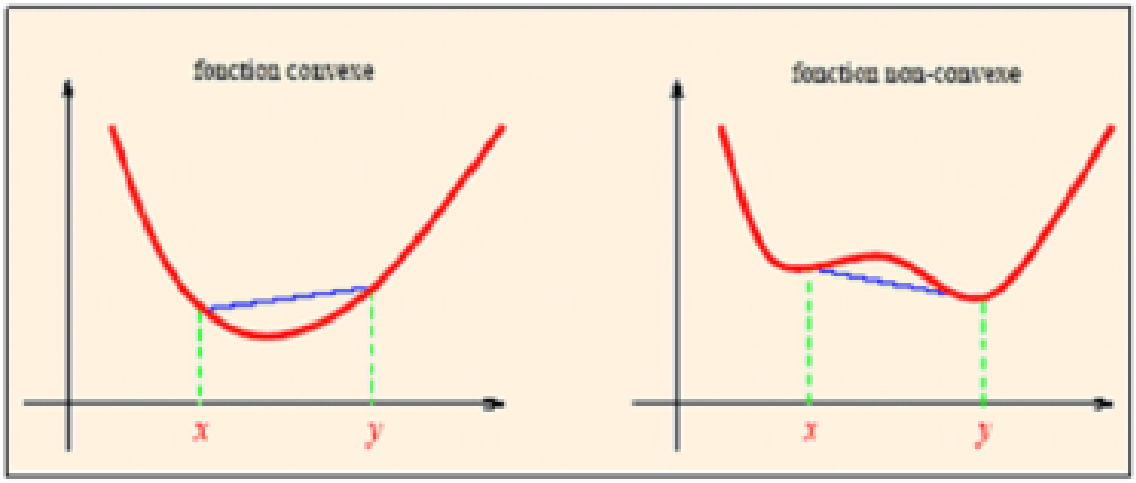


FIG. 1.2 – Fonction convexe et non convexe

Fonction convexe

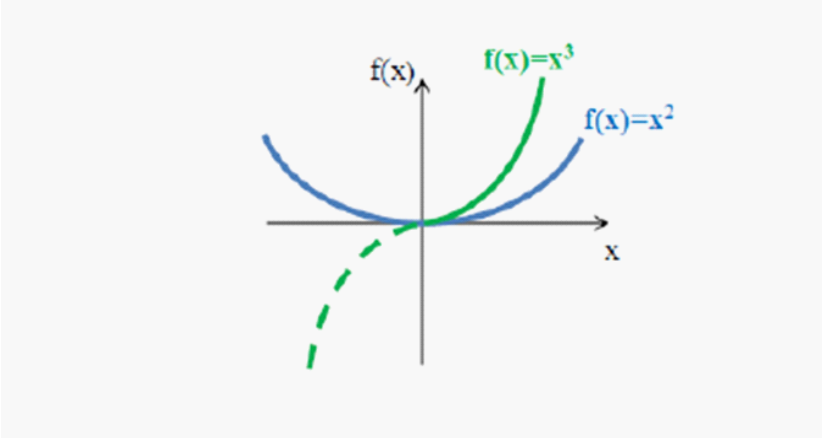
Définition 1.5 *Soit $C \subseteq \mathbb{R}^n$ un ensemble convexe non vide. Une fonction $f : C \to \mathbb{R}$ est convexe si seulement si*

$$\forall x, y \in C^2,\ \forall \lambda \in [0,1],\quad f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

Une fonction f est concave si $-f$ est convexe. On dira que f est strictement convexe dans C si seulement si

$$\forall x, y \in C^2, x \neq y, \forall \lambda \in [0,1],\quad f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y)$$

$f(x) = x^2$
$f'(x) = 2x$
$f''(x) = 2 \rightarrow$ f convexe sur$\mathbb{R}$
$f(x) = x^3$
$f'(x) = 3x^2$
$f''(x) = 6x \quad \rightarrow f$ convexe sur $\mathbb{R}^+$
$\rightarrow f$ non convexe sur $\mathbb{R}$



Définition 1.6 ***(Fonction convexe deux foix différentiable).***

Soit $C \subseteq \mathbb{R}^n$ non vide et $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ est dite deux foix diférentiable au point $x^* \in int(C)$ s'il existe un vecteur $\nabla f(x^*)$ et une matrice symétrique $H(x^*)$ d'ordre (n, n) appelée matrice hessienne, et une fonction $\alpha : \mathbb{R}^n \to \mathbb{R}$ telles que

$$\forall x \in C : f(x) = f(x^*) + \nabla f(x^*)^T (x - x^*) + \frac{1}{2}(x - x^*)^T H(x^*)(x - x^*) + \|x - x^*\|^2 \zeta(x^*, x - x^*) \tag{1.4}$$

ou : $\zeta(x^*, x - x^*) \underset{x \to x^*}{\longrightarrow} 0$. *On peut écrire la matrice hessienne $H(x^*)$* comme suit *:*

$$H(x^*) = \begin{bmatrix} \frac{\partial^2 f(x^*)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(x^*)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x^*)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x^*)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x^*)}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f(x^*)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 f(x^*)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x^*)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x^*)}{\partial x_n \partial x_n} \end{bmatrix}$$

♦ La matrice $H(x^*)$ est dite semi-définie positive si

$$\forall x \in \mathbb{R}^n : x^T H(x^*) x \geq 0$$

♦ La matrice $H(x^*)$ est dite définie positive si

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

Théorème 1.1 *soit $H(x^*)$ la matrice hessienne, on note par $\{\sigma_i\}_{i=1...n}$ses valeurs propres (réelles) on a les équivalences suivantes :*

- $H(x^*) \geq 0 \Longleftrightarrow \sigma_i \geq 0$
- $H(x^*) > 0 \Longleftrightarrow \sigma_i > 0$

1.2 Conditions d'optimalité des problèmes de minimisation sans contraintes :

Considérons le problème d'optimisatin sans contraintes (P)

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

oú $f : \mathbb{R}^n \longrightarrow \mathbb{R}$

1.2.1 Minima locaux et globaux

Définition 1.7 *1) $x^* \in \mathbb{R}^n$ est un minimum local de (P) si et seulement si il existe un voisinage $V_\varepsilon(x^*)$ tel que*

$$f(x^*) \leq f(x) : \forall x \in V_\varepsilon(x^*) \tag{1.5}$$

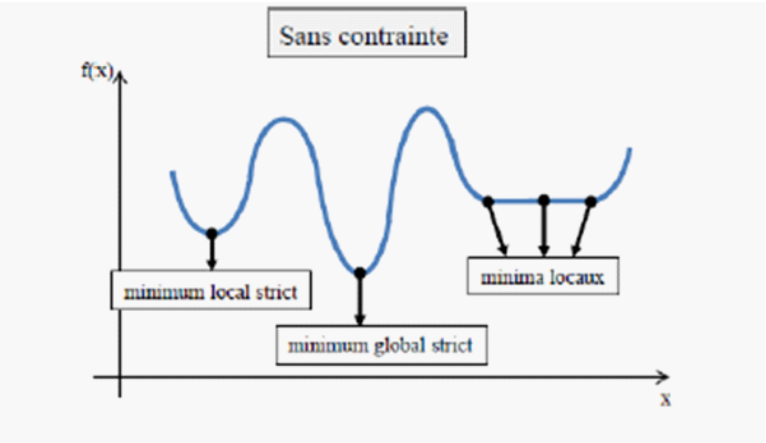
2) $x^* \in \mathbb{R}^n$ est un minimum local strict de (P) si et seulement si il existe un voisinage $V_\varepsilon(x^*)$ tel que

$$f(x^*) < f(x) : \forall x \in V_\varepsilon(x^*), x \neq x^* \tag{1.6}$$

3) $x^* \in \mathbb{R}^n$ est un minimum global de (P) si et seulement si

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

Remarque 1.1 *Dans le cas d'une fonction objectif convexe, il n'y a pas de distinction entre minimum local et global : tout minimum local est également global, comme l'établit le théorème suivant.*



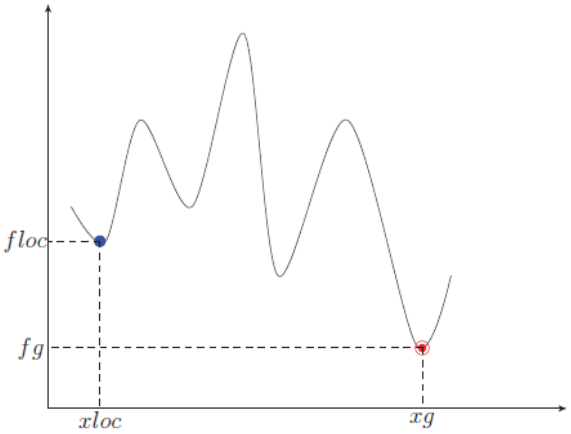


FIG. 1.1 – Optimum global et optimum local

Théorème 1.2 . *Soit* $f : \mathbb{R}^n \to \mathbb{R}$ *une fonction convexe définie sur l'ensemble convexe* X*. Alors, tout minimum local de* f *est également un minimum global. Si* f *est strictement convexe, alors il existe au plus un minimum global de* f

1.3 Direction de descente

Principes généraux : Considérons le problème d'optmisation sans contraintes
$(P) : (P) \min_{x \in \mathbb{R}^n} f(x)$,

ou'$f : \mathbb{R}^n \to \mathbb{R}$ est supposé régulière.

On note aussi $\nabla f(x)$ et $\nabla^2 f(x)$ le gradient et le hessien de f en x pour le produit scalaire euclidien sur $\mathbb{R}^n$.

On s'intéresse ici à une classe d'algorithmes qui sont fondés sur la notion de direction de descente.

Définition 1.8 *Soit d un vecteur non nul de $\mathbb{R}^n$. On dit que d est une direction de descente de f en $x \in \mathbb{R}^n$ si*

$$\nabla f(x)^T d < 0. \tag{1.9}$$

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

$$\theta = \arccos \frac{-\nabla f(x)^T . d}{\|\nabla f(x)\| \, \|d\|} \in \left[0, \frac{\pi}{2}\right[. \tag{1.10}$$

L'ensemble des directions de descente de f en x est le demi espace suivant $\left\{d \in \mathbb{R}^n : \nabla f(x)^T . d < 0\right\}$

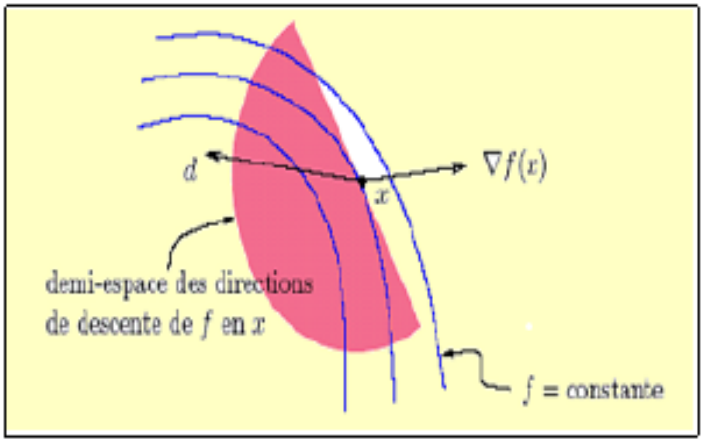


FIG. 1.3 – Demi-espace des directions de descente d de f en x.

On voit que si d est une direction de descente, $f(x + \alpha d) < f(x)$, pour tout $\alpha > 0$ suffisamment petit, et donc que f décroît strictement dans la direction d. De telles directions sont intéressantes en optimisation car, pour faire décroître f, il suffit de faire un déplacement le long de d. Les méthodes à directions de descentes utilisent cette idée pour minimiser une fonction . Elles construisent la suite des itérés $\{x_k\}_{k\geq 1}$, approchant une solution x_k du problème (1.1) par la récurrence :

$$x_{k+1} = x_k + \alpha_k d_k, \text{ pour } k \geq 1$$

Où α_k est appelé le pas et d_k la direction de descente de f en x_k.Pour définir une direction de descente il faut donc spécifier deux choses :

♦ dire comment la direction d_k est calculée, la manière de procéder donne le nom à l'algorithme ;

♦ dire comment on détermine le pas α_k, c'est ce que l'on appelle la recherche linéaire.

Remarque 1.2 *Par la suite l'angle θ_k entre la direction d_k et $-g_k$ jouera un role important dans le processus de la convergence. Pour θ_k on a la relation classique suivante :* $-g_k^T d_k = \|g_k\| \|d_k\| \cos\theta_k$

Définition 1.9 *Soit $f : x \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}, x^* \in \mathbb{R}^n$, $d \in \mathbb{R}^n$ est une direction de déscente au point x^* si et seulement si il existe $\delta > 0$ tel que*

$$f(x^* + \alpha d) < f(x^*) : \quad \forall \alpha \in]0, \delta[\,, \tag{1.11}$$

donnons maintenant une condition suffisante pour que soit d une direction de déscente.

Théorème 1.3 *Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ différentiable au point $x^* \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$ une direction vérifiant la condition suivante :*

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

Alors d est une direction de déscente au point x^*.

Proof. f est différentiable au pointx^*.Donc

$$f(x^* + \alpha d) = f(x^*) + \alpha \nabla f(x^*)^T .d + \alpha \|d\| \zeta(x^*, d)\,,$$

avec $\zeta(x^*, d) \underset{\alpha \longrightarrow 0}{\longrightarrow} 0$ ceci implique que

$$f'(x^*, d) = \lim_{\alpha \to 0} \frac{f(x^* + \alpha d) - f(x^*)}{\alpha} = \nabla f(x^*)^T .d < 0. \tag{1.12}$$

La limite étant strictement négative, alors il existe un voisinage de zéro $V(0)$ tel que

$$\frac{f(x^* + \alpha d) - f(x^*)}{\alpha} < 0, \qquad \forall \alpha \in V(0). \tag{1.13}$$

La relation (1.11) est particulièrement vraie pour tout $\alpha \in]-\delta, +\delta[$. On obtient le résultat cherché en multipliant la relation (1.11) par $\alpha > 0$ ■

Algorithme (méthode à directions de descente- une itération)

Etape 0 :(initialisation)

On suppose qu'au début de l'itération k, on dispose d'un itéré $x_k \in \mathbb{R}^n$

Etape1 :

Test d'arrêt : si $\|\nabla f(x_k)\| \simeq 0$, arrêt de l'algorithme ;

Etape2 :

Choix d'une direction de descente $d_k \in \mathbb{R}^n$;

Etape3 :

Recherche linéaire : déterminer un pas $\alpha_k > 0$ le long de d_k de manière à "faire décroître f suffisamment" ;

Etape4 :

Si la recherche linéaire réussie $x_{k+1} = x_k + \alpha_k d_k$;
remplacer k par $k+1$ et aller à l'étape 1.

1.3.1 Schéma général des algorithmes d'optimisation sans contraintes

Schéma général des algorithmes d'optimisation sans contraintes

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

$$x_{k+1} = x_k + \alpha_k d_k, \qquad \alpha_k \in]0, +\delta[$$

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

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

Un bon choix de d_k et α_k permet ainsi de construire une multitude d'algorithme 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. Dans le cas $d_k = -(H(x_k))^{-1}.\nabla f(x_k)$, on obtient La méthode de Newton, oú la matrice hessienne $H(x_k)$ est définie positive.

Exemple de choix de pas α_k :

On choisit en général α_k de façon optimale, c'est a dire que α_k doit vérifier

$$f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k) : \forall \alpha \in [0, +\infty[\,.$$

En d'autres temps on est ramené à étudier à chaque itération un problème de minimisation d'une variable réelle.C'est ce qu'on appèlle recherche linéaire

1.3.2 Conditions nécéssaires d'optimalité

Etant donné un vecteur x^*, nous souhaiterions être capables de déterminer si ce vecteur est un minimum local ou global de (P). La propriété de différentiabilité de f fournit une première manière de caractériser une solution optimale. Enonçons tout d'abord un théorème, pour établir une première condition nécéssaire d'optimalité.

Condition nécessaire d'optimalité du premier ordre

Théorème 1.4 *Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ différentiable au point $x^* \in \mathbb{R}^n$.Si x^* est un minimum local de (P), alors $\nabla f(x^*) = 0$.*

Proof. C'est une conséqence directe du théorème (1.2) .En effet, supposons que $\nabla f(x^*) \neq 0$. Puisque la direction $d = -\nabla f(x^*)$ est une direction de descente, alors il existe $\delta > 0$ tel que :

$$f(x^* + \alpha d) < f(x^*) : \qquad \forall \alpha \in]0, \delta[.$$

Ceci est contradiction avec le fait que x^* est une solution optimale locale de (P). ■

Remarque 1.3 *si f est convexe, la condition nécéssaire du premier ordre est également suffisante pour que x^* soit un minimum global. Dans le cas où f est deux fois différentiable, une autre condition nécéssaire est donnée par le théorème (1.5). Elle est appelée condition nécéssaire du second ordre car elle fait intervenir la matrice hessienne de f (que nous noterons $\nabla^2 f(x)$, dont les éléments sont ses secondes dérivées partielles).*

Conditions nécéssaires d'optimalité du second ordre

Théorème 1.5 *Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ deux fois différentiable au point $x^* \in \mathbb{R}^n$.Si x^* est un minimum local de (P), alors $\nabla f(x^*) = 0$ et $H(x^*)$ est semi définie positive.*

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

$$f(x^* + \alpha x) = f(x^*) + \frac{1}{2}\alpha^2 x^T H(x^*)x + \alpha^2 \|x\|^2 \zeta(x^*, \alpha x), \quad \zeta(x^*, \alpha x) \underset{\alpha \to 0}{\to} 0.$$

Ceci implique

$$\frac{f(x^* + \alpha x) - f(x^*)}{\alpha^2} = \frac{1}{2} x^T H(x^*) x + \|x\|^2 \zeta(x^*, \alpha x). \tag{1.14}$$

x^* est un optimum local, il existe alors $\delta > 0$ tel que

$$\frac{f(x^* + \alpha x) - f(x^*)}{\alpha^2} \geq 0, \qquad \forall \alpha \in]-\delta, +\delta[\,.$$

Si on prend en considération (1.12) et on passe à la limite quand $\alpha \to 0$, $\alpha \neq 0$,on obtient :

$$x^T H(x^*) x \geq 0, \qquad \forall x \in \mathbb{R}^n.$$

■

1.3.3 Conditions suffisantes d'optimalité

Les conditions données précédemment sont nécéssaires (si f n'est pas convexe), c'est-à-dire qu'elles doivent être satisfaites pour tout minimum local ; cependant, tout vecteur vérifiant ces conditions n'est pas nécessairement un minimum local. Le théorème (1.5) établit une condition suffisante pour qu'un vecteur soit un minimum local, si f est deux fois différentiable.

Condition suffisante d'optimalité du second ordre

Théorème 1.6 *Soient $f : \mathbb{R}^n \to \mathbb{R}$ deux fois différentiable au point $x^* \in \mathbb{R}^n$. Si $\nabla f(x^*) = 0$ et $H(x^*)$ est définie positive, alors x^* est un minimum local strict de (P).*

Remarque 1.4 *La matrice H est définie positive ssi toutes ses valeurs propres sont positives*

Chapitre 2

Recherche linéaire exacte et inexacte

2.1 Recherche linéaire

2.1.1 Introduction

Considérons le problème d'optimisation sans contraintes P.

$$(P) : \min_{x \in \mathbb{R}^n} f(x), \tag{2.1}$$

où $f : \mathbb{R}^n \longrightarrow \mathbb{R}$.

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

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.2}$$

où α_k est solution optimale du problème d'optimisation unidimensionnel suivant :

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

c'est a' dire que α_k vérifie

$$f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k), \forall \alpha > 0,$$

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

$$\alpha \to \varphi(\alpha) = f(x_k + \alpha d_k). \tag{2.3}$$

Il faut noter que dans les problème d'optimisation sans contraintes on a besoin de résoudre à chaque Itération x_k, un problème d'optimisation dans $\mathbb{R}$

Dans ce chapitre nous allons décrire les différentes manières de déterminer un pas $\alpha_k > 0$ le long d'une direction de descente d_k. C'est ce que l'on appèlle faire de la recherche linéaire.

Il existe deux grandes classes de méthodes qui s'intéressent a' l'optimisation unidimensionnelle :

- ▶ les recherches linéaires exactes.
- ▶ les recherches linéaires inexactes.

2.1.2 Objectifs de la recherche linéaire

Il s'agit de réaliser deux objectifs :

Le premier objectif, faire décroître f suffusamment, et pour celà on cherche à vérifier l'inégalité :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \text{“}\textit{un terme négatif}\text{”} \tag{2.4}$$

Le terme négatif joue un rôle-clé dans la convergence de l'algorithme.

Le second objectif, on choisit le pas $\alpha_k > 0$ d'être trop petit, pour assurer la convergence d'algorithme au point stationnaire .

Le premier objectif n'est en effet pas suffisant car l'inégalité (2.4) est en général satisfaite par des pas $\alpha_k > 0$arbitrairement petit. Or ceci peut entraîner une “fausse convergence”, c'est-à- dire la convergence des itérés vers un point non stationnaire,

2.1.3 But de la recherche linéaire

Dans le cas non-quadratique les méthodes de descente (2.2), nécéssitent la recherche d'une valeur de $\alpha_k > 0$ optimale ou non, vérfiant :

$$f(x_k + \alpha_k d_k) \leq f(x_k). \tag{2.5}$$

Rappellons que si f est différentiable, le pas optimalα^* peut être caractérisé par

$$\varphi'(\alpha^*) = 0,$$

$$\varphi(\alpha^*) \leq \varphi(\alpha), \ \textit{pour}\ 0 < \alpha \leq \alpha^*,$$

autrement dit, α^* est un minimum local de φ qui assure de plus la décroissance de f. En fait, dans la plupart des algorithmes d'optimisation modernes, on ne fait jamais de recherche linéaire exacte, car trouver α^* signifie qu'il va falloir calculer un grand nombre de fois la fonction φ, et cela peut être dissuasif du point de vue du temps de calcul. En pratique, on recherche plutôt une valeur de α qui assure une décroissance suffisante de f. Cela conduit à la notion d'intervalle de sécurité.

2.1.4 Intervalle de sécurité

Définition 2.1 *On dit que $[a, b]$ est un intervalle de sécurité s'il permet de classer les valeurs deα de la façon suivante :*

- *Si $\alpha < a$ alors α est considéré trop petit.*
- *Si $b \geq \alpha \geq a$ alors αest satisfaisant.*
- *Si $\alpha > b$ alors t est considéré trop grand.*

Le problème est de traduire de façon numérique sur φ les trois conditions précédentes, ainsi que de trouver un algorithme permettant de déterminer a et b. L'idée est de partir d'un intervalle suffisamment grand pour contenir $[a, b]$, et d'appliquer une bonne stratégie pour itérativement réduire cet intervalle.

2.1.5 Algorithme de base

Etape 0 : (initialisation)
$a = b = 0$,choisir $\alpha_1 > 0$, poser $k = 1$ et aller à l'étape 1 ;

Etape 1 :
Si α_k convient, poser $\alpha^* = \alpha_k$ et on s'arrête.
Si α_k est trop petit on prend $a_{k+1} = \alpha_k$, $b_{k+1} = b$,
et on va à l'étape 2 .
Si α_k est trop grand on prend $b_{k+1} = \alpha_k$, $a_{k+1} = a$,
et on va à l'étape 2 .

Etape 2 :
Si $b_{k+1} = 0$ déterminer $\alpha_{k+1} \in]a_{k+1}, +\infty[$.
Si $b_{k+1} \neq 0$ déterminer $\alpha_{k+1} \in]a_{k+1}, b_{k+1}[$,
remplacer k par $k + 1$ et aller à l'étape 1.

Il faut maintenant préciser quelles sont les relations sur φ qui va nous permettre de caractériser les valeurs de α convenables, ainsi que les techniques utilisées pour réduire l'intervalle.

2.2 Recherches linéaires "exactes"

Comme on cherche à minimiser f, il semble naturel de chercher a minimiser le critère le long de d_k et donc de déterminer le pas α_k comme solution du problème

$$\underset{\alpha>0}{min}\varphi(\alpha)$$

C'est ce que l'on appèlle la règle de Cauchy et le pas déterminé par cette règle est appelé pas de Cauchy ou pas optimal Dans certains cas, on préférera le plus petit point stationnaire de φ qui fait décroître par cette fonction :

$$\alpha_k = \inf\{\alpha > 0 : \varphi'(\alpha) = 0, \varphi(\alpha) < \varphi(0)\}.$$

On parle alors de règle de Curry et le pas déterminé par cette règle est appelé pas de Curry. De manière un peu imprécise, ces deux règles sont parfois qualifiées de recherche linéaire exacte. Elles ne sont utilisées que

dans des cas particuliers, par exemple lorsque φ est quadratique, la solution de la recherche linéaire s'obtient de façon exacte et dans un nombre fini d'itérations.

2.3 Recherches linéaires inexactes

On considère la situation qui est typique pour l'application de la technique de recherche linéaire à l'intérieur de la méthode principale multidimensionnelle. Sur une itération k de la dernière méthode nous avons l'itération courante $x_k \in \mathbb{R}^n$ et la direction de recherche $d_k \in \mathbb{R}^n$ qui est direction de descente pour notre objectif : $f : \mathbb{R}^n \to \mathbb{R}$

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

Le but est de réduire "de façon importante"la valeur de l'objectif par un pas $x_k \to x_{k+1} = x_k + \alpha_k d_k$ de x_k dans la direction d_k. Pour celà de nombreux mathématiciens (Armijo[9], Goldstein, Wolfe[10], Albaali[11], Touati-Ahmed[12], Lemaréchal, Fletcher...) ont élaboré plusieurs règles.

Schéma des recherches linéaires inexactes

Elles reviennent à déterminer, par tâtonnement un intervalle $[a\ , b]$, où $\alpha^* \in [a\ , b]$, dans lequel :

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

Le schéma de l'algorithme est donc :

Algorithme (Schéma général des recherches linéaires inexactes)
Etape 0 : (initialisation)
$a_1 = b_1 = 0$, choisir $\alpha_1 > 0$, poser $k = 1$ et aller à l'étape 1 :
Etape 1 :
Si α_k est satisfaisant (suivant un certain critère) : STOP($\alpha^* = \alpha_k$).
Siα_k est trop petit (suivant un certain critère) : nouvel intervalle : $[a_{k+1} = \alpha_k, b_{k+1} = b]$,
et aller à l'étape 2.
Etape 2 :
Si $b_{k+1} = 0$ déterminer $\alpha_{k+1} \in]a_{k+1}, +\infty[$.
Si $a_{k+1} \neq 0$ déterminer $\alpha_{k+1} \in]a_{k+1}, b_{k+1}[$,
remplacer k par $k + 1$ et aller à l'étape 1.

Il nous reste donc à décider selon quel(s) critère(s) est trop petit ou trop grand ou satisfaisant

2.3.1 Caractérisation de l'intervalle de sécurité

la règle d'Armijo

Une condition naturelle est de demander que f décroisse autant qu'une portion $\delta \in]0, 1[$ de ce que ferait le modèle linéaire de f en x_k. Celà conduit à l'inégalité suivante, parfois appelée condition d'Armijo ou condition de décroissance linéaire :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k \nabla f(x_k)^T d_k. \tag{2.6}$$

Elle est de la forme (2.4), car δ devra être choisi dans $]0, 1[$. On voit bien à la figure (2.1) ce que signifie cette condition. Il faut qu'en α_k, la fonction prenne une valeur plus petite que celle prise par la fonction affine

$$\alpha \rightarrow f(x_k) + \delta \alpha \nabla^T f(x_k) d_k. \tag{2.7}$$

Règle d'Armijo
♦ Si $\varphi(\alpha) \leq \varphi(0) + \delta\varphi'(0)\alpha$, alors α convient.
♦ Si $\varphi(\alpha) > \varphi(0) + \delta\varphi'(0)\alpha$, alors α est trop grand.
On peut noter que l'on a

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

$$\varphi(0) = f(x_k),$$

$$\varphi\prime(0) = \nabla f(x_k)^T d_k.$$

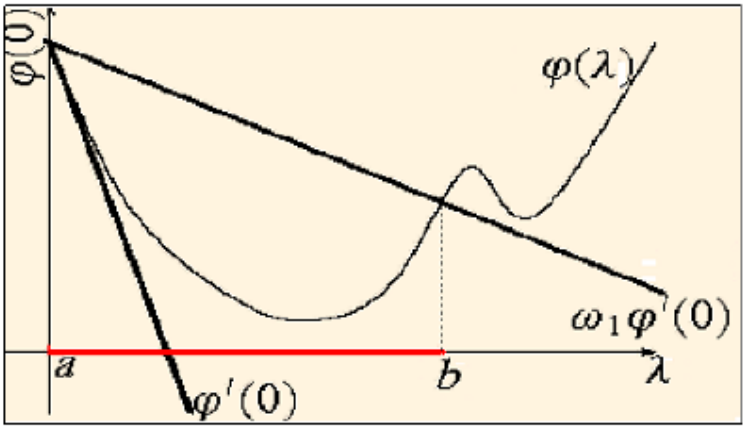


FIG. 2.1 – La règle d'Armijo

Algorithme 2.1 (Règle d'Armijo)

Etape 0 : (initialisation)

$a_1 = b_1 = 0$, choisir $t_1 > 0$, $\omega_1 \in]0, 1[$ poser $k = 1$ et aller à l'étape 1.

Etape 1 :

Si $\varphi(\alpha_k) \leq \varphi(0) + \delta\varphi'(0)\alpha_k$: STOP $(\alpha^* = \alpha_k)$.
Si $\varphi(\alpha_k) > \varphi(0) + \delta\varphi'(0)\alpha_k$, alors
$b_{k+1} = b$, $a_{k+1} = \alpha_k$ et aller à l'étape 2.

Etape 2 :

Si $b_{k+1} = 0$ déterminer $\alpha_{k+1} \in]\ a_{k+1}\ , +\infty\ [$,
Si $b_{k+1} \neq 0$ déterminer $\alpha_{k+1} \in]\ a_{k+1}\ , b_{k+1}[$,
remplacer k par $k + 1$ et aller à l'étape 1.

Remarque 2.1

1) En pratique, la constante δ est prise trés petite, de manière à satisfaire (2.6) le plus facilement possible. Typiquement, $\delta = 10^{-4}$.Notons que cette constante ne doit pas être adaptée aux données du problème et donc que l'on ne se trouve pas devant un choix de valeur délicat.

2) Dans certains algorithmes, il est important de prendre $\delta < \frac{1}{2}$pour que le pas α_k soit accepté lorsque x_k est proche d'une solution.

3) Il est clair d'aprés la figure (2.1) que l'inégalité (2.6) est toujours vérifiée si $\alpha_k > 0$ est suffisamment petit.

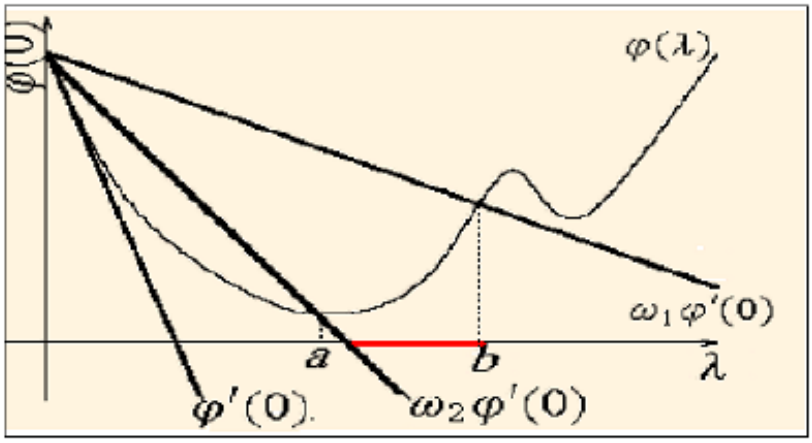


FIG. 2.2 – La règle de Goldestein

On a vu qu'il etait dangereux d'accepter des pas trop petits, cela pouvait conduire a une fausse convergence. Il faut donc un mécanisme supplémentaire qui empêche le pas d'être trop petit. On utilise souvent la technique de rebroussement due à *Armijo* [9, 1966] ou celle de *Goldstein.*

La règle de Goldstein

La règle de *Goldstein* remédie à cet inconvénient (le pas α_k doit être trop petit). Dans celle-ci, en ajoutant une deuxième inégalité à la règle d'Armijo on obtient la règle de *Goldstein.*

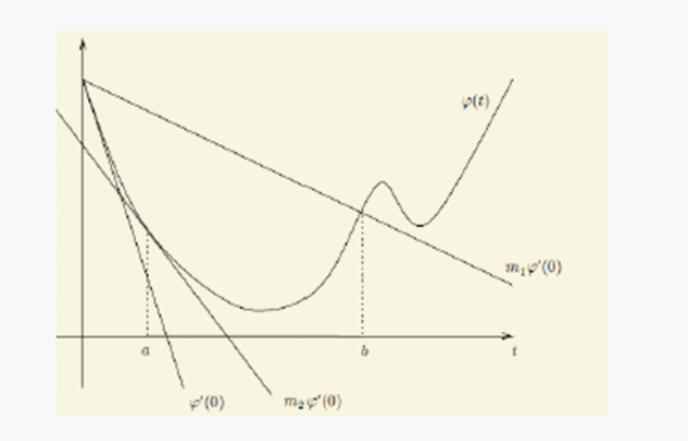
$$f(x_k) + \delta\alpha_k \nabla^T f(x_k) d_k \geq f\left(x_k + \alpha_k d_k\right) \geq f(x_k) + \sigma\alpha_k \nabla^T f(x_k) d_k, \quad (2.8)$$

où δ et σ sont deux constantes vérifiant $0 < \delta < \sigma < 1$, cette inégalité qui empêche le pas d'être trop petit.

la règle de Wolf :

La règle de *Wolfe* fait appel au calcul de $\varphi\prime(t)$, elle est donc en théorie plus coûteuse que la règle de *Goldstein.* Cependant dans de nombreuses applications, le calcul du gradient $\nabla f(x)$ représente un faible coût additionnel en comparaison du coût d'évaluation de $f(x)$, c'est pourquoi cette règle est très utilisée.

Nous allons présenter les conditions de *Wolfe faibles* sur $\alpha > 0$:



Règle de wolf faible

$$f(x_k + \alpha d_k) \leq f(x_k) + \delta\alpha\nabla f(x_k)^T.d_k.$$

$$\nabla f(x_k + \alpha d_k)^T.d_k \geq \sigma\nabla f(x_k)^T.d_k.$$

avec $0 < \delta < \sigma < 1$.

Pour expliquer ces conditions posons

$$l(\alpha) = \phi(0) + (\sigma\phi'(0))\alpha.$$

Règle de *Wolfe faible*

- Si $\phi(\alpha) \leq \phi(0) + \delta\varphi\prime(0)\alpha$ et $\varphi\prime(\alpha) \geq \sigma\varphi\prime(0)$, alors α convient.
- Si $\phi(\alpha) > \phi(0) + \delta\varphi\prime(0)\alpha$, alors α est trop grand.
- Si $\varphi\prime(\alpha) < \sigma\varphi\prime(0)$, alors α est trop petit.

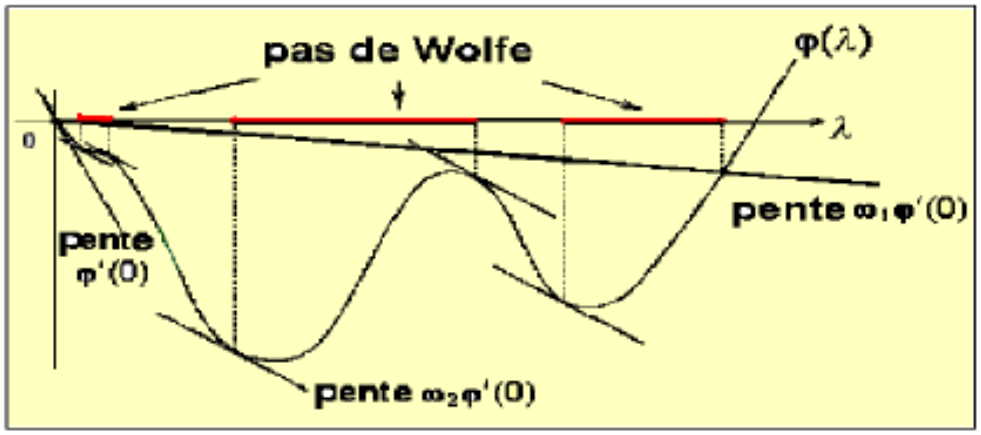


FIG. 2.3 – La règle de Wolfe

Algorithme 2.3 (Règle de *Wolfe*)

Etape 0 : (initialisation)

$a_1 = b_1 = 0$, choisir $t_1 > 0$ $\delta \in]0, 1[$, $\sigma \in]\delta, 1[$, poser $k = 1$ et aller à l'étape 1.

Etape 1 :

Si $\phi(\alpha_k) \leq \phi(0) + \delta\phi'(0)\alpha_k$ et $\phi'(t) \geq \sigma\phi'(0)$: STOP($\alpha^* = \alpha_k$).

Si $\phi(\alpha_k) > \phi(0) + \delta\phi'(0)\alpha_k$, alors

$b_{k+1} = \alpha_k$,$a_{k+1} = a_k$, et aller à l'étape 2.

Si $\phi'(t) < \sigma\phi'(0)$, alors $b_{k+1} = b_k$, $a_{k+1} = \alpha_k$ et aller à l'étape 2.

Etape 2 :

Si $b_{k+1} = 0$ déterminer $\alpha_{k+1} \in]a_{k+1}, +\infty[$

Si $b_{k+1} \neq 0$ déterminer $\alpha_{k+1} \in]a_{k+1}, b_{k+1}[$.

Règle de *Wolfe* Forte On obtient des contraintes plus fortes si l'on remplace $(W2)$ par

$$\left|\nabla f(x_k + \alpha d_k)^T . d_k\right| \leq -\sigma \nabla f(x_k)^T . d_k$$

Les $(W1)$ et $(W3)$ sont les conditions de *Wolfe fortes*. La contrainte $(W3)$ entraîne que $\sigma\phi'(0) \leq \phi'(\alpha) \leq -\sigma\phi'(0)$ c-à-d.$\phi'(\alpha)$ n'est pas"trop"

Remarque 2.2

1) On voit bien que les conditions de *Wolfe fortes* impliquent les conditions de *Wolfe faibles*,en effet :

$$\begin{aligned} \left|\nabla^T f(x_k+\alpha_k d_k)d_k\right| &\leq -\sigma\nabla^T f(x_k)d_k \\ &\Leftrightarrow \sigma\nabla^T f(x_k)d_k \leq \nabla^T f(x_k+\alpha_k d_k)d_k \leq -\sigma\nabla^T f(x_k)d_k \\ &\Rightarrow \sigma\nabla^T f(x_k)d_k \leq \nabla^T f(x_k+\alpha_k d_k)d_k. \qquad (W2) \end{aligned}$$

2) L'existence de valeurs α de vérifiant les conditions de *Wolfe faibles* et *fortes*. est donnée par la Proposition suivante.

Proposition 2.1 .*Soit $f \in C^1$ et d une direction de descente de f en x ,on suppose que f est minorée . Alors, si $0 < \delta < \sigma < 1$, il existe des intervalles dans $\mathbb{R}_+$ qui vérifient les conditions de Wolfe faibles et fortes.*

2.4 Convergence des méthodes à directions de descente

2.4.1 Condition de Zoutendijk

Dans cette section on va étudier la contribution de la recherche linéaire inexacte à la convergence des algorithmes à directions de descente. On dit qu'une règle de recherche

linéaire satisfait la condition de *Zoutendijk*[13] s'il existe une constante $C > 0$ telle que pour tout indice $k \geq 1$ on ait

$$f(x_{k+1}) \leq f(x_k) - C\left\|\nabla f(x_k)\right\|^2 \cos^2\theta_k, \qquad (2.10)$$

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

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

Voici comment on se sert de la condition de *Zoutendijk*.

Proposition 2.2 *Si la suite* $\{x_k\}$ *générée par un algorithme d'optimisation vérifie la condition de Zoutendijk* (2.10) *et si la suite* $\{f(x_k)\}$ *est minorée, alors*

$$\sum_{k\geq 1} \|\nabla f(x_k)\|^2 \cos^2\theta_k < \infty. \tag{2.11}$$

Démonstration

En sommant les inégalités (2.10), on a

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

La série est donc convergente puisqu'il existe une constante C' telle que pour tout k, $f(x_k) \geq C'$.Il ya des propositions précisent les circonstances dans les quelles la condition de *Zoutendijk* (2.10) est vérifie avec les règles de la recherche linéaire exacte (*Cauchy,Curry*) et aussi les règles de la recherche linéaire inexacte (*Armijo, Wolfe*).
qui concernant la proposition qui est vérifie la condition de *Zoutendijk* avec la règle de *Wolfe*.

Proposition 2.3 *Soit* $f : \mathbb{R}^n \to \mathbb{R}$ *une fonction contiuement différentiable dans un voisinage de* $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ *.On considère un algorithme à directions de descente* d_k*, qui génère une suite* $\{x_k\}$ *en utilisant la recherche linéaire de Wolfe (w1)-(w2).*

Alors il existe une constante $C > 0$ *telle que, pour tout* $k \geq 1$*, la condition de Zoutendijk* (2.10) *est vérifiée.*

Démonstration

Noton $g_k = \nabla f(x_k)$ et $g_{k+1} = \nabla f(x_k + \alpha_k d_k)$.
D'après $(W2)$

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k$$

$$\begin{aligned} \Rightarrow \quad & \left(g_{k+1} - g_k\right)^T d_k \geq (\sigma - 1)\, g_k^T d_k \\ = \quad & -(1-\sigma)\, g_k^T d_k = (1-\sigma)\left|g_k^T d_k\right| \\ \Leftrightarrow \quad & (1-\sigma)\left|g_k^T d_k\right| \leq \left(g_{k+1} - g_k\right)^T d_k, \end{aligned}$$

et du fait que f est continement différentiable :

$$\begin{aligned}(1-\sigma)\left|g_k^T d_k\right| &= (1-\sigma)\|g_k\|\|d_k\|\cos\theta_k \\ &\leq \|g_{k+1}-g_k\|\|d_k\| \\ &\Rightarrow \quad (1-\sigma)\|g_k\|\cos\theta_k \leq L\alpha_k\|d_k\| \\ &\Rightarrow \quad \alpha_k\|d_k\| \leq \frac{(1-\sigma)}{L}\|g_k\|\cos\theta_k,\end{aligned}$$

en utilisant (w1), on aura :

$$\begin{aligned}f(x_k+\alpha_k d_k) &\leq f(x_k)+\delta\alpha_k g_k^T d_k \\ &\Rightarrow \quad f(x_k+\alpha d_k) \leq f(x_k)+\delta\alpha g_k^T d_k \leq f(x_k)+\left|\delta\alpha g_k^T d_k\right| \\ &\Rightarrow \quad f(x_k+\alpha d_k) \leq f(x_k)+\delta\alpha\left|g_k^T d_k\right| \leq f(x_k)-\delta\alpha g_k^T d_k \\ &\Rightarrow \quad f(x_k+\alpha d_k) \leq f(x_k)-\delta\alpha\|g_k\|\|d_k\|\cos\theta_k \\ &\Rightarrow \quad f(x_k+\alpha d_k) \leq f(x_k)-\frac{\delta(1-\sigma)}{L}\|g_k\|^2\cos^2\theta_k.\end{aligned}$$

On en déduit (2.10).

Chapitre 3

Méthode du gradient conjugué :

3.1 Optimisation quadratique sans contrainte

3.1.1 définition et théorème fondamentaux

Soit A est une matrice (n, n) symétrique et définie positive et $b \in \mathbb{R}^n$. On

appelle problème de minimisation quadratique sans contrainte, le problème noté (P)

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} x^T A x - b^T x \right\} \tag{p}$$

Théorème 3.1 *Le problème (P) a une solution unique x^*. solution du système linéaire $Ax = b$ c'est-à-dire que x^* vérifié*

$$x^* = A^{-1} b \tag{3.1}$$

Proof. Considérons $f(x) = \frac{1}{2} x^T A x - b^T x$. notons $H(x)$ la matrice Hessienne de f au point x . On a ■

$$H(x) = A, \forall x \in \mathbb{R}^n \tag{3.2}$$

Rappelons le théorème de l'analyse convexe.

Soit $f : S \to \mathbb{R} \subset \mathbb{R}^n$ convexe. Alors f est convexe sur S si et seulement si $H(x)$ est semi définie positive pour tout $x \in S$ De plus si f est strictement quadratique de la forme

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

Alors f est strictement convexe si est seulement si $H(x)$ est définie positive pour tout $x \in \mathbb{R}^n$ Pour

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

on a $H(x) = A$, $\forall x \in \mathbb{R}^n$. Par hypothèse A est définie positive, donc

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

est strictement convexe dans $\mathbb{R}^n$.

Rappelons maintenant ce deuxième résultat de l'optimisation convexe.Soit $S \subset \mathbb{R}^n$ convexe et ouvert et $f : S \to \mathbb{R}$ strictement convexe et (P) le problème d'optimisation convexe suivant

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

Alors $x^* \in \mathbb{R}^n$ est solution optimale de (P) si et seulement si

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

De plus si x^* existe, alors elle est unique. Appliquons ce résultat à notre problème.

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

est strictement convexe.

$S = \mathbb{R}^n$ est convexe et ouvert. $x \in \mathbb{R}^n$ est solution optimale de (P) si et seulement si

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

Or

$$\nabla f(x) = Ax - b$$

Donc

$$\nabla f(x) = 0 \iff Ax - b = 0 \iff Ax = b \qquad (3.3)$$

Donc x^* solution optimale de (P)si et seulement si x^* vérifie

$$Ax^* = b \tag{3.4}$$

A étant définie positive,alors A est inversible. Par conséquent x^* solution de (3.4) est donnée par

$$x^* = A^{-1}b \tag{3.5}$$

Calcul du pas obtenu par une recherche linéaire exacte

A est une matrice (n, n) symétrique et définie positive et $b \in \mathbb{R}^n$ et

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

considérons le problème (P)

$$\left\{ \min_{x \in \mathbb{R}^n} \frac{1}{2}x^T Ax - b^T x \right\} \tag{p}$$

les méthode a direction de recherche linéaire génèrent des suites $\{x_k\}_{k=1,2.....}$ de la manière suivante

- On démarre par $x_1 \in \mathbb{R}^n$ a l'itération k si on a $x_k \in \mathbb{R}^n$
- Le successeur x_{k+1} de x_k donnée par la relation suivante

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

où $d_k \in \mathbb{R}^n$ est une direction de recherche et $\alpha_k \in \mathbb{R}^n$ est le pas de recherche obtenu par une recherche linéaire exacte ou inexacte. Dans le cas d'une recherche linéaire exacte α_k vérifie

$$f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k) \tag{3.6}$$

Notons

$$g_k = \nabla f(x_k) = Ax_k - b \tag{3.7}$$

Théorème 3.2 *Soit A est une matrice (n, n) symétrique et définie positive et $b \in \mathbb{R}^n$ et*

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

. Considérons le problème(P)

$$\min_{x\in\mathbb{R}^n} f(x) = \left\{ \min_{x\in\mathbb{R}^n} \frac{1}{2}x^T Ax - b^T x \right\} \qquad \text{(p)}$$

Supposons qu'à l'itération k on a une direction d_k de descente, c'est à dire que d_k vérifie

$$g_k^t d_k = (Ax_k - b)^t d_k < 0 \tag{3.8}$$

soit $\alpha_k > 0$ obtenue par une recherche linéaire exacte c-à-dire que α_k vérifie

$$f(x_k + \alpha_k d_k) = \min_{\alpha>0} f(x_k + \alpha d_k)$$

Alors

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T A d_k} \tag{3.9}$$

preuve :considérons

$$\varphi(\alpha_k) = f(x_k + \alpha d_k) \tag{3.10}$$

Donc α_k est la solution optimale du problème unidimensionnel suivant

$$\varphi(\alpha_k) = \min_{\alpha\succ 0} \varphi(\alpha) \tag{3.11}$$

A définie positive, Alors

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

est strictement convexe sur $\mathbb{R}$.
Par conséquent

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

est strictement convexe sur $]0, \infty[$ qui est un ouvert convexe.
Donc α_k solution optimale de (3.12) si et seulement α_k vérifie

$$\varphi\prime(\alpha) = 0$$

Or
$\varphi\prime(\alpha) = \nabla f(x_k + \alpha d_k)^T d_k$
$= [A(x_k + \alpha d_k) - b]^T d_k$
$= [Ax_k - b + \alpha A d_k]^T d_k$
$= [g_k + \alpha A d_k]^T d_k$
$= g_k^T d_k + \alpha d_k^T A d_k$
Donc

$$\varphi\prime(\alpha) = 0 \iff \alpha d_k^T A d_k = -g_k^T d_k$$

encore

$$\alpha = -\frac{g_k^T d_k}{d_k^T A d_k}$$

Remarque 3.1 *Si A n'est pas définie positive mais seulement semi définie positive, alors f n'est pas strictement convexe, et $d_k^T A d_k$ peut être égale à zéro. Par conséquent α_k ne sera plus définie par* (3.9).

3.1.2 Méthode des directions conjuguées

Définitions et propriétés générales

Définition 3.1 *Soit A une matrice (n, n) symétrique. Les directions $d_0, d_1, ...d_k$, sont dites A conjuguées si on a :*

$$d_i^T A d_j = 0, \qquad 0 \leq i, j \leq k$$

Théorème 3.3 *Soit A une matrice (n, n) symétrique et définie positive. Si les directions $d_0, d_1, ...d_k$ avec $k \leq n-1$,sont non nuls et A conjuguées, alors ils sont linéairement indépendants.*

Proof. Supposons que ■

$$\alpha_0 d_0 + \alpha_1 d_1 + ... + \alpha_k d_k = 0 \tag{3.12}$$

Multiplions l'égalité (3.12) par d_j^T A , on obtient

$$0 = \sum_{i,j=0}^{k} \alpha_j d_j^T A d_j = \alpha_j d_j^T A d_j \tag{3.13}$$

Or

$$\alpha_j d_j^T A d_j = 0 \implies \alpha_j = 0 \tag{3.14}$$

car A est définie positive et par conséquent $d_j^T\ Ad_j > 0$ est libre.

L'algorithme de directions conjuguées

Soit A une matrice (n, n) symétrique et définie positive et $b \in \mathbb{R}^n$. Considérons le problème de minimisation quadratique sans contrainte, (P), suivant

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} x^T A x - b^T x \right\} \qquad \text{p}$$

Algorithme des directions conjuguées

a) **Initialisation**

On se donne $x_0 \in \mathbb{R}$ quelconque et $d_0, d_1, ...d_{n-1}A$, conjuguées, Poser $k = 0$ et aller à

b) **Étape principale**

Pour $k \geq 0$

Calculez

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T A d_k}$$

et

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

Poser $k = k + 1$ et aller à b.

3.1.3 Méthode du gradient conjugué

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

l'idée initiale était de trouver une suite de direction de descente permettant de résoudre le problème

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

Où f est régulière (continument différentiable)

– Dans la méthode des directions conjuguées,les directions $d_0, ...d_{n-1}$, sont données à l'avance.

– Dans la méthode du gradient conjugué, On démarre d'un point $x_0 \in \mathbb{R}^n$,

$$\begin{aligned} d_0 &= -g_0 \\ &= \nabla f(x_0) \\ &= Ax_0 - b \end{aligned}$$

les directions d_k, $k = 1, .., n-1$ sont calculés à chaque itération à l'itération k

$$d_K = -g_k + \beta_{k-1} d_{k-1}$$

β_{k-1} est obtenu de sorte que d_k et A soit conjugués avec les autres vecteurs $d_j, i = 0, ...k-1$

En d'autre terme on doit avoir

$$d_k^T A d_i = 0, i = 0, ...k-1 \tag{3.15}$$

Dans l'appellation gradient conjugué, on trouve les deux mots gradient et conjugué

a) Le mot gradient est utilisé car d_k est calculé à partir du gradient au point x_k

b) Le mot conjugué est aussi justifié, car et comme on le verra plus loin, les directions

$\{d_k\}_{k=0}^{n-1}$ sont A conjugués

Algorithme du gradient conjugué . "cas quadratique"

Principe de l'algorithme

– On démarre d'un point quelconque $x_0 \in \mathbb{R}^n$

– Calculez

$$\begin{aligned} d_0 &= -g_0 \\ &= b - Ax_0 \\ \alpha_0 &= -\frac{g_0^T d_0}{d_0^T A d_0} \end{aligned}$$

Supposons qu'à l'itération k on ait x_k et d_k . Ceci nous permettra de calculer

$$g_k = Ax_k - b, \alpha_k = -\frac{g_k^T d_k}{d_k^T A d_k},\ x_{k+1} = x_k + \alpha_k d_k, g_{k+1} = Ax_{k+1} - b$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k \tag{3.16}$$

β_k est choisi de sorte que

$$d_{k+1}^T A d_k = 0 \tag{3.17}$$

Puisque

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

alors (3.17) donne

$$\left(-g_{k+1} + \beta_k d_k\right)^T A d_k = 0$$

Ou encore

$$\beta_k d_k^T A d_k = g_{k+1}^T A d_k \tag{3.18}$$

et finalement

$$\beta_k = \frac{g_{k+1}^T A d_k}{d_k^T A d_k} \tag{3.19}$$

Cas quadratique

Soit A une matrice (n, n), symétrique et définie positive. On considère dans ce paragraphe le problème (P) suivant

$$\min\left\{f\left(x\right) : x \in \mathbb{R}^n\right\} = \left\{\frac{1}{2} x^T A x - b^T x : x \in \mathbb{R}^n\right\} \tag{p}$$

Algorithme

Algorithme du gradient conjugué. Cas quadratique

1. choisir $x_0 \in \mathbb{R}^n$
2. Calculez $g_0 = A\ x_0 - b, g_0 = 0$ stop. sinon poser $d_0 = -g_0$ Poser $k = 0$
3. Calculez $\alpha_k = -\frac{g_k^T d_k}{d_k^T d_k}$
4. Calculez $x_{k+1} = x_k + \alpha_k d_k$
5. Calculez $g_{k+1} = A\ x_{k+1} - b$ si $g_{k+1} = 0$ stop
6. Calculez $\beta_k = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}$
7. Calculez $d_{k+1} = -\ g_{k+1} + \beta_k d_k$
8. Poser $k = k + 1$ et allez à 3

Cas des fonctions non quadratiques

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

$$\min\{f(x) : x \in \mathbb{R}^n\} \tag{3.20}$$

Les méthodes du gradient conjugué pour réseaux cette problème sont des méthodes itératifs de la forme

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.21}$$

Le pas $\alpha_k \in \mathbb{R}$ étant déterminé par une recherche linéaire .la direction d_k est définie par la formule de récurrence suivante ($\beta_K \in \mathbb{R}$)

$$d_k = \begin{cases} -g_0 & \text{si} \quad k = 0 \\ -g + \beta_{k-1} d_{k-1} & \text{si} \quad k \geq 1 \end{cases} \tag{3.22}$$

Ces méthodes sont des extensions de la méthode du gradient conjugué linéaire du cas quadratique , si β_k prend l'une des valeurs :

$$\beta_k^{PRP}, \beta_k^{HS}, \beta_k^{LS}, \beta_k^{DY}, \beta_k^{CD}, \beta_k^{FR} \tag{3.23}$$

Remarque 3.2 *Dans le cas où f n'est pas quadratique on a*

$$\beta_k^{FR} \neq \beta_k^{PRP} \neq \beta_k^{HS} \neq \beta_k^{LS} \neq \beta_k^{DY} \neq \beta_k^{CD}$$

Par conséquent, en appliquant l'algorithme du gradient conjugué non quadratique , en utilisant les coefficients β_k , on obtient des suites$\{x_k\}_{k\in\mathbb{N}}$différentes.

Que se passe t'il si f est quadratique strictement convexe et si α_k est obtenue par une recherche linéaire exacte .

Remarque :Dans le cas quadratique on a vu que :

$$\beta_{k+1}^{HS} = \beta_{k+1}^{PRP} = \beta_{k+1}^{FR} = \beta_{k+1}^{CD} = \beta_{k+1}^{DY}$$

Algorithme du gradient conjugué non quadratique

Soit $f : \mathbb{R}^n \to \mathbb{R}$ non quadratique et (PNQ) le problème de minimisation non quadratique sans contrainte suivant

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

Pour construire l'algorithme du gradient conjugué cas non quadratique , on peut s'inspirer de l'algorithme du gradient conjugué quadratique établi au chapitre précédent. Contrairement au cas quadratique , on n'a pas de matrice A. Par conséquent on n'a pas de direction conjugué comme dans le cas quadratique , l'algorithme du gradient conjugué cas non quadratique génère une suite $\{x_k\}_{k\in N}$ de la manière suivante

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

L'algorithme démarre d'un un point $x_0 \in \mathbb{R}^n$ quelconque a l'itération k

Supposons qu'on ait le vecteur $x_k \in \mathbb{R}^n$ et la direction d_{k-1} . Ceci nous permet de calculer $\nabla f(x_k)$ au lieu $g_k = Ax - b$ dans le cas quadratique. Pour avoir x_{k+1} , on a besoin de calculer α_k et d_k calculer de d_k

$$d_k = -\nabla f(x_K)\beta_{k-1}d_{k-1} \tag{3.24}$$

On a huit façon pour calculer β_{k-1}

$$\beta_{k-1}^{HS}, \beta_{k-1}^{PRP}, \beta_{k-1}^{FR}, \beta_{k-1}^{DY}, \beta_{k-1}^{LS}, \beta_{k-1}^{CD}$$

Calculer de α_k :

Ayant obtenu d_k, rappelons que α_k vérifie

$$f(x_k + \alpha_K d_k) = \min\{f(x_k + \alpha d_k) : \alpha \in]0, \infty[\} \tag{3.25}$$

Dans le cas où

$$f(x) = \tfrac{1}{2}x^T Ax - b^T x$$

A symétrique définie positive , α_k solution de (3.26) est donne par la relation suivan

$$\alpha_k = \frac{g_k^T d_k}{d_k^T A d_k} \tag{3.26}$$

Dans le cas où f n'est pas quadratique, α_k ne peut pas être calculée par la form (3.26) . On calcule dans ce cas α_k par d'autre méthode. On utilise par exemple la méthode du dichotomie ou la méthode de point fixe. Comme on le verra plus loin , α_k peut être calculée par une recherche linéaire inexacte d'Armijo ou Goldstien ou Wolfe.

3.1.4 Algorithme du gradient conjugué non quadratique

Algorithme Étape 1

Choisir $x_0 \in \mathbb{R}^n$ quelconque et $\varepsilon > 0$

Étape 2 Poser $k = 0$

Calculer $g_0 = \nabla f(x_0)$ .Poser $d_0 = -g_0$

Étape 3 Calculez α_k en utilisant une recherche linéaire exacte où inexacte d'Armijo ou de Goldstien ou de Wolfe Forte

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

Étape 4

Si : $\|\nabla f(x_{K+1})\| < \varepsilon$

stop,Sinon allez à étape 5

Étape 5

Calculez $g_{k+1} = \nabla f(x_{K+1})$ Calculez β_k par l'une des manière suivantes

$\beta_k = \beta_k^{HS}$ ou $\beta_k = \beta_k^{FR}$ ou $\beta_k = \beta_k^{PRP}$ ou $\beta_k = \beta_k^{CD}$ ou $\beta_k = \beta_k^{LS}$ ou $\beta_k = \beta_k^{DY}$

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

Posez $k = k + 1$ et allez à l'étape 3

Chapitre 4

comparaison des performance d'une nouvelle méthode de gradient conjugué

4.1 introdiction

La méthode de gradient conjugué (CG) a joué un rôle particulier pour la résolution à grande problèmes d'optimisation non linéaire en raison de la simplicité de leur itération et leurs besoins en mémoire très faibles. En fait, la méthode de laCG est pas parmi les algorithmes d'optimisation les plus rapides ou plus robuste pour les problèmes non linéaires disponibles aujourd'hui, mais il reste très populaire pour les ingénieurs et les mathématiciens qui sont intéressés à résoudre les grands problèmes .

La méthode non linéaire de gradient conjugué est conçu pour résoudre les problèmes d'optimisation sans contrainte suivant

$$\min \{f(x) : x \in \mathbb{R}^n\} \tag{4.1}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est une fonction non linéaire, lisse dont le gradient est symbolisée par la fonction g :La formule itérative du la methode de gradient conjugué est donnée par :

$$x_{k+1} = x_k + \alpha_k d_k \tag{4.2}$$

où $\alpha_k$est une longueur de pas obtenu par une recherche linéaire unidimensionnelle, et d_k est le sens de la recherche défini par

$$d_k = \left\{ \begin{array}{ll} -g_1 & si\ k = 1\ , \\ -g_k + \beta_k d_{k-1} & .....si\ k \geq 2, \end{array} \right\} \tag{4.3}$$

où $g_k = \nabla f(x_k)$; et β_K est un scalaire. Il ya au moins six formules pour β_K ,qui sont donnés ci-dessous

$$\beta_K^{HS} = -\frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})} Hestenes - Stiefel\,[1]\,, 1952 \tag{4.4}$$

$$\beta_K^{FR} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}\ FletcherReeves\ \ [2]\,,1964 \tag{4.5}$$

$$\beta_K^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{\|g_{k-1}\|^2}\ Polak - Ribière - Polyak\ \ [3,4]\,, 1969 \tag{4.6}$$

$$\beta_K^{CD} = -\frac{g_k^T g_k}{d_{K-1}^T g_{k-1}}\ descenteconjugué.\ \ [5]\, 1987 \tag{4.7}$$

$$\beta_K^{LS} = -\frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T g_{k-1}}\ Liu - Storey\ \ [6]\,, 1991 \tag{4.8}$$

$$\beta_K^{DY} = \frac{g_k^T g_k}{(g_k - g_{k-1})^T d_{k-1}}\ Dai - yuan\ \ [7]\,, 1999 \tag{4.9}$$

Récemment, Wei et al. [8]a proposé une nouvelle formule

$$\beta_K^{wyl} = \frac{g_k^T \left(g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1} \right)}{g_{k-1}^T g_{k-1}} \qquad [8] \tag{4.10}$$

la formule WYL a non seulement agréable mais aussi des résultats numériques est globalement convergent avec la recherche linéaire exact et les conditions de recherche lineairé de $Grippo - Lucidi$ [15]la méthode proposée est globalement convergent. Donc, il est évidemment que la condition suffisante de descente est important de la méthode WYL. Dans cet section nous allons montrer que la formule WYL avec la recherche linéaire de $Wolfe - Powell$ forte posséder la condition suffisante de descente.Les α_k vérifient les relations de $wolfe$ forte suivantes :

$$f(x_k + \alpha_k d_k) - f\,(x_k) \leq \delta \alpha_k g_K^T d_K \tag{4.11}$$

$$\left|g\left(x_k+\alpha_k d_k\right)^T d_k\right| \leq \sigma\left|g_K^T d_k\right| \tag{4.12}$$

Où $\delta \in \left(0, \frac{1}{2}\right)$ et $\sigma \in (\delta, 1)$

4.1.1 La preuve de la condition suffisante de descente

Le théorème suivant montre que la formule WYL avec la resherche lineaire de SWP possèdent la condition suffisante de descente

Théorème 4.1 *Supposer que les séquences $\{g_k\}$ et $\{d_k\}$ sont générés par la méthode de la forme (4.2),(4.3) et (4.10) et le pas α_k est déterminée par la recherche lineaire de $Wolf-Powell$ fort (4.11) et (4.12), si $\sigma < \frac{1}{4}$ donc la séquence $\{d_k\}$possèdent la condition suffisante de descente*

$$g_K^T d_k \leq -c\left\|g_k\right\|^2 \tag{4.13}$$

Proof. Nous avons d'abord prouver la propriété de descente de $\{d_k\}$:Par (4.3) ■

nous avons $g_K^T d_k = -\left\|g_k\right\|^2 + \beta_K^{wyl} g_k^T d_{k-1}$ combinant cette équation avec $\left(\beta_K^{wyl}\right)$,nous avons

$$\frac{g_k^T d_k}{\left\|g_k\right\|^2} = -1 + \frac{\sigma g_{k-1}^T d_{k-1}}{\left\|g_{k-1}\right\|^2}\left(1 - \frac{g_k^T g_{k-1}}{\left\|g_k\right\|\left\|g_{k-1}\right\|}\right) \leq \frac{g_k^T d_k}{\left\|g_k\right\|^2} \leq -1 + \frac{-\sigma g_{k-1}^T d_{k-1}}{\left\|g_{k-1}\right\|^2}\left(1 - \frac{g_k^T g_{k-1}}{\left\|g_k\right\|\left\|g_{k-1}\right\|}\right) \tag{4.14}$$

puisque $g_1^T d_1 = -\left\|g_1\right\|^2 < 0$ si $g_1 \neq 0$,Supposons maintenant que $d_i; i = 1, 2, 3, ...k-1$

Sont toutes les directions de descente, Ex: $g_k^T d_k < 0$

De l'inégalité de $Cauchy - Schwarz$, nous obtenons

$$0 \leq 1 - \frac{g_k^T g_{k-1}}{\left\|g_k\right\|\left\|g_{k-1}\right\|} \leq 2 \tag{4.15}$$

(4.14) et (4.15) en déduire

$$-1 + 2\sigma\frac{g_{k-1}^T d_{k-1}}{\left\|g_{k-1}\right\|^2} \leq \frac{g_k^T d_k}{\left\|g_k\right\|^2} \leq -1 - 2\sigma\frac{g_{k-1}^T d_{k-1}}{\left\|g_{k-1}\right\|^2} \tag{4.16}$$

En répétant ce processus et le fait $g_1^T d_1 = -\left\|g_k\right\|^2$,nous avons

$$-\sum_{j=0}^{k-1}(2\sigma)^j \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -2+\sum_{j=0}^{k-1}(2\sigma)^j \tag{4.17}$$

puisque
$\sum_{j=0}^{k-1}(2\sigma)^j < \sum_{j=0}^{\infty}(2\sigma)^j = \frac{1}{1-2\sigma}$
(4.17)peut être écrite comme

$$-\frac{1}{1-2\sigma} \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -2+\frac{1}{1-2\sigma} \tag{4.18}$$

En faisant la restriction $\sigma \in \left(0, \frac{1}{4}\right)$ nous avons $g_k^T d_k < 0$,Donc par induction $\forall k \in \mathbb{N} g_k^T d_k < 0$ est vérifiée

Maintenant, nous prouvons la propriété suffisante descente de d_k .si $\sigma \in \left(0, \frac{1}{4}\right)$

set $c = 2 - \frac{1}{1-2\sigma}$ puis

$0 < c < 1$et (4.18) s'avère être

$c - 2 \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -c$

$$c - 2 \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -c \tag{4.19}$$

$\Longrightarrow g_k^T d_k \leq -c\|g_k\|^2$ ou $C = -2 + \frac{1}{1-2\sigma}$
ce qui signifie que (4.13) est vérifiée.

4.1.2 propriétés convergentes

Dans ce paragraphe ,nous allons réintroduire les propriétés de convergence de la formule (4.10) , (4.11) et (4.12)

Les résultats réintroduits sont donnés dans $[16-19]$, de sorte que les preuves de détail sont négligés.

Pour plus de commodité, nous avons d'abord introduire certaines hypothèses et des lemmes qui sont souvent utilisés dans les cours de méthodes de gradient conjugué..

Maintenant nous donnons l'algorithme suivant to d'abord

.Étape 1 :point initiale

$x_1 \in \mathbb{R}^n, \varepsilon \geq 0$,fixe $d_1 = -g_1, si\ \|g_1\| \leq \varepsilon$ alors stop

Étape 2 :Calculer α_k par quelques recherches linéaires

$\sigma = 0.01$,et $\sigma = 0.1$

Étape 3 : defnir $x_{k+1} = x_k + \alpha_k d_k$, si $\|g_{k+1}\| \leq \varepsilon$ alors stop $\varepsilon = 10^{-6}$
Étape 4 :Calculer β_K par quelques formules et générer d_{k+1}par (4.3)
Étape 4 :poser $K = k + 1$,passez à l'étape 2.

Sur les études de méthodes de gradient conjugué la condition suffisante descente (4.13)

joue un rôle important. Malheureusement,cette condition ne peut être satisfaite pour de nombreuses.méthodes de gradient conjugué

Hypothéses A **:** L'ensemble $\Omega := \{x \in \mathbb{R}^n; f(x) \leq f(x_1)\}$est borné où $x_1 \in \mathbb{R}^n$est le point initial.

Hypothéses B **:** Dans certains voisinage N de Ω , $f(x)$ est continûment différentiable et son gradient est lipschitz continu, c'est-à-dire, pour toutes $x, y \in N$, il existe une constante $L \geq 0$ telle que

$$\|g(x) - g(y)\| \leq L\|x - y\| \tag{4.20}$$

Lemme 4.1 . *Supposons que les hypothèses A et B ont vérifiée. considérer la méthode sous la forme d'*(1.2) *et* (1.3) *,où* d_k*satisfait* $g_k^T d_k < 0$ *, pour*

tout k, et α_k est obtenu par SWP (4.11) et (4.12) alors,

$$\sum_{K\geq 1} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < +\infty \tag{4.21}$$

Dèmonstration.
nous avons $g_k^T d_k < 0$ pour tout $K \geq 1$Nous avons aussi de (4.12) et(4.20)

$$-(1-\sigma) g_k^T d_k \leq (g_{k+1} - g_k)^T d_k \leq L\alpha_K \|d_k\|^2 \tag{4.22}$$

$$\alpha_k \geq \frac{-(1-\sigma)}{L} \frac{g_k^T d_k}{\|d_k\|^2} \tag{4.23}$$

laquelle associant (4.11), nous obtenons

$$f(x_k) - f(x_k + \alpha_K d_k) \geq -\delta\alpha_k g_k^T d_k \geq \delta \frac{(1-\sigma)}{L} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} \tag{4.24}$$

En outre,àpartirde Hypothèse(A) nous avons$\{f(x_k)\}$est une suite décroissante et a une limite ci-dessous
dans Ω , et montre $\lim_{k\to\infty} f(x_{k+1}) < +\infty$ et (4.24) présente

$$+\infty > f(x_1) - \lim_{k\to\infty} f(x_{k+1}) = \sum |f(x_k) - f(x_{k+1})| \geq \delta \frac{(1-\sigma)}{L} \sum \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} \tag{4.25}$$

Nous pouvons conclure que (4.21) est vérifiée.

4.2 Résultats numériques et discussion

Dans cette section, nous présentons les résultats numériques de notre méthode proposée basée sur la comparaison avec les méthodesCG de FR, PRP, HS, DY , LS

Ces comparaisons sont basé sur le nombre d'itérations et le temps CPU.

Nous avons examiné les performances numériques des formules proposées sur quelques fonctions quadratiques et non quadratique ; Nous avons considère $\|g_k\| < 10^{-6}$ comme critère d'arrêt.$\delta = 0.01$, et $\sigma = 0.1$ Nous utilisons la recherche linéaire inexacte et en utilisons matlab 2010

Fonctions testes :

Les résultats numériques ont été obtenus sur les fonctions suivantes :

Problème1 :$f(x,y) = x^2+4y^2-4x-8y$ avec $x_0 = (0,0)\,, x^* = (2,1)\,, f(x^*) =$ 0

Problème2 :$f\,(x,y) = (x-2)^4+(x-2y)$avec$x_0 = (0,0)\,, x^* = (2,1)\,, f(x^*) =$ 0

$(x-2)^4+(x-$

$(x+2y-7)^2 + (2x+y-5)^2$

la fonction $f\,(x,y) = (x+2y-7)^2 + (2x+y-5)$ (Bouth function) avec $x^* = (1,3)\,, f(x^*) = 0$

application des méthodes de gradients sur fonction quadratique. Nous testons

les six méthodes GC suivantes :

PRP, FR, WYL , DY, CD, LS

La condition d'arrêt est $\|g_k\| < 10^{-6}$

Dans cette partie, on adopter les notations suivantes :

CPU time : le temps

NI : nombre d'itérations

$\|\|$ la norme du gradient

4.2.1 Probléme 3 :

Considérons le probéme quadratiqu suivante :

$$f(x) = x^2 + 4y^2 - 4x - 8y = \frac{1}{2}x^T \begin{pmatrix} 2 & 0 \\ 0 & 8 \end{pmatrix} x - \begin{pmatrix} 4 \\ 8 \end{pmatrix} x$$

La solution optimal est(2.1)

la méthode du gradient à pas fixe, à pas optimal, puis le gradient conjuguée pour comparer ces méthode on choisit de démarrer les algorithmes que désencadrer au point de coordonnée $x_0 = (0, 0)$

Méthode du gradient à pas fixe

Pour tester cette méthode, on pourra choisir par exemple un pas initial alpha= $0,001$, alpha= $0,8$, alpha= $0,20$

Tableau 1 : Résultats numériques de la méthode du gradient à pas fixe

alpha	*NI*	*CPU time*	*la solution*	$\|g\|$
0.001	999	0.159	(1.72 0.99)	0.5413
0.8	467	0.123	échoué	échoué
0.20	34	0.003	(1.99 ;1)	9×10^{-7}

la Convergence n'est pas assuré dans le cas 2 ou alpha= 0.8 (plus grand)

Dans le troisiéme cas la convergence vers la solution optimal (2.1) est mieux

Méthode du gradient à pas optimale

$\alpha_k = \frac{g_k^T g_k}{g_k^T A g_k}\,(exacte)$

Tableau 2 : Résultats numériques de la méthode du gradient à pas optimale

alpha	*NI*	*CPU time*	*la solution*	$\|g\|$
0,14705	25	0,0087	(1,99 ;1)	$3{,}7 \times 10^{-7}$

l'algorithme converge vers le $x^* = (1,99 \quad 1)$ qui est trés proche du point optimale détecte théoriquement $x^* = (2 \quad 1)$

Méthode du gradient conjugué à pas optimale

$\alpha_k = \frac{g_k^T g_k}{g_k^T A g_k}\,(exacte)$

Tableau 3 : Comparaison la performance de la méthode de *wyl* avec d'autre

méthode	β_K	*le pas* α_K	*NI*	*CPU time*	*la solution*
FR	0,1245	0,147	2	0,00147	(2,1)
PRP	0,1245	0,147	2	0,00120	(2,1)
wyl	0,1245	0,147	2	0,001	(2,1)
HS	0,1245	0,147	2	0.03	(2,1)
DY	0,1245	0,147	2	0.002	(2,1)
CD	0,1245	0,147	2	0.002	(2,1)
LS	0,1245	0,147	2	0.0021	(2,1)

Remarque 4.1 *Dans le cas quadratique avec recherche linéaire exacte on a vu que* $\beta_K^{FR} = \beta_K^{PRP} = \beta_K^{LS} = \beta_K^{CD} = \beta_K^{DY}$

Probléme:Le minimum de la fonction $f(x,y) = (x-2)^4 + (x-2y)^2$ correspondant au point du coordonn à $(2,1)$

Le gradient de cette fonction ce calcul de la maniére suivante :

$\frac{\partial f(x,y)}{\partial x} = 4(x-2)^3 + 2(x-2y)$

$\frac{\partial f(x,y)}{\partial y} = -4(x-2y)$

La norme du gradient ce calcul de la maniére suivante :

$\|g\| = \sqrt{g_x^2 + g_y^2}$

g_x et g_y représentant la composante en x et en y du gradient g

Méthode du gradient à pas fixe

Remarque 4.2 *Tableau* 4 : *résultats numériques pour la fonctionf* (x,y)

point initiale x_0	*le pas*	NI	CPU *time*	*le* min	$\|g\|$
$(-1.50; -0.0175)$	0.01	3674	9.0609	$(1.934; 0.96)$	$9.98 * 10^{-4}$
$(-1.40; -4.45)$	0.01	3686	9.0617	$(1.946; 0.9673)$	$9.99 * 10^{-4}$
$(-1.65; -0.017)$	0.07	*échoué*	0.01756	*échoué*	

Dans le tableau 4, nous notons que la méthode ne converge pas si le pas est est trop grand comme dans le troisiéme cas. Généralement lorsque le pas est fixée, nous avons besoin d'un grand nombre d'itérations, c'est l'inconvénient de cette méthode

Méthode du gradient conjugué Le programme Matlab pour la méhode du gradient conjugué trouvé à [14]

Résultats numériques pour la fonction : $f(x,y) = (x-2)^4 + (x-2y)$

$x_0 = (-1,5; -0,5)$

La condition d'arrét est $\|g_k\| < 10^{-6}$

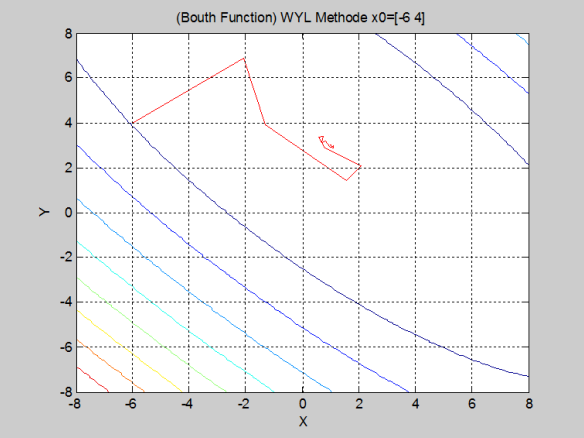
Tableau 5 : Comparaison la performance de la méthode de WYL avec d'autre méthode des gradient conjugué

	WYL	FR	PRP	HS	DY	CD	
NI	9	36	16	20	36	36	
CPU	0.32	1.3864	0.58	0.86	1.43	1.43	
$Xoptimal$	$(1.99; 1)$	$(1.96; 0.98)$	$(1.9; 0.99)$	$(1.93; 0.97)$	$(1.95; 0.98)$	$(1.95; 0.98)$	$($
$\| g \|$	$2.5 * 10^{-4}$	$9.54 * 10^{-4}$	$3.04 * 10^{-4}$	$8.77 * 10^{-4}$	$9.8 * 10^{-4}$	$9.8 * 10^{-4}$	8

La méthode WYL fonctionné mieux que d'autre méthode et converge rapide

vers la solution optimale. Moins d'itération et moins de temps.

Il est clair que la méthode de WYL est un peut mieux que les autre méthode Résultats numériques pour la fonction $f(x,y) = (x+2y-7)^2 +$



$(2x + y - 5)^2$(Bouth function) avec la méthode du gradient conjugué nous avons utilisé le

code Matlab [30]

$x_0 = (-6, 4)$ la condition d'arrét est $\|g_k\| < 10^{-6}$

Tableau 6 : Comparaison la performance de la méthode de WYL avec d'autres méthodes du gradient conjugué

	WYL	FR	PRP	HS	ZDY	CD	LS
NI	41	99	49	*échoué*	*échoué*	53	23
CPU	0.1429	0.2448	0.3202	1.0045	*échoué*	2.1136	0.9623
$Xoptimal$	$(1;3)$	$(1;3)$	$(1;3)$			$(1;3)$	$(1;3)$
$\|g\|$	$1.1 * 10^{-7}$	$2.15 * 10^{-7}$	$8.042 * 10^{-8}$			$3.48 * 10^{-7}$	$2.79 * 10$

Dans le tableau 6, on note que la méthode wyl est plus rapide que les autres méthodes

Conclusion 4.2 *Dans ce travail, nous avons etudie une nouvelle méthode*(WYL) *du gradient conjugué qui garante la condition suffisante de descente et nous avons prouvé qu'elle converge globalement vers la solution optimale sous la recherche linéaire inéxacte Les résultats préliminaires numériques montrent que la performance de la méthode* WYL

Bibliographie

[1] M.R. Hestenes,E. Stiefel, Method of conjugate gradient for solving linear equations, J. Res. Nat. Bur. Stand. 49 (1952) 409ñ436..

[2] R. Fletcher, Reeves, Function minimization by conjugate gradients, Compute. J. 7 (1964) 149ñ154

[3] B.T. Polyak (1969), The conjugate gradient method in extremem problems, Comput. Math. Math. Phys., 9, pp. 94-112...

[4] E. Polak and G. RibiËre (1969), Note sur la convergence de directions conjuguÈes, Rev. FranÁaise Informat. Recherche Operationelle, 3e annÈe, 16, pp. 35-43.

[5] R. Fletcher (1987), Practical methods of optimization, John Wiley&Sons, Chichster

[6] Y. Liu, C.Storey, E¢ cient generalized conjugate gradient algorithms, part 1 : theory, J. Optim. TheoryAppl. 69 (1992) 129ñ137.

[7] Y.H. Dai and Y. Yuan (1999), A non linear conjugate gradient with a strong global convergence property, SIAM J. Optimization, Vol. 10(1),pp.177-182.

[8] Lin Suihua, Yao Shengwei, A new conjugate gradient method of modified LS formula,$Department of Mathematics and Information Science, Guangxi University, 2006.

[9] L. Armijo (1966), Minimzation of function having lipschitz continous . rst partial derivatives, Paci. . . c Journal of Mathematics, Vol. 16(1), pp.1-3.

[10] M.J.D. Powell (1986), Convergence properties of algorithms for nonlinear optimization, SIAM rev., 28, pp. 487-500

[11] M. Al-Baali (1985), Descent property and global convergence of the Fletcher-Reeves method with inexacte line search. IMA J. Num. Anal., Vol. 5, pp.121-124.

[12] D. Touati-Ahmed and C. Storey (1990), Ec/ cient hybrid conjugate gradient techniques, JOTA, 64, pp. 379-397.

[13] G. Zoutendijk,Nonlinear programming computational methods, in : J. Abadie (Ed.), Integer and Nonlinear Programming, North-Holland, Amsterdam, 1970, pp. 37ñ86

[14] :Mickaél S Mémoire de mathématique p :9-11, Université Paris Sud

[15] L. Grippo,S. Lucidi, A globally convergent version of the PolakÒRibieÎre gradient method, Math. Prog.78 (1997) 375Ò391

[16] Wei et al., The convergence properties of some conjugate gradient methods, Appl. Math.Comput. 183 (2006) 1341Ò1350.

[17] The proof of the suc/ cient descent condition of the WeiÒYaoÒLiu conjugate gradient method under the strongWolfe-Powell line search, Appl. Math. Comput. doi :doi :10.1016/j.amc.2006.12.006, 2007

[18] Lin Suihua, Yao Shengwei, A new conjugate gradient method of modified LS formula,$Department of Mathematics and Information Science, Guangxi University, 2006.

[19] Shengwei, Lin Suihua, A new conjugate gradient method combined HS and DY formulas, Department of Mathematics and Information Science, Guangxi University, 2