DESCENTE SUFFISANTE ET CONVERGENCE GLOBALE DE LA METHODE DU GRADIENT CONJUGUE VERSION NICULAI ANDREI

BERKANE ali

16 mai 2014

Table des matières

Résumé

On propose dans ce mémoire une modification de la méthode du gradient conjugué version Dai-Yuan. Cette modification a été étudiée en 2008 par Niculai Andrei. Dans ce travail l'auteur réussit à prouver que la nouvelle méthode génère des directions de descente suffisante et que l'algorithme converge globalement.

Mots clés : Gradient conjugué, Recherche linéaire inexacte de Wolfe, descente suffisante , Méthode de Dai-Yuan, Méthode de Niculai Andrei, Convergence globale.

Abstract

A modification of the Dai-Yuan conjugate gradient algorithm is proposed. For inexact line search the algorithm satifies both sufficient descent and conjugacy conditions. A global convergence result is proved when the wolfe line search conditions are used.

Key words : Dai-Yuan conjugate gradient algorithm. Niculai Andrei conjugate gradient algorithm. Wolfe line search. Global convergence.

INTRODUCTION

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 ([28, 1952]), pour la minimisation de fonctions quadratiques 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 Reeves ([19, 1964]) (*méthode de Fletcher-Reeves*) puis en 1969 par Polak, Ribière ([38, 1969]) et Ployak ([39, 1969]) (*méthode de Polak-Ribière-Ployak*). Une autre variante a été étudiée en 1987 par Fletcher ([17, 1987]) (*Méthode de la descente conjuguée*).

Les méthodes du gradient conjugué engendrent une suite d'itérés $\{x_k\}_{k\in\mathbb{N}}$ comme suit :

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

Où α_k est le pas optimal déterminé par une recherche linéaire inexacte de Wolfe faible, c'est à dire que les pas α_k vériient les deux conditions suivantes :

$$\begin{array}{rcl} f(x_k + \alpha_k d_k) & \leq & f(x_k) + \delta \alpha_k d_k^T g_k d_k^T g_{k+1} \qquad \geq \\ \sigma d_k^T g_k & & \end{array}$$

avec

$$0 < \delta < \sigma < 1$$

Ces méthodes considèrent d_k comme suit :

$$d_k = \begin{cases} -g_1 & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases} \tag{0.4}$$

Où $g_k = \nabla f(x_k)$ et β_k est le paramétre qui caractérise les différentes versions des méthodes du gradient conjugué.

Si on note $y_k = g_{k+1} - g_k$, on obtient les variantes suivantes :

$$\beta_k^{PRP} = \frac{g_k^T y_k}{\|g_{k-1}\|^2} \quad \text{Gradient conjugué variante } \textit{Polak-Ribière-Polyak} \tag{0.5}$$

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \quad \text{Gradient conjugué variante } \textit{Fletcher Reeves} \tag{0.6}$$

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}} \quad \text{Gradient conjugué variante } \textit{descente conjugué} \tag{0.7}$$

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} \quad \text{Gradient conjugué variante de } \textit{Dai-Yuan} \tag{0.8}$$

On étudie dans ce mémoire le problème de la convergence globale de la méthode du gradient conjugué i.e. étudier la limite inf suivante

$$\lim_{k\to\infty} \inf \|g_k\| = 0 \tag{0.9}$$

dans le cas d'une recherche linéaire inexacte de Wolfe et $\beta_k = \beta_k^{DY}$ ou $\beta_k = \beta_k^a$, avec β_k^{DY} (coéfficient de Dai-Yuan) défini par la relation (0.7) et β_k^a (Coéfficient de Niculai Andrei), qu'on définira par la suite. Le même problème a été étudié par Al Baali ([*Al Baali*]) qui a démontré la relation (1.9), dans le cas d'une recherche linéaire inexacte de Wolfe forte, c'est à dire que les pas α_k vérifient les deux conditions suivantes :

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \delta \alpha_k d_k^T g_k \\ \left| d_k^T g_{k+1} \right| &\leq -\sigma d_k^T g_k \end{aligned}$$

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

D'autres auteurs ont démontré qu'on ne peut pas obtenir la relation (0.9), avec des recherches linéaires inexactes de Wolfe faibles (voir [Y.H. Dai et Y. Yuan : Convergence properties of the Fletcher Reeves method, 1996]). Le problème majeur rencontré par l'utilisation des recherches linéaires inexactes de Wolfe faibles est le fait que les directions qu'elles engendrent ne sont pas des directions de descente. Dai et Yuan ([Dai-Yuan 1999]) ont modifié les coéfficients β_k^{FR}, de sorte que les nouvelles directions obtenues avec des recherches linéaires inexactes de Wolfe faibles, soient des directions de descente.

On obtient ainsi une nouvelle famille de gradients conjuguées pour laquelle la relation (0.9) est vraie avec des recherches linéaires inexactes de Wolfe faibles.

La propriété fondamentale de cette méthode est qu'elle génère des directions de descente en utilisant seulement la recherche de wolfe alors que les autres méthodes du gradient conjugué classiques (Fletcher Reeves et d'autres) utilisent la recherche de wolfe forte.

Les directions générées par la méthode du gradient conjugué version Dai-Yuan sont des directions de descente mais ne sont pas des directions de descente suffisante, c'est à dire qu'elles ne vérifient pas la relation suivante :

$$g_k^T d_k \leq -c \|g_k\|^2, \quad \forall k, \quad c > 0 \tag{0.13}$$

En 2008, Neculai Andrei ([A Dai-Yuan conjugate...]) proposa une modification de la méthode de Dai-Yuan et démontre que les directions engendrées sont des directions de descente suffisante.

Niculai Andrei ([A Dai-Yuan conjugate...]) proposa la suite $\{d_k\}_{k\in\mathbb{N}}$ comme suit : $d_0 = -g_0$ et

$$d_{k+1} = -\theta_{k+1}.g_{k+1} + \beta_k^a s_k, \quad k = 0, 1, 2, ... \tag{0.14}$$

avec

$$\theta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{y_k^T \cdot g_{k+1}} \tag{0.15}$$

et

$$\beta_k^a = \frac{1}{y_k^T s_k} \left(g_{k+1} - \delta_k \frac{\|g_{k+1}\|^2}{y_k^T s_k} s_k \right)^T g_{k+1} \tag{0.16}$$

et

$$\delta_k = \frac{y_k^T \cdot g_{k+1}}{g_{k+1}^T g_{k+1}} \tag{0.17}$$

La suprémacie de la méthode de Niculai Andrei est qu'elle assure que les directions d_k sont des directions de descente suffisante, alors que les directions engendrées par Dai et Yuan sont des directions de descente seulement. Les tests numériques ont prouvé que la méthode de Niculai Andrei est plus performante que celle de Dai et Yuan. La méthode de Niculai Andrei converge globlement avec la recherche linéaire inexacte de Wolfe.

Le mémoire comporte cinq chapitres :

On introduit dans le premier chapitre les notions préliminaires d'optimisation sans contraintes.

Le chapitre deux est consacré aux méthodes à direction de descente et les recherches linéaire, on insiste surtout sur les recherches linéaires inexactes dites d'Armijo, Goldstein & Price et de Wolfe.

Le troisième chapitre traite de façon générale les méthodes du gradient conjugué.

Dans le chapitre quatre on étudie la méthode du gradient conjugué version Dai-Yuan. On montre la descente des directions d_k, avec des recherches linéaires inexactes de Wolfe faibles.

Le chapitre5 est consacré à l'étude la méthode du gradient conjugué version Niculai Andrei qui est une modification de la méthode du gradient conjugué version Dai-Yuan. Cette modification assure la descente suffisante (notion plus foret que la simple descente) des directions d_k.

On trouve à la fin du mémoire des tests numériques, une conclusion et des problèmes ouverts.

Chapitre 1

Quelques généralités sur les problèmes d'optimisation sans contraintes

Dans cette partie, nous donnons quelques généralités sur les problèmes de minimisations sans contraintes.

1.1 Définitions

Définition 1.1
- On note par

$$(\nabla^T f(x)) = \frac{\partial f}{\partial x}(x) = \left(\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n}\right)(x) \tag{1.1}$$

le gradient de f au point $x = (x_1, .., x_n)$.
- On appelle Hessien de f la matrice symétrique de d'ordre $n \times n$

$$H(x) = \nabla(\nabla^T f)(x) = \nabla^2 f(x) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)(x), \; i = 1, ..., n, \; j = 1, ..., n \tag{1.2}$$

Définition 1.2
• La matrice H est dite semi-définie positive si

$$\forall x \in \mathbb{R}^n : x^\top H x \geq 0 \tag{1.3}$$

• H est dite définie positive si

$$\forall x \in \mathbb{R}^n \ , \ x \neq 0 : x^\top Hx > 0 \tag{1.4}$$

Définitions 1.3

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

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

• On dit que $\widehat{x}$ est un minimum local du problème (P) si et seulement si :

$$\exists V_\varepsilon(\widehat{x}) \text{ tel que } f(\widehat{x}) \leq f(x), \forall x \in V_\varepsilon(\widehat{x}) \ \text{ et } \varepsilon > 0 \tag{1.5}$$

• On dit que $\widehat{x}$ est un minimum global du problème (P) si et seulement si :

$$f(\widehat{x}) \leq f(x), \forall x \in \ \ \mathbb{R}^n \tag{1.6}$$

• On dit que $\widehat{x}$ est un minimum local strict du problème (P) si et seulement si :

$$\exists V_\varepsilon(\widehat{x}) \ \text{ tel que } f(\widehat{x}) < f(x), \forall x \in V_\varepsilon(\widehat{x}) et\ x \neq \ \widehat{x} \tag{1.7}$$

La figure (1.1) illustre, sur une fonction à une seule variable, les notions d'optimum global et d'optimum localftbpFU4.7046in2.6636in0pt**Point stationnaire, optimum local, optimum global**Figure

1.2 Direction de descente

Définition 1.4
Soit $f : \mathbb{R}^n \to \mathbb{R}$, $\widehat{x} \in \mathbb{R}^n$, d un vecteur non nul de $\mathbb{R}^n$ est dit direction de descente au point $\widehat{x}$ si et seulement s'il existe un réel $\delta > 0$ tel que $\forall \lambda \in]0, \delta[$ on a :

$$f(\widehat{x} + \lambda d) < f(\widehat{x}) \tag{1.8}$$

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

Théorème 1.1
Soit $f : \mathbb{R}^n \to \mathbb{R}$, différentiable au point $\widehat{x} \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$ telle que :

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

Alors d est une direction de descente en $\widehat{x}$.

1.3 Conditions d'optimalité des problèmes d'optimisation sans contraintes

1.3.1 Condition nécessaire d'optimalité d'ordre 1

Théorème 1.2. *Soit $f : \mathbb{R}^n \to \mathbb{R}$ différentiable en $\widehat{x} \in \mathbb{R}^n$, si $\widehat{x}$ est un minimum local de f alors*

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

Démonstration. [3,1966]
Supposons le contraire $\nabla f(\widehat{x}) \neq 0$. Si on pose $d = -\nabla f(\widehat{x})$, on obtient :

$$\nabla^T f(\widehat{x})\ .d = -\left\| \nabla f(\widehat{x}) \right\|^2 < 0$$

Et d'après le théorème (1.1), $\exists\ \delta > 0$ tel que :

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

Mais ceci est contradictoire avec le fait que $\widehat{x}$ soit un minimum local
D'où $\nabla f(\widehat{x}) = 0.\square$

1.3.2 Condition nécessaire d'optimalité d'ordre 2

Théorème 1.3. *Soit $f : \mathbb{R}^n \to \mathbb{R}$ deux fois différentiable au point $\widehat{x} \in \mathbb{R}^n$, si $\widehat{x}$ est un minimum local de f alors, $\nabla f(\widehat{x}) = 0$ et $H(\widehat{x})$ est semi-définie positive.*

Démonstration. [4,1993]
La première proposition est déjà montrer, pour la deuxième proposition on a :
Comme f est deux fois différentiable en $\widehat{x}$ alors

$$\begin{aligned} f(\widehat{x}+\lambda d) &= f(\widehat{x}) + \lambda\ \nabla^T f(\widehat{x}).d + \frac{1}{2}\lambda^2 d^T H(\widehat{x})d + \lambda^2 \|d\|^2 \alpha(\widehat{x},\lambda d) \\ \textit{Où}\ \ \alpha(\widehat{x},\lambda d) &\to 0 \to \textit{pour}\ \ \lambda \to 0 \end{aligned}$$

Ceci implique :

$$\frac{f(\widehat{x}+\lambda d)\ -\ f(\widehat{x})}{\lambda^2} = \frac{1}{2}d^T H(\widehat{x})d + \|d\|^2 \alpha(\widehat{x},\lambda d)\ , \lambda \neq 0$$

Comme $\widehat{x}$ est un minimum local alors
$f(\widehat{x}+\lambda d) \geq f(\widehat{x})$ pour λ suffisamment petit
D'où

$$\frac{1}{2}d^t H(\widehat{x})d + \|d\|^2 \alpha(\widehat{x},\lambda d) \geq 0 \ \textit{pour}\ \lambda\ \textit{petit}$$

En passant à la limite quand $\lambda \to 0$, on obtient que

$$d^t H(\widehat{x})d \geq 0$$

D'où $H(\widehat{x})$ est semi définie positive.$\square$

1.3.3 Condition suffisante d'optimalité

$\diamond$ **Cas où f est non convexe**

Théorème 1.4. *Soit $f : \mathbb{R}^n \to \mathbb{R}$ deux fois différentiable 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 f

Démonstration. [4,1993]

f est deux fois différentiable en $\widehat{x}$, alors $\forall x \in \mathbb{R}^n$, on obtient :

$$f(x) = f(\widehat{x}) + \nabla^T f(\widehat{x})(x - \widehat{x}) + \frac{1}{2}(x - \widehat{x})^T H(\widehat{x})(x - \widehat{x}) + \|x - \widehat{x}\|^2 \alpha(\widehat{x}, x - \widehat{x})$$

Où $\alpha(\widehat{x}, x - \widehat{x}) \to 0$ pour $x \to \widehat{x}$

Supposons que $\widehat{x}$ n'est pas un minimum local strict.

Donc il existe une suite $\{x_k\}_{k\in N}$ convergente vers $\widehat{x}$ telle que

$$f(x_k) \leq f(\widehat{x}), x_k \neq \widehat{x} \quad \forall k$$

Posons $d_k = \frac{(x_k - \widehat{x})}{\|x_k - \widehat{x}\|^2}$, donc $\|d_k\| = 1$ ceci implique :

$$\frac{f(x_k) - f(\widehat{x})}{\|x_k - \widehat{x}\|^2} = \frac{1}{2} d_k^t H(\widehat{x}) d_k + \alpha(\widehat{x}, x_k - \widehat{x}) \leq 0 \qquad \forall k \qquad (*)$$

et comme $\|d_k\| = 1$,$\forall k$ alors $\exists \{d_k\}_{k\in N_1 \subset N}$ telle que $d_k \to d$ pour $k \to \infty$ et $k \in N_1$ donc on a $\|d\| = 1$ .Considérons donc $\{d_k\}_{k\in N_1}$et le fait $\alpha(\widehat{x}, x - \widehat{x}) \to 0$ pour $k \to \infty$ et $k \in N_1$

Alors (*) donne $d^t H(\widehat{x}) d \leq 0$, ce qui contredit le fait que $H(\widehat{x})$ est définie positive car $\|d\| = 1$ $(d \neq 0)$

Donc $\widehat{x}$ est un minimum local strict.$\square$

$\diamond$ **Cas où f est convexe**

Définition 1.5

Soit $f : \mathbb{R}^n \to \mathbb{R}$

a) f est dite convexe dans $\mathbb{R}^n$ si : $\forall x_1, x_2 \in \mathbb{R}^n$ et pour tout $t \in [0, 1]$

$$f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2)$$

b) si l'inégalité précédente est stricte pour tout points x_1 et x_2 distincts et $\forall t \in]0,1[$ alors f est dite strictement convexe.

Remarque 1.1

Si f est convexe, alors tout minimum local est aussi global. De plus si f est strictement convexe, alors tout minimum local devient minimum global unique.

Théorème 1.5. *Soit $f : \mathbb{R}^n \to \mathbb{R}$ convexe et différentiable au point $\widehat{x} \in \mathbb{R}^n$ si $\nabla f(\widehat{x}) = 0$ alors, $\widehat{x}$ est un minimum global de f.*

Démonstration.

f étant convexe on a :

$$f(tx + (1-t)\widehat{x}) \leq tf(x) + (1-t)f(\widehat{x})$$

Ceci implique

$$f(tx + \widehat{x} - t\widehat{x}) \leq tf(x) + (1-t)f(\widehat{x})$$

$$f(\widehat{x} + t(x - \widehat{x})) \leq tf(x) + (1-t)f(\widehat{x})$$

Où encore :

$$f(\widehat{x} + t(x - \widehat{x})) \leq tf(x) + f(\widehat{x}) - tf(\widehat{x})$$

$$f(\widehat{x} + t(x - \widehat{x})) \leq f(\widehat{x}) + t(f(x) - f(\widehat{x}))$$

$$f(\widehat{x} + t(x - \widehat{x})) - f(\widehat{x}) \leq t(f(x) - f(\widehat{x}))$$

Divisons par t on obtient :

$$\frac{f(\widehat{x} + t(x - \widehat{x})) - f(\widehat{x})}{t} \leq f(x) - f(\widehat{x})$$

Et comme f est différentiable au point $\widehat{x}$ on a :

$$f(\widehat{x}+td) = f(\widehat{x})+t\nabla^T f(\widehat{x})d+t\,\|d\|\,\alpha(\widehat{x},td) \quad avec\ \alpha(\widehat{x},td) \to 0\ quand\ t \to 0$$

Donc on obtient :

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

Si on pose $d = x - \widehat{x}$ on obtient :

$$\nabla f(\widehat{x})(x-\widehat{x}) \leq f(x) - f(\widehat{x})$$

Et comme $\nabla f(\widehat{x}) = 0$ on a donc $0 < f(x) - f(\widehat{x}) \Rightarrow f(x) \leq f(\widehat{x})$
D'où $\widehat{x}$ est minimum global. $\square$

Chapitre 2

Méthodes à directions de descente et Recherches linéaires

2.1 Principe général des méthodes à direction de descente

Considérons le problème d'optimisation sans contrainte

$$(P) \qquad \begin{cases} \min\limits_{x\in\mathbb{R}^n} f(x) \end{cases}$$

Où $f : \mathbb{R}^n \to \mathbb{R}$ est supposée régulière

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

$$x_{k+1} = x_k + \lambda_k d_k \quad \text{où } \lambda_k > 0 \tag{2.1}$$

Tout en assurant la propriété $f(x_{k+1}) < f(x_k)$

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

2.1.1 Direction de descente

Définition 2.1

Soit $f : \mathbb{R}^n \to \mathbb{R}$, $\widehat{x} \in \mathbb{R}^n$, d un vecteur non nul de $\mathbb{R}^n$ est dit direction de descente au point $\widehat{x}$ si et seulement s'il existe un réel $\delta > 0$

tel que $\forall \lambda \in]0, \delta[$ on a :

$$f(\widehat{x} + \lambda d) < f(\widehat{x}) \tag{2.2}$$

Théorème 2.1
Soit $f : \mathbb{R}^n \to \mathbb{R}$ *, différentiable au point* $\widehat{x} \in \mathbb{R}^n$ *et* $d \in \mathbb{R}^n$ *telle que :*

$$\nabla^T f(\widehat{x})\ d < 0 \tag{2.3}$$

Alors d *est une direction de descente en* $\widehat{x}$*.*
Démonstration. [1,1985]
Comme f est différentiable en $\widehat{x}$ alors

$f(\widehat{x}+\lambda d) = f(\widehat{x}) + \lambda \nabla^T f(\widehat{x}).d + \lambda \|d\|\ \alpha(\widehat{x}, \lambda d)$ Où $\alpha(\widehat{x}, \lambda d) \to 0$ pour $\lambda \to 0$

Ceci implique :

$$\frac{f(\widehat{x} + \lambda d)\ -\ f(\widehat{x})}{\lambda} = \nabla^T f(\widehat{x}).d\ + \|d\|\ \alpha(\widehat{x}, \lambda d)\ , \lambda \neq 0$$

Et comme $\nabla^T f(\widehat{x})\ d < 0$ et $\alpha(\widehat{x}, \lambda d) \to 0$ pour $\lambda \to 0$, $\exists\ \delta > 0$ tel que
Body Math

$$\nabla^T f(\widehat{x})\ .d\ < 0 \qquad \forall \lambda \in]0, \delta[$$

Et par conséquent on obtient :

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

Donc d est une direction de descente. □
Remarque 2.1
d fait avec l'opposé du gradient $-\nabla f(x)$ un angle θ strictement plus petit que 90° :

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

L'ensemble des directions de descente de f en x

$$\left\{d \in \mathbb{R}^n : \nabla^T f(x).d < 0\right\}$$

forme un demi-espace ouvert de $\mathbb{R}^n$ (illustration à la figure 2.1)

ftbpFU4.9009in2.041in0pt**Demi-espace (translaté) des directions de descente** d **de** f **en** xFigure

De telles directions sont intéressantes en optimisation car, pour faire décroitre f il suffit de faire un déplacement le long de d.

2.1.2 Description des méthodes de descente

Les méthodes à direction de descente utilisent l'idée suivante pour minimiser une fonction.

Elles construisent la suite des itérés $\{x_k\}_{k\geq 1}$, approchant une solution x_k de (P) par la récurrence :

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

où λ_k est appelé le pas et d_k la direction de descente de f en x_k.

Pour définir une méthode à 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 λ_k, c'est ce que l'on appelle " la recherche linéaire "

D'écrivons cette classe d'algorithme de manière précise.

Algorithme 2.1 (méthode à direction de descente-une itération)

Etape 0 : initialisation

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

Etape 1 :

Teste d'arrêt : si $\|\nabla f(x_k)\| \approx 0$, arrêt de l'algorithme .

Etape 2 :

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

Etape 3 :

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

Etape 4 :

Si la recherche linéaire réussit : $x_{k+1} = x_k + \lambda_k d_k$

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

ftbpFU5.1214in2.7121in0pt**Méthode à directions de descente**Figure

2.1.3 Exemples des méthodes à directions de descente

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 :

$$x_{k+1} = x_k + \lambda_k d_k, \quad \lambda_k \in \left]0, +\delta\right[. \tag{2.4}$$

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

$$f\left(x_{k+1}\right) = f\left(x_k + \lambda_k d_k\right) < f\left(x_k\right) \tag{2.5}$$

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

Exemples de chois de directions 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.

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

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

La méthode de Newton correspond à $d_k = -\left(H\left(x_k\right)\right)^{-1}.\nabla f(x_k)$.

d_k direction de descente si la matrice hessienne $H(x_k)$ est définie positive

Exemples de chois de pas λ_k
On choisit en général λ_k de façon optimale, c'est-à-dire que λ_k doit vérifier

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

En d'autres 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.

2.2 Recherche linéaire

Faire de la recherche linéaire veut dire déterminer un pas λ_k le long d'une direction de descente d_k, autrement dit résoudre le problème unidimensionnel :

$$\min f(x_k + \lambda d_k), \quad \lambda \in \mathbb{R}_+ \tag{2.6}$$

Notre intérêt pour la recherche linéaire ne vient pas seulement du fait que dans les applications on rencontre, naturellement, des problèmes unidimensionnels, mais plutôt du fait que la recherche linéaire est un composant fondamental de toutes les méthodes traditionnelles d'optimisation multidimensionnelle.

2.2.1 Objectifs à atteindre

Il s'agit de réaliser deux objectifs
Le premier objectif
Consiste à *faire décroître f suffisamment*. Cela se traduit le plus souvent par la réalisation d'une inégalité de la forme

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \text{"un terme négatif "} \tag{2.7}$$

Le terme négatif, disons ν_k, joue un rôle-clé dans la convergence de l'algorithme utilisant cette recherche linéaire.
L'argument est le suivant.
Si $f(x_k)$ est minorée (il existe une constante c telle que $f(x_k) \geq c \quad \forall\ k$)
alors ce terme négatif tend nécessairement vers zéro ($\nu_k \to 0$) C'est souvent à partir de la convergence vers zéro de cette suite que l'on parvient

à montrer que le gradient lui-même doit tendre vers zéro. Le terme négatif devra prendre une forme bien particulière si on veut pouvoir en tirer de l'information.

En particulier, il ne suffit pas d'imposer $f(x_k + \lambda_k d_k) < f(x_k)$.

Le second objectif

Consiste d'empêcher le pas $\lambda_k > 0$ d'être trop petit, trop proche de zéro.

Le premier objectif n'est en effet pas suffisant car l'inégalité (2.7) et en générale satisfaite par des pas $\lambda_k > 0$ arbitrairement petit.

Or ceci peut entrainer une "*fausse convergence* ", c'est-à-dire la convergence des itérés vers un point non stationnaire.

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

2.2.2 Recherche linéaire exacte

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 λ_k comme solution du problème

$$\{\min \varphi_k(\lambda) \ , \quad \lambda \geq 0 \tag{2.8}$$

C'est ce que l'on appelle la règle de Cauchy et le pas déterminé par cette règle est appelé pas de Cauchy ou pas optimal

(voir figure 2.3). Dans certains cas, on préférera le plus petit point stationnaire de φ_k qui fait décroître cette fonction :

$$\lambda_k = \inf \left\{ \lambda \geq 0 : \varphi_k' \left(\lambda \right) < \varphi_k \left(0 \right) \right\} . \tag{2.9}$$

On parle alors de règle de Curry et le pas déterminé par cette règle est appelé pas de Curry (voir figure 2.3). De manière un peu imprécise, ces deux règles sont parfois qualifiées de recherche linéaire exacte.

ftbpFU4.0075in1.9649in0pt**Règles de Cauchy et de Curry**Figure

Remarque 2.2. Ces deux règles ne sont utilisées que dans des cas particuliers, par exemple lorsque φ_k est quadratique.

Le mot exact prend sa signification dans le fait que si f est quadratique la solution de la recherche linéaire s'obtient de façon exacte et dans un nombre fini d'itérations.

Les inconvénients des recherches linéaires exactes Pour une fonction non linéaire arbitraire,

- il peut ne pas exister de pas de Cauchy ou de Curry,

- la détermination de ces pas demande en général beaucoup de temps de calcul et ne peut de toute façons pas être faite avec une précision infinie,

- l'efficacité supplémentaire éventuellement apportée à un algorithme par une recherche linéaire exacte ne permet pas, en général, de compenser le temps perdu à déterminer un tel pas,

- les résultats de convergence autorisent d'autres types de règles (recherche linéaire inexacte), moins gourmandes en temps de calcul.

2.2.3 Recherche linéaire inexacte

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 la recherche $d_k \in \mathbb{R}^n$ qui est direction de descente pour notre objectif $f : \mathbb{R}^n \to \mathbb{R}$ où

$$\nabla^T f(x).d < 0$$

Le but est de réduire ≪*de façon importante*≫ la valeur de l'objectif par un pas $x_k \to x_{k+1} = x_k + \lambda_k d_k$ de x_k dans la direction d_k

pour cela de nombreux mathématiciens (Armijo , Goldstein ,Wolfe , Albaali, Lamaréchal, Fletcher,...) on élaboré plusieurs régles.

L'objectif de cette section consiste à présenter les principaux testes.

2.2.4 La règle d'Armijo [1966]

Soit $f : \mathbb{R}^n \to \mathbb{R},\ \ x \in \mathbb{R}^n, d \in \mathbb{R}^n$ tel que $\nabla^T f(\widehat{x}).d < 0$

la condition d'Armijo exige à ce que f décroisse de façon suffisante au point $x_k + \lambda_k d_k$

Cette condition est décrite par l'inégalité suivante dite condition d'Armijo

$$f(x_k + \lambda_k d_k) \leq f(x_k) + c\lambda_k \nabla^T f(x_k).d_k \qquad c \in]0,1[\quad (2.10)$$

Définissons la fonction $\varphi : \mathbb{R}_+ \to \mathbb{R}$ avec $\varphi_k(\lambda) = f(x_k + \lambda d_k)$,$\lambda_k \geq 0$

Notons que :

$$\varphi_k^{'}(\lambda) = \nabla^T f(x_k + \lambda d_k).d_k$$

$$\varphi_k^{'}(0) = \nabla^T f(x_k).d_k \; < 0$$

$$\varphi_k(0) = f(x_k)$$

L'équation de la tangente au point $(0, \varphi(0))$ est la suivante :

$$\{\lambda, y\} : y = \varphi(0) + \varphi^{'}(0)(\lambda - 0)$$

$$\widetilde{\varphi}_k(\lambda) = \varphi_k(0) + \varphi_k^{'}(0)\,\lambda \;\; , \;\; \lambda \geq 0$$

L'équation de la tangente devient

$$\widetilde{\varphi}_k(\lambda) = f(x_k) + \lambda \nabla^T f(x_k).d_k$$

Définissons maintenant la fonction $\widehat{\varphi}_k(\lambda)$ comme suit :

$$\begin{aligned} \widehat{\varphi}_k(\lambda) &= f(x_k) + c\lambda \nabla^T f(x_k).d_k 2.11 && (2.1) \\ &= \varphi_k(0) + c\lambda \varphi_k^{'}(0) \qquad c \in]0,1[&& \\ \text{On voit que} \quad &: \varphi_k(\lambda) \leq \widehat{\varphi}_k(\lambda) 2.12 && (2.2) \end{aligned}$$

Remarque 2.3

- La condition (2.12) implique la décroissance de la fonction f c'est-à-dire

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

- $\widehat{\varphi}_k(\lambda)$ est appelle la ligne de décroissance suffisante .

- En général, on s'assure dans la règle d'Armijo que λ_k ne soit pas trop petit car cela va nuire à la convergence de l'algorithme qui pourrait converger prématurément vers un point qui n'est pas stationnaire .

On voit bien à la figure (2.4) ce qui signifie la condition (2.12)

ftbpFU4.8767in2.7605in0pt**Règle d'Armijo**Figure

Teste d'Armijo

◊ Si

$$\varphi_k(\lambda) \leq \varphi_k(0) + c\lambda\varphi_k^{'}(0)$$

$$\text{Autrement dit} \qquad f(x_k + \lambda d_k) \leq f(x_k) + c\lambda\nabla^T f(x_k).d_k$$

Alors λ convient.

◊ Si

$$\varphi_k(\lambda) > \varphi_k(0) + c\lambda\varphi_k^{'}(0)$$

Autrement dit $\qquad f(x_k + \lambda d_k) > f(x_k) + c\lambda\nabla^T f(x_k).d_k$

Alors λ est trop grand. $\square$

$\boxed{\textbf{Algorithme 2.2 (la règle d'Armijo)}}$

Etape initiale : choisir un point initial λ_1, et $c \in]0,1[$

Etapes principales :

Etape 1 :

Si $\varphi(\lambda_1) \leq \varphi(0) + c\lambda_1\varphi'(0)$, aller à l'étape 2
Sinon aller à l'étape 3

Etape 2 :

Soit t le plus grand entier positif avec $\lambda = \lambda_1 2^t$ qui vérifie
$\varphi(\lambda_1 2^t) < \varphi(0) + c\lambda_1 2^t\varphi'(0)$

Etape 3 :

Soit t le plus petit entier positif avec $\lambda = \frac{\lambda_1}{2^t}$ qui vérifie
$\varphi(\frac{\lambda_1}{2^t}) < \varphi(0) + c(\frac{\lambda_1}{2^t})\varphi'(0) \quad \square$

Théorème 2.2

Si $\varphi : \mathbb{R}_+ \to \mathbb{R}$ 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 $c \in]0,1[$ alors l'ensemble des pas vérifiant la régle d'Armijo est non vide.

Démonstration. [10,1996]

On a

$$\begin{aligned} \varphi_k(\lambda) &= f(x_k + \lambda d_k) \\ \psi_c(\lambda) &= f(x_k) + c\lambda\nabla^T f(x_k).d_k \end{aligned}$$

Le développement de Taylor-Yong en $\lambda = 0$ de φ_k est :

$$\varphi_k(\lambda) = f(x_k + \lambda d_k) = f(x_k) + \lambda\nabla^T f(x_k).d_k + \lambda\xi(\lambda) \text{ où } \xi(\lambda) \to 0, \lambda \to 0.$$

et comme $c \in]0,1[$ et $\varphi_k'(0) = \nabla^T f(x_k).d_k < 0$ on déduit :

$$f(x_k) + \lambda \nabla^T f(x_k).d_k < f(x_k) + c\lambda \nabla^T f(x_k).d_k \text{ pour } \lambda > 0$$

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

$$\varphi_k(\lambda) < \psi_c(\lambda)$$

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

$$\begin{cases} \psi_c(\lambda) - \varphi_k(\lambda) > 0 & \text{pour } \lambda \text{ assez petit} \\ \psi_c(\lambda) - \varphi_k(\lambda) < 0 & \text{pour } \lambda \text{ assez grand} \end{cases}$$

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

$$\varphi_k(\overline{\lambda}) = \psi_c(\overline{\lambda}) \text{ et } \varphi_k(\lambda) < \psi_c(\lambda) \text{ pour } 0 < \lambda < \overline{\lambda}$$

Ce qui achève la démonstration. $\square$
Décrivons maintenant la règle de Goldstein&Price

2.2.5 La règle de Goldstein&Price [1969]

La régle de Goldstein s'applique lorsque le gradient de la fonction ne peut être évalué (ou est trop couteux à obtenir)

En ajoutant une deuxième inégalité à la règle d'Armijo on obtient la règle de Goldstein&Price avec $c_{1,}c_2$ sont deux constants vérifiant $0 < c_1 < c_2 < 1$. Les deux inégalités de la règle de Goldstein&Price sont donc :

$$\begin{array}{lrcll} & f(x_k + \lambda_k\, d_k) & \leq & f(x_k) + c_1\lambda_k \nabla^T f(x_k).d_k & c_1 \in]0,1[\; 2.13 \quad (2.3) \\ \text{Et} & f(x_k + \lambda_k d_k) & \geq & f(x_k) + c_2\lambda_k \nabla^T f(x_k).d_k & c_2 \in]c_1,1[\; 2.14 \quad (2.4) \end{array}$$

Autrement dit :

$$\begin{array}{rcl} \varphi_k(\lambda) & \leq & \widehat{\varphi}_k(\lambda) 2.15 \quad (2.5) \\ \text{Et } \varphi_k(\lambda) & \geq & \varphi_k(0) + c_2\varphi_k^{'}(0)\, 2.16 \quad (2.6) \end{array}$$

On voit bien à la figure (2.5) ce que signifie les conditions (2.15) et (2.16)ftbpFU5.4457in2.9663in0pt**Règle de Goldstein&Price**Figure

Teste de Goldstein&Price

$\diamondsuit$Si

$$\begin{aligned} \varphi_k(0) + c_2\lambda\varphi_k'(0) &\leq \varphi_k(\lambda) \leq \varphi_k(0) + c_1\lambda\varphi_k'(0) \\ \text{autrement dit } f(x_k) + c_2\lambda\nabla^T f(x_k).d_k &\leq f(x_k + \lambda d_k) \leq f(x_k) + c_1\lambda\nabla^T f(x_k).d_k \end{aligned}$$

Alors λ convient.

$\diamondsuit$ Si

$$\begin{aligned} \varphi_k(\lambda) &> \varphi_k(0) + c_1\lambda\varphi_k'(0) \\ \text{autrement dit } f(x_k + \lambda d_k) &> f(x_k) + c_1\lambda\nabla^T f(x_k).d_k \end{aligned}$$

Alors λ est trop grand.

$\diamondsuit$ Si

$$\begin{aligned} \varphi_k(\lambda) &< \varphi_k(0) + c_2\lambda\varphi_k'(0) \\ \text{autrement dit } f(x_k + \lambda d_k) &< f(x_k) + c_2\lambda\nabla^T f(x_k).d_k \end{aligned}$$

Alors λ est trop petit. $\square$

Algorithme 2.3 (la règle de Goldstein&Price)

Etape initiale :

On dispose de $\lambda_g = 0, \lambda_d =$ une valeur maximale quelconque, et soit $\lambda \in]\lambda_g, \lambda_d[$

poser $c_1 = 0,1$, $c_2 = 0,7$

Etapes principales :

Etape 1 : calculer $\varphi(\lambda) = f(x_k + \lambda d_k)$

Si $\varphi(\lambda) \leq \varphi(0) + c_1\lambda\varphi'(0)$ alors aller à l'étape 2

Sinon prendre $\lambda_d = \lambda$ et aller à l'étape 4

Etape 2 :

Si $\varphi(\lambda) \geq \varphi(0) + c_2\lambda\varphi'(0)$ stop

Sinon aller à l'étape 3

Etape 3 : poser $\lambda_g = \lambda$

Etape 4 : rechercher un nouveau $\lambda \in]\lambda_g, \lambda_d[$ et retourner à l'étape2 .

□

Théorème 2.3

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 $c_1 \in]0,1[\,, c_2 \in]c_1,1[\,,$ alors l'ensemble des pas vérifiant la règle de Goldstein&Price est non vide.

Démonstration. [10,1996]

On a

$$\begin{aligned}\varphi_k(\lambda) &= f(x_k + \lambda d_k)\\ \psi_{c_1}(\lambda) &= f(x_k) + c_1\lambda\nabla^T f(x_k).d_k\\ \psi_{c_2}(\lambda) &= f(x_k) + c_2\lambda\nabla^T f(x_k).d_k\end{aligned}$$

Le développement de Taylor-Yong en $\lambda = 0$ de φ_k est :

$$\varphi_k(\lambda) = f(x_k + \lambda d_k) = f(x_k) + \lambda\nabla^T f(x_k).d_k + \lambda\xi(\lambda) \text{ où } \xi(\lambda) \to 0, \lambda \to 0.$$

et comme $c_1 \in]0,1[$ et $\varphi_k'(0) = \nabla^T f(x_k) d_k < 0$ on déduit :

$$f(x_k)+\lambda\nabla^T f(x_k).d_k < f(x_k)+c_2\lambda\nabla^T f(x_k).d_k < f(x_k)+c_1\lambda\nabla^T f(x_k).d_k \text{ pour } \lambda > 0$$

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

$$\varphi_k(\lambda) < \psi_{c_2}(\lambda) < \psi_{c_1}(\lambda)$$

De ce qui précède et du fait que φ_k est bornée inférieurement,
et $\psi_{c_1}(\lambda) \to -\infty ; \lambda \to +\infty$, on déduit que la fonction $\psi_{c_1}(\lambda) - \varphi_k(\lambda)$ a la propriété :

$$\begin{cases} \psi_{c_1}(\lambda) - \varphi_k(\lambda) > 0 & \text{pour } \lambda \text{ assez petit} \\ \psi_{c_1}(\lambda) - \varphi_k(\lambda) < 0 & \text{pour } \lambda \text{ assez grand} \end{cases}$$

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

$$\varphi_k(\overline{\lambda}) = \psi_{c_1}(\overline{\lambda}) \text{ et } \varphi_k(\lambda) < \psi_{c_1}(\lambda) \text{ pour } 0 < \lambda < \overline{\lambda}$$

De la même manière, il existe $\widetilde{\lambda} > 0$ tel que :

$$\varphi_k(\widetilde{\lambda}) = \psi_{c_2}(\widetilde{\lambda}) \text{ et } \varphi_k(\lambda) < \psi_{c_2}(\lambda) \text{ pour } 0 < \lambda < \widetilde{\lambda}$$

et comme $\psi_{c_2}(\lambda) < \psi_{c_1}(\lambda)$ pour $\lambda > 0$, forcément $\widetilde{\lambda} < \overline{\lambda}$ et $\lambda = \overline{\lambda}$ satisfait (2.13) et (2.14)

$$\begin{pmatrix} \psi_{c_2}(\widetilde{\lambda}) = \ \varphi_k(\widetilde{\lambda}) < \psi_{c_1}(\lambda) \text{ n'est autre que} \\ f(x_k) + c_2\widetilde{\lambda}\nabla^T f(x_k).d_k = f(x_k + \widetilde{\lambda}d_k) < f(x_k) + c_1\widetilde{\lambda}\nabla^T f(x_k).d_k \end{pmatrix}$$

Ce qu'il fallait de démontrer. □

2.2.6 La règle de Wolfe [1969] :

La règle de Wolfe fait appel au calcul de $\varphi_k'(\lambda)$,elle est donc en théorie plus couteuse que la règle de Goldstein&Price. Cependant dans de nombreuse application, le calcul du $\nabla f(x)$ représente un faible cout additionnel en comparaison du coute d'évaluations de $f(x)$ c'est pourquoi cette règle est très utilisée

Etant donnes deux réels $c_{1,}c_2$ tel que $0 < c_1 < c_2 < 1$,ces condition sont :

$$\begin{aligned} f(x_k + \lambda_k d_k) &\leq f(x_k) + c_1\lambda_k\nabla^T f(x_k).d_k \quad c_1 \in]0,1[\, 2.17 & (2.7) \\ \nabla^T f(x_k + \lambda_k d_k).d_k &\geq c_2\nabla^T f(x_k).d_k \qquad c_2 \in]c_1, 1[\, 2.18 & (2.8) \end{aligned}$$

Autrement dit :

$$\begin{aligned} \varphi_k(\lambda) &\leq \varphi_k(0) + c_1\lambda\varphi_k^{'}(0)\, 2.19 & (2.9) \\ \varphi_k^{'}(\lambda) &\geq c_2\varphi_k^{'}(0)2.20 & (2.10) \end{aligned}$$

On voit bien à la figure (2.6) ce que signifie les conditions (2.19) et (2.20) ftbpFU5.4933in2.7216in0pt**Règle de Wolfe**Figure

Remarque 2.4

- La deuxième condition de Wolfe est appelé condition de courbure.

- Les λ_k sélectionnés par la condition de Wolfe (2.17) peuvent être très petits. Ceci peut avoir des conséquences fâcheuses sur la convergence de l'algorithme. La condition de Wolfe (2.18) évite cet inconvénient et supprime les trés petites valeurs de λ_k

Test de Wolfe

$\diamondsuit$ Si

$$\varphi_k(\lambda) \leq \varphi_k(0) + c_1\lambda\varphi_k'(0) \text{ et } \varphi_k'(\lambda) \geq c_2\varphi_k'(0)$$

Alors λ convient.

$\diamondsuit$ Si

$$\varphi_k(\lambda) > \varphi_k(0) + c_1\lambda\varphi_k'(0)$$

Alors λ est trop grand.

$\diamondsuit$ Si

$$\varphi_k'(\lambda) < c_2\varphi_k'(0)$$

Alors λ est trop petit. $\square$

Algorithme 2.4 (la règle de Wolfe)

Etape initiale :

On dispose de $\lambda_g = 0, \lambda_d =$ une valeur maximale quelconque, et soit $\lambda \in]\lambda_g, \lambda_d[$

poser $c_1 = 0,1$, $c_2 = 0,7$

Etapes principales :

Etape 1 : calculer $\varphi(\lambda) = f(x_k + \lambda d_k)$

Si $\varphi(\lambda) \leq \varphi(0) + c_1\lambda\varphi'(0)$ alors aller à l'étape 2

Sinon prendre $\lambda_d = \lambda$ et aller à l'étape 4

Etape 2 : calculer $\varphi'(\lambda) = \nabla^T f(x_k + \lambda d_k).d_k$

Si $\varphi'(\lambda) \geq c_2\ \varphi'(0)$ stop

Sinon aller à l'étape 3

Etape 3 : poser $\lambda_g = \lambda$

Etape 4 : rechercher un nouveau $\lambda \in]\lambda_g, \lambda_d[$ et retourner à l'étape2 . $\square$

Théorème 2.4

Si $\varphi_k : \mathbb{R}_+ \to \mathbb{R}$;*définie par* $\varphi_k(\lambda) = f(x_k + \lambda d_k)$ *est continument différentiable et bornée inférieurement, si* d_k *est une direction de descente* *en* x_k $(\varphi'_k(0) < 0)$*et si* $c_1 \in \left]0,1\right[, c_2 \in \left]c_1,1\right[$, *alors l'ensemble des pas vérifiant la règle de Wolfe (faible) (2.17) et (2.18) est non vide.*

Démonstration. [10 ;1996]

On a

$$\begin{aligned} \varphi_k(\lambda) &= f(x_k + \lambda d_k) \\ \psi_{c_1}(\lambda) &= f(x_k) + c_1 \lambda \nabla^T f(x_k).d_k \end{aligned}$$

Le développement de Taylor-Yong en $\lambda = 0$ de φ_k est :

$$\varphi_k(\lambda) = f(x_k + \lambda d_k) = f(x_k) + \lambda \nabla^T f(x_k).d_k + \lambda \xi(\lambda) \text{ où } \xi(\lambda) \to 0, \lambda \to 0.$$

et comme $c_1 \in \left]0,1\right[$ et $\varphi'_k(0) = \nabla^T f(x_k).d_k < 0$ on déduit :

$$f(x_k) + \lambda \nabla^T f(x_k).d_k < f(x_k) + c_1 \lambda \nabla^T f(x_k).d_k \quad \text{pour } \lambda > 0$$

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

$$\varphi_k(\lambda) < \psi_{c_1}(\lambda)$$

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

et $\psi_{c_1}(\lambda) \to -\infty; \lambda \to +\infty$, on déduit que la fonction $\psi_{c_1}(\lambda) - \varphi_k(\lambda)$ a la propriété :

$$\begin{cases} \psi_{c_1}(\lambda) - \varphi_k(\lambda) > 0 & \text{pour } \lambda \text{ assez petit} \\ \psi_{c_1}(\lambda) - \varphi_k(\lambda) < 0 & \text{pour } \lambda \text{ assez grand} \end{cases}$$

Donc s'annule au moins une fois pour $\lambda > 0$.

En choisissant le plus petit de ces zéros on voit qu'il existe $\overline{\lambda} > 0$ tel que

$$\varphi_k(\overline{\lambda}) = \psi_{c_1}(\overline{\lambda}) \text{ et } \varphi_k(\lambda) < \psi_{c_1}(\lambda) \text{ pour } 0 < \lambda < \overline{\lambda} \qquad (*)$$

La formule des accroissements finis fournit alors un nombre $\widetilde{\lambda}$,$0 < \widetilde{\lambda} < \overline{\lambda}$ tel que

$$\begin{aligned} \varphi_k(\overline{\lambda}) - \varphi_k(0) &= \overline{\lambda} \varphi'_k(\widetilde{\lambda}) = \overline{\lambda} \nabla^T f(x_k + \widetilde{\lambda} d_k).d_k \\ &\Longrightarrow c_1 \overline{\lambda} \nabla^T f(x_k).d_k = \overline{\lambda} \nabla^T f(x_k + \widetilde{\lambda} d_k).d_k \\ &\Longrightarrow \nabla^T f(x_k + \widetilde{\lambda} d_k).d_k = c_1 \nabla^T f(x_k).d_k \geq c_2 \nabla^T f(x_k).d_k \end{aligned}$$

car $0 < c_1 < c_2 < 1$ et $\nabla^T f(x_k).d_k < 0$

Donc $\widetilde{\lambda}$ satisfait (2.18)
D'autre part $\lambda = \widetilde{\lambda}$ satisfait (2.17),en effet,
$\widetilde{\lambda}$ satisfait (*) $\quad \varphi_k(\widetilde{\lambda}) < \psi_{c_1}(\widetilde{\lambda})$ n'est autre que :

$$f(x_k + \widetilde{\lambda} d_k) < f(x_k) + c_1 \widetilde{\lambda} \nabla^T f(x_k).d_k$$

Ce qu'il fallait démontrer. $\square$

La règle de Wolfe forte [1971]

Pour certains algorithmes il est parfois nécessaire d'avoir une condition plus restrictive que (2.18).

Pour cela la deuxième condition (2.18) est remplacée par

$$\left|\nabla^T f(x_k + \lambda d_k).d_k\right| \leq c_2 \left|\nabla^T f(x_k).d_k\right| = -c_2 \nabla^T f(x_k).d_k.$$

On aura donc les conditions de Wolfe fortes :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + c_1 \lambda_k \nabla^T f(x_k).d_k \tag{2.21}$$

$$\left|\nabla^T f(x_k + \lambda_k d_k).d_k\right| \leq -c_2 \nabla^T f(x_k).d_k \tag{2.22}$$

Autrement dit :

$$\varphi_k(\lambda) \leq \varphi_k(0) + c_1 \lambda \varphi_k'(0) \tag{2.23}$$

$$|\varphi_k'(\lambda)| \leq -c_2 \varphi_k'(0) \tag{2.24}$$

Où $0 < c_1 < c_2 < 1$.

Remarque 2.5

- On voit bien que les conditions de Wolfe fortes impliquent les conditions de Wolfe faibles. Effectivement (2.21) est équivalente à (2.17) tandis que (2.22)

$$\begin{aligned}
\left|\nabla^T f(x_k + \lambda_k d_k).d_k\right| &\leq -c_2 \nabla^T f(x_k).d_k \\
&\Leftrightarrow c_2 \nabla^T f(x_k).d_k \leq \nabla^T f(x_k + \lambda_k d_k).d_k \leq -c_2 \nabla^T f(x_k).d_k \\
&\Rightarrow c_2 \nabla^T f(x_k).d_k \leq \nabla^T f(x_k + \lambda_k d_k).d_k \quad (2.18) \qquad \square
\end{aligned}$$

- Le pas λ_k sélectionné par les conditions (2.17) et (2.18) peut être très loin d'un point optimal ou stationnaire de la fonction φ. la condition (2.22) assure que le pas λ_k se trouve dans le voisinage d'un point stationnaire ou un point optimale de φ.

la règle de Wolfe relaxée [1996]

proposée par Dai et Yuan , cette règle consiste à choisir le pas satisfaisant aux conditions :

$$\begin{aligned} f(x_k+\lambda_k d_k) &\leq f(x_k)+c_1\lambda\nabla^T f(x_k).d_k 2.25 &\quad (2.11)\\ \rho_1\nabla^T f(x_k).d_k &\leq \nabla^T f(x_k+\lambda_k d_k).d_k \leq -\rho_2\nabla^T f(x_k).d_k 2.26 &\quad (2.12)\end{aligned}$$

autrement dit :

$$\varphi_k(\lambda) \leq \varphi_k(0)+c_1\lambda\varphi_k'(0) \tag{2.27}$$

$$\rho_1\varphi_k'(0) \leq \varphi_k'(\lambda) \leq -\rho_2\varphi_k'(0) \tag{2.28}$$

Où $0 < c_1 < \rho_1 < 1$ et $\rho_2 > 0$.

Remarque 2.6

On voit bien que les conditions de Wolfe relaxée impliquent les conditions de Wolfe fortes. Effectivement (2.25) est équivalente à (2.17), tandis que pour le cas particulier $\rho_1=\rho_2=\rho$, (2.26) est équivalente à (2.22). En effet :

$$\begin{aligned} \rho_1\nabla^T f(x_k).d_k &\leq \nabla^T f(x_k+\lambda_k d_k).d_k \leq -\rho_2\nabla^T f(x_k).d_k\\ &\Rightarrow \rho\nabla^T f(x_k).d_k \leq \nabla^T f(x_k+\lambda_k d_k).d_k \leq -\rho\nabla^T f(x_k).d_k\\ &\Rightarrow \left|\nabla^T f(x_k+\lambda_k d_k).d_k\right| \leq -\rho\nabla^T f(x_k).d_k \text{ (2.22)} \qquad \square\end{aligned}$$

Remarque 2.7

Les conditions de Wolfe relaxée impliquent les conditions de Wolfe faibles. Effectivement (2.25) est équivalente à (2.17), tandis que pour le cas particulier $\rho_1=\rho$ et $\rho_2=+\infty$, (2.26) est équivalente à (2.18). En effet :

$$\begin{aligned} \rho_1\nabla^T f(x_k)d_k &\leq \nabla^T f(x_k+\lambda_k d_k)d_k \leq -\rho_2\nabla^T f(x_k)d_k\\ &\Rightarrow \rho\nabla^T f(x_k)d_k \leq \nabla^T f(x_k+\lambda_k d_k)d_k \text{ (2.18)} \qquad \square\end{aligned}$$

2.3 Convergence des méthodes à directions de descente

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

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

$$f(x_{k+1}) \leq f(x_k) - c \|\nabla f(x_k)\|^2 \cos^2 \theta_k \tag{2.29}$$

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

$$\cos \theta_k = \frac{-\nabla^T f(x_k) d_k}{\|\nabla f(x_k)\| \, \|d_k\|} \tag{2.30}$$

Voici comment on se sert de la condition de Zoutendijk.

Proposition 2.1 *Si la suite $\{x_k\}$ générée par un algorithme d'optimisation vérifie la condition de Zoutendijk (2.29) et si la suite $\{f(x_k)\}$ est minorée, alors*

$$\sum_{k \geq 1} \|\nabla f(x_k)\|^2 \cos^2 \theta_k < \infty \tag{2.31}$$

Démonstration. [10,1996]

En sommant les inégalités (2.29), on a

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

et puisque la suite $\{f(x_k)\}$ est minorée, c'est-à-dire $\exists c^{'} > 0$ telle que pour tout k, $f(x_k) \geq c'$alors

$$\sum_{k \geq 1} \|\nabla f(x_k)\|^2 \cos^2 \theta_k < \infty \quad \textit{quand } l \to \infty \qquad \square$$

Les deux propositions suivantes précisent les circonstances dans les quelles la condition de Zoutendijk (2.29) est vérifiée avec les règles d'Armijo et de Wolfe.

Proposition 2.2 *Soit* $f : \mathbb{R}^n \to \mathbb{R}$ *une fonction* $C^{1,1}$(différentiable et sa dérivée vérifie pour une constante L et pour tout x et y :

$\|\nabla f(y) - \nabla f(x)\| \leq L \|y - x\|$) *dans un voisinage de* $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$.

On considère un algorithme à directions de descente d_k*, qui génère une suite* $\{x_k\}$ *en utilisant la recherche linéaire d'Armijo avec* $\lambda_1 > 0$.

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

$$f(x_{k+1}) \leq f(x_k) - c\nabla^T f(x_k).d_k \tag{2.32}$$

ou

$$f(x_{k+1}) \leq f(x_k) - c\|\nabla f(x_k)\|^2 \cos^2 \theta_k \tag{2.33}$$

est vérifiée.

Démonstration. [10,1996]

Si le pas $\lambda_k = \lambda_1$ est accepté, on a (2.32), en effet

d_k direction de descente implique que $f(x_{k+1}) < f(x_k)$ et car λ_1 est uniformément positif alors :

$$f(x_{k+1}) \leq f(x_k) - c\nabla^T f(x_k).d_k$$

Dans le cas contraire ($\lambda_k \neq \lambda_1$),la condition d'Armijo (Wolf1) n'est pas vérifiée avec un pas $\lambda'_k \leq \frac{\lambda_k}{\tau}$, c'est-à-dire

$$f(x_k + \lambda'_k d_k) > f(x_k) + \rho\lambda'_k \nabla^T f(x_k).d_k$$

Comme f est continument différentiable, on a pour tout $\lambda_k > 0$:

$$\begin{aligned} f(x_k + \lambda_k d_k) &= f(x_k) + \lambda_k \nabla^T f(x_k).d_k + \int_0^1 [\nabla f(x_k + t\lambda_k d_k) - \nabla f(x_k)]^T \lambda_k d_k dt \\ &\Rightarrow f(x_k + \lambda_k d_k) \leq f(x_k) + \lambda_k \nabla^T f(x_k).d_k + c\lambda_k^2 \|d_k\|^2 \end{aligned}$$

Où c ¿ 0 est une constante. Avec l'inégalité précédente, et le fait que , on obtient :

$$\begin{cases} f(x_k + \lambda'_k d_k) - f(x_k) > \rho\lambda'_k \nabla^T f(x_k).d_k \\ f(x_k + \lambda'_k d_k) - f(x_k) \leq \lambda'_k \nabla^T f(x_k).d_k + c\lambda'^2_k \|d_k\|^2 \end{cases}$$

$$\begin{aligned} &\Rightarrow \quad \rho\lambda'_k \nabla^T f(x_k).d_k \leq \lambda'_k \nabla^T f(x_k).d_k + c\lambda'^2_k \|d_k\|^2 \\ &\Rightarrow \quad -c\lambda'^2_k \|d_k\|^2 \leq (1-\rho)\lambda'_k \nabla^T f(x_k).d_k \end{aligned}$$

Or

$$\rho < 1 \Rightarrow 0 < 1 - \rho < 1 \Rightarrow \frac{1}{1-\rho} > 1$$

D'où

$$\begin{aligned} \nabla^T f(x_k).d_k &\geq \frac{-c}{1-\rho}\lambda'_k \|d_k\|^2 \Rightarrow -\nabla^T f(x_k).d_k \leq \frac{c}{1-\rho}\lambda'_k \|d_k\|^2 \\ &\Rightarrow \left|\nabla^T f(x_k).d_k\right| = \left\|\nabla^T f(x_k)\right\| \|d_k\| \cos\theta_k \leq \frac{c}{1-\rho}\lambda'_k \|d_k\|^2 \end{aligned}$$

ce qui permet de minorer $\lambda'_k \|d_k\|$ et donc aussi $\lambda_k \|d_k\|$ par une constante fois $\nabla^T f(x_k) \cos\theta_k$. Cette minoration et l'expression suivante de la condition d'Armijo (Wolf 1)

$$f(x_k + \lambda_k d_k) \leq f(x_k) - \frac{\rho}{1-\rho} c\lambda_k \left\|\nabla^T f(x_k)\right\| \|d_k\| \cos\theta_k$$

car $\frac{\rho}{1-\rho} < 1$ on obtient

$$f(x_k + \lambda_k d_k) < f(x_k) - c\left\|\nabla^T f(x_k)\right\|^2 \cos^2\theta_k \qquad \square$$

Proposition 2.3

Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction $C^{1,1}$(différentiable et sa dérivée vérifie pour une constante L et pour tout x et y :
$\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|$) *dans un voisinage de $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$.*
On considère un algorithme à directions de descente d_k, qui génère une suite $\{x_k\}$ en utilisant la recherche linéaire de Wolfe (2.17)-(2.18) avec

$$\lambda_1 > 0$$

Alors il existe une constante $c > 0$ telle que, pour tout $k \geq 1$, la condition de Zoutendijk (2.29) est vérifiée.

Démonstration. [10,1996]
D'après (2.17)

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

$$\begin{array}{cl} \Rightarrow & \left(\nabla f(x_k + \lambda_k d_k) - \nabla f(x_k)\right)^T .d_k \geq (\sigma - 1)\, \nabla^T f(x_k).d_k \\ = & -(1-\sigma)\ \nabla^T f(x_k).d_k = (1-\sigma)\left|\nabla^T f(x_k).d_k\right| \\ \Leftrightarrow & (1-\sigma)\left|\nabla^T f(x_k).d_k\right| \leq \left(\nabla f(x_k + \lambda_k d_k) - \nabla f(x_k)\right)^T .d_k \end{array}$$

et du fait que f est contiument différentiable :

$$\begin{array}{rcl} (1-\sigma)\left|\nabla^T f(x_k).d_k\right| & = & (1-\sigma)\left\|\nabla^T f(x_k)\right\| \|d_k\| \cos\theta_k \\ & \leq & \|\nabla f(x_k + \lambda_k d_k) - \nabla f(x_k)\| \, \|d_k\| \\ & \Rightarrow & (1-\sigma)\left\|\nabla^T f(x_k)\right\| \cos\theta_k \leq L\lambda_k \|d_k\| \\ & \Rightarrow & \lambda_k \|d_k\| \leq \dfrac{(1-\sigma)}{L}\left\|\nabla^T f(x_k)\right\| \cos\theta_k \end{array}$$

En utilisant (2.17), on aura :

$$\begin{array}{rcl} f(x_k + \lambda_k d_k) & \leq & f(x_k) + \rho\lambda_k \nabla^T f(x_k).d_k \\ & \Rightarrow & f(x_k + \lambda d_k) \leq f(x_k) + \rho\lambda \nabla^T f(x_k).d_k \leq f(x_k) + \left|\rho\lambda \nabla^T f(x_k).d_k\right| \\ & \Rightarrow & f(x_k + \lambda d_k) \leq f(x_k) + \rho\lambda \left|\nabla^T f(x_k).d_k\right| \leq f(x_k) - \rho\lambda \nabla^T f(x_k).d_k \\ & \Rightarrow & f(x_k + \lambda d_k) \leq f(x_k) - \rho\lambda \left\|\nabla^T f(x_k)\right\| \|d_k\| \cos\theta_k \\ & \Rightarrow & f(x_k + \lambda d_k) \leq f(x_k) - \dfrac{\rho(1-\sigma)}{L}\left\|\nabla^T f(x_k)\right\|^2 \cos^2\theta_k \end{array}$$

On en déduit (2.29). $\square$

Chapitre 3

Méthodes 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 résoudre 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

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

Où f est régulière (continûment différentiable)

Dans ce chapitre on va décrire toutes ces méthodes, mais avant d'accéder à ces derniers, on va d'abord donner le principe général d'une méthode à directions conjuguées

3.1 Le principe général d'une méthode à directions conjuguées

Donnons la définition de "*conjugaison*" :

Définition 3.1. Soit H une matrice symétrique $n \times n$, définie positive. On dit que deux vecteurs x et y de $\mathbb{R}^n$ sont $H-$conjugués (ou conjugués par rapport à H) s'ils vérifient

$$x^T H y = 0 \tag{3.1}$$

3.1.1 Description de la méthode

Soit $\{d_0, d_1, ..., d_n\}$ une famille de vecteurs H-conjugués. On appelle alors méthode de directions conjuguées toute méthode itérative appliquée à une fonction quadratique strictement convexe de n variables :

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

Avec $x \in \mathbb{R}^n$ et $H \in M_{n\times n}$ est symétrique et définie positive, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$ .conduisent à l'optimum en n étapes au plus. Et cette méthode de la forme :

$$x_0 \text{ donné}$$

$$x_{k+1} = x_k + \lambda_k d_k \tag{3.2}$$

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

Si l'on note $g_k = \nabla f(x_k)$, la méthode se construit comme suit :

Calcul de λ_k

Comme λ_k minimise f dans la direction d_k, on a, $\forall k$:

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

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

Soit :

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

D'où l'on tire :

$$\lambda_k = \frac{-d_k^T (Hx_k + b)}{d_k^T H d_k} \tag{3.3}$$

Théorème 3.1.

Pour tout $x_0 \in \mathbb{R}^n$ la suite $\{x_k\}$ généré par l'algorithme (3.2)-(3.3) converge vers la solution x^* du système linéaire $Hx = b$ en n étapes au plus.

Demonstration. [37,1991]

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

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

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

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

Démontrons que β_k coïncide avec λ_k donnée par (3.3)
Si généré par l'algorithme (3.2)-(3.3) , Alors on obtient :

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

Multiplions cette expression par $d_k^T H$ et on utilise (3.1) on obtient que :

$$d_k^T H (x_k - x_0) = 0$$

Et donc

$$x^* - x_0 = (x^* - x_k) + (x_k - x_0)$$

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

On compare cette relation avec (3.3) et (**) on trouve :

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

Donc $\beta_k = \lambda_k$
Ce qui achéve la démonstration $\square$

3.2 La méthode du gradient conjugué

Soient

$$\begin{aligned} \nabla f(x) &= Hx + b \\ \text{Et } \nabla^2 f(x) &= H \end{aligned}$$

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

Notons que la méthode du gradient conjugué est inspirée de celle du gradient (plus profonde pente).

3.3 Algorithme de la méthode du gradient conjugué

3.3.1 Algorithme de La méthode du gradient conjugué pour les fonctions quadratiques (cas linéaire)

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

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

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

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

$$x_{k+1} = x_k + \lambda_k d_k \tag{3.4}$$

L'idée de la méthode est :

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_{k-1}c'est-à-dire

$$d_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} d_k \tag{3.5}$$

les coefficients β_{k+1}étant choisis de telle manière que d_k soit conjuguée avec toutes les directions précédentes.

Autrement dit :

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

On en déduit :

$$\begin{aligned} d_{k+1}^T H d_k &= 0 \Rightarrow \left(-\nabla f(x_{k+1}) + \beta_{k+1} d_k\right)^T H d_k = 0 \\ &\Rightarrow -\nabla^T f(x_{k+1}) H d_k + \beta_{k+1} d_k^T H d_k = 0 \\ &\Rightarrow \beta_{k+1} = \frac{\nabla^T f(x_{k+1}) H d_k}{d_k^T H d_k} = \frac{g_{k+1}^T H d_k}{d_k^T H d_k} \end{aligned}$$

2-déterminer le pas λ_k :

en particulier, une façon de choisir λ_k peut être de résoudre le problème d'optimisation (à une seule variable)

$$\lambda_k = \min f(x_k + \lambda d_k) \quad , \quad \lambda > 0 \tag{3.6}$$

on en déduit :

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

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

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

$$\Longrightarrow \lambda_k = \frac{-d_k^T (H x_k + b)}{d_k^T H d_k} = \frac{-d_k^T g_k}{d_k^T H d_k} \tag{3.7}$$

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

Algorithme 3.1 (Algorithme du gradient conjugué "linéaire")

Etape 0 : (initialisation)

Soit x_0 le point de départ, $g_0 = \nabla f(x_0) = Hx_0 + b$, 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"
Sinon aller à l'étape 2.

Etape 2 :

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

$$\lambda_k = \frac{-d_k^T g_k}{d_k^T H d_k}$$

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

$$\beta_{k+1} = \frac{g_{k+1}^T H d_k}{d_k^T H d_k} \tag{3.8}$$

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

La validité de l'algorithme du gradient conjugué linéaire

On va maintenant montrer que l'algorithme ci-dessus définit bien une méthode de directions conjuguées.

Théorème 3.2. *A une itération k quelconque de l'algorithme où l'optimum de $f(x)$ n'est pas encore atteint (c'est-à-dire $g_i \neq 0$, $i = 0, 1, ..., k$) on a :*

a-

$$\lambda_k = \frac{g_k^T g_k}{d_k^T H d_k} \neq 0 \tag{3.9}$$

b-

$$\beta_{k+1} = \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k} \tag{3.10}$$

$$= \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \tag{3.11}$$

c- *Les directions $d_0, d_1, ..., d_{k+1}$ engendrées par l'algorithme sont mutuellement conjuguées.*

Démonstration. [35,1981]

On raisonne par récurrence sur k en supposant que $d_0, d_1, ..., d_k$ sont mutuellement conjuguées.

a- Montrons d'abord l'équivalence de (3.7) et de (3.9)

On a : $d_k = -g_k + \beta_k d_{k-1}$.

Donc (3.7) s'écrit :

$$\begin{aligned}\lambda_k &= \frac{-d_k^T g_k}{d_k^T H d_k} \\ &= \frac{-\left[-g_k + \beta_k d_{k-1}\right]^T g_k}{d_k^T H d_k} \\ &= \frac{g_k^T g_k}{d_k^T H d_k} - \beta_k \frac{d_{k-1}^T g_k}{d_k^T H d_k}\end{aligned}$$

Comme $(d_0, d_1, ..., d_{k-1})$ sont mutuellement conjuguées, x_k est l'optimum de $f(x)$.

Donc $d_{k-1}^T g_k = 0$ d'où l'on déduit (3.9).

b- Pour démontrer (3.10) remarquons que :

$$\begin{aligned}g_{k+1} - g_k &= H(x_{k+1} - x_k) = \lambda_k H d_k \\ &\Rightarrow H d_k = \frac{1}{\lambda_k}\left[g_{k+1} - g_k\right]\end{aligned}$$

On a alors :

$$g_{k+1}^T H d_k = \frac{1}{\lambda_k} g_{k+1}^T \left[g_{k+1} - g_k\right]$$

Et en utilisant (3.9)

$$\lambda_k = \frac{g_k^T g_k}{d_k^T H d_k}$$

Il vient

$$\begin{aligned}g_{k+1}^T H d_k &= \frac{d_k^T H d_k}{g_k^T g_k} . g_{k+1}^T \left[g_{k+1} - g_k\right] \\ &\Rightarrow \frac{g_{k+1}^T H d_k}{d_k^T H d_k} = \frac{g_{k+1}^T \left[g_{k+1} - g_k\right]}{g_k^T g_k}\end{aligned}$$

Or de (3.8) on aura :

$$\beta_{k+1} = \frac{g_{k+1}^T H d_k}{d_k^T H d_k} = \frac{g_{k+1}^T \left[g_{k+1} - g_k\right]}{g_k^T g_k}$$

Ce qui démontre (3.10).

(3.11) découle alors du fait que :

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

Car

$$g_k = d_k - \beta_k d_{k-1}$$

Appartient au sous-espace engendré par $(d_0, d_1, ..., d_k)$ et que g_{k+1} est orthogonal à ce sous-espace .

c- Montrons enfin que d_{k+1} est conjuguée par rapport à $(d_0, d_1, ..., d_k)$.

On a bien $d_{k+1}^T H d_k$ car, en utilisant $d_{k+1} = -g_{k+1} + \beta_k d_k$ on aura :

$$\begin{aligned} d_{k+1}^T H d_k &= (-g_{k+1} + \beta_{k+1} d_k)^T H d_k \\ &= -g_{k+1}^T H d_k + \beta_{k+1} d_k^T H d_k \\ &= -g_{k+1}^T H d_k + \frac{g_{k+1}^T H d_k}{d_k^T H d_k} d_k^T H d_k \\ &= 0 \end{aligned}$$

Vérifions maintenant que :

$$d_{k+1}^T H d_i = 0 \quad \text{pour } i = 0, 1, ..., k-1.$$

On a :

$$d_{k+1}^T H d_i = -g_{k+1}^T H d_i + \beta_{k+1} d_k^T H d_i$$

Le seconde terme est nul par l'hypothèse de récurrence ($(d_0, d_1, ..., d_k$)sont mutuellement conjuguées).

Montrons qu'il en est de même du premier terme. Puisque $x_{i+1} = x_i + \lambda_i d_i$ et que $\lambda_i \neq 0$ on a :

$$\begin{aligned} H d_i &= \frac{1}{\lambda_i} (H x_{i+1} - H x_i) \\ &= \frac{1}{\lambda_i} (g_{i+1} - g_i) \end{aligned}$$

En écrivant :

$$\begin{aligned} g_{i+1} &= d_{i+1} - \beta_i d_i \\ g_i &= d_i - \beta_{i-1} d_{i-1} \end{aligned}$$

on voit que Hd_i est combinaison linéaire de d_{i+1}, d_i et de d_{i-1} seulement.

Mais puisque $(d_0, d_1, ..., d_k)$ sont mutuellement conjuguées, on sait que le point x_{k+1} est l'optimum de $f(x)$.

Donc g_{k+1} est orthogonal au sous-espace engendré par $(d_0, d_1, ..., d_k)$ et comme Hd_i appartient à ce sous-espace pour $i = 0, 1, ..., k-1$, on en déduit $g_{k+1}^T Hd_i = 0$.

ce qui achève la démonstration. $\square$

Remarque 3.1.

dans ce cas d_k est une direction de descente puisque

$$\begin{aligned} d_k^T \nabla f(x_k) &= \left(-\nabla f(x_k) + \beta_{k+1} d_{k-1}\right)^T \nabla f(x_k) \\ &= -\nabla f(x_k)^T \nabla f(x_k) + \beta_{k+1} d_{k-1}^T \nabla f(x_k) \\ &= -\left\|\nabla f(x_k)\right\|^2 \quad (\text{car } d_{k-1}^T \nabla f(x_k) = 0 \ ) \end{aligned}$$

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

Les avantages de la méthode du gradient conjugué linéaire

1- la consommation mémoire de l'algorithme est minime : on doit stocker les quatre vecteurs x_k, g_k, d_k, Hd_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}, Hd_{k+1}) et les scalaires λ_k, β_{k+1}.

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

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

Différentes formules de β_{k+1} dans le cas linéaire

Soient

$$\lambda_k = \frac{-d_k^T g_k}{d_k^T H d_k}$$

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

Remarquons que

$$\begin{aligned} g_{k+1} - g_k &= H(x_{k+1} - x_k) = \lambda_k H d_k \\ &\Rightarrow \; H d_k = \frac{1}{\lambda_k}(g_{k+1} - g_k) \end{aligned}$$

$\diamondsuit$**Formule de Hestenes-Stiefel** Cette méthode été découverte en 1952 par Hestenes et Stiefels [28,1952],

$$\begin{aligned} \beta_{k+1}^{HS} &= \frac{g_{k+1}^T H d_k}{d_k^T H d_k} = \frac{g_{k+1}^T (g_{k+1} - g_k)/\lambda_k}{d_k^T (g_{k+1} - g_k)/\lambda_k} \\ &= \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)} \end{aligned}$$

D'où

$$\beta_{k+1}^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \tag{3.12}$$

où $y_k = g_{k+1} - g_k$

$\diamondsuit$**Formule de Fletcher-Reeves** Cette méthode été découverte en 1964 par Fletcher et Reeves [19, 1964],

$$\beta_{k+1}^{FR} = \frac{g_{k+1}^T H d_k}{d_k^T H d_k} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)}$$

Or puisque $\lambda_k = \min f(x_k + \lambda d_k)$, $\forall \lambda > 0$ donc

$$\nabla f(x_k + \lambda_k d_k) = d_k^T g_{k+1} = d_{k-1}^T g_k = 0$$

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

D'autre part

$$g_{k+1}^T g_k = g_{k+1}^T(-d_k + \beta_{k-1}d_{k-1}) = -g_{k+1}^T d_k + \beta_{k-1}g_{k+1}^T d_{k-1}$$

$$= \beta_{k-1}g_{k+1}^T d_{k-1} = \beta_{k-1}(g_k + \lambda_k H d_k)^T d_{k-1}$$

$$= \beta_{k-1}g_k^T d_{k-1} + \lambda_k \beta_{k-1} d_k^T H d_{k-1} \ = 0$$

Car les directions sont mutuellement conjuguées par rapport à H
On aura donc

$$\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \tag{3.13}$$

◇**Formule de Polak-Ribière-Polyak** Cette méthode été découverte en 1969 par Polak, Ribière [38,1969] et Ployak [39, 1969]

$$\begin{aligned}
\beta_{k+1}^{PRP} &= \frac{g_{k+1}^T H d_k}{d_k^T H d_k} = \frac{g_{k+1}^T(g_{k+1} - g_k)}{d_k^T(g_{k+1} - g_k)} \\
&= \frac{g_{k+1}^T(g_{k+1} - g_k)}{d_k^T g_{k+1} - d_k^T g_k} = \frac{g_{k+1}^T(g_{k+1} - g_k)}{-d_k^T g_k} \\
&= \frac{g_{k+1}^T(g_{k+1} - g_k)}{-(-g_k + \beta_{k-1}d_{k-1})^T g_k} = \frac{g_{k+1}^T(g_{k+1} - g_k)}{g_k^T g_k - \beta_{k-1}d_{k-1}^T g_k} \\
&= \frac{g_{k+1}^T(g_{k+1} - g_k)}{\|g_k\|^2}
\end{aligned}$$

D'où

$$\beta_{k+1}^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2} \tag{3.14}$$

$\diamondsuit$**Formule de la descente conjuguée** Cette méthode été découverte en 1987 par Fletcher [17,1987],

$$\beta_{k+1}^{CD} = \frac{g_{k+1}^T H d_k}{d_k^T H d_k} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)}$$

$$= \frac{g_{k+1}^T g_{k+1} - g_{k+1}^T g_k}{d_k^T g_{k+1} - d_k^T g_k} = \frac{\|g_{k+1}\|^2 - g_{k+1}^T g_k}{-d_k^T g_k}$$

D'où

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

$\diamondsuit$**Formule de Dai et Yuan** Cette méthode été découverte en 1999 par Dai et Yuan [9,1999],

$$\beta_{k+1}^{DY} = \frac{g_{k+1}^T H d_k}{d_k^T H d_k} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)}$$

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

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

D'où

$$\beta_{k+1}^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} \tag{3.16}$$

Remarque 3.2. Dans le cas quadratique avec recherche linéaire exacte on a vu que :

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

Dans le cas non quadratique, ces quantités ont en général des valeurs différentes.
$\square$

3.3.2 Algorithme de La méthode du gradient conjugué pour les fonctions quelconques (cas non linéaire)

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

$$\min \{ f(x); \quad x \in \mathbb{R}^n \} \tag{3.17}$$

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

$$x_{k+1} = x_k + \lambda_k d_k \tag{3.18}$$

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

$$d_k = \begin{cases} -g_1 & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases} \tag{3.19}$$

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

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

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

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

$$\beta_{k+1}^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}$$

Où $y_k = g_{k+1} - g_k$.

Algorithme 3.2(des différentes méthodes du gradient conjugué non linéaire)

Etape0 : (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"

Si non aller à l'étape 2.

Etape 2 :

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

λ_k : calculer par la recherche linéaire

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

Où

β_{k+1} : définir selon la méthode

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

La propriété de descente de la méthode du gradient conjugué non linéaire

Cas de recherche linéaire exacte Powell [41,1984] a démontré la satisfaction de la propriété de descente de la fonction objective pour la méthode de Fletcher-Reeves avec recherche linéaire exacte.

Soit $x_{k+1} = x_k + \lambda_k d_k$ la méthode itérative pour résoudre le problème, avec λ_k : calculer par la recherche linéaire exacte, et

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

Alors : quelque soit $\beta_k \in \mathbb{R}$, d_k est une direction de descente,$\forall k \geq 1$

Explication : si le pas λ_{k-1} est un point stationnaire de

$$\lambda \longmapsto f(x_{k-1} + \lambda d_{k-1}).$$

En effet, dans ce cas $d_{k-1}^T g_k = 0$ et on trouve lorsque $g_k \neq 0$:

$$\begin{aligned} d_k^T g_k &= \left(-g_k + \beta_k d_{k-1}\right)^T g_k \\ &= -\left\|g_k\right\|^2 + \beta_k d_{k-1}^T g_k = -\left\|g_k\right\|^2 < 0 \end{aligned}$$

Cependant, Il est fortement déconseillé de faire la recherche linéaire exacte lorsque f n'est pas quadratique : le coût de détermination de k est excessif

Cas de recherche linéaire inexacte Le premier théorème démontrant la descente d'une méthode du gradient conjugué non linéaire était établi par Albaali ([1, 1985]) pour la méthode de Fletcher-Reeves avec la recherche forte de Wolfe avec $\sigma \leq \frac{1}{2}$.

Gilbert et Nocedal ([22, 1992]) ont généralisé ce résultat pour tout algorithme du gradient conjugué dont

$$|\beta_k| \leq \beta_k^{FR}$$

Aucun théorème n'a était établi afin de démontrer la satisfaction de la propriété de descente pour la méthode de Polak-Ribière-Polyak non linéaire. Grippo et Lucidi ([26, 1997]) et P. Armand ([2, 2005]) ont suggéré des modifications dans le choix de λ_k afin d'établir le résultat de la convergence, ainsi la propriété de descente.

Fletcher ([17, 1987]) a démontré que la méthode de la descente conjuguée est une méthode de descente si le pas λ_k est déterminé par la règle forte de Wolfe (4.7)-(4.9) avec $\sigma \leq \frac{1}{2}$.

Dai et Yuan ([10, 1996]) ont démontré que cette méthode avec la règle de Wolfe relaxée (4.7)-(4.10) où $0 < \rho < \sigma_1 < 1$ *et* $0 \leq \sigma_2 \leq 1$, génère des directions de descente suffisante à chaque itération $k \geq 1$.

Dai et Yuan ([13, 1998]) ont démontré qu'à chaque itération $k \geq 1$, la direction recherchée par la méthode de Dai-Yuan avec la recherche de Wolfe faible (4.7)-(4.8), est de descente si la fonction objectif f est strictement convexe.

Chapitre 4

Sur la convergence globale d'une nouvelle classe de méthodes du Gradient Conjugué non linéaire

4.1 Position du problème

Soit $f : \mathbb{R}^n \to \mathbb{R}$ et (P) le problème d'optimisation sans contraintes suivant :

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

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 méthodes du gradient conjugué génèrent des suites $\{x_k\}$ de la forme suivante :

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

Où la direction de recherhe est définie par la formule de récurrence suivante :

$$d_k = \begin{cases} -g_0 & \text{si } k = 0 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 1 \end{cases}$$

Le coefficient β_k détermine la méthode du gradient conjugué. Le pas $\lambda_k \in \mathbb{R}^+$ étant déterminé par une recherche linéaire exacte ou inexacte.

On étudie dans ce mémoire le problème de la convergence globale de la méthode du gradient conjugué, version Fletcher Reeves avec les recherches linéaires inexactes de Wolfe faibles, c'est à dire que les pas λ_k vériient les deux conditions suivantes :

$$\begin{aligned} f(x_k + \lambda d_k) &\leq f(x_k) + \delta\lambda d_k^T g_k 4.0 \\ d_k^T g_{k+1} &\geq \sigma d_k^T g_k \end{aligned} \tag{4.1}$$

avec

$$0 < \delta < \sigma < 1$$

De façon plus précise nous étudions si la relation suivante

$$\lim_{k\to\infty} \inf \|g_k\| = 0 \tag{4.0bis}$$

est vraie ou non. Le même problème a été étudié par Al Baali ([*Al Baali*]) qui a démontré la relation (4.0bis), dans le cas d'une recherche linéaire inexacte de Wolfe forte, c'est à dire que les pas λ_k vérifient les deux conditions suivantes :

$$\begin{aligned} f(x_k + \lambda_k d_k) &\leq f(x_k) + \delta\lambda_k d_k^T g_k \\ \left|d_k^T g_{k+1}\right| &\leq -\sigma d_k^T g_k \end{aligned}$$

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

D'autres auteurs ont démontré qu'on ne peut pas obtenir la relation (4.0bis), avec des recherches linéaires inexactes de Wolfe faibles (voir [Y.H. Dai et Y. Yuan : Convergence properties of the Fletcher Reeves method, 1996]). Le problème majeur rencontré par l'utilisation des recherches linéaires inexactes de Wolfe faibles est le fait que les directions qu'elles engendrent ne sont pas des directions de descente. Dans ce travail, les auteurs ont modifié les coéfficients β_k^{FR}, de sorte que les nouvelles directions obtenues avec des recherches linéaires inexactes de Wolfe faibles, soient des directions de descente. On obtient ainsi une nouvelle famille de gradients conjuguées pour laquelle la relation (4.0bis) est vraie avec des recherches linéaires inexactes de Wolfe faibles.

4.2 Hypothèses et résultats intermediaires

Durant tout ce chapitre on exigera les hypothèses suivantes :

Supposition 4.1.

(i) La fonction f est bornée inférieurement dans $\mathbb{R}^n$

(ii) Dans un voisinage N de $\mathcal{L} = \{x \in \mathbb{R}^n; f(x) \leq f(x_1)\}$, la fonction objectif f est continûment différentiable et son gradient est lipchitzien c'est-à-dire

$$\exists L > 0 \text{ tel que } \quad \|g(x) - g(y)\| \leq L \|x - y\|, \forall x, y \in N \tag{4.4}$$

Remarque 4.1. Notons que ces suppositions impliquent qu'il existe $\gamma > 0$ tel que

$$\|g(x)\| \leq \gamma \ , \forall x \in \mathcal{L} \tag{4.5}$$

Définition 4.1. On dit que d_k est une direction de descente suffisante si

$$d_k^T g_k \leq -C \|g_k\|^2 ; \text{ où } C > 0 \tag{4.6}$$

Rappelons :

les conditions de Wolfe faibles :

$$\begin{aligned} f(x_k + \lambda_k d_k) &\leq f(x_k) + \rho \lambda_k d_k^T g_k 4.7 && (4.2) \\ d_k^T g_{k+1} &\geq \sigma d_k^T g_k 4.8 && (4.3) \end{aligned}$$

Où $\quad 0 < \rho < \sigma < 1.$

Les conditions de Wolfe fortes :

$$\begin{aligned} f(x_k + \lambda_k d_k) &\leq f(x_k) + \rho \lambda_k d_k^T g_k \\ \left|d_k^T g_{k+1}\right| &\leq -\sigma d_k^T g_k 4.9 && (4.4) \end{aligned}$$

Où $\quad 0 < \rho < \sigma < 1.$

Les conditions de Wolfe relaxées :

$$\begin{aligned} f(x_k + \lambda_k d_k) &\leq f(x_k) + \rho \lambda_k d_k^T g_k \\ \sigma_1 d_k^T g_k &\leq d_k^T g_{k+1} \leq -\sigma_2 d_k^T g_k 4.10 && (4.5) \end{aligned}$$

Où $\quad 0 < \rho < \sigma_1 < 1$ et $\sigma_2 > 0.$

Expsons maintenant un théorème fondamental qui assure la satisfaction de la condition de Zoutendijk (voir chapitre2), pour toute méthode du type

(4.2)-(4.3), dans laquelle le pas λ_k est déterminé par la règle de Wolfe faible (4.7)-(4.8). Ce théorème a été démontré par Zoutendijk (47,1970) et Powell (42, 1971).

Théorème 4.1. *Soit x_1 un point de départ pour lequel la supposition 4.1 soit satisfaite. Considérons une méthode du type (4.2) et (4.3) dans laquelle d_k est une direction de descente et le pas λ_k est déterminé par la règle de Wolfe faible (4.7)-(4.8) . Alors pour une telle méthode la condition suivante :*

$$ {}_{k\geq 1}\frac{d_k^T g_k}{\|d_k\|^2} < \infty \tag{4.11}$$

est vérifiée.

Démonstration. [37,1991]

De (4.8) on a :

$$d_k^T y_k = d_k^T (g_{k+1} - g_k) \geq (\sigma - 1)\, d_k^T g_k$$

D'autre part

$$\begin{array}{rcl} d_k^T (g_{k+1} - g_k) & \leq & \|g_{k+1} - g_k\| \, \|d_k\| \\ & \leq & \lambda_k L \|d_k\|^2 \end{array}$$

D'où

$$\lambda_k \geq \left(\frac{\sigma - 1}{L}\right) \frac{d_k^T g_k}{\|d_k\|^2}$$

En remplaçant ceci dans (4.7) on aura :

$$f(x_k) - f(x_k + \lambda_k d_k) \geq -\rho \left(\frac{\sigma - 1}{L}\right) \frac{(d_k^T g_k)^2}{\|d_k\|^2}$$

$$\Longrightarrow f(x_k) - f(x_k + \lambda_k d_k) \geq \mu \frac{(d_k^T g_k)^2}{\|d_k\|^2}$$

où $\mu = \frac{\rho(1-\sigma)}{L}$

Or puisque f est bornée sur N on a :

$${}_{k\geq 1}\frac{d_k^T g_k}{\|d_k\|^2} < \infty$$

Ce qui achève la démonstration. $\square$

4.3 Les nouvelles formules de β_k. Méthode de Dai-Yuan

Une motivation pour nos nouvelles méthodes est la propriété de descente de la méthode de descente conjuguée

$$\beta_{k+1}^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k} \tag{4.12}$$

Remarque 4.2.

La méthode de descente conjuguée produit toujours une direction de descente si les conditions de Wolfe fortes sont satisfaites.

Nous essayons de trouver des méthodes de gradient conjugué qui produisent des directions de descente avec seulement des conditions de **Wolfe faibles.** C'est l'objet du paragraphe suivant :

Cette méthode a été découverte par Dai et Yuan (9, 1999),

4.3.1 Description de la méthode

On suppose que la direction recherche d_k est une direction de descente, C'est à dire qu'on a

$$d_k^T g_k < 0$$

Maintenant nous avons besoin de trouver un coefficient β_k qui définit une direction de descente d_{k+1}. Cela exige que :

$$g_{k+1}^T d_{k+1}^T < 0$$

Soit en remplaçant l'expression de d_{k+1}, on obtient :

$$-\|g_{k+1}\|^2 + \beta_k g_{k+1}^T d_k < 0 \tag{4.13}$$

Supposons que $\beta_k > 0$, et notons

$$r_k = \|g_{k+1}\|^2 / \beta_K$$

L'inégalité (4.13) devient

$$-r_k + g_{k+1}^T d_k < 0 \Leftrightarrow r_k > g_{k+1}^T d_k \tag{4.13bis}$$

Si on prend

$$r_k = d_k^T y_k = d_k^T (g_{k+1} - g_k) = d_k^T g_{k+1} - d_k^T g_k = g_{k+1}^T d_k + \tau, \quad \tau > 0.$$

Il est clair que ce choix implique que

$$r_k > g_{k+1}^T d_k.$$

Si on substitue l'expression de $r_k = d_k^T y_k$ dans $r_k = \|g_{k+1}\|^2 / \beta_K$, on obtient :

$$\beta_k = \frac{\|g_{k+1}\|^2}{d_k^T y_k} \tag{4.14}$$

De (4.3) et (4.14), on a

$$g_{k+1}^T d_{k+1} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} g_k^T d_k = \beta_k g_k^T d_k. \tag{4.15}$$

Ce qui donne

$$\beta_k = \frac{g_{k+1}^T d_{k+1}}{g_k^T d_k} \tag{4.16}$$

4.3.2 Algorithme de Dai et Yuan

Algorithme 4.1 (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"

Si non aller à l'étape 2.

Etape 2 :

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

λ_k vérifie les conditions de **Wolfe faibles**

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

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. $\square$

4.3.3 La propriété de descente de la méthode de Dai-Yuan

Dai et Yuan [13, 1998] ont démontré qu'à chaque itération $k \geq 1$, la direction d_k définie par les relations (4.3) et (4.14) et utilisant une recherche linéaire de Wolfe faible (4.7)-(4.8) est de descente, si la fonction objectif f est strictement convexe.

Dai et Yuan [9,1999] ont généralisé ce résultat pour toute fonction régulière.

Théorème 4.2. *Supposons que la supposition 4.1 soit satisfaite. Pour toute méthode du type (4.2) et (4.3) dont β_k satisfait à (4.14) et le pas λ_k satisfait aux conditions de **Wolfe faibles** :*

$$\begin{aligned} f(x_k + \lambda_k d_k) &\leq f(x_k) + \rho \lambda_k d_k^T g_k \\ \text{et} \quad d_k^T g_{k+1} &\geq \sigma d_k^T g_k \end{aligned}$$

Où $0 < \rho < \sigma < 1$. Alors les directions générées par ces méthodes sont des directions de descente, autrement dit :

$$d_k^T g_k < 0; \quad \forall k \geq 1 \tag{4.17}$$

Démonstration. [9,1999]

La démonstration se fait par récurrence.

1- Pour $k = 1$:

$$d_1^T g_1 = -\|g_1\|^2 < 0$$

2- Supposons que (4.17) est satisfaite pour $k > 1$ et démontrons qu'elle le sera pour $k + 1$:

Supposons que :

$$d_k^T g_k < 0; \quad k > 1$$

En utilisant (4.8), on aura :

$$d_k^T y_k = d_k^T (g_{k+1} - g_k) \geq (\sigma - 1) d_k^T g_k = -(1 - \sigma) d_k^T g_k > 0$$

D'autre part :

$$\begin{aligned}
d_{k+1}^T g_{k+1} &= \left(-g_{k+1} + \beta_{k+1}^{DY} d_k\right)^T g_{k+1} \\
&= -\|g_{k+1}\|^2 + \beta_{k+1}^{DY} d_k^T g_{k+1} \\
&= -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2}{d_k^T y_k} d_k^T g_{k+1} \\
&= -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2}{d_k^T y_k} d_k^T (y_k + g_k) \\
&= -\|g_{k+1}\|^2 + \|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2}{d_k^T y_k} d_k^T g_k \\
&= \frac{\|g_{k+1}\|^2}{d_k^T y_k} d_k^T g_k
\end{aligned}$$

or puisque $d_k^T g_k < 0$; $d_k^T y_k > 0$; il en résulte :

$$d_{k+1}^T g_{k+1} < 0$$

Ce qui achève la démonstration. □

4.3.4 Convergence de la méthode de Dai-Yuan

Dai-Yuan [9, 1999] ont démontré la convergence de la méthode de Dai-Yuan au sens $\lim\limits_{k\to\infty} \inf \|g_k\| = 0$ si le pas λ_k est déterminé par la règle de **Wolfe faible** et la fonction objectif satisfait les conditions citées dans la supposition 4.1.

Théorème 4.3. *Soit x_1 un point de départ pour lequel la supposition 4.1soit satisfaite.*

La suite $\{x_k\}$ générée par l'algorithme 4.1 converge dans le sens

$$\lim_{k\to\infty} \inf \|g_k\| = 0 \tag{4.18}$$

Démonstration. [9,1999]

Supposons que l'algorithme ne se termine pas après un nombre fini d'itération. nous aurons alors

$$\|g_k\| \neq 0 \quad \forall k.$$

Puisque

$$\|g_k\| \geq 0 \quad \forall k.$$

Alors

$$\|g_k\| > 0 \quad \forall k$$

-On montre tout d'abord que toutes les directions de recherche sont de descente c'est-à-dire

$$d_k^T g_k < 0, \quad \forall k \tag{4.19}$$

On démontrera (4.19) par récurence. En effet l'inégalité (4.19) est évidente pour $k = 1$

Supposons que (4.19) soit vraie pour k, on a d'après (4.8)

$$d_k^T y_k \geq (\sigma - 1)\, d_k^T g_k > 0 \tag{4.20}$$

(4.20) et (4.15) impliquent que (4.19) est vérifiée pour $k+1$. Donc (4.19) est vraie $\forall k$.

D'autre part on a :

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

Alors

$$\|d_{k+1} + g_{k+1}\|^2 = \left\|\beta_{k+1}^{DY} d_k\right\|^2$$

$$\Rightarrow \|d_{k+1}\|^2 = \left(\beta_{k+1}^{DY}\right)^2 \|d_k\|^2 - 2d_{k+1}^T g_{k+1} - \|g_{k+1}\|^2 \tag{4.21}$$

On divise les deux côtés par $(d_{k+1}^T g_{k+1})^2$:

$$\frac{\|d_{k+1}\|^2}{(d_{k+1}^T g_{k+1})^2} = \frac{\left(\beta_{k+1}^{DY}\right)^2 \|d_k\|^2}{(d_{k+1}^T g_{k+1})^2} - \frac{2d_{k+1}^T g_{k+1}}{(d_{k+1}^T g_{k+1})^2} - \frac{\|g_{k+1}\|^2}{(d_{k+1}^T g_{k+1})^2}$$

De (4.3) :

$$\begin{aligned} d_{k+1}^T g_{k+1} &= \left(-g_{k+1} + \beta_{k+1}^{DY} d_k\right)^T g_{k+1} \\ &= \frac{\|g_{k+1}\|^2}{d_k^T y_k} d_k^T g_k = \beta_{k+1}^{DY} d_k^T g_k \\ &\Rightarrow \ \beta_{k+1}^{DY} = \frac{d_{k+1}^T g_{k+1}}{d_k^T g_k} \end{aligned}$$

Remplaçant ceci dans (4.21), on aura :

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

D'où

$$\begin{aligned}
\frac{\|d_k\|^2}{(d_k^T g_k)^2} &\leq \frac{1}{\|g_k\|^2} + \frac{\|d_{k-1}\|^2}{\left(d_{k-1}^T g_{k-1}\right)} \\
&\leq \frac{1}{\|g_k\|^2} + \frac{1}{\|g_{k-1}\|^2} + \frac{\|d_{k-2}\|^2}{\left(d_{k-2}^T g_{k-2}\right)} \\
&\ \ \vdots \\
&\leq \sum_{i=1}^{k} \frac{1}{\|g_i\|^2} 4.22
\end{aligned} \tag{4.6}$$

Supposons maintenant que (4.18) n'est pas satisfaite, autrement dit :

$$\exists \omega > 0 \ \text{ tel que } \ \|g_k\| > \omega; \forall k \tag{4.23}$$

D'après (4.22) et (4.23) on a

$$\begin{aligned}
\frac{\|d_k\|^2}{(d_k^T g_k)^2} &\leq \frac{1}{\|g_i\|^2} \leq \frac{1}{\omega^2}\sum_{i=1}^{k} 1 = \frac{1}{\omega^2}k \\
&\Rightarrow \sum_{k\geq 1} \frac{\left(d_k^T g_k\right)^2}{\|d_k\|^2} \geq \omega^2 \sum_{k\geq 1} \frac{1}{k}
\end{aligned}$$

D'où

$$\sum_{k\geq 1} \frac{\left(d_k^T g_k\right)^2}{\|d_k\|^2} = \infty$$

Ce qui contredit la condition de Zoudentijk (4.11)
Ce qui achève la démonstration. □

Chapitre 5

Une modification de l'algorithme Gradient conjugué version Dai-Yuan assurant la descente suffisante

5.1 Position du problème

Soit $f : \mathbb{R}^n \to \mathbb{R}$ et (P) le problème d'optimisation sans contraintes suivant :

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

Dans le chapitre précédent nous avons vu que Dai et Yuan ([ ]) ont proposé la méthode du gradient conjugué suivante. On génère une suite $\{x_k\}_{k\in\mathbb{N}}$ de la manière suivante :

$$x_{k+1} = x_k + \alpha_k d_k \tag{5.1}$$

où la direction de recherhe est définie par la formule de récurrence suivante :

$$d_k = \begin{cases} -g_0 & \text{si } k = 0 \\ -g_k + \beta_k^{DY} d_{k-1} & \text{si } k \geq 1 \end{cases} \tag{5.2}$$

avec

$$\beta_k^{DY} = \frac{g_{k+1}^T \cdot g_{k+1}}{y_k^T s_k} \tag{5.3}$$

où

$$g_k = \nabla f(x_k), \quad y_k = g_{k+1} - g_k, \quad s_k = x_{k+1} - x_k \tag{5.4}$$

La propriété fondamentale de cette méthode est qu'elle génère des directions de descente en utilisant seulement la recherche de wolfe alors que les autres méthodes du gradient conjugué classiques (Fletcher Reeves et d'autres) utilisent la recherche de wolfe forte.

A la fin du chapitre 4 nous avons posé un problème ouvert (problème ouvert1) : Les directions générées par la méthode du gradient conjugué version Dai-Yuan sont elles aussi des directions de descente suffisante, c'est à dire qu'elles vérifient la relation suivante :

$$g_k^T d_k \leq -c \left\| g_k \right\|^2, \quad \forall k, \quad c > 0 \tag{5.5}$$

On n'a pas réussi à démontrer celà.

En 2008, Neculai Andrei ([A Dai-Yuan conjugate...]) proposa une modification de la méthode de Day-Yuan et démontre que les directions engendrées sont des directions de descente suffisante.

5.2 Gradient conjugué version Dai-yuan modifiée

Andrei ([A Dai-Yuan conjugate...]) proposa la suite $\{d_k\}_{k\in\mathbb{N}}$ comme suit : $d_0 = -g_0$ et

$$d_{k+1} = -\theta_{k+1}.g_{k+1} + \beta_k^a s_k, \quad k = 0, 1, 2, ... \tag{5.6}$$

avec

$$\theta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{y_k^T \cdot g_{k+1}} \tag{5.7}$$

et

$$\beta_k^a = \frac{1}{y_k^T s_k} \left(g_{k+1} - \delta_k \frac{\left\| g_{k+1} \right\|^2}{y_k^T s_k} s_k \right)^T g_{k+1} \tag{5.8}$$

et

$$\delta_k = \frac{y_k^T \cdot g_{k+1}}{g_{k+1}^T g_{k+1}} \tag{5.9}$$

Théorème5.1 *Si f est quadratique et α_k est obtenu par une recherche linéaire exacte, alors*

$$\beta_k^a = \beta_k^{DY} \tag{5.10}$$

preuve du théorème 5.1

Si f est quadratique et α_k est obtenu par une recherche linéaire exacte, alors

$$s_k^T g_{k+1} = 0. \tag{5.11}$$

Par conséquent

$$\begin{aligned} \beta_k^a &= \frac{1}{y_k^T s_k}(g_{k+1}^T g_{k+1} - \delta_k \frac{\|g_{k+1}\|^2}{y_k^T s_k} s_k^T g_{k+1}) \\ &= \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} = \beta_k^{DY} \quad \blacksquare \end{aligned}$$

Dans ce travail on considère des fonctions f non linéaires, non quadratiques et α_k est obtenu par une recherche linéaire inexacte de type wolfe. Avant de donner l'algorithme de la nouvelle méthode, démontrons le théorème suivant :

Théorème5.2 ([N.Andrei]) *Si $y_k^T s_k \neq 0$ et $d_{k+1} = -\theta_{k+1}.g_{k+1} + \beta_k^a s_k (d_0 = -g_0)$, avec β_k^a donnée par la relation (5.8), alors on a :*

$$g_{k+1}^T d_{k+1} \leq -\left(\theta_{k+1} - \frac{1}{4\delta_k}\right) \|g_{k+1}\|^2 . \tag{5.12}$$

Preuve du Théorème5.2

Rappelons que

$$d_0 = -g_0, \tag{5.13}$$

donc

$$g_0^T d_0 = -\|g_0\|^2 . \tag{5.14}$$

Donc (5.12) est vérifiée pour $k = 0$.

Multiplions maintenant (5.6) par g_{k+1}^T, on obtient

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\theta_{k+1}\|g_{k+1}\|^2 + \frac{\left(g_{k+1}^T g_{k+1}^T\right)\left(g_{k+1}^T s_k\right)}{y_k^T s_k} 5.15 \\ & \quad -\delta_k \frac{\|g_{k+1}\|^2 \left(s_k^T g_{k+1}\right)^2}{\left(y_k^T s_k\right)^2}. \end{aligned} \tag{5.1}$$

Remarquons maintenant que :

$$\begin{aligned}
\frac{\left(g_{k+1}^T g_{k+1}^T\right)\left(g_{k+1}^T s_k\right)}{y_k^T s_k} &= \frac{\left[\frac{\left(y_k^T s_k\right) g_{k+1}}{\sqrt{2\delta_k}}\right]^T \left[\sqrt{2\delta_k}\left(g_{k+1}^T s_k\right) g_{k+1}\right]}{\left(y_k^T s_k\right)^2} 5.16 \qquad (5.2)\\
&\leq \frac{\frac{1}{2}\left[\frac{1}{2\delta_k}\left(y_k^T s_k\right)^2 \|g_{k+1}\|^2 + 2\delta_k \left(g_{k+1}^T s_k\right)^2 \|g_{k+1}\|^2\right]}{\left(y_k^T s_k\right)^2}\\
&= \frac{1}{4\delta_k}\|g_{k+1}\|^2 + \delta_k \frac{\left(g_{k+1}^T s_k\right)^2 \|g_{k+1}\|^2}{\left(y_k^T s_k\right)^2}.
\end{aligned}$$

Considérons maintenant (5.15) en prenant en considération (5.16), on obtient (5.12).

Remarque 5.1
La direction donnée par (5.6), (5.7), (5.8), (5.9) est une direction de descente. En effet la relation (5.12) implique

$$g_k^T d_k < 0 \quad \forall k \tag{5.17}$$

Remarque 5.2
Dai et Yuan ([9]) ont présenté une méthode dans laquelle $g_k^T d_k < 0$ si la condition $y_k^T s_k > 0$.

Remarque 5.3
Si f est fortement convexe ou si la recherche linéaire est de Wolfe alors $y_k^T s_k > 0$.

Théorème 5.3 ([N. Andrei]) *Considérons l'algorithme du gradient conjugué défini par les relations (5.1), (5.2), (5.6), (5.7), (5.8), (5.9) et α_k est obtenu par une recherche linéaire inexacte de Wolfe. Si pour tout k on a*

$$\theta_{k+1} > 0 \quad et \quad \theta_{k+1} > \frac{1}{4\delta_k}. \tag{5.18}$$

Alors les directions d_k engendrées par cette méthode sont des directions de descente suffisante, i.e.

$$\exists c > 0 : \ \forall k \in \mathbb{N} \quad g_k^T d_k \leq -c\|g_k\|^2. \tag{5.19}$$

Preuve du Théorème 5.3
voir ([N.Andrei, A Dai-Yuan...])

5.3 L'Algorithme du Gradient conjugué version Dai-yuan modifiée

Notons le nouveau algorithme par la notation GCDS (gradient conjugué à descente suffisante).

Algorithme GCDS

Etape1 : Initialisation

Choisir $x_0 \in \mathbb{R}^n$ et les paramètres $0 < \sigma_1 < \sigma_2 < 1$. Calculez $f(x_0)$ et $g_0 = \nabla f(x_0)$.

Poser $d_0 = -g_0$, $\alpha_0 = \frac{1}{\|g_0\|}$, $\quad k = 0$

Etape2 : Test de continuation des iterations

Si $\|g_k\| \leq 10^{-6}$, Stop. Sinon poser $k = k + 1$

Etape3 : Recherche linéaire

Calculez α_k vérifiant les conditions de Wolfe suivantes :

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \sigma_1 \alpha_k d_k^T g_k \\ d_k^T g_{k+1} &\geq \sigma_2 d_k^T g_k \end{aligned}$$

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

Calculer : $f(x_{k+1}),\ g_{k+1},\ s_k = x_{k+1} - x_k,\ \ y_k = g_{k+1} - g_k$

Etape : Calcul des directions

Calculez

$$d = -\theta_{k+1}.g_{k+1} + \beta_k^a s_k, \quad k = 0, 1, 2, ...$$

avec

$$\theta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{y_k^T . g_{k+1}}$$

et

$$\beta_k^a = \frac{1}{y_k^T s_k} \left(g_{k+1} - \delta_k \frac{\|g_{k+1}\|^2}{y_k^T s_k} s_k \right)^T g_{k+1}$$

et

$$\delta_k = \frac{y_k^T . g_{k+1}}{g_{k+1}^T g_{k+1}}$$

Si

$$g_{k+1}^T d \leq -10^{-3} \|d\| \|g_{k+1}\|$$

alors posons $d_{k+1} = d$. Sinon poser $d_{k+1} = -g_{k+1}$. Calculer $\alpha_k = \alpha_{k-1}\frac{\|d_{k-1}\|}{\|d_k\|}$. Poser $k = k+1$ et aller à Etape2

5.4 Analyse de la convergence

Théorème 5.4 ([N. Andrei]) *Supposons qu'il existe deux constantes positives ω et ρ telles que*

$$0 < \omega < \theta_k < \rho : \quad \forall k \in \mathbb{N} \tag{5.20}$$

Supposons aussi que l'ensemble

$$L_{x_0} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$$

est borné et que la condition de Lipschitz soit vérifiée dans l'ensemble L_{x_0}, c'est à dire qu'on a :

$$\forall (x, y) \in L_{x_0}^2 : \quad \|\nabla f(x) - \nabla f(y)\| \leq \mathcal{L}\|x - y\|. \tag{5.21}$$

Alors l'algorithme du gradient conjugué défini par les relations (5.1), (5.2), (5.6), (5.7), (5.8), (5.9) et α_k est obtenu par une recherche linéaire inexacte de Wolfe, on a $g_{k_0} = 0$ pour un certain $k_0 \in \mathbb{N}$ ou bien

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{5.22}$$

Preuve du Théorème 5.4 ([N. Andrei])
Supposons que $g_k \neq 0$ pour tout k et que

$$\liminf_{k\to\infty} \|g_k\| > 0.$$

Definissons

$$\gamma = \inf(\|g_k\| : k \geq 0).$$

Puisque $g_k \neq 0$, alors $\gamma > 0$. La condition de Wolfe implique

$$y_k^T s_k = (g_{k+1} - g_k)^T s_k \geq (\sigma_2 - 1)\, g_k^T s_k = -(1 - \sigma_2)\, g_k^T s_k.$$

Le théorème 5.2 donne

$$g_k^T d_k \leq -\left(\theta_k - \frac{1}{4\delta_{k-1}}\right)\|g_k\|^2 = -\frac{3}{4}\omega\|g_k\|^2.$$

Par conséquent

$$-g_k \geq \frac{3}{4}\omega\gamma^2.$$

(21) et (23) donnent

$$y_k^T s_k \geq \frac{3}{4}\left(1-\delta^2\right)\omega\alpha_k\gamma^2.$$

Remarquons que $g_{k+1}^T s_k = y_k^T s_k + g_k^T < y_k^T s_k$. Donc et en tenant compte de (17) on a

$$g_{k+1}^T s_k \geq \delta^2 g_k^T = -\delta^2 y_k^T s_k + \delta^2 g_{k+1}^T s_k.$$

Puisque $\delta^2 < 1$, alors on a

$$g_{k+1}^T s_k \geq \frac{-\delta^2}{1-\delta^2} y_k^T s_k.$$

Par conséquent,

$$\left|\frac{g_{k+1}^T s_k}{y_k^T s_k}\right| \leq \max\left\{1, \frac{-\delta^2}{1-\delta^2}\right\}.$$

D'autre part

$$\|y_k\| = \|g_{k+1} - g_k\| \leq L\|s_k\|.$$

Donc

$$|\sigma_k| = \frac{\left|y_k^T g_{k+1}\right|}{\|g_{k+1}\|^2} \leq \left\|\frac{y_k}{\|g_{k+1}\|}\right\| \leq \frac{L\|s_k\|}{\|g_{k+1}\|}.$$

Avec celà on obtient

$$\left|\beta_k^a\right| \leq \frac{1}{\left|y_k^T s_k\right|}\left[\|g_{k+1}\|^2 + \frac{\left|\delta_{k+1 s_k}\right|}{\left|y_k^T s_k\right|}\right]$$

$$\leq \frac{4}{3\left(1-\sigma_2\right)\omega\alpha_k\gamma^2}\left[\Gamma^2 + L\Gamma\|s_k\| \max\left\{1, \frac{\sigma_2}{1-\sigma_2}\right\}\right] = E + F\|s_k\| = E + FD,$$

où

$$E = \frac{4\Gamma^2}{3\left(1-\sigma_2\right)\omega\alpha_k\gamma^2}, F = \frac{4L\Gamma}{3\left(1-\sigma_2\right)\omega\alpha_k\gamma^2} \max\left\{1, \frac{\sigma_2}{1-\sigma_2}\right\},$$

$D = \max\{\|y-z\| : y, z \in L\}$ est le diamètre de l'ensemble L et $\Gamma = \max_{x\in L}\|\nabla f(x)\|$.

Donc,

$$\|d_{k+1}\| \leq |\theta_{k+1}| \, \|g_{k+1}\| + |\beta_k^{\alpha}| \, \|s_k\| \leq \Omega\Gamma + (E + FD)\, D.$$

Si on tient compte de la condition de Lipschitz et des conditions de Wolfe on peut prouver que

$$\alpha_k \geq \frac{1-\sigma_2}{L} \frac{\left|g_k^T\right|}{\|d_k\|^2}.$$

Puisque l'ensemble L est borné et puisque la fonction f est aussi bornée inferieurement, donc en considérant (17) et (21) on a :

$$\sum_{k=0}^{\infty} \frac{\left(g_k^T d_k\right)}{\|d_k\|^2} < \infty.$$

Par conséquent, utilisant (22), the propriété de descente yields

$$\sum_{k=0}^{\infty} \frac{\gamma^4}{\|d_k\|^2} \leq \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \leq \sum_{k=0}^{\infty} \frac{16}{9\omega^2} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < \infty,$$

qui contredit (26). Donc,

$$\gamma = \lim \inf_{k\to\infty} \|g_k\| = 0.\blacksquare$$

5.5 Résultats numériques et comparaisons

5.5.1 Critère de performance de Dolan et Moré

On a fait des tests numériques de la nouvelle méthode GCDS sur un échantillon de 750 problèmes tests. On compare le nouveau algorithme avec l'algorithme du gradient conjugué version Dai-Yuan. Nous avons utilisé des recherches linéaires de Wolfe avec $\sigma_1 = 0.0001$ et $\sigma_2 = 0.9$. Le critère d'arret est le suivant : $\|g_k\| \leq 10^{-6}$. Le graphe suivant appelé aussi critère de performance de Dolan et Moré, montre la performance de la nouvelle méthode GCDS par rapport à l'algorithme du gradient conjugué version Dai-Yuan.

5.5.2 Conclusions

Soit $f : \mathbb{R}^n \to \mathbb{R}$ et (P) le problème d'optimisation sans contraintes suivant :

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

Nous avons présenté dans ce mémoire deux algorithmes du gradient conjugué pour résoudre le problème (P). La première méthode est due à Dai et Yuan ([ ]). La seconde méthode est une modification de la méthode de Dai et Yuan et a été proposée par Niculai Andrei en 2008 ([ ]). Les deux méthodes génèrent une suite $\{x_k\}_{k\in\mathbb{N}}$ de la manière suivante :

$$x_{k+1} = x_k + \alpha_k d_k \tag{5.1}$$

où la direction de recherhe est définie par la formule de récurrence suivante :

$$d_k = \begin{cases} -g_0 & \text{si } k = 0 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 1 \end{cases} \tag{5.2}$$

Les deux méthodes different par les coéfficients β_k. Dai et Yuan ont proposé les coéfficients suivants :

$$\beta_k^{DY} = \frac{g_{k+1}^T \cdot g_{k+1}}{y_k^T s_k}$$

Les directions d_k et les coéfficients β_k proposés par Niculai Andrei ont la forme suivante :

$$d_0 = -g_0, \quad d_{k+1} = -\theta_{k+1}.g_{k+1} + \beta_k^a s_k, \quad k = 0, 1, 2, ...$$

avec

$$\theta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{y_k^T \cdot g_{k+1}} \tag{5.7}$$

et

$$\beta_k^a = \frac{1}{y_k^T s_k} \left(g_{k+1} - \delta_k \frac{\|g_{k+1}\|^2}{y_k^T s_k} s_k \right)^T g_{k+1} \tag{5.8}$$

et

$$\delta_k = \frac{y_k^T \cdot g_{k+1}}{g_{k+1}^T g_{k+1}} \tag{5.9}$$

La suprémacie de la méthode de Niculai Andrei est qu'elle assure que les directions d_k sont des directions de descente suffisante, alors que les directions engendrées par Dai et Yuan sont des directions de descente seulement. Les tests numériques ont prouvé que la méthode de Niculai Andrei est plus performante que celle de Dai et Yuan. La méthode de Niculai Andrei converge globlement avec la recherche linéaire inexacte de Wolfe.

5.5.3 Problème ouvert

Supposons que dans la méthode de Niculai Andrei, on remplace les recherches linéaires inexactes de Wolfe par les recherches linéaires inexactes de Goldstein. Les directions d_k sont elles des des directions de descente suffisante ? La méthode de Niculai Andrei converge elle globlement ?

Bibliographie

[1] **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.*

[2] **LP. Armand (2005)**, Modification of the Wolfe Line Search Rules to Satisfy the Descent Condition in the Polak-Ribière-Polyak Conjugate Gradient Method, *Journal of Optimization Theory and Applications, Vol. 131, pp. 103-115.*

[3] **L. Armijo (1966)**, Minimzation of function having lipschitz continous first partial derivatives, *Pacific Journal of Mathematics, Vol. 16(1), pp.1-3.*

[4] **M. S. Bazaraa, H. D. Sherali, et C. M. Shetty (1993)**, Nonlinear Programming, Theory and Algorithms, *Wiley-Interscience.*

[5] **M. Barrault et C.Le. Bris (1999)**, Optimisation Numérique et Différentiation Automatique pour un Problème Industriel, *Notes de cours, Ecole Nationale de Ponts et Chaussées, Paris.*

[6] **M. Bergounioux (2001)**, Optimisation et Contrôle des Systèmes Linéaires, *Dunod.*

[7] **C. W. Carroll (1961)**, The created reponse surface technique for optimizing nonlinear restrained systems, *Operations Res.*, Vol. 9, pp. 169-84.

[8] **A. Cauchy (1847)**, Analyse mathématique, Méthode générale pour la résolution des systèmes d'équations simultanées, *Comptes Rendus de l'Académie des Sciences de Paris, t-25, pp. 536-538.*

[9] **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.*

[10] **Y. H. Dai and Y. Yuan (1996)**, Convergence properties of the Fletche-Reeves method, *IMA J Numer. Anal., Vol.16(2), pp. 155-164.*

[11] **Y. H. Dai and Y. Yuan (1996)**, Convergence properties of the conjugate descent method, *Advances in Mathematics, 6, pp.552-562.*Y.

[12] **H. Dai and Y. Yuan (2001)**, New properties of a new conjugate gradient method, *Numer. Math., Vol. 89(2), pp. 83-98.*

[13] **Y. H. Dai and Y. Yuan (1998)**, Some properties of a new conjugate gradient method, in : *Advances in Nonlinear Programming, ed . Kluwer Academic, Boston, pp. 251-262.*

[14] **Y. H. Dai and Y. Yuan (2002)**, A note on the nonlinear conjugate gradient methode Engineering Computing, *Chinese Academy of Sciences, P. O. Box 2719,Beijing 100080, China)*

[15] **Y. H. Dai and Y. Yuan (2001)**, An Efficient Hybrid Conjugate Gradient Method for Unconstrained Optimization, *Chinese Academyof Sciences, P.O. Box 2719, Beijing 100080, P.R. China*

[16] **T. Diallo (2006)**, Travail pratique de master, Sujet : Etude et Illustration De Méthodes Itératives D'optimisation Non Linéaire, *Responsable : Benjamin Leroy-Beaubeir, Ecole polytechnique Fédérale de Lausanne.*

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

[18] **R. Fletcher and M. J. D. Powell (1963)**, A rapidly convergent descent method for minimization, *Computer. J., 6, pp. 163-168.*

[19] **R. Fletcher and C. Reeves (1964)**, Function minimization by conjugate gradients. *Comput. J., 7, pp.149-154.*

[20] **A. V. Fiacco and G. P. McCormick (1968)**, Nonlinear Programming, *John Wiley, New York.*

[21] **J. C. Gilbert (2007)**, Eléments d'Optimisation Différentiable : Théorie et Algorithmes, *Notes de cours, École Nationale Supérieure de Techniques Avancées, Paris.*

[22] **J.C. Gilbert and J. Nocedal (1992)**, Global convergence properties of conjugate gradient methods for optimization, *SIAM J. Optimization. Vol. 3, No.1, pp. 21-42.*

[23] **A.A. Goldstein (1967)**, Constructive Real Analysis*, A Harper International Edition.*

[24] **A. A. Goldstein (1965)**, On steepest descent, *SIAM J. on Control A, Vol. 3, No. 1, pp. 147-151.*

[25] **A.A. Goldstein and J.F. Price (1969)**, An effective algorithm for minimzation, *Num.Math., 10, pp. 184-189.*

[26] **L. Grippo and S. Lucidi (1997)**, A Globally Convergent Version of the Polak-Ribière Conjugate Gradient Method, *Mathematical Programming, Vol. 78, pp. 375–391.*

[27] **W.W. Hager et H. Zhang (2005)**, A new conjugate gradient method with guaranteed descent and an efficient line search, *SIAM J. Optim., 16,* pp. *170-192.*

[28] **M. R. Hestenes and E.L. Stiefel (1952)**, Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standars Sect., 5(49),*
pp. 409-436.

[29] **Y.F. Hu and C. Story (1991)**, Global convergence result for conjugate gradient methods, *JOTA, 71(2), pp. 399-405.*

[30] **Z. Li, J. Chen and N. Deng (1998),** A New Conjugate Gradient Method and its Global Convergence Properties, *Systems Science and Mathematical Sciences, Vol. 11, pp. 53–60.*

[31] **Laetitia ANDRIEU (2004),**Optimisation sous contrainte en probabilité, THESE présentée pour l'obtention du titre de Docteur *de l'Ecole Nationale des Ponts et Chaussées, décembre*

[32] **G.H. Liu and J.Y. Han and H.X. Yin (1995)**, Global convergence of the Fletcher-Reeves algorithm with an inexact line search, *Appl. math. J. Chinese Univ. Ser. B, 10, pp. 75-82.*

[33] **D. G. Luenberger (1969),** Optimization by Vector Space Methods, *John Wiley and Sons, Inc. New York.*

[34] **M. Minoux (1983)**, Programmation Mathématique, Théorie et Algorithmes, tome 1 , *Dunod.*

[35] **J.J.Moré, B.S.Garbow and K.E. Hillstrom (1981)** , Testing unconstrained optimization software, ACMTransactions on Mathematical Software 7 ,17–41.

[36] **A.S. Nemirovsky et D.B. Yudin (1983),** Problem Complexity and Method Efficiency, *New York, Wiley.*

[37] **J. Nocedal (1991)**, Theory of algorithm for unconstrained optimization, *Acta Numerica, pp. 199-242.*

[38] **E. Polak and G. Ribière (1969)**, Note sur la convergence de directions conjuguées, *Rev. Française Informat. Recherche Operationelle, 3e année, 16, pp. 35-43.*

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

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

[41] **M.J.D. Powell (1984)**, Non convex minimzation calculation and the conjugate gradient method, in : *Lecture Notes in Mathematics 1066 (Springer, Berlin), pp. 122-141.*

[42] **M.J.D. Powell (1971)**, On the convergence of the variable metric algorithm ; *J. Inst. Math. Appl, 7, pp. 21-36.*

[43] **M.J.D. Powell (1977)**, Restart procedures for the conjugate gradient method, *Math Programming, 2 , pp 241-254.*

[44] **D. Touati-Ahmed and C. Storey (1990)**, Efficient hybrid conjugate gradient techniques, *JOTA, 64, pp. 379-397.*

[45] **P. Wolfe (1961)**, A duality theorem for nonlinear programming, *Quart. Appl. Math.*, 19, pp. 239-244.

[46] **P. Wolfe (1969)**, Convergence conditions for ascent methods, *SIAM Review, 11, pp. 226-235.*

[47] **P.Wolfe (1971)**, Conditions for ascent methods some corrections 2, *SIAM Review, 13, pp.185-188.*

[48] **P. Wolfe (1963)**, Methods of Nonlinear Programming, *in Recent Advances in Mathematical Programming (Eds R. L. Graves and P. Wolfe), McGraw-Hill, New York.*

[49] **Y.X.Yuan (1993)**, Numerical Methods for Non linear Programming, *Shanghai scientific & Technical Publishers (in Chinese).*

[50] **G. Zoutendijk (1970),** Nonlinear Programming Computational Methods, *Integer and Nonlinear Programming, North Holland, Amsterdam.*

[51] **G. Zoutendijk (1960)**, Methods of Feasible Directions, *Elsevier, Amesterdam*