



**BADJI MOKHTAR-ANNABA  
UNIVERSITY  
UNIVERSITE BADJI MOKHTAR-  
ANNABA**



**جامعة باجي مختار  
- عنابة -**

**Faculté des Sciences**

**Année : 2019**

**Département de Mathématiques**

# **THÈSE**

Présenté en vue de l'obtention du diplôme de  
Doctorat en Sciences

***POLYNOMES POLAIRES ET GRADIENT CONJUGUE***

**Option  
Mathématiques appliquées**

**Par  
BOUMEDIENE AMINA**

**DIRECTEUR DE THÈSE : BENZINE RACHID** Prof ESTI. ANNABA

Devant le jury

<b>PRESIDENT :</b>	Djebabla ABdelhak	M.C. A	U.B.M. ANNABA
<b>EXAMINATEURS :</b>	Berkane Ahmed	M.C. A	U.B.M. ANNABA
	Salmi Abdelouahab	M.C. A	U.B.M. ANNABA
	Laskri Yamina	Prof	ESTI. ANNABA
	Belloufi Mohamed	M.C. A	U. Souk Ahras

---

# Gradient conjugué et polynômes polaires

Présentée par Boumediene Amina

Dirigé par :Prof.Benzine Rachid

ESTI Annaba

7 mars 2019

# TABLE DES MATIÈRES

<b>1 OPTIMISATION SANS CONTRAINTES</b>	<b>6</b>
1.1 Notions de base	6
1.2 Schémas général des algorithmes d'optimisation sans contraintes	8
1.2.1 Exemples de choix de directions de descente	8
1.2.2 Exemple de choix de pas $\alpha_k$	8
1.3 Conditions d'optimalité des problèmes d'optimisation sans contraintes	9
1.3.1 Condition nécessaire d'optimalité du premier ordre	9
1.3.2 Conditions nécessaires d'optimalité du second ordre	10
1.3.3 Conditions suffisantes d'optimalité du second ordre	10
1.4 Les modes de convergence	12
<b>2 OPTIMISATION DANS <math>\Re</math>. RECHERCHES LINEAIRES EXACTES</b>	<b>13</b>
2.1 Recherche linéaire	13
2.2 Objectifs à atteindre	14
2.3 Choix optimal du pas $\alpha_k$	15
2.4 Optimisation dans $\Re$ . Recherches Linéaires Exactes	16
2.4.1 L'intervalle d'incertitude	16
2.4.2 Schémas général des recherches linéaires exactes utilisant les inter- valles d'incertitude	16
2.4.3 La méthode progressive-retrogressive ou Forward-Backward Algo- rithm	17
2.4.4 Deux méthodes d'optimisation unidimensionnelle sans dérivées.	22
2.5 Avantages et Inconvénients des recherches linéaires exactes	25

<b>3</b>	<b>OPTIMISATION DANS <math>\Re</math>. RECHERCHES LINEAIRES INEXACTES</b>	<b>27</b>
3.1	Introduction et rappels	27
3.1.1	Le schéma des recherches linéaires inexactes	27
3.2	La règle d'Armijo	28
3.3	La règle de Goldstein et Price	30
3.4	La règle de Wolfe	31
3.5	La règle de Wolfe fort	32
3.6	La règle de Wolfe relaxée.	33
<b>4</b>	<b>GRADIENT CONJUGUE. CAS QUADRATIQUE</b>	<b>34</b>
4.1	Optimisation quadratique sans contraintes	34
4.1.1	Definition et théorèmes fondamentaux	34
4.1.2	Calcul du pas obtenu par une recherche linéaire exacte	35
4.2	Méthode des directions conjuguées	37
4.2.1	Définitions et propriétés générales	37
4.2.2	L'Algorithme des directions conjuguées	38
4.3	Méthode du gradient conjugué. Cas quadratique	40
4.3.1	Introduction	40
4.3.2	Algorithme du gradient conjugué. Cas quadratique	41
4.3.3	Propriétés du gradient conjugué quadratique	42
4.3.4	Convergence de l'Algorithme du gradient conjugué quadratique	42
<b>5</b>	<b>GRADIENT CONJUGUE NON QUADRATIQUE</b>	<b>44</b>
5.1	Introduction et différentes formes du gradient conjugué	44
<b>6</b>	<b>SYNTHESE DES RESULTATS DE CONVERGENCES DES METHODES</b>	<b>51</b>
	<b>DU GRADIENT CONJUGUE NON QUADRATIQUE</b>	<b>51</b>
6.1	Hypothèses C1 et C2 et théorème de Zoutendijk	51
6.2	Rôle joué par la direction de recherche initiale	54
6.2.1	Méthode de Dai-Yuan généralisée	68
6.3	Les méthodes où $g_{k+1}^T y_k$ figure dans le numérateur de $\beta_k$	69
6.3.1	Méthode de Liu et Storey LS	73
6.4	Méthodes hybrides et familles paramétriques du gradient conjugué	74
<b>7</b>	<b>LA NOUVELLE METHODE CGBBB</b>	<b>79</b>
7.1	Intrduction	79
7.2	Algorithme de méthode CGBBB	81
7.3	Propriété de la descente	81

7.4	Convergence globale de l'Algorithme CGBBB	82
7.5	Résultats numériques et comparaisons	84
<b>8</b>	<b>PROGRAMMES EN FORTRAN</b>	<b>86</b>
8.1	Programme en fortran 77 d' Armijo	86
8.2	Programme en fortran 77 de Goldschtien	87
8.3	Programme en fortran 77 de Wolfe	88
8.4	Programme en fortran 90 de la méthode du gradient conjugué	88

DÉDICACES

*À mes parents*

## REMERCIEMENTS

*Je tiens à adresser mes remerciements les plus chaleureux et ma profonde gratitude à mon encadreur Monsieur **R. Benzine**, professeur à l'ESTI Annaba pour m'avoir proposé le sujet de cette thèse. C'est grâce à sa grande disponibilité, ses conseils, ses orientations, et ses encouragements que j'ai pu mener à bien ce travail.*

*Mes remerciements vont également à..Monsieur **A. Djebabla**, M.C. A à l'université d'Annaba, pour avoir bien voulu me faire l'honneur d'accepter de présider le jury.*

*De même je remercie Mr **A. Berkane**. M.C. A Université Badji Mokhtar Annaba, Mr **A. Salmi**. M.C. A. Université Badji Mokhtar Annaba, Mme **Y. Laskri**, professeur à l'ESTI Annaba, Mr **M.Belloufi**. M.C. A Université de Souk Ahras, pour l'honneur qu'ils m'ont fait de bien vouloir accepter de faire partie du jury de cette thèse.*

*Enfin, je n'oublie pas de remercier toutes les personnes qui m'ont facilité la tâche et tous ceux que j'ai connu à l'institut de mathématiques qui ont rendu mes séjours au département agréables.*

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

$$(P) : \min \{f(x) : x \in \mathbb{R}^n\} \quad (1)$$

La méthode du gradient conjugué génère une suite  $\{x_k\}_{k \in \mathbb{N}}$  de la manière suivante :

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \dots, \quad (2)$$

$\alpha_k > 0$  est le pas qu'on calcule à chaque itération  $k$  en utilisant des recherches linéaires exactes ou inexactes. Les recherches linéaires exactes sont obtenues par la relation :

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k) \quad (3)$$

Les directions  $d_k$  sont définies par

$$d_k = \begin{cases} -g_k & \text{si } k = 0 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 1 \end{cases} \quad (4)$$

Belloufi et Benzine ([49]) ont proposé une nouvelle direction  $d_k^{BB}$  définie comme suit :

$$d_k^{BB} = \begin{cases} -\frac{g_k}{\|g_k\|^2} & \text{if } k = 0 \\ -\frac{1}{\|g_k\|^2} g_k + d_{k-1} & \text{if } k \geq 1 \end{cases} \quad (5)$$

Belloufi et Benzine ([49]) ont obtenu leur résultat de convergence en utilisant la recherche linéaire inexacte de Wolfe forte suivante :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \quad (6)$$

---

$$\left| g(x_k + \alpha_k d_k)^T d_k \right| \leq -\sigma g_k^T d_k \quad (7)$$

où  $\delta \in \left] 0, \frac{1}{2} \right[$  et  $\sigma \in ]\delta, 1[$ .

Dans ce travail, on propose un nouveau algorithme qu'on a appelé : Conjugate gradient Boumediene Benzine Belloufi et noté CGBBB. On démontre la convergence globale et on prouve qu'il est numériquement performant. Pour arriver à ce but, on utilise une recherche linéaire exacte (3).

**Mots clés :** Optimisation sans contraintes, gradient conjugué, algorithme, recherche linéaire inexacte, Règle d'Armijo, Règle de Wolfe (forte et faible), recherche linéaire exacte, convergence globale, Algorithme, Convergence globale, Recherche linéaire inexacte, , Méthode de Hestenes-Stiefel, Méthode de Fletcher-Reeves, Méthode de Polak-Ribière-Polyak, Méthode de la descente conjuguée, Méthode de Dai-Yuan, Méthode hybride du Gradient conjugué

Nonlinear conjugate gradient (CG) methods are widely used for solving large scale unconstrained optimization problems. Many studies have been devoted to develop and improve these methods. In this thesis, we aim to study the global convergence of the CGBBB conjugate gradient method with exact line search

**Key words :** Unconstrained optimization Conjugate gradient, Algorithm, Global convergence, Inexact line search, Armijo line search, Strong Wolfe line search, Weak Wolfe line search, global convergence, Hestenes-Stiefel Method, Fletcher-Reeves Method, Polak-Ribière-Polyak Method, Conjugate descent Method, Dai-Yuan Method , Hybrid Conjugate Gradient Method.

تعتبر طريقة التدرج المترافق من أهم الطرق المستعملة لحل مسائل الأمثلة بدون قيود ذات البعد الكبير.

قدمنا في هذه الرسالة خوارزمية جديدة لطريقة التدرج المترافق باستخدام خط بحث مضبوط، سميناها خوارزمية بومدين ، بن زين ، بلوفي. حققت هذه الطريقة شرط الانحدار الكافي كما أن شرط التقارب الشامل تم برهانه. دلت النتائج العددية على أن الطريقة الجديدة حققت نتائج عددية ذات كفاءة عالية بالمقارنة طرق عددية أخرى معروفة.

**كلمات مفتاحية:** التدرج المترافق، معامل التدرج المترافق، الخوارزمية، الانحدار الكافي ، التقارب الشامل، البحث الخطي الدقيق، طريقة فانتشر ريفز، طريقة بولاك ريبيار بولياك ، طريقة النزول المتعامد.

L'optimisation peut être définie comme la science qui détermine la meilleure solution à certains problèmes mathématiquement définis, qui sont souvent des modèles de problèmes concrets (économie, physique, chimie, biologie...).

L'optimisation recouvre l'étude des critères d'optimalité pour les différents problèmes, la détermination des méthodes algorithmiques des solutions, l'étude de la structure de telles méthodes et l'expérimentation de l'ordinateur avec ces méthodes avec des problèmes de la vie courante .

D'un point de vue mathématique, l'optimisation consiste à rechercher le *minimum* ou le *maximum* d'une fonction avec ou sans contraintes.

L'optimisation possède ses racines au 18ième siècle dans les travaux de :

- Taylor, Newton , Lagrange, qui ont élaboré les bases des développements limités.
- Cauchy ([9]) fut le premier à mettre en œuvre une méthode d'optimisation, méthode du pas de descente, pour la résolution de problèmes sans contraintes.

Il faut attendre le milieu du vingtième siècle, avec l'émergence des calculateurs et surtout la fin de la seconde guerre mondiale pour voir apparaître des avancées spectaculaires en termes de techniques d'optimisation. A noter, ces avancées ont été essentiellement obtenues en Grande Bretagne.

De nombreuses contributions apparaissent ensuite dans les années soixante. G.Zoutendijk ([20]), C. W. Carroll ([11]), P.Wolfe ([54]), R. Fletcher et M. J.D. Powell ([45]), A. A. Goldstein ([2]) et A.V.Fiacco et G.P. McCormick ([5]) pour la programmation non linéaire ainsi que E. Polak et G. Ribière([15]), B.T. Polyak ([8]).

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

$$(P) : \min \{f(x) : x \in \mathbb{R}^n\} \tag{8}$$

---

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 ([25]), pour la minimisation des 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 ([17]) (*méthode de Fletcher-Reeves*) puis en 1969 par Polak, Ribière ([15]) et Polyak ([8]) (*méthode de Polak-Ribière-Polyak*). Une autre variante a été étudiée en 1987 par Fletcher ([57]) (*Méthode de la descente conjuguée*).

Toutes ces méthodes génèrent une suite  $\{x_k\}_{k \in \mathbb{N}}$  de la façon suivante :

$$x_{k+1} = x_k + \alpha_k d_k \quad (9)$$

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

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

$$d_k = \begin{cases} -g_k & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases} \quad (10)$$

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

Les différentes valeurs attribuées à  $\beta_k$  définissent les différentes formes du gradient conjugué.

Si on note  $y_{k-1} = g_k - g_{k-1}$ ,  $s_k = x_{k+1} - x_k$  on obtient les variantes suivantes :

1- Gradient conjugué variante Hestenes - Stiefel(HS)[25]

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$$

2- Gradient conjugué variante Fletcher Reeves(FR)[17]

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$$

3- Gradient conjugué variante Polak-Ribière-Polyak(PRP)[15,8]

$$\beta_k^{PRP} = \frac{g_k^T y_k}{\|g_{k-1}\|^2}$$

---

4- Gradient conjugué variante descente – Fletcher (CD)[57]

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

5- Gradient conjugué variante Liu - Storey(LS)[80]

$$\beta_k^{LS} = -\frac{g_{k+1}^T y_k}{d_k^T g_k}$$

6- Gradient conjugué variante de *Dai-Yuan*(DY)[78]

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

7- Gradient conjugué variante de Dai-Liao (DL)[69]

$$\beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{y_k^T s_k}$$

8- Gradient conjugué variante Hager-Zhang(HZ)[63]

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

9- Gradient conjugué variante de Z. Wei [87]

$$\beta_k^* = \frac{g_k^T \left( g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1} \right)}{\|g_{k-1}\|^2}$$

10- Gradient conjugué variante Rivaie-Mustafa-Ismail-Leong(RMIL)[42]

$$\beta_k^{RMIL} = \frac{g_k^T (g_k - g_{k-1})}{\|d_{k-1}\|^2}$$

Rappelons que les premiers résultats de convergence de la méthode du gradient conjugué ont d'abord été établis avec des recherches linéaires inexactes de Wolfe fortes, c'est à dire que les pas  $\alpha_k$  dans (9), doivent vérifier aussi les deux relations suivantes :

$$f(x + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k \nabla f(x_k)^T d_k \quad (11)$$

---


$$|\nabla f(x + \alpha_k d_k)^T \cdot d_k| \leq -\sigma \nabla f(x_k)^T d_k \quad (12)$$

avec

$$0 < \rho < \sigma < 1$$

Le premier théorème démontrant la descente d'une méthode du gradient conjugué non linéaire fut établi par Albaali ([1]). Albaali a utilisé  $\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$  et la recherche linéaire inexacte de Wolfe forte (11), (12). avec  $\sigma \leq \frac{1}{2}$ .

Gilbert et Nocedal ([30]) ont généralisé ce résultat pour tout algorithme du gradient conjugué dont

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

Ces résultats furent par la suite généralisés en considérant des recherches linéaires inexactes du type Wolfe faibles, c'est à dire que les pas  $\alpha_k$  dans (9), doivent vérifier les deux relations plus faibles que (11), et (12), suivantes :

$$f(x + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k \nabla f(x_k)^T d_k \quad (13)$$

$$\nabla f(x + \alpha_k d_k)^T \cdot d_k \geq \sigma \nabla f(x_k)^T d_k \quad (14)$$

Dans ce cas, le principal résultat obtenu est celui de Dai et Yuan( [73]).

Si on considère les variantes Polak-Ribière-Polyak (PRP) et Hestenes et Steifel (HS) non linéaire du gradient conjugué, peu de résultats de convergence complets ont été établis. Grippo et Lucidi ([38]) et P. Armand (2005) ont suggéré des modifications dans le choix de  $\alpha_k$  afin d'établir le résultat de la convergence.

Belloufi et Benzine ([49]) ont proposé une nouvelle direction  $d_k^{BB}$  définie comme suit :

$$d_k^{BB} = \begin{cases} -\frac{g_k}{\|g_k\|^2} & \text{if } k = 0 \\ -\frac{1}{\|g_k\|^2} g_k + d_{k-1} & \text{if } k \geq 1 \end{cases} \quad (15)$$

Belloufi et Benzine ([49]) ont obtenu leur résultat de convergence en utilisant la recherche linéaire inexacte de Wolfe forte suivante :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \quad (16)$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k \quad (17)$$

---

où  $\delta \in \left]0, \frac{1}{2}\right[$  et  $\sigma \in ]\delta, 1[$ .

On propose dans cette thèse un nouveau algorithme qu'on a appelé : Conjugate gradient Boumediene Benzine Belloufi et noté CGBBB Algorithm. On démontre la descente et la convergence globale de notre nouveau algorithme.. Pour arriver à ce but, on utilise une recherche linéaire exacte du type (3).

Une étude numérique a été proposée en effectuant près de 700 tests numériques sur notre nouveau algorithme et en les comparant avec d'autres algorithmes assez connus, en utilisant la méthode de Dolan et Moré CPU performance. On montre que l'algorithme CGBBB est plus performant que l'algorithme Polak-Ribiere-Polyak (CGPRP) et l'algorithme de la descente conjuguée (CGDC).

Cette thèse contient une introduction et sept chapitres. On introduit dans le premier chapitre les notions de base. On y aborde quelques notions sur les problèmes de minimisation sans contraintes et leurs algorithmes ainsi que les conditions d'optimalité.

Le deuxième chapitre est consacré à l'étude des recherches linéaires exactes. Les recherches linéaires inexactes sont abordées au chapitre 3. On y trouve les recherches linéaires de Wolfe, de Goldstein, D'Armijo.

Le chapitre 4 est consacré à une introduction à la méthode du gradient conjugué pour les fonctions quadratiques strictement convexes.

La méthode du gradient conjugué pour les fonctions non quadratiques est exposée dans le chapitre 5. Le chapitre 6 contient une synthèse des résultats de convergence des différentes méthodes du gradient conjugué.

Le chapitre 7 contient le résultat original de cette thèse. On propose un nouveau algorithme qu'on a appelé : Conjugate gradient Boumediene Benzine Belloufi et noté CGBBB. On démontre la convergence globale et on prouve qu'il est numériquement performant. Pour arriver à ce but, on utilise une recherche linéaire exacte. Les résultats numériques exposés à la fin du paragraphe montrent que cette méthode est plus performante que d'autres méthodes du gradient conjugué assez connues notamment la méthode CGBB (conjugate gradient Belloufi Benzine) et CGPRP(conjugate gradient Polak Ribiere Poliak).

Dans le chapitre 8 on trouve les programmes Fortran des différentes méthodes exposées dans cette thèse.

# CHAPITRE 1

## OPTIMISATION SANS CONTRAINTES

### 1.1 Notions de base

**Définition 1** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , on appelle problème de minimisation sans contraintes le problème (  $P$  ) suivant :

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

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

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

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

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

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

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

**Remarque 1.1.1** Dans le cas où  $f$  est convexe, alors tout minimum local est aussi global. De plus si  $f$  est strictement convexe, alors tout minimum local devient non seulement global mais aussi unique.

---

## Direction de descente

**Définition 2** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\hat{x} \in \mathbb{R}^n$ ,  $d \in \mathbb{R}^n$  est dite direction de descente au point  $\hat{x}$  si et seulement si il existe  $\delta > 0$  tel que

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

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

**Théorème 1.1.1** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  continument différentiable au point  $\hat{x} \in \mathbb{R}^n$ . S'il existe un vecteur  $d$  tel que la dérivée directionnelle de  $f$  dans la direction  $d$  au point  $\hat{x}$  (notée  $f'(\hat{x}; d)$ ) est strictement inférieure à zéro

$$\nabla^T f(\hat{x})d \prec 0 \tag{1.1}$$

alors  $d$  est une direction de descente de  $f$  en  $\hat{x}$ .

**Preuve.**  $f$  est différentiable au point  $\hat{x}$ . donc

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

avec  $\Psi(\hat{x}, \alpha d) \rightarrow 0$  quand  $\alpha \rightarrow 0$ , ce qui implique que

$$\lim_{\alpha \rightarrow 0} \frac{f(\hat{x} + \alpha d) - f(\hat{x})}{\alpha} = \nabla^T f(\hat{x})d \prec 0.$$

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

$$\frac{f(\hat{x} + \alpha d) - f(\hat{x})}{\alpha} \prec 0, \quad \forall \alpha \in ]-\delta, \delta[ \tag{1.2}$$

La relation (1.2) est particulièrement vraie pour tout  $\alpha \in ]0, \delta[$ . On obtient  $f(\hat{x} + \alpha d) \prec f(\hat{x})$  pour tout  $\alpha \in ]0, \delta[$  ■

---

## 1.2 Schémas général des algorithmes d'optimisation sans contraintes

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

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

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

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

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

### 1.2.1 Exemples de choix 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.

**Remarque 1.2.1**  $d_k = -\nabla f(x_k)$  est une direction de descente, car :

$$\nabla f(x_k)^t \cdot d_k = -\nabla f(x_k)^t \cdot \nabla f(x_k) = -\|\nabla f(x_k)\|^2 < 0$$

✓ Si choisit  $d_k = -(H(x_k))^{-1} \cdot \nabla f(x_k)$ , on obtient la méthode de Newton

**Remarque 1.2.2** Si la matrice hessienne  $H(x_k)$  est définie positive alors :

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

est aussi une direction de descente.

### 1.2.2 Exemple de choix de pas $\alpha_k$

On choisit en général  $\alpha_k$  de façon optimale, c'est à dire que  $\alpha_k$  doit vérifier

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

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.

---

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

Etant donné un vecteur  $\hat{x}$ , nous souhaiterons être capable de déterminer si ce vecteur est un minimum local ou global de la fonction  $f$ . La propriété de différentiabilité continue de  $f$  fournit une première manière de caractériser une solution optimale. Énonçons tout d'abord un théorème, sur lequel s'appuiera le corollaire suivant pour établir une première condition nécessaire d'optimalité :

### 1.3.1 Condition nécessaire d'optimalité du premier ordre

Les conditions que nous allons donner sont des conditions différentielles qui portent sur la dérivée de la fonction à minimiser.

**Théorème 1.3.1** *Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  différentiable au point  $\hat{x} \in \mathbb{R}^n$ . Si  $\hat{x}$  est un minimum local de  $(P)$  alors*

$$\nabla f(\hat{x}) = 0 \tag{1.3}$$

**Preuve.** On démontre par l'absurde, on suppose que  $\nabla f(\hat{x}) \neq 0$ , si on suppose  $d = -\nabla f(\hat{x})$ , on obtient :

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

alors il existe  $\delta > 0$  tel que

$$f(\hat{x} + \alpha d) < f(\hat{x}), \quad \forall \alpha \in ]0, \delta[. \tag{1.4}$$

Ce qui donne une contradiction avec le fait que  $\hat{x}$  est un minimum local, d'où  $\nabla f(\hat{x}) = 0$ . ■

---

### 1.3.2 Conditions nécessaires d'optimalité du second ordre

**Théorème 1.3.2** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois différentiable au point  $\hat{x} \in \mathbb{R}^n$ . Si  $\hat{x}$  est un minimum local de  $(P)$  alors  $\nabla f(\hat{x}) = 0$  et la matrice hessienne de  $f$  au point  $\hat{x}$ , qu'on note  $H(\hat{x})$ , est semi définie positive.

**Preuve.** Le théorème (1.3.1) montre la première proposition, pour la deuxième proposition on a :

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

$$f(\hat{x} + \alpha d) = f(\hat{x}) + \frac{1}{2}\alpha^2 d^T H(\hat{x})d + \alpha^2 \|d\|^2 \Psi(\hat{x}, \alpha d),$$

avec  $\Psi(\hat{x}, \alpha d) \rightarrow 0$ , quand  $\alpha \rightarrow 0$ . Ce ci implique

$$\frac{f(\hat{x} + \alpha d) - f(\hat{x})}{\alpha^2} = \frac{1}{2}d^T H(\hat{x})d + \|d\|^2 \Psi(\hat{x}, \alpha d), \quad (1.5)$$

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

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

Si on passe à la limite quand  $\alpha \rightarrow 0$ ,  $\alpha \neq 0$ , on obtient

$$d^T H(\hat{x})d \geq 0, \quad \forall d \in \mathbb{R}^n.$$

Ce qui termine la preuve. ■

### 1.3.3 Conditions suffisantes d'optimalité du second ordre

**Théorème 1.3.3** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois différentiable au point  $\hat{x} \in \mathbb{R}^n$ . Si  $\nabla f(\hat{x}) = 0$  et  $H(\hat{x})$  est définie positive alors  $\hat{x}$  est un minimum local strict de  $(P)$ .

**Preuve.**  $f$  étant deux fois différentiable au point  $\hat{x}$ , on aura pour tout  $x \in \mathbb{R}^n$

$$f(x) = f(\hat{x}) + \frac{1}{2}(x - \hat{x})^t H(\hat{x})(x - \hat{x}) + \|(x - \hat{x})\|^2 \Psi(\hat{x}, (x - \hat{x})). \quad (1.6)$$

---

avec  $\Psi(\hat{x}, (x - \hat{x})) \xrightarrow{x \rightarrow \hat{x}} 0$  ( $\nabla f(\hat{x}) = 0$ ).

Supposons que  $\hat{x}$  n'est pas un optimum local strict. Alors il existe une suite  $\{x_k\}_{k \in \mathbb{N}}$  telle que  $x_k \neq \hat{x} : \forall k$ , et

$$x_k \neq \hat{x} : \forall k, x_k \xrightarrow{k \rightarrow \infty} \hat{x} \text{ et } f(x_k) \leq f(\hat{x}). \quad (1.7)$$

Dans (1.6) prenons  $x = x_k$ , divisons le tout par  $\|(x - \hat{x})\|^2$  et notons

$$d_k = \frac{(x - \hat{x})}{\|(x - \hat{x})\|}, \|d_k\| = 1,$$

on obtient

$$\frac{f(x_k) - f(\hat{x})}{\|(x_k - \hat{x})\|^2} = \frac{1}{2} d_k^T H(\hat{x}) d_k + \Psi(\hat{x}, (x_k - \hat{x})), \quad \Psi(\hat{x}, (x_k - \hat{x})) \xrightarrow{k \rightarrow \infty} 0 \quad (1.8)$$

(1.5) et (1.6) impliquent

$$\frac{1}{2} d_k^T H(\hat{x}) d_k + \Psi(\hat{x}, (x_k - \hat{x})) \leq 0, \quad \forall k.$$

D'autre part la suite  $\{d_k\}_{k \in \mathbb{N}}$  est bornée ( $\|d_k\| = 1, \forall n$ ). Donc il existe une sous suite  $\{d_k\}_{k \in N_1 \subset \mathbb{N}}$  telle que

$$d_k \xrightarrow{k \rightarrow \infty, k \in N_1} \tilde{d}.$$

Finalement lorsque  $k \rightarrow \infty, k \in N_1$ , on obtient

$$\frac{1}{2} \tilde{d}^T H(\tilde{x}) \tilde{d} \leq 0.$$

La dernière relation et le fait que  $\tilde{d} \neq 0$  ( $\|\tilde{d}\| = 1$ ) impliquent que la matrice hessienne  $H(\tilde{x})$  n'est pas définie positive. Ceci est en contradiction avec l'hypothèse. ■

---

## 1.4 Les modes de convergence

La convergence globale d'un algorithme ayant été établie, nous nous intéressons maintenant à l'évaluation de son efficacité d'un point de vue pratique, l'efficacité d'un algorithme dépend du nombre d'itérations nécessaires pour obtenir une approximation à  $\varepsilon$  près ( $\varepsilon$  fixé à l'avance) de l'optimum  $\hat{x}$ . Si l'on compare entre eux plusieurs algorithmes, et si l'on admet que le temps de calcul par itération est sensiblement le même pour tous, le meilleur est celui qui nécessitera le plus petit nombre d'itérations. Malheureusement, il se révèle impossible de dégager des conclusions générales de ce genre de comparaison. Suivant le point de départ choisi, la nature de la fonction à optimiser, la valeur de la tolérance choisie, la hiérarchie des algorithmes peut varier considérablement. Si l'on veut dégager un critère ayant une certaine valeur d'absolu, il faut par conséquent recourir à un autre type d'analyse, c'est l'objet de l'étude de la *convergence asymptotique* c'est-à-dire du comportement de la suite  $\{x_k\}$  au voisinage du point limite  $\hat{x}$ . Ceci conduit à attribuer à chaque algorithme un indice d'efficacité appelé sa *vitesse de convergence*. Nous introduisons maintenant les différents types de convergence. Plaçons nous dans  $\mathbb{R}^n$ , où  $\| \cdot \|$  désigne la norme euclidienne et considérons une suite  $\{x_k\}$  convergeant vers  $\hat{x}$ .

**Définition 3** Soit  $\{x^k\}_{k \in \mathbb{N}}$  une suite dans  $\mathbb{R}^n$  convergeant vers  $x^*$ .

✓ Si  $\limsup_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = \alpha < 1$ , on dit que  $\{x^k\}_{k \in \mathbb{N}}$  converge vers  $x^*$  linéairement avec le taux  $\alpha$ .

Lorsque  $\|x^{k+1} - x^*\| \simeq \alpha \|x^k - x^*\|$ , la convergence est dite linéaire asymptotique.

✓ Si  $\limsup_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^\gamma} < +\infty$ ,  $\gamma > 1$  la convergence est dite superlinéaire d'ordre  $\gamma$ .

✓ Si  $\limsup_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^\gamma} < +\infty$ ,  $\gamma > 1$  la convergence est dite superlinéaire d'ordre  $\gamma$ .

## CHAPITRE 2

# OPTIMISATION DANS $\mathfrak{R}$ . RECHERCHES LINEAIRES EXACTES

Notre problème consiste à minimiser une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . On s'intéresse à la classe des méthodes à directions de descente, ou on ne peut pas garantir la convergence vers un minimum, même local. C'est pourquoi, ces algorithmes sont couplés à une recherche linéaire. Ainsi, on retrouve dans l'algorithme la propriété de convergence globale vers un minimum local. On suppose connaître la direction de descente  $d_k$  au point  $x_k$ .

### 2.1 Recherche linéaire

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

$$\min_{\alpha \in \mathbb{R}} f(x_k + \alpha d_k) \quad (2.1)$$

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. D'habitude, nous avons le schéma suivant d'une méthode de minimisation sans contraintes multidimensionnelle :

En regardant le comportement local de l'objectif  $f$  sur l'itération courante  $x_k$ , la méthode choisit la "direction du mouvement"  $d_k$  (qui, normalement, est une direction de

---

descente de l'objectif :  $\nabla^T f(x).d < 0$ ) et exécute un pas dans cette direction :

$$x_k \longmapsto x_{k+1} = x_k + \alpha_k d_k \quad (2.2)$$

afin de réaliser un certain progrès en valeur de l'objective, c'est-à-dire, pour assurer que :  $f(x_k + \alpha_k d_k) \leq f(x_k)$ . Et dans la majorité des méthodes le pas  $\alpha_k$  dans la direction  $d_k$  est choisi par la minimisation unidimensionnelle de la fonction :  $\varphi_k(\alpha) = f(x_k + \alpha d_k)$ . Ainsi, la technique de recherche linéaire est une brick de base fondamentale de toute méthode multidimensionnelle.

## 2.2 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

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

Le terme négatif,, 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  $f(x_k) \geq C$  pour tout  $k$ ), alors ce terme négatif tend nécessairement vers zéro : 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  $f(x_k + \alpha_k d_k) < f(x_k)$ .

### Le second objectif

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

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

- ❶ Les recherches linéaires exactes.
- ❷ Les recherches linéaires inexactes ou économiques.

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

$$\varphi_k(\alpha) = f(x_k + \alpha d_k) . \alpha \in [0. + \infty[ \quad (2.3)$$

**Définition 4** *Supposons qu'à l'itération  $k$  on ait un point  $x_k \in \mathbb{R}^n$  et une direction de*

---

descente  $d_k \in \mathbb{R}^n$ . Soit  $\alpha \in ]0, \delta[$  telle que

$$\forall \alpha \in ]0, \delta[ \quad f(x_k + \alpha d_k) < f(x_k) \quad (2.4)$$

On appelle *taux de décroissance* de  $f$  du point  $x_k$  au point  $x_k + \alpha d_k$  le nombre positif  $\tau(f, x_k, \alpha, d_k)$  suivant :

$$\tau(f, x_k, \alpha, d_k) = f(x_k) - f(x_k + \alpha d_k) = \varphi_k(0) - \varphi_k(\alpha) \quad (2.5)$$

## 2.3 Choix optimal du pas $\alpha_k$

$f, x_k$  et  $d_k$  étant fixes,  $\tau(f, x_k, \alpha, d_k)$  dépend de  $\alpha$ . La plus grande valeur de  $\tau(f, x_k, \alpha, d_k)$  mesure la meilleure descente de  $f$  du point  $x_k$  au point  $x_k + \alpha d_k$ . Puisque  $f(x_k)$  est fixe et ne dépend de  $\alpha$ , la plus grande valeur de  $\tau(f, x_k, \alpha, d_k)$  est atteinte pour la plus petite valeur de  $f(x_k + \alpha d_k)$ , c'est à dire

$$\begin{aligned} & \max \{ \tau(f, x_k, \alpha, d_k) : \alpha \in ]0, +\infty[ \} \\ &= \max \{ \varphi_k(0) - \varphi_k(\alpha) : \alpha \in ]0, +\infty[ \} \\ &= \min \{ \varphi_k(\alpha) : \alpha \in ]0, +\infty[ \} \end{aligned}$$

En d'autres termes, on peut choisir  $\alpha_k$  de façon optimale, c'est à dire qu'en partant du point  $x_k$  et la direction  $d_k$ ,  $f$  décroît le mieux possible au point  $x_{k+1} = x_k + \alpha_k d_k$ . Ceci est possible si on impose à  $\alpha_k$  de vérifier la condition suivante :

$$\forall \alpha \in ]0, +\infty[ : \quad f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k) \quad (2.6)$$

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

$$\varphi_k(\alpha) = f(x_k + \alpha d_k) : \quad \alpha \in ]0, +\infty[ \quad (2.7)$$

(2.6) et (2.7) donnent

$$\varphi_k(\alpha_k) \leq \varphi_k(\alpha), \quad \forall \alpha \in ]0, +\infty[ \quad (2.8)$$

ou encore  $\alpha_k$  est solution optimale de :

$$\varphi_k(\alpha_k) = \min \{ \varphi_k(\alpha) : \alpha \in ]0, +\infty[ \} \quad (2.9)$$

---

Trouvez  $\alpha_k$  vérifiant (2.9) s'appelle recherche linéaire exacte. Ce travail s'effectue à chaque itération  $k$ .

Donc à chaque itération  $k$ , on doit résoudre un problème d'optimisation dans  $\mathbb{R}$ .

## 2.4 Optimisation dans $\mathfrak{R}$ . Recherches Linéaires Exactes

### 2.4.1 L'intervalle d'incertitude

**Définition 5** Soit  $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ . Considérons le problème unidimensionnel suivant :

$$\min_{\alpha \in ]0, +\infty[} \varphi(\alpha) = \varphi(\alpha^*)$$

L'intervalle  $[a, b] \subset ]0, +\infty[$  est dit intervalle d'incertitude si la solution minimale  $\alpha^*$  de  $\varphi(\alpha)$  appartient à  $[a, b]$ , mais sa valeur exacte n'est pas connue.

La notion d'intervalle d'incertitude joue un rôle central dans les algorithmes de recherches linéaires exactes et particulièrement dans la méthode du nombre d'or. Cette méthode exige au démarrage la connaissance d'un intervalle d'incertitude initial  $[a_0, b_0]$ .

### 2.4.2 Schémas général des recherches linéaires exactes utilisant les intervalles d'incertitude

Le principe général de toutes ces méthodes est de construire une suite d'intervalles  $[a_k, b_k]$  telles que :

**a)** On démarre à partir d'un intervalle d'incertitude initial  $[a_0, b_0]$  et  $\varepsilon > 0$  qui détermine la longueur finale de l'intervalle d'incertitude  $[a_k, b_k]$

( En d'autres termes l'algorithme s'arrête lorsque  $b_k - a_k < \varepsilon$ ).

**b)**  $\forall k : [a_k, b_k]$  est un intervalle d'incertitude. En d'autres termes  $\alpha^* \in [a_k, b_k]$  pour tout  $k$

**c)**  $\forall k : [a_{k+1}, b_{k+1}] \subset [a_k, b_k]$

**d)** L'algorithme s'arrête si  $b_k - a_k < \varepsilon$

**e)** Comme solution  $\alpha^*$  on prend n'importe quel nombre  $\bar{\alpha} \in [a_k, b_k]$ . On pourrait prendre

par exemple

$$\alpha^* \simeq \bar{\alpha} = \frac{a_k + b_k}{2}, \quad a_k \text{ et } b_k \text{ tels que : } b_k - a_k < \varepsilon.$$

**Conclusion :**

On démarre d'un intervalle d'incertitude initial  $[a_0, b_0]$  et  $\varepsilon > 0$  (longueur finale de l'intervalle  $[a_f, b_f]$ ). A chaque itération  $k$ , on enlève une partie de l'intervalle d'incertitude  $[a_k, b_k]$ , pour obtenir le nouveau intervalle d'incertitude  $[a_{k+1}, b_{k+1}]$ , et ainsi de suite jusqu'à arriver au dernier intervalle d'incertitude  $[a_f, b_f]$  vérifiant  $b_f - a_f < \varepsilon$ .

Ceci étant, une question naturelle se pose. Quel procédé nous permet de déterminer la partie de l'intervalle  $[a_k, b_k]$  à enlever. La réponse à cette question fera l'objet des méthodes suivantes

### 2.4.3 La méthode progressive-retrogressive ou Forward-Backward Algorithm

Nous avons vu au paragraphe précédent, que la notion d'intervalle d'incertitude joue un rôle central dans les algorithmes de recherches linéaires exactes et particulièrement dans la méthode du nombre d'or. Cette méthode ainsi que la méthode de dichotomie et la méthode de Fibonacci exigent au démarrage la connaissance d'un intervalle d'incertitude initial  $[a_0, b_0]$ .

La méthode suivante dite méthode **progressive-retrogressive ou en termes anglo-saxon : Forward-Backward Algorithm** consiste à déterminer un intervalle d'incertitude initial  $[a, b]$ . Son principe se résume à trouver trois points  $\alpha, \eta$  et  $\beta$  tels que  $\varphi$  décroît de  $\alpha$  à  $\eta$  et croît de  $\eta$  à  $\beta$ . En d'autres termes on devrait choisir  $\alpha, \eta$  et  $\beta$  tels que :

$$\varphi(\eta) < \varphi(\alpha) \quad \text{et} \quad \varphi(\eta) < \varphi(\beta).$$

Sous ces conditions, on choisit

$$[a, b] = [\alpha, \beta]$$

L'Algorithme suivant permet de déterminer un intervalle d'incertitude initial  $[a, b]$ .

Algorithme (méthode progressive-retrogressive) (forward-backward method)

**Etape1 : Initialisation**

Choisir  $\alpha_0 \in [0, +\infty[$ ,  $h_0 > 0$ . Calculez  $\varphi(\alpha_0)$ . Poser  $k = 0$  et allez à **Etape2**

**Etape2 : Comparaison des valeurs de la fonction objectif**

Posez  $\alpha_{k+1} = \alpha_k + h_k$ . Calculez  $\varphi_{k+1} = \varphi(\alpha_{k+1})$ . Si  $\varphi_{k+1} < \varphi_k$ , allez à **Etape3**, sinon allez à **Etape4**

**Etape3 : Etape Progressive (Forward Step)**

Posez  $h_{k+1} = 2h_k$ ,  $\alpha = \alpha_k$ ,  $\alpha_k = \alpha_{k+1}$ ,  $\varphi_k = \varphi_{k+1}$ ,  $k = k + 1$  et Allez à **Etape2**

**Etape4 : Etape retrogressive (Backward Step) ou finalisation**

Si  $k = 0$ , intervertir la direction. Poser  $h_k = -h_k$ ,  $\alpha_k = \alpha_{k+1}$  et allez à **Etape2**.

---

Sinon Poser

$$a = \min(\alpha, \alpha_{k+1}), \quad b = \max(\alpha, \alpha_{k+1})$$

Output  $[a, b]$ . L'intervalle cherché est  $[a, b]$ .

Stop.

### Déscription de l'Algorithme : forward-backward method

#### Principe général

Comme on l'a signalé avant, l'Algorithme précédent : **Forward-Backward Algorithm** consiste à déterminer un intervalle d'incertitude initial  $[a, b]$ .

**A l'itération**  $k$ , l'Algorithme s'arrête si a réussi à trouver trois points  $\alpha_k, \alpha_{k+1}, \alpha_{k+2}$  tels que  $\varphi$  décroît strictement de  $\alpha_k$  à  $\alpha_{k+1}$  et croit strictement de  $\alpha_{k+1}$  à  $\alpha_{k+2}$ . En d'autres termes on devrait choisir  $\alpha_k, \alpha_{k+1}, \alpha_{k+2}$  tels que :

$$\varphi(\alpha_{k+1}) < \varphi(\alpha_k) \quad \text{et} \quad \varphi(\alpha_{k+1}) < \varphi(\alpha_{k+2}).$$

Sous ces conditions, on choisit

$$[a, b] = [\alpha_k, \alpha_{k+2}]$$

**Théorème 2.4.1** *Si au démarrage on a*

**Cas1 :**  $\varphi(\alpha_1) < \varphi(\alpha_0)$ , alors l'algorithme progresse à droite du point  $\alpha_0$ . On aura pour tout  $k$

$$\alpha_{k+1} = \alpha_k + 2^k h_0$$

*Si au contraire on a*

**Cas2 :**  $\varphi(\alpha_1) > \varphi(\alpha_0)$ , alors l'algorithme progresse en faisant une marche arrière à partir du point  $\alpha_0$ . On a pour tout  $k$

$$\alpha_{k+1} = \alpha_k - 2^k h_0$$

#### Détails

L'Algorithme démarre par un point initial  $\alpha_0 \in [0, +\infty[$  et un pas initial  $h_0 > 0$ , le même pour tout l'algorithme. On calcule  $\varphi(\alpha_0)$ , on pose  $k = 0$  et on va à

#### Etape2.

On calcule  $\alpha_1 = \alpha_0 + h_0$  et  $\varphi(\alpha_1)$  et on compare les 2 valeurs  $\varphi(\alpha_0)$  et  $\varphi(\alpha_1)$ . Deux cas sont possibles :

**Cas1 :**  $\varphi(\alpha_1) < \varphi(\alpha_0)$

L'Algorithme progresse à droite du point  $\alpha_1$ , c'est à dire que tous les autres points

---

$\alpha_k, k = 2, 3, \dots$  se trouvent à droite du point  $\alpha_1$

$$\begin{aligned}\alpha_2 &= \alpha_1 + h_0 \\ \alpha_3 &= \alpha_2 + h_0 \\ &\dots\dots\dots \\ \alpha_k &= \alpha_{k-1} + h_0\end{aligned}$$

**Cas2 :**  $\varphi(\alpha_1) > \varphi(\alpha_0)$

L'Algorithme progresse en faisant une marche arrière, à gauche du point  $\alpha_0$ , c'est à dire que tous les autres points  $\alpha_k, k = 2, 3, \dots$  se trouvent à gauche du point  $\alpha_0$

$$\begin{aligned}\alpha_2 &= \alpha_0 - h_0 \\ \alpha_3 &= \alpha_2 - h_0 \\ &\dots\dots\dots \\ \alpha_k &= \alpha_{k-1} - h_0\end{aligned}$$

### Explication et détails

**Cas1 :**  $\varphi(\alpha_1) < \varphi(\alpha_0)$  ((**On va progresser dans toutes les itérations en faisant des marches avant :**  $\alpha_{k+1} = \alpha_k + h_k$ ))

D'après l'Algorithme on va à l'Étape 3 dans laquelle on fait doubler le pas  $h_1 = 2h_0$  (progression). L'itération devient  $k = k + 1$ , c'est à dire  $k = 0 + 1 = 1$  et on revient à l'**Étape2(0)**.

#### **Étape2(0)**

on calcule  $\alpha_{k+1} = \alpha_2$ , puis  $\varphi(\alpha_2)$ .

Si  $\varphi(\alpha_2) > \varphi(\alpha_1)$ , on va à l'**Étape4(0)**

#### **Étape4(0)**

A L'étape4(0), puisque  $k = 1$ , l'algorithme va s'arreter et  $[a, b] = [\alpha_0, \alpha_2]$

Si  $\varphi(\alpha_2) < \varphi(\alpha_1)$ , on va à l'**Étape3(0)**

#### **Étape3(0)**

A l'étape3, on augmente le pas  $h_2 = 2h_1 = 2^2h_0$ , On pose  $k = k + 1 = 1 + 1 = 2$  et on va à l'**Étape2(bis)**

#### **Étape2(bis)**

on calcule  $\alpha_{k+1} = \alpha_k + h_k = \alpha_3 = \alpha_2 + h_2 = \alpha_2 + 2^2h_0 = \alpha_2 + 4h_0$ , puis  $\varphi(\alpha_3)$ .

Si  $\varphi(\alpha_3) > \varphi(\alpha_2)$ , on va à l'étape4 et l'algorithme s'arrête et  $[a, b] = [\alpha_1, \alpha_3]$ .

Si  $\varphi(\alpha_3) < \varphi(\alpha_2)$ , on va à l'étape3 une deuxième fois qu'on note **Étape3 (bis)**

**Étape3 (bis)** on augmente le pas  $h_3 = 2h_2 = 2^3h_0$ , On pose  $k = k + 1 = 2 + 1 = 3$

---

et on va à l'**Etape2(tris)**

**Etape2 (bis)** on calcule  $\alpha_{k+1} = \alpha_k + h_k = \alpha_4 = \alpha_3 + h_3 = \alpha_3 + 2^3 h_0 = \alpha_2 + 8h_0$ , puis  $\varphi(\alpha_4)$ .

Si  $\varphi(\alpha_4) > \varphi(\alpha_3)$ , on va à l'étape4 et l'algorithme s'arrête et  $[a, b] = [\alpha_2, \alpha_4]$ .

Si  $\varphi(\alpha_3) < \varphi(\alpha_2)$ , on va à l'étape3 une troisième fois qu'on note **Etape3 (tris)** et on fait une autre progression du pas et ainsi de suite jusqu'à avoir l'intervalle final  $[a, b]$ .

**Cas2** :  $\varphi(\alpha_1) > \varphi(\alpha_0)$  (**On va progresser dans toutes les itérations en faisant des marches arrières** :  $\alpha_{k+1} = \alpha_k - h_k$ )

D'après l'Algorithme, on va à l'Etape4(0)

**Etape4(0)**

Puisque  $k = 0$ , l'algorithme ne s'arrête pas à cette étape.

On change de direction en posant  $h_k = -h_k$ , c'est à dire  $h_0$  devient  $-h_0$ ,  $\alpha_0$  devient  $\alpha_1$ . Donc on pose  $h_0 = -h_0$ ,  $\alpha_0 = \alpha_1$  et on va à l'étape2(0)

**Etape2(0)**

$\alpha_1 = \alpha_0 + h_0 = \alpha_1 - h_0 = \alpha_0$  (Donc  $\alpha_0$  devient  $\alpha_1$  et  $\alpha_1$  devient  $\alpha_0$ ). Puisqu'on est dans le cas2 et que  $\alpha_0$  est devenu  $\alpha_1$  et  $\alpha_1$  est devenu  $\alpha_0$ ,

on a donc

$\varphi(\alpha_1) < \varphi(\alpha_0)$ . Par conséquent on va à l'Etape3(0)

**Etape3(0)**

$h_1 = 2h_0 = -2h_0$ ,  $\alpha_0 = \alpha_1$ ,  $\varphi(\alpha_0) = \varphi(\alpha_1)$ ,  $k = k + 1 = 0 + 1 = 1$ . Allez à Etape2(bis)

**Etape2(bis)** (progression en marche arrière)

$\alpha_2 = \alpha_1 + h_1 = \alpha_0 - 2h_0$  (On progresse en faisant une marche arrière à partir du point  $\alpha_0$ ). On calcule  $\varphi(\alpha_2)$  et on compare  $\varphi(\alpha_2)$  et

$\varphi(\alpha_1) = \varphi(\alpha_0)$ .

Si  $\varphi(\alpha_2) > \varphi(\alpha_0)$ , on va à l'étape4(bis)

**Etape4 (bis)**

$k = 1$ . Donc l'Algorithme s'arrête et  $[a, b] = [\alpha_2, \alpha_1]$

Si  $\varphi(\alpha_2) < \varphi(\alpha_0)$ , on va à l'étape3(bis)

**Etape3(bis)**

$h_2 = 2h_1 = -4h_0$  et  $\alpha_3 = \alpha_2 + h_2 = \alpha_2 - 4h_0$  (on progresse en marche arrière). On calcule  $\varphi(\alpha_3)$  et on compare  $\varphi(\alpha_3)$  et  $\varphi(\alpha_2)$ .

Si  $\varphi(\alpha_3) > \varphi(\alpha_2)$ , on va à l'étape 4 et l'algorithme s'arrête,  $[a, b] = [\alpha_3, \alpha_1]$

Si  $\varphi(\alpha_3) < \varphi(\alpha_2)$ , on va à **Etape3(tris)** ((on fait une nouvelle progression en marche arrière) en posant

$h_3 = 2h_2 = -8h_0$  et  $\alpha_4 = \alpha_3 + h_3 = \alpha_3 - 8h_0$  et ainsi de suite.

---

## Fonctions unimodales

**Définition 6** Soit  $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ .  $\varphi$  est dite unimodale sur l'intervalle  $[a, b]$ , s'il existe  $\alpha^* \in ]a, b[$  unique tel que :

- a)  $\varphi$  est strictement décroissante sur  $[a, \alpha^*]$
- b)  $\varphi$  est strictement croissante sur  $[\alpha^*, b]$

L'intervalle  $[a, b]$  s'appelle intervalle d'unimodalité associé à la fonction  $\varphi$ .

**Remarque 2.4.1** Si  $\varphi$  est unimodale sur l'intervalle  $[a, b]$ , alors  $\varphi$  admet une solution minimale unique  $\alpha^* \in ]a, b[$  et  $\varphi$  est strictement décroissante sur  $[a, \alpha^*]$  et strictement croissante sur  $[\alpha^*, b]$ .

On va voir dans le théorème suivant que la connaissance de deux valeurs d'une fonction unimodale sur  $[a, b]$  permet de réduire l'intervalle d'incertitude.

**Théorème 2.4.2** Soit  $\varphi : \mathbb{R} \rightarrow \mathbb{R}$  telle que  $\varphi$  est unimodale sur l'intervalle  $[a, b]$ . Soient  $\lambda, \mu \in [a, b]$  tels que  $\lambda < \mu$

- 1- Si  $\varphi(\lambda) \leq \varphi(\mu)$ , alors  $\varphi$  est unimodale sur l'intervalle  $[a, \mu]$ .  $[a, \mu]$  est le nouveau intervalle d'incertitude associé à  $\varphi$ .
- 2- Si  $\varphi(\lambda) > \varphi(\mu)$ , alors  $\varphi$  est unimodale sur l'intervalle  $[\lambda, b]$ .  $[\lambda, b]$  est le nouveau intervalle d'incertitude associé à  $\varphi$ .

**Conséquence importante du théorème 2.3.2.**

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

---

## 2.4.4 Deux méthodes d'optimisation unidimensionnelle sans dérivées.

### La méthode de dichotomie.

La méthode de dichotomie est simple. Elle se base sur le théorème 1. Le choix des points  $\lambda_k$  et  $\mu_k$  se fait de la façon suivante :

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

On fixe  $\varepsilon > 0$  au démarrage, assez petit. A l'itération  $k$ ,  $\lambda_k$  est le milieu de  $[a_k, b_k]$  moins  $\varepsilon$ .  $\mu_k$  est le milieu de  $[a_k, b_k]$  plus  $\varepsilon$ . En d'autres termes

$$\begin{aligned}\lambda_k &= \frac{a_k + b_k}{2} - \varepsilon \\ \mu_k &= \frac{a_k + b_k}{2} + \varepsilon\end{aligned}$$

#### Algorithmme de la méthode de dichotomie)

##### Initialisation :

Choisir  $\varepsilon > 0$  et la longueur finale de l'intervalle d'incertitude  $\ell$ .

$[a_1, b_1]$  étant l'intervalle initial.

Poser  $k = 1$  et aller à l'étape 1.

##### Etape 1 :

Si  $b_k - a_k < \ell$  stop. Le minimum appartient à  $[a_k, b_k]$ .

Sinon poser :

$$\begin{aligned}\lambda_k &= \frac{a_k + b_k}{2} - \varepsilon \\ \mu_k &= \frac{a_k + b_k}{2} + \varepsilon\end{aligned}$$

et aller à l'étape 2

##### Etape 2 :

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

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

---

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

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

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

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

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

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

Dans la méthode du nombre d'or, on exige que  $\lambda_k$  et  $\mu_k$  vérifient les deux conditions suivantes :

#### Condition 1

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

$$b_k - \lambda_k = \mu_k - a_k \quad (2.10)$$

#### Condition 2

Le choix de  $\lambda_{k+1}$  et  $\mu_{k+1}$  se fait de la façon suivante :

**A l'itération  $k + 1$  on a**

a)  $\lambda_{k+1}$  coïncide avec  $\mu_k$

ou bien

b)  $\mu_{k+1}$  coïncide avec  $\lambda_k$ .

#### Conséquence importante

En d'autres termes, la méthode du nombre d'or calcule une seule valeur à l'itération  $k + 1$ .

En effet.

a) Si  $\lambda_{k+1} = \mu_k$ , on calcule à l'itération  $k + 1$  seulement la valeur  $\varphi(\mu_{k+1})$ , car  $\varphi(\lambda_{k+1}) = \varphi(\mu_k)$  a été calculée à l'itération  $k$ .

b) Si  $\mu_{k+1} = \lambda_k$ , on calcule à l'itération  $k + 1$  seulement la valeur  $\varphi(\lambda_{k+1})$ , car  $\varphi(\mu_{k+1}) = \varphi(\lambda_k)$  a été calculée à l'itération  $k$ .

Avec ces deux conditions, on obtient les théorèmes suivants :

**Théorème 2.4.3** Si  $\lambda_k$  et  $\mu_k$  vérifient la condition (2.10), alors il existe  $\alpha \in ]0, 1[$  tel que

$$\mu_k = a_k + \alpha (b_k - a_k), \quad (2.11)$$

$$\lambda_k = a_k + (1 - \alpha) (b_k - a_k). \quad (2.12)$$

et dans les 2 cas possibles :  $\varphi(\lambda_k) > \varphi(\mu_k)$  ( $[a_{k+1}, b_{k+1}] = [\lambda_k, b_k]$ ) ou  $\varphi(\lambda_k) \leq \varphi(\mu_k)$  ( $[a_{k+1}, b_{k+1}] = [a_k, \mu_k]$ ), la longueur de l'intervalle  $[a_{k+1}, b_{k+1}]$  diminuera  $\alpha$  fois par rapport à la longueur de  $[a_k, b_k]$ , c'est à dire qu'on a :

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

**Théorème 2.4.4** Sous les conditions 1 et 2 citées ci dessus, on a

**Cas1 : Si  $\varphi(\lambda_k) \leq \varphi(\mu_k)$ , alors on a**

$$\left\{ \begin{array}{l} a_{k+1} = a_k \\ b_{k+1} = \mu_k \\ \mu_{k+1} = \lambda_k \\ \lambda_{k+1} = a_k + (1 - \alpha) (\mu_k - a_k) \end{array} \right. \quad (2.13)$$

**Cas2 : Si  $\varphi(\lambda_k) > \varphi(\mu_k)$ , alors on a**

$$\left\{ \begin{array}{l} a_{k+1} = \lambda_k \\ b_{k+1} = b_k \\ \lambda_{k+1} = \mu_k \\ \mu_{k+1} = a_k + \alpha (b_k - \lambda_k) \end{array} \right. \quad (2.14)$$

Le nombre  $\alpha \in ]0, 1[$  intervenant à chaque itération de la méthode du nombre d'or est solution de l'équation

$$\alpha^2 + \alpha - 1 = 0$$

et sa valeur approximative est

$$\alpha \simeq 0.618$$

---

**Algorithme de la méthode du nombre d'or.)**

**Algorithme**

**Etape initiale :**

Donner  $[a_1, b_1]$  le premier intervalle d'incertitude et choisir  $\ell > 0$  la longueur finale de l'intervalle d'incertitude final  $[a_k, b_k]$ . Poser  $\alpha = 0.618$  et  $(1 - \alpha) = 1 - 0.618 = 0.382$   
Calculer

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

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

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

**Etape principale :**

**1-** Si  $b_k - a_k < \ell$ , stop. La solution optimale appartient à  $[a_k, b_k]$ .

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

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

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

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

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

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

## 2.5 Avantages et Inconvénients des recherches linéaires exactes

**Avantages**

Le principal avantage des recherches linéaires exactes est de calculer  $\alpha_k$  optimal, c'est  $\alpha_k = \alpha^*$ , vérifiant

$$\varphi_k(\alpha_k) = \varphi_k(\alpha^*) = \min\{\varphi_k(\alpha) : \alpha \in ]0, +\infty[ \}$$

En d'autres termes, si  $\alpha_k = \alpha^*$  la fonction  $f$  décroît le mieux en passant du point  $x_k$  au point  $x_k + \alpha_k d_k = x_k + \alpha^* d_k$ , c'est à dire

$$\forall \alpha \in ]0, +\infty[ : [f(x_k) - f(x_k + \alpha_k d_k)] = [f(x_k) - f(x_k + \alpha^* d_k)] \geq [f(x_k) - f(x_k + \alpha d_k)]$$

---

## **Inconvénients**

Effectuer une recherche linéaire exacte à chaque itération, est difficilement réalisable en pratique et assez couteux en temps et en mémoire.

## CHAPITRE 3

# OPTIMISATION DANS $\mathfrak{R}$ . RECHERCHES LINEAIRES INEXACTES

### 3.1 Introduction et rappels

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

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

Le but est de réduire «de façon importante» la valeur de l'objectif par un pas  $x_k \mapsto x_{k+1} = x_k + \alpha_k d_k$  de  $x_k$  dans la direction  $d_k$ . Pour cela de nombreux mathématiciens (Armijo, Goldstein, Wolfe, Albaali, Lemaréchal, Fletcher...) ont élaborés plusieurs règles. L'objectif de cette section consiste à présenter les principales tests.

#### 3.1.1 Le schéma des recherches linéaires inexactes

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

$$\varphi(\alpha_k) < \varphi(0) \quad \Leftrightarrow \quad (f(x_k + \alpha_k d_k) < f(x_k)).$$

Le schéma de l'algorithme est donc :

**Algorithme.**

**Etape 0 : (initialisation)**

---

$a_1 = b_1 = 0$ , choisir  $\alpha_1 > 0$ ; poser  $k = 1$  et aller à l'étape 1.

**Etape 1 :**

Si  $\alpha_k$  est satisfaisant (suivant un certain critère) : STOP ( $\hat{\alpha} = \alpha_k$ ).

Si  $\alpha_k$  est trop petit (suivant un certain critère) : nouvel intervalle :  $[a_{k+1} = \alpha_k, b_{k+1} = b]$   
et aller à l'étape 2.

Si  $\alpha_k$  est trop grand (suivant un certain critère) : nouvel intervalle :  $[a_{k+1} = a, b_{k+1} = \alpha_k]$   
et aller à l'étape 2.

**Etape 2 :**

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

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

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

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

## 3.2 La règle d'Armijo

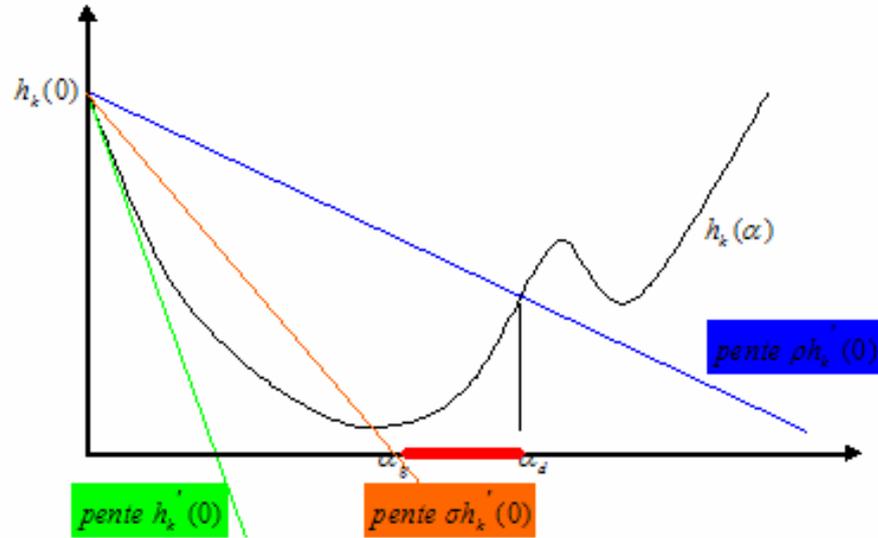
On bute à réduire «de façon importante »la valeur de l'objectif par un pas  $x_k \mapsto x_{k+1} = x_k + \alpha_k d_k$  de  $x_k$  dans la direction de descente  $d_k$ , tel que  $f(x_k + \alpha_k d_k) < f(x_k)$ . Or cette condition de décroissance stricte n'est pas suffisante pour minimiser  $\varphi$  au moins localement. Une condition naturelle est de demander que  $f$  décroisse autant qu'une portion  $\delta \in ]0, 1[$  parfois appelée condition d'Armijo ou condition de décroissance linéaire :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k \nabla f(x_k)^T d_k. \quad (3.1)$$

Elle est de la forme ( $f(x_k + \alpha_k d_k) < f(x_k)$ ), car  $\alpha_k$  devra être choisi dans  $]0, 1[$ . On voit bien à la figure suivante ce que signifie cette condition. Il faut qu'en  $\alpha_k$ , la fonction prenne une valeur plus petite que celle prise par la fonction affine

$$\alpha \rightarrow f(x_k) + \delta \alpha_k \nabla f(x_k)^T d_k. \quad (3.2)$$

### Règle d'Armijo



1.pdf

Fig 3.1 :La Règle d'Armijo

- ◆ Si  $\varphi(\alpha) \leq \varphi(0) + \delta \varphi'(0) \alpha$ , alors  $\alpha$  convient,
- ◆ Si  $\varphi(\alpha) > \varphi(0) + \delta \varphi'(0) \alpha$ ; alors  $\alpha$  est trop grand.

On peut noter que l'on a

$$\begin{aligned}\varphi(\alpha) &= f(x_k + \alpha d_k), \\ \varphi(0) &= f(x_k), \\ \varphi'(0) &= \nabla f(x_k)^T d_k.\end{aligned}$$

### Algorithme 3.1

#### Etape 0 : (initialisation)

$a_1 = b_1 = 0$ , choisir  $\alpha_1 > 0$ ,  $\delta \in ]0, 1[$  poser  $k = 1$  et aller à l'étape 1 ;

#### Etape 1 :

Si  $\varphi(\alpha_k) \leq \varphi(0) + \delta \varphi'(0) \alpha_k$  : stop ( $\hat{\alpha} = \alpha_k$ )

Si  $\varphi(\alpha_k) > \varphi(0) + \delta \varphi'(0) \lambda_k$ , alors  $b_{k+1} = b$ ,  $a_{k+1} = k$  et aller à l'étape 2 ;

#### Etape 2 :

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

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

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

☞ En pratique, la constante  $\delta$  est prise très petite, de manière à satisfaire (3.1) le plus facilement possible. Typiquement,  $\delta = 10^{-4}$ . Notons que cette constante ne doit pas être adaptée aux données du problème et donc que l'on ne se trouve pas devant un choix de valeur délicat,

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

⇒ Il est clair d'après la figure (3.1) que l'inégalité (3.1) est toujours vérifiée si  $\alpha_k > 0$  est suffisamment petit,

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

### 3.3 La règle de Goldstein et Price

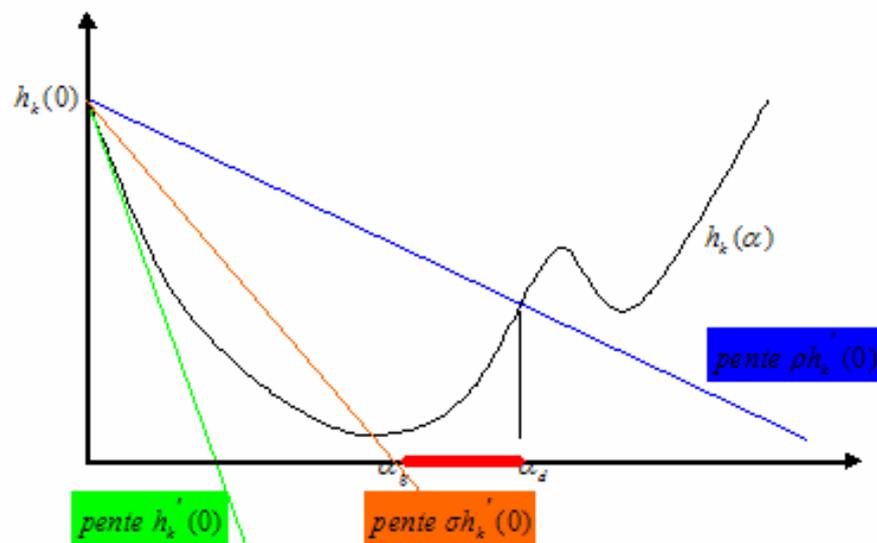
Dans la règle d'*Armijo* on assure la décroissance de la fonction objectif à chaque pas, mais c'est ne pas suffisant. Les conditions de *Goldstein* et *Price* suivantes sont, comme on va le prouver, suffisantes pour assurer la convergence sous certaines conditions et indépendamment de l'algorithme qui calcule le paramètre. Dans celle-ci, en ajoutant une deuxième inégalité à la règle d'*Armijo* on obtient la règle de *Goldstein*.

$$f(x_k) + \delta \alpha_k \nabla^T f(x_k) d_k \geq f(x_k + \alpha_k d_k) \geq f(x_k) + \sigma \alpha_k \nabla^T f(x_k) d_k. \quad (3.3)$$

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

#### Règle de Goldstein

*goldstein*



2.pdf

Fig 3.2 :La Règle de Goldstein

- ♣ Si  $\varphi(\lambda) < \varphi(0) + \sigma\varphi'(0)\lambda$ , alors  $\lambda$  est trop petit
- ♣ Si  $\varphi(\alpha) > \varphi(0) + \delta\varphi'(0)\alpha$ , alors  $\alpha$  est trop grand.
- ♣ Si  $\varphi(0) + \delta\varphi'(0)\alpha \geq \varphi(\alpha) \geq \varphi(0) + \sigma\varphi'(0)\alpha$ , alors convient.

On peut noter que l'on a

$$\begin{aligned}\varphi(\alpha) &= f(x_k + \alpha d_k) \\ \varphi(0) &= f(x_k).\end{aligned}$$

#### AAlgorithme 2.2 (Règle de Goldstein et Price)

##### Etape 0 : (initialisation)

$a_1 = b_1 = 0$ , choisir  $\alpha_1 > 0$ ,  $\delta \in ]0, 1[$ ,  $\sigma \in ]\delta, 1[$ , poser  $k = 1$  et aller à l'étape 1 ;

##### Etape 1 :

Si  $\varphi(0) + \sigma\varphi'(0)\alpha \leq \varphi(\alpha_k) \leq \varphi(0) + \delta\varphi'(0)\alpha_k$  : stop ( $\hat{\alpha} = \alpha_k$ ).

Si  $\varphi(\alpha_k) > \varphi(0) + \delta\varphi'(0)\alpha_k$ , alors  $b_{k+1} = \alpha_k$ ,  $a_{k+1} = a_k$ , et aller à l'étape 2.

Si  $\varphi(\alpha_k) < \varphi(0) + \sigma\varphi'(0)\alpha_k$ , alors  $b_{k+1} = b_k$ ,  $a_{k+1} = \alpha_k$  ; et aller à l'étape 2.

##### Etape 2 :

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

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

**Théorème 3.3.1** Si  $\varphi : \mathbb{R}_+ \longrightarrow \mathbb{R}$  définie par  $\varphi(\alpha) = f(x_k + \alpha d_k)$  est continue et bornée inférieurement, si  $d_k$  est une direction de descente en  $x_k$  et si  $\delta \in ]0, 1[$ ,  $\sigma \in ]\delta, 1[$ , alors l'ensemble des pas vérifiant la règle de Goldstein et Price est non vide.

## 3.4 La règle de Wolfe

Les conditions de la règle de *Goldstein et Price* peuvent exclure un minimum ce qui est peut être un inconvénient. La règle de *Wolfe* fait appel au calcul de  $\varphi'(\alpha)$ , elle est donc en théorie plus coûteuse que la règle de *Goldstein*. Cependant dans de nombreuses applications, le calcul du gradient  $\nabla f(x)$  représente un faible coût additionnel en comparaison du coût d'évaluation de  $f(x)$ , c'est pourquoi cette règle est très utilisée. Nous allons présenter les conditions de *Wolfe* faibles sur  $\alpha > 0$  :

$$f(x_k + \alpha d_k) \leq f(x_k) + \delta\alpha \nabla f(x_k)^T \cdot d_k, \quad (3.4)$$

$$\nabla f(x_k + \alpha d_k)^T \cdot d_k \geq \sigma \nabla f(x_k)^T \cdot d_k. \quad (3.5)$$

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

### Règle de Wolfe

*dewolfe*

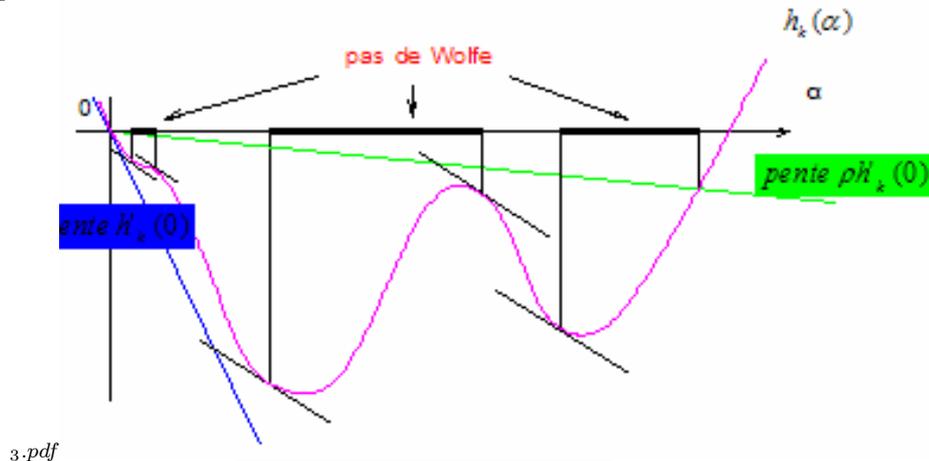


Fig 3.2 :La Règle de Wolfe

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

On voit bien à la figure (3.3) ce que signifie cette condition.

#### Algorithme 3.3 (Règle de Wolfe)

##### Etape 0 : (initialisation)

$a_1 = b_1 = 0$ , choisir  $\alpha_1 > 0$   $\delta \in ]0, 1[$ ,  $\sigma \in ]\delta, 1[$ , poser  $k = 1$  et aller à l'étape 1

##### Etape 1 :

Si  $\varphi(\alpha_k) \leq \varphi(0) + \delta \varphi'(0) \alpha_k$  et  $\varphi'(\alpha_k) \geq \sigma \varphi'(0)$  : STOP ( $\hat{\alpha} = \alpha_k$ ).

Si  $\varphi(\alpha_k) > \varphi(0) + \delta \varphi'(0) \alpha_k$ , alors  $b_{k+1} = \alpha_k$ ,  $a_{k+1} = a_k$ , et aller à l'étape 2.

Si  $\varphi'(\alpha_k) < \sigma \varphi'(0)$ , alors  $b_{k+1} = b_k$ ,  $a_{k+1} = \alpha_k$ ; et aller à l'étape 2.

##### Etape 2 :

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

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

## 3.5 La règle de Wolfe fort

On obtient des contraintes plus fortes si l'on remplace (3.5) par

$$|\nabla f(x_k + \alpha d_k)^T \cdot d_k| \leq -\sigma \nabla f(x_k)^T \cdot d_k. \quad (3.6)$$

---

Les equations (3.4) et(3.6) sont les conditions de Wolfe fortes. La contrainte (3.6) entraîne que  $\sigma\varphi(0) \leq \varphi'(\alpha) < -\sigma\varphi(0)$  c'est a dire.  $\varphi'(\alpha)$  n'est pas "trop" positif.

### 3.6 La règle de Wolfe relaxée.

Proposée par *Dai* et *Yuan* (1996), cette règle consiste à choisir le pas satisfaisant aux conditions :

$$(f(x_k + \alpha d_k) \leq f(x_k) + \delta\alpha\nabla^T f(x_k).d_k, \quad (3.7)$$

$$\sigma_1\nabla^T f(x_k).d_k \leq \nabla^T f(x_k + \alpha_k d_k).d_k \leq -\sigma_2\nabla^T f(x_k).d_k. \quad (3.8)$$

Où  $0 < \delta < \sigma_1 < 1$  et  $\sigma_2 > 0$ .

On voit bien que les conditions de *Wolfe* relaxée impliquent les conditions de *Wolfe* fortes. Effectivement (3.8) est équivalente à (3.6) , tandis que pour le cas particulier  $\sigma_1 = \sigma_2 = \sigma$ , (3.8) est équivalente à (3.6). En effet :

$$\begin{aligned} \sigma_1\nabla^T f(x_k).d_k &\leq \nabla^T f(x_k + \alpha_k d_k).d_k \leq -\sigma_2\nabla^T f(x_k).d_k \\ \implies \sigma\nabla^T f(x_k).d_k &\leq \nabla^T f(x_k + \alpha d_k).d_k \leq -\sigma\nabla^T f(x_k).d_k \\ \implies |\nabla^T f(x_k + \alpha d_k).d_k| &\leq -\sigma\nabla^T f(x).d_k \quad (3.6). \end{aligned}$$

## CHAPITRE 4

# GRADIENT CONJUGUE. CAS QUADRATIQUE

## 4.1 Optimisation quadratique sans contraintes

### 4.1.1 Définition et théorèmes fondamentaux

**Définition 7** Soit  $Q$  est une matrice  $(n, n)$  symétrique et définie positive et  $b \in \mathbb{R}^n$ . On appelle problème de minimisation quadratique sans contraintes, le problème noté (PQSC)

$$\text{suivant : } \min_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} x^T Q x - b^T x \right\} \quad (\text{PQSC})$$

**Théorème 4.1.1** Le problème (PQSC) a une solution unique  $\hat{x}$ . solution du système linéaire  $Qx = b$ , c'est à dire que  $\hat{x}$  vérifie

$$\hat{x} = Q^{-1}b \quad (4.1)$$

**Preuve.** Considérons  $f(x) = \frac{1}{2} x^T Q x - b^T x$ . Notons  $H(x)$  la matrice Hessienne de  $f$  au point  $x$ . On a

$$H(x) = Q, \quad \forall x \in \mathbb{R}^n$$

Rappelons le théorème de l'Analyse convexe. Soit  $f : S \rightarrow \mathbb{R}$ ,  $S \subset \mathbb{R}^n$  convexe. Alors  $f$  est convexe sur  $S$  si et seulement si

$H(x)$  est semi définie positive pour tout  $x \in S$ . De plus si  $f$  est quadratique de la forme  $f(x) = \frac{1}{2} x^T Q x - b^T x$ . Alors  $f$  est

---

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

Pour  $f(x) = \frac{1}{2}x^T Qx - b^T x$ , on a  $H(x) = Q$ ,  $\forall x \in \mathbb{R}^n$ . Par hypothèse  $Q$  est définie positive, donc  $f(x) = \frac{1}{2}x^T Qx - b^T x$  est strictement convexe dans  $\mathbb{R}^n$ .

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

$$: \min_{x \in S} f(x) \quad (4.2)$$

Alors  $\hat{x} \in S$  est solution optimale de  $(PC)$  si et seulement si on a

$$\nabla f(\hat{x}) = 0 \quad (4.3)$$

De plus si  $\hat{x}$  existe, alors elle est unique.

Appliquons ce résultat à notre problème.  $f(x) = \frac{1}{2}x^T Qx - b^T x$  est strictement convexe.  $S = \mathbb{R}^n$  est convexe et ouvert.  $x \in \mathbb{R}^n$  est

solution optimale de  $(PQSC)$  si et seulement si  $\nabla f(x) = 0$ . Or  $\nabla f(x) = Qx - b$ . Donc

$$\nabla f(x) = 0 \iff Qx - b = 0 \iff Qx = b \quad (4.4)$$

Donc  $\hat{x}$  solution optimale de  $(PQSC)$  si et seulement si  $\hat{x}$  vérifie

$$Q\hat{x} = b. \quad (4.5)$$

$Q$  étant définie positive, alors  $Q$  est inversible. Par conséquent  $\hat{x}$  solution de  $(PQSC)$  est donnée par

$$\hat{x} = Q^{-1}b. \quad (4.6)$$

■

### 4.1.2 Calcul du pas obtenu par une recherche linéaire exacte

Soit  $Q$  est une matrice  $(n, n)$  symétrique et définie positive et  $b \in \mathbb{R}^n$  et  $f(x) = \frac{1}{2}x^T Qx - b^T x$ . Considérons le problème  $(PQSC)$

$$\min_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2}x^T Qx - b^T x \right\} \quad (PQSC)$$

Les méthodes à directions de recherches linéaires génèrent des suites  $\{x_k\}_{k=1,2,\dots}$  de la manière suivante. On démarre par  $x_1 \in \mathbb{R}^n$ . A l'itération  $k$  si on a  $x_k \in \mathbb{R}^n$ . Le successeur  $x_{k+1}$  de  $x_k$  est donné par la relation suivante

$$x_{k+1} = x_k + \alpha_k d_k \quad (4.7)$$

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

$$f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k) \quad (4.8)$$

Notons

$$g_k = \nabla f(x_k) = Qx_k - b \quad (4.9)$$

**Théorème 4.1.2** Soit  $Q$  est une matrice  $(n, n)$  symétrique et définie positive et  $b \in \mathbb{R}^n$  et  $f(x) = \frac{1}{2}x^T Qx - b^T x$ . Considérons le problème (PQSC)

$$\min_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2}x^T Qx - b^T x \right\} \quad (PQSC)$$

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

$$g_k^T d_k = (Qx_k - b)^T d_k < 0 \quad (4.10)$$

Soit  $\alpha_k > 0$  obtenue par une recherche linéaire exacte c'est à dire que  $\alpha_k$  vérifie

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

Alors

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k} \quad (4.11)$$

**Preuve.** Considérons

$$\varphi(\alpha) = f(x_k + \alpha d_k) \quad (4.12)$$

Donc  $\alpha_k$  est la solution optimale du problème unidimensionnel suivant

$$\varphi(\alpha_k) = \min_{\alpha > 0} \varphi(\alpha) \quad (4.13)$$

$Q$  définie positive. Alors  $f(x) = \frac{1}{2}x^T Qx - b^T x$  est strictement convexe sur  $\mathbb{R}^n$ . Par conséquent  $\varphi(\alpha) = f(x_k + \alpha d_k)$  est strictement

convexe sur  $]0, +\infty[$  qui est un ouvert convexe. Donc  $\alpha_k$  solution optimale de (4.12) si et seulement si  $\dot{\varphi}(\alpha_k) = 0$ . Or

$$\begin{aligned}\dot{\varphi}(\alpha) &= \nabla f(x_k + \alpha d_k)^T d_k = [Q(x_k + \alpha d_k) - b^T] d_k \\ &= [Qx_k - b + \alpha Qd_k]^T d_k = [g_k + \alpha Qd_k]^T d_k \\ &= g_k^T d_k + \alpha d_k^T Qd_k.\end{aligned}$$

Donc

$$\dot{\varphi}(\alpha) = 0 \iff \alpha d_k^T Qd_k = -g_k^T d_k$$

ou encore

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

■

**Remarque 4.1.1** Si  $Q$  n'est pas définie positive mais seulement semi définie positive, alors  $f$  n'est pas strictement convexe, et  $d_k^T Qd_k$  peut être égale à zéro. Par conséquent  $\alpha_k$  ne sera plus définie par (4.10).

## 4.2 Méthode des directions conjuguées

### 4.2.1 Définitions et propriétés générales

**Définition 8** Soit  $Q$  une matrice  $(n, n)$  symétrique. Les directions  $d_0, d_1, \dots, d_n$  sont dites  $Q$  conjuguées si on a

$$d_i^T Qd_j = 0, \quad 0 \leq i, j \leq k \quad (4.14)$$

**Théorème 4.2.1** Soit  $Q$  une matrice  $(n, n)$  symétrique et définie positive. Les directions  $d_0, d_1, \dots, d_k$ , avec  $k \leq n - 1$ , sont non nuls et  $Q$  conjuguées, alors ils sont linéairement indépendants.

---

**Preuve.** Supposons que

$$\alpha_0 d_0 + \alpha_1 d_1 + \dots + \alpha_k d_k = 0 \quad (4.15)$$

Multiplions l'égalité (4.15) par  $d_j^T Q$ ,  $0 \leq j \leq k$ , on obtient

$$0 = \sum_{i=1}^k \alpha_i d_j^T Q d_i = \alpha_j d_j^T Q d_j \quad (4.16)$$

or

$$\alpha_j d_j^T Q d_j = 0 \implies \alpha_j = 0$$

car  $Q$  est définie positive et par conséquent  $d_j^T Q d_j > 0$ . Donc le système  $\{d_0, d_1, \dots, d_k\}$  est libre. ■

## 4.2.2 L'Algorithme des directions conjuguées

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

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} x^T Q x - b^T x \right\} \quad (PQSC)$$

### Algorithme

#### Initialisation

On se donne  $x_0 \in \mathbb{R}^n$  quelconque et  $d_0, d_1, \dots, d_{n-1}, Q$  conjuguées. Poser  $k = 0$  et aller à l'étape principale

#### Etape principale

Pour  $k \geq 0$

Calculez

$$g_k = \nabla f(x_k) = Qx_k - b$$

Si  $g_k = 0$  Stop.

Sinon calculez

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

et

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

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

**Théorème 4.2.2** Partant d'une point initial  $x_0 \in \mathbb{R}^n$  quelconque, l'algorithme des directions conjuguées précédent converge vers la solution optimale unique  $\hat{x}$  du problème (PQSC) dans  $n$  itérations, c'est à dire qu'on a

$$x_n = \hat{x} \quad \text{et} \quad Qx_n = Q\hat{x} = b \quad (4.17)$$

**Remarque 4.2.1** Si on démarre du point  $x_1$ , alors la solution optimale est atteinte au point  $x_{k+1}$ , c'est à dire qu'on aura

$$\hat{x} = x_{n+1}$$

**Théorème 4.2.3** Soit  $Q$  une matrice  $(n, n)$  symétrique et définie positive et  $b \in \mathbb{R}^n$  et  $f(x) = \frac{1}{2}x^T Qx - b^T x$ . Considérons la suite  $\{x_k\}_{k=1,2,\dots}$  de la manière suivante. On démarre par  $x_1 \in \mathbb{R}^n$ . A l'itération  $k$ , Le successeur  $x_{k+1}$  de  $x_k$  est donné par la relation suivante

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

où  $d_k \in \mathbb{R}^n$  est une direction de recherche et  $\alpha_k \in \mathbb{R}^+$  est le pas de recherche obtenu par une recherche linéaire exacte,  $\alpha_k$  vérifie

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k) \quad (4.18)$$

Notons

$$g_{k+1} = \nabla f(x_{k+1}) = Qx_{k+1} - b \quad (4.19)$$

Alors

$$g_{k+1} = g_k + \alpha_k Qd_k \quad (4.20)$$

et

$$g_{k+1}^T d_k = 0, \quad k = 0, 1, \dots, n-1 \quad (4.21)$$

---

**Théorème 4.2.4** Dans la méthode des directions conjuguées, on a:

$$g_{k+1}^T d_i = 0, \quad k = 0, 1, \dots, n-1, \quad i = 0, \dots, k \quad (4.22)$$

## 4.3 Méthode du gradient conjugué. Cas quadratique

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

$$\min \{f(x) : x \in \mathbb{R}^n\} = \min \left\{ \frac{1}{2} x^T Q x - b^T x : x \in \mathbb{R}^n \right\} \quad (\text{PQSC})$$

### 4.3.1 Introduction

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

Dans la méthode du gradient conjugué, On démarre d'un point  $x_0 \in \mathbb{R}^n$ ,  $d_0 = -g_0 = -\nabla f(x_0) = -Qx_0 - b$ . les directions  $d_k$ ,  $k = 1, \dots, n-1$  sont calculés à chaque itération.

A l'itération  $k$ ,  $d_k = -g_k + \beta_{k-1} d_{k-1}$

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

$\beta_{k-1}$  est obtenu de sorte que  $d_k$  soit  $Q$  conjugué avec les autres vecteurs  $d_i$ ,  $i = 0, \dots, k-1$ . En d'autres termes on doit avoir

$$d_k^T Q d_i = 0 \quad i = 0, \dots, k-1. \quad (4.23)$$

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

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

b) Le mot conjugué est aussi justifié, car et comme on le verra plus loin, les directions  $\{d_k\}_{k=0}^{n-1}$  sont  $Q$  conjuguées.

---

### 4.3.2 Algorithme du gradient conjugué. Cas quadratique

#### Principe de l'Algorithme

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

Calculez  $d_0 = -g_0 = b - Qx_0$ ,  $\alpha_0 = -\frac{g_0^T d_0}{d_0^T Q d_0}$

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

$$g_k = Qx_k - b, \quad \alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}, \quad x_{k+1} = x_k + \alpha_k d_k, \quad \text{et} \quad g_{k+1} = Qx_{k+1} - b, \quad d_{k+1} = -g_{k+1} + \beta_k d_k \quad (4.24)$$

$\beta_k$  est choisi de sorte que

$$d_{k+1}^T Q d_k = 0 \quad (4.25)$$

Puisque  $d_{k+1} = -g_{k+1} + \beta_k d_k$ , alors (4.25) donne

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

ou encore

$$\beta_k d_k^T Q d_k = g_{k+1}^T Q d_k \quad (4.26)$$

et finalement

$$\beta_k = \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k} \quad (4.27)$$

#### Algorithme du gradient conjugué. Cas quadratique

1. choisir  $x_0 \in \mathbb{R}^n$ .
2. Calculez  $g_0 = Qx_0 - b$ . Si  $g_0 = 0$  stop. Sinon poser  $d_0 = -g_0$ . Poser  $k = 0$
3. Calculez  $\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}$ .
4. Calculez  $x_{k+1} = x_k + \alpha_k d_k$ .
5. Calculez  $g_{k+1} = Qx_{k+1} - b$ . si  $g_{k+1} = 0$  stop.
6. Calculez  $\beta_k = \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k}$ .
7. Calculez  $d_{k+1} = -g_{k+1} + \beta_k d_k$ .
8. Poser  $k = k + 1$  et allez à 3 .

---

### 4.3.3 Propriétés du gradient conjugué quadratique

La propriété fondamentale du gradient conjugué cas quadratique est que les directions  $\{d_k\}_{k=0}^{n-1}$  sont  $Q$  conjugués. Ces directions vérifient comme on l'a vu dans l'algorithme

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

avec

$$\beta_k = \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k}$$

D'après le théorème 4.2.1, l'algorithme du gradient conjugué, version quadratique converge vers la solution optimale en  $n$  itérations

Avant d'enoncer et de démontrer ce résultat, démontrons les lemmes suivants :

**Lemme 4.3.1** Notons par

$$g_k = \nabla f(x_k) = Qx_k - b$$

où  $x_k$ ,  $k = 0, \dots, n-1$  sont obtenus par l'algorithme du gradient conjugué quadratique. Alors on a

$$g_{k+1}^T g_j = 0, \quad k = 0, 1, \dots, n-1, \quad j = 0, 1, \dots, k$$

**Théorème 4.3.1** Les directions  $\{d_0, d_1, \dots, d_{n-1}\}$  engendrées par l'algorithme du gradient conjugué quadratique sont  $Q$  conjuguées.

### 4.3.4 Convergence de l'Algorithme du gradient conjugué quadratique

**Théorème 4.3.2** Soit  $Q$  une matrice  $(n, n)$ , symétrique et définie positive et (PQSC) le problème de minimisation sans contraintes quadratique suivant

$$\min \{f(x) : x \in \mathbb{R}^n\} = \min \left\{ \frac{1}{2}x^T Qx - b^T x : x \in \mathbb{R}^n \right\} \quad (\text{PQSC})$$

Démarrant d'un point  $x_0 \in \mathbb{R}^n$  quelconque, considérons la suite  $\{x_k\}$  générée par l'algorithme du gradient conjugué quadratique définie par

$$g_k = \nabla f(x_k) = Qx_k - b, \quad k = 0, 1, \dots$$

$$\beta_k = \frac{g_{k+1}^T Qd_k}{d_k^T Qd_k}, \quad k = 0, 1, \dots \quad (4.28)$$

$$d_k = \begin{cases} -g_0 & \text{si } k = 0 \\ -g_k + \beta_{k-1}d_{k-1} & \text{si } k \geq 1 \end{cases} \quad (4.29)$$

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Qd_k} \quad (4.30)$$

et

$$x_{k+1} = x_k + \alpha_k d_k, \quad (4.31)$$

Alors la suite  $\{x_k\}$  converge en  $n$  itérations vers la solution optimale  $\hat{x}$  du problème (PQSC), c'est à dire que  $x_n$  vérifie  $x_n = \hat{x}$  et

$$Q\hat{x} = Qx_n = b \quad (4.32)$$

**Preuve.** D'après le théorème 6, les directions  $\{d_0, d_1, \dots, d_{n-1}\}$  engendrées par l'algorithme du gradient conjugué

quadratique, sont  $Q$  conjuguées. L'algorithme du gradient conjugué est donc une méthode à directions conjuguées. D'autre

part la fonction  $f(x) = \frac{1}{2}x^T Qx - b^T x$  est strictement convexe et  $\mathbb{R}^n$  est un convexe ouvert. Donc d'après le théorème 3, la suite  $\{x_k\}$

converge en  $n$  itérations vers la solution optimale  $\hat{x}$  du problème (PQSC), c'est à dire que  $x_n$  vérifie  $x_n = \hat{x}$  et

$$Q\hat{x} = Qx_n = b.$$

■

## CHAPITRE 5

# GRADIENT CONJUGUE NON QUADRATIQUE

### 5.1 Introduction et différentes formes du gradient conjugué

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . non quadratique .On cherche à résoudre le problème non quadratique sans contraintes suivant

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

Parmi les plus anciennes méthodes utilisées pour résoudre les problèmes du type  $(PNQSC)$  on peut citer la méthode du gradient conjugué. Cette méthode est surtout utilisée pour résoudre les problèmes de grande taille.

Cette méthode à été découverte en 1952 par Hestene et Steifel ([25]), pour la minimisation des fonctions quadratiques strictement convexes.

Plusieurs mathématiciens ont étendus cette méthode pour le cas non quadratique . Ceci a été réalisé pour la première fois, en 1964 par Fletcher et Reeves ([17]) puis en 1969 par Polak , Ribière ([15]) et Polyak ([8])

D'autres variantes on été étudiées plus tard. Une autre variante a été étudiée en 1987 par Fletcher

Toutes ces méthodes génèrent une suite de la façon suivante :

$$x_{k+1} = x_k + \alpha_k d_k \quad (5.1)$$

Le pas  $\alpha_k \in \mathbb{R}$  est déterminé par une optimisation unidimensionnelle ou recherche linéaire exacte ou inexacte du type Armijo, Goldstein ou Wolfe.

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

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

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

Les différentes valeurs attribuées à  $\beta_k$  définissent les différentes formes du gradient conjugué

Si on note

$$y_k = g_{k+1} - g_k, \quad S_k = x_{k+1} - x_k \quad (5.3)$$

On obtient les variantes suivantes :

$$\begin{aligned} \beta_{k+1}^{HS} &= \frac{g_{k+1}^T y_{k+1}}{d_k^T y_k} && \text{Gradient conjugué variante Hestenes-Stiefel (HS) en 1952} \\ \beta_{k+1}^{FR} &= \frac{\|g_{k+1}\|^2}{\|g_k\|^2} && \text{Gradient conjugué variante Fletcher Reeves (FR) en 1964} \\ \beta_{k+1}^{PRP} &= \frac{g_{k+1}^T y_{k+1}}{\|g_k\|^2} && \text{Gradient conjugué variante Polak-Ribière-Polyak (PRP) en 1969} \\ \beta_{k+1}^{CD} &= \frac{\|g_{k+1}\|^2}{d_k^T g_k} && \text{Gradient conjugué variante descente conjuguée (CD) en 1987} \\ \beta_{k+1}^{LS} &= \frac{g_{k+1}^T y_k}{d_k^T g_k} && \text{Gradient conjugué variante de Dai- Yuan (Liu-Storey) en 1991} \\ \beta_{k+1}^{CD} &= \frac{\|g_{k+1}\|^2}{d_k^T g_k} && \text{Gradient conjugué variante de Dai-Yuan (DY) en 1999} \\ \beta_{k+1}^{HZ} &= \left( y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k} \right) \frac{g_{k+1}}{d_k^T y_k} && \text{Gradient conjugué variante Hager-Zhang (HZ) en 2005} \\ \beta_{k+1}^{RMIL} &= \frac{g_{k+1}^T (g_{k+1} - g_k)}{\|d_k\|^2} && \text{Gradient conjugué variante Rivaie-Mustafa-Ismail-Lelong (RMIL) en 2010} \end{aligned}$$

**Remarque 5.1.1** Dans le cas où  $f$  n'est pas quadratique on a

$$\beta_k^{HS} \neq \beta_k^{FR} \neq \beta_k^{PRP} \neq \beta_k^{CD} \neq \beta_k^{LS} \neq \beta_k^{DY} \neq \beta_k^{HZ} \neq \beta_k^{RMIL} \quad (5.4)$$

Par conséquent, en appliquant l'algorithme du gradient conjugué non quadratique, en utilisant les coefficients  $\beta_k$  figurant dans (5.4), on obtient des suites  $\{x_k\}_{k \in \mathbb{N}}$  différentes.

Que se passe-t-il si  $f$  est quadratique strictement convexe et si  $\alpha_k$  est obtenue par une recherche linéaire exacte. La réponse à cette question se trouve dans le théorème suivant

**Théorème 5.1.1** Si  $f(x) = \frac{1}{2}x^T Qx - b^T x$ , avec  $Q$  symétrique définie positive,  $x \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^n$  et si  $\alpha_k$  est obtenue par une recherche linéaire exacte. Notons

$$\beta_k = \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k},$$

alors on a

$$\beta_k = \beta_k^{HS} = \beta_k^{FR} = \beta_k^{PRP} = \beta_k^{CD} = \beta_k^{LS} = \beta_k^{DY} = \beta_k^{HZ} = \beta_k^{RMIL} \quad (5.5)$$

et l'algorithme du gradient conjugué quadratique génère la même suite  $\{x_k\}_{k \in \mathbb{N}}$ .

**Preuve.**

a) Montrons que  $\beta_k = \beta_k^{HS}$

En effet. Rappelons que si  $(x) = \frac{1}{2}x^T Qx - b^T x$ , avec  $Q$  symétrique définie positive,  $x \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^n$  et si  $\alpha_k$  est obtenue par une recherche linéaire exacte, alors on a

$$\beta_k = \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k} \quad (5.6)$$

On va transformer (5.6) de différentes façons de sorte que la matrice  $Q$  disparaisse. D'après la formule

[(4.20) -Théorème4.2.3-Chapitre4] on a

$$g_{k+1} - g_k = \alpha_k Q d_k \quad (5.7)$$

(5.7) implique

$$Q d_k = \frac{g_{k+1} - g_k}{\alpha_k} \quad (5.8)$$

Par conséquent (5.6) devient

$$\beta_k = \frac{g_{k+1}^T \frac{g_{k+1} - g_k}{\alpha_k}}{d_k^T \frac{g_{k+1} - g_k}{\alpha_k}} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)} = \frac{g_{k+1}^T y_k}{d_k^T y_k} = \beta_k^{HS} \quad (5.9)$$

**b) Montrons que  $\beta_k^{HS} = \beta_k^{FR}$**

En effet. Puisque

$$d_k = -g_k + \beta_{k-1} d_{k-1} \quad (5.10)$$

alors

$$\beta_k^{HS} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)} = \frac{g_{k+1}^T g_{k+1} - g_{k+1}^T g_k}{(-g_k + \beta_{k-1} g_{k+1}^T d_{k-1})^T (g_{k+1} - g_k)} \quad (5.11)$$

$$= \frac{g_{k+1}^T g_{k+1} - g_{k+1}^T g_k}{-g_{k+1}^T g_k + g_k^T g_k + \beta_{k-1} g_{k+1}^T d_{k-1} - \beta_{k-1} g_k^T d_{k-1}} \quad (5.12)$$

Or d'après le lemme 4.3.1-Chapitre 4

, on a

$$g_{k+1}^T g_k = 0 \quad (5.13)$$

et d'après la relation [(4.22)-Chapitre 4-Théorème 4.2.4] on a

$$g_{k+1}^T d_{k-1} = 0 \quad \text{et} \quad g_k^T d_{k-1} = 0. \quad (5.14)$$

Par conséquent (5.11), (5.12) et (5.14) donnent

$$\beta_k^{HS} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} = \beta_k^{FR} \quad (5.15)$$

**c) Montrons que  $\beta_k^{HS} = \beta_k^{PRP}$**

En effet. On a

$$\beta_k^{HS} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)} = \frac{g_{k+1}^T y_k}{(-g_k + \beta_{k-1} d_{k-1})^T (g_{k+1} - g_k)} = \frac{g_{k+1}^T y_k}{g_k^T g_k} = \beta_k^{PRP} \quad (5.16)$$

■

### Exercice 1

Montrez que

$$\beta_k = \beta_k^{CD} = \beta_k^{LS} = \beta_k^{DY} = \beta_k^{HZ} = \beta_k^{RMIL}$$

---

## Algorithme du gradient conjugué non quadratique.

### Introduction.

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  non quadratique et (PNQSC) le problème de minimisation non quadratique sans contraintes suivant :

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

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

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

L'algorithme démarre d'un point  $x_0 \in \mathbb{R}^n$  quelconque.

### A l'itération $k$

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

**Calcul de  $d_k$  :**

$$d_k = -\nabla f(x_k) + \beta_{k-1} d_{k-1} \quad (5.17)$$

On a huit façons pour calculer  $\beta_{k-1}$

$$\beta_{k-1}^{HS} = \frac{\nabla f(x_k)^T (\nabla f(x_k) - \nabla f(x_{k-1}))}{d_{k-1}^T (\nabla f(x_k) - \nabla f(x_{k-1}))}$$

$$\beta_{k-1}^{PR} = \frac{\nabla f(x_k)^T (\nabla f(x_k) - \nabla f(x_{k-1}))}{\|\nabla f(x_{k-1})\|^2}$$

$$\beta_{k-1}^{FR} = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}$$

$$\beta_{k-1}^{DY} = \frac{\|\nabla f(x_k)\|^2}{d_{k-1}^T (\nabla f(x_k) - \nabla f(x_{k-1}))}$$

$$\beta_{k-1}^{LS} = -\frac{\nabla f(x_k)^T (\nabla f(x_k) - \nabla f(x_{k-1}))}{d_{k-1}^T \nabla f(x_{k-1})}$$

---


$$\beta_{k-1}^{CD} = -\frac{\|\nabla f(x_k)\|^2}{d_{k-1}^T \nabla f(x_{k-1})}$$

$$\beta_{k-1}^{HZ} = \left( (\nabla f(x_k) - \nabla f(x_{k-1})) - 2d_{k-1} \frac{\|\nabla f(x_k) - \nabla f(x_{k-1})\|^2}{d_{k-1}^T (\nabla f(x_k) - \nabla f(x_{k-1}))} \right)^T \frac{\nabla f(x_k)}{(\nabla f(x_k) - \nabla f(x_{k-1}))^T y_{k-1}}$$

$$\beta_{k-1}^{RMIL} = \frac{(\nabla f(x_k))^T (\nabla f(x_k) - \nabla f(x_{k-1}))}{\|d_{k-1}\|^2}$$

### Calcul de $\alpha_k$

Ayant obtenu  $d_k$ , rappelons que  $\alpha_k$  vérifie

$$f(x_k + \alpha_k d_k) = \min \{f(x_k + \alpha d_k) : \alpha \in ]0, +\infty[ \}. \quad (5.18)$$

Dans le cas où  $f(x) = \frac{1}{2}x^T Qx - b$ ,  $Q$  symétrique définie positive,  $\alpha_k$  solution de (5.18), est donnée par la relation suivante

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}. \quad (5.19)$$

Dans le cas où  $f$  n'est pas quadratique,  $\alpha_k$  ne peut pas être calculée par la formule (5.19). On calcule dans ce cas  $\alpha_k$  par d'autres méthodes. On utilise par exemple la méthode du nombre d'or ou la méthode de dichotomie. Comme on le verra plus loin,  $\alpha_k$  peut être calculée par une recherche linéaire inexacte d'Armijo ou Goldstein ou Wolfe.

### Algorithme

#### Etape1

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

#### Etape2

Poser  $k = 0$

Calculez  $g_0 = \nabla f(x_0)$ . Posez  $d_0 = -g_0$

#### Etape3

Calculez  $\alpha_k$  en utilisant une recherche linéaire exacte ou inexacte d'Armijo ou de Goldstein ou de Wolfe ou de Wolfe Forte

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

#### Etape4

Si  $\|\nabla f(x_{k+1})\| < \varepsilon$ , Stop,  $x^* = x_{k+1}$ . Sinon allez à Etape5

#### Etape5

Calculez  $g_{k+1} = \nabla f(x_{k+1})$

---

Calculez  $\beta_k$  par l'une des manières suivantes

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

Calculez

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

Posez  $k = k + 1$  et allez à Etape3

## CHAPITRE 6

# SYNTHESE DES RESULTATS DE CONVERGENCES DES METHODES DU GRADIENT CONJUGUE NON QUADRATIQUE

Dans ce chapitre on va essayer de présenter une synthèse sur les différents résultats de convergence des méthodes du gradient conjugué pour la minimisation des fonctions sans contraintes. Ces méthodes seront utilisées avec une recherche linéaire inexacte. L'analyse couvre quatre classes des méthodes qui sont globalement convergentes pour des fonctions régulières non nécessairement convexes. Dans la première famille, ce sont certaines propriétés de la méthode Fletcher-Reeves qui jouent un rôle crucial, tandis que la seconde famille concerne la méthode de Polak-Ribière-Polyak qui a une propriété importante. La troisième concerne la méthode de la descente conjuguée et dans la dernière famille on va présenter quelques propriétés de Dai-Yan. On terminera ce chapitre par quelques développements récents de la méthode du gradient conjugué.

### 6.1 Hypothèses C1 et C2 et théorème de Zoutendijk

Hypothèses C1 et C2 (de Lipschitz et de bornitude).

Rappelons dans ce qui suit les deux conditions suivantes que toutes les méthodes du gradient conjugué doivent vérifier (ou moins l'une d'entre elles) pour prouver les résultats de convergence.

**Condition C1 :**

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . On dit que  $f$  vérifie la condition C1 si  $f$  est continument différentiable dans un voisinage  $V(\Gamma)$  de  $\Gamma = \{x \in \mathbb{R}^n : f(x) \leq f(x_1) \quad x_1 \in \mathbb{R}^n \text{ point initial.}\}$  et si

$\nabla f(x)$  vérifier la condition de Lipschitz dans  $V(\Gamma)$ , c'est à dire, il existe une constante  $L$  telle que

$$\|g(x) - g(y)\| \leq L \|x - y\|, \quad \text{pour tout } x, y \in V(\Gamma).$$

**Condition C2 :**

i) L'ensemble  $\Gamma = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$  est borné, i e, il existe une constante  $B < \infty$  telle que

$$\|x\| \leq B, \quad \forall x \in \Gamma$$

**Théorème de Zoutendijk).**

L'outil de base utilisé par les différentes variantes du gradient conjugué avec une recherche linéaire inexacte est le théorème suivant de Zoutendijk

**Théorème 6.1.1 (Zoutendijk) ([23])** *Considérons une méthode itérative quelconque générant une suite  $\{x_k\}$  de la forme :*

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

*$d_k$  étant une direction de descente et  $\alpha_k$  est une recherche linéaire inexacte de Wolfe. Supposons aussi que  $f$  vérifie la Condition C1. Alors on a*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \tag{6.1}$$

**Utilisation du théorème de Zoutendijk pour démontrer la convergence globale).**

La condition  $\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty$  est équivalente à  $\sum_{k=1}^{\infty} \cos^2(\theta_k) \|g_k\|^2 < \infty$  où  $\theta_k$  est l'angle que fait  $d_k$  avec  $-g_k$ . Il est évident de voir que si

$$\cos(\theta_k) \geq \delta > 0$$

pour tout  $k$ , alors on a

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

Dans les différentes méthodes du gradient conjugué on n'arrive pas à démontrer le résultat

précédent i.e.,  $\lim_{k \rightarrow \infty} \|g_k\| = 0$ , mais seulement

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (6.2)$$

La condition (6.2) implique qu'il existe une sous suite de  $\{\|g_k\|\}$  qui tend vers zéro.

### Conditions suffisantes associés au théorème de Zoutendijk pour démentrer la relation (6.2)

Pour démontrer (6.2) on associe au théorème de Zoutendijk les deux hypothèses suivantes :

#### Hypothèse1

La descente suffisante est assurée, i.e., il existe une constante  $C$  indépendante de  $k$  telle que

$$g_k^T d_k \leq -C \|g_k\|^2$$

#### Hypothèse 2

Il existe une constante  $\beta$  telle que

$$\|d_k\|^2 \leq \beta k$$

On a donc

$$\left\{ \begin{array}{l} \text{Relation de Zoutendijk } \left( \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \right) \\ \text{Hypothèse1} \\ \text{Hypothèse2} \end{array} \right\} \implies \left\{ \liminf_{k \rightarrow \infty} \|g_k\| = 0 \right\}$$

### Remarque importante

► La condition (**Hypothèse1**) est plus forte que la descente. En effet

$$g_k^T d_k \leq -c \|g_k\|^2 \implies g_k^T d_k < 0.$$

► La condition (**Hypothèse1**) n'est pas nécessaire pour avoir  $\liminf_{k \rightarrow \infty} \|g_k\| = 0$ . Dai et Yuan ([68]) ont montré la convergence globale seulement avec la condition de descente :

$$g_k^T d_k < 0.$$

► D'autre part la condition (**Hypothèse1**), i.e.  $g_k^T d_k \leq -C \|g_k\|^2$  n'est pas suffisante

---

toute seule, pour avoir la convergence globale.

**Définition 9** Nous dirons qu'une méthode de gradient conjugué converge globalement si l'une des deux conditions suivantes est vérifiée

- a)  $g_{k_0} = 0$ , pour un certain  $k_0 \in \mathbb{N}$
- b)  $\liminf_{k \rightarrow \infty} \|g_k\| = 0$ .

Une autre version du théorème de Zoutendijk se trouve dans ([23]). Ce sera l'objet du théorème suivant

**Théorème 6.1.2 [Zoutendijk bis] ([23])** Considérons une méthode itérative quelconque générant une suite  $\{x_k\}$  de la forme :

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

$d_k$  étant une direction de descente et  $\alpha_k$  est une recherche linéaire inexacte de Wolfe forte. Supposons aussi que  $f$  vérifie la Condition C1.

Alors on a

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

ou

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty$$

## 6.2 Rôle joué par la direction de recherche initiale

Commençons d'abord par donner un petit aperçu sur le rôle joué par la direction de recherche initiale.

Il est essentiel de prendre  $d_0 = -g_0$  dans un algorithme de CG. En 1972, Crowder et Wolfe [26] ont donné un exemple en 3 dimensions qui montre que le taux de convergence est linéaire si la direction de la recherche initiale n'est pas la direction de la plus grande pente, même pour une fonction quadratique fortement convexe. En 1976, Powell [45] a obtenu un résultat encore plus fort, il a montré que si la fonction objectif est une fonction quadratique convexe et si la direction de recherche initiale est une direction de descente arbitraire, alors, soit la condition d'optimalité est obtenue dans  $n + 1$  itérations, ou la

vitesse de convergence est seulement linéaire. En outre, en analysant la relation entre  $x_0$  et  $d_0$ , il s'ensuit que la convergence linéaire est plus fréquente que la convergence finie.

Afin de parvenir à une convergence finie pour une direction de recherche initiale arbitraire, Nazareth [34] a proposé un algorithme CG basée sur une récurrence à trois termes :

$$d_{k+1} = -y_k + \frac{y_k^T y_k}{d_k^T y_k} d_k + \frac{y_{k-1}^T y_k}{d_{k-1}^T y_{k-1}} d_{k-1} \quad (6.3)$$

avec  $d_{-1} = 0$  et  $d_0$  une direction de descente arbitraire. Si  $f$  est une fonction quadratique convexe, alors pour tout  $\alpha_k$ , les directions de recherche générés par (6.3) sont conjugués par rapport au Hessian de  $f$ . Toutefois, cette innovation intéressante n'a pas vu une utilisation importante dans la pratique.

Les méthodes dans lesquelles le terme  $\|g_{k+1}\|^2$  figure dans le numérateur de  $\beta_k$

Les méthodes **FR**, **DY** et **CD** ont toutes comme numérateur commun le terme  $\|g_{k+1}\|^2$ . Une différence fondamentale qui caractérise ces méthodes par rapport aux autres méthodes du gradient conjugué où  $\beta_k$  est calculé différemment, est que les théorèmes de convergence globale nécessitent seulement la condition de Lipschitz : **Condition C1** et n'ont pas besoin de la condition de bornétude **Condition C2**.

#### Méthode de Fletcher-Reeves

Cette méthode été découverte en 1964 par Fletcher et Reeves ([17]),  $\beta_k$  est égale à :

$$\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \quad (6.4)$$

### Synthèse des principaux résultats de convergence pour la méthode de Fletcher-Reeves

Le premier résultat de convergence globale de la méthode de **FR** a été donnée par Zoutendijk ([23]) en 1970. Il a prouvé que la méthode Fletcher-Reeves converge globalement quand  $\alpha_k$  obtenu par une recherche linéaire exacte, en d'autres mots. En 1977, Powell a fait remarquer que la méthode de Fletcher-Reeves, avec une recherche linéaire exacte, était sensible numériquement. Autrement dit, l'algorithme pourrait prendre de nombreuses mesures courtes sans faire d'importants progrès au minimum. La mauvaise performance de la méthode de **FR** dans les applications a été souvent attribuée à ce phénomène de brouillage.

Le premier résultat de convergence globale de la méthode de **FR** pour une recherche linéaire inexacte a été donné par Al- Baali ([1]) en 1985. En utilisant la recherche linéaire inexacte de Wolfe forte avec  $\sigma < \frac{1}{2}$ , il a prouvé que la méthode **FR** génère des directions

---

de descente suffisantes. Plus précisément, il a prouvé que :

$$\frac{1 - 2\sigma + \sigma^{k+1}}{1 - \sigma} \leq \frac{-g_k^T d_k}{\|g_k\|^2} \leq \frac{1 - \sigma^{k+1}}{1 - \sigma}, \quad \text{pour tout } k \geq 0$$

Comme conséquence, la convergence globale a été démontrée en utilisant la condition Zoutendijk. Pour  $\sigma = 1/2$ ,  $d_k$  est une direction de descente, toutefois, l'analyse n'a pas établi la descente suffisante.

Touati Ahmed et Story ([14]) ont généralisé ce résultat pour

$$0 \leq \beta_k \leq \beta_k^{FR} \tag{6.5}$$

Gillbert et Nocedal ([30]) ont généralisé ce résultat pour

$$|\beta_k| \leq \beta_k^{FR} \tag{6.6}$$

Dans Liu et al. ([80]), la preuve de la convergence d'Al-Baali est étendue au cas  $\sigma = 1/2$ . Dai et Yuan ([78]) ont montré que dans les versions **FR** consécutives, au moins une itération satisfait la propriété de descente suffisante. En d'autres termes,

$$\max \left\{ \frac{-g_k^T d_k}{\|g_k\|^2}, \frac{-g_{k-1}^T d_{k-1}}{\|g_{k-1}\|^2} \right\} \geq \frac{1}{2}$$

Le théorème de Zoutendijk bis peut également être utilisé pour obtenir un résultat de convergence globale pour les méthodes **FR** mis en œuvre avec une recherche inexacte de Wolfe forte et  $\sigma \leq 1/2$ , puisque les directions de recherche sont toujours des directions de descente.

Dans ([73]), Dai et Yuan montrent qu'avec la méthode de **FR**, les conditions Wolfe fortes n'engendrent pas en général des directions de descente quand  $\sigma > 1/2$ , même pour la fonction  $f(x) = \lambda \|x\|^2$ , où  $\lambda > 0$  est une constante. Par conséquent, la contrainte  $\sigma \leq 1/2$  doit être imposée pour assurer la descente. Dans les implémentations typiques des conditions Wolfe, il est souvent plus efficace de choisir  $\sigma$  proche de 1. Par conséquent, la contrainte  $\sigma \leq 1/2$ , nécessaire pour assurer la descente, représente une restriction importante dans le choix des paramètres de la recherche linéaire. D'autre part, Dai et Yuan montrent dans [71] que, lorsque  $\sigma > 1/2$  et  $g_k^T d_k > 0$ ,  $-d_k$  peut être utilisé pour une direction de recherche, et si  $g_k^T d_k = 0$ , alors la recherche linéaire peut être ignorée par le choix  $x_{k+1} = x_k$ . S'il existe une constante  $\gamma$  telle que  $\|g_k\| \leq \gamma$ , sous la condition de Lipschitz, la méthode de **FR**, avec une recherche linéaire de Wolfe et ces ajustements spéciaux quand  $g_k^T d_k \geq 0$ , est globalement convergente.

Dans ([84]), la recherche linéaire de Wolfe forte est étendue à une recherche linéaire de Wolfe faible. La convergence globale est obtenue lorsque  $\sigma_1 + \sigma_2 \leq 1$ . Pour une recherche linéaire de Wolfe forte,  $\sigma_1 = \sigma_2 = \sigma$ , dans ce cas, la contrainte  $\sigma_1 + \sigma_2 \leq 1$  implique que  $\sigma \leq 1/2$ . Par conséquent, la condition  $\sigma_1 + \sigma_2 \leq 1$  est plus faible que la contrainte de Wolfe forte  $\sigma \leq 1/2$ . Et il est possible de prendre  $\sigma_1$  proche de 1, en prenant  $\sigma_2$  près de 0

### **Résultats d'El Baali**

Comme on l'a remarqué auparavant le premier et le plus important résultat de convergence de la méthode de Fletcher Reeves (FR), avec une recherche linéaire inexacte est celui d'El Baali ([1]). Nous exposons dans ce qui suit en détail ses résultats.

**Théorème 6.2.1 (El Baali)** ([1]) *Supposons que L'hypothèse Condition C1 soit satisfaite. Considérons une méthode du type (5.1) et (5.2) avec  $\beta_k = \beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$  et le pas  $\alpha_k$  satisfaisant à la règle de Wolfe forte (3.4), (3.6) où  $\sigma \in \left] 0, \frac{1}{2} \right[$ . Alors on a*

$$\frac{-1}{1-\sigma} \leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma-1}{1-\sigma}; \quad k = 1, \dots \quad (6.7)$$

**Preuve.** La démonstration se fait par récurrence

① Pour  $k = 1$  :

$$\frac{d_1^T g_1}{\|g_1\|^2} = \frac{-\|g_1\|^2}{\|g_1\|^2} = -1$$

D'autre part :

$$0 < \sigma < \frac{1}{2} \Rightarrow \begin{cases} \frac{-1}{1-\sigma} \leq -1 \\ \frac{2\sigma-1}{1-\sigma} \geq -1 \end{cases}$$

② Supposons que (6.7) est satisfaite pour  $k > 1$  et démontrons qu'elle le sera pour  $k + 1$  :

Supposons que :

$$\frac{-1}{1-\sigma} \leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma-1}{1-\sigma}; \quad k = 1, \dots \quad (6.8)$$

On a :

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

D'autre part, on aura :

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

d'où

$$\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} = -1 + \frac{\beta_{k+1}}{\beta_{k+1}^{FR}} \frac{d_k^T g_{k+1}}{\|g_k\|^2} \quad (6.9)$$

En utilisant la condition de la recherche linéaire (3.6) on aura :

$$|d_k^T g_{k+1}| \leq -\sigma d_k^T g_k \Rightarrow \sigma |\beta_{k+1}| d_k^T g_k \leq \beta_{k+1} d_{k+1}^T g_{k+1} \leq -\sigma |\beta_{k+1}| d_k^T g_k$$

soit en remplaçant ceci dans (6.9), on obtient :

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

De (6.8) on aura :

$$-1 - \frac{|\beta_{k+1}| \sigma}{\beta_{k+1}^{FR} (1 - \sigma)} \leq \frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} \leq -1 + \frac{|\beta_{k+1}| \sigma}{\beta_{k+1}^{FR} (1 - \sigma)}$$

et de (6.6) :

$$\frac{-1}{1 - \sigma} < \frac{|\beta_{k+1}|}{\beta_{k+1}^{FR}} < 1$$

Par conséquent on aura :

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

Ce qui achève la démonstration. ■

**Hypothèse 6.2.1** *Sous les mêmes hypothèses et conditions du Théorème précédent, on a pour tout  $k \geq 1$ ;  $d_k$  est une direction de descente suffisante i.e., il existe une constante  $C > 0$  telle que*

$$g_k^T d_k \leq -C \|g_k\|^2$$

**Preuve.** On a d'après (6.7)

$$\begin{aligned} \frac{-1}{1 - \sigma} &\leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma - 1}{1 - \sigma} \\ \Rightarrow d_k^T g_k &\leq \frac{2\sigma - 1}{1 - \sigma} \|g_k\|^2 = -\frac{1 - 2\sigma}{1 - \sigma} \|g_k\|^2 = -C \|g_k\|^2, \end{aligned}$$

avec  $C = \frac{1 - 2\sigma}{1 - \sigma}$  ■

**Théorème 6.2.2** ([1]) Supposons que L'hypothèse **Condition C1** soit satisfaite. Considérons une méthode du type (5.1) et (5.2) avec  $\beta_k = \beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$  et le pas  $\alpha_k$  satisfaisant à la règle de Wolfe forte (3.4)-(3.6) où  $\sigma \in \left]0, \frac{1}{2}\right[$ . Alors cette méthode est globalement convergente, dans le sens suivant :

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad (6.10)$$

**Preuve.** Puisque les conditions du [théorème 6.3.1] sont satisfaites alors on a :

$$\frac{-1}{1 - \sigma} \leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma - 1}{1 - \sigma} \Rightarrow -\sigma d_{k-1}^T g_{k-1} \leq \frac{\sigma}{1 - \sigma} \|g_{k-1}\|^2$$

D'autre part de (3.6)

$$|d_k^T g_{k+1}| \leq -\sigma d_k^T g_k \Rightarrow |d_{k-1}^T g_k| \leq -\sigma d_{k-1}^T g_{k-1}$$

d'où

$$|d_{k-1}^T g_k| \leq -\sigma d_{k-1}^T g_{k-1} \leq \frac{\sigma}{1 - \sigma} \|g_{k-1}\|^2 \quad (6.11)$$

De (6.7), (5.2) et (6.11) :

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

Posons  $\hat{\sigma} = \frac{1 + \sigma}{1 - \sigma} \geq 1$ , on aura :

$$\begin{aligned} \|d_k\|^2 &\leq \hat{\sigma} \|g_k\|^2 + (\beta_k^{FR})^2 \|d_{k-1}\|^2 \\ &\leq \hat{\sigma} \|g_k\|^2 + (\beta_k^{FR})^2 \left[ \hat{\sigma} \|g_{k-1}\|^2 + (\beta_k^{FR})^2 \|d_{k-2}\|^2 \right] \end{aligned}$$

---


$$\leq \hat{\sigma} \|g_k\|^4 \sum_{j=2}^k \|g_j\|^{-2} + \hat{\sigma} \|g_k\|^4 \|g_1\|^{-2} = \hat{\sigma} \|g_k\|^4 \sum_{j=1}^k \|g_j\|^{-2} \quad (6.12)$$

Supposons que  $g_k$  est borné en dehors du zéro ( $\lim_{k \rightarrow \infty} \inf \|g_k\| \neq 0$ ), c'est-à-dire :

$$\|g_k\| \geq \omega > 0; \forall k \Rightarrow \|g_k\|^{-2} \leq \omega^{-2}$$

de (6.12) on a :

$$\begin{aligned} \|d_k\|^2 &\leq \hat{\sigma} \|g_k\|^4 \sum_{j=1}^k \|g_j\|^{-2} \leq \hat{\sigma} \frac{\lambda^4}{\omega^2} \sum_{j=1}^k 1 \\ &\Rightarrow \|d_k\|^2 \leq \hat{\sigma} \frac{\lambda^4}{\omega^2} k \end{aligned}$$

d'où

$$\sum_{k \geq 1} \frac{1}{\|d_k\|^2} \geq \frac{\omega^2}{\hat{\sigma} \lambda^4} \sum_{k \geq 1} \frac{1}{k} > \infty \quad (6.13)$$

Ce qui veut dire que  $\sum_{k \geq 1} \frac{1}{\|d_k\|^2}$  est divergente.

D'autre part, puisque les conditions du Théorème 4.1, et du théorème 4.2 sont satisfaites, alors on a :

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

et

$$c_1 \frac{\|g_k\|}{\|d_k\|} \leq \cos \theta_k \leq c_2 \frac{\|g_k\|}{\|d_k\|}$$

d'où

$$\begin{aligned} \sum_{k \geq 1} c_1^2 \frac{\|g_k\|^2}{\|d_k\|^2} \|g_k\|^2 &\leq \sum_{k \geq 1} \cos^2 \theta_k \|g_k\|^2 < \infty \\ &\Rightarrow \sum_{k \geq 1} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty \\ &\Rightarrow \sum_{k \geq 1} \frac{\omega^4}{\|d_k\|^2} < \infty \\ &\Rightarrow \sum_{k \geq 1} \frac{1}{\|d_k\|^2} < \infty \end{aligned}$$

---

Ce qui contredit (6.13), d'où le résultat. Donc

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

■

**Remarque 6.2.1** *La méthode de Fletcher-Reeves avec une recherche linéaire exacte génère des directions de descente.*

En effet, à chaque itération  $k \geq 1$ , on a :

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

Puisque

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

Donc  $\alpha_k$  vérifie la condition nécessaire d'optimalité :

$$h'_k(\alpha_k) = d_k^T \nabla f(x_k + \alpha_k d_k) = d_k^T g_{k+1} = 0, \quad \forall k \geq 1$$

### Méthode de descente conjuguée

Cette méthode a été découverte en 1987 par Fletcher ([57]),  $\beta_k$  est égale à :

$$\beta_{k+1}^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k} \quad (6.14)$$

La méthode de descente conjuguée **CD** proposée par Fletcher ([57]) est étroitement liée à la méthode Fletcher Reeves **FR**. Avec une recherche linéaire exacte,  $\beta_k^{FR} = \beta_k^{CD}$ . Une différence importante entre **FR** et **CD** est que, avec **CD**, la descente suffisante est assurée pour une recherche linéaire de Wolfe forte et la contrainte  $\sigma \leq 1/2$  avec **FR**, n'est pas nécessaire pour la méthode **CD**. En outre, pour une recherche linéaire qui satisfait aux conditions de Wolfe relaxée (3.7)-(3.8) avec  $\sigma_1 \leq 1$  et  $\sigma_2 = 0$ , il peut être démontré que :

$$0 \leq \beta_k^{CD} \leq \beta_k^{FR}.$$

Par conséquent, à partir de l'analyse ([1]) ou le théorème de Zoutendijk bis, la convergence globale est assurée. D'autre part, si  $\sigma_1 \geq 1$  ou  $\sigma_2 > 0$ , Dai et Yuan [66] construisent

des exemples où  $\|d_k\|^2$  augmente de façon exponentielle et le procédé de CD converge vers un point où le gradient ne s'annule pas. En particulier, la méthode **CD** peut ne pas converger vers un point stationnaire pour une recherche linéaire de Wolfe forte.

**Théorème 6.2.3** ([57]) *Supposons que l'hypothèse **Condition C1** soit satisfaite. Considérons une méthode du type (5.1) et (5.2) avec*

*$\beta_k = \beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k}$  et le pas  $\alpha_k$  satisfait aux conditions de Wolfe relaxée (3.7)-(3.8) :*

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

*avec  $0 < \rho < \sigma_1 < 1$*

*et  $0 \leq \sigma_2 \leq 1$ . Alors la méthode génère des directions de descente suffisante à chaque itération  $k \geq 1$ .*

**Preuve.** On a

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

D'autre part de (3.8), on a

$$\begin{aligned} \sigma_1 d_k^T g_k &\leq d_k^T g_{k+1} \leq -\sigma_2 d_k^T g_k \\ \Rightarrow 1 - \sigma_2 &\leq 1 + \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} \leq 1 + \sigma_1 \end{aligned}$$

d'où

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

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

$$d_k^T g_k \leq -C \|g_k\|^2 \quad \text{où } C = 1 - \sigma_2 > 0$$

et donc  $d_k$  est une direction de descente suffisante. ■

**Théorème 6.2.4** *Supposons que L'hypothèse **Condition C1** soit satisfaite. Considérons une méthode du type (5.1) et (5.2) avec  $\beta_k = \beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k}$  et le pas  $\alpha_k$  satisfait aux conditions de Wolfe relaxée (3.7)-(3.8) :*

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

avec

$0 < \rho < \sigma_1 < 1$  et  $\sigma_2 = 0$ ; est globalement convergente, dans le sens suivant :

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

**Preuve.** Du théorème précédent on a :

$$\begin{aligned} 1 - \sigma_2 &\leq \frac{-d_k^T g_k}{\|g_k\|^2} \leq 1 + \sigma_1 \\ \Rightarrow 1 &\leq \frac{-d_k^T g_k}{\|g_k\|^2} \leq 1 + \sigma_1 \\ \Rightarrow (1 + \sigma_1)^{-1} &\leq \frac{\|g_k\|^2}{-d_k^T g_k} \leq 1 \\ \Rightarrow (1 + \sigma_1)^{-1} &\leq \frac{\|g_{k+1}\|^2 \|g_k\|^2}{-d_k^T g_k \|g_{k+1}\|^2} \leq 1 \\ \Rightarrow (1 + \sigma_1)^{-1} &\leq \frac{\beta_{k+1}^{CD}}{\beta_{k+1}^{FR}} \leq 1 \\ \Rightarrow \beta_{k+1}^{CD} &\leq \beta_{k+1}^{FR} \end{aligned}$$

Donc  $\beta_{k+1}^{CD}$  vérifie l'inégalité (6.6).

D'après le théorème 4.2 1 on a :

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

■

### Méthode de Dai-Yuan

Cette méthode été découverte par Y. H. Dai et Y. Yuan ([71]) en 1996,  $\beta_k$  est égale à :

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}; \quad y_k = g_{k+1} - g_k \quad (6.15)$$

Cette méthode est fondamentalement différente de la méthode Fletcher Reeves et aussi de la méthode CD. Avec une recherche linéaire de Wolfe standard, la méthode de **DY** génère toujours des directions de descente. En plus elle est globalement convergente avec des hypothèses très générales sur la fonction objectif  $f$ . On exige seulement que  $f$  soit continuellement différentiable et que le gradient soit Lipschitzien, c'est à dire que  $f$  vérifie l'hypothèse **Condition C1**.

### Théorèmes de convergence de Dai et Yuan exigeant seulement la condition de Wolfe et la condition de Lipschitz C1

**Théorème 6.2.5** [31] *Supposons que la condition de Lipschitz C1 soit satisfaite. Considérons des méthodes du type (5.1) et (5.2) où  $\beta_k$  satisfait à  $\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}$ ;  $y_k = g_{k+1} - g_k$  et le pas  $\alpha_k$  satisfait aux conditions de Wolfe faible :*

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

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

$$d_k^T g_k < 0; \quad \forall k \geq 1 \quad (6.16)$$

**Preuve.** La démonstration se fait par récurrence.

1) Pour  $k = 1$  :

---


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

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

2) Supposons que (6.16) 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 (3.7), on aura :

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

D'autre part :

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

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. ■

### Résultat de Dai et Yuan sur la descente suffisante

Dans [16], Dai a analysé la méthode de **DY** de façon plus approfondie. Le résultat suivant donne des conditions suffisantes pour obtenir des directions de descente suffisantes.

**Théorème 6.2.6** *Supposons que la condition de Lipschitz **C1** soit satisfaite. Considérons des méthodes du type (5.1) et (5.2) où  $B_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}$ ;  $y_k = g_{k+1} - g_k$  et  $\alpha_k$  satisfait aux conditions de faible :*

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

avec  $0 < \rho < \sigma < 1$ . Alors la suite  $\{x_k\}$  générée par l'algorithme de Dai-Yuan converge globalement i.e.

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad (6.17)$$

### Preuve

En utilisant le théorème précédent, on aura :

$$\sum_{k \geq 1} \cos^2 \theta \|g_k\|^2 < \infty \quad (6.18)$$

d'autre part on a :

$$\begin{aligned} \|d_{k+1} + g_{k+1}\|^2 &= \|\beta_{k+1}^{DY} d_k\|^2 \\ \Rightarrow \|d_{k+1}\|^2 &= (\beta_{k+1}^{DY})^2 \|d_k\|^2 - 2d_{k+1}^T g_{k+1} - \|g_{k+1}\|^2 \end{aligned} \quad (6.19)$$

De (5.2) :

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

remplaçant ceci dans (6.19), on aura :

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

d'où

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

Supposons maintenant que (6.18) n'est pas satisfaite, autrement dit :

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

On aura :

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

d'où

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

ce qui contredit (6.18). Ceci achève la démonstration.

---

## 6.2.1 Méthode de Dai-Yuan généralisée

Dans le procédé de l'analyse de la méthode de **DY**, Dai et Yuan ont généralisé leurs résultats pour toutes les méthodes du gradient conjugué où  $\beta_k$  peut se mettre sous la forme suivante :

$$\beta_k = \beta_k^{DYG} = \frac{\Phi_{k+1}}{\Phi_k}$$

Le procédé de FR correspond au choix  $\Phi_k = \|g_k\|^2$ . En utilisant (1.3),  $\beta_k^{DY}$  peut être réécrit sous la forme :

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

Par conséquent, le procédé de DY à la forme (DYG) avec  $\Phi_k = g_k^T d_k$ . Le résultat suivant a été établi par Dai et Yuan dans [29, 31] :

**Théorème 6.2.7** [36] *Considérons une méthode du type (3.29) et (3.30) avec  $\beta_k = \beta_k^{DYG} = \frac{\Phi_{k+1}}{\Phi_k}$ ,  $d_k$  est une direction de descente pour tout  $k$ . Supposons aussi que l'hypothèse Condition C1 soit satisfaite. Si la relation de Zoutendijk ( $\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty$ ) est vérifiée et si l'une des trois relations suivantes*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\Phi_k^2} = \infty \quad \text{ou} \quad \sum_{k=0}^{\infty} \frac{\|g_k\|^2}{\Phi_k^2} = \infty \quad \text{ou} \quad \sum_{k=1}^{\infty} \prod_{i=1}^k \beta_i^{-2} = \infty$$

*a lieu, alors la suite générée est globalement convergente, i.e.,*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

### Applications du théorème à la méthode de Dai Yuan

Comme corollaire de ce résultat, la méthode de DY est globalement convergente, si elle est mise en œuvre avec une recherche linéaire inexacte de Wolfe. En effet, on a  $\Phi_k = g_k^T d_k$ . Par conséquent

$$\sum_{k=0}^N \frac{(g_k^T d_k)^2}{\Phi_k^2} = N + 1 \implies \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\Phi_k^2} = \lim_{n \rightarrow \infty} \sum_{k=0}^N \frac{(g_k^T d_k)^2}{\Phi_k^2} = \infty$$

---

## Applications du théorème à la méthode de Fletcher Reeves

la méthode de FR est globalement convergente, si elle est mise en œuvre avec une recherche linéaire inexacte de Wolfe forte avec  $\sigma \leq 1/2$ . En effet, on a  $\Phi_k = \|g_k\|^2$ . Par conséquent

$$\sum_{k=0}^N \frac{\|g_k\|^2}{\Phi_k^2} = N + 1 \implies \sum_{k=0}^{\infty} \frac{\|g_k\|^2}{\Phi_k^2} = \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{\|g_k\|^2}{\Phi_k^2} = \infty$$

Notez qu'une méthode générale de gradient conjugué peut être exprimée sous la forme (4.13) en prenant  $\Phi_0 = 1$ , et

$$\Phi_k = \prod_{j=1}^k \beta_j, \text{ pour } k > 0$$

## 6.3 Les méthodes où $g_{k+1}^T y_k$ figure dans le numérateur de $\beta_k$

### Méthode de Polak-Ribière-Polyak

Cette méthode a été proposée en 1969 par Polak et Ribière ([15]) et Polyak ([8]),  $\beta_k$  est égale à :

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}; \quad y_k = g_{k+1} - g_k$$

Dans [55] la convergence globale de la méthode **PRP** fut établie lorsque  $f$  est fortement convexe et la recherche linéaire est exacte. Pour une fonction non linéaire, Powell ([45]) a montré que la méthode **PRP** est globalement convergente si

- La taille du pas  $s_k = x_{k+1} - x_k$  tend vers zéro.
- La recherche en ligne est exacte.
- La condition de Lipshitz  $C1$  est vérifiée.

D'autre part, Powell a montré plus tard ([43]), en utilisant un contre exemple à 3 dimensions, que, avec une recherche linéaire exacte, la méthode **PRP** pourrait avoir un cycle infini, sans converger vers un point stationnaire. Par conséquent, il est nécessaire d'imposer la condition (a) pour avoir la convergence.

Dans le cas où la direction de recherche est une direction de descente, Yuan ([36]) a établi la convergence globale de la méthode **PRP** pour les fonctions fortement convexes associées à une recherche linéaire inexacte de Wolfe standard. Cependant, pour une recherche linéaire inexacte de *Wolfe forte*, Dai ([67]) a donné un contre exemple qui montre

---

que, même lorsque la fonction objectif est fortement convexe et  $\sigma \in (0, 1)$  est suffisamment petit, la méthode **PRP** peut encore générer des directions de recherche  $d_k$  ascendantes i.e.,  $g_k^T d_k > 0$ .

En résumé, la convergence de la méthode de **PRP** pour une fonction non linéaire est incertaine; l'exemple de Powell montre que lorsque la fonction n'est pas fortement convexe, la méthode **PRP** peut ne pas converger, même avec une recherche linéaire exacte. L'exemple de Dai montre que même pour une fonction fortement convexe, la méthode **PRP** peut ne pas générer une direction de descente avec une recherche linéaire inexacte.

### Modification de la méthode PRP en agissant sur le coefficient $\beta_k$

Sur la base des connaissances tirées de l'exemple de Powell, Dai a suggéré ([70]) la modification suivante dans le paramètre de mise à jour pour la méthode **PRP** :

$$\beta_k^{PRP+} = \max \{ \beta_k^{PRP}, 0 \}.$$

Dans ([30]) Gilbert et Nocedal ont prouvé la convergence de la méthode **PRP** +.

L'analyse de Gilbert et Nocedal s'applique à une classe d'algorithmes du gradient conjugué qui possèdent la propriété suivante :

Considérons une méthode itérative de la forme (5.1) et (5.2), et supposons la suite des gradients  $\{g_k\}$  vérifie la condition suivante :

il existe deux constantes positives  $\gamma$  et  $\bar{\gamma}$  telles que :  $\forall k \in \mathbb{N} : 0 < \gamma \leq \|g_k\| \leq \bar{\gamma}$

### Définition de la propriété (★)

On dit que la propriété (★) est vérifiée s'il existe des constantes  $b > 1$  et  $\lambda > 0$  telle que :

$$\forall k \in \mathbb{N} : |\beta_k| \leq b \quad \text{et} \quad \|s_k\| \leq \lambda \quad \text{implique} \quad |\beta_k| \leq \frac{1}{2b}$$

Le résultat suivant est prouvé dans ([12]).

---

**Théorème 6.3.1** *Considérons une méthode itérative du gradient conjugué de la forme (5.1) et (5.2) qui satisfait aux conditions suivantes :*

a)  $\beta_k \geq 0$

b) *Les directions de recherche sont de descente i.e il existe une constante  $C > 0$  telle que*

$$g_k^T d_k \leq C \|g_k\|^2 \quad : k = 1, 2, \dots$$

c) *La condition de Zoutendijk  $(\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty)$  est vérifiée*

d) *La propriété (★) est vérifiée.*

c) *La condition de Liptchiz C1 et la condition de bornitude C2 sont assurées.*

*Alors la suite générée par cet algorithme est globalement convergente i.e.*

$$\liminf \|g_k\| = 0 \tag{6.20}$$

Comme corollaire de ce résultat, Si on associe à la méthode PRP+, des directions de recherche satisfaisant la condition de descente suffisante DS et des recherche linéaires de Wolfe faible, alors la méthode PRP + est globalement convergente.

Dans ([78]), Dai et Yuan ont donné des exemples pour montrer que la *la condition de bornitude C2* est nécessaire pour obtenir la convergence globale i.e.,  $\liminf_{k \rightarrow \infty} \|g_k\| = 0$ . D'autre part, la contrainte  $\beta_k \geq 0$  ne peut pas être affaiblie par  $\max\{\beta_k^{PRP}, -\varepsilon\}$  pour tout choix de  $\varepsilon > 0$ .

Dans [34], il est montré que la condition de descente suffisante dans le théorème précédent, peut être affaiblie à une condition de descente simple du type  $g_k^T d_k < 0$ , si on utilise une recherche linéaire inexacte de Wolfe forte.

### **Modification de la méthode PRP en agissant sur le pas $\alpha_k$**

La méthode PRP+ a été introduite pour remédier à la défaillance de la convergence de la méthode PRP lorsque elle est mise en œuvre avec une recherche linéaire inexacte de Wolfe. Une autre approche pour corriger l'échec de convergence, est de conserver la formule de mise à jour PRP, mais modifier la recherche linéaire elle même. Suivant cette voie, Grippo et Lucidi( [38]) ont proposé une nouvelle recherche linéaire de type Armijo.

où  $j \geq 0$  est le plus petit entier possédant les propriétés suivantes

$$f(x_{k+1}) \leq f(x_k) - \delta \alpha_k^2 \|d_k\|^2, \quad (6.21)$$

$$-c_1 \|g_{k+1}\|^2 \leq g_{k+1}^T d_{k+1} \leq c_2 \|g_{k+1}\|^2 \quad (6.22)$$

où  $0 < C_2 < 1 < C_1$ ,  $0 < \lambda < 1$  et  $\tau > 0$  sont des constantes. Avec cette nouvelle recherche linéaire, ils prouvent la convergence globale de la méthode **PRP**. Dans un document plus récent [15], ils combinent la recherche linéaire précédente avec une technique de «région de confiance».

Dans une autre voie de recherche, il est montré dans [22] que la méthode **PRP** est globalement convergente lorsque la recherche linéaire utilise un pas de progression constant :  $\alpha_k = \eta < 1/4L$ , où  $L$  est une constante de Lipschitz associée à  $f$ . Dans [63] Sun et Zhang donnent un résultat de convergence globale avec  $\alpha_k = -\delta \frac{g_k^T d_k}{d_k^T Q_k d_k}$ , où  $Q_k$  est une matrice définie positive dont la plus petite valeur propre  $\nu_{\min}$  vérifie  $\nu_{\min} > 0$ ,  $\delta \in (0, \nu_{\min}/L)$ ,  $L$  est la constante de Lipschitz associée à  $\nabla f$ .

Pour ces choix de pas, les directions de recherche ne sont plus conjuguées lorsque  $f$  est quadratique. Par conséquent, ces méthodes doivent être considérées comme des méthodes de plus grande pente, plutôt que des méthodes de gradient conjugué.

## Méthode de Hestenes et Stiefel HS

### Introduction

Cette méthode a été proposée en 1952, dans sa version linéaire par Hestenes et Stiefel ([20]). C'est d'ailleurs le premier article publié qui parle de la méthode du gradient conjugué. Les auteurs utilisent cette méthode itérative pour résoudre des systèmes linéaires,  $\beta_k$  est égale à :

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}; \quad y_k = g_{k+1} - g_k \quad (6.23)$$

La méthode **HS** possède la propriété de conjugaison suivante

$$d_{k+1}^T y_k = 0 \quad (6.24)$$

indépendamment de la recherche linéaire.

---

## Convergence de la méthode HS

Pour une recherche linéaire exacte,  $\beta_k^{HS} = \beta_k^{PRP}$ . Par conséquent, les propriétés de convergence de la méthode de **HS** doivent être similaires aux propriétés de convergence de la méthode PRP. En particulier, si on prend en considération l'exemple de Powell [60], la méthode **HS** associée à une recherche linéaire exacte, peut ne pas converger pour une fonction non linéaire. Il est facile de vérifier que si les directions de recherche satisfont à la condition de descente suffisante et si on utilise une recherche linéaire Wolfe, alors la méthode de HS satisfait la propriété (★).

Comme la méthode PRP +, si nous notons

$$\beta_k^{HS+} = \max \{ \beta_k^{HS}, 0 \}, \quad (6.25)$$

il résulte du théorème 5.1, que la méthode HS+ est globalement convergente.

### 6.3.1 Méthode de Liu et Storey LS

La méthode Liu et Storey LS est également identique à la méthode PRP pour une recherche de linéaire exacte. Bien que peu de recherches ont été faites sur ce choix pour le paramètre de mise à jour, à l'exception de l'article ([52]), nous nous attendons à ce que les techniques développées pour l'analyse de la méthode PRP devraient s'appliquer à la méthode **LS**.

---

## 6.4 Méthodes hybrides et familles paramétriques du gradient conjugué

### Les méthodes hybrides

#### Méthodes hybrides utilisant FR et PRP

Comme nous l'avons vu, la première série de méthodes **FR**, **DY** et **CD** ont des propriétés de convergence remarquables, mais ils ne sont pas assez performants en pratique à cause du phénomène du brouillage. D'autre part, comme on l'a remarqué, le second ensemble de méthodes **PRP**, **HS**, **LS** peuvent ne pas converger. Cependant numériquement et partiquement, elles sont plus performantes que les méthodes **FR**, **DY** et **CD**. Par conséquent, des combinaisons des deux types de méthodes ont été proposées pour tenter d'exploiter les caractéristiques intéressantes de chaque ensemble. Touati-Ahmed et Storey [14] ont proposé la méthode hybride qui suit :

$$\beta_k = \begin{cases} \beta_k^{PRP} & 0 \leq \beta_k^{PRP} \leq \beta_k^{FR} \\ \beta_k^{FR} & \text{sinon} \end{cases}$$

Ainsi, lorsque les itérations coïncent ou bloquent, le paramètre de mise à jour PRP est utilisé. Par les mêmes motivations, Hu et Storey [79] ont proposé

$$\beta_k = \max \{0, \min \{ \beta_k^{PRP}, \beta_k^{FR} \} \}$$

Dans [79], il est précisé que  $\beta_k^{PRP}$  peut être négatif, même pour des fonctions fortement convexes. Nocedal et Gilbert [30] ont proposé

$$\beta_k = \max \{ -\beta_k^{FR}, \min \{ \beta_k^{PRP}, \beta_k^{FR} \} \}$$

Avec ce procédé hybride,  $\beta_k$  peut être négatif car  $\beta_k^{FR}$  est toujours positif ou nul. On notera que dans les résultats numériques présentés dans [38], les performances de ce procédé hybride n'étaient pas meilleure que celles de PRP+.

Les résultats de convergence pour les méthodes du gradient conjugué qui peuvent être bornés en fonction de la méthode FR sont développés dans [39, 47, 49]. En particulier, le résultat suivant est établi dans [47] :

**Théorème 6.4.1** *Considérons une méthode itérative du gradient conjugué de coefficient  $\beta_k$  de la forme (3.29) - (3.30), qui utilise une recherche linéaire inexacte de Wolfe forte (2.8)-(2.9) avec  $\sigma \leq 1/2$ . Supposons aussi que l'hypothèse de Lipschitz  $C1$  soit vérifiée et qu'on a  $2\sigma |\beta_k| \leq \beta_k^{FR}$ . Alors les directions de recherche engendrées sont toujours les directions de descente et la suite associée à cet algorithme est globalement convergente i.e.*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

**Remarque 6.4.1** *Dans le théorème précédent, avec la condition  $2\sigma |\beta_k| \leq \beta_k^{FR}$ , on a obtenu la descente et la convergence globale. Dans [39, 49], on a imposé une condition plus forte, en l'occurrence  $2\sigma |\beta_k| < \beta_k^{FR}$  et on a obtenu un résultat plus fort, la convergence globale et la descente suffisante.*

### Méthodes hybrides utilisant DY et HS

Rappelons que la méthode de DY possède des résultats de convergence globale plus performantes que celles de FR. Par conséquent, Dai et Yuan ([78]) ont étudié la possibilité de combiner DY avec d'autres méthodes CG. Utilisant une recherche linéaire de Wolfe et pour  $\beta_k \in [-\eta\beta_k^{DY}, \beta_k^{DY}]$ , où  $\eta = (1 - \sigma)(1 + \sigma)$ , ils démontrent la convergence globale lorsque la condition de Lipschitz  $C1$  est assurée.

Les deux méthodes hybrides suivantes ont été proposés dans ([30]) :

$$\beta_k = \max \left\{ - \left( \frac{1 - \sigma}{1 + \sigma} \right) \beta_k^{DY}, \min \{ \beta_k^{HS}, \beta_k^{DY} \} \right\}$$

et

$$\beta_k = \max \{ 0, \min \{ \beta_k^{HS}, \beta_k^{DY} \} \}$$

Les expériences numériques dans ([20]) indiquent que la seconde méthode hybride a donné des résultats plus performants que la méthode PRP<sup>+</sup>.

Une autre méthode hybride a été proposée par Dai dans ([17]). Elle emploie soit le modèle de DY ou celui de CD :

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

Il montre que ce système hybride génère des directions de descente, indépendamment de la recherche linéaire. Cette propriété de descente est plus forte que celle du modèle de DY lui-même, où la descente est valable pour une recherche linéaire de Wolfe. Dai montre

que pour ce système hybride,  $\beta_k \in [0, \beta_k^{DY}]$ . Cette propriété, ainsi que celle de la descente du modèle hybride, implique la convergence des méthodes pour des recherche linéaires typiques.

### Les méthodes unifiées

Les méthodes quasi-Newtonniennes ont été regroupées en familles dites de Broyden ([9]) et ont été étudiées ensemble. De la même manière, Dai et Yuan ([26, 31]), ont unifié plusieurs méthodes du gradient conjugué et ont proposé une famille à un paramètre des méthodes du gradient conjugué en posant

$$\beta_k^{DY-unif} = \frac{\|g_{k+1}\|^2}{\lambda_k \|g_k\|^2 + (1 - \lambda_k) d_k^T y_k} \quad (6.26)$$

où  $\lambda_k \in [0, 1]$  est un paramètre. Les méthodes unifiées sont en fait une combinaison convexe des méthodes FR et DY. Le procédé de FR correspond à  $\lambda_k = 1$ , tandis que la méthode de DY correspond à  $\lambda_k = 0$ . Dans ([27]), cette famille est étudiée en considérant  $\lambda_k \in ]-\infty, +\infty[$ . Si la condition de Lipschitz  $C1$  est vérifiée et si on utilise une recherche linéaire de Wolfe avec

$$\sigma_1 - 1 \leq (\sigma_1 + \sigma_2) \lambda_k \leq 1$$

alors on a la convergence globale pour chaque membre de la famille.

En considérant des combinaisons convexes des numérateurs et des dénominateurs de  $\beta_k^{FR}$  et  $\beta_k^{HS}$ , Nazareth ([35]) propose et de façon indépendante, une famille à deux paramètres des méthodes du gradient conjugué avec :

$$\beta_k^{N-unif} = \frac{\mu_k \|g_{k+1}\|^2 + (1 - \mu_k) g_{k+1}^T y_k}{\lambda_k \|g_k\|^2 + (1 - \lambda_k) d_k^T y_k} \quad (6.27)$$

où  $\lambda_k, \mu_k \in [0, 1]$ . Si on attribue à  $\lambda_k$  et  $\mu_k$  les valeurs extrêmes 0 ou 1, on obtient les méthodes du gradient conjugué FR, DY, PRP, et SH. Par exemple

$$\begin{aligned} \lambda_k = 1, \mu_k = 0 &\implies \beta_k^{N-unif} = \frac{g_{k+1}^T y_k}{\|g_k\|^2} = \beta_k^{PRP} \\ \lambda_k = 0, \mu_k = 1 &\implies \beta_k^{N-unif} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} = \beta_k^{DY} \\ \lambda_k = 1, \mu_k = 1 &\implies \beta_k^{N-unif} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} = \beta_k^{FR} \end{aligned}$$

Constatant que les six méthodes du gradient conjugué qu'on a vu précédemment ont deux numérateurs et trois dénominateurs, Dai et Yuan ([29]) ont introduit une nouvelle famille de gradients conjugués encore plus large à trois paramètres, ils ont choisi  $\beta_k$  comme

suit :

$$\beta_k = \beta_k^{DY3} = \frac{\mu_k \|g_{k+1}\|^2 + (1 - \mu_k) g_{k+1}^T y_k}{(1 - \lambda_k - \omega_k) \|g_k\|^2 + \lambda_k d_k^T y_k - \omega_k d_k^T g_k} \quad (6.28)$$

où  $\mu_k, \lambda_k, \omega_k \in [0, 1]$ .

Cette famille à trois paramètres comprend les six méthodes standard du gradient conjugué, les familles à 1 ou 2 paramètres vues ci dessus et de nombreuses méthodes hybrides comme cas particuliers. Afin de veiller à ce que les directions de recherche générés par cette famille soient des directions de descente, le critère de redémarrage de Powell ([58]) est utilisé. Posons  $d_k = -g_k$ , si

$$|g_k^T g_{k-1}| > \xi \|g_k\|^2,$$

où  $\xi > 0$  est une certaine constante fixe. Si on utilise une recherche linéaire inexacte de Wolfe forte (2.8)-(2.9) avec

$$(1 + \xi) \sigma \leq \frac{1}{2}$$

Dai et Yuan ([29]) ont montré que les directions de recherche sont des directions de descente. Les résultats de convergence globale sont également établis.

Dans ([19]) Dai et Liao modifient le numérateur du paramètre de mise à jour pour obtenir HS

$$\beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k} = \beta_k^{HS} - t \frac{g_{k+1}^T s_k}{d_k^T y_k}, \quad s_k = x_{k+1} - x_k = \alpha_k d_k. \quad y_k = g_{k+1} - g_k \quad (6.29)$$

où  $t > 0$  est une constante. Pour une recherche de ligne exacte,  $g_{k+1}$  est orthogonal à  $s_k = x_{k+1} - x_k = \alpha_k d_k$ . Ainsi, pour une recherche linéaire exacte,  $\beta_k^{DL} = \beta_k^{HS}$ . Par conséquent la méthode du gradient conjugué DL se réduit à la méthode HS et la méthode PRP. Encore une fois, en raison de l'exemple de Powell, la méthode DL peut ne pas converger pour une recherche linéaire exacte. Semblable à la méthode PRP + , Dai et Liao ont également modifié leur formule de la façon suivante pour assurer la convergence :

$$\beta_k^{DL+} = \max \left\{ \frac{g_{k+1}^T y_k}{d_k^T y_k}, 0 \right\} - t \frac{g_{k+1}^T s_k}{d_k^T y_k} \quad (6.30)$$

Si les hypothèses de Lipschitz  $C1$  et de bornétude  $C2$  sont vérifiées et si  $d_k$  satisfait à la condition de descente suffisante (2.3), il est montré dans [19] que la méthode DL+, mise en œuvre une recherche linéaire inexacte de Wolfe forte (2.8)-(2.9), est globalement convergente.

Très récemment, dans un nouveau développement de cette stratégie de mise à jour, Yabe et Takano proposent le choix suivant pour le paramètre de mise à jour, basée sur

---

une condition sécante modifiée donnée par Zhang et al. ([62]) :

$$\beta_k^{YT} = \frac{g_{k+1}^T (z_k - ts_k)}{d_k^T z_k} \quad (6.31)$$

où

$$z_k = y_k + \left( \frac{\rho \theta_k}{s_k^T u_k} \right) u_k$$

$$\theta_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k$$

$\rho \geq 0$  est une constante et  $u_k \in \mathbb{R}^n$  satisfait  $s_k^T u_k \neq 0$  ; par exemple,  $u_k = d_k$  . Encore une fois, et suivant la démarche appliquée à la méthode PRP+, Yabe et Takano ont modifié leur formule de la façon suivante pour assurer la convergence :

$$\beta_k^{YT+} = \max \left\{ \frac{g_{k+1}^T z_k}{d_k^T z_k}, 0 \right\} - t \frac{g_{k+1}^T s_k}{d_k^T z_k} \quad (6.32)$$

Ils montrent que la méthode YT+ est globalement convergente si les hypothèses de Lipschitz  $C1$  et de bornétude  $C2$  sont vérifiées et  $d_k$  satisfait à la condition de descente suffisante, et si une recherche linéaire inexacte de Wolfe forte (2.8)-(2.9) est employée avec

$$0 \leq \rho < \frac{1 - \sigma}{3(1 + \sigma - 2\delta)}$$

Sellami, Laskri et Benzine ([7]) ont proposé une nouvelle famille à deux paramètres des méthodes du gradient conjugué pour l'optimisation sans contraintes et ont étudié ses propriétés et sa convergence.  $\beta_k^*$  est de la forme générale suivante :

$$\beta_k^* = \frac{\phi_k}{\phi'_{k-1}}, \quad (6.33)$$

où  $\phi'_{k-1}$  satisfait

$$\phi'_{k-1} = (1 - \lambda_k - \mu_k) \|g_{k-1}\|^2 + (\lambda_k + \mu_k)(-g_{k-1}^T d_{k-1}). \quad (6.18)$$

et

$$\phi_k = \lambda \|g_k\|^2 + (1 - \lambda)(-g_k^T d_k), \quad \lambda \in [0, 1]. \quad (6.34)$$

Avec ces données, nous obtenons

$$\beta_k^* = \frac{\|g_k\|^2}{(1 - \lambda_k - \mu_k) \|g_{k-1}\|^2 + \mu_k(-d_{k-1}^T g_{k-1}) + \lambda_k(y_{k-1}^T d_{k-1})}$$

## 7.1 Intrduction

Considérons la méthode d'optimisation sans contraintes suivant :

$$\min_{x \in \mathbb{R}^n} f(x) \quad (7.1)$$

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est une fonction continuellement différentiable. Les différentes formes du gradient conjugué sont parmi les meilleures méthodes utilisées pratiquement pour résoudre les problèmes de type (1), spécialement quand la dimension de l'espace  $n$  est très grande. La méthode du gradient conjugué génère une suite  $\{x_k\}_{k \in \mathbb{N}}$  de la manière suivante :

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \dots, \quad (7.2)$$

Le pas  $\alpha_k \in \mathbb{R}$  étant déterminé par une recherche linéaire exacte vérifier

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k) \quad (7.3)$$

Les directions  $d_k$  sont calculées de façon récurrente par la formule suivant

$$d_k = \begin{cases} -g_k & \text{si } k = 0 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 1 \end{cases} \quad (7.4)$$

où  $g_k$  est le gradient de  $f(x)$  au point  $x_k \in \mathbb{R}^n$  et  $\beta_k$  est un scalaire. On connaît quelques formules classiques de calcul des  $\beta_k$ . Citons les plus célèbres : Celles de Hestenes-Stiefel (HS) en 1952, de Fletcher-Reeves en 1964, de Polak-Ribiere-Polyak (PRP) en

1969. Citons aussi la méthode conjugate descent method (CGD) en 1987, la méthode Liu-Storey (LS) en 1992, et la méthode the Dai-Yuan (DY) en 2000. Les paramètres  $\beta_k$  sont calculés en utilisant les formules suivantes : en 1969. Citons aussi la méthode conjugate descent method (CGD) en 1987, la méthode Liu-Storey (LS) en 1992, et la méthode the Dai-Yuan (DY) en 2000. Les paramètres  $\beta_k$  sont calculés en utilisant les formules suivantes : en 1969. Citons aussi la méthode conjugate descent method (CGD) en 1987, la méthode Liu-Storey (LS) en 1992, et la méthode the Dai-Yuan (DY) en 2000. Les paramètres  $\beta_k$  sont calculés en utilisant les formules suivantes : en 1969. Citons aussi la méthode conjugate descent method (CGD) en 1987, la méthode Liu-Storey (LS) en 1992, et la méthode the Dai-Yuan (DY) en 2000. Les paramètres  $\beta_k$  sont calculés en utilisant les formules suivantes :

$$\begin{aligned}\beta_k^{HS} &= \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, & \beta_k^{FR} &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, & \beta_k^{PRP} &= \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \\ \beta_k^{CD} &= -\frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}}, & \beta_k^{LS} &= -\frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}}, & \beta_k^{DY} &= \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}},\end{aligned}$$

La convergence des différentes méthodes du gradient conjugué a été étudié par plusieurs mathématiciens. Citons les plus importants résultats. Powell ([44]) et Zoutendijk (1970, [20]) ont prouvé la convergence de la méthode du gradient conjugué version Fletcher Reeves en utilisant la recherche linéaire exacte. Al-Baali ((1985) [1]) généralisa ce résultat pour les recherches linéaires inexactes de Wolfe faible forte. Touati-Ahmed et Storey (1990, [14]), Gilbert et Nocedal (1992, [30]) ont étudié la convergence globale du gradient conjugué version PRP avec la recherche linéaire inexacte de Wolfe faible.

Belloufi et Benzine ([49]) ont proposé une nouvelle direction  $d_k^{BB}$  définie comme suit :

$$d_k^{BB} = \begin{cases} -\frac{g_k}{\|g_k\|^2} & \text{if } k = 0 \\ -\frac{1}{\|g_k\|^2} g_k + d_{k-1} & \text{if } k \geq 1 \end{cases} \quad (7.5)$$

Belloufi et Benzine ([49]) ont obtenu leur résultat de convergence en utilisant la recherche linéaire inexacte de Wolfe forte suivante :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \quad (7.6)$$

$$\left| g(x_k + \alpha_k d_k)^T d_k \right| \leq -\sigma g_k^T d_k \quad (7.7)$$

où  $\delta \in \left] 0, \frac{1}{2} \right[$  et  $\sigma \in ]\delta, 1[$ .

On propose de ce paragraphe un nouveau algorithme qu'on a appelé : Conjugate gradient Boumediene Benzine Belloufi et noté CGBBB Algorithm. On démontre dans cette partie la descente suffisente et la convergence globale de notre nouveau algorithme.. Pour arriver à ce but, on utilise une recherche linéaire exacte du type (7.3).

On termine ce paragraphe en effectuant près de 700 tests numériques sur notre nouveau algorithme et on les compare avec d'autres algorithmes assez connus, en utilisant la méthode de Dolan et Moré CPU performance. On montre que l'algorithme CGBBB est plus performant que l'algorithme Polak-Ribiere-Polyak (CGPRP) et l'algorithme de la descente conjuguée (CGDC).

## 7.2 Algorithme de méthode CGBBB

Algorithme

**Etape1 :** Donné  $x_0 \in \mathbb{R}^n$ , posé  $d_0 = \frac{-g_0}{\|g_0\|^2}$ ,  $k = 0$ .

**Etape2 :** Si  $\|g_0\| = 0$  stope sinon passer à l'Etape2.

**Etape3 :** Poser  $x_{k+1} = x_k + \alpha_k d_k$  où  $d_k$  est définie par (7.5) et  $\alpha_k$  est définie par (7.3)

**Etape4 :** Poser  $k = k + 1$  et aller à l'Etape1.

## 7.3 Propriété de la descente

Nous sommes en mesure maintenant de démontrer que la méthode CGBBB possède la propriété suivante dite de descente suffisante qui nous sera utile pour prouver la convergence globale de notre algorithme :

**Théorème 7.3.1** *Supposons que les suites  $\{x_k\}$  et  $\{d_k\}$  sont générées par l'algorithme CGBBB en se basant sur (7.2), et (7.5) où le pas  $\alpha_k$  est calculée à partir de la recherche linéaire exacte (7.3). Alors on a :*

$$g_k^T d_k = -1, \quad k = 0, 1, 2, \dots \quad (7.8)$$

**Preuve.** On démontre ce théorème par récurrence. Si  $k = 0$ , alors on a

$$g_0^T d_0 = g_0^T \left( \frac{-g_0}{\|g_0\|^2} \right) = -\frac{\|g_0\|^2}{\|g_0\|^2} = -1 < 0.$$

Donc (7.6) est vraie.

Supposons que (7.6) soit vraie pour  $k \geq 1$ . En utilisant (7.5) et en multipliant par  $g_{k+1}$  on obtient

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

Remarquons maintenant que pour les recherches linéaires exactes nous avons toujours  $g_{k+1}^T d_k = 0$ .

Par conséquent :

$$g_{k+1}^T d_{k+1} = -1, \quad k = 0, 1, 2, \dots$$

■

## 7.4 Convergence globale de l'Algorithme CGBBB

Pour énoncer et démontrer le théorème de convergence globale nous avons besoin de l'hypothèse suivante :

### Hypothèse 1

(i) Soit  $x_0 \in \mathbb{R}^n$ , un point de démarrage de l'Algorithme quelconque et  $\Omega = \{x \in \mathbb{R}^n / f(x) \leq f(x_0)\}$ . suppose que l'ensemble  $\Omega$  est borné.

(ii) Dans un voisinage  $N$  de  $\Omega$ , on suppose que la fonction objectif  $f$  est continuellement différentiable et que son gradient  $g(x) = \nabla f(x)$  est continuellement Lipschitzien, c'est à dire qu'il existe une constante  $\ell > 0$  telle que

$$\|g(x) - g(y)\| \leq \ell \|x - y\|, \quad (x, y) \in N^2 \quad (7.9)$$

Sous l'Hypothèse 1, on obtient le lemme suivant qui a été prouvé par Zoutendijk ([7]).

### **Lemme 7.4.1** (Zoutendijk[23])

Supposons que les suites  $\{x_k\}$  et  $\{d_k\}$  sont générées par l'algorithme CGBBB en se basant sur (7.2), (7.3), and (7.5), où le pas  $\alpha_k$  est calculé à partir de la recherche linéaire exacte (7.3). Supposons que Hypothèse 1 soit satisfaite. Alors on obtient la relation suivante dite condition de Zoutendijk :

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \quad (7.10)$$

Avec ces résultats on obtient le résultat de convergence globale suivant associé à l'Algorithme CGBBB :

**Théorème 7.4.1** *Supposons que  $L$ , hypothèse 1 soit vérifiée, Soit  $\{x_k\}_{k \in \mathbb{N}}$  la suite générée par l'Algorithme CGBBB, où  $\alpha_k$  est obtenue par la recherche linéaire exacte (7.3), Alors on a*

$$\lim_{k \rightarrow \infty} \|g_k\| = 0 \quad \text{ou} \quad \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty$$

**Preuve.** En effet. Supposons le contraire. Alors il existe une constante  $C > 0$  telle que

$$\|g_k\| \geq C. \tag{7.11}$$

Nous aurons alors

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

et

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

Pour les recherches linéaires exactes on a  $g_k^T d_{k-1} = 0$ . Donc

$$\begin{aligned} \|d_k\|^2 &= \frac{1}{\|g_k\|^2} + \left( \frac{1}{\|g_{k-1}\|^2} + \|d_{k-1}\|^2 \right), \\ &= \frac{1}{\|g_k\|^2} + \frac{1}{\|g_{k-1}\|^2} + \dots + \frac{1}{\|g_1\|^2} + \|d_0\|^2, \end{aligned}$$

Remarquons que  $\|d_0\|^2 = \frac{1}{\|g_0\|^2}$ . Alors on obtient :

$$\|d_k\|^2 = \frac{1}{\|g_k\|^2} + \frac{1}{\|g_{k-1}\|^2} + \dots + \frac{1}{\|g_1\|^2} + \frac{1}{\|g_0\|^2},$$

,

ou encore

$$\|d_k\|^2 = \sum_{i=0}^k \frac{1}{\|g_i\|^2} \tag{7.12}$$

(7.11) implique

$$\frac{1}{\|g_i\|^2} \leq \frac{1}{C^2}.$$

Par conséquent on a :

$$\|d_k\|^2 = \sum_{i=0}^k \frac{1}{\|g_i\|^2} \leq \sum_{i=0}^k \frac{k}{C^2}$$

D'après la relation(7.8) on a

$$(g_k^T d_k)^2 = 1,$$

Ceci implique

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \frac{k}{C^2} \quad (7.13)$$

et par conséquent

$$\frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \frac{C^2}{k} \quad (7.14)$$

Il est bien connu que

$$\sum_{k=0}^{\infty} \frac{1}{k} = +\infty \quad (7.15)$$

(7.14) et (7.15) impliquent :

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} = +\infty$$

Ceci contredit la condition de Zoutendijk du lemm2 1. Le théorème2 est donc prouvé.

■

## 7.5 Résultats numériques et comparaisons

Dans cette section on expose les résultats des tests numériques obtenus par implémentation de l'Algorithme CGBBB. Ces tests ont été obtenus en utilisant les programmes Fortran des fonctions tests de la référence ([10]). Considérant les mêmes critères que dans ([12]), nous avons sélectionné 52 fonctions tests en optimisation sans contraintes ([10], Cute Library)). Pour chaque fonction test nous effectués 20 tests numériques en prenant en considération un nombre croissant de variables :  $n = 2, 10, 30, 50, 70, 100, 300, 500, 700, 900, 1000, 2000, 6000, 7000, 8000, 9000, 10000$ . On utilise dans notre algorithme la recherche linéaire exacte (7.3) et le critère d'arrêt :  $\|\nabla f(x_k)\| < 10^{-6}$ . Le nombre d'itérations se limite à 100000.

On utilise la méthode de Dolan et Moré CPU performance.

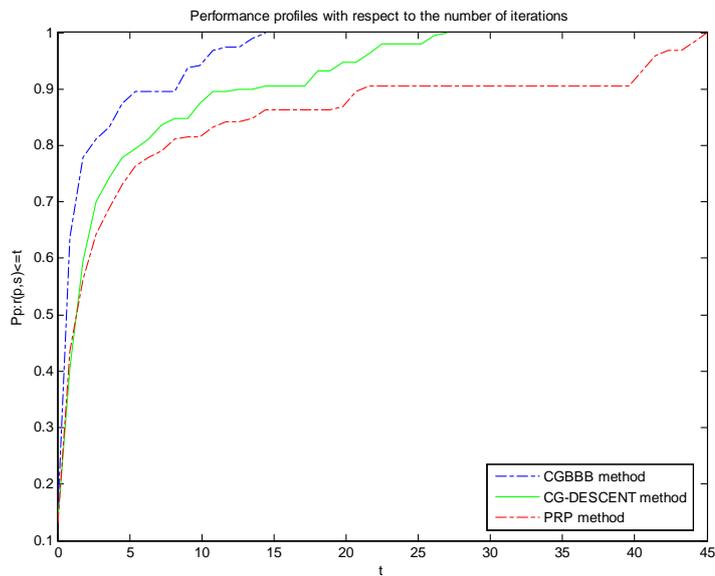
Figure 1 montre que Dolan et Moré CPU performance profile de l'Algorithme CGBBB, est plus performant que l'algorithme Polak-Ribiere-Polyak (CGPRP) et l'algorithme de

---

la descente conjuguée (CGDC).

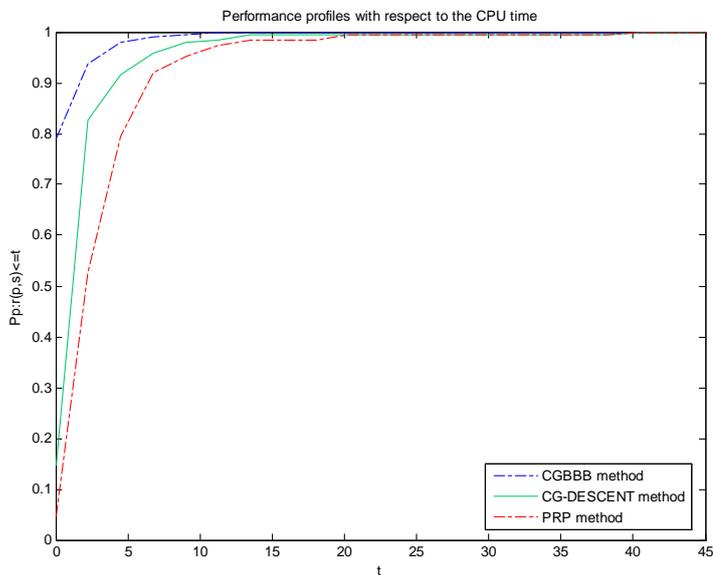
Quand on compare 2 algorithmes,, le plus performant est celui dont la courbe est au dessus de la courbe du second algorithme.

*iteration*



4.pdf

*time*



5.pdf

## 8.1 Programme en fortran 77 d' Armijo

```

c*****
      Programme d' Armijo
c*****
      real x(2),xp(2),g(2),gp(2),f0,alph,f,p,s,a,b,pf,t,y,phopt
      integer i,k,test,n
      sig=0.3;ro=0.1;b=10000000;alph=10;test=0;k=0
      print*, 'le dimention n=';read*,n
      do i=1,n print*, 'x0(',i,')=';read*, xp(i) end do
10  if (test==0.and.k.le.10) then call fonc(f0,gp,xp)
      do i=1,n x(i)=xp(i)-alph*gp(i) end do
      call fonc(f,g,x);p=0
      do i=1,n p=p+gp(i)*gp(i) end do
      s=f0-ro*alph*p
      if (f.gt.s) then b=alph else phopt=alph;test=1 endif
      print*, 'alpha(',k,')=', alph;alph=b/2;k=k+1
      go to 10
      end if
      print*, 'alpha optimal=', phopt
      end
c*****
      subroutine fonc(f,g,x)

```

---

```

real f,g(2),x(2)
f=(x(1))**2+(x(2))**4;g(1)=2*x(1);g(2)=4*(x(2))**3
end
c*****
c*****

```

## 8.2 Programme en fortran 77 de Goldschtien

```

c*****
Programme de Goldschtien
c*****
real x(2),xp(2),g(2),gp(2),f0,alph,f,p,s,a,b,pf,t,y,phopt
integer i,k,test,n
ro=0.1;delt=0.3;b=10000000;alph=10;test=0;k=0
print*, 'le dimention n=';read*,n
do i=1,n print*, 'x0(',i,')=';read*, xp(i) end do
10 if (test==0.and.k.le.1000) then call fonc(f0,gp,xp)
do i=1,n x(i)=xp(i)-alph*gp(i) end do
call fonc(f,g,x);p=0
do i=1,n p=p+gp(i)*gp(i) end do
s=f0-ro*alph*p;y=f0-delt*alph*p
if (f.gt.s) then b=alph else if (f.lt.y) then a=alph else
phopt=alph;test=1 endif
endif
print*, 'alpha(',k,')=', alph;alph=(a+b)/2;k=k+1
go to 10
end if
print*, 'alpha optimal=', phopt
end
c*****
subroutine fonc(f,g,x)
real f,g(2),x(2)
f=(x(1))**2+(x(2))**4;g(1)=2*x(1);g(2)=4*(x(2))**3
end
c*****
c*****

```

---

## 8.3 Programme en fortran 77 de Wolfe

```
c*****
      Programme de Wolfe
c*****
      real x(2),xp(2),g(2),gp(2),f0,alph,f,p,s,a,b,pf,t,y,phopt
      integer i,k,test,n
      sig=0.3;ro=0.1;b=10000000;alph=10;test=0;k=0;print*, 'le dimention n='
      read*,n
      do i=1,n print*, 'x0(',i,')='; read*, xp(i) end do
10 if (test==0.and.k.le.10) then call fonc(f0,gp,xp)
      do i=1,n x(i)=xp(i)-alph*gp(i) end do
      call fonc(f,g,x);p=0
      do i=1,n p=p+gp(i)*gp(i) end do
      s=f0-ro*alph*p
      if (f.gt.s) then b=alph else pf=0
      do i=1,n pf=pf+g(i)*gp(i) end do
      t=-pf;y=sig*(-p)
      if (t.lt.y) then a=alph else phopt=alph;test=1 endif
      endif
      print*, 'alpha(',k,')=', alph;alph=(a+b)/2;k=k+1
      go to 10
      end if
      print*, 'alpha optimal=', phopt;x=xp
      end
c*****
      subroutine fonc(f,g,x)
      real f,g(2),x(2)
      f=(x(1))**2+(x(2))**4;g(1)=2*x(1);g(2)=4*(x(2))**3
      end
c*****
c*****
```

## 8.4 Programme en fortran 90 de la méthode du gradient conjugué

---

```

c*****
Programme du gradient conjugué
c*****
program CGWP
implicit none
integer, parameter : :n=10000
! test : nombre des direction engendré par la méthode du GC
! et qui ne sont pas des directions de descente
! cont : nombre des évaluations fonctionnelles
integer ik,j,k,contf,test,contg,B
! Y : la suite, g : le gradient, d : la direction de déplacement
! eps : epcilon (critère de convergence)
double precision y(n),g(n),gp(n)
double precision d(n),eps
double precision alphd,alph0
double precision b,w,alphg,f0,f1
integer (2) hour,minut,second,hstedt
integer (2) hour2,minut2,second2,hstedt2
call gettim(hour,minut,second,hstedt)
print*,hour,minut,second,hstedt
print*,'ENTRER Y'
read*, B
do ik=1,n
y(ik)=0.5
enddo
print*,'ENTRER EPS'
eps=1.e-10
b=0.9 ;w=0.1 ;contf=1 ;test=0 ;contg=1
g=df(y) ;f0=f(y) ;d=-g ;j=1 ;k=1
do while(sqrt(dot_product(g,g))>eps)
c*****
! recherche linéaire
c*****
alphg=0 ;alphd=100 ;alph0=1
gp=g
do
contf=contf+1

```

---

```

f1=f(y+alph0*d)
if(f1<=f0+w*alph0*dot_product(gp,d))then
contg=contg+1
g=df(y+alph0*d)
if(dot_product(g,d)>=b*dot_product(gp,d))then
exit
else
alphg=alph0 ;alph0=(alphg+alphd)/2.
endif
else
alphd=alph0 ;alph0=(alphg+alphd)/2.
endif
enddo
c*****
! le nouveau point
c*****
y=y+alph0*d
k=k+1
if(j<n)then
if(B=1)then
d=-g+(dot_product(g,g-gp)/dot_product(d,g-gp))*d
end if
if(B=2)then
d=-g+(dot_product(g,g)/dot_product(gp,gp))*d
end if
if (B=1)then
d=-g+(dot_product(g,g)/dot_product(gp,gp))*d
end if
j=j+1
if(dot_product(g,d)>=0)then
j=1
d=-g
test=test+1
endif
else
d=-g
j=1

```

---

```

endif
c*****
!test de la décroissance négligeable
c*****
if(f0-f1<1.e-14)then
print*
print*, 'THE NORM OF GRADIENT'
print '(e20.2)',sqrt(dot_product(g,g))
g=0
endif
f0=f1
enddo
c*****
! affichage du temps
c*****
call gettim(hour2,minut2,second2,hse dt2)
print*,hour2,minut2,second2,hse dt2
print*,hour2-hour
print*,minut2-minut
print*,second2-second
print*,hse dt2-hse dt
c*****
!affichage des résultats
c*****
print*
print*, 'THE NUMBER OF ITERATION'
print*,k
print*, 'THE FUNCTION VALUE'
print '(e20.2)',f(y)
print*, 'NOMBRE D EVALUATIONS FONCTIONNELLES',contf
print*, 'NOMBRE D EVALUATIONS GRADIENT',contg
print*, 'test=',test
! les procédures intérieures
contains
end
*****
*****

```

- [1] A. Al-Baali, (1985). Descent property and global convergence of Fletcher-Reeves method with inexact line search. *IMA.J.Numer. Anal.*, 5;121-124.
- [2] A.A. Goldstein and J.F.,(1969). Price, An effective algorithm for minimization, *Num.Math.*, 10 , pp. 184-189.
- [3] A. A. Goldstein, (1965). On steepest descent, *SIAM J. Control* 3;147-151..
- [4] A. Buckley, Conjugate gradient methods, in *Nonlinear Optimization* , M. J. D. Powell, ed., Academic Press, London, pp. 17–22.
- [5] A. V. Fiacco and G. P., (1968). McCormick, *Nonlinear Programming*, John Wiley, New York.
- [6] A. A. Goldstein, (1965), On steepest descent, *SIAM J. on Control* A, Vol. 3, No. 1 pp. 147-151.
- [7] B. Sellami, Y. Laskri et R. Benzine (2013). A new two-parameter family of nonlinear conjugate gradient methods. *Optimization*, <http://dx.doi.org/10.1080/02331934.2013.830118>.
- [8] B. T. Polyak,(1969), The conjugate gradient method in extreme problems, *USSR Comp. Math. Math. Phys.*, 9 pp. 94–112.
- [9] 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.
- [10] C. G. Broyden, (1970).The convergence of a class of double-rank minimization algorithms 1. general considerations, *J. Inst. Math. Appl.*, 6, pp. 76–90.
- [11] C. W. Carroll, (1961). The created response surface technique for optimizing nonlinear restrained systems, *Operations Res.*, Vol. 9 , pp. 169-84.

- 
- [12]. C. Gilbert , *Eléments d'optimisation différentiable : théorie et algorithmes*, Notes de cours, École Nationale Supérieure de Techniques Avancées, Paris.
- [13] D. F. Shanno,(1978). On the convergence of a new conjugate gradient algorithm, *SIAM J. Numer. Anal.*, 15 ,pp. 1247–1257.
- [14] D. Touati-Ahmed, C. Storey, (1990). Efficient hybrid conjugate gradient techniques, *J. Optim. Theory Appl.* 64 ;379-397.
- [15] E. Polak and G. Ribière,(1969). Note sur la convergence de directions conjuguées, *Rev. Francaise Informat Recherche Opertionelle*, 3e Année 16, pp. 35–43.
- [16] E. M. L. Beale, ( 1972) . A derivative of conjugate gradients, in *Numerical Methods for Nonlinear Optimization*, F. A. Lootsma, ed., Academic Press, London, pp. 39–43.
- [17] Fletcher, R. and Reeves, (1964). Function minimization by conjugate gradients. *Com-  
pute. J.*7 ; 149-154.7.
- [18] G. Fasano, (2005). Planar conjugate gradient algorithm for large-scale unconstrained optimization. Part 1 : Applications, *Journal of Optimization Theory and Applications* 125, pp. 543–558.
- [19] G. Yuan, X. Lu,Z. Wei (2009), A conjugate gradient method with descent direction for unconstrained optimization, *J. Comput. Appl. Math.* 233 ;519-530
- [20] G. Zoutendijk, (1970). Nonlinear programming computational methods, in ;Aba9.M.R.Hestenes, E.Stiefel, *Method of conjugate gradient for solving linear equations*, *J.Res.Nat.Bur.Stand.*49 ;405-436.
- [21] G. H. Liu, J. Y. Han and H. X. Yin, (1960). Global convergence of the Fletcher-Reeves algorithm with an inexact line search, *Appl. Math. J. Chinese Univ. Ser. B*, 10 , pp. 75–82
- [22] G. Zoutendijk, (1960). *Methods of Feasible Directions*, Elsevier, Amesterdam.
- [23] G. Zoutendijk, (1970). *Nonlinear Programming, Computational Methods*, in *Integer and Nonlinear Programming*, J. Abadie, ed., North-Holland, Amsterdam, pp. 37–86.
- [24] H. Yabe and M. Takano, (2004). Global convergence properties of nonlinear conjugate gradient methods with modi ?ed secant condition, *Comput. Optim. Appl.*, 28, pp. 203–225.
- [25 ]H. Fan, Z. Zhu et A. Zhou, (1995). A New Descent Nonlinear Conjugate Gradient Method for Unconstrained Optimization, *Applied Mathematics*, 2, 1119-1123.
- [26] H. P. Crowder and P. Wolfe, (1969). Linear convergence of the conjugate gradient method, *IBM J. Res. Dev.*, 16, pp. 431–433.
- [27] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. (1995). Toint : Constrained and unconstrained testing environments, *ACM Trans. Math. Software*, 21, pp. 123–160.

- 
- [28] J. Sun, X.Q. Yang, and X.D. Chen, (2005). Quadratic cost flow and the conjugate gradient method, *European Journal of Operational Research* 164 104–114.
- [29] J. Sun and J. Zhang, (2001). Global convergence of conjugate gradient methods without line search, *Ann. Oper. Res.*, 163 , pp. 161–173
- [30] J. C Gilbert, J. Nocedal (1992), Global convergence properties of conjugate gradient methods for optimization, *SIAM J.Optim.*2 ;21-42
- [31] J. Z. Zhang and C. X. Xu, (2001). Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations, *J. Comput. Appl. Math.*, 137, pp. 269–278.
- [32] J. Z. Zhang, N. Y. Deng, and L. H. (1999). Chen, new quasi-Newton equation and related methods for unconstrained optimization, *J. Optim. Theory Appl.*, 102.pp. 147–167.
- [33] J. M. Perry, (1977). A class of conjugate gradient algorithms with a two-step variable-metric memory, Discussion Paper 269, Center for Mathematical Studies in Economics and Management Sciences, Northwestern University, Evanston, Illinois, (1977).
- [34] J. L. Nazareth, (1999). Conjugate-gradient methods, *Encyclopedia of Optimization*, C. Floudas and P. Pardalos, eds., Kluwer Academic Publishers, Boston,
- [35] J. L. Nazareth, (1977). A conjugate direction algorithm without line searches, *J. Optim. Theory Appl.*, 23 . pp. 373–387.
- [36] J. Y. Han, G. H. Liu, and H. X. Yin, (1998). Convergence properties of conjugate gradient methods with strong Wolfe linesearch, *Systems Sci. Math. Sci.*, 11 , pp. 112–116.
- [37] J. W. Daniel, (1967). The conjugate gradient method for linear and nonlinear operator equations, *SIAM J. Numer. Anal.*, 4 , pp. 10–26.
- [38] L. Grippo and S. Lucidi, (1966). A globally convergent version of the Polak-Ribière conjugate gradient method, *Math. Prog.*, 78 , pp. 375–391.
- [39] L. Armijo, (1966). Minimization of function having lipschitz continous first partial derivatives, *Pacific Journal of Mathematics*, Vol. 16, pp.1-3.
- [40] L. 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
- [41] L. Grippo and S. Lucidi, (2003). Convergence conditions, line search algorithms and trust region implementations for the Polak-Ribière conjugate gradient method, Technical Report 25-03, Dipartimento d'Informatica et Sistemistica, Università di Roma "La Sapienza", November .

- 
- [42] M. Rivaie, M. Mamat , L. W. (2012). June ,and I. Mohd, A new class of nonlinear conjugate gradient coefficients with global convergence properties. *Applied Mathematics and Computation* 218 , 11323–11332.
- [43] M. J. D. Powell, (1984). Nonconvex minimization calculations and the conjugate gradient method, *Numerical Analysis, Lecture Notes in Mathematics*, Vol. 1066, Springer-Verlag, Berlin, pp. 122–141.
- [44] M. J. D. Powell,(1977). Restart procedures of the conjugate gradient method, *Math. Prog.*, 2, pp. 241–254.
- [45] M. J. D. Powell, (1976). Some convergence properties of the conjugate gradient method, *Math. Prog.*, 11, pp. 42–49.
- [46] M. R. Hestenes and E. L. Stiefel, (1952). Methods of conjugate gradients for solving linear systems, *J. Research Nat. Bur. Standards*, 49, pp. 409–436.
- [47] M. S. Bazaraa and H. D. Sherali, et C. M. Shetty , (1993). *Nonlinear Programming, Theory and Algorithms*, Wiley-Interscience(1993).
- [48] M. Belloufi and R. Benzine, (2013). Global Convergence Properties of the HS Conjugate Gradient Method. *Applied Mathematical Sciences*, Vol. 7, 142. pp. 7077-7091
- [49] M. Belloufi, Rachid Benzine, (2014). Descent property and global convergence of a new search direction method for unconstrained optimization. *Numerical Functional Analysis and optimization*
- [50] M. Belloufi, R. Benzine, and Y. Laskri, (2015). Modification of the Armijo line search to satisfy the convergence properties of HS method, *An International Journal of Optimization and Control : Theories Applications (IJOCTA)*, 141, pp. 145-152 .
- [51] N. Andrei, (2008). An unconstrained optimization test function collection, *Adv. Model. Optim.* 10;147-161.
- [52] N. Andrei, (2009). Open problems in nonlinear conjugate gradient algorithms for unconstrained optimization, *ICI Technical, Report*, 8-13.
- [53] N. Andrei,(2009). Accelerated conjugate gradient algorithm with finite difference Hessian/ vector product approximation for unconstrained optimization. *J. Comput. Appl. Math.* 320;570-582.
- [54] P. Wolfe, ((1969). Convergence conditions for ascent method, *SIAM Rev* 11 ;149-154.
- [55] P. Wolfe, (1971). Convergence conditions for ascent methods. II : Some corrections, *SIAM Review*, 13. pp. 185–188.

- 
- [56] P. Wolfe, (1961). A duality theorem for nonlinear programming, *Quart. Appl. Math.*, 19 , pp. 239-244.
- [57] R. Fletcher, (1987). *Practical Methods of Optimization vol. 1 : Unconstrained Optimization*, John Wiley Sons, New York,
- [58] Rivaie, M.,Mamat, M., Lelong, W.J. and Mohd,I.(2012). Anew class of nonlinear conjugate gradient with global convergence properties. "*Appl.Math. comput.*218 ;11323-11332.
- [59] R. Fletcher and C. Reeves, (1964). Function minimization by conjugate gradients, *Comput. J.*, 7, pp. 149–154.
- [60] R. Fletcher and M. J. D. Powell, (1963). A rapidly convergent descent method for minimization, *Computer. J.*, 6 , pp. 163-168.
- [61] R. A. Horn, and C. R. (1990). *Johnson, Matrix Analysis*, Cambridge University Press,
- [62] W. W. Hager and H. Zhang, (2004). CG DESCENT, a conjugate gradient method with guaranteed descent, January 15, (to appear in *ACM TOMS*).
- [63] W.W. Hager. H.C. Zhang, (2003). Anew conjugate gradient method with guaranteed descent and efficient line search, *SIAM J.Optim*16 ;170-192.
- [64] X.D. Chen and J. Sun, (2002). Global convergence of a two-parameter family of conjugate gradient methods without line search, *Journal of Computational and Applied Mathematics* 146 , pp. 37–45.
- [65] Y. H. Dai, (1997). *Analyses of conjugate gradient methods*, Ph.D. thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences.
- [66] Y. H. Dai, (2001). New properties of a nonlinear conjugate gradient method, *Numer. Math.*, 8 , pp. 83–98.
- [67] Y. H. Dai, (2002)A nonmonotone conjugate gradient algorithm for unconstrained optimization, *J. Syst. Sci. Complex.*, 15 , pp. 139–145.
- [68] Y. H. Dai, J. Y. Han, G. H. Liu, D. F. Sun, H. X. Yin, and Y. Yuan, (1999). Convergence properties of nonlinear conjugate gradient methods, *SIAM J. Optim.*, 10, pp. 345–358.
- [69] Y. H. Dai and L. Z. Liao, (2001). New conjugacy conditions and related nonlinear conjugate gradient methods, *Appl. Math. Optim.*, 43, pp. 87–101.
- [70] Y. H. Dai and Q. Ni, (2003). Testing different conjugate gradient methods for large-scale unconstrained optimization, *J. Comput. Math.*, 21, pp. 311–320.

- 
- [71] Y. H. Dai and Y. Yuan, (1996). Convergence properties of the Fletcher-Reeves method, *IMA J. Numer. Anal.*, 16 , pp. 155–164.
- [72] Y. H. Dai and Y. Yuan, (1995). Further studies on the Polak-Ribière-Polyak method, Research report ICM-95-040, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences.
- [73] Y. H. Dai and Y. Yuan, (1999). A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optim.*, 10 , pp. 142–148.
- [74] Y. H. Dai and Y. Yuan, (1998). A class of globally convergent conjugate gradient methods, Research report ICM-98-030, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences,
- [75] Y. H. Dai and Y. Yuan, (1998). Extension of a class of nonlinear conjugate gradient methods, Research report ICM-98-049, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences,
- [76] Y. H. Dai and Y. Yuan, (2001). A three-parameter family of hybrid conjugate gradient method, *Math. Comp.*, 70 , pp. 1155–1167.
- [77] Y. H. Dai and Y. Yuan, (2001). An efficient hybrid conjugate gradient method for unconstrained optimization, *Ann. Oper. Res.*, 103, pp. 33–47.
- [78] Y. H. Dai and Y. Yuan, (2003). A class of globally convergent conjugate gradient methods, *Sci. China Ser. A*, 46, pp. 251–261.
- [79] Y. F. Hu and C. Storey, (1991). Global convergence result for conjugate gradient methods, *J. Optim. Theory Appl.*, 71 , pp. 399–405.
- [80] Y. Liu and C. Storey, (1991). Efficient generalized conjugate gradient algorithms, Part 1 : Theory, *J. Optim. Theory Appl.*, 69 , pp. 129–137.
- [81] Y. Yuan, (1993). *Numerical Methods for Nonlinear Programming*, Shanghai Scientific Technical Publishers.
- [82] Y. Yuan, (1993). Analysis on the conjugate gradient method, *Optim. Methods Softw.*, 2, pp.19–29
- [83] Z.J. Shi and J. Shen, (2005). Convergence of descent method without line search, *Appl. Math. Comput.* 167 (2005) 94–107.
- [84] Z.J. Shi, (2002). Restricted PR conjugate gradient method and its global convergence, *Advances in Mathematics* 31 (1) (2002) 47–55 (in Chinese).
- [85] Z.J. Shi and J. Shen, (2007). Convergence of Liu-Storey conjugate gradient method, *European Journal of Operational Research*, 182(2), 552– 560 .

---

[86] Z.J. Shi and J. Shen, (2007). Convergence of Polak-Rebière-Polyak conjugate gradient method, *Nonlinear Analysis*, 66 (2007), 1428–1441.

[87] Z. Wei, S. Yao and L. Liu, (2006). The Convergence Properties of Some New Conjugate Gradient Methods, *Applied Mathematics and Computation*, Vol. 183, No. 2, 2006, pp.1341-1350.