وزارة التعليم العالي والبحث العلمي

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

**BADJI MOKHTAR-ANNABA**

**UNIVERSITY**

**UNIVERSITE BADJI MOKHTAR ANNABA**


جامعة باجي مختار

-عنابة-

**Faculté des Sciences**

**Année : 2024/2025**

**Département de Mathématiques**


# THÈSE

Présentée en vue de l'obtention du diplôme de Doctorat
en sciences

**Etude sur les méthodes du gradient conjugué et quasi Newton en utilisant la recherche linéaire non monotone.**

**Filière**
**Mathématiques**

**Spécialité**
**Mathématiques appliquées**


**Par**
**Hadji Ghania**

**DIRECTEUR DE THÈSE:** Laskri Yamina **Prof ENSTI -Annaba**


**Devant le jury**

| | | | |
|---|---|---|---|
| **PRESIDENT :** | Kilani Brahim | **MCA** | **UBM Annaba** |
| **EXAMINATEUR :** | Lakhdari Abdelghani | **MCA** | **ENSTI -Annaba** |
| **EXAMINATEUR :** | Sahari Mohamed Lamine | **MCA** | **UBM Annaba** |
| **EXAMINATEUR :** | Boulares Hamid | **Prof** | **Université Guelma** |
| **EXAMINATEUR** | Boumediene Amina | **MCA** | **UMCM Souk Ahras** |

# Etude sur les méthodes du gradient conjugué et quasi Newton en utilisant la recherche linéaire non monotone.

**Hadji Ghania**
*Université Badji Mokhtar Annaba*
*Département de mathématiques*

2024/2025

Directeur du thèse : Laskri Yamina (Prof)

# الملخص

في هذه الاطروحة، نقترح طريقة هجينة جديدة تعتمد على التدرج المترافق لحل مشكلات الأمثلة الكبيرة غير المقيدة. تدمج هذه الطريقة بشكل محدب بين تقنيتين غير خطيتين للتدرج المترافق وهما طريقة بولاك- ريبيار-بولاياك وطريقةارميل.

من خلال الجمع بين هاتين الطريقتين، نحصل على اتجاه تنازلي في كل تكرار، مما يضمن التقارب الكلي وفقا لشروط وولف القوية.

تظهر الاختبارات العددية التي أجريت فعالية وقوة هذه الطريقة الجديدة.

**الكلمات المفتاحية**

الأمثلة غير المقيدة، البحث الخطي لوولف، طريقة التدرج المترافق الهجينة, التقارب الكلي

# Résumé

Dans cette thèse, nous proposons une nouvelle méthode hybride de gradient conjugué pour résoudre des problèmes d'optimisation de grande taille sans contraintes. Cette approche combine de manière convexe deux techniques non linéaires de gradient conjugué déjà existantes ;
la méthode PRP (Polak-Ribiere-Polyak) et la méthode RMIL+. En réunissant ces deux méthodes, nous obtenons une direction de descente à chaque itération, ce qui assure une convergence globale sous les conditions de Wolfe fortes. Les tests numériques réalisés illustrent l'efficacité et la robustesse de la méthode de cette nouvelle approche.

**Mots clés :** Optimisation sans contraintes, recherche linéaire de Wolfe, méthode du gradient conjugué hybride, convergence globale.

# Abstract

In this thesis, we propose a new hybrid conjugate gradient method for solving large unconstrained optimization problems. This approch convexly combines two existing nonlinear conjugate gradient methods (the PRP method and the RMIL+ method), which produces a descent direction at each iteration and converges globally under strong Wolfe conditions. Numerical experiments are performed to test the efficiency of the new method, which confirms the power of this method.

**Key words** :Unconstrained optimization , Wolfe linear search, hybrid conjugate gradient method, global convergence.

.

# Remerciements

*Avant tout, je remercie "Allah" le tout puissant pour nous avoir donné l'opportunité et de me donner la force et la patience pour terminer ce travail.*

*Je tiens tout d'abord à remercier profondément mon encadreur Madame* **Laskri Yamina**, *professeur à l'ENSTI Annaba, pour la gentillesse et la patience qu'elle a manifesté à mon égard. Je la remercie, aussi, pour m'avoir guidé, encouragé, conseillé, tout au long de la réalisation de cette thèse.*

*Je tiens à remercier également Mr* **Kilani Brahim** professeur *à l'université Badji Mokhtar Annaba, pour l'honneur qu'il m'a fait en acceptant de présider le jury et aussi pour tout ce qu'il m'a appris.*

*Je remercie aussi tous les membres du jury pour leurs contributions. Je cite,Mr* ***Lakhdari Abdelghani*** *maître de conférences à l'ENSTI Annaba, Mr* ***Sahari Mohamed Lamine***, *maître de conférences à l'université Badji Mokhtar de Annaba, Mr* ***Boulares Hamid*** *professeur à l'université de Guelma et Mme Boumediene Amina maître de conférences à l'université de Souk Ahras, qui ont pris de leurs temps pour lire et juger ce travail, ainsi que pour leur déplacement le jour de la soutenance.*

*Mes remerciements ne seraient pas complets sans mentionner* **Pr. Benzine Rachid** *dont je lui souhaite la bonne santé.*

*Je n'oublie pas de remercier toutes les personnes qui m'ont facilité la tâche et tous ceux que j'ai connu au département de mathématiques d' Annaba et de Souk Ahras, mentionnant particulièrement Mr* ***Barrouk Bachir***.

*Comme je remercie aussi ma grande famille pour son soutien et pour son aide dans la préparation de cette thèse et pour leurs encouragements... Merci !.*

# Dédicace

*Je dédie ce modeste travail :*

*À ma chère mère* **Yamina**

*À la mémoire de mon chère père*

*À la mémoire de mon époux*

*À mes enfants* ***Houcem eddine***, ***Zakaria***, ***Seif elislam*** *et* ***Meriem Nour*** *source de ma plus grande fierté,*

# TABLE DES MATIÈRES

# Introduction générale

L'optimisation consiste à trouver les meilleurs résultats possibles pour une fonction donnée, souvent en minimisant ou en maximisant celle-ci en fonction de certaines variables

Dans la vie quotidienne, nous somme toujours confrontés à des problèmes d'optimistation simple ou bien complexe ;car on cherche une meilleur solution aux problémes qui jalonnent notre présence ; où on veut améliorer le temps de travail, nos zones de stockage ou encore le chemin que nous devons emprunter pour nous rendre quelque part.

Cette amélioration peut être utilisée pour résoudre une variété de problèmes procédés, comme planifier les ressources, prendre des décisions financières, planifier la production et allouer les ressources. Elle peut également être utilisée pour aborder des problèmes en science et dans d'autres domaines.

La maximisation des profits dans les entreprises en modifiant les tarifs et les quantités de production

La réduction des coûts de production consiste à identifier les méthodes les plus efficaces pour fabriquer des biens ou offrir des services. Il est également possible de diminuer les coûts liés au transport et au stockage en planifiant la logistique et la chaîne d'approvisionnement,.

Les services de livraison et de transport peuvent réduire les temps de trajet et les dépenses en carburant en adoptant des stratégies d'optimisation des itinéraires. Cela repose sur plusieurs techniques et outils :

1. Outils de planification des itinéraires :

Ces systèmes s'appuient sur des algorithmes avancés pour déterminer les trajets les plus efficaces en tenant compte de divers paramètres tels que :

Le trafic en temps réel

La distance

Les conditions météorologiques

Les limitations ou fermetures de routes

2. Données en temps réel :

Les technologies GPS et les informations sur le trafic permettent d'ajuster les trajets à la minute

près, évitant ainsi les congestions et les retards.

3. Consolidation des livraisons :

Le regroupement des commandes destinées à des zones proches permet d'optimiser les déplacements et de réduire les trajets superflus.

4. Optimisation des horaires :

Planifier les livraisons durant les périodes de faible circulation contribue à réduire les temps de trajet et les coûts d'usure des véhicules.

5. Usage de l'intelligence artificielle :

Les systèmes basés sur l'apprentissage automatique analysent les données historiques et actuelles pour prévoir les tendances et ajuster les itinéraires de manière optimale.

Avantages de l'optimisation des trajets :

Diminution des dépenses liées au carburant

Amélioration de l'efficacité des conducteurs

Réduction de l'impact environnemental

Amélioration de la satisfaction des clients grâce à une meilleure ponctualité

La répartition des ressources dans les projets, y compris la gestion du personnel et des équipements, est cruciale pour améliorer l'efficacité et limiter les retards.

La mise en place de réseaux de télécommunications vise à assurer la vitesse et la fiabilité des communications, tout en réduisant les coûts d'infrastructure.

Optimiser les campagnes marketing permet de cibler de manière efficace les publics et d'optimiser le retour sur investissement.

La conception de portefeuilles d'investissement pour maximiser le rendement tout en minimisant le risque.

L'optimisation des traitements médicaux pour personnaliser les thérapies et maximiser les chances de succès tout en minimisant les effets secondaires.

L'optimisation des processus industriels pour maximiser la production et minimiser les déchets et les temps d'arrêt.

Il existe deux types principaux d'optimisation mathématique : l'optimisation ou amélioration sans contraintes et l'optimisation avec contraintes. Dans la peremière ; il n'y a pas de restrictions

sur les variables, tandis que dans la seconde,il existe des limitations sur les variables qui doivent être respectées pour obtenir une solution valide.

L'histoire de l'optimisation sans contraintes remonte à plusieurs siècles, bien que les concepts modernes d'optimisation sans contraintes aient émergé au 20 ème siècle avec le progrès de la théorie mathématique et la croissance des ordinateurs.

Au 19 ème siècle, les mathématiciens ont commencé à étudier les méthodes pour maximiser ou minimiser des fonctions sans restrictions, ce qui a donné naissance à la notion d'optimisation sans contraintes. Cependant, les méthodes utilisées à l'époque étaient souvent inefficaces et limitées en termes de complexité et de précision.

Au cours du 20 ème siècle, les mathématiciens et les informaticiens ont commencé à développer des algorithmes plus avancés pour l'optimisation sans contraintess, en utilisant des techniques telles que la descente du gradien et la méthode du gradient conjugué. Les ordinateurs ont également permis de traiter les données plus rapidement et de manière plus précise, ce qui a contribué à la croissance de l'amélioration sans contraintes.

D'une manière formelle, il est possible d'écrire le problème d'optimisation sans contraintes. noté $(P)$ comme suit :

Soit $J : \mathbb{R}^n \to \mathbb{R}$. Notre objectif est de créer une solution pour aborder le problème de minimisation suivant, qui ne présente aucune contrainte :

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

En trouvant un élément $\hat{x}$ de $\mathbb{R}^n$ tel que $J(\hat{x}) \leq J(x)$ pour tout $x \in \mathbb{R}^n$.

L'introduction de la méthode du gradient conjugué a été remonté à 1952. par Hestenes et Stiefel afin de réduire au minimum les fonctions quadratiques purement convexes.Plusieurs mathématiciens ont par la suite élargi cette méthode aux autres cas :

Fletcher-Reeves (1964) [38] : Formulé par Fletcher et Reeves, cet algorithme est une version consacrée de la descente de gradient conjuguée. Il calcule les coefficients de conjugaison en utilisant la norme du gradient actuel et précédent.

Technique Polak-Ribière-Polyak (1969)[34] : Proposée par Polak, Ribière et Polyak, cette technique valorise Fletcher-Reeves. Il modifie sa trajectoire en réponse à la fluctuation du gradient,

aidant ainsi à gérer des scénarios spécifiques dans lesquels l'algorithme de Fletcher-Reeves pourrait faiblir.

Fletcher (1987)[36] a examiné les techniques de convergence des lignées, évaluant leur efficacité et leurs contraintes dans divers états.

Ces procédures sont essentielles pour aborder des tâches d'optimisation sans restriction, en particulier dans les scénarios où l'ampleur du problème rend les techniques de gradient traditionnelles inapplicables.

Toutes ces variations créent une succession $\{x_k\}_{k\in\mathbb{N}}$ d'après la relation ci-dessous. :

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

Une optimisation unidimensionnelle ou une recherche linéaire, qu'elle soit précise ou inexacte, permet de déterminer le pas $\lambda_k \in \mathbb{R}$ .

On effectue des calculs récurrents des directions $d_k$ en utilisant les formules qui suivent :

$$d_k = \begin{cases} -R_0 & \text{si } k = 0 \\ -R_k + \beta_{k-1} d_{k-1} & \text{si } k \geq 1 \end{cases}.$$

$R_k = \nabla J(x_k)$ et $\beta_k \in \mathbb{R}$.

Si on note $y_{k-1} = R_k - R_{k-1}$,

Méthode de Polak-Ribière-Polyak

$$\beta_k^{PRP} = \frac{R_{k+1}^T y_k}{\|R_k\|^2},$$

Méthode de Fletcher-Reeves

$$\beta_k^{FR} = \frac{\|R_{k+1}\|^2}{\|R_k\|^2},$$

$$\beta_k^{CD} = \frac{\|R_{k+1}\|^2}{-d_k^T R_k}, \text{Gradient descendant conjugué}$$

$$\beta_k^{DY} = \frac{\|R_{k+1}\|^2}{d_k^T y_k}, \text{GC de } \textit{Dai-Yuan.}$$

$$\beta_k^{HS} = \frac{R_{k+1}^T y_k}{d_k^T y_k}, \text{GC variante Hestenes-Stiefel.}$$

$$\beta_k^{LS} = \frac{R_{k+1}^T y_k}{R_k^T d_k}, \text{Variante du GC proposée par Liu et Storey}$$

$$\beta_k^{HZ} = \frac{R_{k+1}}{d_k^T y_k}\left(y_k - 2d_k\frac{||y_k||^2}{d_k^T y_k}\right)^T, \text{GC de Hager et Zhang}$$

$$\beta_k^{RMIL} = \frac{R_k^T\left(R_k - R_{k-1}\right)}{\|d_{k-1}\|^2} \text{ GC selon la variante Rivaie-Mustafa-Ismail-Leong(RMIL)}$$

$$\beta_k^{RMIL+} = \frac{R_k^T\left(y_{k-1} - d_{k-1}\right)}{\|d_{k-1}\|^2}$$

Cette thèse contient une introduction et cinq chapitres.

## Chapitre 1

Ce chapitre présente les principes fondamentaux des techniques d'optimisation sans contraintes qui reposent sur les directions de descente et les recherches linéaires. Ces techniques créent des ensembles de points$\{x_k\}_{k\in\mathbb{N}}$ qui se dirigent vers un minimum de la fonction à optimiser.

## Chapitre 2

Les méthodes à directions de descente sont présentées dans ce chapitre. Ce sont des algorithmes d'optimisation utilisés pour minimiser une fonction objectif. Ces méthodes se basent sur l'idée de chercher une solution en se déplaçant dans une direction qui réduit la valeur de la fonction à chaque itération. On trouve ses applications en apprentissage automatique , en ingénierie et en mathématiques appliquées pour résoudre des problèmes d'optimisation ou d'équations non linéaires.

## Chapitre 3

L'algorithme de gradient conjugué est présenté dans ce chapitre , c'est une technique employée pour résoudre des problèmes d'optimisation quadratique, surtout lorsque la matrice du système est grande, symétrique et définie positive. Cette méthode se révèle efficace car elle évite le stockage et l'inversion explicite des matrices, ce qui la rend particulièrement adaptée aux systèmes de grande taille.

## Chapitre 4

Ce chapitre donne une importance capitale dans cette étude car il renferme les résultats de convergence essentiels concernant les méthodes de minimisation sans contraintes utilisant le gradient conjugué préféré pour les systèmes linéaires bien conditionnés ou utilisant un préconditionneur, les méthodes classiques adaptées aux problèmes simples ou aux données bien conditionnées et les méthodes hybrides qui sont idéales pour les problèmes complexes ou mal

conditionnés, nécessitant flexibilité et robustesse , qui proposent des approches variées avec des caractéristiques et des performances distinctes.

L'analyse comparative constitue une méthode précieuse pour évaluer diverses approches, outils ou stratégies en les confrontant selon des critères spécifiques.

Au dernier chapitre, on s' intéresse aux resultats numériques de notre étude, nous allons comparer la nouvelle méthode HLB avec les deux autre méthodes PRP et RMIL+ en fonction du temps et du nombre d'itérations.

# CHAPITRE 1

## Rappels

L'ojectif de l'optimisation sans contraintes est de trouver un minimum d'une fonction $J$ appelée fonction objectif mais en pratique, selon les propriétés de $J$, il est rare de pouvoir trouver un minimum global, ni même de pouvoir assurer que la solution candidate obtenue est un minimum local. Ainsi, les méthodes d'optimisation traditionnelles recherchent plutôt un point stationnaire, c'est-à-dire un point où le gradient s'annule. Néanmoins, dans certains cas, il est possible de savoir si un minimum local ou global existe et s'il est unique.

La formulation générale d'un problème d'optimisation nécessite :

-Une fonction objectif , fonction de coût ou un critère à minimiser, notée $J : \mathbb{R}^n \to \mathbb{R}$

- $S$ qui est un ensemble des éléments admissibles pour un problème, ou bien l'ensemble des contraintes il peut s'agir d' un ouvert non vide de $\mathbb{R}^n$,d' un convexe inclu dans $\mathbb{R}^n$ où bien d'un ensemble qui est défini par des égalités et (ou) des inégalités fonctionnelles dans lequel nous allons chercher l'optimum.

Minimiser la fonction $J$ sur l'ensemble $S$ signifie trouver un élément $x* \in S$ tel que : $J(x^*) \leq J(x)$, pour tout $x \in S$. En d'autres termes, $x^*$ est une solution admissible qui rend la valeur de la fonction $J$ aussi petite que possible parmi toutes les solutions appartenant à $S$.

$x*$ est la solution optimale du problème d'optimisation. $J(x^*)$ représente la valeur minimale de la fonction objectif. L'objectif est de déterminer cette solution $x^*$en respectant les contraintes

définissant $S$.

$$J(x^*) = \min_{y \in S} J(y) \Longleftrightarrow J(x^*) \leq J(y), \forall y \in S. \tag{1.0.1}$$

Considérant donc le problème formulé de la manière suivante. :

$$\min \left\{ J\left(x\right) : x \in \mathbb{R}^n \right\}. \tag{1.0.2}$$

Tel que $J$ est de la forme quadratique :

$$J(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle \tag{1.0.3}$$

$$= \frac{1}{2}x^t Ax - b^t x, x \in \mathbb{R}^n, A \in M_{n*n}. \tag{1.0.4}$$

Ou bien $J$ est une fonction quelconque.

Un algorithme de descente est une méthode d'optimisation qui vise à trouver le minimum d'une fonction f en ajustant progressivement les valeurs de ses paramètres. À partir d'un point de départ $x_0$ , il produit une série de points $(x_k)_{k\in\mathbb{N}}$ qui devraient idéalement se rapprocher du minimum souhaité.

$$x_{k+1} = x_k + \lambda_k d_k, \tag{1.0.5}$$

et telle que :

$$\forall k \in \mathbb{N}, J(x_{k+1}) \leq J(x_k).$$

$$\forall k \in \mathbb{N}, J(x_k + \lambda_k d_k) < J(x_k). \tag{1.0.6}$$

Où $d_k$ est un vecteur non nul de $\mathbb{R}^n$ choisi à chaque itération, qui indique la direction dans laquelle on souhaite ajuster pour réduire la valeur de la fonction appelé direction de descente, et

la recherche linéaire $\lambda_k$ est un scalaire strictement positif (également appelé pas de descente ou longueur de pas) qui détermine l'ampleur du déplacement dans la direction choisie.

## 1.1 Existence et unicité de problème $(p)$

Avant d'étudier les propriétés de la solution du problème$(P)$, il faut s'assurer de leur existence. Le théorème le plus connu sur l'existence des minima et des maxima est :

**Théorème 1.1.1** ***(Théorème de Weierstrass )***

*Soient $S$ un ensemble compact (fermé et borné) non vide de $\mathbb{R}^n$ et $J : S \to \mathbb{R}$ une fonction continue sur $S$, alors $J$ accepte au moins un min $\hat{x}$ sur $S$. c.a.d, il existe un point $\hat{x}$ de $S$ minimum global de $J$ sur $S$ i.e. :*

$$\forall x \in S \quad J(\hat{x}) \leq J(x). \tag{1.1.1}$$

**Preuve.** Soit $(x_n)$ une suite minimise $J$ sur $S$ c'est-à-dire d'éléments de $S$ telle que

$$x_n \in S, \forall n \in \mathbb{N} \text{ et } \lim_{n\to+\infty} J(x_n) = \inf_{x\in S} J(x).$$

Comme $S$ est borné, la suite minimisante soit bornée, donc on peut extraite une sous-suite notée $(x_n)$ convergente vers un élément $\hat{x}$ de $S$ car $S$ est un ensemble fermé. Cette suite extraite vérifie

$$\lim_{n\to+\infty} J(x_n) = J(\hat{x}).$$

Car $J$ est continue, d'où par unicité de la limite

$$J(\hat{x}) = \inf_{x\in S} J(x).$$

Donc $J$ réalise son optimum dans $S$. ■

**Théorème 1.1.2** ***(Théorème d'existence)***

*Soit $J : \mathbb{R}^n \to \mathbb{R}$ une fonction continue et coercive alors $J$ admet au moins un minimum sur $\mathbb{R}^n$. Autrement dit ,il existe un point $\hat{x}$ de $\mathbb{R}^n$ minimum global de $J$ sur $\mathbb{R}^n$*

$$\forall x \in \mathbb{R}^n, J(x) \geq J(\hat{x}). \tag{1.1.2}$$

**Preuve.** Soit $(x_n)$ une suite minimisante dans $\mathbb{R}^n$ c'est à dire

$$\lim_{n\to+\infty} J(x_n) = \inf_{x\in\mathbb{R}^n} J(x).$$

Comme $J$ est coercive, la suite $(x_n)$ ne peut pas partir à l'infini donc elle est bornée et il existe une sous-suite extraite notée $(x_n)$ de $\mathbb{R}^n$ qui converge vers un $\hat{x}$ de $\mathbb{R}^n$ : $x_n \to \hat{x} \in \mathbb{R}^n$.
Or $J$ est continue, donc

$$\lim_{n\to+\infty} J(x_n) = J(\hat{x}),$$

d'où par unicité de la limite

$$J(\hat{x}) = \inf_{x\in\mathbb{R}^n} J(x).$$

Donc ; $J$ réalise son minimum dans $\mathbb{R}^n$. ■

**Théorème 1.1.3** ***(Théorème d'unicité)***

*Si de plus $J : \mathbb{R}^n \to \mathbb{R}$ strictement convexe, alors il existe un unique minimum $\hat{x} \in \mathbb{R}^n$ de $J$ tel que :*

$$\forall x \in \mathbb{R}^n, J(x) \geq J(\hat{x}). \tag{1.1.3}$$

**Preuve.** Soit $J$ strictement convexe, supposons qu'il existe $\hat{x}_1$, $\hat{x}_2 \in \mathbb{R}^n$ deux minimum de la fonction $J$ sur $\mathbb{R}^n$ tels que

$$\hat{x}_1 \neq \hat{x}_2 \text{ et } J(\hat{x}_1) = J(\hat{x}_2) = \inf_{x\in\mathbb{R}^n} J(x).$$

Soit

$$\tilde{x} = \lambda\hat{x}_1 + (1-\lambda)\hat{x}_2 \,;\, \tilde{x} \in \mathbb{R}^n, \lambda \in [0,1]\,.$$

Et comme $J$ est strictement convexe

$$\begin{aligned} J(\tilde{x}) &\leq \lambda J(\hat{x}_1) + (1-\lambda)J(\hat{x}_2) \\ J(\tilde{x}) &\leq J(\hat{x}_2) \\ J(\tilde{x}) &\leq \inf_{x\in\mathbb{R}^n} J(x), \end{aligned}$$

Ceci fournit une contradiction, ce qui est impossible ; donc $\hat{x}_1 = \hat{x}_2$. ■

## 1.2 Direction de descente

### 1.2.1 Définition d'une direction de descente

Les techniques d'optimisation itératives, souvent désignées sous le nom de méthodes de descente, sont des approches numériques conçues pour minimiser les valeurs de fonctions différentiables définies sur des espaces euclidiens ou, plus généralement, sur des espaces hilbertiens.

Fonctionnement général :

Ces algorithmes opèrent de manière itérative, améliorant progressivement la solution.

À chaque étape, un mouvement est effectué dans une direction qui assure une diminution de la valeur de la fonction. Les méthodes de descente sont appliquées dans divers contextes , tels que l'apprentissage automatique ou la résolution de problèmes d'ingénierie. Elles peuvent également être adaptées à des cas particuliers, y compris des fonctions non différentiables ou des contraintes, ces algorithmes permettent d'approcher progressivement le seuil minimal requis..

**Définition 1.2.1** *Soit* $J : \mathbb{R}^n \to \mathbb{R}$ *et* $d$ *un vecteur de* $\mathbb{R}^n / \{0\}$. *Le vecteur* $d$ *est dit une direction de descente au point* $\hat{x} \in \mathbb{R}^n$ *si et seulement si*

$$\exists \delta > 0, \forall \lambda \in ]0, \delta[\,, J(\hat{x} + \lambda d) < J(\hat{x}). \tag{1.2.1}$$

**Théorème 1.2.1** *soit* $J : \mathbb{R}^n \to \mathbb{R}$ *une fonction différentiable au point* $\hat{x} \in \mathbb{R}^n$ *et* $d \in \mathbb{R}^n$ *tel que*

$$\langle \nabla J(\hat{x}).d \rangle < 0. \tag{1.2.2}$$

*alors* $d$ *est une direction de descente de* $J$ *en* $\hat{x} \in \mathbb{R}^n$

**Preuve.** Soit $J$ une fonction différentiable au point $\hat{x} \in \mathbb{R}^n$, donc

$$\begin{aligned} J(\hat{x} + \lambda d) &= J(\hat{x}) + \lambda \nabla^T J(\hat{x})\, d + \lambda ||d|| \alpha(\hat{x}, \lambda d) \\ \forall x &\in S, \text{ où } \alpha(\hat{x}, \lambda d) \underset{\lambda \to 0}{\to} 0\ . \end{aligned}$$

Donc

$$\frac{J(\hat{x} + \lambda d) - J(\hat{x})}{\lambda} = \nabla^T J(\hat{x})\, d + ||d|| \alpha(\hat{x}, \lambda d).$$

Et

$$\lim_{\lambda \to 0} \frac{J(\hat{x} + \lambda d) - J(\hat{x})}{\lambda} = \nabla^T J(\hat{x})\, d.$$

Et comme

$$\langle \nabla J(\hat{x}).d \rangle = \nabla^T J(\hat{x})\, d.$$

Et

$$\langle \nabla J(\hat{x}).d \rangle < 0.$$

Donc

$$\begin{aligned} \exists \delta &> 0, \text{ tq } \lim_{\lambda \to 0} \frac{J(\hat{x}+\lambda d) - J(\hat{x})}{\lambda} < 0, \ \forall \lambda \in ]0, \delta[ \\ \exists \delta &> 0, \ \forall \lambda \in ]0, \delta[ \, , J(\hat{x}+\lambda d) < J(\hat{x}) \, . \end{aligned}$$

Donc le vecteur $d$ est une direction de descente. ■

**Remarque 1.2.1** *1- Par définition du gradient, d fait avec l'opposé du gradient* $(-\nabla J(x))$ *un angle* $\theta$ *strictement plus petit que* $90°$ *appelé* ***angle de descente*** *:*

$$\theta = \arccos \frac{-\nabla^T J(x) \, .d}{\|\nabla J(x)\| \, . \, \|d\|} \in \left[0, \frac{\pi}{2}\right[ . \tag{1.2.3}$$

*2- L'ensemble des directions de descente de* $J$ *en* $x$

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

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

**Remarque 1.2.2** *L'angle* $\theta_k$ *entre la direction* $d_k$ *et* $-\nabla J(x_k)$ *joue un rôle important dans le processus de la convergence. Pour* $\theta_k$ *on a la relation classique suivante :*

$$-\nabla^T J(x_k) \, .d_k = \|\nabla J(x_k)\| \, \|d_k\| \cos \theta_k$$

*Supposons* $J : \mathbb{R}^n \to \mathbb{R}$ *différentiable. un vecteur* $d \in \mathbb{R}^n$ *est une direction de descente de* $J$ *au point* $x$ *ssi*

$$\begin{aligned} \forall \beta &< 1, \exists \eta > 0 \text{ tel que :} \\ \forall t &\in ]0, \eta] \, , J(x+td) < J(x) + t\beta \nabla J(x)^T d < J(x). \end{aligned}$$

*Cette inégalité garantit une décroissance au minimum de la fonction* $J$ *dans la direction* $d$*.*

### 1.2.2 Choix de la direction de descente

Dans le domaine des algorithmes d'optimisation, la sélection de la direction de descente peut se faire selon deux approches principales :

Utilisation du gradient

Cette technique repose directement sur le gradient de la fonction que l'on cherche à minimiser. Comme le gradient indique la direction de la plus forte augmentation de la fonction, sa direction opposée constitue naturellement une voie de diminution.

Méthodes courantes :

- Descente de gradient : Cette méthode permet d'approcher vers le minimum, surtout pour les fonctions convexes.

Ces approches tirent parti d'informations supplémentaires, comme la courbure locale ou des propriétés de conjugaison.

- Méthode de Newton : La direction de descente est déterminée en prenant en compte la courbure via la matrice Hessienne. Bien qu'elle soit efficace près d'un minimum, cette méthode peut s'avérer coûteuse en termes de calculs.

- Méthodes de conjugaison : Ces méthodes produisent des directions successives qui respectent certaines propriétés d'orthogonalité ou de conjugaison, ce qui favorise une convergence plus rapide, notamment pour les problèmes quadratiques.

- Approximations de la Hessienne : Les méthodes quasi-Newton, comme BFGS, utilisent des approximations pour réduire les calculs liés à la Hessienne tout en maintenant une efficacité.

### 1.2.3 Critère d'arrêt

Soit $\hat{x}$ un minimum local du critère $J$ à optimiser. Supposons que l'on adopte comme critère d'arrêt dans l'algorithme de descente le modèle idéal : $x_k = \hat{x}$. Dans un scénario parfait (c'est-à-dire en considérant que tous les calculs sont exacts et que la capacité de calcul est illimitée), l'algorithme s'arrêterait après un nombre fini d'itérations ou produirait une suite infinie $x_0, x_1, ..., x_k, ...$ de points dans $\mathbb{R}^n$ qui converge vers $\hat{x}$.

En pratique, il est essentiel de définir un critère d'arrêt pour garantir que l'algorithme se termine toujours après un nombre fini d'itérations et que le dernier point calculé soit suffisamment proche de $\hat{x}$. Dans le cadre de l'optimisation différentiable sans contraintes, nous allons vérifier si

$$||\nabla J(x_k)|| < \varepsilon.$$

### 1.2.4 Algorithmes à Direction de descente

Un algorithme de descente de gradient est caractérisé par les paramètres $d$ et $\lambda$ : la manière dont on calcule la direction de descente $d$ détermine le nom de l'algorithme, tandis que la détermination du pas $\lambda$ est appelée recherche linéaire, qui peut être réalisée de différentes manières à préciser. Un exemple d'algorithme utilisant des directions de descente est illustré par l'algorithme suivant.

La sélection d'une itération de départ $x_0 \in \mathbb{R}^n$ et d'un petit paramètre $\varepsilon$.

Initialisation $k = 0$.

1. Faire le test de convergence : si $||\nabla J(x_k)||_2 < \varepsilon$ arrêt de l'algorithme.

2.Déterminer une direction de descente $d_k$.

3.Trouver un pas $\lambda_k > 0$ assez petit tel que la fonction $J$ décroisse suffisamment.

4. Prendre une nouvelle itération. $x_{k+1} = x_k + \lambda_k d_k$.

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

## 1.3 Conditions d'optimalité

La condition d'optimalité est une condition mathématique qui doit satisfaite pour qu'un point soit un minimum (ou maximum) d'une fonction. Plus précisément, pour qu'un point soit un **minimum local** d'une fonction, il doit vérifier la condition d'optimalité suivante :

★ Le gradient de la fonction en ce point doit être égal à zéro, c'est-à-dire que toutes les dérivées partielles de la fonction doivent être nulles.

En outre, pour qu'un point soit un **minimum global** d'une fonction, il doit vérifier la condition d'optimalité suivante :

★ Le point doit être un minimum local et la fonction doit être convexe dans un voisinage de ce point. Cette condition est appelée la condition de second ordre ou la condition du second gradient.

La condition d'optimalité est importante, car elle permet de trouver des points optimaux d'une fonction en utilisant des algorithmes d'optimisation qui cherchent des points vérifiant cette condition.

### 1.3.1 Condition d'optimalité nécessaires du premier ordre (CON1) :

**Théorème 1.3.1** *Soit $J : \mathbb{R}^n \to \mathbb{R}$ une fonction différentiable au point $\hat{x}$ de $\mathbb{R}^n$. Si $\hat{x}$ est un optimum local du probléme $(P)$ alors*

$$\nabla J(\hat{x}) = 0.$$

**Preuve.** Supposons le contraire c'est-à-dire $\nabla J(\hat{x}) \neq 0$ donc on un vecteur $d = -\nabla J(\hat{x})$ qui est une direction de descente i.e

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

Et $\hat{x} \in \mathbb{R}^n$ est un optimum local du $J$ donc :

$$\exists V_\epsilon(\hat{x}), \forall x \in V_\epsilon(\hat{x}), J(x) \geq J(\hat{x}).$$

Soit $V_\epsilon(\hat{x})$ un voisinage quelconque de $\hat{x}$ :

$$V_\epsilon(\hat{x}) \cap [\hat{x}, \hat{x} + \lambda \nabla J(\hat{x})] \neq \emptyset, \lambda \in ]0, \delta[ \Rightarrow \exists x_\lambda = \hat{x} + \lambda d \in V_\epsilon(\hat{x}), J(x_\lambda) < J(\hat{x}).$$

Contradiction avec le fait que $\hat{x} \in \mathbb{R}^n$ est une solution optimal locale du $(p)$ .
On conclut que $\nabla f(\hat{x}) = 0$. ■

**Remarque 1.3.1** *1- La réciproque du théorème précédent n'est pas vraie.*
*2- Tout point $\hat{x} \in \mathbb{R}^n$ vérifiant $\nabla J(\hat{x}) = 0$ est appelé « point critique » ou « point stationnaire ».*
*3- La relation $\nabla J(\hat{x}) = 0$ est aussi appelée « équation d'Euler ».*
*4- Ce théorème n'a pas de sens si la fonction $J$ n'est pas différentiable.*
*5- Il faut bien noter que la condition du premier ordre ne fournit qu'une condition nécessaire d'optimalité. Autrement dit, l'application de ce critère fournira en général trop de points parmi lesquels il faudra déterminer la ou les solutions du problème $(P)$ usuellement en comparant les valeurs de la fonction objectif $J$ en ces points.*
*6- Si $J$ est convexe, la condition nécessaire du premier ordre est également suffisante pour que $\hat{x} \in \mathbb{R}^n$ soit un minimum de $J$.*

**Théorème 1.3.2** *Soit $J : \mathbb{R}^n \to \mathbb{R}$ une fonction convexe et différentiable au point $\hat{x}$ de $\mathbb{R}^n$. un point $\hat{x}$ réalise un optimum de $J$ sur $\mathbb{R}^n$ si et seulement si $\nabla J(\hat{x}) = 0$.*

**Preuve.** On a vu que la condition est toujours nécessaire, montrons qu'elle est suffisante.
Soit $\hat{x} \in \mathbb{R}^n$ tel que $\nabla J(\hat{x}) = 0$ et comme $J$ est convexe donc

$$\forall x \in \mathbb{R}^n, J(x) \leq J(\hat{x}) + (\nabla J(\hat{x}))^t(x - \hat{x}) \Rightarrow J(x) \geq J(\hat{x})$$

On a donc immédiatement le fait que $\hat{x}$ réalise un minimum de $J$ sur $\mathbb{R}^n$ Nous donnons maintenant une condition nécessaire permettant de préciser encore les éventuels minima. Cette condition va faire intervenir la dérivée seconde de $J$ ■

### 1.3.2 Condition d'optimalité nécessaires du deuxième ordre (CON2) :

**Théorème 1.3.3** *soit $J : \mathbb{R}^n \to \mathbb{R}$ une fonction différentiable au point $\hat{x}$ de $\mathbb{R}^n$.Si $J$ possède un minimum local en $\hat{x}$ alors :*
*1-* $\nabla J(\hat{x}) = 0$.
*2-La matrice Hessienne $H(\hat{x})$ soit semi définie positive.*

**Preuve.** Soit $J$ une fonction deux fois différentiable au point $\hat{x}$ de $\mathbb{R}^n$ et $d$ un vecteur de $\mathbb{R}^n$ , donc pour $\lambda$ assez petit, on a

$$\begin{aligned} J(\hat{x} + \lambda d) &= J(\hat{x}) + \lambda(J(\hat{x}), d) + \frac{1}{2}\lambda^2.d^T H(\hat{x}).d + \lambda^2||d||^2.\alpha(\hat{x}, \lambda d), \\ \forall x &\in S, \text{ ou } \alpha(\hat{x}, \lambda d) \underset{\lambda \to 0}{\to} 0. \end{aligned}$$

Le point $\hat{x} \in \mathbb{R}^n$ est le minimum de $J$ c'est-à-dire $\nabla f(\hat{x}) = 0$

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

$$\Rightarrow \frac{1}{2}d^T H(\hat{x}).d + ||d||^2.\alpha(\hat{x}, \lambda d) \geq 0, \forall \lambda \in ]0, \delta[ .$$

Par passage à la limite lorsque $\lambda \to 0 \Rightarrow \frac{1}{2}d^T H(\hat{x}).d \geq 0, \forall d \in \mathbb{R}^n$.
Donc, la matrice Hessienne $H(\hat{x})$ est semi définie positive. ■

### 1.3.3 Condition d'optimalité suffisante du deuxième ordre (COS2) :

Les conditions données précédemment sont nécessaires, c'est-à-dire qu'elles doivent être satisfaites pour tout minimum local, cependant, tout vecteur vérifiant ces conditions n'est pas nécessairement un minimum local. Le théorème suivant établit une condition suffisante pour qu'un vecteur soit un minimum local, si $J$ est deux fois continuellement différentiable.

**Théorème 1.3.4** *Soit $J : \mathbb{R}^n \to \mathbb{R}$ une fonction deux fois différentielle au point $\hat{x} \in \mathbb{R}^n$. Si $\nabla J(\hat{x}) = 0$ et la matrice Hessienne $H(\hat{x})$ est définie positive, alors $J$ posséde un minimum local stricte en $\hat{x} \in \mathbb{R}^n$.*

**Preuve.** Soit $J$ une fonction deux fois différentiable au point $\hat{x} \in \mathbb{R}^n$, donc $J$ s'écrit sous la forme :

$$\begin{aligned} J(x) &= J(\hat{x}) + \nabla^T J(\hat{x})(x-\hat{x}) + \frac{1}{2}(x-\hat{x})^T H(\hat{x})(x-\hat{x}) + ||x-\hat{x}||^2.\alpha(\hat{x}, x-\hat{x}), \\ \forall x &\in \mathbb{R}^n, \text{ ou } \alpha(\hat{x}, \lambda d) \underset{\lambda \to 0}{\to} 0. \end{aligned}$$

Supposons que $\hat{x} \in \mathbb{R}^n$ n'est pas un minimum local stricte c'est-à-dire

$$\forall V_\epsilon(\hat{x}), \exists x_k \in V_\epsilon(\hat{x}), J(x_k) \leq J(\hat{x})$$

Notons

$$d_k = \frac{x_k - \hat{x}}{||x_k - \hat{x}||^2} \text{ , donc } ||d_k||^2 = 1$$

D'aprés le théorème de Bolzano Weierstrass

$$\exists N_1 \in N, d_k \underset{k \to \infty}{\to} \tilde{d}, k \in N_1.$$

Donc

$$J(x_k) - J(\hat{x}) = \frac{1}{2}(x_k - \hat{x})^T H(\hat{x})(x_k - \hat{x}) + ||x_k - \hat{x}||^2.\alpha(\hat{x}, x_k - \hat{x}).$$

$$\frac{J(x_k) - J(\hat{x})}{||x_k - \hat{x}||^2} = \frac{1}{2} d_k^T H(\hat{x}) d_k + \alpha(\hat{x}, \lambda d).$$

Or

$$J(x_k) \leq J(\hat{x}) \Rightarrow J(x_k) - J(\hat{x}) \leq 0.$$

Donc

$$\forall k, \frac{1}{2} d_k^T H(\hat{x}) d_k + \alpha(\hat{x}, \lambda d) \leq 0.$$

Passant à la limite

$$k \to +\infty \Rightarrow \tilde{d}^T H(\hat{x}) \tilde{d} \leq 0, \tilde{d} \neq 0 \text{ car } d_k \to \tilde{d} \text{ et } ||d_k|| = 1.$$

Donc, la matrice hessienne $H(\hat{x})$ n'est pas définie positive. ■

## 1.4 La recherche linéaire

On a vu que dans le cas non-quadratique les méthodes de descente :

$$x_{k+1} = x_k + \lambda d_k; \qquad \lambda > 0, \tag{1.4.1}$$

nécessitent la recherche d'une valeur de $\lambda > 0$ optimale ou non, vérifiant :

$$J(x_k + \lambda d_k) \leq J(x_k). \tag{1.4.2}$$

On définit la fonction $\varphi_k(\lambda) = J(x_k + \lambda d_k)$.

Si $J$ est différentiable, le pas optimal $\hat{\lambda}$ peut être caractérisé par

$$\begin{cases} \varphi'\left(\hat{\lambda}\right) = 0, \\ \varphi\left(\hat{\lambda}\right) \leq \varphi(\lambda), \quad \text{pour} \quad 0 \leq \lambda \leq \hat{\lambda}, \end{cases} \tag{1.4.3}$$

autrement dit, $\hat{\lambda}$ est un minimum local de $\varphi$ qui assure de plus la décroissance de $J$.

Le choix de ce pas répond généralement à deux objectifs souvent contradictoires :

★ Trouver le meilleur pas possible.

★ Effectuer le moins de calculs possibles.

On distingue deux classes de recherche linéaire

♦ recherche linéaire exacte.

♦ recherche linéaire inexacte où économique.

### 1.4.1 Recherches linéaires exactes

On pose

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

Donc on définie la fonction $\varphi_k$ par :

$$\begin{aligned} \varphi_k \quad &: \quad ]0, +\infty[ \to \mathbb{R} \\ \lambda \quad &\to \quad \varphi_k(\lambda) = J(x_k + \lambda d_k). \end{aligned} \tag{1.4.4}$$

$$\varphi_k(0) = J(x_k). \tag{1.4.5}$$

Avec

$$\varphi_k'(\lambda) = \nabla^T J(x_k + \lambda d_k) d_k. \tag{1.4.6}$$

$$\varphi_k'(0) = \nabla^T J(x_k) d_k < 0. \tag{1.4.7}$$

On cherche le pas de déplacement $\lambda_k$,tel que :

$$\begin{aligned} J(x_k + \lambda d_k) &< J(x_k) \\ \varphi_k(\lambda) &< J(x_k) \\ \varphi_k(\lambda) &< \varphi_k(0). \end{aligned}$$

Alors on cherche une solution du problème suivant :

$$\begin{cases} \min_{\lambda>0} \varphi_k(\lambda) = J(x_k + \lambda d_k) \end{cases} \tag{1.4.8}$$

★Le cas où la fonction $J$ est **différentiable** alors, le pas optimal $\lambda_k$ caractérisé par :

$$\varphi'(\lambda_k) = 0.$$

★Le cas où $J$ est une **fonction quadratique** i.e

$$J(x) = \frac{1}{2} x^T H x + x^T b$$

avec $H \in M^{n*n}$ symétrique et définie positive, par résolution de l'équation (1.4.8) on trouve la soution :

$$\lambda_k = -\frac{\nabla^T J(x_k) d_k}{d_k^T H d_k}.$$

**Remarque 1.4.1** *Ils ne sont utilisés que dans des cas particuliers, par exemple lorsque $\varphi$ est quadratique, la solution de la recherche linéaire s'obtient de façon exacte et dans un nombre fini d'itérations.*

**Définition 1.4.1** ***(Intervalle d'incertitude)***

*Soit* $\varphi : \mathbb{R} \to \mathbb{R}$ , $\min_{\lambda>0}\varphi(\lambda) = \varphi(\lambda^*)$

*L'intervalle* $[a, b] \subset ]0, +\infty[$ *est appelé l'intervalle d'incertitude si* $\lambda^* \in [a, b]$ , *mais la valeur exacte de* $\lambda^*$ *n'est pas connue.*

**Avantage et inconvénient de la recherche linéaire exacte**

L'avantage majeur de cette méthode est sa simplicité.

Pour une fonction non linéaire arbitraire :

♦ La détermination de pas demande en général beaucoup de temps de calcul et ne peut de toutes façons pas être faite avec une précision infinie,

♦ L'efficacité supplémentaire éventuellement apportée à un algorithme par une recherche linéaire précise ne permet pas, en général, de compenser le temps perdu à déterminer un tel pas.

♦ Les résultats de convergence autorisent d'autres types de recherche (recherche linéaire inexacte), moins gourmandes en temps de calcul

### 1.4.2 recherche linéaire inexacte

Dans la plupart des algorithmes d'optimisation modernes, on ne fait jamais de recherche linéaire exacte, car trouver $\lambda_k$ signifie qu'il va falloir calculer un grand nombre de fois la fonction $\varphi$ et cela peut être dissuasif du point de vue du temps de calcul. En pratique, on recherche plutôt une valeur de $\lambda$ qui assure une décroissance suffisante de $J$. Cela conduit à la notion d'intervalle de sécurité.

**Définition 1.4.2** *On dit que* $[a, b]$ *est un intervalle de sécurité s'il permet de classer les valeurs de* $\lambda$ *de la façon suivante :*

♦ *Si* $\lambda < a$ *alors* $\lambda$ *est considéré trop petit.*

♦ *Si* $a \leq \lambda \leq b$ *alors* $\lambda$ *est satisfaisant.*

♦ *Si* $\lambda > b$ *alors* $\lambda$ *est considéré trop grand.*

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

**Schéma des recherches linéaires inexactes**

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

$$\varphi(\lambda_k) < \varphi(0) \left(J(x_k + \lambda_k d_k) < J(x_k)\right). \tag{1.4.9}$$

Le schéma de l'algorithme est donc :

**Algorithme :(Schéma des recherches linéaires inexactes)**

**Etape 0 : (initialisation)**

$a_1 = b_1 = 0$, choisir $\lambda_1 > 0$ ; poser $k = 1$ et aller à l'étape 1.

**Etape 1 :**

Si $\lambda_k$ est satisfaisant (suivant un certain critère ) : STOP ($\lambda = \lambda_k$).

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

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

**Etape 2 :**

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

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

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

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

**La règle d'Armijo**

Soit $J:\mathbb{R}^n\to\mathbb{R}, d\in\mathbb{R}^n$ telles que $\nabla^T J(x_k)d_k<0$ : Définisons la fonction $\varphi:\mathbb{R}\to\mathbb{R}$ telque $\varphi$ vérifier (1.4.4) , (1.4.5) , (1.4.6) et (1.4.7)

Posons

$$\varphi_1(\lambda)=\varphi(0)+\acute{\varphi}(0)\lambda.$$

Donc

$$\varphi_1(\lambda)=J(x_k)+\nabla^T J\left(x_k\right)d_k\lambda.$$

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

$$\hat{\varphi}(\lambda) = J(x_k)+\delta\lambda\nabla^T J\left(x_k\right)d_k; 0<\delta<1 \tag{1.4.10}$$

$$\hat{\varphi}(\lambda) = \varphi(0)+\delta\acute{\varphi}(0)\lambda \tag{1.4.11}$$

On cherche $\hat{\lambda}_k$ tel que

$$\varphi(\hat{\lambda}_k)\leq\hat{\varphi}(\hat{\lambda}_k)$$

c'est à dire

$$\begin{aligned} J\left(x_k+\hat{\lambda}_k d_k\right) &\leq J(x_k)+\delta\hat{\lambda}_k\nabla^T J\left(x_k\right)d_k \\ J(x_k)-J\left(x_k+\hat{\lambda}_k d_k\right) &> \gamma\hat{\lambda}_k \\ J\left(x_k+\hat{\lambda}_k d_k\right) &\leq J(x_k)-\gamma\hat{\lambda}_k \end{aligned}$$

avec $\gamma=-\delta\nabla^T J\left(x_k\right)d_k$

**★Sens graphique de $\lambda_k$-Armijo**

Noton par $T_{M_0}$ le tangent au graphe de $\varphi(\lambda_k)$ qui passe par le point $(0,\varphi_k(0))$

$$\begin{aligned} T &= \{(\lambda,y), y-\varphi_k(0)=\acute{\varphi_k}(0)(\lambda-0)\} \\ &= \{(\lambda,y), y=\acute{\varphi_k}(0)\lambda+\varphi_k(0)\}. \end{aligned}$$

Noton par $\Delta_{M_0}$ la droite qui passe par le point $(0,\varphi_k(0))$ et qui à $\delta\acute{\varphi_k}(0)$ comme coefficient directeur

$$\begin{aligned} \Delta &= \{(\lambda,y), y-\varphi_k(0)=\delta\acute{\varphi_k}(0)(\lambda-0)\} \\ &= \{(\lambda,y), y=\delta\acute{\varphi_k}(0)\lambda+\varphi_k(0)\}. \end{aligned}$$

On à

$$\begin{aligned}\varphi_k'(0) &< 0 \Rightarrow \delta\varphi_k'(0) > \varphi_k'(0)\\ &\Rightarrow \lambda\delta\varphi_k'(0) > \varphi_k'(0)\\ &\Rightarrow \varphi_k(0) + \lambda\delta\varphi_k'(0) > \varphi_k(0) + \varphi_k'(0).\end{aligned}$$

Donc $\lambda \in ]0, +\infty[$ , $\Delta_{M_0}$ est au dessus de $T_{M_0}$

**Algorithme : (Règle d'Armijo)**

**Etape 0 : (initialisation)**

$a_1 = b_1 = 0$, choisir $\lambda_1 > 0$, $\delta \in ]0, 1[$ poser $k = 1$

Calculer $\varphi_k\left(\hat{\lambda}_k\right) = J\left(x_k + \hat{\lambda}_k d_k\right)$et $\varphi_k(0) = J(x_k)$ et aller à l'étape 1 ;

**Etape 1 :**

Si $\varphi_k\left(\hat{\lambda}_k\right) \leq \varphi_k(0) + \delta\varphi_k'(0)\hat{\lambda}_k$ alors

calculer le plus grand entier $n_0$ tel que $\varphi_k\left(2^{n_0}\hat{\lambda}_k\right) \leq \varphi_k(0) + \delta\varphi_k'(0)\, 2^{n_0}\hat{\lambda}_k$

et prendre $\lambda_{Armijo} = 2^{n_0}\hat{\lambda}_k$

Sinon $\varphi\left(\hat{\lambda}_k\right) > \varphi(0) + \delta\varphi'(0)\hat{\lambda}_k$ ,alors

calculer le plus petit entier $n_0$ tel que $\varphi_k\left(\dfrac{\hat{\lambda}_k}{2^{n_0}}\right) \leq \varphi_k(0) + \delta\varphi_k'(0)\dfrac{\hat{\lambda}_k}{2^{n_0}}$

et prendre $\lambda_{Armijo} = \dfrac{\hat{\lambda}_k}{2^{n_0}}$.

**Théorème 1.4.1** . *Si $\varphi_k : \mathbb{R}_+ \to \mathbb{R}$;définie par $\varphi_k(\alpha) = J(x_k + \lambda d_k)$ est continue et bornée inférieurement, si $d_k$ est une direction de descente en $x_k$ $(\varphi_k'(0) < 0)$ et si $\rho \in ]0, 1[$, alors l'ensemble des pas vérifiant la règle d'Armijo est non vide.*

**Preuve.** On a

$$\begin{aligned}\varphi_k(\lambda) &= J(x_k + \lambda d_k)\\ \psi_\rho(\lambda) &= J(x_k) + \rho\lambda_k \nabla J(x_k)^T d_k\end{aligned}$$

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

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

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

$$J(x_k) + \lambda_k \nabla J(x_k)^T d_k < J(x_k) + \rho\lambda_k \nabla J(x_k)^T d_k \text{ pour } \lambda > 0$$

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

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

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

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

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

$$\varphi_k(\hat{\lambda}) = \psi_\rho(\hat{\lambda}) \text{ et } \varphi_k(\lambda) < \psi_\rho(\lambda) \text{ pour } 0 < \lambda < \hat{\lambda}$$

Ce qui achève la démonstration. ■

**La règle de Goldstein et Price**

Dans la règle d'*Armijo* on assure la décroissance de la fonction objectif à chaque pas, mais ce n'est pas suffisant.

Les conditions de *Goldstein* et *Price* suivant 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*.

$$J(x_k + \lambda_k d_k) \leq J(x_k) + \delta\lambda_k \nabla^T J(x_k) d_k \tag{1.4.12}$$

$$\varphi_k(\lambda_k) \leq \varphi_k(0) + \delta\lambda_k \varphi_k'(0) \tag{1.4.13}$$

$$J(x_k + \lambda_k d_k) \geq J(x_k) + (1-\delta)\lambda_k \nabla^T J(x_k) d_k \tag{1.4.14}$$

$$\varphi_k(\lambda_k) \geq \varphi_k(0) + (1-\delta)\lambda_k \varphi_k'(0) \tag{1.4.15}$$

Où $\delta$ est un constante vérifiant $0 < \delta < \frac{1}{2}$ et $\frac{1}{2} < 1 - \delta < 1$, cette inégalité qui empêche le pas de ne pas être trop petit.

**★Sens graphique de $\lambda_k$-Goldstein**

♦$\varphi_k(\lambda_{Gold}) \leq \varphi_k(0) + \delta\lambda_{Gold}\varphi_k'(0)$

♦$\varphi_k(\lambda_{Gold}) \geq \varphi_k(0) + (1-\delta)\lambda_{Gold}\varphi_k'(0)$,

telque $0 < \delta < \frac{1}{2}$ et $\frac{1}{2} < 1 - \delta < 1$ donc $1 - \delta > \delta$

♦$\varphi_k'(0) < 0$.

Donc

$$\begin{aligned} \varphi_k'(0)(1-\delta) &< \varphi_k'(0)\delta \\ &\Rightarrow \varphi_k(0) + \varphi_k'(0)(1-\delta) < \varphi_k(0) + \varphi_k'(0)\delta \end{aligned}$$

Alors la droite $\Delta_2$ situé au dessus de la droite $\Delta_1$Avec

$$\Delta_1 = \{(\lambda, y), y = \varphi_k(0) + \varphi_k'(0)\delta\}$$

$$\Delta_2 = \{(\lambda, y), y = \varphi_k(0) + \varphi_k'(0)(1-\delta)\}$$

Donc $\lambda_{Gold} \in ]\beta_1, \beta_2[$,

telque $\beta_1$ est l'abscisse du point d'intersection de graphe $C_{\varphi_k}$ et la droite $\Delta_2$

et $\beta_2$ est l'abscisse du point d'intersection de graphe $C_{\varphi_k}$ et la droite $\Delta_1$

**★Règle de Goldstein**

♦ Si $\varphi_k(\lambda_k) \leq \varphi_k(0) + \delta\lambda_k\varphi_k'(0)$, alors $\lambda_k$ est trop petit

♦ Si $\varphi_k(\lambda_k) \geq \varphi_k(0) + (1-\delta)\lambda_k\varphi_k'(0)$ , alors $\lambda_k$ est trop grand.

♦ Si $\varphi_k(0) + \delta\lambda_k\varphi_k'(0) \geq \varphi_k(\lambda_k) \geq \varphi_k(0) + (1-\delta)\lambda_k\varphi_k'(0)$, alors convient.

**Algorithme :(Règle de Goldstein et Price)**

**Etape 0 : (initialisation)**

$a_1 = b_1 = 0$, choisir $\lambda_1 > 0$ $\delta \in \left]0, \frac{1}{2}\right[$, $(1-\delta) \in \left]\frac{1}{2}, 1\right[$, poser $k = 1$ et aller à l'étape 1 ;

**Etape 1 :**

Si $\varphi_k(0) + \delta\lambda_k\varphi_k'(0) \geq \varphi_k(\lambda_k) \geq \varphi_k(0) + (1-\delta)\lambda_k\varphi_k'(0)$ : STOP $(\hat{\lambda} = \lambda_k)$.

Si $\varphi_k(\lambda_k) \geq \varphi_k(0) + (1-\delta)\lambda_k\varphi_k'(0)$, alors $b_{k+1} = \lambda_k, a_{k+1} = a_k$,

et aller à l'étape 2.

Si $\varphi_k(\lambda_k) \leq \varphi_k(0) + \delta\lambda_k\varphi_k'(0)$, alors $b_{k+1} = b_k, a_{k+1} = \lambda_k$ ;

et aller à l'étape 2.

**Etape 2 :**

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

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

**Théorème 1.4.2** *Si* $\varphi : \mathbb{R}_+ \longrightarrow \mathbb{R}$ *définie par* $\varphi(\lambda_k) = J(x_k + \lambda d_k)$ *est continue et bornée inférieurement, si* $d_k$ *est une direction de descente en* $x_k$ *et si* $\rho \in ]0, 1[, \delta \in ]\rho, 1[$, *alors l'ensemble des pas vérifiant la règle de Goldstein et Price est non vide.*

**Preuve.** On a

$$\begin{aligned} \varphi_k(\lambda) &= J(x_k + \lambda d_k) \\ \psi_\rho(\lambda) &= J(x_k) + \rho\lambda\nabla J(x_k)^T d_k \\ \psi_\delta(\lambda) &= J(x_k) + \delta\lambda\nabla J(x_k)^T d_k \end{aligned}$$

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

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

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

$$J(x_k) + \lambda\nabla J(x_k)^T d_k < J(x_k) + \delta\lambda\nabla J(x_k)^T d_k < J(x_k) + \rho\lambda\nabla J(x_k)^T d_k \text{ pour } \alpha > 0$$

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

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

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

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

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

$$\varphi_k(\hat{\lambda}) = \psi_\rho(\hat{\lambda}) \text{ et } \varphi_k(\lambda) < \psi_\rho(\lambda) \text{ pour } 0 < \lambda < \hat{\lambda}$$

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

$$\varphi_k(\tilde{\lambda}) = \psi_\delta(\tilde{\lambda}) \text{ et } h_k(\lambda) < \psi_\delta(\lambda) \text{ pour } 0 < \lambda < \tilde{\lambda}$$

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

$$\left( \begin{array}{c} \psi_\delta(\tilde{\lambda}) = \ \varphi_k(\tilde{\lambda}) < \psi_\rho(\alpha) \text{ n'est autre que} \\ J(x_k) + \delta\tilde{\lambda}\nabla J(x_k)^T d_k = J(x_k + \tilde{\lambda} d_k) < J(x_k) + \rho\tilde{\lambda}\nabla J(x_k)^T d_k \end{array} \right)$$

Ce qu'il fallait démontrer. ■

**La règle de Wolfe**

Une condition d' recherche linéaire inexacte éxige que $\lambda_k$ soit calculer de façon à diminuer suffisamment la fonction objectif $J$, donnée par :

$$J\left(x_k + \lambda_k d_k\right) \leq J(x_k) + \delta\lambda_k \nabla^T J\left(x_k\right) d_k$$

avec $\delta \in ]0, 1[$ .

En plus de la condition d'Armijo, il est important d'imposer une condition qui permet d'éviter les pas trop petits.

La règle de *Wolfe* fait appel au calcul de $\varphi'(\lambda)$, Cependant dans de nombreuses applications, le calcul du gradient $\nabla J(x)$ représente un faible coût additionnel en comparaison du coût d'évaluation de $J(x)$, c'est pourquoi cette règle est très utilisée.

Si $\hat{\lambda}$ le pas optimal à faire en $x$ dans la direction $d$, on a :

$$\nabla^T J(x_k) d_k < 0 \text{ et } \nabla^T J(x_k + \hat{\lambda} d_k) d_k = 0$$

La dérivée $\nabla^T J(x_k + \lambda_k d_k) d_k$ augmente donc par rapport à sa valeur négative initiale.

Afin d'assurer des pas suffisamment grand, l'idée va être de demander à la dérivée directionnelle d'augmenter suffisamment.

Ces deux conditions sont :

$$J\left(x_k+\lambda_k d_k\right) \leq J(x_k)+\delta\lambda_k \nabla^T J\left(x_k\right) d_k \qquad \delta \in ]0,1[\,,$$

$$\nabla^T J(x_k+\lambda_k d_k).d_k \geq \sigma \nabla^T J(x_k).d_k \qquad \sigma \in ]\delta,1[\,,$$

avec $\delta$ et $\sigma$ deux réels tel que $0<\delta<\sigma<1$,

**Remarque 1.4.2** - *La deuxième condition de Wolfe est appelé condition de courbure.*
*- Les $\lambda_k$ sélectionnés par la $1^{er}$ condition de Wolfe peuvent être trés petits. ceci peut avoir des conséquences fâcheuses sur la convergence de l'algorithme. La $2^{eme}$ condition de Wolfe évite cet inconvénient et supprime les trés petites valeurs de $\lambda_k$.*

**La règle de Wolfe faible**

$$J(x_k+\lambda d_k) \leq J(x_k)+\delta\lambda \nabla^T J(x_k).d_k, \tag{1.4.16}$$

Avec $\delta \in \left]0,\frac{1}{2}\right[$, $(1-\delta) \in \left]\frac{1}{2},1\right[$.Pour expliquer ces conditions, posons :

$$\Psi\left(\lambda\right)=\varphi\left(0\right)+\left(\delta\varphi'(0)\right)\lambda. \tag{1.4.17}$$

La fonction $\Psi\left(\lambda\right)$ est une fonction linéaire à pente négative $\delta\nabla^T J\left(x_k\right) d_k$ mais comme $\delta \in ]0,1[$, $\Psi\left(\lambda\right)$ se trouve au-dessus du graphe de $\varphi$ pour les petites valeurs positives de $\lambda$.
La condition suffisante de diminution indique que $\lambda$ est acceptable seulement si $\varphi(\lambda) \leq \Psi\left(\lambda\right)$
En pratique,$\delta$ est choisi pour être assez faible, par exemple $\delta=10^{-4}$.La condition suffisante de diminution ne suffit pas pour assurer que l'algorithme fait des progrès raisonnable, ses progrès sont assurés pour toutes les valeurs suffisamment petites de $\lambda$. Pour exclure des mesures trop petites, nous introduisons une deuxième condition, la condition de courbure, qui est necessaire pour $\lambda_k$ satisfaite

$$\nabla^T J(x_k+\lambda d_k).d_k \geq \sigma \nabla^T J(x_k).d_k. \tag{1.4.18}$$

pour certains constants $\sigma \in ]0,1[$,et $\sigma>\delta$, donc la condition de la courbure assure que la pente de la fonction $\varphi$ en $\lambda_k$ est supérieure à $\sigma$ dans la pente initiale $\varphi'(0)$ .C'est logique car

si la pente $\varphi'(\lambda)$ est fortement négatif, nous avons une indication que nous pouvons réduire $J$ significativement en déplacant le long de la direction choisie.

La diminution suffisante combinée à la condition de la courbure sont connues comme ; les conditions de Wolfe faible données par :

$$J\left(x_k+\lambda_k d_k\right) \leq J(x_k)+\delta\lambda_k \nabla^T J\left(x_k\right) d_k \tag{1.4.19}$$

$$\nabla^T J(x_k+\lambda_k d_k).d_k \geq \sigma \nabla^T J(x_k).d_k. \tag{1.4.20}$$

avec $\sigma \in ]0,1[$ ,et $\sigma > \delta$.

Le pas de déplacement peut satisfaire les conditions de Wolfe sans être particulièrement proche du minimumm de $\varphi_k$.Cependant, nous pouvons modifier la condition de courbure pour forcer $\lambda_k$ à se trouver au moins au voisinage du minimumm local ou le point fixe de $\varphi_k$.

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

$$J(x_k+\lambda d_k) \leq J(x_k)+\delta\lambda \nabla^T J(x_k).d_k, \tag{1.4.21}$$

$$\sigma_1 \nabla^T J\left(x_k\right).d_k \leq \nabla^T J(x_k+\lambda_k d_k).d_k \leq -\sigma_2 \nabla^T J\left(x_k\right).d_k. \tag{1.4.22}$$

Où $0 < \sigma_1 < \sigma_2 < 1$.

**La règle de Wolfe forte (1971)** Pour certains algorithmes il est parfois nécessaire d'avoir une condition plus restricive que (1.4.20)

Pour cela la deuxième condition (1.4.20) est remplacée par :

$$\begin{aligned} \left|\nabla^T J(x_k+\lambda d_k).d_k\right| &\leq \sigma\left|\nabla^T J(x_k).d_k\right| \\ &\leq -\sigma \nabla^T J(x_k).d_k \end{aligned} \tag{1.4.23}$$

avec $\sigma \in ]0,1[$ ,et $\sigma > \delta$.La seule différence avec les conditions de Wolfe c'est de permettre à la dérivée $\varphi'(\lambda_k)$ d'être trop positive. Par conséquent, nous excluons les points qui sont loin d'être les points fixes de $\varphi$.

On aura donc les conditions de Wolfe fortes :

$$J\left(x_k+\lambda_k d_k\right) \leq J(x_k)+\delta\lambda_k \nabla^T J\left(x_k\right) d_k \tag{1.4.24}$$

$$\left|\nabla^T J(x_k+\lambda d_k).d_k\right| \leq -\sigma\nabla^T J(x_k).d_k \tag{1.4.25}$$

**Algorithme : (Règle de Wolfe)**

**Etape 0 : (initialisation)**

$a_1 = b_1 = 0$, choisir $\lambda_1 > 0$ $\sigma \in ]0,1[$ ,et $\sigma > \delta$, poser $k = 1$ et aller à l'étape 1

**Etape 1 :**

Si $\varphi_k(\lambda_k) \leq \varphi_k(0) + \delta\lambda_k\acute{\varphi_k}(0)$ et $\varphi_k'(\lambda_k) \geq \sigma\varphi_k(0)$ : STOP ($\hat{\lambda} = \lambda_k = \lambda_{wolf}$).

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

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

**Etape 2 :**

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

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

La contraintes $(E2)$ entraîne que $\sigma\varphi(0) \leq \varphi'(\lambda) < -\sigma\varphi(0)$ c'est a dire. $\varphi'(\lambda)$ n'est pas "trop"positif.

**Remarque 1.4.3** - *On voit bien que les conditions de wolfe fortes impliquent les conditions de Wolfe faibles. Effectivement* (1.4.24) *est équivalente à* (**??**) *tandis que* (1.4.25)

$$\begin{aligned}\left|\nabla^T J(x_k+\lambda d_k).d_k\right| &\leq -\sigma\nabla^T J(x_k).d_k \\ &\Leftrightarrow \sigma\nabla^T J(x_k).d_k \leq \nabla^T J(x_k+\lambda d_k).d_k \leq -\sigma\nabla^T J(x_k).d_k \\ &\Rightarrow \sigma\nabla^T J(x_k).d_k \leq \nabla^T J(x_k+\lambda d_k).d_k \end{aligned} \tag{1.4.26}$$

*Le pas $\lambda_k$ sélectionné par les conditions* (**??**) *et* (1.4.26) *peut être très loin d'un point optimal ou stationnaire de la fonction $\varphi$. La condition* (1.4.25) *assure que le pas $\lambda_k$ se trouve dans le voisinage d'un point stationnaire ou un point optimale de $\varphi$.*

**Remarque 1.4.4** - *On voit bien que les conditions de* ***Wolfe relaxée*** *impliquent les conditions de* ***Wolfe fortes****. Effectivement* (1.4.21) *est équivalente à* (1.4.24)*, tandis que pour le cas*

*particulier* $\sigma_1 = \sigma_2 = \sigma$, (1.4.22) *est équivalente à* (1.4.25). *En effet :*

$$\begin{aligned} \sigma_1 \nabla^T J(x_k).d_k \quad &\leq \quad \nabla^T J(x_k + \lambda_k d_k).d_k \leq -\sigma_2 \nabla^T J(x_k).d_k \\ &\Longrightarrow \quad \sigma \nabla^T J(x_k).d_k \leq \nabla^T J(x_k + \lambda d_k).d_k \leq -\sigma \nabla^T J(x_k).d_k \\ &\Longrightarrow \quad \left|\nabla^T J(x_k + \lambda d_k).d_k\right| \leq -\sigma \nabla^T J(x).d_k \qquad (E2) \end{aligned}$$

*Si* $\varphi_k : \mathbb{R}_+ \to \mathbb{R}$;*définie par* $\varphi(\lambda_k) = J(x_k + \lambda d_k)$ *est dérivable et bornée inférieurement, si* $d_k$ *est une direction de descente en* $x_k$ *et si* $\rho \in ]0, 1[\,, \sigma \in ]\rho, 1[\,$, *alors l'ensemble des pas vérifiant la règle de Wolfe (faible) est non vide.*

**Preuve.** On a

$$\begin{aligned} \varphi(\lambda_k) &= J(x_k + \lambda d_k) \\ \psi_\rho(\lambda) &= J(x_k) + \rho\lambda \nabla J(x_k)^T d_k \end{aligned}$$

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

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

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

$$J(x_k) + \lambda \nabla J(x_k)^T d_k < J(x_k) + \rho\lambda \nabla J(x_k)^T d_k \text{ pour } \alpha > 0$$

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

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

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

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

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

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

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

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

$$\begin{aligned} \varphi_k(\tilde{\lambda}) - \varphi_k(0) &= \tilde{\lambda}\varphi_k'(\hat{\lambda}) = \tilde{\lambda}\nabla J(x_k + \hat{\lambda}d_k)^T d_k \\ &\Rightarrow \rho\tilde{\lambda}\nabla J(x_k)^T d_k = \tilde{\lambda}\nabla J(x_k + \hat{\lambda}d_k)^T d_k \\ &\Rightarrow \nabla J(x_k + \hat{\lambda}d_k)^T d_k = \rho\nabla J(x_k)^T d_k \geq \sigma\nabla J(x_k)^T d_k \end{aligned}$$

car $0 < \rho < \sigma < 1$ et $\nabla J(x_k)^T d_k < 0$ .

Donc $\hat{\lambda}$ satisfait (1.4.26)

D'autre part, $\lambda = \hat{\lambda}$ satisfait (**??**), en effet,

$\hat{\lambda}$ satisfait (*) $h_k(\hat{\lambda}) < \psi_\rho(\hat{\lambda})$ n'est autre que :

$$J(x_k + \hat{\lambda}d_k) < J(x_k) + \rho\hat{\lambda}\nabla J(x_k)^T d_k$$

■

# CHAPITRE 2

# Les méthodes à directions de descente

Ce chapitre aborde une catégorie importante d'algorithmes destinés à résoudre des problèmes d'optimisation sans contraintes. Considérons le problème de minimiser la fonction $J(x)$ où $x$ appartient à $\mathbb{R}^n$. Un algorithme itératif est une méthode qui permet de résoudre ce problème en générant une séquence de vecteurs $(x_0, x_1, \ldots, x_n)$ dans $\mathbb{R}^n$, en suivant un ensemble d'instructions et une condition d'arrêt. En d'autres termes, à partir d'un point de départ $x_0$ , on construit une suite $(x_n)$ qui converge vers la solution du problème.

$$(p) : \{\min J(x), x \in \mathbb{R}^n\}. \tag{2.0.1}$$

Un algorithme est défini comme une application $K$ de $\mathbb{R}^n$ vers $\mathbb{R}^n$, qui associe à chaque point $x_k$ le point $x_{k+1} = K(x_k)$. L'étude de la convergence est similaire à l'analyse des propriétés de l'application $K$.

Il est essentiel de comprendre que la plupart des algorithmes d'optimisation, qu'ils intègrent des contraintes ou non, suivent une approche générale bien définie ; à chaque étape, ils s'efforcent de se rapprocher du minimum en résolvant un sous-problème de minimisation.

# 2.1 Convergence des méthodes à direction de descente et recherche linéaire

## 2.1.1 Mode de convergence

**Définition 2.1.1** ***(Convergence globale)***

*On dit qu'un algorithme décrit par une application $K$ est globalement convergent (où encore, possède la propriété de la convergence globale) si, quelque soit le point de départ $x_0$ choisi, la suite $\{x_k\}_{k\in\mathbb{N}}$ générée par cet algorithme (où une sous suite) converge vers un point $\hat{x}$ satisfaisant les conditions nécessaires d'optimalité.($\nabla J(\hat{x}) = 0$ et $H(\hat{x})$ est semi définie positive).*

**Remarque 2.1.1** *On a plusieurs modes de convergence :*

*1- La suite $(x_k)$ converge vers $\hat{x}$ linéairement aves un taux $\alpha < 1$ si*

$$\lim_{k\to\infty} \sup \frac{||x_{k+1} - \hat{x}||}{||x_k - \hat{x}||} = \alpha.$$

*2- La convergence est dite asymptotique si :*

$$||x_{k+1} - \hat{x}|| \tilde{=} \alpha ||x_k - \hat{x}||.$$

*3- La convergence est dite super-linéaire si :*

$$\lim_{k\to\infty} \frac{||x_{k+1} - \hat{x}||}{||x_k - \hat{x}||} = 0.$$

*4- La convergence est dite super-linéaire d'ordre $\gamma > 1$ si :*

$$\lim_{k\to\infty} \frac{||x_{k+1} - \hat{x}||}{||x_k - \hat{x}||} < \infty.$$

*Si $\gamma = 2$ la convergence de l'algorithme est dite quadratique.*

## 2.1.2 Hypothèses H1 et H2

Il est important de rappeler les deux conditions que toutes les méthodes du gradient conjugué doivent respecter (ou au moins l'une d'elles) pour prouver les résultats de convergence.

**Condition H1**

Soit $J : \mathbb{R}^n \to \mathbb{R}$. On dit que $J$ vérifie la condition $C1$ si $J$ est continuement différentiable dans un voisinage $V(\Gamma)$ de $\Gamma = \{x \in \mathbb{R}^n : J(x) \leq J(x_0) : x_0 \in \mathbb{R}^n \text{ point initial}\}$ et si $\nabla J(x)$ vérifie la condition de Lipschitz dans $V(\Gamma)$, c'est à dire, il existe une constante $L$ telle que

$$||\nabla J(x) - \nabla J(y)|| \leq L||x - y|| : \text{ pour tout } x, y \in V(\Gamma). \tag{2.1.1}$$

**Condition C2**

L'ensemble $\Gamma = \{x \in \mathbb{R}^n : J(x) \leq J(x_1) : x_0 \in \mathbb{R}^n \text{ point initial}\}$ est borné, i.e, il existe une constante $B < \infty$ telle que

$$||x|| \leq B : \forall x \in \Gamma. \tag{2.1.2}$$

### 2.1.3 Convergence globale des algorithmes

Pour démontrer la convergence globale des algorithmes de descente avec recherche linéaire, Il s'agit de montrer que :

$$\lim_{k \to +\infty} ||\nabla J(x_k)|| = 0.$$

Ce résultat signifie que tout point d'accumulation $\hat{x}$ de la suite $(x_k)_{k \in \mathbb{N}}$ est un point critique de $J$ .

**Remarque 2.1.2** *La convergence globale des méthodes de descente avec analyse linéaire ne garantit que la convegence de sous-suites des itérés vers un point critique, alors que la convergence de la suite elle même n'est pas garantie.*

**Principe de démonstration**

Pour montrer la convergence des algorithmes de descente, il suffit de montrer que l'algorithme vérifie une inégalité du type :

$$J(x_k) - J(x_{k+1}) \geq c||\nabla J(x_k)||^2,$$

ou $c > 0$ est une constante réelle. En sommant ces inégalités pour $k$ variant de 0 à $N-1$, on obtient :

$$\forall N \in \mathbb{N}, J(x_0) - J(x_N) \geq c \sum_{k=0}^{N-1} ||\nabla J(x_k)||^2.$$

Avec $\sum_{k=0}^{N-1} ||\nabla J(x_k)||^2$ est une somme partielle d'une série à termes positifs.
Si $J$ est borné inférieurement, alors nécessairement $J(x_0) - J(x_N)$ est majorée et donc la somme partielle est majorée, et donc la série $\sum_{k=0}^{N-1} ||\nabla J(x_k)||^2$ converge, ce qui implique :

$$\lim_{k \to +\infty} ||\nabla J(x_k)|| = 0.$$

### 2.1.4 Condition de Zoutendijk.

Nous allons explorer comment la recherche linéaire inexacte contribue à la convergence des algorithmes de descente. Il s'agit d'une contribution importante, car la recherche linéaire à elle seule ne peut pas garantir la convergence des itérations. Il est clair que le choix de la direction de descente est crucial. Cela correspond à une condition, souvent désignée sous le nom de condition de Zoutendijk, qui permet d'extraire des informations qualitatives intéressantes.
On dit qu'une régle d' recherche linéaire satisfait la condition de Zoutendijk ; s'il existe une constante $c > 0$ telle que pour tout indice $k \geq 1$ on a :

$$J(x_{k+1}) \leq J(x_k) - c||\nabla J(x_k)||^2 \cos^2 \theta_k. \tag{2.1.3}$$

où $\theta_k$ est l'angle que fait la direction de descente $d_k$ avec la direction du gradient $-\nabla J(x_k)$, définie par :

$$\langle -\nabla J(x_k), d_k \rangle = -\nabla^T J(x_k).d_k = ||\nabla J(x_k)||.||d_k||. \cos \theta_k.$$

Donc

$$\cos \theta_k = \frac{-\nabla^T J(x_k).d_k}{\|\nabla J(x_k)\| . \|d_k\|}. \tag{2.1.4}$$

**Théorème 2.1.1** *Si la suite $\{x_k\}$ générée par un algorithme d'optimisation vérifie la condition de Zoutendijk (2.1.2) et si la suite $\{J(x_k)\}$ est minorée, alors*

$$\sum_{k \geq 1} \|\nabla J(x_k)\|^2 \cos^2 \theta_k < \infty. \tag{3-6}$$

**Preuve.** En sommant les inégalités (2.1.3), on a

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

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

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

■

Ce théorème implique que si la direction de descente $d_k$ n'est pas orthogonale à la direction de $-\nabla J(x_k)$ à l'infini i.e. si :

$$\exists c > 0, \forall k \in \mathbb{N}, \cos(\theta_k) \geq c,$$

alors la série $\sum_{k\geq 1} \|\nabla J(x_k)\|^2$ converge, et on obtient ainsi la convergence globale de l'algorithme de descente considéré.

**Utilisation du théorème de Zoutendjik pour démontrer la convergence globale**

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

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

pour tout $k$, alors on a

$$\lim_{k\to\infty} ||R_k|| = 0. \tag{2.1.5}$$

Dans les différentes méthodes du gradient conjugué on n'arrive pas à démontrer le résultat précédent i.e, $\lim_{k\to\infty} ||R_k|| = 0$, mais seulement

$$\liminf_{k\to\infty} ||R_k|| = 0. \tag{2.1.6}$$

La condition (2.1.6) implique qu'il existe une sous-suite de $\{||R_k||\}$ qui tend vers zéro.

**Conditions suffisantes associés au théorème de Zoutendijk pour démontrer (2.1.6)**

Pour démontrer (2.1.6) on associe au théorème de Zoutendijk les 2 hypothèses suivantes

**Hypothèse 1**

La decente suffisante est assuré, i.e, il existe une constante $c$ indépendante de $k$ telle que

$$R_k^T d_k \leq -c||R_k||^2. \tag{2.1.7}$$

**Hypothèse 2**

Il existe une constante $\beta$ tel que

$$||d_k||^2 \leq \beta k. \tag{2.1.8}$$

On a donc

$$\left\{ \begin{array}{c} \text{relation de Zoutendijk } \sum_{k=0}^{\infty} \frac{(R_k^T d_k)^2}{||d_k||^2} < \infty \\ \text{Hypothèse 1} \\ \text{Hypothèse 2} \end{array} \right\} \Rightarrow \left\{ \liminf_{k\to\infty} ||R_k|| = 0 \right\}.$$

**Remarque 2.1.3** *La condition* ***Hypothèse 1*** *est plus forte que la descente. En effet*

$$R_k^T d_k \leq -c||R_k||^2 \Rightarrow R_k^T d_k < 0.$$

*La condition* ***Hypothèse 1*** *n'est pas nécessaire pour avoir* $\liminf_{k\to\infty} |R_k|| = 0$. *Dai et Yuan ont montré la convergence globale seulement avec la condition de descente* $R_k^T d_k < 0$.

**Définition 2.1.2** *Nous dirons qu'une méthode du gradient conjuqué converge globalement si l'une des deux conditions suivantes est vérifiée :*

***a)*** $R_{k_0} = 0$, *pour un certain* $k_0 \in \mathbb{N}$.

***b)*** $\liminf_{k\to\infty} ||R_k|| = 0$.

## 2.2 Méthode du gradient (méthode de la plus forte pente)

### 2.2.1 Principe de la méthode

L'algorithme du gradient est un algorithme d'optimisation qui repose sur des principes différentiables. Il fonctionne de manière itérative, ce qui signifie qu'il s'améliore progressivement. À chaque étape, il effectue un mouvement dans la direction opposée au gradient pour minimiser la fonction. Cela le classe parmi les algorithmes de descente directionnelle. L'algorithme du gradient, également connu sous le nom de méthode de la pente la plus forte ou descente la plus

raide, tire son nom du fait que le gradient représente la pente de la fonction linéarisée au point d'intérêt, ce qui correspond localement à sa pente maximale, un concept basé sur le produit scalaire.

Voici l'approche générale d'une technique de descente :

$$\begin{aligned} x_0 &\in \mathbb{R}^n \text{ donné} \\ x_{k+1} &= x_k + \lambda_k d_k \ , d_k \in \mathbb{R}^n - \{0\} \text{ et } \lambda_k \in \mathbb{R}^{+*}. \end{aligned}$$

Où $\lambda_k$ est choisi de manière à ce que

$$J\left(x_k + \lambda_k d_k\right) \leq J\left(x_k\right), \quad \forall \lambda > 0.$$

On trouve la direction de descente par un développement de Taylor d'ordre 2 de la fonction $J$ entre deux itération

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

$$J(x_k + \lambda_k d_k) = J(x_k) + \lambda_k \nabla^T J(x_k) d_k + \varepsilon(\lambda_k d_k).$$

Comme on veut $J\left(x_k + \lambda_k d_k\right) \leq J\left(x_k\right)$ on peut choisir une première approximation

$d_k = -\nabla J(x_k)$ La méthode ainsi obtenue s'appelle l'algorithme du Gradient.

Le schéma général généré par cette méthode est donnée par :

$$\begin{aligned} x_0 &\in \mathbb{R}^n \text{ donné} \\ x_{k+1} &= x_k - \lambda_k \nabla J(x_k), \ \lambda_k \in \mathbb{R}^{+*}. \end{aligned}$$

Avec le pas $\lambda_k$ est peut être :

1- pas optimal $\lambda_k = \hat{\lambda}$

2- pas d'Armijo

3- pas de Goldstien

4- pas de Wolfe

5- pas constant

### 2.2.2 Convergence de la méthode du gradient

**Théorème 2.2.1** *Soit $J$ une fonction $C^1(\mathbb{R}^n, \mathbb{R})$, coercive et strictement convexe. On suppose qu'il existe une constante $M > 0$ tel que*

$$\forall (x, y) \in \mathbb{R}^n * \mathbb{R}^n, ||\nabla J(x) - \nabla J(y)|| \leq M||x - y||.$$

*Alors, si on choisit le pas $\lambda_k$ dans un intervalle $[\beta_1, \beta_2]$ tel que $0 < \beta_1 < \beta_2 < \frac{2}{M}$, la méthode du gradient converge vers le minimum de $J$.*

**Preuve.** La fonction $J$ admet un minimum $\hat{x}$ unique sur $\mathbb{R}^n$ caracérisé par
$\nabla J(\hat{x}) = 0$ puisque $J$ est strictement convexe. Montrons que la suite $(x_k)$ engendrée par l'algorithme converge vers $\hat{x}$, on a :

$$J(y) = J(x) + \langle \nabla J(x), y - x \rangle + \int_0^1 \langle \nabla J(x_k + t(x_{k+1} - x_k)) - \nabla J(x_k), x_{k+1} - x_k \rangle dt.$$

Comme $x_{k+1} = x_k - \lambda_k \nabla J(x_k)$, on obtient

$$\begin{aligned} J(x_{k+1}) - J(x_k) &\leq -\frac{1}{\lambda_k}||x_{k+1} - x_k||^2 + \int_0^1 ||\nabla J(x_k + t(x_{k+1} - x_k)) - \nabla J(x_k)||.||x_{k+1} - x_k||dt \\ &\leq -\frac{1}{\lambda_k}||x_{k+1} - x_k||^2 + \frac{M}{2}||x_{k+1} - x_k||^2 \\ &\leq \left(\frac{M}{2} - \frac{1}{\lambda_k}\right)||x_{k+1} - x_k||^2. \end{aligned}$$

Si on choisit $\lambda_k$ dans un intervalle $[\beta_1, \beta_2]$ tel que $0 < \beta_1 < \beta_2 < \frac{2}{M}$, nous obtenons alors :

$$J(x_{k+1}) - J(x_k) \leq \left(\frac{M}{2} - \frac{1}{\lambda_k}\right)||x_{k+1} - x_k||^2.$$

La suite $J(x_k)$ est alors strictement décroissante, comme elle est minorée car :

$$J(x_k) \leq J(\hat{x}), \forall k$$

elle converge.

Cela entraine d'une part que $(J(x_{k+1}) - J(x_k))$ tends vers 0 et d'autre part, la suite $(x_k)$ est bornée (car $J$ est coercive). O peut donc exraire une sous suite convergeant vers $\hat{x}$. De plus, comme :

$$||x_{k+1} - x_k||^2 \leq \left(\frac{1}{\lambda_k} - \frac{M}{2}\right)^{-1} .(J(x_{k+1}) - J(x_k)).$$

La suite $(x_{k+1} - x_k)$ tend également vers 0.

Par conséquent :

$$\nabla J(x_k) = \frac{x_{k+1} - x_k}{\lambda_k} \to 0.$$

Par continuité de $\nabla J(\hat{x}) = 0$, donc $\hat{x}$ est l'unique minimum $\hat{x}$ de $J$. Ceci étant vrai pour toute valeur d'adhérence de la suite $(x_k)$ cela prouve que toute la suite $(x_k)$ converge vers $\hat{x}$. ■

**Définition 2.2.1** ***(Lignes de niveau de direction du gradient)***

*Soit $J : \mathbb{R}^n \longrightarrow \mathbb{R}$ et $k \in \mathbb{R}$ on appel ligne de niveau d'ordre $k$ associe à $J$ et on le note $C_k$ l'ensemble des point de $\mathbb{R}^n$ suivant :*

$$C_k = \{x \in \mathbb{R}^n; J(x) = k\}. \tag{2.2.1}$$

**Théorème 2.2.2** *soit $J : \mathbb{R}^n \longrightarrow \mathbb{R}$ différentiable et $(x_0, y_0) \in \mathbb{R}^2$, considérons la ligne de niveau :*

$$C_{J(x_0,y_0)} = \{x \in \mathbb{R}^n; J(x, y) = J(x_0, y_0)\}.$$

*Soit $T_{(x_0,y_0)}$ la tangente à la courbe $C_{J(x_0,y_0)}$ et passant par le point $(x_0, y_0)$. alors $T_{(x_0,y_0)}$ est orthogonale à la direction du gradient.*

**Preuve.**

$$\begin{aligned}
T_{(x_0,y_0)} &= \left\{(x, y) \in \mathbb{R}^2, \frac{\delta J}{\delta x}(x_0, y_0)(x - x_0) + \frac{\delta J}{\delta y}(x_0, y_0)(y - y_0) = 0\right\} \\
T_{(x_0,y_0)} &= \left\{(x, y) \in \mathbb{R}^2, \frac{\delta J}{\delta x}(x_0, y_0)x - \frac{\delta J}{\delta x}(x_0, y_0)x_0 + \frac{\delta J}{\delta y}(x_0, y_0)y - \frac{\delta J}{\delta y}(x_0, y_0)y_0 = 0\right\} \\
T_{(x_0,y_0)} &= \left\{(x, y) \in \mathbb{R}^2, \frac{\delta J}{\delta x}(x_0, y_0)x + \frac{\delta J}{\delta y}(x_0, y_0)y - \frac{\delta J}{\delta x}(x_0, y_0)x_0 - \frac{\delta J}{\delta y}(x_0, y_0)y_0 = 0\right\}
\end{aligned}$$

C'est l'équation de la droite alors on conclut :

le vecteur unitaire $\vec{n} = \left(\frac{\delta J}{\delta x}(x_0, y_0), \frac{\delta J}{\delta y}(x_0, y_0)\right)^T = \nabla J(x_0, y_0)^T$

le vecteur directeur $\vec{d} = \left(-\frac{\delta J}{\delta y}(x_0, y_0), \frac{\delta J}{\delta x}(x_0, y_0)\right)$. ■

**Proposition 2.2.1** *Dans la méthode du gradient à pas optimal on à :*

$$\nabla^T J(x_{k+1}) \nabla J(x_k) = 0. \tag{2.2.2}$$

### 2.2.3 Avantages et inconvénients de la méthode du Gradient

**Avantages**

- Au démarrage, la méthode du gradient descent de façon appréciable.

- Le calcul est assez simple.

- On a de bon résultat de convergence.

**Inconvénients**

- Lorsque $x_k$ s'approche de $\hat{x}$ tel que $\nabla J(\hat{x}) = 0$ (i.e $\nabla J(x_k) \tilde{=} 0$) alors $J(x_k)$ décroit très lentement.

## 2.3 Méthode du Gradient Conjugué

La technique des directions conjuguées, en particulier à travers les méthodes du gradient conjugué (GC), peut être considérée comme un compromis entre la méthode de la pente maximale et la méthode de Newton. Cette approche a pour but d'accélérer la convergence de la méthode du gradient le plus raide, qui est généralement lente, tout en surmontant les difficultés liées au calcul et au stockage de la matrice hessienne, comme c'est le cas avec la méthode de Newton. C'est principalement en 1952 et 1956 que R. Hestenes et E. Stiefel ont présenté cette technique pour des problèmes d'algèbre linéaire. La généralisation de la méthode du gradient conjugué à des cas non quadratiques a été réalisée par R. Fletcher et M. Reeves en 1964.
Les techniques de gradient conjugué ont pour objectif de minimiser la fonction objectif, en commençant par suivre le chemin de la pente la plus raide $-\nabla J(x_k)$ à partir d'un point de départ donné $x_0$. Ensuite, elles définissent des directions qui ont la particularité suivante : la nouvelle direction déterminée est une combinaison linéaire de la direction de la pente maximale et de la direction précédente.

### 2.3.1 Principe de la méthode du Gradient Conjugué

Le concept clé de la technique du gradient conjugué consiste à optimiser la méthode traditionnelle du gradient, qui s'appuie sur la pente maximale, afin d'accélérer la vitesse de convergence.

En d'autres termes, nous formulons la direction de descente de la manière suivante :

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

avec $R_k = \nabla J(x_k)$

La direction de descente est décomposée en deux parties : dans le $1^{er}$ terme : on maintient $\nabla J(x_k)$ pour garder tous les avantages du gradient et on ajoute un $2^{eme}$ terme $\beta_{k-1} d_{k-1}$ pour accélérer la vitesse de convergence.

Alors l'idée est : à chaque étape, l'algorithme cherche une direction qui est conjuguée à toutes les directions de recherche précédentes. Cette direction est ensuite utilisée pour calculer une approximation de la solution du système dans cette direction. Ensuite, l'algorithme modifie cette approximation pour qu'elle soit conjuguée à la direction de recherche précédente.

**Définition 2.3.1** *Soit $Q$ une matrice d'ordre $n * n$ symétrique, définie positive. Les directions $d_0, d_1, ..., d_k$ sont dites $Q$ Conjuguées s'ils vérifient :*

$$d_i^T Q d_j = 0 \quad \forall i, j \in [0, k] \;\; et \; i \neq j. \tag{2.3.1}$$

**Théorème 2.3.1** *Soit $Q(n * n)$ une matrice symétrique définie positive et $\{d_0, d_1, ..., d_k\}$ $Q$ conjuguée ; $d_i \neq 0$. Alors $\{d_0, d_1, ..., d_k\}$ est libre.*

**Preuve.** On montre que $\beta_0 = \beta_1 = ... = \beta_k = 0$, tel que

$$\beta_0 d_0 + \beta_1 d_1 + ... + \beta_k d_k = 0. \tag{2.3.2}$$

On multiplie (2.3.2) par $d_0^T Q$ on obtient :

$$\beta_0 d_0^T Q d_0 + \beta_1 d_0^T Q d_1 + ... + \beta_k d_0^T Q d_k = 0,$$

avec $d_0^T Q d_1 = ... = d_0^T Q d_k = 0$ car $\{d_0, d_1, ..., d_k\}$ Q conjuguée.

Alors $\beta_0 d_0^T Q d_0 = 0 \Rightarrow \beta_0 = 0$.

En suite on multiplie (3-14) par $d_1^T Q$ on obtient $\beta_1 d_1^T Q d_1 = 0 \Rightarrow \beta_1 = 0$,

Et on multiplie (2.3.2) jusqu'à $K^{ieme}$ itération on trouve $\beta_k d_k^T Q d_k = 0 \Rightarrow \beta_k = 0$.

Alors le système $\{d_0, d_1, ..., d_k\}$ est libre. ■

**Corollaire 2.3.1** *Soit $\{d_0, d_1, ..., d_k\}$ un systéme de direction $Q$-Conjugué, alors $\{d_0, d_1, ..., d_k\}$ est une base de $\mathbb{R}^n$.*

### 2.3.2 Méthode du gradient conjugué dans le cas non-quadratique

**Principe de la méthode**

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

$$\min \{J(x), x \in \mathbb{R}^n\}.$$

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

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

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

$$d_k = \begin{cases} -R_1 & \text{si } k = 1 \\ -R_k + \beta_{k-1} d_{k-1} & \text{si } k \geq 2 \end{cases}$$

**Défférentes formules de $\beta_k$**

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

$$\beta_k^{PRP} = \frac{R_{k+1}^T y_k}{||R_k||^2} \tag{2.3.3}$$

$$\beta_k^{FR} = \frac{||R_{k+1}||^2}{||R_k||^2} \tag{2.3.4}$$

$$\beta_k^{CD} = \frac{||R_{k+1}||^2}{-d_k^T R_k} \tag{2.3.5}$$

$$\beta_k^{DY} = \frac{||R_{k+1}||^2}{d_k^T y_k} \tag{2.3.6}$$

$$\beta_k^{LS} = \frac{R_{k+1}^T y_k}{-R_k^T d_k} \tag{2.3.7}$$

$$\beta_k^{HS} = \frac{R_{k+1}^T y_k}{d_k^T y_k} \tag{2.3.8}$$

$$\beta_{k+1}^{HZ} = (y_k - 2d_k \frac{||y_k||^2}{d_k^T y_k})^T \frac{R_{k+1}}{d_k^T y_k} \tag{2.3.9}$$

Où $y_k = R_{k+1} - R_k$.

Dans le cas où $J$ n'est pas quadratique on a

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

Les méthodes de Fletcher et Reeves (FR)[24], de Dai et yuan (DY)[44] et de la conjugué de descente (CD) proposé par Fletcher [10] ont une forte convergence . D'autre par les méthodes de polak-Ribière [9] et polyak (PRP)[19], de Hestenes et Stiefel (HS) [35]ou de LIU et Storey (LS)[39] ne sont pas toujours convergentes, mais ils ont souvent de meilleurs performances de calcul. L'une des méthodes de gradient conjugé qui est forte en théorie est suggérée par Hager et Zhang[40].

**Algorithme des différentes méthodes du gradient conjugué non linéaire**

**Etape 0 :(initialisation)**

Soit $x_0$ le point de départ, $R_0 = \nabla J(x_0)$, poser $d_0 = - R_0$.

Poser $k = 0$ et aller à l'étape 1.

**Etape 1 :**

Si $R_k = 0$ : STOP ($\hat{x} = x_k$).

Sinon aller à l'étape 2

**Etape 2 :**

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

$\lambda_k$ : calculer par la recherche linéaire

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

$\beta_k$ : définie selon la méthode.

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

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

**Cas de la recherche linéaire exacte :** Powell a démontré la satisfaction de la propriété de descente de la fonction objectif pour la méthode de Fletcher-Reeves avec recherche linéaire précise.

Soit $x_{k+1} = x_k + \lambda_k d_k$ la méthode itérative pour résoudre le probléme, avec $\lambda_k$ : calculé par la recherche linéaire précise, et

$$d_k = \begin{cases} -R_1 & \text{si } k = 1 \\ -R_k + \beta_{k-1} d_{k-1} & \text{si } k \geq 2 \end{cases}$$

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

Cependant, il est fortement conseillé de ne pas procéder à une recherche linéaire précise lorsque $J$ n'est pas quadratique : le coût pour déterminer $k$ est trop élevé.

Dans le cas d'une recherche linéaire imprécise : le premier théorème établissant la descente d'une méthode du gradient conjugué non linéaire a été formulé par Albaali (1985) pour la méthode de Fletcher-Reeves, en utilisant la recherche Wolfe forte avec $\sigma \leq (1/2)$.

Gilbert et Nocedal (1992) ont élargi cette découverte à tous les algorithmes de gradient conjugué dont :

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

Il n'existe aucun théorème prouvant que la méthode non linéaire de Polak-Ribiére-Polyak respecte la propriété de descente. Fletcher (1987) a montré que l'approche de la descente conjuguée est une méthode de descente si le paramètre $\lambda_k$ est défini selon la règle stricte de Wolfe avec $\sigma \leq (1/2)$.

Dai et Yuan (1996) ont démontré que cette méthode, qui utilise la règle de Wolferelaxée, génère des directions de descente adéquates à chaque itération pour$k \geq 1$. De plus, Dai et Yuan (1998) ont prouvé qu'à chaque itération $k \geq 1$, l'orientation souhaitée par la méthode de Dai-Yuan, en utilisant la recherche Wolfe faible, est décroissante si la fonction objectif $J$ présente une stricte convexité.

# CHAPITRE 3

# Algorithme du gradient conjugué linéaire

L'algorithme du gradient conjugué linéaire a été spécialement développé pour minimiser les fonctions quadratiques convexes ou pour résoudre des systèmes d'équations algébriques linéaires avec des matrices définies positives. Hestenes et Stiefel ont introduit cet algorithme en 1952. Considérons la fonction quadratique

$$J(x) = \frac{1}{2}x^T Ax - b^T x \tag{3.0.1}$$

où $A \in \mathbb{R}^{n\times n}$ est une matrice définie symétrique et positive, et $b \in \mathbb{R}^n$ est un vecteur connu. On a :

$$\nabla J(x) = Ax - b,\ \nabla^2 J(x) = A \tag{3.0.2}$$

Remarquez que le Hessien de la fonction est indépendant de $x$. Puisque le Hessian $A$ est symétrique et défini positif, d'après les conditions d'optimalité pour un minimum d'une fonction différentiable, il s'ensuit qu'il existe un minimum unique $x^*$. Observons que $x^*$ est la solution du système linéaire

$$Ax = b$$

Sachant que la fonction $J$ est quadratique, d'après le théorème de Taylor, pour tout $t \in \mathbb{R}$ et tout $y,\ z \in \mathbb{R}^n$, on obtient l'identité suivante :

$$J(y + tz) = J(y) + t\nabla J(y)^T z + \frac{t^2}{2}z^T Az \tag{3.0.3}$$

## 3.1 Recherche linéaire

L'algorithme de gradient conjugué linéaire est exactement une recherche linéaire avec un choix spécial de directions. Étant donné l'approximation actuelle $x_j$ du minimum $x^*$ ainsi qu'un vecteur directeur $d_j$, un algorithme de recherche linéaire calcule l'approximation suivante $x_{j+1}$ en utilisant les deux étapes suivantes :

1. Trouvez le pas $\alpha_j = \arg\min J(x_j + \alpha d_j)$
2. Définir $x_{j+1} = x_j + \alpha_j d_j$

En supposant $x_0$ un point initial , alors, en appliquant $k$ étapes de la méthode de recherche linéaire ci-dessus, $k$ itérations sont obtenues : $\{x_0, x_1, ..., x_{k+1}\}$. Le pas $\alpha_j$ est calculé comme suit :

$$\alpha_j = \frac{-d_j^T r_j}{d_j^T A d_j} \tag{3.1.1}$$

où $r_j = Ax_j - b$ est le résidu en $x_j$.

**Définition 3.1.1** *L'ensemble des directions $\{d_0, ..., d_{k+1}\}$ est un ensemble de directions conjuguées si et seulement si $d_j^T A d_i = 0$ pour tout $i = 0, ..., k+1, j = 0, ..., k+1$ et $j \neq j$.*

Maintenant, pour tout $k = 1, ...$, nous définissons les espaces vectoriels et affines suivants :

$$W_k = Vect\{d_0, ..., d_{k+1}\} \tag{3.1.2}$$

$$U_k = x_0 + W_k = \{z \in \mathbb{R}^n : z = x_0 + w_k; w_k \in W_k\} \tag{3.1.3}$$

Notons $W_0 = \{0\}$ et $U_0 = \{x_0\}$.

**Proposition 3.1.1** *Supposons que $d_i^T A d_j = 0$ pour tout $0 \leq j < i$, où $i$ est un entier fixe, et que $\{x_0, ..., x_i\}$ sont calculés par l'algorithme de recherche linéaire. Alors*

$$d_j^T r_j = d_j^T \nabla f(y) \tag{3.1.4}$$

*pour tous $y \in U_i$*

**Preuve.** Tout d'abord, notez que puisque$\{x_0, ..., x_i\}$ sont calculés par l'algorithme de recherche linéaire, il s'ensuit que $x_i \in U_i$. Si $y \in U_i$, alors, de la définition de $U_i$, il s'ensuit que $x_i y \in W_i$ et donc $d_i^T A(x_i - y) = 0$. Par conséquent,

$$d_i^T(r_i - \nabla f(y)) = d_i^T(Ax_i - b + Ay + b) = d_i^T A(x_i - y) = 0$$

ce qui prouve (3.1.4) . ■

## 3.2 Propriété fondamentale de la méthode de recherche linéaire avec directions conjuguées

Évidemment, à chaque étape, l'algorithme de recherche linéaire minimise $J(x)$ uniquement dans une direction fixe. Cependant, si les directions sont conjuguées selon la définition 3.1, alors un résultat plus fort peut être prouvé, comme dans le théorème 3.1 ci-dessous : Un choix de directions conjuguées dans la méthode de recherche linéaire conduit à l'obtention d'un minimiseur $x_k$ pour tout l'espace $U_k$.

**Théorème 3.2.1** *Si les directions dans l'algorithme de recherche de ligne sont conjuguées et* $\{x_0, x_1, ..., x_k\}$ *sont les itérations générées après* $k$ *étapes de l'algorithme de recherche linéaire, alors*

$$x_j = \arg\min_{x \in U_j} f(x)$$

*pour tout* $1 \leq j \leq k$.

**Preuve.** Le théorème est prouvé par récurrence. Pour $k = 1$, le résultat est obtenu à partir de la définition de $x_1$ comme minimiseur sur $U_1$. Supposons que pour $k = i$,

$$x_j = \arg\min_{y \in U_j} J(y)$$

pour tous les $1 \leq j \leq i$. Nous devons montrer que si $x_{i+1} = x_i + \alpha_i d_i$, alors

$$x_{j+1} = \arg\min_{y \in U_{j+1}} J(y)$$

Par définition de $U_{i+1}$, tout $x \in U_{i+1}$ peut s'écrire $x = y + \alpha d_i$, où $\alpha \in \mathbb{R}$ et $y \in U_i$. Maintenant, en utilisant (3.0.3) et la proposition 3.1, on obtient

$$\begin{aligned} J(x) &= J(y + \alpha d_i) = J(y) + \alpha d_i^T \nabla J(y) + \frac{\alpha^2}{2} d_i^T A d_i \\ &= J(y) + \left[ \alpha d_i^T \nabla J(y) + \frac{\alpha^2}{2} d_i^T A d_i \right] \end{aligned} \tag{3.2.1}$$

Remarquons que (3.2.1) est une fonction découplée. Le premier terme du membre de droite de (3.2.1) ne dépend pas de $\alpha$, et le deuxième terme ne dépend pas de $y$. Donc

$$\min_{x \in U_{j+1}} J(x) = \min_{y \in U_{j+1}} J(y) + \min_{\alpha \in \mathbb{R}} \left[ \alpha d_i^T \nabla J(y) + \frac{\alpha^2}{2} d_i^T A d_i \right] \tag{3.2.2}$$

Mais le membre de droite de (3.2.2) est minimisé lorsque $y = x_i$ et

$$\alpha = \alpha_i = \alpha_j = \frac{-d_i^T r_i}{d_i^T A d_i}$$

c'est-à-dire que le côté gauche de (3.2.2) est minimisé exactement pour $x_{i+1} = x_i + \alpha_i d_i$. En d'autres termes, $x_{i+1}$ est le minimiseur de $J$ sur l'ensemble $\{x : x = x_0 + Vect\{d_0, ..., d_{k+1}\}\}$ ■

Pour montrer l'importance de rendre les directions de recherche mutuellement conjuguées par rapport à $A$, énonçons et prouvons d'abord un résultat technique impliquant uniquement les deux premières itérations de l'algorithme du gradient conjugué.

**Proposition 3.2.1** *Après deux itérations de la méthode du gradient conjugué, le gradient $g_2 = Ax_2 - b$ satisfait*

$$d_1^T g_2 = d_0^T g_2 = 0$$

**Preuve.** Après la première itération, le nouveau point est $x_1$. Par conséquent, $R_1 = Ax_1 - b$. Puisque la recherche linéaire exacte, nous avons aussi $d_0^T R_1 = 0$.

Considérons maintenant la deuxième itération. $A$ cette itération, l'algorithme va générer un point $x_2 = x_1 + \alpha d_1$ où $R_2 = Ax_2 - b$ et $d_1^T R_2 = 0$. Mais on a

$$d_0^T R_2 = d_0^T \left( Ax_1 + \alpha A d_1 - b \right) = d_0^T R_1 + \alpha d_0^T A d_1$$

Le premier terme, $d_0^T R_1$, du côté droit de l'égalité ci-dessus est nul en raison de la recherche linéaire lors de la première itération. Le deuxième terme, $\alpha d_0^T A d_1$, est nul car $d_0$ et $d_1$ sont conjugués par rapport à $A$. ■

Ce résultat montre qu'après deux itérations, le gradient est orthogonal aux deux directions de recherche $d_0$ et $d_1$. De même, le résultat ci-dessus peut être généralisé pour prouver la proposition suivante.

**Proposition 3.2.2** *Après $k$ itérations de la méthode du gradient conjugué, le gradient $R_k = Ax_k - b$ satisfait*

$$d_j^T R_k = 0, \ \textit{pour } j = 1, 2, ..., k$$

Cette proposition implique qu'après $k$ itérations, le gradient $R^k$ est restreint au sous-espace de dimension $(n-k)$ orthogonal aux vecteurs $d_0, ..., d_{k-1}$. De là, on peut obtenir l'importante propriété de terminaison finie de la méthode du gradient conjugué.

**Proposition 3.2.3** *La méthode du gradient conjugué résout un $n \times n$ système algébrique linéaire $Ax = b$ en $n$ itérations au maximum.*

**Preuve.** La proposition 3.3 implique que, après $n$ itérations, $R_n$ est orthogonal aux $n$ vecteurs $d_0, ..., d_{N-1}$. Mais cela signifie que $R_n$ doit se situer dans un sous-espace de dimension zéro et donc $R_n = 0$, ce qui prouve que $Ax_n = b$. ■

La propriété de terminaison finie n'est garantie que lorsque les calculs sont exacts. En pratique, la conclusion de la proposition 3.3 peut ne pas être exactement satisfaite lorsque les itérations sont effectuées en arithmétique réelle, sujette à des erreurs d'arrondi. Par conséquent, pour résoudre certains $n \times n$ systèmes algébriques linéaires, la méthode du gradient conjugué nécessite un peu plus de $n$ itérations.

## 3.3 L'algorithme de gradient conjugué linéaire

Le théorème 3.1 fournit les bases nécessaires pour développer l'algorithme du gradient conjugué linéaire en générant des directions conjuguées. Tout d'abord, une relation récurrente générale est démontrée pour engendrer un ensemble de directions conjuguées. Ensuite, cette relation récurrente est simplifiée pour aboutir à une expression plus concise. Enfin, nous exposons en détail l'algorithme du gradient conjugué linéaire.

**Proposition 3.3.1** *Soit $d_0 = r_0$ et pour $k = 1, 2, ...$ et*

$$d_k = -r_k + \sum_{j=0}^{k-1} \frac{d_j^T A r_k}{d_j^T A d_j} d_j \tag{3.3.1}$$

*Alors, $d_j^T A d_m = 0$ pour tout $0 \leq m < i \leq k$.*

**Preuve.** Par récurrence, on montre que(3.3.1) engendre des directions conjuguées. Pour $k = 1$, $d_1^T A d_0 = 0$. Supposons que pour $k = i$, les vecteurs $\{d_0, ..., d_i\}$ sont conjugués deux à deux. Il faut montrer que $d_{i+1}^T A d_m = 0$ pour tout $m \leq i$. Considérons $m \leq i$. Alors

$$\begin{aligned} d_{i+1}^T A d_m &= -r_{i+1}^T A d_m + \sum_{j=0}^{i} \frac{d_j^T A r_{i+1}}{d_j^T A d_j} d_j^T A d_m \\ &= -r_{i+1}^T A d_m + \frac{d_m^T A r_{i+1}}{d_m^T A d_m} d_m^T A d_m = 0, \end{aligned}$$

ce qui prouve la proposition. ■

**Proposition 3.3.2** *Soient $\{d_0, ..., d_k\}$ les directions engendrées par (3.3.1). Alors*

*(i) $W_k = span\{r_0, ..., r_{k-1}\}$*

*(ii) $r_m^T r_j = 0$, pour tout $0 \leq j < m \leq k$,*

*(iii) $d_m^T r_j = -r_k^T r_k$, pour tout $0 \leq j \leq k$,*

*(iv) La direction $d_k$ satisfait*

$$d_k = -r_k + \beta_{k-1} d_{k-1}, \tag{3.3.2}$$

*où*

$$\beta_{k-1} = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} \tag{3.3.3}$$

*De (3.4) et de la proposition 3.6 (iii), il résulte que*

$$\alpha_k = \frac{-d_k^T r_k}{d_k^T A d_k} = \frac{r_k^T r_k}{d_k^T A d_k} \tag{3.3.4}$$

*D'autre part, le vecteur résiduel $r_{k+1}$ peut s'écrire comme suit :*

$$r_{k+1} = A x_{k+1} - b = A x_k - b + \alpha_k A d_k = r_k + \alpha_k A d_k, \tag{3.3.5}$$

Avec ceux-ci, en utilisant la proposition 3.3 et la remarque 3.1, on présenté l'algorithme de gradient conjugué linéaire suivant :

**Algorithm 3.3.1** *Gradient conjugué linéaire*

1. *Sélectionnez un point initial* $x_0$ *et* $\varepsilon > 0$ *suffisamment petit*
2. *Définir* $r_0 = Ax_0 - b, d_0 = r_0$ *et* $k = 0$,
3. *Si* $\|r_k\| \leq \varepsilon$, *alors arrêtez, sinon, continuez avec l'étape 4*
4. *Calculer :* $\alpha_k = \frac{r_k^T r_k}{d_k^T A d_k}$, $x_{k+1} = x_k + \alpha_k d_k$, $r_{k+1} = r_k + \alpha_k A d_k$, $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$, $d_{k+1} = -r_{k+1} + \beta_k d_K$
5. *Posé* $k = k + 1$ *et allez à l'étape 3* ■

L'algorithme est simple et très facile à mettre en œuvre. Le résidu initial est le même que la première direction de descente du gradient. Si la solution initiale $x_0$ est nulle, alors $r_0 = -b$ et $d_0 = b$. Remarquez que si $A$ n'est pas symétrique et défini positif, alors le dénominateur dans $\alpha_k$ peut s'annuler, ce qui entraîne la rupture des itérations.

## 3.4 Taux de convergence de l'algorithme du gradient conjugué linéaire

Dans ce qui suit, une estimation du taux de convergence de l'algorithme du gradient conjugué linéaire est présentée. Pour cela, la réduction d'erreur dans l'algorithme du gradient conjugué linéaire est discutée. Ensuite, sur la base des polynômes de Chebyshev, on donne une estimation du taux de convergence.

**Proposition 3.4.1** *La relation suivante est vraie :*

$$W_k = span\left\{r_0, ..., A^{k-1} r_0\right\} \tag{3.4.1}$$

Dans ce qui suit, nous présentons une estimation d'erreur générale qui relie $\|x^* - x_k\|_A$ et $\|x^* - x_0\|_A$ , où pour tout $y \in \mathbb{R}^n$ ; $\|y\|_A^2 = y^T A y$ : On, note $\overline{P}_k$ l'ensemble des polynômes de degré inférieur ou égal à $k$.

**Proposition 3.4.2** *L'estimation suivante est valable :*

$$\|x^* - x_k\|_A = \inf_{P \in \overline{P}_k, P(0)=1} \|P(A)\,(x^* - x_0)\|_A \tag{3.4.2}$$

**Remarque 3.4.1** ***Preuve.*** *Puisque* $r_k$ *est orthogonal à* $W_k$, *pour tout* $y \in W_k$ *on a :*

$$(x^* - x_k)^T A y = r_k^T y = 0 \tag{3.4.3}$$

*Désignant* $w_k = x_k - x_0$ *et* $e_0 = x^* - x_0$, *nous obtenons*

$$0 = (x^* - x_k)^T Ay = (e_0 - w_k)^T Ay$$

*pour tout* $y \in W_k$. *Donc,* $w_k = x_k - x_0$ *est une projection* $A-$*orthogonale de* $e_0$ *sur* $W_k$. *Ainsi,*

$$\|e_0 - w_k\|_A = \min_{w \in W_k} \|e_0 - w\|_A$$

*Mais, d'après la proposition 3.7, on sait que* $w = Q_{k-1}(A)r_0$ *pour un polynôme* $Q_{k-1} \in \bar{P}_{k-1}$ *est l'ensemble des polynômes de degré inférieur ou égal à* $k-1$. *Aussi,* $Ae_0 = -r_0$ *et* $e_0 - w = (I + Q_{k-1}(A)A)\, e_0$, *et donc,*

$$\|x^* - x_k\|_A = \|e_0 - w_k\|_A = \min_{P \in \overline{P}_k, P(0)=1} \|P(A)e_0\|_A \tag{3.4.4}$$

*ce qui complète la preuve.* ■

Ce taux de convergence est assez général et ne prend pas en compte la connaissance de la distribution des valeurs propres de $A$. Afin d'affiner les résultats ci-dessus et d'obtenir une estimation qualitative du côté droit de (3.4.4), observons que pour $A$ symétrique et défini positif la décomposition spectrale suivante peut s'écrire comme suit :

$$A = U\Lambda U^T$$

où $U$ est une matrice orthogonale dont les colonnes sont les vecteurs propres de $A$, et $\Lambda$ est une matrice diagonale avec les valeurs propres positives de $A$, $\lambda_1, \lambda_2, ..., \lambda_n$ sur la diagonale. Puisque $UU^T = U^TU = I$ par orthogonalité de $U$, pour tout $j$ on a,

$$A^j = U\Lambda^j U^T$$

Donc,

$$P_k(A) = UP_k(\Lambda)U^T$$

Définir $A^{\frac{1}{2}} = U\Lambda^{\frac{1}{2}}U^T$. Remarquons $\|x\|_A^2 = x^TAx = \left\|A^{\frac{1}{2}}x\right\|_2^2$.Par conséquent, pour tout $x \in \mathbb{R}^n$ on a :

$$\|P_k(A)x\|_A = \left\|A^{\frac{1}{2}}P_k(A)x\right\|_2 \leq \|P_k(A)\|_2 \left\|A^{\frac{1}{2}}x\right\|_2 \leq \|P_k(A)\|_2 \|x\|_2 .$$

Ceci, combiné à (3.4.4), implique que pour tout polynôme $P_k(\lambda)$,

$$\|x^* - x_k\|_A \leq \min_{P_k \in \bar{P}_k, P_k(0)=1} \max_{1 \leq j \leq n} |P_k(\lambda_j)| \, \|e_0\|_A \tag{3.4.5}$$

où $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ sont les valeurs propres de $A$. L'inégalité ci-dessus montre que minimiser l'erreur dans le gradient conjugué linéaire correspond à minimiser le polynôme $P_k(\lambda)$ sur toute la plage de valeurs propres $[\lambda_1, \lambda_n]$ . Ceci peut être accompli via les polynômes de Chebyshev. Les polynômes de Tchebychev de première espèce sur $[-1, 1]$ sont définis comme

$$T_k(\zeta) = \cos(k \arcsin(\zeta)), k = 0, 1, ...$$

Il est facile de voir que $T_k(\zeta)$ est un polynôme si les identités trigonométriques suivantes sont utilisées :

$$\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$$

$$\cos(\alpha + \beta) + \cos(\alpha - \beta) = 2\cos(\alpha)\cos(\beta)$$

Notons $\theta = ar\cos(\zeta)$, alors

$$\begin{aligned} T_0(\zeta) &= \cos(0\theta) = 1 \\ T_1(\zeta) &= \cos(1\theta) = \zeta \\ T_2(\zeta) &= \cos(2\theta) = \cos^2(\theta) - \sin^2(\theta) = 2\cos^2(\theta) - 1 = 2\zeta - 1, \\ T_{k+1}(\zeta) + T_{k-1}(\zeta) &= \cos((k+1)\theta) + \cos((k-1)\theta) \\ &= 2\cos(k\theta)\cos(\theta) = 2\zeta T_k(\zeta) \end{aligned}$$

Donc,

$$T_0(\zeta) = 1, T_1(\zeta) = \zeta \tag{3.4.6}$$

$$T_{k+1}(\zeta) = 2\zeta T_k(\zeta) - T_{k-1}(\zeta) \tag{3.4.7}$$

pour tout $\zeta \in \mathbb{R}^n$.

De (3.4.7) pour $\zeta$ fixé, il résulte que

$$T_k(\zeta) = c_1(\eta_1(\zeta))^k + c_2(\eta_2(\zeta))^k, k = 0, 1, ...$$

où $\eta_1(\zeta)$ et $\eta_2(\zeta)$ sont les racines de l'équation caractéristique

$$\eta^2 - 2\zeta\eta + 1 = 0$$

Les constantes $c_1$ et $c_2$ sont déterminées à partir des conditions initiales (3.4.6). Par conséquent,

$$T_k(\zeta) = \frac{1}{2}\left[\left(\zeta + \sqrt{\zeta^2 - 1}\right)^k + \left(\zeta - \sqrt{\zeta^2 - 1}\right)^k\right] \tag{3.4.8}$$

On observe que $|T_k(\zeta)| \leq 1$ pour tout $\zeta \in [-1, 1]$. Le polynôme qui minimise (3.4.5) sur l'intervalle $[\lambda_1, \lambda_n]$ est

$$S_k(\lambda) = \left[T_k\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)\right]^{-1}\left[T_k\left(\frac{\lambda_n + \lambda_1 - 2\lambda}{\lambda_n - \lambda_1}\right)\right] \tag{3.4.9}$$

Pour le prouver, supposons qu'il existe un autre polynôme de degré $k$, $Q_k$, qui minimise mieux (3.4.5) on pose l'intervalle $[\lambda_1, \lambda_n]$, tel que ,$Q_k(0) = 1$,

$$Q_k(\lambda) < \left[T_k\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)\right]^{-1}$$

Le polynôme $P_k - Q_k$ doit avoir un zéro à $\lambda = 0$ et à $k$ zéros des polynômes, ce qui signifie que ce polynôme doit avoir $k + 1$ zéros, ce qui est une contradiction. Par conséquent, $S_k$ de (3.4.9) doit être le polynôme minimisant sur l'intervalle $[\lambda_1, \lambda_n]$ . Par conséquent, de (3.4.5), on a

$$\|x^* - x_k\|_A \leq \left[T_k\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)\right]^{-1}\|x^* - x_0\|_A \tag{3.4.10}$$

**Théorème 3.4.1** *L'erreur après $k$ itérations de l'algorithme du gradient conjugué linéaire peut être limitée comme suit :*

$$\|x^* - x_k\|_A \leq \frac{2}{\left(\frac{\sqrt{k}+1}{\sqrt{k}-1}\right)^k + \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k}\|x^* - x_0\|_A \leq 2\left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k\|x^* - x_0\|_A \tag{3.4.11}$$

*où $k = k(A)$ est le nombre de condition de $A$.*

**Preuve.** Le but est de calculer

$$\left[T_k\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)\right]^{-1}.$$

De (3.4.8), pour $\zeta = \frac{\lambda_n+\lambda_1}{\lambda_n-\lambda_1} = \frac{k+1}{k-1}$, on a

$$\zeta \pm \sqrt{\zeta^2-1} = \frac{k+1}{k-1} \pm \frac{2\sqrt{k}}{k-1} = \frac{k+1\pm 2\sqrt{k}}{k-1} = \frac{\left(\sqrt{k}\pm 1\right)^2}{k-1} = \frac{\left(\sqrt{k}\pm 1\right)^2}{\left(\sqrt{k}-1\right)\left(\sqrt{k}+1\right)} = \frac{\sqrt{k}\pm 1}{\sqrt{k}\mp 1}$$

Donc,

$$T_k\left(\frac{\lambda_n+\lambda_1}{\lambda_n-\lambda_1}\right) = \frac{1}{2}\left[\left(\frac{\sqrt{k}+1}{\sqrt{k}-1}\right)^k + \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k\right]$$

D'où,

$$\left[T_k\left(\frac{\lambda_n+\lambda_1}{\lambda_n-\lambda_1}\right)\right]^{-1} = \frac{2}{\left[\left(\frac{\sqrt{k}+1}{\sqrt{k}-1}\right)^k + \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k\right]} \leq 2\left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k \tag{3.4.12}$$

La preuve est complétée en remplaçant (3.4.12) dans (3.4.10). ■

Connaissant seulement les valeurs élevées et les plus petites de $A$, la borne (3.4.11) est la meilleure possible. Le théorème 3.2 montre que l'erreur $\|x^* - x_k\|_A$ est bornée par une suite qui converge vers zéro. De plus, la convergence est monotone, ce qui explique pourquoi l'algorithme du gradient conjugué linéaire est considéré comme une méthode itérative. En tant que méthode itérative, les performances de l'algorithme du gradient conjugué linéaire dépendent à la fois de $b$ et du spectre de $A$ [voir Kelley (1995)[22], Greenbaum (1997) [11]. Le gradient conjugué linéaire fonctionnera bien si $j$ est proche de 1, et il peut fonctionner très mal si $j$ est grand. Géométriquement, $j$ est grand si les surfaces de niveau ellipsoïdales de la fonction quadratique sont très loin d'être sphériques.

Si des informations supplémentaires sur les valeurs propres de $A$ dans l'intervalle $[\lambda_1, \lambda_n]$ sont disponibles, alors l'estimation (3.4.11) peut être améliorée. Supposons, par exemple, que $A$ possède une valeur propre beaucoup plus grande que les autres, c'est-à-dire $\lambda_1 \leq ... \leq \lambda_{n-1} \ll \lambda_n$ ; soit $\lambda_n/\lambda_1 \gg 1$ : Considérons un polynôme $P_k$ qui est le produit d'un facteur linéaire qui est nul à $\lambda_n$ et du polynôme de Tchebychev appelé et décalé de degrés $(k-1)$sur l'intervalle $[\lambda_1, \lambda_{n-1}]$ :

$$P_k(\lambda) = \frac{T_{k-1}\left(\frac{\lambda_{n-1}+\lambda_1-2\lambda}{\lambda_{n-1}-\lambda_1}\right)}{T_{k-1}\left(\frac{\lambda_{n-1}+\lambda_1}{\lambda_{n-1}-\lambda_1}\right)}\frac{\lambda_n-\lambda}{\lambda_n} \tag{3.4.13}$$

Comme le second facteur de (3.4.13) est nul à $\lambda_n$ et inférieur à un en valeur absolue à chacune des autres valeurs propres, alors la valeur absolue maximale de ce polynôme dans tout le spectre $\{\lambda_1, ..., \lambda_n\}$ de $A$ est inférieure à la valeur absolue maximale du premier facteur sur $\{\lambda_1, ..., \lambda_{n-1}\}$. Par conséquent, en utilisant des arguments similaires à ceux du théorème 3.2, il s'ensuit que

$$\|x^* - x_k\|_A \leq 2\left(\frac{\sqrt{k_{n-1}} - 1}{\sqrt{k_{n-1}} + 1}\right)^{k-1} \|x^* - x_0\|_A \tag{3.4.14}$$

où $k_{n-1} = \frac{\lambda_{n-1}}{\lambda_1}$ . Axelsson et Lindskog (1986) est donnée une étude détaillée du cas des valeurs propres isolées.

De même, si la matrice $A$ n'a que quelques grandes valeurs propres, disons, $\lambda_1 \leq ... \leq \lambda_{n-m} \ll \lambda_{n-m+1} \leq ... \leq \lambda_n$, i.e $\lambda_{n-m+1}/\lambda_{n-m} \gg 1$, alors on peut considérer un polynôme $P_k$ qui est le produit d'un facteur de degré $m$ qui est nul à chaque grande valeur propre et d'un polynôme de Tchebychev mis à l'échelle et décalé de degré $k$ sur l'intervalle $[\lambda_1, \lambda_{n-m}]$ . En limitant la taille de ce polynôme, il en résulte que

$$\|x^* - x_k\|_A \leq 2\left(\frac{\sqrt{k_{n-m}} - 1}{\sqrt{k_{n-m}} + 1}\right)^{k-m} \|x^* - x_0\|_A \tag{3.4.15}$$

où $k_{n-m} = \frac{\lambda_{n-m}}{\lambda_1}$ [Voir Greenbaum (1997) [2], Van der Vorst (1993) [17].

**Remarque 3.4.2** *Il est généralement vrai que si les valeurs propres de la matrice $A$ apparaissent dans $m$ groupes distincts, alors les itérations de l'algorithme du gradient conjugué linéaire résoudront approximativement le problème en environ $m$ étapes.*

## 3.5 Comparaison du taux de convergence du gradient conjugué linéaire et de la descente la plus raide

Comme nous l'avons déjà vu dans la proposition 3.4, l'algorithme du gradient conjugué linéaire a une propriété de terminaison quadratique (finie), c'est-à-dire que pour les fonctions quadratiques convexes, l'algorithme du gradient conjugué linéaire avec recherche linéaire exacte se termine après $n$ itérations. Dans (3.4.10), (3.4.11), (3.4.14) et (3.4.15), certaines formules pour les taux de convergence de l'algorithme du gradient conjugué linéaire ont été présentées, montrant que

le taux de convergence des algorithmes du gradient conjugué linéaire n'est pas pire que celui de l'un des algorithmes de descente la plus raide, c'est-à-dire qu'il n'est pas pire que celui du linéaire.

Dans la suite, nous allons comparer l'algorithme du gradient conjugué linéaire et l'algorithme de descente la plus raide soumis à la réduction des valeurs de la fonction au fil des itérations (Sun & Yuan, 2006) [42]. Considérons la fonction quadratique

$$J(x) = \frac{1}{2}x^T Ax \tag{3.5.1}$$

où $A \in \mathbb{R}^{n\times n}$ est symétrique et défini positif. Dans ce cas, l'expression explicite du pas est

$$\alpha_k = \frac{d_k^T Ax_k}{d_k^T Ad_k} = -\frac{d_k^T R_k}{d_k^T Ad_k} \tag{3.5.2}$$

Donc,

$$\begin{aligned} J(x_{k+1}) &= \frac{1}{2}x_{k+1}^T Ax_{k+1} && (3.5.3)\\ &= \frac{1}{2}\left(x_k + \alpha_k d_k\right)^T A\left(x_k + \alpha_k d_k\right) \\ &= \frac{1}{2}x_k^T Ax_k - \frac{1}{2}\frac{\left(d_k^T R_k\right)^2}{d_k^T Ad_k} && (3.5.4)\end{aligned}$$

Maintenant, pour l'algorithme de descente la plus raide $d_k = -R_k$ et à partir de (3.5.3) ,

$$J(x_k^{DR}) = \frac{1}{2}x_k^T Ax_k - \frac{1}{2}\frac{\|R_k\|^4}{R_k^T AR_k} \tag{3.5.5}$$

Par contre,, pour l'algorithme du gradient conjugué linéaire $d_k = -R_k + \beta_{k+-1}d_{k-1}$ et d'après (3.5.3),

$$J(x_k^{GC}) = \frac{1}{2}x_k^T Ax_k - \frac{1}{2}\frac{\|R_k\|^4}{d_k^T Ad_k} \tag{3.5.6}$$

Puisque

$$\begin{aligned} d_k^T Ad_k &= \left(-R_k + \beta_{k-1}d_{k-1}\right)^T A\left(-R_k + \beta_{k-1}d_{k-1}\right) \\ &= R_k^T AR_k + \beta_{k-1}^2 d_{k-1}^T Ad_{k-1} \\ &\leq R_k^T AR_k \end{aligned}$$

il en résulte que

$$J(x_k^{GC}) \leq J(x_k^{DR}).$$

Par conséquent, l'algorithme du gradient conjugué linéaire réduit la valeur de la fonction de minimisation $J$ au moins autant que l'algorithme de descente la plus raide. Étant donné que l'algorithme de descente la plus raide a un taux de convergence linéaire, puisque l'algorithme du gradient conjugué linéaire a un taux de convergence qui n'est pas pire que le taux linéaire. De (3.5.6), il résulte que pour l'algorithme du gradient conjugué linéaire, la fonction objectif est strictement diminuée au cours des itérations.

**Préconditionnement des algorithmes de gradient conjugué linéaire**

Le préconditionnement est une technique d'accélération de la méthode du gradient conjugué. L'idée est de changer les variables de $x$ à $\hat{x}$ par une matrice non singulière $C \in \mathbb{R}^{n\times n}$, c'est-à-dire

$$\hat{x} = Cx. \tag{3.5.7}$$

La fonction quadratique J donnée par (3.0.1) est transformée comme

$$\left(C^{-T}AC^{-1}\right)\hat{x} = C^{-T}b \tag{3.5.8}$$

peut être résolu. Le taux de convergence de l'algorithme 3.1 dépendra de la structure des valeurs propres de la matrice $C^{-T}AC^{-1}$ plutôt que de celles de $A$. Par conséquent, le but du préconditionnement est de choisir $C$ de telle manière que la structure des valeurs propres de la matrice $C^{-T}AC^{-1}$ soit meilleure pour la théorie de convergence présentée ci-dessus. Il existe plusieurs possibilités pour choisir la matrice préconditionnement $C$. Par exemple, $C$ peut être sélectionné de telle sorte que le numéro de condition de $C^{-T}AC^{-1}$ soit plus petit que le nombre de conditions de $A$. Dans ce cas, la constante dans (3.4.11) est bien plus petite. Une autre possibilité est de choisir la matrice $C$ de telle manière que les valeurs propres de $C^{-T}AC^{-1}$ soient groupées. Dans ce cas, selon la discussion ci-dessus, le nombre d'itérations pour trouver une bonne solution approximative n'est pas beaucoup plus grand que le nombre de groupes.

Une implémentation pratique du gradient linéaire préconditionnement ne nécessite pas explicitement la transformation (3.5.7), mais la matrice $M = C^TC$, qui est une matrice définie positive et symétrique , comme le montre l'algorithme suivante :

**Algorithm 3.5.1** *Gradient conjugué linéaire préconditionné*

| | |
|---|---|
| *étape 1.* | *Sélectionnez un point initial* $x_0$*, avec la condition préalble* |
| | $M$ *et* $\varepsilon > 0$ *suffisamment petit* |
| *étape 2.* | *Posez* $r_0 = Ax_0 - b$*. Résolvez le système* $My_0 = r_0$*. Posez* |
| | $d_0 = y_0$ *et* $k = 0$ |
| *étape 3.* | $S\ \|r_0\| \leq \varepsilon$ *arrêtez, sinon continuez avec l'étape 4.* |
| *étape 4.* | *Calculer :* $\alpha_k = \frac{r_k^T y_k}{d_k^T A d_k},\ x_{k+1} = x_k + \alpha_k d_k,\ r_{k+1} = r_k + \alpha_k A d_k,$ *Résoudre* $My_{k+1} = r_{k+1},\ \beta_k = \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k},\ d_{k+1} = -y_{k+1} + \beta_k d_k$ |
| *étape 5.* | *Posez* $k = k + 1$ *et alez à l'étape 3* ■ |

La différence principale entre les méthodes de gradient conjugué linéaire préconditionnées et non préconditionnées est la nécessité de résoudre le système linéaire de la forme $My = r$ à l'étape 4 de l'algorithme présidente.

La sélection du préconditionneur $M$ n'est pas facile. Les préconditionneurs efficaces sont basés sur une compréhension approfondie de la structure du problème. La sélection de $M$ est basée sur différents objectifs, comme l'efficacité de $M$ pour obtenir une bonne structure des valeurs propres du problème préconditionné, un stockage et un calcul peu coûteux de $M$, une solution peu coûteuse de $My = r$, etc. Souvent, le préconditionneur est défini de telle manière que le système $My = r$ soit une version simplifiée du système original $Ax = b$.Un préconditionneur simple est le préconditionnement de Jacoby, où $M$ est l'inverse des éléments diagonaux de $A$. D'autres préconditionneurs sont basés sur les méthodes itératives stationnaires, telles que le préconditionneur symétrique de Gauss-Seidel. Une autre approche consiste à appliquer une factorisation de Cholesky creuse à la matrice $A$ et à abandonner les petits éléments des facteurs et/ou à n'autoriser qu'une quantité fixe de stockage de facteurs. De tels préconditionneurs sont appelés factorisation de Cholesky incomplète.L'idée est la suivante : au lieu de calculer le facteur de Cholesky exact $L$ qui satisfait $A = LL^T$, on calcule un facteur d'approximation $\bar{L}$ de sorte que $A = \bar{L}\bar{L}^T + E$ où $E$ soit petit. Par conséquent, avec cette factorisation incomplète, $A \approx \bar{L}\bar{L}^T$ Maintenant, en choisissant $C = \bar{L}^T$, il en résulte que $M = \bar{L}\bar{L}^T$ et

$$C^T A C - ^1 = \bar{L}^{-1} A \bar{L}^{-T} \approx I,$$

démontrer que la distribution des valeurs propres de $C^T A C^{-1}$ est favorable. Dans ce cas, la résolution du système $My = r$ se réduit à la résolution de deux systèmes triangulaires [voir :

Axelsson (1994), Golub et Van Loan (1996), Kelley (1995), Nocedal et Wright (2006)].

Pour l'algorithme du gradient conjugué linéaire préconditionné, à partir de (3.5.3), on remarque que

$$J(x_{k+1}^{CG}) = \frac{1}{2}x_k^T A x_k - \frac{1}{2}\frac{\left(R_k^T v_k\right)^2}{d_k^T A d_k},$$

où $v_k = M^{-1}R_k$. Par conséquent, le taux de convergence de l'algorithme de gradient conjugué linéaire préconditionné est également linéaire.

# CHAPITRE 4

# Synthèse des resultats de convergence des différentes méthodes du gradient conjugué

## 4.1 Les méthodes classiques

Les méthodes **FR**, DY et CD ont toutes comme numérateur commun le terme $\|R_{k+1}\|^2$. Une différence fondamentale qui caractérise ces méthodes par rapport aux autres méthodes du gradient conjugué où $\beta_k$ est calculé différement, est que les théorèmes de convergence globale nécessitent seulement la condition de Lipschitz : **Condition** $\mathcal{H}1$ et n'ont pas besoin de la condition de bornétude **Condition** H2 .

### 4.1.1 Méthode de Fletcher-Reeves

Cette méthode été découverte en 1964 par Fletcher et Reeves [38], $\beta_k$ est égale à :

$$\beta_{k+1}^{FR} = \frac{\|R_{k+1}\|^2}{\|R_k\|^2} \tag{4.1.1}$$

Le premier résultat de convergence globale de la méthode de **FR** a été donnée par Zoutendijk [8] en 1970. Il a prouvé que la méthode Fletcher-Reeves converge globalement quand $\alpha_k$ est une recherche linéaire précise . En 1977, Powell a fait remarquer que la méthode de Fletcher-Reeves, avec une recherche linéaire précise, 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 les conditions 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 :

$$\frac{1-2\sigma+\sigma^{k+1}}{1-\sigma} \leq \frac{-R_k^T d_k}{\|R_k\|^2} \leq \frac{1-\sigma^{k+1}}{1-\sigma},$$

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'recherche n'a pas établi la descente suffisante.

Touati Ahmed et Story [44] ont généralisé ce résultat pour

$$0 \leq \beta_k \leq \beta_k^{FR} \tag{4.1.2}$$

Gilbert et Nocedal [10] ont généralisé ce résultat pour

$$|\beta_k| \leq \beta_k^{FR} \tag{4.1.3}$$

Dans Liu et al.[51] , la preuve de la convergence d'Al-Baali [1]est étendue au cas $\sigma = 1/2$. Dai et Yuan [46] ont montré que dans les versions **FR** consécutives, au moins une itération satisfait la propriété de descente suffisante.

**Résultats d'El Baali**

Comme on l'a remarqué auparavant le premier et le plus important résultat de convergence de la méthode Fletcher Reeves (FR), avec une recherche linéaire inexacte est celui d'El Baali [1].

### 4.1.2 Méthode de descente conjuguée

Cette méthode été découverte en 1987 par Fletcher [36], $\beta_k$ est égale à :

$$\beta_{k+1}^{CD} = \frac{\|R_{k+1}\|^2}{-d_k^T R_k} \tag{4.1.4}$$

Avec une recherche linéaire précise, $\beta_k^{FR} = \beta_k^{CD}$ . 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 avec $\sigma_1 \leq 1$ et $\sigma_2 = 0$, il peut être démontré que :

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

D'autre part, si $\sigma_1 \geq 1$ ou $\sigma_2 > 0$, Dai et Yuan [47] 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 minimum.

### 4.1.3 Méthode de Dai-Yuan

Cette méthode été découverte par Y. H. Dai et Y. Yuan [47] en 1999, $\beta_k$ est égale à :

$$\beta_k^{DY} = \frac{\|R_{k+1}\|^2}{d_k^T y_k}; \quad y_k = R_{k+1} - R_k \tag{4.1.5}$$

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 aves des hypothèses très générales sur la fonction objectif $J$. On exige seulement que $J$ soit continueument différentiable et que le gradient soit Lipschitzien, c'est à dire que $J$ vérifie l'hypothèse **Condition** H1e de la méthode de FR dans les applications a été souvent attribuée à ce phénomène.

### 4.1.4 Méthode de Polak-Ribière-Polyak

Cette méthode a été proposée en 1969 par Polak , Ribière et Polyak [34], $\beta_k$ est égale à :

$$\beta_k^{PRP} = \frac{R_{k+1}^T y_k}{\|R_k\|^2}; \quad y_k = R_{k+1} - R_k \tag{4.1.6}$$

La convergence globale de la méthode **PRP** fut établie lorsque $J$ est fortement convexe et la recherche linéaire est précise. Pour une fonction non linéaire, Powell [26] a montré que la méthode **PRP** est globalement convergente si

(a) la taille du pas $s_k = x_{k+1} - x_k$ tend vers zéro,

(b) la recherche en ligne est précise

(c) la condition de Lipshitz $H1$ est vérifiée

D'autre part, Powell a montré plus tard , en utilisant un contre exemple à 3 dimensions, que, avec une recherche linéaire précise, 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 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 a donné un contre exemple qui montre que, même lorsque la fonction objectif est fortement convexe et $\sigma \in (0,1)$ est sufisamment petit, la méthode **PRP** peut encore générer des directions de recherche $d_k$ ascendantes i.e., $R_k^T d_k > 0$.

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 précise. 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 coéfficient $\beta_k$**

Sur la base des connaissances tirées de l'exemple de Powell, Dai a suggéré la modification suivante dans le paramètre de mise à jour pour la méthode **PRP** :

$$\beta_k^{PRP+} = \max\left\{\beta_k^{PRP}, 0\right\}. \tag{4.1.7}$$

Dans [10] Gilbert et Nocedal ont prouvé la convergence de la méthode **PRP +**.

Modification de la méthode PRP en agissant sur le pas de la recherche linéaire $\alpha_k$

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 ont

proposé une nouvelle recherche linéaire de type Armijo de la forme suivante :

$$\alpha_k = \max\left\{\lambda^j \frac{\tau\left|R_k^T d_k\right|}{\|d_k\|^2}\right\}$$

où $j \geq 0$ est le plus petit entier possedant la propriété suivante

$$J(x_{k+1}) \leq J(x_k) - \delta\alpha_k^2 \|d_k\|^2, \tag{4.1.8}$$

et

$$-c_1 \|R_{k+1}\|^2 \leq R_{k+1}^T d_{k+1} \leq c_2 \|R_{k+1}\|^2 \tag{4.1.9}$$

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 , ils combinent la recherche linéaire précédente avec une technique de «région de confiance". Dans une autre voie de recherche, il est montré 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 $J$. Sun et Zhang donnent un résultat de convergence globale avec $\alpha_k = -\delta\frac{R_k^T d_k}{d_k^T Q_k d_k}$ , 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 $\nabla J$. Pour ces choix de pas, les directions de recherche ne sont plus conjugués lorsque $J$ est quadratique. Par conséquent, ces méthodes doivent être considérées comme des méthodes de plus grande pente, plutôt que des méthodes de gradient conjugué.

### 4.1.5 Méthode de Hestenes et Stiefel HS

Cette méthode a été proposée en 1952, dans sa version linéaire par Hestenes et Steifel . C'est d'ailleurs le premier article publié qui parle de la méthode du gradient conjugué. Les auteurs utlisent cette méthode itérative pour résoudre des systèmes linéaires, $\beta_k$ est égale à :

$$\beta_k^{HS} = \frac{R_{k+1}^T y_k}{d_k^T y_k}; \quad y_k = R_{k+1} - R_k \tag{4.1.10}$$

La méthode **HS** possède la propriété de conjugaison suivante

$$d_{k+1}^T y_k = 0 \tag{4.1.11}$$

indépendamment de la recherche linéaire.

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 , la méthode **HS** associée à une recherche linéaire exacte, peut ne pas converger pour une fonction non linéaire.

### 4.1.6 Méthode de Liu et Storey LS

La méthode Liu et Storey LS est également identique à la méthode PRP pour une recherche de ligne précise. Bien que peu de recherches ont été faites sur ce choix pour le paramètre de mise à jour, à l'exception de l'article [51], nous nous attendons à ce que les techniques développées pour la recherche de la méthode PRP devraient s'appliquer à la méthode **LS**.

### 4.1.7 Méthode de RMIL et RMIL+

Rivaie et al. [40] ont proposé un nouveau coefficient CG simple connu sous le nom de RMIL qui vérifie la condition de descente suffisante et possède des propriétés de convergence globale. Le résultat de la convergence globale est établi en utilisant la recherche linéaire exacte.

Cependant, le résultat serait différent si nous utilisons la recherche linéaire inexacte.

Cette méthode est notée comme :

$$\beta_k^{RMIL} = \frac{R_k^T \left(R_k - R_{k-1}\right)}{\|d_{k-1}\|^2}$$

Le nouveau $\beta_k$, qui est une extension du $\beta_k^{RMIL}$ et que nous avons nommé $\beta_k^{RMIL+}$conserve le dénominateur de $\beta_k^{RMIL}$ et ajoute une autre direction de recherche précédente négative au numérateur. D'où,

$$\beta_k^{RMIL+} = \frac{R_k^T \left(y_{k-1} - d_{k-1}\right)}{\|d_{k-1}\|^2}$$

Cette dernière méthode possédera les propriétés de convergence globale basées sur la recherche linéaire exacte et inexacte.

### 4.1.8 Les Méthodes hybrides

#### Méthodes hybrides utilisant FR et PRP

Comme nous l'avons vu, les méthodes **FR**, **DY** et **CD** ont des propriétés de convergence remarquables, mais ils ne sont pas assez performantes 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 [44] 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 coincent ou bloquent, le paramètre de mise à jour PRP est utilisé. Par les mêmes motivations, Hu et Storey ont proposé

$$\beta_k = \max\left\{0, \min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$$

Dans [10], il est précisé que $\beta_k^{PRP}$ peut être négatif, même pour des fonctions fortement convexes. Nocedal et Gilbert ont proposé

$$\beta_k = \max\left\{-\beta_k^{FR}, \min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$$

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 présentées, les performances de ce procédé hybride n'étaientt pas meilleure que celles de PRP+.

En particulier, le résultat suivant est établi dans [37] :

**Théorème 4.1.1** *Considérons une méthode itérative du gradient conjugué de coéfficient $\beta_k$ de la forme (4.1.10) et (4.1.11), qui utilise une recherche linéaire inexacte de Wolfe forte (1.4.24) , (1.4.25) avec $\sigma \leq 1/2$. Supposons aussi que l'hypothèse de Lipschitz H1 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.*

$$\liminf_{k\to\infty} \|R_k\| = 0$$

### 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 ont étudié la possibilité de combiner DY avec d'autres méthodes CG. Utilisant une recherche linéaire de Wolfe et pour $\beta_k \in \left[-\eta\beta_k^{DY}, \beta_k^{DY}\right]$, où $\eta = (1-\sigma)(1+\sigma)$, ils démontrent la convergence globale lorsque la condition de Lipschitz $H1$ est assurée.

Les deux méthodes hybrides suivantes ont été proposés comme suit :

$$\beta_k = \max\left\{-\left(\frac{1-\sigma}{1+\sigma}\right)\beta_k^{DY}, \min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$$

et

$$\beta_k = \max\left\{0, \min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$$

Les expériences numériques indiquent que la seconde méthode hybride a donné des résultats plus performants que la méthode PRP +.

Une autre méthode hybride a été proposée par Dai. Elle emploie soit le modèle de DY ou celui de CD :

$$\beta_k = \frac{\|R_{k+1}\|^2}{\max\left\{d_k^T y_k, -R_k^T d_k\right\}}$$

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, $\beta_k \in \left[0, \beta_k^{DY}\right]$. Cette propriété, ainsi que celle de la descente du modèle hybride, implique la convergence des méthodes pour des recherche linéaires typiques.

Récemment, de nombreux auteurs algériens ont étudié la combinaison convexe de deux méthodes ou plus. Parmi eux, nous mentionnons A.Hallal [14], [13], S.Delladji [7], [6], S.Hanachi [16], A.Hamdi [15], N.Chenna [30] et I.Guefassa [12]. Alors que Chaib et Bechouat [3]ont introduit deux méthodes modifiées appelées MCB1 et MCB2.

# CHAPITRE 5

# La nouvelle méthode d'hybridation du gradient conjugué CGHLB

## 5.1 Principe de la nouvelle méthode d'hybridation

Des méthodes de gradient conjugué sont connues et un excellent aperçu d'entre elles est donné par Hager et Zhang [10].Le scalaire $\beta_k$ est choisi de telle sorte que la méthode décrite par (1.0.2), (1.0.5) soit réduite à un gradient conjugué linéaire lorsque J est quadratique convexe d'où une recherche linéaire précise sera utilisée.

Pour les fonctions générales, une formule différente pour le scalaire $\beta_k$donne des méthodes distinctes de gradient conjugué non linéaire ([39], [24], [35], [5], [4], [8]).

La formule de 1964 de Fletcher-Reeves est généralement considérée comme le premier algorithme non linéaire depuis leur article [9] qui se concentre sur l'optimisation non linéaire, tandis que l'article de 1952 [5] de Hestenes-Stiefel se concentre sur les systèmes linéaires symétriques, définis positifs.

Certaines de ces méthodes, telles que Fletcher et Reeves (FR) [9], Dai etYuan (DY) [24] et la descente conjuguée (CD) [9] ont de fortes propriétés de convergence,mais ils peuvent avoir des performances pratiques modestes en raison du brouillage .D'autre part , les méthodes de Polak et Ribière [39] et Polyak (PRP) [4], Hestenes et Stiefel (HS) [5] ou Liu et Story (LS) [51] peuvent ne pas être généralement convergents, mais ils ont souvent de meilleures performances

de calcul.

Dans le processus d'obtention de méthodes de gradient conjugué plus robustes et efficaces, certains chercheurs ont suggéré l'algorithme de gradient conjugué hybride qui combinait les bonnes caractéristiques des méthodes impliquées dans l'hybridation.

La première méthode de gradient conjugué hybride a été donnée par Touati-Ahmed et Story(1990) [44] pour éviter le phénomène de brouillage.

Les chercheurs ont été motivés par les travaux d'Andrei [1], [21] ; Dai et Yuan [23];.Zhang et Zhou [29] et leur paramètre $\beta_k$ est calculé comme une combinaison convexe de$\beta_k^{FR}$ et $\beta_k^*$ autre algorithmes, c'est-à-dire

$$\beta_k^N = (1-\theta_k)\,\beta_k^{FR} + \theta_k \beta_k^*$$

La recherche linéaire de Wolfe a été utilisée pour déterminer le pas $\alpha_k > 0$ et la nouvelle méthode s'est avérée plus robuste numériquement par rapport à la FR et à d'autres méthodes. La convergence globale a été établie dans certaines conditions appropriées.

Rappelons qu'Andrei dans [1] a proposé un nouvel algorithme de gradient conjugué hybride où le paramètre $\beta_k$ est calculé comme une combinaison convexe des algorithmes de gradient conjugué Polak - Ribière - Polyak et Dai - Yuan, c'est-à-dire

$$\beta_k^N = (1-\theta_k)\,\beta_k^{PRP} + \theta_k \beta_k^{DY}$$

et $\theta_k$ est présenté pour satisfaire la condition de conjugaison

$$\theta_k = \theta_k^{CCOMB} = \frac{(y_k^t R_{k+1})\,(y_k^t s_k) - (y_k^t R_{k+1})\,(R_k^t R_k)}{(y_k^t R_{k+1})\,(y_k^t s_k) - \|R_{k+1}\|^2 \|R_k\|^2}$$

ou $s_k = x_{k+1} - x_k$,et pour satisfaire la direction Newton ; il prend

$$\theta_k = \theta_k^{NDOMB} = \frac{(y_k^t R_{k+1} - s_k^t R_{k+1})\,\|R_k\|^2 - (y_k^t R_{k+1})\,(y_k^t s_k)}{\|R_{k+1}\|^2 \|R_k\|^2 - (y_k^t R_{k+1})\,(y_k^t s_k)}$$

mais dans la combinaison de HS et DY de la direction de Newton, il a mis

$$\theta_k = \frac{-s_k^t R_{k+1}}{R_k^t R_{k+1}}$$

D'autre part, à partir de la direction de Newton avec condition sécante modifiée (hybride M-Andrei), Andrei a proposé une autre méthode

$$\beta_k^{HYBRIDM} = (1-\theta_k)\,\beta_k^{HS} + \theta_k \beta_k^{DY}$$

ou

$$\theta_k = \frac{\left(\frac{\delta\eta_k}{s_k^t s_k} - 1\right) s_k^t R_{k+1} - \frac{y_k^t R_{k+1}}{y_k^t s_k}\delta\eta_k}{R_k^t R_{k+1} + \frac{R_k^t R_{k+1}}{y_k^t s_k}\delta\eta_k}$$

$\delta$ est un paramètre .Dans [27] Salah Gazi Shareef et Hussein Ageel Khatab ont introduit un nouvel hybride CG

$$\beta_k^{New} = (1-\theta_k)\,\beta_k^{PRP} + \theta_k \beta_k^{BA}$$

ou $\beta_k^{BA}$ est sélectionné dans [20].

Dans cette section on présente une nouvelle méthode hybride pour résoudre un probléme d'optimisation non linéaire sans contraintes en utilisant la méthode du gradient conjugué, qui est une combinaison convexe entre la méthode du gradient conjugué dePolak-Ribière-Polyak(PRP) et la méthode du gradient conjugué de Rivaie-Mustafa-Ismail-Leong(RMIL+).

Cette méthode posséde la propriété de descente suffisante et la convergence globale avec la recherche linéaire satisfait les conditions de Wolfe forte.

Considérons le probléme de minimisation d'une fonction $J : \mathbb{R}^n \to \mathbb{R}$, continûment différentiable et non linéaire :

$$\min\{J(x), x \in \mathbb{R}^n\}. \tag{5.1.1}$$

Pour résoudre ce problème on utilise une suite $\{x_k\}$ définie par :

$$x_{k+1} = x_k + s_k \quad \text{où } s_k = \lambda_k d_k, \;\; k = 0, 1, ..., n. \tag{5.1.2}$$

Où $\lambda_k > 0$ est le pas de la recherche linéaire, satisfait les conditions de Wolfe forte :

$$J(x_k + \lambda_k d_k) - J(x_k) \leq \delta\lambda_k \nabla^T J(x_k)\, d_k, \tag{5.1.3}$$

$$\left|\nabla^T J(x_k + \lambda d_k).d_k\right| \leq -\sigma \nabla^T J(x_k).d_k, \tag{5.1.4}$$

où $0 < \delta < \sigma < 1$.

Et $d_k$ la direction de descente définie par la formule de récurrence suivante ;

$$d_k = \begin{cases} -R_K & \text{si } k = 0, \\ -R_k + \beta_{k-1} d_{k-1} & \text{si } k \geq 1, \end{cases} \tag{5.1.5}$$

avec $R_k = \nabla J(x_k)$, et $\beta_k$ est le paramètre du gradient conjugué, qui détermine les différentes méthodes du gradient conjugué (donnée par les formules (3.4.4)(3-18) jusqu'à (3-24)).

L'objectif de cette nouvelle méthode est de trouver une combinaison convexe entre la méthode dePRP et la méthode de RMIL+, cette combainaison nous permet de trouver une meilleure descente et une meilleure convergence vers la solution optimal.

## 5.2 Combinaison convexe

On traitant la combinaison convexe des paramétres du gradient conjugué des méthodes PRP et RMIL+.

On définie $\beta_k^{HLB}$ par :

$$\beta_k^{HLB} = \theta_k \beta_k^{PRP} + (1-\theta_k)\beta_k^{RMIL+}. \tag{5.2.1}$$

Où $\theta_k \in [0,1]$ est appelé le paramétre d'hybridation.

Si $\theta_k = 0$ alors

$$\beta_k^{HLB} = \beta_k^{RMIL+}.$$

Si $\theta_k = 1$ alors

$$\beta_k^{HLB} = \beta_k^{PRP}.$$

Si $0 < \theta_k < 1$ alors $\beta_k^{HLB}$ est une combinaison de $\beta_k^{PRP}$ et $\beta_k^{RMIL+}$.

La direction de descente associée à cette méthode est donnée par :

$$d_k^{HLB} = \begin{cases} -R_0 \text{ si } k = 0 \\ -R_k + \beta_{k-1}^{HLB} d_{k-1} \text{ si } k \geq 1 \end{cases} \tag{5.2.2}$$

**Théorème 5.2.1** *Si les relations* (5.2.1) *et* (5.2.2) *sont valables, alors*

$$d_{k+1}^{HLB} = \theta_k d_{k+1}^{PRP} + (1-\theta_k) d_{k+1}^{RMIL+}. \tag{5.2.3}$$

**Preuve.** Multiplier (5.2.3) par $y_k^T$ et par l'utilisation de condition de conjugaison $y_k^T d_{k+1}^{HLB} = 0$, on trouve :

$$0 = -R_{k+1}^t y_k + (1-\theta_k)\frac{R_{k+1}^t y_k}{\|R_k\|^2} d_k^t y_k + \theta_k \frac{R_{k+1}^t (R_{k+1} - R_k - d_k)}{\|d_k\|^2} d_k^t y_k$$

alors

$$\theta_k = \frac{R_{k+1}^t y_k \|R_k\|^2 \|d_k\|^2 - \left(R_{k+1}^t y_k\right)\left(d_k^t y_k\right)\|d_k\|^2}{\left(\left(R_{k+1}^t (y_k - d_k)\right)\|R_k\|^2 - \left(R_{k+1}^t y_k\right)\|d_k\|^2\right)\left(d_k^t y_k\right)}$$

On peut fixer $\beta_k^{HLB}$ comme suit :

$$\beta_k^{HLB} = \begin{cases} (1-\theta_k)\,\beta_k^{PRP} + \theta_k \beta_k^{RMIL+} & \text{if } 0 < \theta_k < 1 \\ \beta_k^{PRP} & \text{if } \theta_k \leq 0 \\ \beta_k^{RMIL+} & \text{if } \theta_k \geq 1 \end{cases} \tag{5.2.4}$$

■

## 5.3 Algorithme de la méthode CGHLB

**Etape 0 :** Sélectionnez $x_0 \in \mathbb{R}^n, \epsilon > 0$, et $0 < \delta < \sigma < 1$.

Calculer $J(x_0)$, et $R_0$. Considére $d_0 = -R_0$.

**Etape 1 :** Si $||\mathbf{R}_k|| < \epsilon$, alors STOP.

**Etape 2 :** Calculer $\lambda_k > 0$ satisfaisant aux conditions de Wolfe forte.

Calculer $x_{k+1}, J_{k+1}, R_{k+1}, y_k$.

**Etape 3 :** Si$\left(\left(R_{k+1}^t\left(y_k - d_k\right)\right)\|R_k\|^2 - \left(R_{k+1}^t y_k\right)\|d_k\|^2\right)\left(d_k^t y_k\right) = 0$ , alors $\theta_k = 0$, sinon $\theta_k$ comme dans (5.2.4).

**Etape 4 :** Calculer $\beta_k^{HLB}$ comme dans (5.2.1).

**Etape 5 :** calculer $d_{k+1}^{HLB} = -R_{k+1} + \beta_k^{HLB} d_k^{HLB}$.

**Etape 6 :** Si le critère de redémarrage de Powell $|R_{k+1}^T R_k| \geq 0.2||R_{k+1}||^2$. est satisfait,

alors $d_{k+1} = -R_{k+1}$,

sinon définir $d_{k+1} = d$.

**Etape 7 :** Poser $k = k+1$, et aller à l'étape2.

## 5.4 Propriété de descente suffisante et la convergence globale

Les théorèmes mentionnés ci-dessous affirment que la méthode CGHLB satisfait à la condition de descente sufisante, où on distingue trois cas :

★**Premièrement**

Poser $\theta_k = 0$, alors

$$d_{k+1}^{HLB} = d_{k+1}^{RMIL+} = -R_{k+1} + \frac{R_k^T\left(y_{k-1} - d_{k-1}\right)}{\|d_{k-1}\|^2} d_k.$$

★**Deuxièmement**

Pour $\theta_k = 1$, alors

$$d_{k+1}^{HLB} = d_{k+1}^{PRP} = -R_{k+1} + \frac{R_{k+1}^T y_k}{\|R_k\|^2} d_k.$$

★**Finalement**

Si $0 < \theta_k < 1$, il existe deux nombres réels $\mu_1$ et $\mu_2$ tels que $0 < \mu_1 \leq \theta_k \leq \mu_2 < 1$.

Alors le théoreme suivant montre que notre methode assure la condition de descente lorsque $0 < \theta_k < 1$

**Théorème 5.4.1** [1], [27], [26] *l'algorithme* (1.0.2) , (1.0.5) *et* (5.2.4)*supposons que* $d_k$ *est une direction de descente*$(g_k^t d_k < 0)$*, et* $\alpha_k$ *est determiné par une recherche linéaire de Wolfe forte* (1.4.24) ; (1.4.25)*. si* $0 < \theta_k < 1$*alors la direction* $d_{k+1}$*donnée par* (5.2.3)*est une direction de descente suffisante.*

**Preuve.** Multipliez les deux cotés par (5.2.3) $R_{k+1}$ nous obtenons :

$$R_{k+1}^T d_{k+1} = -\|R_{k+1}\|^2 + (1-\theta_k)\frac{R_{k+1}^t y_k}{\|R_k\|^2} d_k^t R_{k+1} + \theta_k \frac{R_{k+1}^t (R_{k+1} - R_k - d_k)}{\|d_k\|^2} d_k^t g_{k+1}$$

$$\begin{aligned} R_{k+1}^T d_{k+1} &= -(1-\theta_k+\theta_k)\|R_{k+1}\|^2 \\ &\quad + (1-\theta_k)\frac{R_{k+1}^t y_k}{\|R_k\|^2} d_k^t R_{k+1} + \theta_k \frac{R_{k+1}^t (R_{k+1} - R_k - d_k)}{\|d_k\|^2} d_k^t R_{k+1} \end{aligned}$$

$$\begin{aligned} R_{k+1}^T d_{k+1} &= \left[-(1-\theta_k)\|R_{k+1}\|^2 + (1-\theta_k)\frac{R_{k+1}^t y_k}{\|R_k\|^2} d_k^t R_{k+1}\right] \\ &\quad + \left[-(\theta_k)\|R_{k+1}\|^2 + \theta_k \frac{R_{k+1}^t (R_{k+1} - R_k - d_k)}{\|d_k\|^2} d_k^t R_{k+1}\right] \end{aligned}$$

$$\begin{aligned} R_{k+1}^T d_{k+1} &= (1-\theta_k)\left[-\|R_{k+1}\|^2 + \frac{R_{k+1}^t y_k}{\|R_k\|^2} d_k^t R_{k+1}\right] \\ &\quad + (\theta_k)\left[-\|R_{k+1}\|^2 + \frac{R_{k+1}^t (R_{k+1} - R_k - d_k)}{\|d_k\|^2} d_k^t R_{k+1}\right] \end{aligned}$$

et comme $0 < \theta_k < 1$ alors :

$$\begin{aligned} R_{k+1}^T d_{k+1} &\leq \left[-\|R_{k+1}\|^2 + \frac{R_{k+1}^t y_k}{\|R_k\|^2} d_k^t R_{k+1}\right] && (5.4.1) \\ &\quad + \left[-\|R_{k+1}\|^2 + \frac{R_{k+1}^t (R_{k+1} - R_k - d_k)}{\|d_k\|^2} d_k^t R_{k+1}\right] && (5.4.2) \end{aligned}$$

La preuve est complète. si le pas $\alpha_k$ est choisi par une recherche linéaire exacte qui necessite $R_{k+1}^T d_k = 0$.

Si $\alpha_k$ est que la recherche linéaire soit inexacte $\left(R_{k+1}^T d_k \neq 0\right)$ donc on obtient :

$$R_{k+1}^T d_{k+1} \leq \left[-c_1 \left\|R_{k+1}\right\|^2\right] + \left[-c_2 \left\|R_{k+1}\right\|^2\right]$$

car les algorithmes de $(PRP)$ et $(RMIL+)$ obtiennent la propriété de la descente suffisente. ■

**Remarque 5.4.1** ***Preuve.*** *Finalement nous obtenons :*

$$R_{k+1}^T d_{k+1} \leq -c \left[\left\|R_{k+1}\right\|^2\right]$$

*ou* $c = c_1 + c_2$ *;et* $c_1 > 0$ *;* $c_2 > 0$ . ■

**★Hypothèse 1**

On dit que $J$ est borné sur l'ensemble

$$S = \{x \in \mathbb{R}^n : J(x) \leq J(x_0) : x_0 \in \mathbb{R}^n \text{ point initial}\},$$

c'est à dire qu'il existe une constante $B > 0$ telle que

$$||x|| \leq B : \forall x \in S. \tag{5.4.3}$$

**★Hypothèse 2**

Le gradient $\nabla J(x)$ vérifie la condition de Lipschitz, c'est à dire, il existe une constante $L > 0$ telle que

$$||\nabla J(x) - \nabla J(y)|| \leq L||x - y|| : \text{ pour tout } x, y \in \mathbb{R}^n. \tag{5.4.4}$$

Les hypothèses (1) et (2) impliquent qu'il existe une constante positive $\gamma$ telle que :

$$||R(x)|| \leq \gamma, \ \forall x \in \mathbb{R}^n. \tag{5.4.5}$$

**Lemme 5.4.1** *supposons que les hypothèses*1 *et* 2 *satisfaites.*

*Considérons une méthode de la forme* (5.2.1)*, où* $d_k$ *est une direction de descente et* $\lambda_k$ *satisfait les conditions de wolfe forte* (5.1.3) *et* (5.1.4)*.*

*Si on a :* $\sum_{k\geq 0} \frac{1}{||d_k||^2} = +\infty$.

*Alors* $\liminf_{k\to\infty} \|R_k\| = 0$

**Lemme 5.4.2** *Supposons que $d_k$ est une direction de descente et $\nabla J$ satisfait les condition de Lipschitz*

$$||\nabla J(x) - \nabla J(x_k)|| \leq L||x - x_k||.$$

*quelque soit $x$ dans le segment de ligne reliant $x_k$ et $x_{k+1}$, ou $L$ est un constant.*

**Lemme 5.4.3** *Si la recherche linéaire satisfait les condition de Wolfe, alors*

$$\lambda_k \geq \frac{1-\sigma}{L}\frac{|R_k^T d_k|}{||d_k||^2}. \tag{5.4.6}$$

**Preuve.** Il s'agit de la $2^{eme}$ condition de Wolfe forte

$$\left|R_{k+1}^T.d_k\right| \leq -\sigma R_k^T d_k, \tag{5.4.7}$$

et la condition Lipschitzienne

$$||\nabla J(x) - \nabla J(y)|| \leq L||x - y|| : \text{ pour tout } x, y \in \mathbb{R}^n, \tag{5.4.8}$$

et l'inégalité de Cauchy-Bankovsky-Schwaertz

$$\sum_{k=1}^{n} |v_k u_k| \leq \left(\sum_{k=1}^{n} |u^k|^2\right)^{\frac{1}{2}} \left(\sum_{k=1}^{n} |v^k|^2\right)^{\frac{1}{2}}.$$

On trouve

$$\begin{aligned}
\sigma R_k^T d_k &\leq R_{k+1}^T.d_k \leq -\sigma R_k^T d_k \qquad (5.4.9)\\
\sigma R_k^T d_k - R_k^T.d_k &\leq R_{k+1}^T.d_k - R_k^T.d_k \leq -\sigma R_k^T d_k - R_k^T.d_k\\
(\sigma-1)R_k^T.d_k &\leq (R_{k+1}^T - R_k^T).d_k \leq -\sigma R_k^T d_k - R_k^T.d_k\\
(\sigma-1)R_k^T.d_k &\leq y_k^T.d_k
\end{aligned}$$

D'autre part

$$\begin{aligned}
|y_k^T.d_k| &\leq ||y_k||.||d_k||\\
&\leq ||R_{k+1} - R_k||.||d_k||\\
&\leq L||x_{k+1} - x_k||.||d_k||\\
&\leq L.s_k.||d_k||\\
&\leq L.\lambda_k.||d_k||^2
\end{aligned}$$

Donc

$$\begin{aligned}(\sigma-1)R_k^T.d_k &\leq L.\lambda_k.||d_k||^2\\ \frac{(\sigma-1)R_k^T.d_k}{L||d_k||^2} &\leq \lambda_k\\ \lambda_k &\geq \frac{(\sigma-1)R_k^T.d_k}{L||d_k||^2}\\ \lambda_k &\geq \frac{1-\sigma}{L}\frac{|R_k^T d_k|}{||d_k||^2}\end{aligned}$$

■

**Lemme 5.4.4** *supposons que les hypothèses (1) et (2) satisfait.*

*Considérons une méthode de la forme (5.2.1), où $d_k$ est une direction de descente et $\lambda_k$ satisfait les conditions de wolfe forte (5.1.3) et (5.1.4).*

*Alors on a*

$$\sum_{k>0}\frac{(R_k^T d_k)^2}{||d_k||^2} < +\infty.$$

**Lemme 5.4.5** *Supposons que $d_k$ est une direction de descente et $\nabla J$ satisfait les condition de Lipschitz*

$$||\nabla J(x) - \nabla J(x_k)|| \leq L||x - x_k||.$$

*quelque soit $x$ dans le segment de ligne reliant $x_k$ et $x_{k+1}$, ou $L$ est un constant.*

*Si la recherche linéaire satisfait les condition de Wolfe, alors*

$$\lambda_k \geq \frac{1-\sigma}{L}\frac{|R_k^T d_k|}{||d_k||^2}. \tag{5.4.10}$$

**Remarque 5.4.2** ***Preuve.*** *Il s'agit de la $2^{eme}$ condition de Wolfe forte (5.1.4)*

$$\left|R_{k+1}^T.d_k\right| \leq -\sigma R_k^T d_k,$$

*et la condition Lipschitzienne*

$$||\nabla J(x) - \nabla J(y)|| \leq L||x-y|| : \ \textit{pour tout } x, y \in \mathbb{R}^n,$$

*et l'inégalité de Cauchy-Bankovsky-Schwaertz*

$$\sum_{k=1}^{n}|v_k u_k| \leq \left(\sum_{k=1}^{n}|u^k|^2\right)^{\frac{1}{2}}\left(\sum_{k=1}^{n}|v^k|^2\right)^{\frac{1}{2}}.$$

*On trouve*

$$\begin{aligned}
\sigma R_k^T d_k &\leq R_{k+1}^T.d_k \leq -\sigma R_k^T d_k \qquad (*)\\
\sigma R_k^T d_k - R_k^T.d_k &\leq R_{k+1}^T.d_k - R_k^T.d_k \leq -\sigma R_k^T d_k - R_k^T.d_k\\
(\sigma-1)R_k^T.d_k &\leq (R_{k+1}^T - R_k^T).d_k \leq -\sigma R_k^T d_k - R_k^T.d_k\\
(\sigma-1)R_k^T.d_k &\leq y_k^T.d_k
\end{aligned}$$

*D'autre part*

$$\begin{aligned}
|y_k^T.d_k| &\leq ||y_k||.||d_k||\\
&\leq ||R_{k+1} - R_k||.||d_k||\\
&\leq L||x_{k+1} - x_k||.||d_k||\\
&\leq L.s_k.||d_k||\\
&\leq L.\lambda_k.||d_k||^2 \qquad (5.4.11)
\end{aligned}$$

*Donc*

$$\begin{aligned}
(\sigma-1)R_k^T.d_k &\leq L.\lambda_k.||d_k||^2\\
\frac{(\sigma-1)R_k^T.d_k}{L||d_k||^2} &\leq \lambda_k\\
\lambda_k &\geq \frac{(\sigma-1)R_k^T.d_k}{L||d_k||^2}\\
\lambda_k &\geq \frac{1-\sigma}{L}\frac{|R_k^T d_k|}{||d_k||^2}
\end{aligned}$$

■

En utilisant le lemme ci-dessus, le résultat suivant peut être prouvé.

Supposant que la fonction $J$ est uniformément convexe i.e. qu'ilexiste une constante$\Gamma \geq 0$ sachant que :

pour tout $x, y \in \Omega$

$$(R(x) - R(y))^t (x - y) \geq \Gamma \|x - y\|^2$$

et le pas $\alpha_k$ est obtenu par la recherche linéaire de Wolfe forte.

Par la convexité uniforme de la fonction qui satisfait les hypothèses précédentes, on peut prouver que la norme de $d_{k+1}$est inférieurement bornée.

En utilisant le lemme précédent, on obtient le théorème suivant :

**Théorème 5.4.2** *[1],[27],[26]Supposons quel'hypothèse*1 *soit vraie et considérons les algorithmes*; $(2,3)$ *et* $(2,5)$ *,ou* $0 \leq \theta_k \leq 1$ *et* $\alpha_k$ *est obtenu par la recherche linéaire de Wolfe forte .*(1.4.24) *et* (1.4.25) *.*
*Si* $d_k$ *tend vers zéro et il existe des constantes non négatives* $\eta_1$ *et* $\eta_2$ *telle que ;*

$$\|R_k\|^2 \geq \eta_1 \|s_k\|^2 \, et \, \|R_{k+1}\|^2 \leq \eta_2 \|s_k\| \tag{5.4.12}$$

*et* $f$ *est une fonction uniformément convexe, alors ;*

$$\lim_{k\to\infty} R_k = 0 \tag{5.4.13}$$

**Preuve.** De (5.4.11) il s'ensuit que

$$y_k^t s_k \geq \Gamma \|s_k\|^2$$

comme $0 \leq \theta_k \leq 1$ ,de convexité uniforme et (**??**) on obtient

$$\begin{aligned}
\left|\beta_k^{HLB}\right| &\leq \left|\frac{R_{k+1}^t y_k}{\|R_k\|^2}\right| + \left|\frac{R_{k+1}^t (R_{k+1} - R_k - d_k)}{\|d_k\|^2}\right| \\
&\leq \frac{\left|R_{k+1}^t y_k\right|}{\|R_k\|^2} + \frac{\left|R_{k+1}^t y_k\right|}{\|d_k\|^2} + \frac{\left|R_{k+1}^t d_k\right|}{\|d_k\|^2} \\
&\leq \frac{\|R_{k+1}\| \|y_k\|}{\|R_k\|^2} + \frac{\|R_{k+1}\| \|y_k\|}{\|d_k\|^2} + \frac{\|R_{k+1}\| \|d_k\|}{\|d_k\|^2}
\end{aligned}$$

de la condition de Lipschitz

$$\begin{aligned}
\|y_k\| &\leq l \|s_k\| \\
\left|\beta_k^{HLB}\right| &\leq \frac{\|R_{k+1}\| \|y_k\|}{\eta_1 \|s_k\|^2} + \frac{\|R_{k+1}\| \|y_k\|}{\|d_k\|^2} + \frac{\|R_{k+1}\|}{\|d_k\|} \\
&\leq \frac{\mu l \|s_k\|}{\eta_1 \|s_k\|^2} + \frac{\mu l \|s_k\| \alpha_k^2}{\|s_k\|^2} + \frac{\mu \alpha_k}{\|s_k\|} \\
&= \frac{\mu l}{\eta_1 \|s_k\|} + \frac{\mu l \alpha_k^2}{\|s_k\|} + \frac{\mu \alpha_k}{\|s_k\|}
\end{aligned}$$

D'où

$$\|d_{k+1}\| \leq \|R_{k+1}\| + \left|\beta_k^{HLB}\right| \|d_k\|$$

$$\leq \mu + \frac{\mu l \|s_k\|}{\eta_1 \alpha_k \|s_k\|} + \frac{\mu l \|s_k\| \alpha_k^2}{\alpha_k \|s_k\|} + \frac{\mu \alpha_k \|s_k\|}{\alpha_k \|s_k\|}$$

$$= 2\mu + \mu l \alpha_k + \frac{\mu l}{\eta_1 \alpha_k}$$

ce qui implique que (5,4,6) est vrai.Par conséquent, par lemme 1, nous avons (5,4,7), qui pour les fonctions uniformément convexes équivaut à $(5, 4, 15)$ ■

## 5.5 Resultats Numériques et Discussion

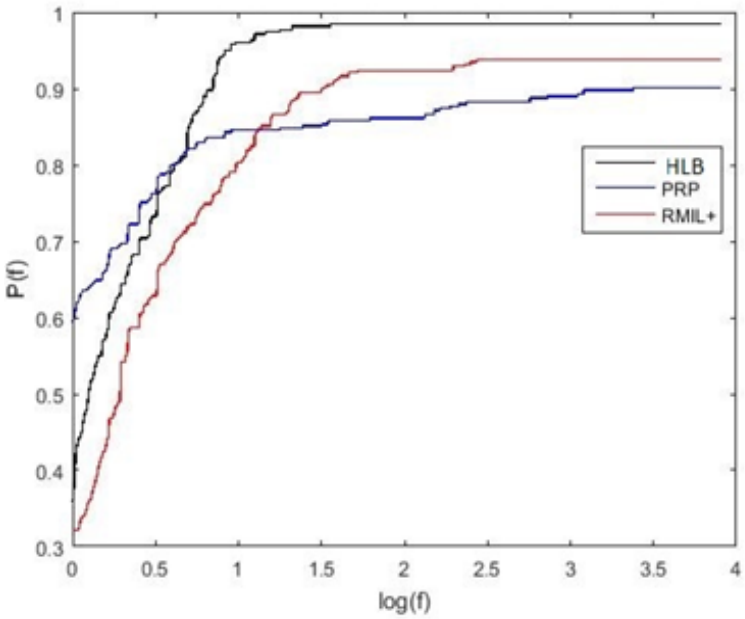
Dans les présentes expériences numériques, nous analysons l'efficacité de $CGHLB$ par rapport aux méthodes classiques : $PRP$ et $RMIL+$.Ces comparaisons sont basés sur le nombre d'itérations et le temps CPU par seconde à atteindre

l'optimum. Toutes les comparaisons sont effectuées avec deux ou trois points initiaux différents et un nombre différent de variables allant de 2 à 20000 . Tous les variables ont été expérimentées à chaque test de fonction[1] . Pour les tests numériques, les paramètres de recherche linéaires de Wolfe fortes ont été expérimentalement fixé à $\rho = 10^{-3}$ et $\delta = 10^{-4}$. Nous arrêtons l'itération si l'inégalité ;

$$||R(x_k)||_\infty \leq \varepsilon,$$

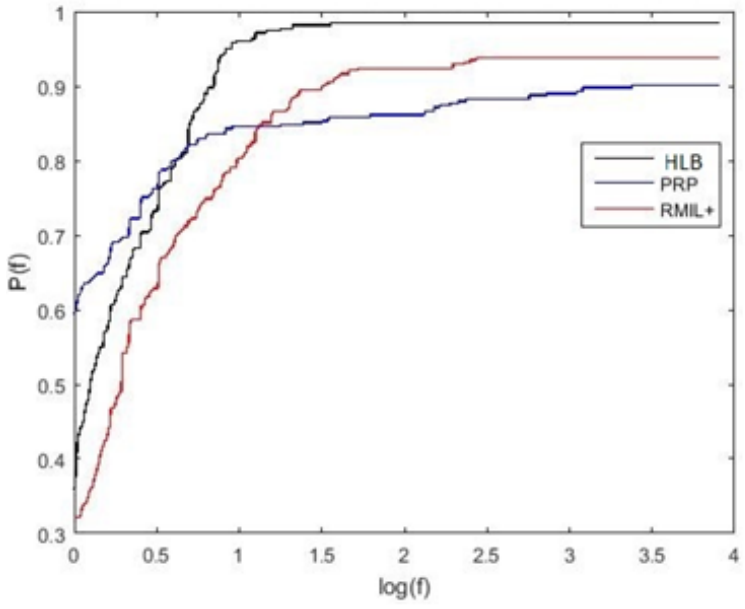
avec $\varepsilon = 10^{-6}$ est satisfaite . Lorsque le nombre d'itérations dépasse 2000 ou le temps d'exécution du PROCESSEUR a dépassé 500 secondes, le test est considéré comme ayant échoué.Les figures (Fig5.1) et (Fig5.2) permettent de conclure que l'algorithme CGHLB est plus efficace que les méthode PRP et RMIL+.avec la plus courte durée de temps CPU. Le pourcentage le plus élevé de comparaison réussie est avec $CGHLB$ à $98,34\%$, suivi de $RMIL+$ avec$93,72\%$.Cependant, la comparaison réussie des taux pour $PRP$ est faible à $90,05\%$.Par conséquent, notre méthode ($CGHLB$) résout avec succès les problèmes testés, et elle est compétitive avec les méthodes du gradient conjugué bien connu d'optimisation sans contraintes

Les profils de performance ont montré que la nouvelle méthode hybride est la meilleur en termes




5.png

FIG. 5.1 – Profile de Performance basé sur le temps CPU




5.png

FIG. 5.2 – Profile de Performance basé sur le nombre d'iterations

d'efficacité et de robustesse.

**Tabel 1**. A list of test problems.

| No. | Function | Dimension | Initial points |
|---|---|---|---|
| 1 | Diagonal 1 | $2, 5, 10, 20, 50$ | $(-1, ..., -1)\,;(0, ..., 0)\,;(1, ..., 1)$ |
| 2 | Diagonal 2 | $2, 10, 50, 100, 200$ | $(-4, ..., -4)\,;(0, ..., 0)\,;(-4, ..., -4)$ |
| 3 | Diagonal 4 | $100, 500, 1000, 10000, 20000$ | $(-4, ..., -4)\,;(1, ..., 1)\,;(10, ..., 10)$ |
| 4 | Penalty | $2, 10, 100, 500, 1000$ | $(-4, ..., -4)\,;(0, ..., 0)\,;(-4, ..., -4)$ |
| 5 | Himmelbleu | $100, 500, 5000, 10000, 20000$ | $(0, ..., 0)\,;(2, ..., 2)\,;(5, ..., 5)$ |
| 6 | Exponential | $2, 4, 6, 8, 10, 12, 14, 16, 18, 20$ | $(-1, ..., -1)\,;(1, ..., 1)$ |
| 7 | Rosenbrock | $100, 1000, 4000, 10000, 20000$ | $(0, ..., 0)\,;(5, ..., 5)\,;(10, ..., 10)$ |
| 8 | Hager | $2, 10, 200, 500, 1000$ | $(0, ..., 0)\,;(2, ..., 2)\,;(4, ..., 4)$ |
| 9 | Perquadratic | $2, 10, 100, 250, 500$ | $(-4, ..., -4)\,;(1, ..., 1)\,;(5, ..., 5)$ |
| 10 | Raydan 2 | $1000, 5000, 10000, 15000, 20000$ | $(-2, ..., -2)\,;(1, ..., 1)\,;(2, ..., 2)$ |
| 11 | Sumsquares | $100, 500, 1000, 10000, 20000$ | $(-4, ..., -4)\,;(1, ..., 1)\,;(4, ..., 4)$ |
| 12 | Power | $2, 4, 8, 10, 15, 20, 25$ | $(1, ..., 1)\,;(2, ..., 2)$ |
| 13 | Qing | $2, 10, 50, 100, 200$ | $(-2, ..., -2)\,;(1, ..., 1)\,;(2, ..., 2)$ |
| 14 | Quadratic | $2, 10, 100, 250, 500$ | $(-4, ..., -4)\,;(1, ..., 1)\,;(5, ..., 5)$ |
| 15 | Rastrinig | $2, 10, 20, 50, 100, 150, 200$ | $(-4, ..., -4)\,;(4, ..., 4)$ |
| 16 | Sphere | $2, 10, 20, 50, 100, 150, 200$ | $(-4, ..., -4)\,;(4, ..., 4)$ |

De nombreuses études ont été consacrées au développement et à l'amélioration des méthodes de gradient conjugué hybride. Dans cette thèse, nous avons présenté une nouvelle hybridation convexe des algorithmes de gradient conjugué $PRP$ et $RMIL+$ notée $CGHLB$. La convergence globale de notre méthode est démontrée pour $0 < \theta < 1$ . les expériences numériques révèlent que notre méthode atteint l'optimum en moins de nombre d'itérations et temps CPU comparés à $RMIL+$ et $PRP$.

# Conclusion générale

Ce travail de recherche se concentre sur une problématique majeure en analyse numérique et en optimisation. Il se distingue par une étude détaillée de plusieurs méthodes itératives visant à résoudre de manière efficace des problèmes d'optimisation sans contrainte. Cette approche constitue une avancée notable dans le domaine, en apportant des perspectives nouvelles et des solutions adaptées aux enjeux actuels.

Les résultats obtenus sont présentés avec rigueur, accompagnés d'analyses approfondies qui illustrent leur portée et leur utilité pratique. Les performances des techniques proposées sont validées par des simulations numériques précises, démontrant leur potentiel à améliorer les pratiques existantes et à enrichir les connaissances scientifiques dans ce champ d'étude.

# Perspectives

Il serait pertinent, dans une perspective future, d'examiner plus en détail l'apport potentiel des techniques d'apprentissage automatique dans l'enrichissement des méthodes itératives étudiées. Ces techniques offrent des capacités intéressantes en matière de généralisation et de traitement de grands volumes de données. Par ailleurs, l'adaptation de ces approches à des problèmes d'optimisation comportant des contraintes constitue une voie de recherche prometteuse, bien que complexe, qui impliquerait la conception de mécanismes adaptatifs et résilients. Enfin, une analyse rigoureuse des fondements théoriques, notamment en ce qui concerne la stabilité et la convergence des algorithmes, pourrait permettre d'en accroître la performance et de renforcer leur pertinence dans des applications concrètes.

# BIBLIOGRAPHIE


[1] **Al-Baali, M.** (1985). *Descent property and global convergence of the Fletcher—Reeves method with inexact line search*. IMA Journal of Numerical Analysis, 5(1), 121-124.

[2] **Axelsson, O., & Lindskog, G**. (1986). *On the rate of convergence of the preconditioned conjugate gradient method*. Numerische Mathematik, 48, 499-523.

[3] **Chaib, Y., & Bechouat, T**. (2023). *Two modified conjugate gradient methods for solving unconstrained optimization and application*. RAIRO-Operations Research, 57(2), 333-350.

[4] **D.Azé, J.B.Hiriart-Urruty** (2010), *recherche variationnelle et optimisation*.

[5] **Dahito, M.-A** (2018), *La méthode des résidus conjugués pour calculer les directions en optimisation continue* [Mémoire de maîtrise, École Polytechnique de Montréal].

[6] D**elladji, S., Belloufi, M., & Sellami, B**. (2021). *Behavior of the combination of PRP and HZ methods for unconstrained optimization*. Numerical Algebra, Control and Optimization, 11(3), 377-389.

[7] **Delladji, S., Belloufi, M., & Sellami, B**. (2021). *New hybrid conjugate gradient method as a convex combination of FR and BA methods*. Journal of Information and Optimization Sciences, 42(3), 591-602.

[8] **G. Zoutendijk** (1970), *Nonlinear programming, computational methods*, edited by J. Abadie, Integer and Nonlinear Programming, North-Holland, Amsterdam 37–86.

[9] **G.Laurent** (2012), *Optimisation sans contraintes de fonctions continues non linéaires*,

[10] **Gilbert, J. C., & Nocedal, J.** (1992). *Global convergence properties of conjugate gradient methods for optimization*. SIAM Journal on optimization, 2(1), 21-42.

[11] **Greenbaum, A**. (1997). *Iterative methods for solving linear systems*. Society for Industrial and Applied Mathematics.

[12] **Guefassa, I., Chaib, Y., & Bechouat, T**. (2024). *Another hybrid conjugate gradient method as a convex combination of WYL and CD methods*. Monte Carlo Methods and Applications.

[13] **Hallal, A., Belloufi, M., & Badreddine, S**. (2023). *Using a new hybrid conjugate gradient method with descent property*. Journal of Information and Optimization Sciences.

[14] **Hallal, A., Belloufi, M., & Sellami, B.** (2022). *An efficient new hybrid CG-method as convex combination of DY and CD and HS algorithms*. RAIRO-operations Research, 56(6), 4047-4056

[15] **Hamdi, A., Sellami, B., & Belloufi, M**. (2021). *New hyrid conjugate gradient method as a convex combination of HZ and CD methods*. Asian-European Journal of Mathematics, 14(10), 2150187.

[16] **Hanachi, S. B., Sellami, B., & Belloufi, M**. (2022). *New iterative conjugate gradient method for nonlinear unconstrained optimization*. RAIRO-Operations Research, 56(4), 2315-2327.

[17] **Hulshof, J., & Vandervorst, R. C. A. M**. (1993). *Differential systems with strongly indefinite variational structure*. Journal of Functional analysis, 114(1), 32-58.

[18] **J. C. Gilbert** (2007), *Eléments d'optimisation diffèrentiable : thèorie et algorithmes*, Note de cours, école Nationale Supèrieure de Techniques Avancèes, Paris.

[19] **J. Guillement** (2011),*Cours d'optimisation M15 Master Pro 1ére année*, Université Nantes.

[20] **J. Moré and E.D. Dolan** (2002), *Benchmarking optimization software with performance ?*. Math. Program. 91 201–2013

[21] **J.Parreaux** (2019), *Méthode du gradient à pas optimal*, Université Aix-Marseille.

[22] **Kelley, C. T**. (1995). *Iterative methods for linear and nonlinear equations*. Society for Industrial and Applied Mathematics.

[23] **L. Armijo** (1966), *Minimization of functions having Lipschitz-continuous first partial de rivatives*. Pacifc J.Math.

[24] **Liu, Y., & Storey**, **C** (1991). Efficient generalized conjugate gradient algorithms, part 1 : theory. Journal of optimization theory and applications, 69, 129-137.

[25] **M. Bierlaire** (2006), *Introduction à l'optimisation différentiable*. Presses polytechniques et universitaires romandes

[26] **M. Powell** (1984), *Nonconvex minimization calculations and the conjugate gradient method*. Numer. Anal. 122–141.

[27] **M. Rudolph Hestenes and Eduard Stiefel** (1952).*Methods of conjugate gradients for solving linear system*, volume 49.NBS Washington , DC.

[28] **M.Bergounioux** (2001), *Optimisation et contrôle des système linéaire*, Dunod, Paris.

[29] **M.Cerf** (2018),*Cours Techniques d'optimisation*, Université Versailles Saint Quentin en Yvelines

[30] **Nasreddine, C., Badreddine, S., & Mohammed, B.** (2022). *A new hybrid conjugate gradient method of unconstrained optimization methods*. Asian-European Journal of Mathematics, 15(04), 2250070.

[31] **P. Wolfe** (1969), *Convergence conditions for ascent methods*. SIAM Rev.11 226–235.

[32] **P. Wolfe** (1971), *Convergence conditions for ascent methods. ii : Some corrections*. SIAM Rev. 13 185–188.

[33] **P.Ciarlet**, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, Paris.

[34] **Polak, E., & Ribiere, G.** (1969). Note sur la convergence de méthodes de directions conjuguées. Revue française d'informatique et de recherche opérationnelle. Série rouge, 3(16), 35-43.

[35] **Polyak, B. T.** (1969). The conjugate gradient method in extremal problems. USSR Computational Mathematics and Mathematical Physics, 9(4), 94-112.

[36] **R. Fletcher** (1987), *Practical Methods of Optimization vol. 1 : Unconstrained Optimization*,John Wiley & Sons, New York.

[37] **R.Benzine** (16 avril 2016), *Optimisation sans contraintess, Tome2 : Methodes du gradient conjugue.*

[38] **R.Fletcher and Colin M Reeves** (1964). *Function minimization by conjugate gra dients.* The computer journal, 7(2) :149-154.

[39] **Rivaie, M., Mamat, M., & Abashar, A.** (2015). A new class of nonlinear conjugate gradient coefficients with exact and inexact line searches. Applied Mathematics and Computation, 268, 1152-1163.

[40] **Rivaie, M., Mamat, M., June, L. W., & Mohd, I.** (2012). A new class of nonlinear conjugate gradient coefficients with global convergence properties. Applied Mathematics and Computation, 218(22), 11323-11332.

[41] **S.Boyed-L.Vandenberghe**(2004), *Convex Optimisation*, Journale univeritaire de Combridge.

[42] **Sun, W., & Yuan, Y. X**. (2006). *Optimization theory and methods : nonlinear programming (Vol. 1)*. Springer Science & Business Media.

[43] **T.Dumont, C.Léonard, X.Mary, H.Mohamed**, *Cours d'optimisation*,Université Paris Ouest-Nanterre-La Défense.

[44] **Touati-Ahmed, D., & Storey, C**. (1990). *Efficient hybrid conjugate gradient techniques.* Journal of optimization theory and applications, 64, 379-397.

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

[46] **Y. Dai and Y. Yuan** (1998), **Some properties of a new conjugate gradient method. In Advances in Nonlinear Programming**. Springer 251–262.

[47] **Y. Dai and Y. Yuan** (1999) , *A nonlinear conjugate gradient method with a strong global convergence property*. SIAM J. Optim. 10 177–182.

[48] **Y. Dai and Y. Yuan** (2001), *A three-parameter family of nonlinear conjugate gradient methods*. Math. Comput. 70 1155–1167.

[49] **Y. Dai, J. Han, G. Liu, D. Sun, H. Yin and Y. Yuan** (2000) , *Convergence properties of nonlinear conjugate gradient methods*. SIAM J. Optim. 10 345–358.

[50] **Y. Dai and Y. Yuan** (2003), *A class of globally convergent conjugate gradient methods*. Sci. China Ser. A : Math. 46 251–261.

[51] **Y. Liu and C. Storey** (1991), *Efficient generalized conjugate gradient algorithms, Part 1 : Theory*, JOTA 69 ,pp. 129–137.