

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique
Université 8 Mai 1945 Guelma

Faculté des Mathématiques et de l'Informatique
et des Sciences de la Matière
Département de Mathématiques



Mémoire

Présenté en vue de l'obtention du diplôme de

Master en Mathématiques

Option : EDP et Analyse numérique

Par :

Amira Abadlia

Intitulé

**Sur la convergence d'une famille de
méthodes du gradient conjugué unifiées**

Dirigé par : Dr. Romaisa Mellal

Devant le jury

PRESIDENT

Dr. Ounesse Nawel

MCA

Univ-Guelma

EXAMINATEUR

Dr. Chawi Abd Erezzak

MCA

Univ-Guelma

Session Juin 2020

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

REMERCIEMENTS

Tout d'abord, je remercie **ALLAH** le tout puissant d'avoir me donner le courage, la volonté et la patience de mener à terme le présent travail.

J'adresse tout particulièrement mes remerciements à Dr
« **Romaissa MELLAL** »,
qui m'a encadrée, tout au long de ce mémoire. Je vous remercie pour votre précieuse présence assistance, votre disponibilité et votre patience. J'ai pris un grand plaisir à travailler avec vous.

Je remercie les membres de jury ici réuni, d'avoir accepté de participer à cette soutenance, et la lecture attentive qu'ils ont réservé à mon mémoire, et notamment pour leur engagement dans ces circonstances difficiles que nous traversons.

Je remercie l'Université de **08 Mai 1945 Guelma** d'avoir accepté ma candidature au Master 2.

Je remercie le département de Mathématiques et tous ses employés, le chef de département et les professeurs de nous accueillir et d'essayer de faciliter toutes les procédures que nous avons rencontrées.

Je vous remercie mes chers parents,
pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez.

Je tiens à exprimer ma profonde gratitude et sincères remerciements à mon fiancé Monsieur
« **Mohamed GOUASMIA** »
(professeur à ENSET Skikda) qui m'a beaucoup aidé à terminer mon travail. Je vous remercie de m'avoir continuellement encouragé, pour votre soutien scientifique et moral.

Je remercie finalement tous ceux qui m'ont soutenu et qui ont été à mes côtés durant ma carrière.

Abadlia Amira

ملخص

نعرض في هذه المذكرة الطريقة العددية للتدرج المترافق الخطية وغير الخطية على مسائل الأمثلة بدون قيود خاصة منها الأشكال التربيعية، تهتم هذه الطريقة بالبحث عن القيم الحدية للمسائل الرياضية، حيث قدمنا أولاً عموميات تخص الشروط الضرورية و أخرى كافية من أجل القيم الحدية لتابع f معرف على \mathbb{R}^n ، بعدها قمنا بدراسة تحليلية لبعض النقاط المهمة كسرعة التقارب واتجاه النزول المتعلقة بالتدرج المترافق، كما ذكرنا الطريقة الموحدة لها ونظريات تقاربها.

الكلمات المفتاحية: طريقة التدرج المترافق، اتجاه الانخفاض، سرعة التقارب.

Résumé

Dans tous les domaines des sciences, nous rencontrons à ce que nous appelons les problèmes d'optimisation, que les mathématiques ont essayé de résoudre et de traiter de nombreuses méthodes. La méthode du gradient conjugué est considérée comme l'une des méthodes les plus utilisées pour résoudre ce genre de problème notamment les problèmes de grande taille.

Dans ce mémoire, on présente une synthèse des différentes variantes de la méthode du gradient conjugué non-linéaire avec la recherche linéaire inexacte de Wolfe. Il s'agit des méthodes de **FR**, **DY** et une méthode du gradient conjugué unifiée.

Toutes ces variantes possèdent la propriété de la descente des directions générées ainsi la convergence globale. Des simulations numériques seront présentées afin d'illustrer la convergence de ces dernières.

Mots clés : Optimisation sans contraintes, gradient conjugué, recherche linéaire.

Abstract

In all areas of science we come across what we call optimization problems, which mathematics has tried to solve and to deal with many methods. The conjugate gradient method is the most used to solve this kind of problem.

In this thesis, we present a synthesis of the different variants of the non-linear conjugate gradient method with Wolfe's inexact linear search. This methods are **FR**, **DY** and a unified conjugate gradient method.

All these variants have the property of the descent of the directions generated, thus global convergence. Numerical simulations will be presented to illustrate the convergence of the latter.

Key words : Unconstrained optimization, conjugate gradient, line search.

Table des matières

1	Préliminaire	1
1.1	Généralités sur l'optimisation sans contraintes	1
1.2	Problèmes d'optimisation sans contraintes	3
1.2.1	Conditions d'optimalité des problèmes d'optimisation sans contraintes	4
1.3	Méthode à direction de descente	4
1.3.1	Algorithme itératif	5
1.3.2	Algorithme de descente	6
2	La méthode du gradient conjugué linéaire	7
2.1	Méthode du gradient	7
2.1.1	Algorithme de la méthode du gradient	7
2.1.2	Convergence de la méthode du gradient	8
2.2	Méthode du gradient conjugué	8
2.2.1	Détermination du pas α_k obtenu par une recherche linéaire exacte . . .	9
2.2.2	Détermination du coefficient β_k	10
2.2.3	Algorithme du gradient conjugué linéaire	10
2.2.4	Convergence de méthode du gradient conjugué linéaire	11
2.2.5	Vitesse de convergence	13
3	La méthode du gradient conjugué non-linéaire	15
3.1	La recherche linéaire inexacte de Wolfe	15
3.1.1	Principe de la règle de Wolfe	16
3.2	Convergence de la classe des méthodes à direction de descente	16
3.2.1	Hypothèses	16
3.2.2	Théorème de Zoutendijk	17
3.3	Algorithme de la méthode du gradient conjugué pour les fonctions quelconques	18
3.4	Méthode du gradient conjugué de Fletcher-Reeves	19
3.4.1	Algorithme de la méthode de Flecher-Reeves	19

3.4.2	Théorèmes de convergence de Fletcher et Reeves	20
3.5	Méthode du gradient conjugué de Dai-Yuan	23
3.5.1	Algorithme de la méthode de Dai-Yuan avec la règle de Wolfe	24
3.5.2	Théorèmes de convergence de Dai et Yuan	24
4	Famille de méthodes du gradient conjugués non linéaires unifiées	27
4.1	Variante de la classe des méthodes du gradient conjugué unifiées	27
4.1.1	Hypothèses	28
4.2	Algorithme de la famille des méthodes du gradient conjugué non linéaires unifiées	29
4.3	Cas particulier de la famille des méthodes du gradient conjugué non linéaires unifiées	29
4.3.1	Méthode de Flecher-Reeves	31
4.3.2	Méthode de Dai-Yuan	31
4.4	Convergence de la famille des méthodes du gradient conjugué non linéaire unifiées	31
5	Simulations numériques	37

Introduction

La méthode du gradient conjugué est considérée comme une amélioration de la méthode du gradient. Elle est l'une des méthodes les plus utilisées pour résoudre les problèmes d'optimisation non-linéaire surtout de grande taille.

Il est bien connu que la méthode du gradient consiste à partir d'un point x_0 calculé à l'itération n , on cherche le point suivant x_{n+1} sur la droite passant par x_n et dirigée par $d_n = -\nabla f(x_n)$. Ainsi, seule l'information fournie par le gradient de f à l'état n est utilisée pour déterminer la prochaine direction de descente.

Le principe de la méthode de gradient conjugué consiste, au contraire, à utiliser tous les gradients calculés précédemment par l'algorithme $\nabla f(x_0), \nabla f(x_1), \dots, \nabla f(x_n)$ pour déterminer la prochaine direction de descente. Cette méthode est basée sur le concept de conjugaison et conduit à l'optimum en un nombre fini d'itérations.

En analyse numérique, la méthode du gradient conjugué est un algorithme qui sert à résoudre des systèmes d'équations linéaires dont la matrice est symétrique définie positive, autrement dit pour la minimisation des fonctions quadratiques strictement convexes. Cette méthode a été proposée en 1952 simultanément par [EDUARD STIEFEL](#) et [MAGNUS HESTENES](#).

Plusieurs mathématiciens ont étendu cette méthode pour la minimisation des fonctions de plusieurs variables à valeurs réelles régulières non-nécessairement convexes. Ceci a été réalisé pour la première fois en 1964 par [FLETCHER](#) et [REEVES](#), (méthode de Fletcher-Reeves). Une autre variante a été proposée par [DAI](#) et [YUAN](#) en 1999 (méthode de Dai-Yuan).

En 2003, ce fut la première initiative d'unification de toutes ces méthodes dans une même classe et étudiée par la suite ces propriétés et sa convergence par [Y.DAI](#) et [Y.YUAN](#).

Les méthodes du gradient conjugué sont parmi les plus anciennes méthodes utilisées pour résoudre les problèmes de minimisation non linéaire, sans contraintes, de la forme suivante :

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

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Dans ce but, on va générer une suite $\{x_k\}$ comme suit :

On démarre par un vecteur de départ x_0 donné. A l'itération k on a la valeur de $x_k \in \mathbb{R}^n$, et x_{k+1} est calculé par la relation suivante :

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

le pas $\alpha_k \in \mathbb{R}$ est déterminé par une recherche linéaire exacte ou inexacte, les vecteurs d_0, d_1, \dots, d_n sont des directions de descente recherchée est définie par la formule de récurrence

suivante :

$$d_k = \begin{cases} -g_0 & \text{si } k = 0, \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 1; \end{cases} \quad (3)$$

$g_k = \nabla f(x_k)$ est le gradient de f au point x_k et le coefficient $\beta_k \in \mathbb{R}$ où ces différentes valeurs définissent les différentes forme du gradient conjugué.

Dans ce mémoire, on présente une synthèse de quelques variantes de la méthode du gradient conjugué non-linéaire avec la recherche linéaire inexacte de Wolfe. Il s'agit des méthodes de **FR**, **DY** et une méthode du gradient conjugué unifiée.

Toutes ces variantes possèdent la propriété de la descente des directions générées ainsi la convergence globale. Des simulations numériques seront présentées afin d'illustrer la convergence de ces dernières.

Ce mémoire est composé de cinq chapitre :

- Dans le premier chapitre, on va cité quelques notions de base qui seront utilisées par la suite, comme : les directions de descente, les conditions nécessaires et suffisantes d'optimalité,... .
- Dans le deuxième chapitre, nous avons rappelé la méthode du gradient conjugué linéaire, pour la minimisation des fonctions quadratiques strictement convexes.
- Le troisième chapitre est consacré à une étude de deux classes de méthodes du gradient conjugué non-linéaire qui sont globalement convergentes pour des fonctions de plusieurs variables à valeurs réels régulières non-nécessairement convexes. La première méthode a été introduite par Fletcher et Reeves [9] en 1964, (méthode de Fletcher-Reeves), et la deuxième a été proposée par Dai et Yuan [7], (méthode de Dai-Yuan) en 1999.
- Quant au quatrième chapitre, il traite une nouvelle variante de la méthode du gradient conjugué non linéaires nommée "la famille des méthodes du gradient conjugué non linéaire unifiée", proposée en 2003 par Dai et Yuan [8].
- On termine ce manuscrit par des simulations numériques, afin de comparer numériquement les variantes étudiées pour des fonctions test bien connues.

Préliminaire

L'objectif de ce chapitre est de rappeler quelques notions de base qui seront utilisées par la suite.

1.1 Généralités sur l'optimisation sans contraintes

L'optimisation consiste en la recherche de minimum (ou maximum) d'une certaine quantité, appelé coût ou objectif. Dans ce mémoire, on supposera que le coût dépend de n variables réelles, rassemblées en un vecteur $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, et qui fournissent une valeur $f(x)$ où f est une fonction de \mathbb{R}^n dans \mathbb{R} . Les variables (x_1, x_2, \dots, x_n) seront autorisées à prendre n'importe quelle valeur dans \mathbb{R}^n , et c'est ce qu'on appelle l'optimisation sans contraintes. On écrira les problèmes d'optimisation sans contraintes sous la forme générale suivante :

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1.1)$$

On dit que le problème (1.1) admet une solution s'il existe un choix de variable $x^* \in \mathbb{R}^n$ tel que

$$\forall x \in \mathbb{R}^n; f(x^*) \leq f(x). \quad (1.2)$$

On dit que x^* est un minimiseur (ou point de minimum) de f sur \mathbb{R}^n , et que $f(x^*)$ est un minimum de f sur \mathbb{R}^n .

Bilan Face au problème (1.1)

- L'existence d'un minimum est liée à la continuité de f .
- L'unicité du minimiseur x^* est liée à la convexité (stricte) de f .
- L'équation satisfaite par x^* est associée à la dérivée de f .

Toutes ces propriétés joueront des rôles analogues en dimension n , la dérivée sera remplacée par dérivées partielles ou dérivées directionnelles.

Définition 1.1.1. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, on appelle **GRADIENT** de la fonction f au point $x_0 \in \mathbb{R}^n$ le vecteur de \mathbb{R}^n noté ∇f et est défini par :

$$\nabla f(x_0) = \left(\frac{\partial f}{\partial x_1}(x_0), \frac{\partial f}{\partial x_2}(x_0), \dots, \frac{\partial f}{\partial x_n}(x_0) \right) \quad (1.3)$$

Définition 1.1.2. Soit $O \subseteq \mathbb{R}^n$ un ouvert et $f : O \rightarrow \mathbb{R}$ une application. Soit $x \in O$. Alors, on dit que f est **DIFFÉRENTIABLE** au point x s'il existe une application linéaire $L \in \mathcal{L}(\mathbb{R}^n, \mathbb{R})$ telle que, pour tout $h \in \mathbb{R}^n$ avec $x + h \in O$,

$$f(x + h) = f(x) + L(h) + o(\|h\|) \quad (1.4)$$

Si une telle application existe, on l'appelle différentielle de f au point x et on la note $Df(x)$.

Définition 1.1.3. Soit $d \in \mathbb{R}^n \setminus \{0\}$. on dit que f admet une **DÉRIVÉE DIRECTIONNELLE** dans la direction d , au point x , si l'application $t \in \mathbb{R} \mapsto f(x + td)$ est dérivée en 0. Si c'est le cas, on note cette dérivée

$$\frac{\partial f}{\partial d}(x) = \lim_{t \rightarrow 0} \frac{f(x + td) - f(x)}{t} \quad (1.5)$$

Dans ce cas, si $d = e_k$ est l'un des vecteurs de base de \mathbb{R}^n , on appelle cette dérivée directionnelle la $k^{\text{ème}}$ **DÉRIVÉE PARTIELLE** de f au point x , que l'on note

$$\frac{\partial f}{\partial e_k}(x) = \lim_{t \rightarrow 0} \frac{f(x_1, x_2, \dots, x_k + t, \dots, x_n) - f(x_1, x_2, \dots, x_n)}{t} \quad (1.6)$$

Définition 1.1.4. On dit qu'un ensemble \mathcal{K} est **CONVEXE** si

$$\forall \lambda \in [0; 1], \quad \forall (x, y) \in \mathcal{K}^2, \quad (1 - \lambda)x + \lambda y \in \mathcal{K} \quad (1.7)$$

Cela signifie que les deux points x et y dans \mathcal{K} et le segment lié entre x et y est inclus dans \mathcal{K} tout entier.

Définition 1.1.5. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une application, on dit que :

1. l'application f est **CONVEXE** sur \mathbb{R}^n si :

$$\forall \lambda \in [0, 1]; \quad \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n; \quad f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad (1.8)$$

2. l'application f est **STRICTEMENT CONVEXE** sur \mathbb{R}^n si :

$$\forall \lambda \in [0, 1]; \quad \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \text{ avec } x \neq y; \quad f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y) \quad (1.9)$$

Définition 1.1.6. a) On dit qu'une matrice carrée $A \in \mathcal{M}_{n \times n}$ est **SYMÉTRIQUE** si seulement si :

$$A^t = A \quad (1.10)$$

b) On dit qu'une matrice carrée $A \in \mathcal{M}_{n \times n}$ est **DÉFINIE POSITIVE** si seulement si :

$$\forall x \in \mathbb{R}^n, \quad x^t A x > 0 \quad (1.11)$$

Définition 1.1.7. Soit A une matrice symétrique $n \times n$, $b \in \mathbb{R}^n$ et c une constante.

On appelle **FORME QUADRATIQUE** la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ définie par :

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c \quad (1.12)$$

Lorsque la matrice A est définie positive (resp. semi-définie positive), on dira que $f(x)$ est une forme quadratique définie positive (resp. semi-définie positive).

Définition 1.1.8. Soit $A \in \mathcal{M}_{n \times n}$ une matrice symétrique, définie positive et $x, y \in \mathbb{R}^n$.

On dit que les deux vecteurs x et y sont conjugués par rapport à A (ou A -conjugués), s'ils vérifient :

$$x^t Ay = (x, y)_A = 0 \quad (1.13)$$

c-à-d ils sont orthogonaux pour ce produit scalaire.

Remarque Une famille de vecteurs sont conjugués par rapport à A s'ils sont orthogonaux au sens de ce produit scalaire. En particulier, ils forment donc une famille libre.

1.2 Problèmes d'optimisation sans contraintes

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

$$(P) \left\{ \min_{x \in \mathbb{R}^n} f(x) \right.$$

Considérons le problème (P)

Définition 1.2.1. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, une fonction continûment différentiable.

a) soit $x^* \in \mathbb{R}^n$, x^* est dite **SOLUTION OPTIMALE GLOBALE** (autrement dit minimum global) de (P) si et seulement si :

$$\forall x \in \mathbb{R}^n, f(x^*) \leq f(x) \quad (1.14)$$

b) soit $x^* \in \mathbb{R}^n$, x^* est dite **SOLUTION OPTIMALE LOCALE** (autrement dit minimum local) de (P) si et seulement s'il existe un voisinage $V_\epsilon(x^*)$ de x^* tel que

$$\forall x \in V_\epsilon(x^*), f(x^*) \leq f(x) \quad (1.15)$$

c) soit $x^* \in \mathbb{R}^n$, x^* est dite **SOLUTION OPTIMALE LOCALE STRICTE** de (P) si et seulement s'il existe un voisinage $V_\epsilon(x^*)$ de x^* tel que

$$\forall x \in V_\epsilon(x^*) \text{ et } x \neq x^*, f(x^*) < f(x) \quad (1.16)$$

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

Les conditions d'optimalité sont des équations, des inéquations ou des propriétés que vérifient les solutions de (P) (conditions nécessaires) ou qui assure à un point d'être solution de (P) (conditions suffisantes).

Conditions nécessaires d'optimalité

Théorème 1.2.1. [3] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable au point $x^* \in \mathbb{R}^n$. Si x^* est un minimum local de (P) alors $\nabla f(x^*) = 0$.

Théorème 1.2.2. [15] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable au point $x^* \in \mathbb{R}^n$. Si x^* est un minimum local de (P) alors $\nabla f(x^*) = 0$ et la matrice hessienne de f au point x^* , qu'on note $H(x^*)$, est semi définie positive.

Conditions suffisantes d'optimalité

Théorème 1.2.3. [15] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable au point $x^* \in \mathbb{R}^n$. Si $\nabla f(x^*) = 0$ et $H(x^*)$ est définie positive, alors x^* est un minimum local strict de (P) .

Théorème 1.2.4. [15] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ telle que f est convexe et différentiable. Alors x^* est un minimum globale de f si et seulement si

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

Remarque Dans le cas où f est convexe, il n'y a pas de distinction entre minimum local et global c'est-à-dire tout minimum local est aussi globale. De plus si f est strictement convexe, alors tout minimum local devient non seulement global mais aussi unique.

1.3 Méthode à direction de descente

Dans cette section nous allons considérer une fonction $f : \mathbb{R}^n \mapsto \mathbb{R}$, que l'on supposera différentiable et strictement convexe pour garantir l'existence d'un unique $x^* \in \mathbb{R}^n$ tel que

$$\text{Pour tout } x \in \mathbb{R}^n; \quad f(x^*) \leq f(x)$$

Rappelons que dans ce cas, le point x^* est caractérisé par l'équation suivante

$$\nabla f(x^*) = 0 \tag{1.17}$$

Trouver x^* est donc équivalent à résoudre (1.17). Cependant, en général, il n'est pas possible de déterminer une formule explicite pour x^* à partir du système d'équations (1.17), car

ces équations peuvent être non linéaire par rapport aux coordonnées $x_1^*, x_2^* \dots, x_n^*$ de x^* . C'est pourquoi on est amené à chercher une valeur approchée qui se présentent sous la forme d'algorithmes de descente.

1.3.1 Algorithme itératif

Définition 1.3.1. Un **ALGORITHME ITÉRATIF** est défini par une application vectorielle $\mathcal{A} : \mathbb{R}^n \mapsto \mathbb{R}^n$ qui génère une suite de vecteurs $(x_k)_{k \geq 0}$, à l'aide d'une construction de la forme

$$\begin{cases} x_0 \in \mathbb{R}^n & \text{étant donné (phase d'initialisation de l'algorithme)} \\ \text{Calculer } x_{n+1} = \mathcal{A}(x_n) & n\text{-ème itération.} \end{cases}$$

Ce que l'on espère, c'est que la suite $(x_n)_{n \in \mathbb{N}}$ converge vers le minimiseur x^* cherché, alors l'algorithme converge vers la solution du problème de minimisation (1.17). Si un algorithme converge, on pourra mesurer son efficacité suivant deux critères :

1. sa VITESSE DE CONVERGENCE, qui mesure la " rapidité " avec laquelle la suite $(x_n)_{n \in \mathbb{N}}$ converge vers le point x^* .
2. sa COMPLEXITÉ CALCULATOIRE, qui mesure le coût des opérations nécessaires pour obtenir chaque itération. Le coût global est alors donné par le coût d'une itération multiplier par le nombre d'itérations nécessaire pour obtenir la solution escomptée avec une certain ε fixée a priori.

La précision ε est associé à un CRITÈRE D'ARRÊT, permettant à l'algorithme de s'arrêter et de fournir une valeur approchée x_n du minimiseur, que l'on jugera acceptable. Sachant que la solution exacte satisfait l'équation (1.17), ce critère d'arrêt pourra prendre par exemple suivant :

$$\|\nabla f(x_n)\| \leq \varepsilon \quad (1.18)$$

Ainsi, l'algorithme fournira comme résultat le premier vecteur x_n obtenu, satisfaisant la condition (1.18).

Définition 1.3.2. Supposons connu une suite $(x_n)_{n \geq 0}$, obtenue à l'aide d'un algorithme itératif, et telle que $\lim_{t \rightarrow \infty} x_n = x^*$. Pour tout $n \in \mathbb{N}$, on définit l'erreur E_n associée par l'itération n par

$$E_n = \|x^* - x_n\| \quad (1.19)$$

NOTATIONS – On dira que le vitesse de convergence de l'algorithme est LINÉAIRE si

$$\exists \rho \in [0, 1[, \forall x_0 \in \mathbb{R}^n; \quad E_{n+1} \leq \rho E_n. \quad (1.20)$$

Cette propriété s'écrit de manière équivalente :

$$\exists \rho \in [0, 1[, \forall x_0 \in \mathbb{R}^n; \quad E_n \leq \rho^n E_0 \quad (1.21)$$

Pour cette raison, on dira également que si une méthode satisfait (1.20), la convergence est GÉOMÉTRIQUE (l'erreur se comporte comme une suite géométrique de raison inférieure strictement à 1).

- La méthode sera dite D'ORDRE p si elle satisfait une relation de type

$$\exists \rho \in [0, 1[, \forall x_0 \in \mathbb{R}^n; \quad E_{n+1} \leq \rho E_n^p \tag{1.22}$$

Si $p = 2$, on dira que la vitesse de convergence est QUADRATIQUE.

1.3.2 Algorithme de descente

Les algorithmes que nous allons considérer pour les problèmes d'optimisation ont la forme générale suivante

$$x_0 \text{ étant donné, calculer } x_{n+1} = x_n + \alpha_n d_n \tag{1.23}$$

Le vecteur d_n s'appelle la direction de descente, et le réel $\alpha_n > 0$ le pas de la méthode à la n -ème itération. On pratique, on choisira la direction et le pas afin que l'inégalité suivante soit satisfaite :

$$f(x_{n+1}) \leq f(x_n) \tag{1.24}$$

De tels algorithmes sont appelés ALGORITHME DE DESCENTE.

Définition 1.3.3. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable au point $x \in \mathbb{R}^n$, on dit que $d \in \mathbb{R}^n$ est une **DIRECTION DE DESCENTE** de f en $x \in \mathbb{R}^n$ si :

$$\nabla^T f(x) \cdot d < 0 \tag{1.25}$$

La direction de descente fait avec l'opposé du gradient $-\nabla f(x)$ un angle Θ strictement plus petit que 90°

$$\theta = \arccos \frac{-\nabla^T f(x) \cdot d}{\|\nabla^T f(x)\| \|d\|} \in [0, \frac{\pi}{2}[. \tag{1.26}$$

L'ensemble des directions de descente de f en x ,

$$\{d \in \mathbb{R}^n : \nabla^T f(x) \cdot d < 0\} \tag{1.27}$$

L'ensemble des directions de descente de f en x , $\{d \in \mathbb{R}^n : \nabla^T f(x_k) \cdot d < 0\}$ forme un demi-espace ouvert de \mathbb{R}^n

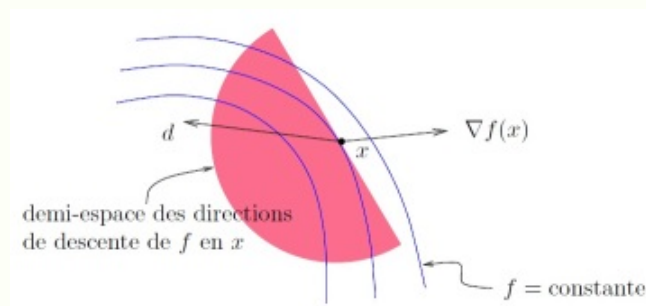


FIGURE 1.1 – Illustration des directions de descente

La méthode du gradient conjugué linéaire

La méthode du gradient conjugué est un algorithme développé dans le but de résoudre des systèmes d'équations linéaires dont la matrice est symétrique définie positive. Il s'agit d'une méthode itérative à direction de descente qui converge en un nombre fini d'itérations. Elle a été découverte en 1952 par **HESTENES** et **STEIFEL** [13]. Avant de décrire la méthode du gradient conjugué linéaire on va d'abord étudier brièvement la méthode du gradient.

2.1 Méthode du gradient

Le principe de cette méthode remonte au moins à Cauchy [4] en 1847. La méthode du gradient est une méthode à directions de descente. En effet, on peut facilement vérifier que la direction $d_k = -\nabla f(x_k)$ est une direction de descente.

L'algorithme du gradient est également connu sous le nom d'algorithme de la plus forte pente ou de la plus profonde descente parce que le gradient est la pente de la fonction au point courant et est donc, localement, sa plus forte pente.

2.1.1 Algorithme de la méthode du gradient

Algorithme 2.1

Etape 0 : (initialisation)

Soit x_0 le point de départ, poser $d_0 = -\nabla f(x_0)$
poser $k = 0$ et aller à l'étape 1.

Etape 1 :

si $\|\nabla f(x_k)\| \simeq 0$: STOP ($x^* = x_k$). "Test d'arrêt"
si non aller à l'étape 2.

Etape 2 :

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

$$\alpha_k \text{ est déterminé par une recherche linéaire} \tag{2.1}$$

$$d_k = -\nabla f(x_k) \quad (2.2)$$

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

2.1.2 Convergence de la méthode du gradient

Le théorème suivant garantit la convergence des algorithmes du gradient.

Théorème 2.1.1. [15] Soit $f : \mathbb{R}^n \mapsto \mathbb{R}$ une fonction différentiable, strictement convexe. On suppose que $\nabla f : \mathbb{R}^n \mapsto \mathbb{R}^n$ est une application L -lipschitzienne, pour une constante $L > 0$, on considère deux réels a, b tel que $0 < a < b < \frac{2\delta}{L}$ et l'on se donne une suite de pas α_n tel que $\forall n \in \mathbb{N}; \alpha_n \in [a, b]$. Alors, pour toute valeur initiale $x_0 \in \mathbb{R}$, la méthode de gradient définie par l'itération

$$x_{n+1} = x_n - \alpha_n \nabla f(x_n) \quad (2.3)$$

converge, de plus, la convergence est géométrique.

2.2 Méthode du gradient conjugué

Les méthodes du gradient conjugué [13] ont été introduites en 1952 par **HESTENES** et **STEIFEL**, dans le but de résoudre les grands systèmes linéaires. Elles reposent sur le concept des directions conjuguées parce que les gradients successifs sont orthogonaux entre eux et aux directions précédentes.

L'idée initiale était de trouver une suite à directions de descente qui sont mutuellement conjuguées par rapport à la fonction quadratique permettant de résoudre le problème

$$\min\{f(x) : x \in \mathbb{R}^n\} \quad (2.4)$$

Où f est une fonction quadratique strictement convexe. Donc minimiser $f(x)$ revient à minimiser $q(x)$, où :

$$q(x) = \frac{1}{2}x^T A x - b^T x + c \quad (2.5)$$

avec $x \in \mathbb{R}^n$ et $A \in \mathcal{M}_{n \times n}$ est symétrique et définie positive, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$.

La méthode du gradient conjugué linéaire est une méthode itérative qui génère une suite $\{x_k\}$ de la manière suivante.

On démarre par un vecteur de départ x_0 donné. A l'itération k si on a $x_k \in \mathbb{R}^n$, le successeur x_{k+1} de x_k est donné par la relation suivante :

$$x_{k+1} = x_k + \alpha_k d_k; \quad (2.6)$$

Où $\{d_0, d_1, \dots, d_n\}$ est une famille de directions de descente possédant la propriété d'être mutuellement conjuguées par rapport à la fonction quadratique $q(x)$.

La direction de descente recherchée est définie par la formule de récurrence suivante :

$$d_k = \begin{cases} -g_0 = \nabla q(x_0) & \text{si } k = 0, \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 1; \end{cases} \quad (2.7)$$

Notons par g_k le gradient de f au point x_k , le pas $\alpha_k \in \mathbb{R}^+$ étant déterminé par une recherche linéaire (optimisation unidimensionnel) et le coefficient $\beta_k \in \mathbb{R}$ est déterminé de telle sorte que les directions d_k seront mutuellement conjuguées par rapport à la forme quadratique.

2.2.1 Détermination du pas α_k obtenu par une recherche linéaire exacte

Une façon de choisir α_k consiste à résoudre le problème d'optimisation unidimensionnel suivant :

$$\alpha_k = \min f(x_k + \alpha d_k), \quad \alpha > 0, \quad (2.8)$$

Notons

$$g_k = \nabla f(x_k) = \nabla q(x_k) = Ax_k - b \quad (2.9)$$

Donc le pas α_k vérifie la condition nécessaire d'optimalité et on a $\forall k$:

$$d_k^T \nabla q(x_{k+1}) = d_k^T (Ax_{k+1} - b) = 0; \quad (2.10)$$

D'où :

$$d_k^T A(x_k + \alpha_k d_k) - d_k^T b = 0; \quad (2.11)$$

Après simplification on trouve :

$$\alpha_k = \frac{-d_k^T (Ax_k - b)}{d_k^T A d_k} \quad (2.12)$$

$$= \frac{-d_k^T (Ax_0 - b)}{d_k^T A d_k} \quad (2.13)$$

$$= \frac{-d_k^T g_k}{d_k^T A d_k} \quad (2.14)$$

2.2.2 Détermination du coefficient β_k

Les coefficients β_k sont choisis de telle sorte que d_k soit conjuguée avec toutes les directions précédentes. En d'autres termes on doit avoir :

$$d_{k+1}^T Ad_k = 0$$

on en déduit :

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

2.2.3 Algorithme du gradient conjugué linéaire

Nous résumons tout ce qui précède avec l'algorithme suivant :

Algorithme 2.2

Etape 0 : (initialisation)

Soit x_0 le point de départ, $g_0 = \nabla q(x_0) = Ax_0 - b$, poser $d_0 = -g_0$, poser $k = 0$ et aller à l'étape 1.

Etape 1 :

si $g_k = 0$: STOP ($x^* = x_k$). "Test d'arrêt" si non aller à l'étape 2.

Etape 2 :

Prendre $x_{k+1} = x_k + \alpha_k d_k$, avec :

$$\alpha_k = \frac{-d_k^T \nabla q(x_k)}{d_k^T Ad_k}, \quad (2.15)$$

$$d_{k+1} = -\nabla q(x_{k+1}) + \beta_{k+1} d_k, \quad (2.16)$$

$$\beta_{k+1} = \frac{\nabla q(x_{k+1})^T Ad_k}{d_k^T Ad_k}. \quad (2.17)$$

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

2.2.4 Convergence de méthode du gradient conjugué linéaire

La méthode du gradient conjugué linéaire converge vers la solution optimale en n itérations. Pour démontrer sa validité, nous avons d'abord besoin d'énoncer les lemmes suivants :

Lemme 2.2.1. Notons par

$$g_k = \nabla q(x_k) = Ax_k - b \quad (2.18)$$

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

$$g_{k+1}^T g_i = 0; \quad k = 0, \dots, n-1; \quad j = 0, \dots, k \quad (2.19)$$

Lemme 2.2.2. Les directions d_0, d_1, \dots, d_{n-1} engendrées par l'algorithme du gradient conjugué linéaire sont A conjuguées.

Théorème 2.2.1. [15] A une itération k quelconque de l'algorithme ou l'optimum de $q(x)$ n'est pas encore atteint (c-à-d $g_i \neq 0, i = 0, 1, \dots, k$) on a :

a)

$$\alpha_k = \frac{g_k^T g_k}{d_k^T A d_k}, \quad (2.20)$$

b)

$$\beta_{k+1}^T = \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k} \quad (2.21)$$

$$= \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \quad (2.22)$$

c) Les directions d_0, d_1, \dots, d_{k+1} engendrées par l'algorithme sont mutuellement conjuguées.

Démonstration. La preuve de ce théorème se fait par récurrence sur k . Supposons que d_0, d_1, \dots, d_k sont mutuellement conjuguées.

a) Montrons d'abord l'équivalence de (2.20) et de (2.14) :

On a : $d_k = -g_k + \beta_k d_{k-1}$ et (2.14) s'écrit :

$$\begin{aligned} \alpha_k &= \frac{-d_k^T g_k}{d_k^T A d_k} \\ &= \frac{-[-g_k + \beta_k d_{k-1}]^T g_k}{d_k^T A d_k} \\ &= \frac{g_k^T g_k}{d_k^T A d_k} - \beta_k \frac{d_{k-1}^T g_k}{d_k^T A d_k} \end{aligned}$$

Comme $(d_0, d_1, \dots, d_{k-1})$ sont mutuellement conjuguées, x_k est l'optimum de $q(x)$ sur la variété \mathcal{V}_k passant par x_0 et engendrée par $(d_0, d_1, \dots, d_{k-1})$. Donc $d_{k-1}^T g_k = 0$ d'où l'on déduit (2.20).

b) Pour démontrer (2.21) remarquons que :

$$\begin{aligned} g_{k+1} - g_k &= A(x_{k+1} - x_k) = \alpha_k Ad_k \\ \Rightarrow Ad_k &= \frac{1}{\alpha_k} [g_{k+1} - g_k]. \end{aligned}$$

On a alors :

$$g_{k+1}^T Ad_k = \frac{1}{\alpha_k} g_{k+1}^T [g_{k+1} - g_k],$$

et en utilisant (2.20)

$$\alpha_k = \frac{g_k^T g_k}{d_k^T Ad_k},$$

il vient

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

Or

$$\beta_{k+1} = \frac{g_{k+1}^T Ad_k}{d_k^T Ad_k} = \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k},$$

Ce qui démontre (2.21).

Alors du fait que :

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

Car

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

Appartient au sous-espace engendré par (d_0, d_1, \dots, d_k) et que g_{k+1} est orthogonal à ce sous-espace, de cela nous obtenons la formule (2.22).

c) Montrons enfin que d_{k+1} est conjuguée par rapport à (d_0, d_1, \dots, d_k) , on a :

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

Vérifions maintenant que : $d_{k+1}^T Ad_i = 0$ pour $i = 0, 1, \dots, k-1$.

On a :

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

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

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

$$Ad_i = \frac{1}{\alpha_i}(Ax_{i+1} - Ax_i) = \frac{1}{\alpha_i}(g_{i+1} - g_i).$$

En écrivant :

$$g_{i+1} = d_{i+1} - \beta_i d_i,$$

$$g_i = d_i - \beta_{i-1} d_{i-1},$$

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

Mais puisque (d_0, d_1, \dots, d_k) sont mutuellement conjuguées, on sait que le point x_{k+1} est l'optimum de $q(x)$ sur la variété \mathcal{V}_{k+1} , engendrée par (d_0, d_1, \dots, d_k) .

Donc g_{k+1} est orthogonal au sous-espace engendrée par (d_0, d_1, \dots, d_k) et comme Ad_i appartient à ce sous-espace pour $i = 0, 1, \dots, k-1$, on en déduit $g_{k+1}^T Ad_i = 0$ ce qui achève la démonstration. □

2.2.5 Vitesse de convergence

Le résultat suivant de Luenenberger [14], donne une estimation de la décroissance de la norme d'erreur $x_k - x^*$, en terme κ du conditionnement de la matrice A :

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}; \quad (2.23)$$

où : $\lambda_{\max}(A)$, $\lambda_{\min}(A)$ sont respectivement, les plus grande, petite valeurs propres de la matrice A . La norme utilisée est la suivante :

$$\|x\|_A = (x^T A x)^{\frac{1}{2}}; \quad (2.24)$$

Théorème 2.2.2. [14] Soit κ le conditionnement de A . La suite $\{x_k\}$ générée par l'algorithme du gradient conjugué linéaire vérifie l'estimation

$$\|x_{k+1} - x^*\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|x_1 - x^*\|_A; \quad (2.25)$$

Remarque 2.2.1. L'algorithme du gradient conjugué linéaire est surtout utile pour résoudre des grands systèmes creux, en effet il suffit de savoir appliquer la matrice A à un vecteur.

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

De façon générale la vitesse de convergence dépend de la distance entre les valeurs propres i.e le conditionnement de A .

La méthode du gradient conjugué non-linéaire

Dans ce chapitre on va présenter deux classes de méthodes du gradient conjugué non-linéaire qui sont globalement convergentes pour des fonctions de plusieurs variables à valeurs réels régulières non-nécessairement convexes. La première méthode a été introduite par **FLETCHER** et **REEVES** en 1964 , (méthode de Fletcher-Reeves). La deuxième a été proposée par **DAI** et **YUAN**, (méthode de Dai-Yuan) en 1999.

Avant de présenter ces méthodes nous allons étudier brièvement la recherche linéaire de Wolfe. Cette dernière sera utilisée pour la détermination du pas des méthodes du gradient conjugué non linéaire dans notre travail.

3.1 La recherche linéaire inexacte de Wolfe

Cette règle a été proposée par Wolfe [18] en 1969 dans le but de minimiser une fonction d'une variable réelle à valeur réelle.

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

$$\min_{\alpha \in \mathbb{R}} f(x_k + \alpha d_k) \quad (3.1)$$

Autrement dit, le pas dans la direction d_k est choisi par la minimisation unidimensionnelle de la fonction :

$$\varphi_k(\alpha) = f(x_k + \alpha d_k) \quad (3.2)$$

Dans le cas où la fonction f est quadratique définie positive alors α_k est donnée par la relation (2.14).

Mais si f n'est pas quadratique, on calcule dans ce cas α_k par une recherche linéaire inexacte.

Et pour cela, nous allons essayer de présenter brièvement la règle de Wolfe car elle sera utilisée pour démontrer la convergence globale des méthodes de Fletcher-Reeves et de Day-Yuan.

3.1.1 Principe de la règle de Wolfe

Définition 3.1.1. On dit que le pas α_k vérifie la recherche linéaire inexacte de **WOLFE FAIBLE** si on a :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g_k^T d_k \quad (3.3)$$

où $0 < \rho < 1$ et

$$g_{k+1}^T d_k > \sigma g_k^T d_k \quad (3.4)$$

où $0 < \rho < \sigma < 1$

Définition 3.1.2. On dit que le pas α_k vérifie la règle de **WOLFE FORTE** si on a :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g_k^T d_k \quad (3.5)$$

où $0 < \rho < 1$ et

$$|g_{k+1}^T d_k| < \sigma g_k^T d_k \quad (3.6)$$

où $0 < \rho < \sigma < 1$

Existence du pas de Wolfe

Le théorème suivant assure l'existence du pas de Wolfe

Théorème 3.1.1. [11] Si $\varphi_k : \mathbb{R}_+ \rightarrow \mathbb{R}$, définie par (3.2) est continue et bornée inférieurement, si d_k est une direction de descente en x_k ($\varphi_k'(0) < 0$) et si $\rho \in]0, 1[$ et $\sigma \in]\rho, 1[$, alors l'ensemble des pas vérifiant la règle de Wolfe est non vide.

3.2 Convergence de la classe des méthodes à direction de descente

Avant de citer le théorème de Zoutendijk qui assure la convergence des méthodes à direction de descente, on va citer deux hypothèses importantes.

3.2.1 Hypothèses

Pour que toutes les méthodes du gradient conjugué non linéaire atteignent la convergence, elles doivent vérifier les deux conditions suivantes :

Condition de Lipschitz C1

Proposition 3.2.1. On dit que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ vérifie la **CONDITION C1** si f est continûment différentiable dans un voisinage $V(\Lambda)$ tel que $\Lambda = \{x \in \mathbb{R}^n : f(x) \leq f(x_1) ; x_1 \in \mathbb{R}^n \text{ point initial}\}$ et si $\nabla f(x)$ vérifie la condition de Lipschitz dans $V(\Lambda)$, c-à-d, il existe une constante $M > 0$ telle que

$$\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\| \quad : \quad \text{pour tout } x, y \in V(\Lambda)$$

Condition de la bornitude C2

Proposition 3.2.2. L'ensemble $\Lambda = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ est borné, i.e, il existe une constante $L < \infty$ telle que

$$\|x\| \leq L \quad : \quad \forall x \in \Lambda$$

3.2.2 Théorème de Zoutendijk

Ce théorème est considéré comme l'outil le plus utilisé par les différentes variantes du gradient conjugué avec une recherche linéaire inexacte.

Théorème 3.2.1. [19] Considérons une méthode itérative quelconque générant une suite $\{x_k\}$ de la forme :

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

d_k étant une direction de descente et α_k est une recherche linéaire inexacte de Wolfe. Supposons aussi que f vérifie la condition **C1**, alors on a :

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \quad (3.7)$$

Remarques On a :

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

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

pour tout k , alors on a :

$$\lim_{k \rightarrow \infty} \|g_k\| = 0 \quad (3.8)$$

Dans les différentes méthodes du gradient conjugué, on a :

$$\begin{aligned} \lim_{k \rightarrow \infty} \|g_k\| &= 0 \\ \Updownarrow \\ \liminf_{k \rightarrow \infty} \|g_k\| &= 0 \end{aligned} \quad (3.9)$$

Donc pour démontrer la convergence, il suffit de s'assurer que la relation (3.9) est réalisée. Une autre version du théorème de Zoutendijk :

Théorème 3.2.2. [19] *Considérons une méthode itérative quelconque générant une suite $\{x_k\}$ de la forme :*

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

*d_k étant une direction de descente et α_k est une recherche linéaire inexacte de Wolfe. Supposons aussi que f vérifie la condition **C1**, alors on a :*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \quad (3.10)$$

ou

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty \quad (3.11)$$

3.3 Algorithme de la méthode du gradient conjugué pour les fonctions quelconques

Nous présentons dans la suite le schéma général de l'algorithme du gradient conjugué non linéaire pour des fonctions quelconques.

Algorithme 3.1**Etape 0 : (initialisation)**

Soit x_0 le point de départ, $g_0 = \nabla f(x_0)$, poser $d_0 = -g_0$,
poser $k = 0$ et aller à l'étape 1.

Etape 1 :

si $g_k = 0$: STOP ($x^* = x_k$). "Test d'arrêt" si non aller à l'étape 2.

Etape 2 :

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

α_k : calculer par la recherche linéaire

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

où

β_k : défini selon la méthode.

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

3.4 Méthode du gradient conjugué de Fletcher-Reeves

Cette méthode été découverte par **FLETCHER** et **REEVES** [9] en 1964, où le β_k est égale :

$$\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \quad (3.12)$$

3.4.1 Algorithme de la méthode de Flecher-Reeves

L'algorithme cette méthode est le suivant :

Algorithme 3.2**Etape 0 : (initialisation)**

Soit x_0 le point de départ, $g_0 = \nabla f(x_0)$, poser $d_0 = -g_0$,
poser $k = 0$ et aller à l'étape 1.

Etape 1 :

si $g_k = 0$: STOP ($x^* = x_k$). "Test d'arrêt" si non aller à l'étape 2.

Etape 2 :

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

$$\alpha_k = \min_{\alpha > 0} f(x_k + \alpha d_k) \quad (3.13)$$

$$d_{k+1} = -g_{k+1} + \beta_k^{FR} d_k \quad (3.14)$$

où

$$\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \quad (3.15)$$

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

3.4.2 Théorèmes de convergence de Fletcher et Reeves

Il y a eu de nombreuses études et résultats sur la convergence globale de la méthode de **FR** pour une recherche linéaire exacte.

Mais le premier et le plus important résultat de convergence globale de cette méthode pour une recherche linéaire inexacte a été donné par **EL-BAALI** [1] en 1985.

Théorème 3.4.1. [9] *Supposons que les hypothèses 1 et 2 soient vérifiées. Considérons une méthode du type (2.6) et (2.7), avec $\beta_k = \beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$ et le pas α_k satisfaisant à la règle de Wolfe forte (3.5) et (3.6) où $\sigma \in]0, \frac{1}{2}[$. Alors on a :*

$$\frac{-1}{1-\sigma} \leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma-1}{1-\sigma} \quad ; \quad k = 1, \dots \quad (3.16)$$

Démonstration. La démonstration se fait par récurrence :

1) Pour $k = 1$:

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

D'autre part :

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

2) Supposons que (3.16) est satisfaite pou $k > 1$ et démontrons qu'elle le sera pour $k + 1$:
Supposons que :

$$\frac{-1}{1-\sigma} \leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma-1}{1-\sigma} \quad ; \quad k = 1, \dots \quad (3.17)$$

On a :

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

D'autre part, on a :

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

d'où :

$$\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} = -1 + \frac{\beta_{k+1}}{\beta_k^{FR}} \frac{d_k^T g_{k+1}}{\|g_k\|^2} \quad (3.18)$$

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

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

Soit en remplaçant ceci dans (3.18), on obtient :

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

De (3.17), on aura :

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

et de (3.18) :

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

Par conséquent on aura :

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

Ce qui achève la démonstration.

Théorème 3.4.2. [9]

Supposons que les hypothèses 1 et 2 soient vérifiées. Considérons une méthode du type (2.6) et (2.7) avec $\beta_k = \beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$ et le pas α_k satisfaisant à la règle de Wolfe forte (3.5) et (3.6). Alors cette méthode est globalement convergente, dans le sens suivant :

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad (3.19)$$

Démonstration. Puisque les conditions du théorème (3.4.1) sont satisfaites alors on a :

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

D'autre part de (3.6)

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

d'où

$$|d_{k-1}^T g_k| \leq -\sigma d_{k-1}^T g_{k-1} \leq \frac{\sigma}{1-\sigma} \|g_{k-1}\|^2 \quad (3.20)$$

De (3.16) et (3.20) :

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

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

$$\begin{aligned} \|d_k\|^2 &\leq \hat{\sigma} \|g_k\|^2 + (\beta_k^{FR})^2 \|d_{k-1}\|^2 \\ &\leq \hat{\sigma} \|g_k\|^2 + (\beta_k^{FR})^2 [\hat{\sigma} \|g_{k-1}\|^2 + (\beta_k^{FR})^2 \|d_{k-2}\|^2] \\ &\leq \hat{\sigma} \|g_k\|^4 \sum_{j=2}^k \|g_j\|^{-2} + \hat{\sigma} \|g_k\|^4 \|g_1\|^{-2} = \hat{\sigma} \|g_k\|^4 \sum_{j=1}^k \|g_j\|^{-2} \quad (3.21) \end{aligned}$$

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

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

De (3.21) on a :

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

$$\|d_k\|^2 \leq \hat{\sigma} \frac{\lambda^4}{\omega^2} k$$

d'où :

$$\sum_{k \geq 1} \frac{1}{\|d_k\|^2} \geq \frac{\omega^2}{\hat{\sigma} \lambda^4} \sum_{k \geq 1} \frac{1}{k} > \infty \quad (3.22)$$

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

D'autre part, puisque les conditions de théorème (3.4.1), et de la théorème (3.4.2) sont satisfaites, alors on a :

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

et

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

d'où

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

Ce qui contredit (3.22), d'où le résultat. Donc

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

□

3.5 Méthode du gradient conjugué de Dai-Yuan

Cette méthode a été découverte par Y.H.Dai et Y.Yuan [7] en 1999, où β_k est égale à :

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} ; y_k = g_{k+1} - g_k \quad (3.23)$$

3.5.1 Algorithme de la méthode de Dai-Yuan avec la règle de Wolfe

L'algorithme de cette méthode est le suivant :

Algorithme 3.3

Etape 0 : (initialisation)

Soit x_0 le point de départ, $g_0 = \nabla f(x_0)$, poser $d_0 = -g_0$,
poser $k = 0$ et aller à l'étape 1.

Etape 1 :

si $g_k = 0$: STOP ($x^* = x_k$). "Test d'arrêt" si non aller à l'étape 2.

Etape 2 :

Définir $x_{k+1} = x_k + \alpha_k d_k$, avec α_k vérifie les conditions (3.3), (3.4) et

$$d_{k+1} = -g_{k+1} + \beta_k^{DY} d_k \quad (3.24)$$

où

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T (g_{k+1} - g_k)} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} \quad (3.25)$$

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

3.5.2 Théorèmes de convergence de Dai et Yuan

Le théorème de convergence de Dai et Yuan exigeant seulement la condition de Wolfe et les hypothèses 1 et 2.

Théorème 3.5.1. [8] Supposons que les hypothèses 1 et 2 soient vérifiées. Considérons des méthodes du type (2.6) et (2.7) où β_k satisfait à $\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}$; $y_k = g_{k+1} - g_k$ et le pas α_k satisfait aux conditions de Wolfe faible (3.3), (3.4) : Alors toutes les directions générées par ces méthodes sont de descente, autrement dit :

$$d_k^T g_k < 0 ; \forall k \geq 1 \quad (3.26)$$

Démonstration. La démonstration se fait par récurrence.

1) Pour $k = 1$:

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

2) Supposons que (3.26) est satisfaite pour $k > 1$ et démontrons qu'elle le sera pour $k + 1$:

Supposons que :

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

En utilisant (3.4), on aura :

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

D'autre part :

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

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

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

Ce qui achève la démonstration. □

Théorème 3.5.2. [7] *Supposons que les hypothèses 1 et 2 soient vérifiées. Considérons des méthodes du type (2.6) et (2.7) où β_k satisfait à $\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}$; $y_k = g_{k+1} - g_k$ et le pas α_k satisfait aux conditions de Wolfe (3.3), (3.4) :*

La suite $\{x_k\}$ générée par l'algorithme de Dai-Yuan converge globalement i.e

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad (3.27)$$

Démonstration. On a d'après le théorème de Zoutendijk :

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

D'autre part on a :

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

$$\implies \|d_{k+1}\|^2 = (\beta_{k+1}^{DY})^2 \|d_k\|^2 - 2d_{k+1}^T g_{k+1} - \|g_{k+1}\|^2 \quad (3.29)$$

De (2.7) :

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

Remplaçant ceci dans (3.29), on aura :

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

d'où

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

Supposons maintenant que (3.28) n'est pas satisfaite, autrement dit :

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

On aura :

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

ce qui contredit (3.28). Ceci achève la démonstration. □

Famille de méthodes du gradient conjugués non linéaires unifiées

Dans ce chapitre, on va étudier une nouvelle variante de la méthode du gradient conjugué non linéaires nommée **la famille des méthodes du gradient conjugué non linéaire unifiée** [8]. Proposée en 2003 par Dai et Yuan [8], cette classe de méthodes du gradient conjugué n'est qu'une combinaison convexe des méthodes du gradient conjugué non linéaire de Fletcher-Reeves et de Dai-Yuan. La convergence globale de cette famille a été démontré par **Dai-Yun** [8], en utilisant la recherche linéaire inexacte de Wolfe.

4.1 Variante de la classe des méthodes du gradient conjugué unifiées

Dans ce mémoire on s'intéresse à la résolution des problèmes de minimisation non linéaire sans contraintes sous la forme suivante :

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

Où f est une fonction de plusieurs variables à valeur réelle. Dans ce but, les différentes variantes du gradient conjugué génèrent une suite $\{x_n\}_{n \in \mathbb{N}}$ de la façon suivante :

$$x_{k+1} = x_k + \alpha_k d_k \quad (4.2)$$

Où le pas $\alpha_k \in \mathbb{R}_+^*$ est déterminé par une recherche linéaire exacte ou inexacte. Dans notre travail on va déterminer le pas par la recherche linéaire inexacte de Wolfe.

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

$$d_k = \begin{cases} -g_1 & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases} \quad (4.3)$$

où $g_k = \nabla f(x_k)$ où $\beta_k \in \mathbb{R}$ est donnée par une des variantes du gradient conjugué, citons :

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \quad \text{Gradient conjugué variante Fletcher Reeves} \quad (4.4)$$

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T (g_k - g_{k-1})} \quad \text{Gradient conjugué variante de Dai-Yuan} \quad (4.5)$$

En 2003, Y.Dai et Y.Yuan ont unifié ces variantes en une, on notera variante de Dai-Yuan unifiée où :

$$\beta_k^{unif} = \frac{\phi_k}{\phi_{k-1}} \quad (4.6)$$

Où

$$\phi_k = \lambda \|g_k\|^2 + (1 - \lambda)(-g_k^T d_k); \quad \lambda \in [0, 1] \quad (4.7)$$

Autrement dit

$$\beta_k^{unif} = \frac{\lambda \|g_k\|^2 + (1 - \lambda)(-g_k^T d_k)}{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(-g_{k-1}^T d_{k-1})} \quad (4.8)$$

4.1.1 Hypothèses

Afin d'assurer l'existence du pas de Wolfe, nous imposons les hypothèses suivantes :

Hypothèse 1

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continûment différentiable dans un voisinage $V(\Lambda)$ tel que $\Lambda = \{x \in \mathbb{R}^n : f(x) \leq f(x_1) \ ; \ x_1 \in \mathbb{R}^n \text{ point initial}\}$ et $\nabla f(x)$ vérifié la condition de Lipschitz dans $V(\Lambda)$, c-à-d, il existe une constante $M > 0$ telle que

$$\|\nabla f(x) - \nabla f(y)\| \leq M \|x - y\| \quad : \quad \text{pour tout } x, y \in V(\Lambda) \quad (4.9)$$

Hypothèse 2

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

$$\|x\| \leq L \quad : \quad \forall x \in \Lambda \quad (4.10)$$

Si f satisfait à hypothèse1 et hypothèse2, alors il existe une constante positive γ telle que :

$$\|g(x)\| \leq \gamma, \text{ pour tout } x \in \Lambda$$

4.2 Algorithme de la famille des méthodes du gradient conjugué non linéaires unifiées

Algorithme 4.1

Etape 0 : (initialisation)

Soit x_0 le point de départ, $g_0 = \nabla f(x_0)$, poser $d_0 = -g_0$,
poser $k = 0$ et aller à l'étape 1.

Etape 1 :

si $g_k = 0$: STOP ($x^* = x_k$). "Test d'arrêt" si non aller à l'étape 2.

Etape 2 :

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

$$\alpha_k = \min_{\alpha > 0} f(x_k + \alpha d_k) \quad (4.11)$$

$$d_{k+1} = -g_{k+1} + \beta_k^{unif} d_k \quad (4.12)$$

où

$$\beta_k^{unif} = \frac{\lambda \|g_k\|^2 + (1 - \lambda)(-g_k^T d_k)}{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(-g_{k-1}^T d_{k-1})} \quad (4.13)$$

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

4.3 Cas particulier de la famille des méthodes du gradient conjugué non linéaires unifiées

Sachant que la variante de la famille du gradients conjugués non linéaire unifiées est formée à partir d'une combinaison convexe de la méthode de Fletcher-Reeves et la méthode de Dai-Yuan. Il est évident que ces deux variantes sont des membres de cette famille. Afin de prouver ceci, nous allons énoncer un lemme dans lequel la variante de la famille du gradients conjugué non linéaire unifiées peut être écrite sous une autre forme.

Lemme 4.3.1. [8]

La variante de la classe de méthode du gradient conjugué non linéaire unifiées

$$\beta_k^{unif} = \frac{\phi_k}{\phi_{k-1}} = \frac{\lambda \|g_k\|^2 + (1 - \lambda)(-g_k^T d_k)}{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(-g_{k-1}^T d_{k-1})} \quad (4.14)$$

Où $\lambda \in [0, 1]$, peut être écrite sous la forme suivante

$$\beta_k^{unif} = \frac{\|g_k\|^2}{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(d_{k-1}^T y_{k-1})} \quad (4.15)$$

$$\phi_k = \|g_k\|^2 \frac{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(-g_{k-1}^T d_{k-1})}{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(d_{k-1}^T y_{k-1})} \quad (4.16)$$

Démonstration. en effet, d'après (4.3) on a

$$g_k^T d_k = -\|g_k\|^2 + \beta_k g_k^T d_{k-1} \quad (4.17)$$

substituant (4.17) dans (4.7) en aura

$$\begin{aligned} g_k^T d_k &= -\|g_k\|^2 + \frac{\lambda \|g_k\|^2 + (1 - \lambda)(-g_k^T d_k)}{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(-g_{k-1}^T d_{k-1})} g_k^T d_{k-1} \\ &= \frac{-\lambda \|g_{k-1}\|^2 \|g_k\|^2 - (1 - \lambda)(-g_{k-1}^T d_{k-1}) \|g_k\|^2 + [\lambda \|g_k\|^2 + (1 - \lambda)(-g_k^T d_k)] g_k^T d_{k-1}}{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(-g_{k-1}^T d_{k-1})} \end{aligned} \quad (4.18)$$

d'autre part

$$y_{k-1} = g_k - g_{k-1} \Leftrightarrow g_{k-1} = g_k - y_{k-1},$$

donc, de (4.18)

$$\begin{aligned} g_k^T d_k [\lambda \|g_{k-1}\|^2 + (1 - \lambda)(-g_k + y_{k-1})d_{k-1}] \\ = -\lambda \|g_{k-1}\|^2 \|g_k\|^2 - (1 - \lambda)(-g_{k-1}^T d_{k-1}) \|g_k\|^2 + [\lambda \|g_k\|^2 + (1 - \lambda)(-g_k^T d_k)] g_k^T d_{k-1}, \end{aligned}$$

de la relation précédente, on a

$$g_k^T d_k = -\|g_k\|^2 \frac{\lambda (\|g_{k-1}\|^2 - g_k^T d_{k-1}) + (1 - \lambda)(-g_{k-1}^T d_{k-1})}{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(d_{k-1}^T y_{k-1})} \quad (4.19)$$

donc

$$g_k^T d_k [\lambda \|g_{k-1}\|^2 + (1 - \lambda)d_{k-1}^T y_{k-1}] = -\|g_k\|^2 [\lambda (\|g_{k-1}\|^2 - g_k^T d_{k-1}) + (1 - \lambda)(-g_{k-1}^T d_{k-1})] \quad (4.20)$$

alors, de (4.20) nous déduisons une forme équivalente de β_k

$$\beta_k = \frac{\|g_k\|^2}{\lambda \|g_{k-1}\|^2 + (1 - \lambda)d_{k-1}^T y_{k-1}} \quad (4.21)$$

Pour obtenir l'expression (4.16) de ϕ_k , substituant (4.19) dans (4.7), nous aurons

$$\phi_k = \|g_k\|^2 \frac{\lambda \|g_{k-1}\|^2 + (1 - \lambda)(-g_{k-1}^T d_{k-1})}{\alpha \|g_{k-1}\|^2 + (1 - \alpha)d_{k-1}^T y_{k-1}} \quad (4.22)$$

□

4.3.1 Méthode de Flecher-Reeves

Rappelons la variante de la méthode de Flecher-Reeves donnée par la formule :

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \quad (4.23)$$

On voit bien que cette variante correspond au cas $\lambda = 1$.

En effet, en remplaçant λ par 1 dans la variante de la famille du gradients conjugué non linéaire unifiées on aura :

$$\beta_k^{unif} = \frac{\|g_k\|^2}{\lambda\|g_{k-1}\|^2 + (1-\lambda)(d_{k-1}^T y_{k-1})} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \quad (4.24)$$

d'où

$$\beta_k^{unif} = \beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \quad (4.25)$$

4.3.2 Méthode de Dai-Yuan

Rappelons la variante de la méthode de Dai-Yuan donnée par la formule :

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} \quad (4.26)$$

On voit bien que cette variante correspond au cas $\lambda = 0$.

En effet, en remplaçant λ par 0 dans la variante de la famille du gradients conjugué non linéaire unifiées on aura :

$$\beta_k^{unif} = \frac{\|g_k\|^2}{\lambda\|g_{k-1}\|^2 + (1-\lambda)(d_{k-1}^T y_{k-1})} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} \quad (4.27)$$

d'où

$$\beta_k^{unif} = \beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} \quad (4.28)$$

4.4 Convergence de la famille des méthodes du gradient conjugué non linéaire unifiées

Dans cette section nous allons présenter le théorème qui assure la convergence de cette classe de méthode.

Dans ce but on va citer des lemmes qui seront utilisées dans l'étude de convergence de la classe des méthodes du gradient conjugué non linéaires unifiées.

Afin de simplifier on note :

$$t_k = \frac{\|d_k\|^2}{\phi_k^2} \quad (4.29)$$

et

$$r_k = -\frac{g_k^T d_k}{\phi_k} \quad (4.30)$$

Lemme 4.4.1. [6] Supposons que x_1 soit un point initial pour lequel l'hypothèse 1 et 2 soient satisfaites. Considérons une méthode itérative du type (4.2) où d_k est une direction de descente et α_k vérifiant les conditions de Wolfe faibles (3.3), (3.4). Alors :

$$\sum_{k \geq 1} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \quad (4.31)$$

Lemme 4.4.2. [6] Supposons que $\{a_i\}$ et $\{b_i\}$ soient des suites à termes positifs.

Si

$$\sum_{k \geq 1} a_k = \infty \quad (4.32)$$

et pour tout $k \geq 1$ on ait :

$$b_k \leq c_1 + c_2 \sum_{i=1}^k a_i \quad (4.33)$$

où c_1 et c_2 sont des constantes positives, alors on a :

$$\sum_{k \geq 1} \frac{a_k}{b_k} = \infty \quad (4.34)$$

Lemme 4.4.3. [6] Considérons la fonction unidimensionnelle suivante :

$$\rho(t) = \frac{a + bt}{c + dt}; \quad t \in \mathbb{R} \quad (4.35)$$

avec $a, b, c, d \neq 0$ sont des nombres réels donnés. Si

$$bc - ad > 0 \quad (4.36)$$

alors $\rho(t)$ est strictement croissante pour $t < -\frac{c}{d}$ et $t > -\frac{c}{d}$. Sinon si

$$bc - ad < 0 \quad (4.37)$$

$\rho(t)$ est strictement décroissante pour $t < -\frac{c}{d}$ et $t > -\frac{c}{d}$.

Lemme 4.4.4. [8] Considérons une méthode définie par (4.2), (4.3) où β_k est donnée par la formule (4.14) et t_k définie par (4.29). Alors

$$t_k = -2 \sum_{i=1}^k \frac{g_i^T d_i}{\phi_i^2} - \sum_{i=1}^k \frac{\|g_i\|^2}{\phi_i^2}. \quad (4.38)$$

pour tout $k \geq 1$.

Démonstration. – Pour $k = 1$ la relation (4.38) est satisfaite puisque $d_1 = g_1$.

– Pour $i \geq 2$ de (4.3) on a :

$$d_i + g_i = \beta_i d_{i-1} \quad (4.39)$$

Rendant carré les deux côtés d'équation (4.39), nous obtenons

$$\|d_i\|^2 = -\|g_i\|^2 - 2g_i^T d_i + \beta_i^2 \|d_{i-1}\|^2 \quad (4.40)$$

On divise (4.40) par ϕ_i^2 et on applique (4.14) et (4.29) on aura

$$\begin{aligned} \frac{\|d_i\|^2}{\phi_i^2} &= t_i = -\frac{\|g_i\|^2}{\phi_i^2} - 2\frac{g_i^T d_i}{\phi_i^2} + \frac{\phi_i^2}{\phi_{i-1}^2} \frac{\|d_{i-1}\|^2}{\phi_i^2} \\ &= t_{i-1} - 2\frac{g_i^T d_i}{\phi_i^2} - \frac{\|g_i\|^2}{\phi_i^2} \end{aligned}$$

En additionnant sur i , on aura

$$t_k = t_1 - 2 \sum_{i=2}^k \frac{g_i^T d_i}{\phi_i^2} - \sum_{i=2}^k \frac{\|g_i\|^2}{\phi_i^2}$$

D'autre part $d_1 = -g_1$ et $t_1 = \frac{\|g_1\|^2}{\phi_1^2}$, la relation précédente est équivalent à (4.38). Ce qui prouve qu'elle est vraie pour tout $k \geq 1$. \square

Théorème 4.4.1. [8] *Supposons que l'hypothèse 1 soit vérifiée. Considérons une méthode définie par (4.2),(4.3) avec β_k de la forme (4.14) et r_k définie par (4.30). Si pour tout k , d_k est une direction de descente et le pas α_k est déterminé par la recherche linéaire de Wolfe (3.3), (3.4) et si*

$$\sum_{k \geq 1} r_k^2 = \infty \quad (4.41)$$

alors,

$$\liminf_{k \rightarrow \infty} \|g_k\| = \infty \quad (4.42)$$

Démonstration. D'après (4.40)

$$\|d_i\|^2 \geq -\|g_i\|^2 - 2g_i^T d_i \quad (4.43)$$

En multipliant le côté droit de l'équation (4.43) par $\frac{\|g_i\|^2}{\|g_i\|^2}$ on aura

$$-2g_i^T d_i - \|g_i\|^2 \leq \frac{(g_i^T d_i)^2}{\|g_i\|^2} \quad (4.44)$$

Divisons (4.44) par ϕ_i^2 et appliquons (4.38) on aura

$$t_k \leq \sum_{i=1}^k \frac{(g_i^T d_i)^2}{\|g_i\|^2 \phi_i^2} \quad (4.45)$$

de (4.30)

$$t_k \leq \sum_{i=1}^k \frac{r_i^2}{\|g_i\|^2} \quad (4.46)$$

Supposons maintenant que

$$\liminf_{k \rightarrow \infty} \|g_k\| \neq 0 \quad (4.47)$$

De la formule (4.47), il existe une constante $\delta > 0$ telle que $\|g_k\|^2 \geq \delta$ pour tout k . Ainsi de la formule, de (4.46), on tire

$$t_k \leq \frac{1}{\delta^2} \sum_{i=1}^k r_i^2$$

En combinant la relation (4.41) dans le lemme (4.2) on obtient

$$\sum_{i \geq 1} \frac{r_i^2}{t_i^2} = \sum_{i \geq 1} \frac{(g_i^T d_i)^2 \phi_i^2}{\|d_i\|^2 \phi_i^2} = \sum_{i \geq 1} \frac{(g_i^T d_i)^2}{\|d_i\|^2} = \infty \quad (4.48)$$

Les formules (4.48) et (4.38) nous conduise à une contradiction. Donc (4.42) est satisfaite. \square

Citons le théorème fondamental de convergence de la famille de méthodes du gradient conjugué non linéaire unifiées avec la recherche linéaire inexacte de Wolfe

Théorème 4.4.2. [8] *Supposons que les hypothèses 1 et 2 sont vérifiées. Considérons une méthode définie par (4.2),(4.3) avec β_k de la forme (4.14) et r_k définie par (4.30). Si pour tout k , d_k est une direction de descente et le pas α_k est déterminé par la règle de Wolfe (3.3), (3.4) et si*

$$\sum_{k \geq 1} \frac{\|g_k\|^2}{\phi_k^2} = \infty \quad (4.49)$$

alors, nous avons

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad (4.50)$$

Démonstration. D'après (4.29)

$$t_k \geq 0 ; \text{ pour tout}$$

donc, de (4.38)

$$\begin{aligned} -2 \sum_{i=1}^k \frac{g_i^T d_i}{\phi_i^2} &\geq \sum_{i=1}^k \frac{\|g_i\|^2}{\phi_i^2} \Leftrightarrow -4 \sum_{i=1}^k \frac{g_i^T d_i}{\phi_i^2} \geq 2 \sum_{i=1}^k \frac{\|g_i\|^2}{\phi_i^2} \\ &\Leftrightarrow -4 \sum_{i=1}^k \frac{g_i^T d_i}{\phi_i^2} - \sum_{i=1}^k \frac{\|g_i\|^2}{\phi_i^2} \geq \sum_{i=1}^k \frac{\|g_i\|^2}{\phi_i^2} \end{aligned}$$

de (4.44) on a

$$4 \sum_{i=1}^k \frac{(g_i^T d_i)^2}{\|g_i\|^2 \phi_i^2} \geq -4 \sum_{i=1}^k \frac{g_i^T d_i}{\phi_i^2} - \sum_{i=1}^k \frac{\|g_i\|^2}{\phi_i^2} \geq \sum_{i=1}^k \frac{\|g_i\|^2}{\phi_i^2}$$

alors, si (4.49) est satisfaite nous avons aussi

$$\sum_{k \geq 1} \frac{(g_k^T d_k)^2}{\|g_k\|^2 \phi_k^2} = \infty \quad (4.51)$$

Puisque (4.45) est satisfaite, d'après (4.51) et le lemme (4.1.1) on obtient

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

Donc la relation (4.38) est vérifiée et par conséquent (4.50) est satisfaite. Ce qui achève la démonstration. □

Simulations numériques

Dans ce chapitre, on va étudier les performances des méthodes FR, DY et la classe de méthodes du gradient conjugués non linéaires unifiées avec la recherche linéaire de Wolfe forte. Dans ce but, on va les comparer pour les fonctions TRIDIA et DBVF. L'algorithme de la recherche linéaire inexacte de Wolfe forte assure que la direction est de descente.

Dans nos simulations numériques on a limité le nombre des itérations nécessaires dans l'algorithme de la recherche linéaire inexacte de Wolfe forte, et le pas initial est égal à un.

Les paramètres sont donnés : $\rho = 0.1, \sigma = 0.01, \beta = 0.95$ et $\epsilon = 0.1$. Tandis que la valeur de λ est précisée à chaque fois.

Dans le cas où le dénominateur de la variante du gradient conjugué β_k est nulle on redémarre en prenant la direction de la plus profonde pente. Powell a éclairé les bien effets du redémarrage dans l'algorithme.

Les programmes sont écrits sous scilab 6.2.0 et exécutés dans un ordinateur Intel PC i5 avec une mémoire 3.10 GHz et un système Windows sept 64 bits.

Pour chaque test le critère d'arrêt est $\|g_k\| \leq \epsilon$, ou $\epsilon = 0.1$.

fonction TRIDIA (cute)

Cette fonction teste est définie comme suit :

$$f(x) = \gamma(\delta x_1 - 1)^2 + \sum_{i=2}^n (\alpha x_i - \beta x_{i-1})^2, \quad (5.1)$$

$$\alpha = 2, \beta = 1, \gamma = 1, \delta = 1 \text{ and } x_0 = [1, \dots, 1].$$

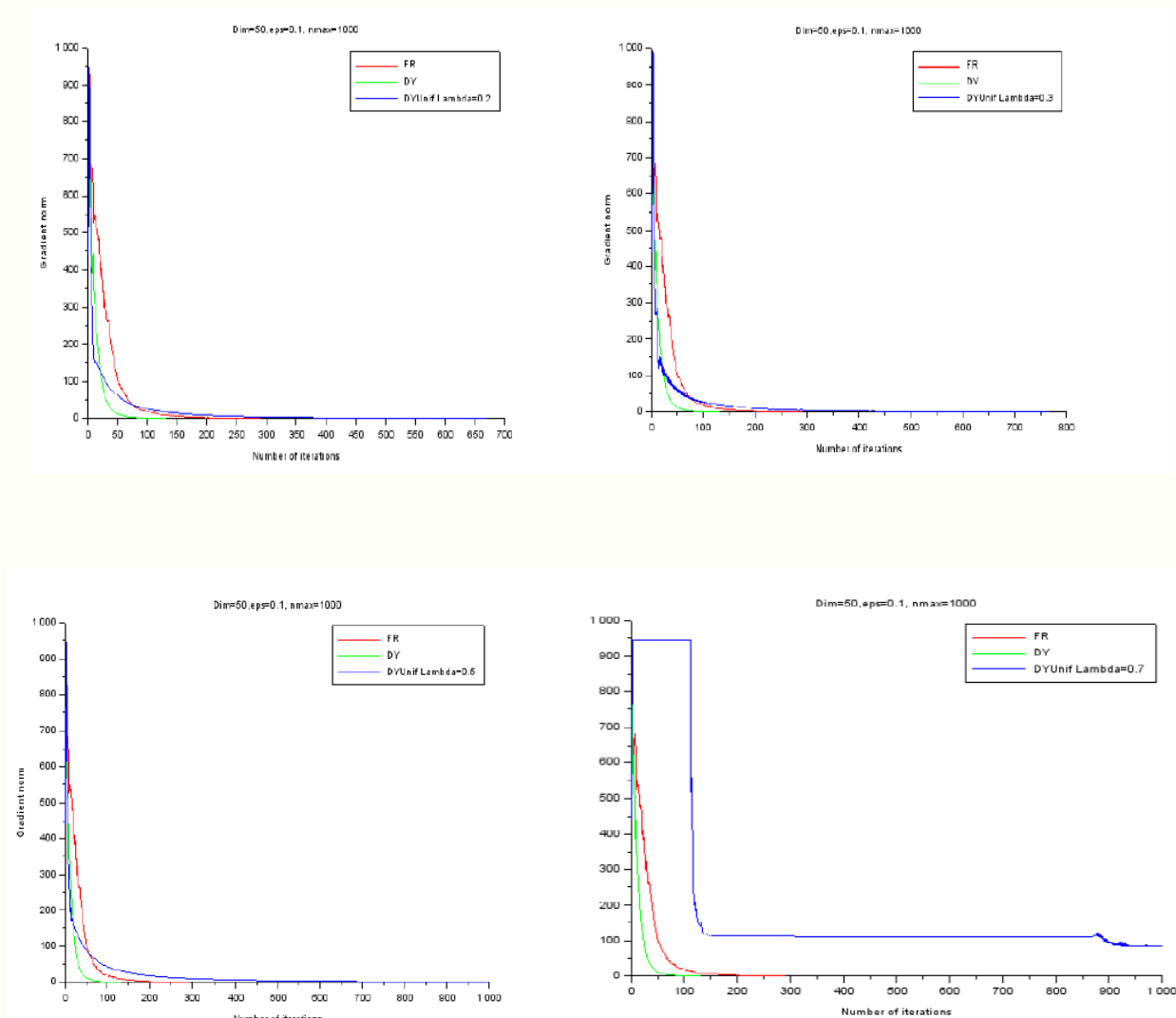


FIGURE 5.1 – Comparaison des méthodes de FR, DY et DYunif pour la fonction TRIDIA

La figure 5.1 représente une comparaison des méthodes de FR, DY et de DY unifiée, dont λ prend les valeurs 0.2, 0.3, 0.5 et 0.7. On a exécuté ces méthodes pour la fonction TRIDIA, en prenant la dimension $n = 50$, la précision $\epsilon = 0.1$ et le nombre maximale d'itérations $nmax = 1000$. On voit bien que notre cas la classe de méthodes de Dai-Yuan unifiée converge dans un nombre d'itérations supérieure à celles de FR et DY et que pour le cas où $\lambda = 0.7$, elle diverge.

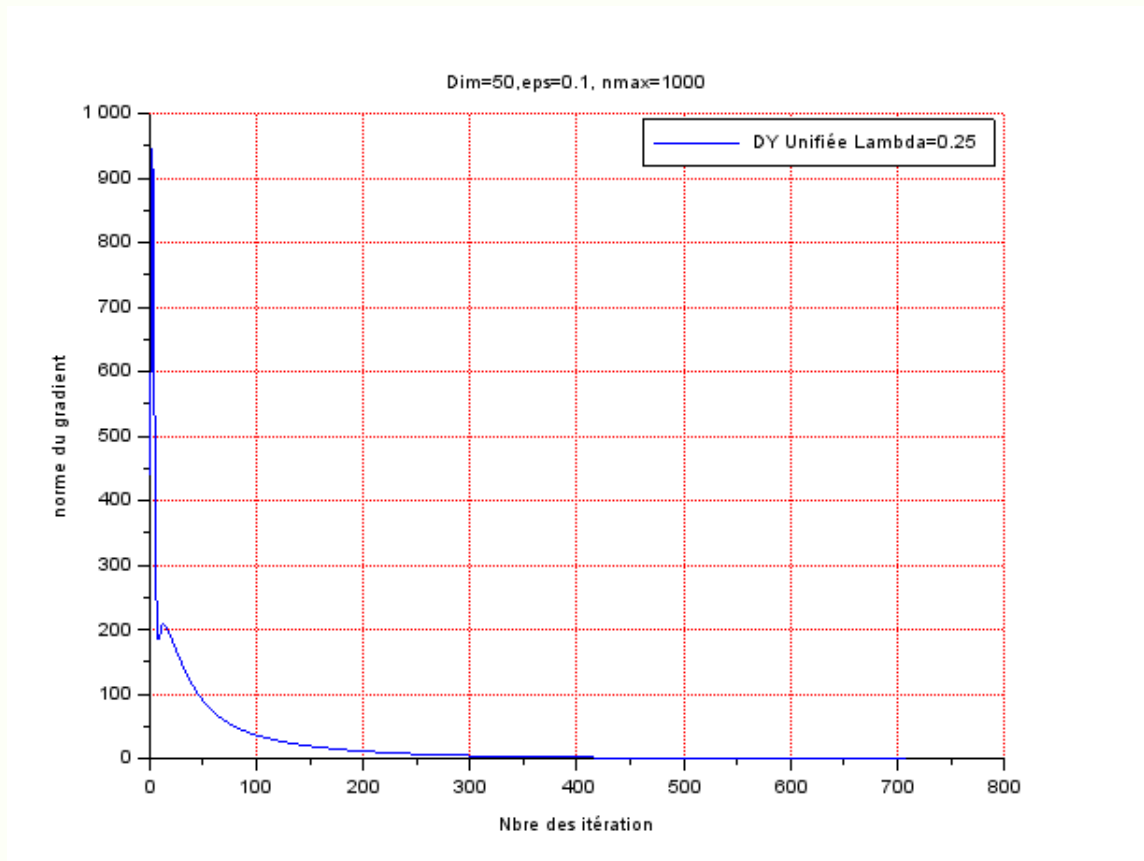


FIGURE 5.2 – Application de la variante de DYunif pour la fonction TRIDIA

La figure montre la convergence de la méthode de DY unifiée pour $\lambda = 0.25$ appliquée à la fonction test TRIDIA pour le cas où $n = 50$ et $\epsilon = 0.1$. La convergence se fait avant 800 itérations.

fonction DBVF (Discrete boundary value)

Cette fonction teste est définie comme suit :

$$f_i(x) = 2x_i - x_{i-1} - x_{i+1} + h^2(x_i + t_i + 1)^3/2 \tag{5.2}$$

$$h = 1/(n + 1), \quad t_i = ih, \quad x_0 = x_{n+1} = 0$$

$$\text{et } x_0 = (\xi_i) \text{ where } \xi_i = t_i(t_i - 1).$$

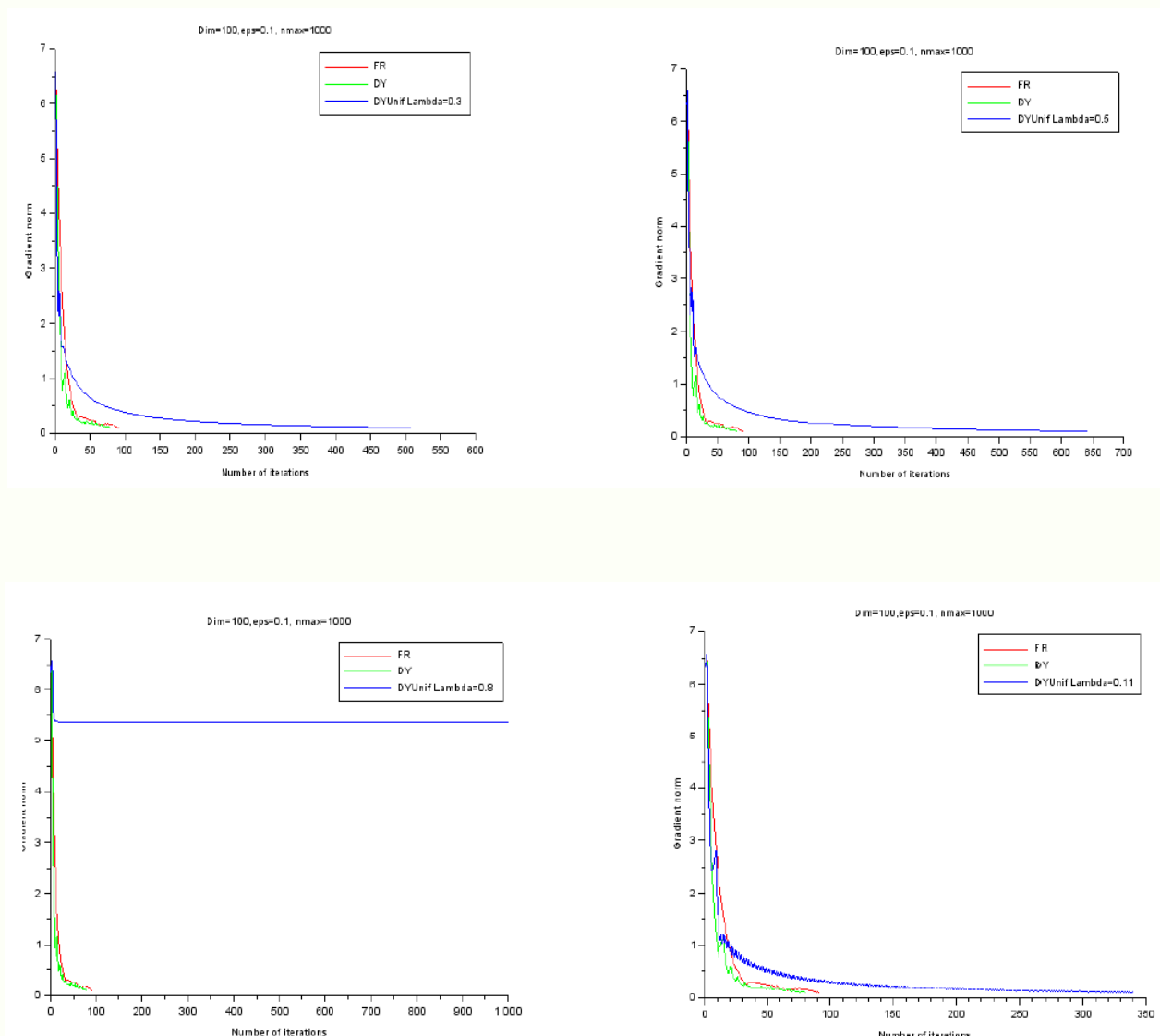


FIGURE 5.3 – Comparaison des méthodes de FR, DY, DYunif pour la fonction DBVF

La figure 5.3 représente une comparaison des méthodes de FR, DY et de DY unifiée, dont λ prend les valeurs 0.3, 0.5, 0.8 et 0.11. On a exécuté ces méthodes pour la fonction DBVF, en prenant la dimension $n = 50$, la précision $\epsilon = 0.1$ et le nombre maximale d'itérations $nmax = 1000$. On voit bien que dans notre cas la classe de méthodes de Dai-Yuan unifiée

converge dans un nombre d'itérations supérieure à celles de FR et DY et que pour le cas où $\lambda = 0.8$, elle diverge.

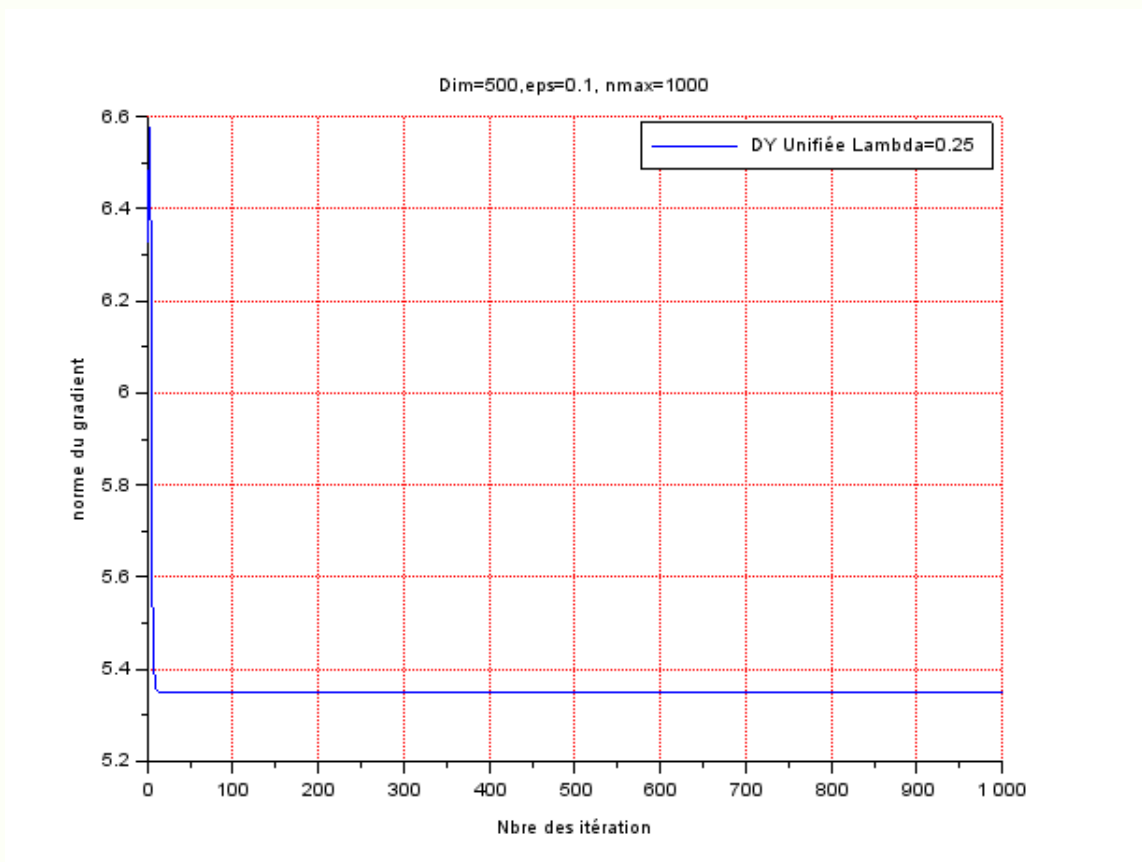


FIGURE 5.4 – Application de la variante de DYunif pour la fonction DBVF

De la figure 5.4 on observe que la méthode de DY unifiée pour $\lambda = 0.25$, appliquée à la fonction test DBVF pour le cas où $n = 50$ et $\epsilon = 0.1$ diverge.

Conclusion

Conclusion

Dans ce mémoire, on s'intéresse à la méthode du gradient conjugué non-linéaire.

On a étudié les méthodes du gradient conjugué non-linéaires de Fletcher-Reeves, Dai-Yuan et une nouvelle classe de méthode du gradient conjugué nommée : famille de méthodes de Dai-Yuan unifiée.

La variante de cette dernière est une combinaison convexe des variantes de FR et DY.

Les algorithmes et les théorèmes qui assurent la convergence des méthodes citées au dessus sont présentés.

On a terminé notre travail par des simulations numériques sur les fonctions test TRIDIA et DBFV.

Nos résultats montre que la méthode DY unifiée est moins performante que celles de FR et DY pour les cas étudiés.

Bibliographie

- [1] M. Al-Baali, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, *IMA J. Numer. Anal.*, 5 (1985), pp. 121-124.
- [2] M. Al-Baali. New property and global convergence of the Fletcher-Reeves method with inexact line searches. *IMA J. Numer. Anal.* 5(1985) 122-124.
- [3] M. S. Bazaraa, H. D. Sherali, et C. M. Shetty, *Nonlinear Programming, Theory and Algorithms*, Wiley-Interscience (1993).
- [4] A. Cauchy, *Analyse mathématique, Méthode générale pour la résolution des systèmes d'équations simultanées*, *Comptes Rendus de l'Académie des Sciences de Paris*, (1847), t-25, pp. 536-538.
- [5] Y. H. Dai and Y. Yuan, Convergence properties of the conjugate descent method, *Adv. Math. (China)*, 26 (1996), pp. 552-562.
- [6] Y. H. Dai and Y. Yuan. Some properties of a new conjugate gradient method. *Advances in Nonlinear Programming* (Kluwer, Boston, 1998), pp. 251-262.
- [7] Y.H. Dai and Y. Yuan. A nonlinear conjugate gradient with a strong global convergence property. *SIAM J. Optimization*, Vol. 10(1),(1999) pp.177-182.
- [8] Y. H. Dai and Y. Yuan. A class of globally convergent conjugate gradient methods. *Sci. China Series. Math. Ser. A.* 2003 ;46 :251-261.
- [9] R. Fletcher and C. Reeves. Function minimization by conjugate gradients, *Comput. J.* 7 : 149-154.
- [10] R. Fletcher, *Practical Methods of Optimization vol. 1 : Unconstrained Optimization*, John Wiley Sons, New York, 1987.
- [11] J.C. Gilbert, *Éléments d'Optimisation Différentiable : Théorie et Algorithmes*, Notes de cours, École Nationale Supérieure de Techniques Avancées, Paris (2007).
- [12] J. C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods. for optimization. *SIAM. J. Optimization.* Vol. 2 No. 1 (1992), pp. 21-42.
- [13] M. R. Hestenes and E. L. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards Sect. 5*, 49 (1952), 409-436.

-
- [14] D. G. Luenberger, Optimization by Vector Space Methods, John Wiley and Sons, Inc. New York (1969).
- [15] M. Minoux, Programmation Mathématique, Théorie et Algorithmes, tome 1 , Dunod (1983).
- [16] M.J.D. Powell, Convergence properties of algorithms for nonlinear optimization, SIAM rev.28, (1983) pp. 487-500.
- [17] D. F. Shanno. Conjugate gradient methods with inexact searches. Math. Oper. Res. 3 (1978), 244-256.
- [18] P.Wolfe. Convergence conditions for ascent method, SIAM Rev. 1969 .
- [19] G. Zoutendijk. Nonlinear Programming. Computational Methods, in : J. Abadie (Ed.), Integer and Nonlinear Programming, North-Holland, Amsterdam, 1970, pp.3786.