People's Democratic Republic Ministry of Algeria
Higher Education and Scientific Research
Echahid Echeikh Larbi Tebessi University- Tebessa
Faculty of the Exactes Sciences and Sciences of
Nature and life
Departement: Mathematics

End Of Study Thesis
For Obtaining The MASTER Diploma
Domaine: Mathematics and Computer Sciences
Field: Mathematics
Option: Partial Differential Equations and Applications
Topic

# Acceleration Of The Convergence Of The Gradient Method By Using The Conjugate Gradient

Presented by:

**Amina BAHI**

the jury :

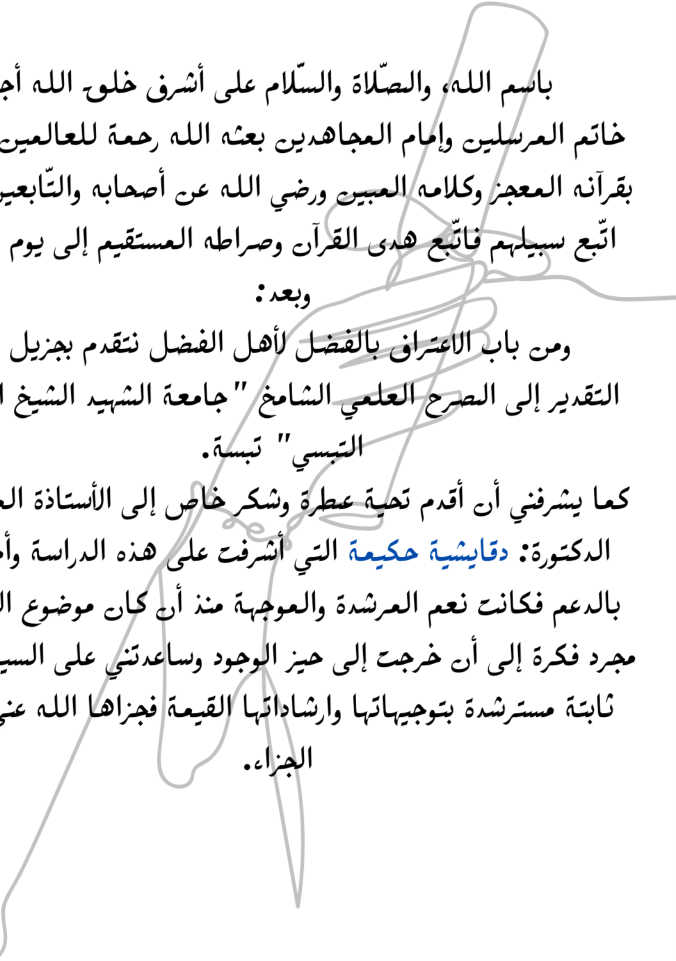
| | | | |
|---|---|---|---|
| **Mrs.Fatiha MESLOUB** | PROF | Larbi Tebessi University | President |
| **Mrs.Hakima DEGAICHIA** | MCA | Larbi Tebessi University | Supervisor |
| **Mrs.Rachida MEZHOUD** | MCA | Larbi Tebessi University | Examiner |

Date of Graduation : 09/06/2024

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

وَقُل رَّبِّ زِدْنِي عِلْماً

# الشكر والتقدير



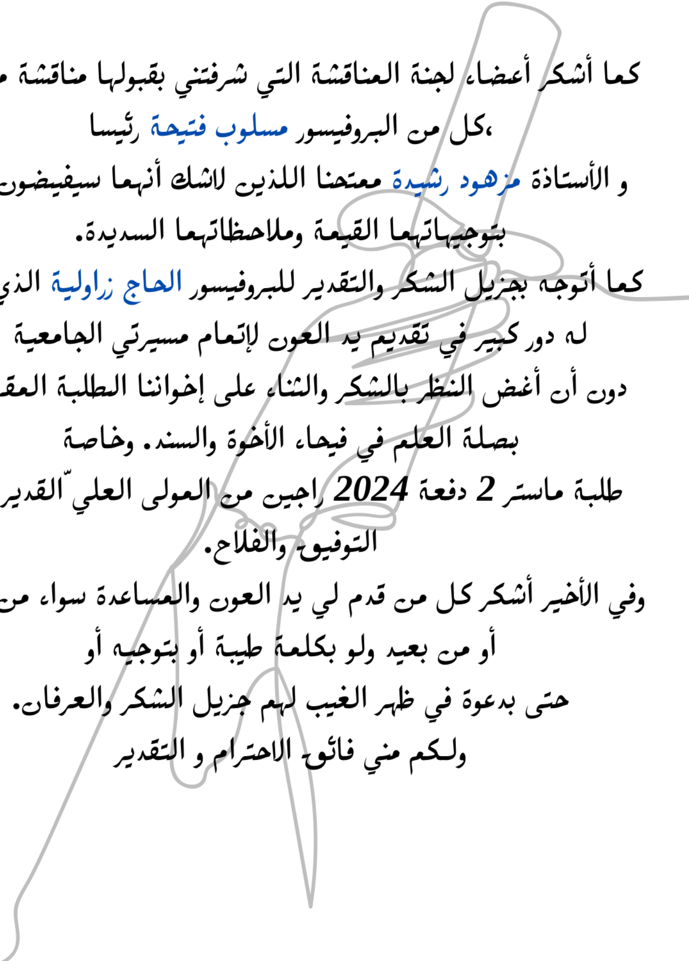
باسم الله، والصّلاة والسّلام على أشرف خلق الله أجمعين، خاتم المرسلين وإمام المجاهدين بعثه الله رحمة للعالمين، وأيّده بقرآنه المعجز وكلامه المبين ورضيِ الله عن أصحابه والتّابعين، ومن اتّبع سبيلهم فاتّبع هدى القرآن وصراطه المستقيم إلى يوم الدّين

وبعد:

ومن باب الاعتراف بالفضل لأهل الفضل نتقدم بجزيل الشكر و التقدير إلى الصرح العلمي الشامخ "جامعة الشهيد الشيخ العربي التبسي" تبسة.

كما يشرفني أن أقدم تحية عطرة وشكر خاص إلى الأستاذة المشرفة: الدكتورة: دقايشية حكيمة التي أشرفت على هذه الدراسة وأمدتني بالدعم فكانت نعم المرشدة والموجهة منذ أن كان موضوع الدراسة مجرد فكرة إلى أن خرجت إلى حيز الوجود وساعدتني على السير بخطى ثابتة مسترشدة بتوجيهاتها وارشاداتها القيمة فجزاها الله عني خير الجزاء.

# الشكر والتقدير



كما أشكر أعضاء لجنة المناقشة التّي شرفتني بقبولها مناقشة مذكرتّي
،كل من البروفيسور مسلوب فتيحة رئيسا
و الأستاذة مزهود رشيدة ممتحنا اللذين لاشك أنهما سيفيضون عليا
بتوجيهاتّهما القيمة وملاحظاتّهما السديدة.
كما أتّوجه بجزيل الشكر والتّقدير للبروفيسور الحاج زراولية الذي كان
له دور كبير في تقديم يد العون لإتّمام مسيرتّي الجامعية
دون أن أغض النظر بالشكر والثناء على إخواننا الطلبة العقربين
بصلة العلم في فيحاء الأخوة والسند. وخاصة
طلبة ماستر 2 دفعة 2024 راجين من المولى العلي ّالقدير كل
التّوفيق والفلاح.
وفي الأخير أشكر كل من قدم لي يد العون والمساعدة سواء من قريب
أو من بعيد ولو بكلمة طيبة أو بتّوجيه أو
حتّى بدعوة في ظهر الغيب لهم جزيل الشكر والعرفان.
ولكم مني فائق الاحترام و التّقدير

# الإِهْدَاء

الحمد لله حبا وشكرا وامتنانا على البدء و الختام

﴿ وَأَنْ لَيْسَ لِلْإِنْسَانِ إِلَّا مَا سَعَى ﴾

أرى مرحلتي الدراسية قد شارفت على الإنتهاء بالفعل ، بعد تعب ومشقة دامت سنين في سبيل الحلم والعلم حملت في طياتها أمنيات الليالي، ها أنا اليوم أقف على عتبة تخرجي أقطف ثمار تعبي وأرفع قبعتي بكل فخر،فاللّهم لك الحمد قبل أن ترضى ولك الحمد إذا رضيت ولك الحمد بعد الرضا، لأنك وفقتني على إتمام هذا النجاح وتحقيق حلمي.

وبكل حُب أهدي ثمرة نجاحي وتخرجي:

إلى من لا أرى الأمل إلا من عينيها... إلى من ركع العطاء أمام قدميها... وكان دعائها سر نجاحي...وحبها بلسم جراحي، وما فتئت ترسعني حتى وجدتني هنا لأحقق لها ما رسمت..إلى أغلى إنسانة في الوجود...إلى من هي جواز سفري إلى الجنة حبا وبرا وطاعة..."أمي الغالية أطال الله في عمرها"

إلى قدوتي الأولى... ونبراسي الذي ينير دربي...إلى من أعطاني ولم يزل يعطيني بلا حدود...إلى من كان سندي و قوتي و ملاذي بعد الله... إلى من رفعت رأسي عاليا افتخارا به..." أبي العزيز أدامه الله ذخرا لي"

إلى وحيدتي و مؤنستي و أختي " منار" ..إلى من شد الله بهم عضدي إخوتي "البخاري" و "مسلم" حفظهم الله وأنار دروبهم

إلى أخي الغالي و زوج أختي الغالية "بدر الدين "

إلى الكتاكيت الصغار "أنس و سيدرا "، والغالي " عبد المطلب " حفظهم الله

لكل من كانوا سندا لي في طريقي و رفيقاتي في رحلتي الجامعية و من شجعني على مواصلة الدرب "سندس" "نريمان" "كاملة" "جيهان" "مريم"

إلى رفيقات العمر "مروى" "تهاني"

أمينة باهي

# Abstract

The conjugate gradient method is considered one of the most important methods used to speed up the gradient algorithm, for this purpose, several related algorithms have been developed. We will present a new method that accelerates the convergence of the gradient method (the higher slope method) using a new version of the conjugate gradient and a powerful non-exact linear Wolf search. It will be shown that this algorithm generates descent trends and converges globally.

**Keywords:** examples without restrictions, gradient method, algorithm, general convergence, linear search, imprecise linear search, imprecise linear search for Armijo, Armijo algorithm, imprecise strong linear search for Wolf, imprecise weak linear search for Wolf, gradient method, conjugate gradient method.

# Résumé

La méthode du gradient conjugué est considérée comme l'une des méthodes les plus importantes utilisées pour accélérer l'algorithme du gradient, et à cette fin, de nombreux algorithmes connexes ont été développés.

On exposera une nouvelle méthode qui accélère la convergence de la méthode du gradient (méthode de la plus forte pente) en utilisant une nouvelle version du gradient conjugué et une recherche linéaire inexacte de Wolfe forte. On montrera que cet algorithme génère des directions de descente et converge globalement.

**Les mots clés :** exemples sans restrictions, méthode du gradient, algorithme, convergence générale, recherche linéaire, recherche linéaire imprécise, recherche linéaire imprécise pour Armijo, algorithme Armijo, recherche linéaire forte imprécise pour Wolfe, recherche linéaire faible imprécise pour Wolfe, méthode du gradient, méthode du gradient conjugué.

# ملخص

تعتبر طريقة التدرج المترافق واحدة من أهم الطرق المستخدمة لتسريع خوارزمية التدرج، ولهذا الغرض، تم تطوير العديد من الخوارزميات ذات الصلة.
سنقدم طريقة جديدة تسرع تقارب طريقة التدرج (طريقة المنحدر الأعلى) باستخدام إصدار جديد من التدرج المترافق وبحث خطي قوي غير دقيق وولف. سيظهر أن هذه الخوارزمية تولد اتجاهات النسب والتقارب الكلي.

**الكلمات المفتاحية:** الأمثلة بدون قيود، طريقة التدرج، خوارزمية ، التقارب العام،البحث الخطي، البحث الخطي غير الدقيق، البحث الخطي غير الدقيق لـ أرميجو، خوارزمية أرميجو، البحث الخطي القوي غير الدقيق لـ وولف، البحث الخطي الضعيف غير الدقيق لـ وولف، طريقة التدرج، طريقة التدرج المترافق.

# Contents

# Notations & abbreviations

| | |
|---|---|
| $\mathbb{R}^n$ | n-dimensional Euclidean (real) space |
| $T$ | transpose of a vector or matrix |
| $x$ | $x = [x_1, x_2 ... x_n]^T$ |
| $\{...\}$ | set |
| $f(x), f$ | objective function |
| $\hat{x}$ | local minimizer |
| $f(\hat{x})$ | minimum function value |
| $C^1$ | set of continuous differentiable functions |
| $C^2$ | set of continuous and twice differentiable function |
| $\subset, \subseteq$ | subset of |
| $\|.\|$ | absolute value |
| $[a, b]$ | closed interval between the real numbers a and b |
| $\mathbb{R}$ | set of real numbers |
| $det A$ | determinant of matrix A |
| $\mathbb{B}$ | Euclidean closed unit ball |
| $A^{-1}$ | Inverse of matrix A |
| $I$ | identity matrix |
| $x^T y$ | scalar product of the vectors $x$ and $y$ |
| $\\|x\\|$ | Euclidean norm of $x$ |
| $\{x_k\}$ | sequence in $\mathbb{R}^n$ |
| $f'(x)$ | derivative of $f$ at $x$ |

# Introduction

Let be $f : \mathbb{R}^n \to \mathbb{R}$ and $(P)$ the problem of nonlinear, unconstrained minimization as follows:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Note

$$g_k = \nabla f(x_k)$$

To solve the problem $(P)$, the majority of methods generate an $\{x_k\}_{k \in N}$ suite in the following form:

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

where $d_k$ is a descent direction and $\alpha_k$ is the pitch obtained by performing a one-dimensional optimization. In the conjugate gradient methods the descent directions are of the form :

$$d_k = -g_k + \beta_k d_{k-1} \tag{2}$$

where the scalar $\beta_k$ characterise the different variants of the conjugate gradient. If $\beta_k = 0$, then we get the gradient method. Another choice of directions is given by

$$d_k = -B_k^{-1} g_k \tag{3}$$

where $B_k$ is a nonsingular symetric matrix. Important cases include:

$$B_k = I \text{ (Steepest descent method)}$$
$$B_k = \nabla^2 f(x_k) \text{ (Newton's method)}$$

The quasi Newton methods are also of form (0.3). All these methods are implemented taking into consideration that $d_k$ is a descent direction i.e.

$$d_k^T g_k < 0$$

The convergence properties of the methods are descent directions and linear searches depend the right choice of $d_k$ and step $\alpha_k$. The angle that the $d_k$ direction and gradient direction makes is fundamental.Thatâs why weâre undoing

$$\cos(\theta_k) = \frac{-d_k g_k}{\|g_k\| \|g_k\|}$$

We will choose $\alpha_k$ so that we get a decrease on health of the function $f$, but at the same time it is necessary that this calculation is not coteux in time and memory. The optimal choice is obtained by choosing $\alpha$ as the optimal solution for the variable function $\varphi(\alpha)$ deffinit by

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

Exact linear searches consist of calculating $\alpha_k$ as a solution to the following one-dimensional problem:

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

Unfortunately, exact linear searches are difficult to perform practically and are costly in time and memory. The strategy we will apply in this part is to choose $\alpha_k$ verifying the following two conditions:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \sigma_1 \alpha_k g_k^T d_k \tag{4}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 g_k^T d_k \tag{5}$$

where $0 < \sigma_1 < \sigma_2 <$. T 1he first relation (4) (Armijo condition), ensures that the function sufficiently decreases. The second condition (5) warns that the step $\alpha_k$ becomes small. Both conditions (4) and (5) are called Wolfe conditions.
You can also choose $\alpha_k$ checking the following conditions:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \sigma \alpha_k g_k^T d_k \tag{6}$$

$$f(x_k + \alpha_k d_k) \geq f(x_k)(1 - \sigma) \alpha_k g_k^T d_k \tag{7}$$

where $0 < \alpha < \dfrac{1}{2}$ . (6) and (7) are called Goldstein conditions. The gradient method is one of the simplest and most celebrated methods of constraint-free optimization. For many problems the gradient method becomes slow when approaching a stationary point. There are many methods that remedy this problem. Instead of considering $d_k = -\nabla f(x_k)$ , you can move along $d_k = -D_k \nabla f(x_k)$ , or the long $d_k = -g_k h_k$ where $D_k$ is a properly selected matrix and $h_k$ is an appropriate vector.

Benzine, Djeghaba and Rahali tried to solve this problem by another method, accelerating the convergence of the gradient method.

To achieve this goal, they developed a new algorithm they named the epsilon steepest descent algorithm, in which the formula of Florent Cordellier and Wynn play They also proved global convergence using exact linear and Armijo research.

In this work we accept the convergence of the gradient method and we study the global convergence using the epsilon algorithm and the inaccurate linear searches of wolfe veriffiant (4) and (5). We called the new algorithm: Wolfe epsilon steepest descent algorithm.

With 700 numerical tests we have shown that the new algorithm is more performing than the other two already studied i.e. the steep elpsilon algorithm with exact linear or Armijo searches.

# Chapter 1

# Unconstrained Optimization

The problem we are studying here is the search of the minimum of a real function $f$ of $n$ variables $x_1, x_2, ..., x_n$.

**Definition 1.1** *([14]) Let be $f : \mathbb{R}^n \to \mathbb{R}$ which to all $x \in \mathbb{R}^n$,$x = (x_1, x_2, ..., x_n)^t$ associates the real value*

$$f(x) = f(x_1, x_2, ..., x_n)$$

*We are looking to solve the problem (P) :*

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

*It is therefore a question of determining a point $\hat{x}$ of $\mathbb{R}^n$ such that :*

- *1. The point $\hat{x} \in \mathbb{R}^n$ is called a global minimum solution of (P) if and only if*

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

  *Here $f(\hat{x})$ is called the global minimum value.*

- *2. The point $\hat{x} \in \mathbb{R}^n$ is called a local minimum solution of (P) if and only if there exists a neighborhood $V_\epsilon(\hat{x})$ such that*

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

*Here $f(\hat{x})$ is called minimum value.*

- *3. The point $\hat{x} \in \mathbb{R}^n$ is called a strict local minimum solution of (P) if and only if there exists a neighborhood $V_\epsilon(\hat{x})$ such that*

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

*Here $f(\hat{x})$ is called a strict local minimum value.*

## 1.1 Descent Direction

**Definition 1.2** *([14]) Either $f : \mathbb{R}^n \to \mathbb{R}$, $\hat{x} \in \mathbb{R}^n\,, d \in \mathbb{R}^n$ is said to be the direction of descent at the point $\hat{x}$ if and only if there exists a strictly positive number $\delta > 0$ such that*

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

*Let's give a sufficient condition for $d$ to be a descent of direction .*

**Theorem 1.1** *([14]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable at the point $\hat{x} \in \mathbb{R}^n$ and $d \in \mathbb{R}^n$ one direction checking the following condition :*

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

*then $d$ is a direction of descent at the point $\hat{x}$.*

**Proof.** $f$ is differentiable at the point $\hat{x}$ then $f$ continues and $\nabla f(\hat{x})$ exists,therefore

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

so

$$\begin{aligned} f(\hat{x} + \lambda d) - f(\hat{x}) &= \lambda \nabla f(\hat{x})^T.d + \lambda \|d\| \alpha(\hat{x}, \lambda d) \\ \Rightarrow \frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} &= \nabla f(\hat{x})^T.d + \lambda \|d\| \alpha(\hat{x}, \lambda d) \\ \Rightarrow \lim_{\lambda \to 0} \frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} &= \lim_{\lambda \to 0} (\nabla f(\hat{x})^T.d + \lambda \|d\| \alpha(\hat{x}, \lambda d)) \end{aligned}$$

with

$$\alpha(\hat{x}, \lambda d) \xrightarrow[\lambda \to 0]{} 0$$

so

$$f'(\hat{x}, d) = \lim_{\lambda \to 0} \frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} = \nabla f(\hat{x})^T.d < 0$$

the limit being strictly negative, then there exists a neighborhood of zero $V(0) = ]-\delta, +\delta[$ such that

$$\frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} < 0, \forall \lambda \in ]-\delta, +\delta[$$

The relation (2.1) is particularly true for all $\lambda \in ]0, +\delta[$. We obtain the desired result by multiplying the relation (2.1) by $\lambda > 0$. ■

## 1.2 General scheme of algorithms

**Definition 1.3** *Let $d_k$ be a direction of descent at the point $x_k$ we can consider the point $x_{k+1}$ the successor of $x_k$ as follows :*

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

***Start:*** *$x_0 \in \mathbb{R}^n$, $d_0$ :*

$$\nabla f(x_0)^t.d_0 < 0$$
$$x_1 = x_0 + \lambda_0 d_0$$

*$\lambda_0$ checks:*

$$f(x_0 + \lambda_0 d_0) < f(x_0)$$

***Iteration k:*** *$(x_k, d_k)$ such that $\nabla f(x_k)^t.d_k < 0$ and $\lambda_k$ such that:*

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

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

*The choice of $d_k$ and $\lambda_k$ makes it possible to build a multitude of optimization algorithms.*

**-Example of choosing descent directions**

If we choose

$$d_k = -\nabla f(x_k),$$

with

$$\nabla f(\hat{x}_k) \neq 0,$$

we obtain the gradient method.
Of course, $d_k = \nabla f(x_k)$ is a direction of descent, indeed :

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

Also if we choose $d_k = -(H(x_k))^{-1}\nabla(x_k)$ such that:
$H(x_k)$ the Hessian matrix. $(H(x_k) \in M_{n\times n})$ ,$\nabla f(x_k)$ the gradient vector.
$(\nabla f(x_k) \in M_{n\times 1})$, we obtain the Newton method.
If the matrix $H(x_k)$ is positive definite, so

$$\nabla f(x_k)^t d_k = -\nabla f(x_k)^t(H(x_k))^{-1}\nabla f(x_k) < 0$$

**-Example of the choice of steps $\lambda_k$**

We choose $\lambda_k$ to check

$$f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k), \quad \forall \lambda \in ]0, \delta[$$

the search for a real variable $\lambda_k$, which is called linear search .

## 1.3 Results of existence and uniqueness

Before studying the properties of the solution (or solutions) of $(P)$, it is necessary to make sure of their existence. We will then give results of uniqueness.

**Definition 1.4** *We say that $f : \mathbb{R}^n \to \mathbb{R}$ is coercive if*

$$\lim_{\|x\| \longrightarrow +\infty} f(x) = +\infty$$

*Here $\|.\|$ denotes any norm of $\mathbb{R}^n$ We will denote $\|.\|_p$ $(p \in \mathbb{N})$ the norm $l_p$ of $\mathbb{R}^n$*

$$\forall x = (x_1, ..., x_n) \in \mathbb{R}^n, \quad \|x\|_p = \left[\sum_{i=1}^{n} |x_i|^p\right]^{\frac{1}{p}}.$$

*The infinite norm of $\mathbb{R}^n$ is*

$$\forall x = (x_1, ..., x_n) \in \mathbb{R}^n, \quad \|x\|_\infty = \max_{1\leq i\leq n} |x_i|.$$

**Theorem 1.2** *(**Existence**): $f : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ be proper, continuous and coercive, then $(P)$ admits at least one solution.*

**Proof.** Let $d = \inf(p)$ ; $d > +\infty$ because $f$ is proper. Let $(x_p)_{p\in\mathbb{N}} \in \mathbb{R}^n$ be a minimizing sequence, that is to say such that

$$\lim_{p\longrightarrow+\infty} f(x_p) = d$$

Let's show that $(x_p)$ is bounded.If this were not the case we could extract from this suite a sub-suite (still noted $(x_p)$) such that

$$\lim_{p\longrightarrow+\infty} \|x_p\| = +\infty$$

By coercivity of $f$, we would have

$$\lim_{p\longrightarrow+\infty} f(x_p) = +\infty$$

which contradicts the fact that

$$\lim_{p\longrightarrow+\infty} \|x_p\| = d < +\infty$$

As $(x_p)$ is bounded, we can then extract a sub-sequence from it (again noted $(x_p)$) which converges to $\overline{x} \in \mathbb{R}^n$ By continuity of $f$ , we then have

$$d = \lim_{p\longrightarrow+\infty} f(x_p) = f(\overline{x}).$$

In particular $d > -\infty$ is $\overline{x}$ a solution of the problem $(P)$. ■

**Theorem 1.3** ***(Uniqueness )*** *:$f$ : $\mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be strictly convex. Then problem $(P)$ admits at most one solution.*

**Proof.** Suppose that $f$ admits at least one minimum $m$ and are $x_1 \neq x_2$ (in $\mathbb{R}^n$) achieving this minimum :

$$f(x_1) = f(x_2) = m\ .$$

By strict convexity of function $f$, we then have:

$$f\left(\frac{x_1+x_2}{2}\right) < \frac{1}{2}(f(x_1) + f(x_2)) = m\ ;$$

This contradicts the fact that $m$ is the minimum.Therefore, $x_1 = x_2$ .Finally, we will give a criterion for a function to be strictly convex and coercive. ■

**Theorem 1.4** *Let $f$ be a function $C^1$ of $\mathbb{R}^n$ in $\mathbb{R}$. Suppose that there exists $\alpha > 0$ such that:*

$$\forall (x,y) \in \mathbb{R}^n \times \mathbb{R}^n\ (\nabla f(x) - \nabla f(y), x-y) \geq \alpha \|x-y\|^2 \tag{1.1}$$

*Then $f$ is strictly convex and coercive , in particular problem (P) admits a unique solution.*

**Proof.** the Condition (1.1) implies that $\nabla f$ is monotone and that $f$ is convex. Moreover, we have the strict convexity of $f$.

Finally, $f$ is coercive: indeed, applying the Taylor formula with integral remainder:

$$f(y) = f(x) + \int_0^1 \frac{d}{dt} f(x + t(y-x))dt = f(x) + \int_0^1 (\nabla f(x + t(y-x)), y - x)dt. \tag{1.2}$$

so

$$f(y) = f(x) + (\nabla f(x), y - x) + \int_0^1 (\nabla f(x + t(y-x)) - (\nabla f(x), y - x))dt. \tag{1.3}$$

According to (1.1), we obtain

$$f(y) \geq f(x) + (\nabla f(x), y - x) + \int_0^1 t\alpha \|x - y\|^2 dt \tag{1.4}$$

Finally

$$f(y) \geq f(x) - \|\nabla f(x)\| \|y - x\| + \frac{\alpha}{2} \|x - y\|^2. \tag{1.5}$$

Let's fix $x = 0$ for example; it is then clear that $f$ is coercive. Therefore $f$ admits a unique minimum at $x^*$ on $\mathbb{R}^n$ characterized by

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

The condition (1.1) leads us to the following definition:

**Definition 1.5** *(Elliptic function) : We say that $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is elliptic if the condition (1.1) is satisfied, i.e. $\exists \alpha > 0$ such that*

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \; (D^2 f(x)y, y) \geq \alpha \|x - y\|^2$$

*$\alpha$ is the ellipticity constant.*

**Proposition 1.1** *: A function $f:\mathbb{R}^n \longrightarrow \mathbb{R}$ twice differentiable on $\mathbb{R}^n$ is elliptic if and only if*

$$\forall (x,y) \in \mathbb{R}^n \times \mathbb{R}^n \; (D^2 f(x)y, y) \geq \alpha \|y\|^2$$

**Proof.** We use again the Taylor formula applied to the function

$$\varphi : t \rightarrow \varphi(t) = f(x+ty).$$

■

We must now give conditions to be able to calculate the or the solutions. We will try to show that this solution is the solution of certain equations, so that it will be easier to calculate it.

## 1.4 Optimality conditions

The objective function must satisfy two sets of conditions in order to have a minimum, namely, first and second-order conditions. The first-order conditions are in terms of the first derivatives, i.e., the gradient.

### 1.4.1 Necessary condition for first-order optimality

**Theorem 1.5** *either $f:\mathbb{R}^n \rightarrow \mathbb{R}$ differentiable at the point $\hat{x} \in \mathbb{R}^n$, if $\hat{x}$ is a local minimal solution, then $\nabla f(\hat{x}) = 0$.*

**Proof.** Suppose $\hat{x}$ is a local minimum solution, then

$$f(\hat{x}) \leq f(x), \forall x \in V(\hat{x}) \tag{1.6}$$

Suppose the opposite,

$$\nabla f(\hat{x}) \neq 0,$$

then $-\nabla f(\hat{x})$ is a direction of descent, then there $\exists \epsilon > 0$ such that $\forall \alpha \in ]0, \delta[$ :

$$f\left(\hat{x} + \alpha(-\nabla f(\hat{x}))\right) < f(\hat{x})$$

We set $\hat{x} + \alpha(-\nabla f(\hat{x})) = \overline{x}$ ,then

$$f(\overline{x}) < f(\hat{x})$$

So $\exists \overline{x} \in V(\hat{x})$ such that :

$$f(\overline{x}) < f(\hat{x}) \tag{1.7}$$

a contradiction between (1.3) and (1.4).
from or $f$ differentiable and

$$f(\hat{x}) \leq f(x), \forall x \in V(\hat{x})$$

So

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

■.

### 1.4.2 Necessary condition for second-order optimality

**Theorem 1.6** *Either $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at the point $\hat{x} \in R^n$, if $\hat{x}$ is a local minimum of $(P)$, then $\nabla f(\hat{x}) = 0$ and the Hessian matrix of $f$ at the point $\hat{x}$, denoted by $H(\hat{x})$, is positive semi-definite..*

**Proof.** Let be any $x \in \mathbb{R}^n$, since $f$ is twice differentiable at the point $\hat{x}$ , we will have for all $\lambda \neq 0$

$$f(\hat{x} + \lambda x) = f(\hat{x}) + \frac{1}{2}\lambda^2 x^T H(\hat{x})x + \lambda^2 \|x^2\| \alpha(\hat{x}, \lambda x), \alpha(\hat{x},\, \lambda x) \underset{\lambda \to 0}{\longrightarrow} 0$$

This implies

$$\frac{f(\hat{x} + \lambda x) - f(x)}{\lambda^2} = \frac{1}{2}x^T H(\hat{x})x + \|x^2\| \alpha(\hat{x}, \lambda x) \tag{1.8}$$

$\hat{x}$ is a local optimum, then there exists $\delta > o$ such that

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

if we take into consideration (1.5) and we go to the limit when $\lambda \to 0, \lambda \neq 0$, we get

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

■.

### 1.4.3 Sufficient condition of optimality

**Theorem 1.7** *let $f : \mathbb{R}^n \to \mathbb{R}$ twice differentiable at the point $\hat{x} \in \mathbb{R}^n$ If $\nabla f(\hat{x}) = 0$ and $H(\hat{x})$ is positive, then $\hat{x}$ is a strict local minimum from (P).*

**Proof.** $f$ being twice differentiable to the point $\hat{a}$ , we will have for everything $x \in \mathbb{R}^n$

$$f(x) = f(\hat{x}) = \frac{1}{2}(x-\hat{x})^T H(\hat{x})(x-\hat{x}) + \|x-\hat{x}\|^2 \alpha(\hat{x},(x-\hat{x})), \tag{1.9}$$

$$\alpha(\hat{x},(x-\hat{x})) \underset{x\to\hat{x}}{\to} 0, (\nabla f(\hat{x})).$$

Suppose that $\hat{x}$ is not an optimum strict local.
Then there is a sequence $\{x_k\}_{k\in N^*}$ , such as $x_k \neq \hat{x} : \forall k$ and

$$x_k \neq \hat{x} : \forall k, x_k \underset{k\to\infty}{\to} \hat{x}, f(x_k) \leq f(\hat{x}) \tag{1.10}$$

In (1.6) let's take $x = x_k$ divide everything by $\|(x-\hat{x})\|^2$ and write $d_k = \dfrac{(x_k-\hat{x})}{\|(x_k-\hat{x})\|}$ , we get

$$\frac{f(x_k)-f(\hat{x})}{\|(x_k-\hat{x})\|^2} = \frac{1}{2}d_k^T H(\hat{x})d_k + \alpha(\hat{x},(x_k-\hat{x})), \alpha(\hat{x},(x_k-\hat{x})) \underset{k\to\infty}{\to} 0. \tag{1.11}$$

(1.7) and (1.8) imply

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

on the other hand, the sequence $\{d_k\}_{k\in N^*}$ is bounded $(\|d_k\|) = 1, \forall n)$. So there is a sub continuation drtrenicn such that $\{d_k\}_{k\in N_1\subset N}$.

$$d_k \underset{k\to\infty,k\in N_1}{\to} \overline{d}.$$

Finally, when $k\to\infty, k\in N_1$, we obtain

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

The last relation and the fact that $\overline{d} \neq 0(\|\overline{d}\| = 1)$ imply that the hessian matrix $H(\hat{x})$ is not positive definite. This is in contradiction with the assumption. ∎

## 1.5 One-dimensional optimization

One-dimensional optimization (linear search) consists of finding $\lambda_k$ so as to reduce the function $f$ Sufficiently along this direction.
This "sufficient" will be quantified in the following in the description of the so called conditions of Armijo, Wolfe, Goldstein $\&$ Price (linear searches inaccurate).
But first we expose the principle of descent method:

### 1.5.1 Principle of descent method

The principle of a descent method consists in making the following iterations- brags:

$$x_{k+1} = x_k + \lambda_k d_k, \quad k > 0 \tag{1.12}$$

while ensuring the ownership

$$f(x_{k+1}) < f(x_k).$$

The vector $d_k$ is the direction of descent in $x_k$ . The scalar $\lambda_k$ is called the step of the method at iteration $k$.
We can characterize the descent directions in $x_k$ using the gradient.

**Proposition 1.2** *Let $d \in \mathbb{R}^n$ Be verifying*

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

*then $d$ is a direction of descent in $x$ .*

**Proof.** we have for $\lambda > 0$

$$f(x + \lambda d) = f(x) + \lambda \nabla f(x)^t d + \lambda \varepsilon(\lambda)$$

so if we write

$$\frac{f(x + \lambda d) - f(x)}{\lambda} = \nabla f(x)^t d + \varepsilon(\lambda)$$

we can clearly see that for $\lambda$ sufficiently small we will have

$$f(x+\lambda d) - f(x) < 0.$$

■

Or that $d$ makes with the opposite of the gradient $-\nabla f(x)$ a strict angle-smaller than $90°$:

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

All the descent directions of $f$ en $x$

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

forms an open half-space of $\mathbb{R}^n$ (illustration in Figure 1.1).

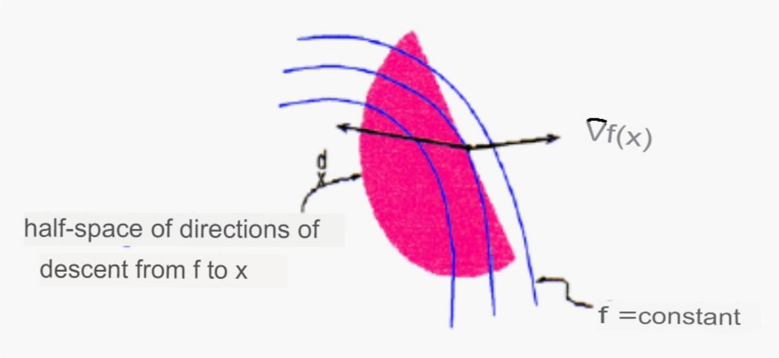


Figure 1.1: Half-space (translat ) of the descent direction $d$ from $f$ to $x$.

Such directions are interesting in optimization because to make $f$ decoist; just move along $d$. The descent-oriented methods use this idea to minimize a function In the method (1.9) the choice of $\lambda_k$ is related to the function:

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

As in the method of the direction of descent, the trajectory of the solution follows a zigzag pattern. If is chosen such that $f(x_k + d_k)$ let be the minimum in each iteration, then the successive directions are orthogonal.

Indeed

if we note $g(x) = \nabla f(x)$

$$\begin{aligned}\frac{df(x_k+\lambda d_k)}{d\lambda} &= \sum_{i=1}^{n}\frac{\partial f(x_k+\lambda d_k)}{\partial x_{ki}}\frac{d(x_{ki}+\lambda d_{ki})}{d\lambda}\\ &= \sum_{i=1}^{n} g_i(x_k+\lambda d_k)d_{ki}\\ &= \mathrm{g}(\mathrm{x}_k+\lambda d_k)^t d_k\end{aligned}$$

where $g\,(x_k+d_k)$ is the gradient at the point $x_k+d_k$.
In particular, one way to choose $\lambda_k$ may be to solve the problem optimization (with a single variable)

$$\min_{\lambda>0}\varphi(\lambda). \tag{1.13}$$

If the step $\tilde{\lambda}_k$ obtained in this way is called the optimal step then we can write:

$$\varphi'(\tilde{\lambda}_k)=\nabla f(x_k+\tilde{\lambda}_k d_k)^t d_k=0$$

that is to say

$$g(x_k+\tilde{\lambda}_k d_k)^t d_k=0$$

or else

$$d_{k+1}^t d_k=0$$

where

$$d_{k+1}=-g(x_k+\tilde{\lambda}_k d_k)=-g_{k+1}$$

is the direction of descent at the point $x_k+\tilde{\lambda}d_k$ .So the successive directions $d_k$ and $d_{k+1}$ are orthogonal as shown in Figure (1.2).

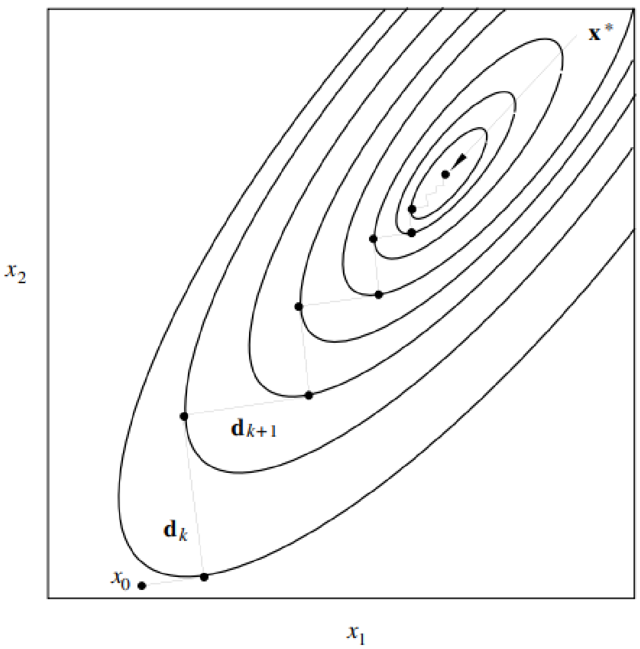


Figure 1.2: trajectory of a typical solution in a different way, in the directrion of the patient.

To define a direction of descent it is therefore necessary to specify two things:

* tell how the direction $d_k$ is calculated.This choice directly influences in the appointment of the algoritem.

* To say how we determine the step $\lambda_k$ is what we call:the search linear.

## Algorithm (method with direction of descent-one iteration)

**Step 0:** (Initialization)It is

Assumed that at the begining of iteration k, an iterated $x_k \in \mathbb{R}^n$

**Step 1:**

Stop test : if $\|\nabla f(x_k)\| \cong 0$, Stop the algorithm.

**Step 2:**

Choice of a direction of descent $d_k \in \mathbb{R}^n$

**Step 3:**

Linear search : determine a step $\lambda_k > 0$ along $d_k$ in such a way to "make $f$ decrease sufficientlly"

**Step 4:**

If the linear search is finished $x_{k+1} = x_k + \lambda_k d_k$, replace $k$ by $k + 1$ and go to step 1.

### 1.5.2 Linear Search

Performing a linear search means solving the one-dimensional problem (1.10), where the objective is to :

* Decrease f sufficiently, which most often translates into achieving an inequality of the form

$$f(x_k + \lambda_k d_k) \leq f(x_k) + "anegativeterm" \tag{1.14}$$

The negative term, let's say $\nu_k$ , plays a key role in the convergence of the algorithm using this linear search. The argument is as follows.
If $f(x_k)$ is lower bounded ($\exists c$ a constant such that $f(x_k) \geq c$ for all $k$), then $\nu_k$ must necessarily tend towards zero $(\nu_k \to 0)$. It is often from the convergence to zero of this sequence that we manage to show that the gradient itself must tend towards zero. The negative term will have to take a very particular form if we want to be able to derive information from it.
In particular, it is not enough to impose $f(x_k + \lambda_k d_k) < f(x_k)$.

* Prevent the step $A_0$ from being too small, too close to zero.
The first objective is indeed not sufficient because inequality (1.11) is generally satisfied by steps $\lambda_k > 0$ arbitrarily small.
However, this can lead to a "false convergence", that is to say the convergence of the iterates towards a non-stationary point. We give an overview in this part of the linear searches that we will use later. We have classified them into two categories

### 1.5.3 Exact Linear Searches

In this case, the optimal solution A is calculated exactly (from a theoretical point of view because in practice we generally only obtain an approximation). We will give the algorithm for linear search by dichotomy and of the golden number.

**The uncertainty interval**

**Definition 1.6** *Consider the following one-dimensional problem:*

$$\underset{\lambda \in [a,b]}{Minimize}\, \varphi(\lambda)$$

**Definition 1.7** *The interval $[a, b]$ is said to be an uncertainty interval if the minimum $\tilde{\lambda}$ of $\varphi(\lambda)$ belongs to $[a, b]$, but its exact value is not known.*

**Theorem 1.8** *let $\varphi : \mathbb{R} \to \mathbb{R}$ be strictly quasi-convex on $[a, b]$.*
*let $\lambda, \mu \in la, b[, \lambda < \mu$*

- ***1)** if $\varphi(\lambda) > \varphi(\mu)$, then $\varphi(z) \geq \varphi(\mu)$; $\forall z \in [a, \lambda]$.*
- ***2)** if $\varphi(\lambda) \leq \varphi(\mu)$, then $\varphi(z) \geq \varphi(\lambda)$;$\forall z \in [\mu, b]$.*

### *Important consequences of theorem 1.8:*

***1.** If $\varphi(\lambda) > \varphi(\mu)$, then the new uncertainty interval is: $[\lambda, b]$. (We delete $[a, \lambda[$).*
***2.** If $\varphi(\lambda) \leq \varphi(\mu)$, then the new uncertainty interval is: $\forall z \in [\mu, b]$. (We delete $[\mu, b]$).*
*This is the basic idea for the construction of optimization algorithms unidimentional without derivative calculation. At each iteration we do dimi reduce the uncertainty interval until we arrive at a final interval of length less than*

## The dichotomy method

### Algorithm of the dichotomy method

**Initialization:** Choose $\epsilon > 0$ and $l$ final length of the uncertainty interval, $[a, b]$ being the initial interval.
Set $k = 1$ (iteration counter) and go to step 1.
**Step 1:** If $b_k - a_k < \epsilon$, stop.The minimum belongs to $[a_k, b_k]$.
Otherwise ask:

$$\lambda_k = \frac{a_k + b_k}{2} - \epsilon$$
$$\mu_k = \frac{a_k + b_k}{2} + \epsilon$$

and go to step 2.
**Step 2:** If $\varphi(\lambda_k) > \varphi(\mu_k)$ then $a_{k+1} = a_k, b_{k+1} = \mu_k$.
Otherwise $a_{k+1} = \lambda_k, b_{k+1} = b_k$.
Replace $k$ with $k + 1$ and go to step 1.

## The golden number method

The golden number method improves the dichotomy method, in dimireducing the number of observations, at each iteration.

## Algorithm of The golden number method:

**Initial step:** choose $l > 0$ final length of the uncertainty interval and $[a_1, b_1]$, $\alpha = 0,618$,calculate $\lambda_1$ and $\mu_1$ such that:

$$\lambda_1 = a_1 + (1-\alpha)(b_1 - a_1).$$
$$\mu_1 = a_1 + \alpha(b_1 - a_1).$$

Set $k = 1$ and go to the main step.
**Main step:**
**(1)** If $b_k - a_k < l$ stop, take $\alpha^* \in [a_k, b_k]$. If $\varphi(\lambda_k) > \varphi(\mu_k)$ go to (2), otherwise go to (3).
**(2)** Ask $a_{k+1} = \lambda_k, b_{k+1} = \mu_k$, $\mu_{k+1} = a_{k+1} + (b_{k+1} - a_{k+1})$, calculate $\varphi(\mu_{k+1})$, and go to (4).
**(3)** Ask $a_{k+1} = a_k, b_{k+1} = \mu_k, \mu_{k+1} = \lambda_k, \lambda_{k+1} = a_{k+1} + (1-\alpha)(b_{k+1} - a_{k+1})$, calculate $\varphi(\lambda_{k+1})$ and go to (4).
**(4)** Set $k = k + 1$, and go to (1).

### 1.5.4 Inexact Line Searches

Exact linear searches, despite the fact that they only lead to an approximate optimal solution, they do not require a lot of observations at each iteration of the main algorithm .In the 60s, 70s, 80s, math scientists have succeeded in developing linear research that is less expensive, but at the same time respects the descent of the function.
Let us now describe in detail the three most inaccurate linear searches more important. It is about the inexact linear searches of Armijo, of Goldstein and de Wolfe.

### 1.5.5 Inexact Line Searches of Armijo (1966)

Let $f : \mathbb{R}^n \to \mathbb{R}, x_k \in \mathbb{R}^n, d_k \in \mathbb{R}^n$ a direction of descent $(\nabla f(x_k)^t d_k < 0)$.
The rule of Armijo requires that $f$ decreases sufficiently to the point $x_k + \lambda_k d_k$ .This condition is described by the following inequality called condition of armijo:

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \epsilon \lambda_k \nabla f(x_k)^t d_k, \epsilon \in ]0.1[ \tag{Armijo}$$

That is to say that the reduction of $f$ must be proportional at the same time to $\lambda_k$ and to the directional derivative $\nabla f(x_k)^t d_k$.

### 1.5.6 Graphical interpretation of the Armijo condition

Let's define the function

$$\varphi : \mathbb{R} \to \mathbb{R}$$

by

$$\varphi(\lambda_k) = f(x_k + \lambda_k d_k), \lambda_k \geq 0$$

Note that:

$$\varphi^{'}(\lambda) = \nabla f(x_k + \lambda_k d_k)^t d_k,$$
$$\varphi^{'}(0) = \nabla f(x_k)^t d_k < 0,$$
$$\varphi(0) = f(x_k).$$

The equation of the tangent at the point $(0, \varphi(0))$ is as follows:

$$\{\lambda, y\} : y = \varphi(0) + \varphi^{'}(0)(\lambda - 0)$$
$$\widetilde{\varphi}(\lambda_k) = f(x_k) + \nabla f(x_k)^t d_k \lambda_k$$

Let's Pose

$$\widetilde{\varphi}(\lambda) = \varphi(0) + \varphi^{'}(0)\lambda$$

The equation of the tangent becomes:

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

Now let's define the function $\hat{\varphi}(\lambda)$ as follows:

$$\hat{\varphi}(\lambda) = \varphi(0) + \epsilon\lambda\varphi^{'}(0) = f(x_k) + \epsilon\lambda\nabla f(x_k)^t d_k, \epsilon \in [0, 1[$$

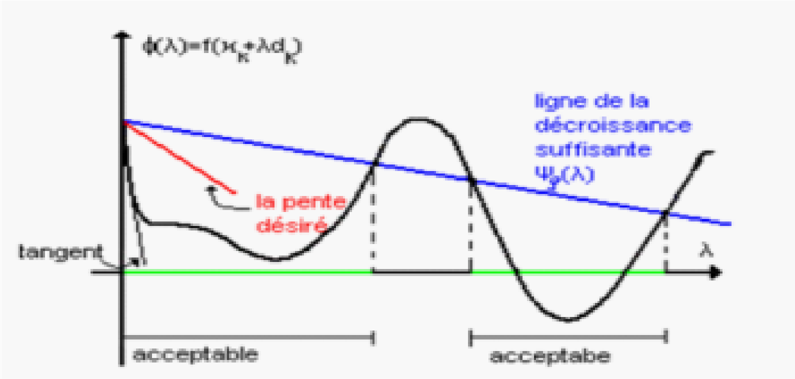


Figure 1.3: Armijo Rule.

We are looking $\overline{\lambda}_k$ for such that

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

**Remark 1.1**

- ***1***-*The condition* $\varphi(\overline{\lambda}_k) \leq \hat{\varphi}(\overline{\lambda}_k)$ *implies the decrease of the function* $F$.

- ***2***-*Indeed*

$$\begin{gathered}\varphi(\overline{\lambda}_k) \leq \hat{\varphi}(\overline{\lambda}_k)\\ f(x_k + \overline{\lambda}d_k) \leq f(x_k) + \epsilon\overline{\lambda_k}\nabla f(x_k)d_k < f(x_k)\end{gathered}$$

*because the direction d is a direction of descent.*

- ***3***- *When we take* $\overline{\lambda_k}$ *very close to zero it will harm the convergence and the speed of convergence. Indeed*

$$\begin{gathered}f(x_k + \overline{\lambda}d_k) = f(x_k) + \overline{\lambda_k}\nabla f(x_k)d_k + \overline{\lambda_k}\alpha(x_k, \tilde{\lambda}d_k)\\ f(x_k + \overline{\lambda}d_k) - f(x_k) = \tilde{\lambda_k}[\nabla f(x_k)d_k + \alpha(x_k, \overline{\lambda}d_k)]\end{gathered}$$

*if*

$$\begin{gathered}\overline{\lambda_k} \longrightarrow 0\\ \underset{\bar{\lambda}_k \longrightarrow 0}{\alpha(x_k, \overline{\lambda}d_k)} \longrightarrow 0\end{gathered}$$

*so*

$$f(x_k + \overline{\lambda_k}d_k) \simeq f(x_k).$$

### 1.5.7 Algorithm (Armijo's Rule)

**Step 0:(Initialization)**

$\alpha_{g,1} = \alpha_{d,1} = 0$, choose $\alpha_1 > 0, \rho \in ]0, 1[$ set $k = 1$ and go to step 1.

**Step 1:**

if $\varphi_k(\alpha_k) \leq \varphi_k(0) + \rho\varphi_k^{'}(0)\alpha_k$ : STOP $(\alpha^* = \alpha_k)$.

if $\varphi_k(\alpha_k) > \varphi_k(0) + \rho\varphi_k^{'}(0)\alpha_k$, then

$\alpha_{d,k+1} = \alpha_d, \alpha_{g,k+1} = \alpha_k$ and go to step 2.

**Step 2:**

if $\alpha_{d,k+1} = 0$ determine $\alpha_{k+1} \in ]\alpha_{g,k+1}, +\infty[$

if $\alpha_{d,k+1} \neq 0$ determine $\alpha_{k+1} \in ]\alpha_{g,k+1}, \alpha_{d,k+1}[$

replace $k$ with $k + 1$ and go to step $1$ .

**Remark 1.2** *It is clear from Figure 1.3- Armijo's Rule that the armijo equality is always checked if:* $\alpha_k \succ 0$ *is small enough. indeed, in the opposite case, we would have a sequence of strictly positive* $\{\alpha_{k,i}\}_{i\succeq 1}$ *converging to 0 when* $i \to \infty$ *and such that*

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

*does not take place for* $\alpha_k = \alpha_{k,i}$.
*By subtracting* $f(x_k)$ *in the two members, dividing by* $\alpha_{k,i}$ *and by passing to the limit when* $i \to \infty$*, we would find*

$$\nabla^T f(x_k) d_k \geq \rho \nabla^T f(x_k) d_k$$

*which would contradict the fact that* $d_k$*, is a direction of descent* $(\rho < 1)$.

**Theorem 1.9** *If* $\varphi_k : \mathbb{R}_+ \to \mathbb{R}$ *, defined by* $\varphi_k(\alpha) = f(x_k + \alpha d_k)$ *is continuous and bounded on the outside, if* $d_k$ *is a direction of descent in* $x_k(\varphi_k^{'}(0) < 0)$ *and if* $\rho \in ]0, 1[$*, then the set of steps verifying the rule d'armijo is not empty.*

**Proof.** We have

$$\varphi_k(\alpha) = f(x_k + \alpha d_k)$$
$$\Psi_\rho(\alpha) = f(x_k) + \rho\alpha_k \nabla^T f(x_k) d_k$$

The Taylor-Yong expansion in $\alpha = 0$ of $\varphi_k$ is:

$$\varphi_k(\alpha) = f(x_k + \alpha d_k) = f(x_k) + \rho\alpha_k \nabla^T f(x_k) d_k + \alpha\xi(\alpha)$$

where

$$\xi(\alpha) \to 0, \alpha \to 0$$

and as $\rho \in ]0, 1[$ and $\varphi_k^{'}(0) = \nabla^T f(x_k) d_k < 0$ we deduce:

$$f(x_k) + \alpha_k \nabla^T f(x_k) d_k < f(x_k) + \rho\alpha_k \nabla^T f(x_k) d_k$$

for $\alpha > 0$ We see that for $\alpha > 0$ quite small we have:

$$\varphi_k(\alpha) < \Psi_\rho(\alpha)$$

From the above and the fact that $\varphi_k$ is bounded inferiorly, and

$$\Psi_\rho(\alpha) \to -\infty, \, (\alpha) \to +\infty,$$

we deduce that the function $\Psi_\rho(\alpha) - \varphi_k(\alpha)$ at the property:

$$\begin{cases} \Psi_\rho(\alpha) - \varphi_k(\alpha) \succ 0 \text{ for a small enough} \\ \Psi_\rho(\alpha) - \varphi_k(\alpha) \prec 0 \text{ for a large enough} \end{cases}$$

so cancels at least once for $\alpha > 0$:
By choosing the smallest of these zeros we see that there are $\bar{\alpha} > 0$ such that

$$\varphi_k(\bar{\alpha}) = \Psi_\rho(\bar{\alpha}) \text{ and } \varphi_k(\alpha) < \Psi_\rho(\alpha) \text{ for } 0 < \alpha < \bar{\alpha}.$$

Which completes the demonstration. ■

## Goldstein's Inexact linear Search (1967)

The step $\lambda_k$, is acceptable by the inexacte linear Goldestein search, if it satisfies the following two Goldesteinlet Goldestein2 conditions:

$$f(x_k + \lambda_k d_k) \leq f(x_k) + c\lambda_k (x_k)^t . d_k c \in ]0, \frac{1}{2}[ \qquad \text{(Goldstein1)}$$

$$f(x_k + \lambda_k d_k) \geq f(x_k) + (1-c).\lambda_k . \nabla f(x_k)^t . d_k \qquad \text{(Goldstein2)}$$

**Interpretation of the Goldstein1 relationship:**
The Goldsteinl condition is exactly the Armijo condition studied. yes, of course.This condition ensures a sufficient decrease in the function $f$.
**Interpretation of the Goldstein2 relationship:**
provided that Goldstein2 avoids at step $\lambda_k$ being too small (see the figure ssous) .This is a great contribution to the convergence process.

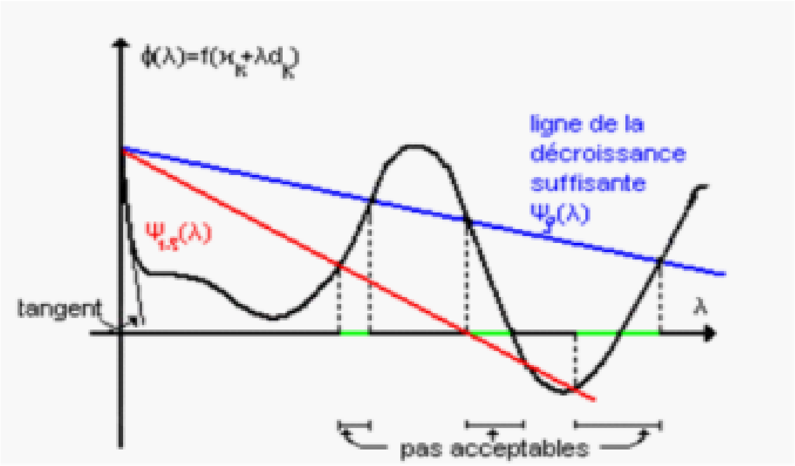

$\phi(\lambda)=f(x_k+\lambda d_k)$
ligne de la décroissance suffisante $\psi_\beta(\lambda)$
$\psi_{1,\beta}(\lambda)$
tangent
$\lambda$
pas acceptables

Figure 1.4: Goldstein rule.

Figure 1.5 shows, on an example, all the points satisfying
the two Goldstein conditions.

## The Goldstein Algorithm:

The Algorithm tries to find $\lambda_k \in ]\beta_1, \beta_2[$. We start with an intervalle $[a_0, b_o]$ quite large. We take $\lambda_0 \in ]\beta_1, \beta_2[$:

- if $\lambda_0$ checked Goldsteinl and Goldstein2 then $\lambda_k \in ]\beta_1, \beta_2[$ and we stop.

- If $\lambda_0 > \beta_1$, then $\lambda_0$ is not Goldstein, then we take $b_1 = \lambda_0$ and$a_1 = bo$ and $\lambda_1 = \dfrac{a_1 + b_1}{2}$ and we start again with $\lambda_1$.

- If $\lambda_0 < \beta_1$ then $\lambda_0$ is not Goldstein2, we take $a_1 = \lambda_0$,$b_1 = b_0$ and $\lambda_1 = \dfrac{a_1 + b_1}{2}$ and we test $\lambda_1$ again.

## At iteration k

Suppose we have $[a_k, b_k]$ and $\lambda_k = \dfrac{a_k + b_k}{2}$

If $\lambda_k$ checked Goldstein1 and Goldstein2; $\lambda_k \in ]\beta_1, \beta_2[$. Stop.

If $\lambda_k$ is not Goldstein1 then $\lambda_k > \beta_2$

We take $b_{k+1} = \lambda_k; a_k + 1 = a_k; \lambda_{k+1} = \dfrac{a_k + 1 + b_{k+1}}{2}$.

If $k$ is not Goldstein2 then $\lambda_k < \beta_1$ . We take $a_{k+1} = \lambda_k; b_{k+1} = b_k; \lambda_{k+1} = \dfrac{a_{k+1} + b_{k+1}}{2}$.

The following algorithm is thus obtained :

## Algorithm from Goldstein

**STEP 1 (Initialization )**

Choose $\alpha_0 \in [0, 10^{100}]$ and $\rho \in 0, 1[$. Ask $a_0 = 0, b_0 = 10^{100}$

Set $k = 0$ and go to STEP 2.

**STEP 2 (Goldsteinl Test)**

Iteration $k$ we have $[a_k, b_k]$ and $a_k$, calculate $\varphi_k(\alpha_k)$

If $\varphi_k(\alpha_k) \leq \varphi_k(0) + \rho\alpha_k\varphi_k^{'}(0)$, go to STEP 3.

Otherwise

Ask $a_{k+1} = \alpha_k, b_{k+1} = b_k$, and go to STEP 4

**STEP 3 (Gold Test 02)**

if $\varphi_k(\alpha_k) \geq \varphi_k(0) + (1-\rho)\alpha_k\varphi_k^{'}(0)$, stop. $\alpha^* = \alpha_k$

Otherwise

Ask $a_{k+1} = \alpha_k, b_{k+1} = b_k$ and go to STEP 4

**STEP 4**

Pose $\alpha_{k+1} = \dfrac{a_{k+1} + b_{k+1}}{2}$.

Set $k = k + 1$ and go to STEP 2.

**Wolfe's inexact linear Search (1969)**

## <u>Inexact linear weak Wolfe search</u>

The step $\lambda_k$ is acceptable by Wolfe's inexact linear search or Wolfe simply, if it satisfies the following two conditions low :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + c_1\lambda_k\nabla^t f(x_k).d_k, \;\; c_1 \in ]0, 1[ \tag{Wolf1}$$

$$\nabla f(x_k + \lambda_k d_k)^t \geq c_2\nabla f(x_k)^t.d_k, \;\; c_2 \in ]c_1, 1[ \tag{Wolf2}$$

**Interpretation of the Wolf1 relationship**

the Wolf1 condition is exactly Armijo's condition, this condition ensures a sufficient decrease in the function $f$.

**Interpretation of the Wolf2 relationship**

The selected $\lambda_k$ by the Wolf1 condition can be very small. This can have disastrous consequences on the convergence of the algorithm. The condition Wolf2 avoids this drawback and removes very small values from $\lambda_k$. (see the fugire below).

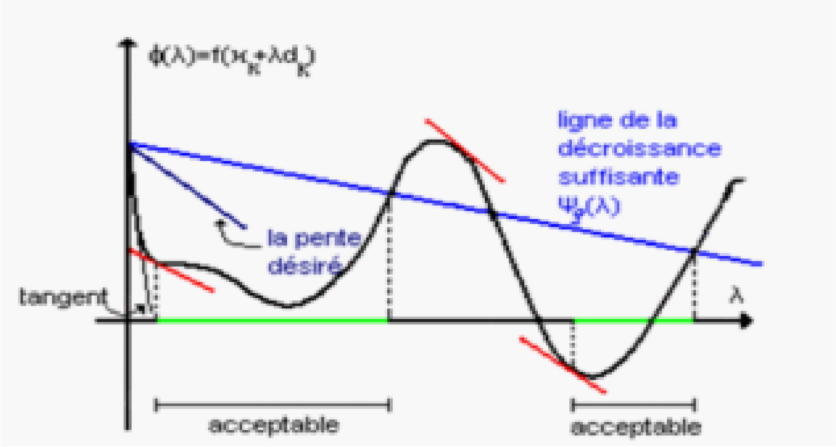


Figure 1.5: Wolfe rule.

Figure 1.5 shows on an example the set of points satisfying Wolfe's conditions $c-1=0.1; c_2=0.7$ (Lemarechal 1980).

**Inexacte linear strong Wolfe search**

The step $\lambda_k$ is acceptable by Wolfe's inaccurate linear search ,

if it satisfies the following two Wolfe fort1 and Wolfe fort2 conditions :

$$f(x_k+\lambda_k d_k) \le f(x_k) + c_1\lambda_k \nabla^t f(x_k) d_k, c_1 \in ]0,1[ \qquad \text{(Wolfe fort1)}$$

$$|\nabla f(x_k+\lambda_k d_k)^t.d_k| \ge c_2.|\nabla f(x_k)^t.d_k|, c_2 \in ]c_1,1[ \qquad \text{(Wolef fort2)}$$

## Interpretation of the Wolfe fort1 relationship

The Wolfe fort1 condition is exactly the Wolfe1 or Armijo condition.

This condition ensures a sufficient decrease in the function $f$.

## Interpretation of the Wolfe fort2 relationship

The Wolf fort2 condition implies Wolf2. The step $\lambda_k$ selected by the Wolf1 and Wolf2 conditions may be very far from an optimal point or stationary of the $\varphi$ function . The Wolf fort2 condition ensures that the pitch has $\lambda_k$ is in the vicinity of a stationary point or an optimal point of $\varphi$ .

## Wolfe's algorithm

### STEP 1 (Initialization)

Take $\alpha_0 \in [0, 10^{90}]$ , calculate $\varphi(0), \varphi_0'(0)$. Take $\rho = 0.1$ (or $\rho = 0,1$ or $\rho = 0.001$ or $\rho = 10^{-4}$) $\theta = 0.9$ (or even smaller )

Set $\alpha_0 = 0, b_0 = 10^{99}, k = 0$ and go to STEP 2

**STEP 2 (test of (Wolfe1))**

Calculate $\varphi(\alpha_k)$. If $\varphi(\alpha_k) \leq \varphi(0) + \rho\alpha_k\varphi^{'}(0)$, go to STEP 3. Otherwise. Ask $a_k + 1 = a_k, b_k + 1 = \alpha_k$ and go to STEP 4

**STEP 3 (test (Wolfe2) or (Wolfeforte2) )**

Calculate $\varphi^{'}(\alpha_k)$. If $\varphi^{'}(\alpha_k) \geq \theta\varphi^{'}(0)(|\varphi^{'}_k(\alpha_k)| \leq -\theta\varphi^{'}(0))$. STOP

To take $\tilde{\alpha} = \alpha_k$. Otherwise Ask $a_k + 1 = a_k, b_k + 1 = b_k$ and go to STEP4

**STEP 4 (calculation of $\alpha_{k+1}$)**

$\alpha_{k+1} = \dfrac{a_{k+1} + b_{k+1}}{2}$

Set $k = k + 1$ and go to STEP 2.

## 1.6 Convergence of methods.

### 1.6.1 The Zoutendijk condition

Now we will study the contribution of the inaccurate linear search in the convergence of algorithms with descending directions. It's only a contribution, because linear research alone cannot ensure the convergence of iterates . It is well understood that the choice of the direction of decente also plays a role. This translates into a so called Zoutendijk condition, from which we can draw some interesting qualitative information.

An inaccurate linear search rule is said to satisfy the condition Zoutendijk if there exists a constant $C > 0$ such that for any index $k \leq 1$ we have from

$$f(x_{k+1}) \leq f(x) - C\|\nabla f(x_k)\|^2 \cos^2\theta_k \tag{1.15}$$

where $\theta_k$ is the angle that $d_k$ makes with $-\nabla f(x_k)$, defined by

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

(1.16)

Here is how we use the condition condition from Zoutendijk.

**Theorem 1.10** ***(from Zoutendijk)***

If the sequence $\{x_k\}$ generated by an optimization algorithm verifies the conditiontion of Zoutendijk (1.12) and if the sequence $f(x_k)\}$ is reduced, then

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

**Proof.** By summing the quantities $\|\nabla f(x_k)\|^2\cos^2\theta_k$ while pretaking into consideration (1.13), we have

$$\sum_{k\geq 1}^{l}\|\nabla f(x_k)\|^2 cos^2\theta_k\leq\frac{1}{C}(f(x_1)-f(x_{l+1}))\tag{1.17}$$

The series is thus convergent since there exists a constant $C"$ such for all $k$, $f(x_k)\geq C"$. ■

**Important consequence of Zoudentijk's theorem**

The condition (1.14) implies

$$\|\nabla f(x_k)\|^2\cos^2\theta_k\to 0\,(k\to\infty)\tag{1.18}$$

This limit can be used to deduce the convergence of the algorithm.
Indeed, if our algorithm generates a sequence $\{x_k\}$ of the form :

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

If the choice of $d_k$ is such that

$$\cos\theta_k\geq\delta>0,\forall k$$

then it follows from (1.15) that

$$\lim\|\nabla f(x_k)\|=0$$

The following two proposals specify the circumstances in which the condition of Zoutendijk (1.12) is verified with the rules of Armijo and Wolf.

**Proposition 1.3** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function continuously differentciable in a neighborhood of*

$$T = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}.$$

*We consider an algorithm with descent directions $d_k$, which generates a following $\{x_k\}$ using Armijo's linear search, with*

$$\alpha_1 > 0$$

*Then there exists a constant $C > 0$ such that, for any $k \geq 1$, one of the conditions*

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

*or*

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

*is verified.*

**Proposition 1.4** *Either $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function in a neighborhood of*

$$T = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}.$$

*We consider an algorithm with descent directions $d_k$, which generates a continued $\{x_k\}$ using the Wolfe linear search (Wolfe1) and (Wolfe2). Then there remains a constant $C > 0$ such that, for any $k \geq 1$, the condition of Zoutendijk (1.12) is verified.*

### 1.6.2 Global convergence

**Definition 1.8** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable . Suppose that we built a sequence $\{x_k\}$, using an optimization algorithm without constraints described in the model (algorithm model). We will say that the algorithm converges globally if we have :*

$$\lim_{k\to\infty} \inf \|\nabla f(x_k)\| = 0$$

**Remark 1.3** *Authors sometimes require for the same definition the next stronger relationship :*

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$$

### 1.6.3 Notion of convergence speed

The global convergence of an algorithm having been established, we are now interested in evaluating its effectiveness. From a practical point of view, the effectiveness of an algorithm depends on the number of necessary iterations to obtain an approximation to within $\epsilon$ ($\epsilon$ fixed in advance) of the optimum $x^*$.

If we compare between them, several algorithms, and if we admit that the calculation time per iteration is approximately the same for all, the best is the one that will require the smallest number of iterations.

Unfortunately, it turns out to be impossible to draw general conclusions of this kind of comparison.

Depending on the chosen starting point, the nature of the function to be optimized, the value of the chosen tolerance, the hierarchy of the algorithms may vary considerably.

If we want to identify a criterion having a certain absolute value, we must therefore resort to another type of analysis: this is the object of the study of the asymptotic convergence, that is to say of the behavior of the sequence $\{x_k\}$ in the vicinity of the limit point $x^*$.

This leads to assigning to each algorithm an efficiency index called its speed of convergence.

**Remark 1.4** *we are once brought to express the convergence speed of $\{x_k\}$ sequel by studying, not the way $\|x_k - x^*\|$ tends to 0, but the way the sequence $\{f(x)\}$ tends to $f(x^*)$ where f the function that we minimized.*

# Chapter 2

# Gradinet and conjugate gradient methods

## 2.1 Gradient method (Steepest-descent method)

This method was discovered by Cauchy in 1847 ([10]). It is natural to wonder about the origin or justification of such an appellation (steepest slope method). Let us consider a point $x \in \mathbb{R}^n$, if $\nabla f(x_k) \neq 0$, then the direction $d_k = -\nabla f(x_k)$ is a direction of descent (see Theorem 4.1[45] and remark 4.1[45]). The following Theorem goes show us that this is actually the best direction of descent. In other words the decrease of the function will be the strongest following the direction: $-\nabla f(x_k)$.

**Theorem 2.1** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$, is differentiable at point x, and suppose that $\nabla f(x_k) \neq 0$. Let's consider the optimal problem*

$$\underset{\|d\| \leq 1}{Minimize}\ f'(x, d)$$

*where $f'(x, d)$ is the directional derivative of $f$ at the point $x$ and in the direction $d$. Then the optimal solution of this problem is given by*

$$\tilde{d} = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$$

**Proof.** Since

$$f'(x, d) = \lim_{\lambda \to 0_+} \frac{f(x + \lambda d) - f(x)}{\lambda} = \nabla f(x)^t d.$$

Our problem therefore amounts to minimizing $\nabla f(x)^t d$ in $\{d : \|d\| \leq 1\}$. The shwartz inequality gives

$$|\nabla f(x)^t d| \leq \|\nabla f(x)\| \|d\|.$$

so

$$\nabla f(x)^t d \geq 0,$$

we have of course

$$-\nabla f(x)^t d \leq \|\nabla f(x)\| \|d\|.$$

If

$$\nabla f(x)^t d \leq 0,$$

(3.1) implies that

$$-\nabla f(x)^t d \leq \|\nabla f(x)\| \|d\|.$$

Therefore we always have

$$\nabla f(x)^t d \geq -\|\nabla f(x)\| \|d\|.$$

For $\|d\| \leq 1$, we have

$$\|\nabla f(x)\| \|d\| \leq \|\nabla f(x)\| \Rightarrow -\|\nabla f(x)\| \|d\| \geq -\|\nabla f(x)\|.$$

So :$\forall d : \|d\| \leq 1$ we have

$$\nabla f(x)^t d \geq -\|\nabla f(x)\|$$

On the other hand,: $\|\tilde{d}\| = 1$ and $d$ verifies:

$$\nabla f(x)^t \tilde{d} = \nabla f(x)^t (-\frac{\nabla f(x)}{\|\nabla f(x)\|}) = -\|\nabla f(x)\|.$$

■

**Interpretation of the Theorem 3.0.2 [45] :**

We will start from theorem 7.1[45] to give an intuitive idea about the call: method of the highest slope. Indeed, according to the theorem 7.1 [45] we have :

$$f'(x,d) \geq f'(x,\tilde{d}) : \forall d, \ \|d\| \leq 1$$

Either by using the definition of the directional derivative

$$\underset{\lambda \to 0_+}{lim} \frac{f(x+\lambda d) - f(x)}{\lambda} \geq \underset{\lambda \to 0_+}{lim} \frac{f(x+\lambda \tilde{d}) - f(x)}{\lambda}$$

This last inequality implies that there exists $\delta > 0$ such that

$$[f(x+\lambda d) - f(x)] - [f(x+\lambda \tilde{d}) - f(x)] \geq 0, \quad \forall \lambda \in ]-\delta, +\delta[$$

or again

$$f(x+\lambda d) \geq f(x+\lambda \tilde{d}), \forall \lambda \in ]-\delta, +\delta[ \quad \text{and} \quad \forall d, \ \|d\| \leq 1.$$

### 2.1.1 Algorithm of the steepest slope method

This algorithm is very simple. It follows the following scheme.

**Algorithm of the steepest slope method**

▶ **Initial step** :

$Choose \epsilon > 0$. Choose an initial point $x_1$. Put $k = 1$ and go to the main stage.

▶ **Main step** :

$If \, \|\nabla f(x)\| < \epsilon$ stop. Otherwise set $d_k = -\nabla f(x_k)$ and let the optimal solution of linear search

$$Min \, \{f(x_k, +\lambda d_k); \lambda \geq 0\}.$$

Pose $x_{k+1} = x_k + \lambda_k d_k$. Replace $k$ with $k + 1$ and repeat the main step.

### 2.1.2 Disadvantages of the steepest slope method

**Slowness of the method in the vicinity of stationary points**

This method works efficiently in the first steps of the algorithm. Unfortunately, as soon as we approach the stationary point, the method becomes very slow. We can intuitively explain this phenomenon by the following considerations

$$f(x_k, +\lambda d) = f(x_k) + \lambda \nabla f(x_k)^t d + \lambda \|d\| \alpha(x_k; \lambda d)$$

where $\alpha(x_k; \lambda d) \to 0$ when $\lambda d \to 0$.

If $d = -\nabla f(x_k)$, we obtain :$x_{k+1} = x_k - \lambda \nabla f(x_k)$ and consequently

$$f(x_{k+1}) - f(x_k) = \lambda[-\|\nabla f(x_k)\|^2 + \|\nabla f(x_k)\| \alpha(x_k; \lambda \nabla f(x_k))]$$

From the previous expression, it can be seen that when irg approaches a stationary point, and if $f$ is continuously differentiable, then $\|\nabla f(x_k)\|$ is close to zero. Done the term to the right approaches zero, independently of $\lambda$, and consequently $f(x_{k+1})$ does not move away not a lot of $f(x_k)$ when we go from the point $x_k$, to the point $x_{k+1}$

**The phenomenon of Zigzagging**

It is not easy to verify that for the gradient method we always have

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

that is to say that the sequence $\{x_k\}$ generated by the algorithm of the gradient method,zigzag. This creates a phenomenon of slowing down in the routing of the points $x_K$ towards the optimal solution.

### 2.1.3 Some remedies

**Change of direction**
Instead of taking as the direction of descent, the direction :

$$d_k = -\nabla f(x_k),$$

we take directions of the form

$$d_k = -D.\nabla f(x_k),$$

where $D$ is a suitably chosen matrix ($D$ could be, for example, the inverse of the Hesian matrix at the point $x_k$, that is to say $(H(x_k))^{-1}$).
Another choice could be made in the following way :

$$d_k = -\nabla f(x_k) + g_k,$$

where $g_k$, is an appropriate vector.
**Acceleration of convergence**
We can also accelerate the convergence of the gradient method. For this we trans- forms, thanks to an algorithm for accelerating convergence, the sequence $\{x_k\}$ into a sequence $\{y_k\}$ which would converge towards the same limit as the following $\{x_k\}$, but would converge more quickly-dement. If we denote by $x^*$this limit comments, we express this rapidity by the limit next :

$$\lim_{k \longrightarrow \infty} \frac{y_k - x^*}{x_k - x^*} = 0$$

**Example 2.1** *Let the following quadratic function be:* $f(x) = \frac{1}{2}x^t Ax - b^t x$ *with* $A > 0$ *(that is,* $A$ *is a positive definite matrix), we note* $g(\rho) = f(x_k + \rho d_k)$*, where the optimal* $\rho_k$*, is characterized by* $g'(\rho_k) = 0$ *so we have*

$$\nabla f(x_k + \rho_k d_k)^t d_b = (A(x_k + \rho d_k) - b)^t d_k = 0$$

*Either*

$$\nabla f(x_k + \rho_k A d_k)^t d_b = 0 \Rightarrow \rho_k = -\frac{((x))^t.d_k}{d_k^t.Ad_k} > 0$$

*because $d_k$ is a direction of descent and $d_k^t Hasd_k > 0$.*
*The optimal step gradient method can be written as: $x_{k+1} = x_k + \rho_k d_k$ with*

$$d\begin{cases} {}_k = b - Ax_k & (2.1)\\ \rho_k = \dfrac{d_k^T.d_k}{d_k^T.A} & (2.2)\end{cases}$$

## 2.2 Conjugate gradient method

This method is mainly used for large problems. This method was discovered in 1952 by Hestenes and Steifel ([32]), for the minimization of functions strictly convex quadratics. Several mathematicians have extended this method for the nonlinear case. This has been made for the first time, in 1964 by Fletcher and Reeves ([26]) (Fletcher's method- Reeves) then in 1969 by Polak, RibiÃ¨re ([45]) and Polyak ([41]) (Polak-RibiÃ¨re method- Polyak). Another variant was studied in 1987 by Fletcher ([29]) (Method of the conjugated descent). Let's mention other new algorithms that can be found in ([18], [6], [38], [34], [23],[37], [52], [20], [13], [4], [20], [2], [24], [23], [5], [53])

### 2.2.1 Quadratic optimization without constraints

**Definition 2.1** *Let $Q$ be a symmetric and positive definite matrix $(n; n)$ and $b \in \mathbb{R}^n$. We call quadratic minimization problem without constraints, the problem noted $(PQSC)$ next :*

$$\left\{ \min_{x\in\mathbb{R}^n} \frac{1}{2}x^T Qx - b^T x \right\} \tag{2.3}$$

**Theorem 2.2** *The problem $(PQSC)$ has a unique solution $\hat{x}$,*
*solution of the linear system $Qx = b$, that is to say that $\hat{x}$ verifies*

$$\hat{x} = Q^{-1}b \tag{2.4}$$

### 2.2.2 Calculation of the pitch obtained by an exact linear search

Let Q be a symmetric and positive definite $(n, n)$ matrix and $b \in \mathbb{R}^n$ and

$$f(x) = \frac{1}{2}x^TQx - b^Tx.$$

Consider the problem (PQSC)

$$\underset{x\in\mathbb{R}^n}{min} f(x) = \underset{x\in\mathbb{R}^n}{min} \left\{ \frac{1}{2}x^TQx - b^Tx \right\} \tag{PQSC}$$

The methods with linear research directions generate sequences $\{x_k\}_{k=1,2,\dots}$ of the following way. We start with $x_1 \in \mathbb{R}^n$. At iteration $k$, if we have $x_k \in \mathbb{R}^n$, the successor $x_{k+1}$ of $x_k$ is given by the following relation

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.5}$$

on $d_k \in \mathbb{R}^n$ is a search direction and $\alpha_k \in \mathbb{R}^+$ is the search step obtained by an exact or inaccurate linear search. In the case of an exact linear search $\alpha_k$ check

$$f(x_k + \alpha_k d_k) = \min_{\alpha>0} f(x_k + \alpha d_k) \tag{2.6}$$

Let's note

$$g_k = \nabla f(x_k) = Qx_k - b \tag{2.7}$$

**Theorem 2.3** *Let $Q$ be a symmetric and positive definite $(n, n)$ matrix and $b \in \mathbb{R}^n$ and*

$$f(x) = \frac{1}{2}x^TQx - b^Tx. \tag{2.8}$$

*Consider the problem (PQSC)*

$$\min_{x\in\mathbb{R}^n} f(x) = \min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}x^TQx - b^Tx \right\} \tag{PQSC}$$

*Suppose that at iteration $k$ we have a direction $d_k$, of descent, that is to say that $d_k$, verifies*

$$g_k^T d_k = (Qx_k - b)^T d_K < 0 \tag{2.9}$$

*let $\alpha_k > 0$ be obtained by an inaccurate linear search, that is to say that $\alpha_k$ verifies*

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

*so*

$$\alpha_k = -\frac{g_k^T d_k}{d_K^T Q d_K} \tag{2.10}$$

### 2.2.3 Conjugate directions method

**Definition 2.2** *Let $Q$ be a symmetric $(n, n)$ matrix. The directions $d_0, d_1, ....d_k$ are said to be $Q$ conjugates if we have*

$$d_i^T Q d_j = 0, 0 \leq i,\, j \leq k \tag{2.11}$$

**Theorem 2.4** *Let $Q$ be a symmetric and positive definite matrix $(n; n)$. If the directions $d_0, d_1, ....d_k$; with $k \leq n - 1$; are non-zero and $Q$ conjugates, then they are linearly independent.*

### 2.2.4 The Algorithm of conjugate directions

Let $Q$ be a symmetric and positive definite $(n; n)$ matrix and $b \in \mathbb{R}^n$. Consider the problem of quadratic minimization without constraints, $(PQSC)$, according to :

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

**Algorithm of conjugate directions**

***Initialization**

We give ourselves any $x_0 \in \mathbb{R}^n$ and $(d_0, d_1, ....d_{n-1})$ , $Q$ conjugates. Set $k = 0$ and go to the **main step**

***Main step**

For $k \geq 0$

Calculate

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

If $g_k = 0$. Stop.
Otherwise calculate

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

Set

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

ask $k = k + 1$ and go to the **main step**.

**Theorem 2.5** *Starting from an initial point $x_0 \in \mathbb{R}^n$, the previous conjugate directions algorithm converges to the single optimal solution $\hat{x}$ of the problem $(PQSC)$ in $n$ iterations, that is to say that we have*

$$x_n = \hat{x} \text{ and } Qx_n = Q\hat{x} = b \tag{2.12}$$

**Remark 2.1** *If we start from the point $x_1$, then the optimal solution is reached at point $x_{n+1}$, that is to say that we will have*

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

**Theorem 2.6** *Let $Q$ be a symmetric and positive definite $(n, n)$ matrix and $b \in \mathbb{R}^n$ and*

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

*Consider the sequence $\{x_k\}_{k=1,2,....}$ in the following way . We start with $x_1 \in \mathbb{R}^n$ At iteration $k$, the successor $x_k + 1$ of $x_k$ is given by the relation next*

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

*where $d_k \in \mathbb{R}^n$ is a search direction and $\alpha_k \in \mathbb{R}^+$ is the search step obtained by an exact linear search, $\alpha_k$ verifies*

$$f(x_k + \alpha_k d_k) = \min_{\alpha>0} f(x_k + \alpha d_k) \tag{2.13}$$

*Let's note*

$$g_{k+1} = \nabla f(x_{k+1}) = Qx_{k+1} - b \tag{2.14}$$

*So*

$$g_{k+1}d_k = g_k + \alpha_k Q d_k \tag{2.15}$$

*and*

$$g_{k+1}^T d_i = 0, k = 0, 1, ..., n-1 \tag{2.16}$$

**Theorem 2.7** *On the conjugate direction method, we have*

$$g_{k+1}^T d_i = 0, k = 0, 1, ..., n-1, i = 0, .....k \tag{2.17}$$

### 2.2.5 Conjugate gradient method. quadratic case

Let $Q$ be a matrix $(n, n)$, symmetric and positive definite. We consider in this paragraph the following problem $(PQSC)$

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

In the conjugate directions method, the directions $d_0, ...., d_{n-1}$ are given to advance.
In the conjugate gradient method, We start from a point $x_0 \in \mathbb{R}^n$,

$$d_0 = -g_0 = \nabla f(x_0) = Qx_0 - b.$$

The directions $d_k$,$k = 1, ...n-1$ are calculated at each iteration.
At iteration $k$

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

$\beta_{k-1}$ is obtained so that $d_k$, is $Q$ conjugated with the other vectors $d_i$,$i = 0, ..., k-1$.
In other words we must have

$$d_k^T Q d_i = 0 \ i = 0, ...k-1. \tag{2.18}$$

In the appellation conjugated gradient, we find the two words : gradient and conjugate.

**a)** The word gradient is used because $d_k$ is calculated from the gradient at the point $x_k$.

**b)** The conjugated word is also justified, because and as will be seen later, the directions $\{d_k\}_{k=0}^{n-1}$ are subjugated.

### 2.2.6 Conjugate gradient algorithm. quadratic case

**Principle of the Algorithm**

We start from any point $x_0 \in \mathbb{R}^n$ .
Calculate $d_0 = -g_0 = b - Qx_0$ , $\alpha_0 = -\dfrac{g_0^T d_0}{d_0^T Q d_0}$
Suppose that at iteration $k$ we have : $x_k$ and $d_k$. This will allow us to calculate

$$g_k = Qx_k - b, \alpha_k = -\frac{g_k^T d_k}{d_K^T Q d_K}, x_{k+1} = x_k + \alpha_k d_k, g_{k+1} = Qx_{k+1} - b, d_{k+1} = -g_{k+1} + \beta_k d_k \tag{2.19}$$

$\beta_k$ is chosen so that

$$d_{k+1}^T . Q d_k = 0 \tag{2.20}$$

Since $d_{k+1} = -g_{k+1} + \beta_k d_k$, then (2.17) gives

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

or again

$$\beta_k d_{k+1}^T . Q d_k = g_{k+1}^T . Q d_k \tag{2.21}$$

and finally

$$\beta_k = \frac{g_{k+1}^T Q d_k}{d_k^T . Q d_k} \tag{2.22}$$

**Algorithm**

**Conjugate gradient algorithm. Quadratic case**

1. Choose $x_0 \in \mathbb{R}^n$.

2. Calculate $g_0 = Qx_0 - b$. If $g_0 = 0$ stop. Otherwise ask $d_0 = -g_0$ Ask $k = 0$

3. Calculate $\alpha_k = -\dfrac{g_k^T Q d_k}{d_k^T Q d_k}$

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

5. Calculate $g_{k+1} = Qx_{k+1} - b$. if $g_{k+1} = 0$ stop.

6. Calculate $\beta_k = \dfrac{g_{k+1}^T Q d_k}{d_k^T Q d_k}$

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

8. Put $k = k + 1$ and go to 3 .

### 2.2.7 Properties of the quadratic conjugate gradient

The fundamental property of the quadratic case conjugate gradient is that the directions $\{d_k\}_{k=0}^{n-1}$ are $Q$ conjugates. These directions verify as we have seen in the algotithm

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

with

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

According to theorem 4.2, the conjugate gradient algorithm, quadratic version converges to the optimal solution in $n$ iterations. Let's summarize these two results in the following two theorems:

**Theorem 2.8** *The directions $\{d_0, d_1, ...., d_{n-1}\}$ generated by the gradient algorithm quadratic conjugate are $Q$ conjugates.*

**Theorem 2.9** *Let $Q$ be a $(n, n)$, symmetric and positive definite and (PQSC) the following quadratic constraint-free minimization problem*

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

*Starting from any point $x_0 \in \mathbb{R}^n$ , consider the sequence generated by the algorithm of the quadratic conjugate gradient defined by*

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

$$\beta_k = \frac{g_{k+1}^TQd_k}{d_k^T.Qd_k},\ k = 0, 1, ... \tag{2.23}$$

$$d_k = \begin{cases} -g_0 & \textit{si } k = 0 \quad (2.24) \\ g_k + \beta_{k-1}d_{k-1} & \textit{si } k \geq 1 \quad (2.25) \end{cases}$$

$$\alpha_k = -\frac{g_k^Td_k}{d_k^TQd_k} \tag{2.26}$$

*and*

$$x_{k+1} = x_k + \alpha_kd_k \tag{2.27}$$

*Then the sequence $\{x_k\}$ converges in $n$ iterations towards the optimal solution $\hat{x}$ of the problem (PQSC), that is to say that $x_n$, verifies $x_n = \hat{x}$ and*

$$Q\hat{x} = Qx_n = b \tag{2.28}$$

## 2.3 Conjugate gradient method.Non-quadratic case

### 2.3.1 Introduction and different forms of the conjugate gradient non-quadratic

Let $f : \mathbb{R}^n \to \mathbb{R}$ be non-quadratic. We seek to solve the non-quadratic problem without constraints $(PNQSC)$ next :

$$\min\{f(x) : x \in \mathbb{R}^n\} \tag{2.29}$$

Among the oldest methods used to solve problems of the type $(PNQSC)$, we can mention the conjugate Gradient method. This method is mainly used for large problems. This method was discovered in 1952 by Hestenes and Steifel ([32]), for the minimization of strictly convex quadratic functions. Several mathematicians have extended this method for the non-quadratic case. This was achieved for the first time, in 1964 by Fletcher and Reeves ([26]) (Fletcher-Reeves method) and then in 1969 by Polak, Ribière ([46]) and Ployak ([42]) (Polak Ribière-Ployak method). Other variants were studied later ([28],[55],[31]) Another variant was studied in 1987 by Fletcher ([30]) (Conjugated descent method). All these methods generate an $\{x_k\}_{k\in\mathbb{N}}$ sequence as follows :

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.30}$$

The age step $\alpha_k \in \mathbb{R}$ is determined by a one-dimensional optimization or search exact or inaccurate linear of the Armijo, Goldstein or Wolfe type.
The directions $d$, are calculated recurrently by the following formulas :

$$d_k = \begin{cases} -g_0 & \text{si } k = 0 \quad (2.31) \\ g_k + \beta_{k-1} d_{k-1} & \text{si } k \geq 1 \quad (2.32) \end{cases}$$

with $g_k = \nabla f(x_k)$ and $\beta_k \in \mathbb{R}$.
The different values assigned to $\beta_k$ define the different shapes of the conjugate gradient
If we note

$$y_{k-1} = g_k - g_{k-1},\ s_k = x_{k+1} - x_k \tag{2.33}$$

the following variants are obtained :
1952 ([32])- Conjugate gradient Hestenes - Stiefel variant(HS)

$$\beta_K^{HS} = \frac{g_{K+1}^T y_k}{d_k^T y_k} \tag{2.34}$$

1964 ([26])-Conjugate gradient variant Fletcher Reeves(FR)

$$\beta_K^{FR} = \frac{\|g_{K+1}\|^2}{\|g_k\|^2} \tag{2.35}$$

1969 ([42].[46])- Conjugate gradient Polak-Ribière-Polyak variant(PRP)

$$\beta_K^{PRP} = \frac{g_{K+1}^T y_k}{\|g_k\|^2} \tag{2.36}$$

1987 ([30])- Conjugate gradient conjugate descent variant - Fletcher (CD)

$$\beta_K^{CD} = -\frac{\|g_K\|^2}{d_{k-1}^T g_{k-1}} \tag{2.37}$$

1991 ([36])- Conjugate gradient Liu - Storey variant(LS)

$$\beta_K^{LS} = -\frac{g_{K+1}^T y_k}{d_k^T g_k} \tag{2.38}$$

1999 ([11])- Conjugate gradient variant of Dai-Yuan(DY)

$$\beta_K^{DY} = \frac{\|g_{K+1}\|^2}{d_k^T y_k} \tag{2.39}$$

2005([31])- Conjugate gradient Hager-Zhang variant(HZ) -(-24)

$$\beta_K^{HZ} = (y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k})^T \frac{g_{K+1}}{d_k^T y_k} \tag{2.40}$$

2012([48])- Conjugate gradient variant Rivaie-Mustafa-Ismail-Leong(RMIL)[60]

$$\beta_{K-1}^{RMIL} = \frac{g_K^T (g_k - g_{k-1})}{\|d_{k-1}\|^2} \tag{2.41}$$

**Remark 2.2** *In the case where $f$ is not quadratic we have*

$$\beta_k^{HS} \neq \beta_k^{FR} \neq \beta_k^{PRP} \neq \beta_k^{CD} \neq \beta_k^{LS} \neq \beta_k^{DY} \neq \beta_k^{HZ} \neq \beta_k^{RMIL} \tag{2.42}$$

*Therefore, by applying the non-quadratic conjugate gradient algorithm, using the coefficients $\beta_k$ appearing in (2.39), we obtain sequences $\{x_k\}_{k\in\mathbb{N}}$ different.*
*What happens if $f$ is strictly convex quadratic and if $\alpha_k$ is obtained by an exact linear search. The answer to this question can be found in the following theorem.*

**Theorem 2.10** *If $f(x) = \frac{1}{2}x^TQx - b^Tx$, with a positive definite symmetric $Q$, $x \in \mathbb{R}^n$; $b \in \mathbb{R}^n$ and if $\alpha_k$ is obtained by an exact linear search. Let's note*

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

*so we have*

$$\beta_k^{HS} = \beta_k^{FR} = \beta_k^{PRP} = \beta_k^{CD} = \beta_k^{LS} = \beta_k^{DY} = \beta_k^{HZ} = \beta_k^{RMIL} \tag{2.43}$$

*and the quadratic conjugate gradient algorithm generates the same sequence $\{x_k\}_{k\in\mathbb{N}}$.*

## Non-quadratic conjugate gradient algorithm

### Introduction

Let $f : \mathbb{R}^n \to \mathbb{R}$ be non-quadratic and $(PNQSC)$ the minimization problem not quadratic without constraints following :

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

To construct the non-quadratic conjugate case gradient algorithm, we can draw inspiration from the quadratic conjugate gradient algorithm established in the previous chapter. Unlike the quadratic case, we do not have a matrix $Q$. Therefore we do not have conjugate $Q$ directions. As in the quadratic case, the algorithm of the conjugate gradient non-quadratic case generates a sequence $\{x_k\}_{k\in\mathbb{N}}$ in the following way :

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

The algorithm starts from any point $x_0 \in \mathbb{R}^n$.

**At iteration $k$**

Suppose that we have the vector $x_k \in \mathbb{R}^n$ and the direction $d_{k-1}$ This allows us to calculate $\nabla f(x_k)$ instead of $g_k = Qx_k - b$ in the quadratic case. To have $x_{k+1}$, we need to calculate $\alpha_k$ and $d_k$.

**Calculation of $d_k$:**

$$d_k = -\nabla f(x_k) + \beta_{k-1} d_{k-1} \tag{2.44}$$

We have eight ways to calculate $\beta_{k-1}$

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

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

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

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

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

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

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

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

**Calculation of $\alpha_k$**

Having obtained $d_k$ , recall that $\alpha_k$ checks

$$f(x_k + \alpha_k d_k) = min\{f(x_k + \alpha d_k) : \alpha \in ]0, +\infty[\} \tag{2.45}$$

In the case where $f(x) = \frac{1}{2}x^T Qx - b$, positive definite symmetric $Q$, a, solution of (2.42), is given by the following relation

$$\alpha_k = -\frac{g_k^T}{d_k^T Q d_K} \tag{2.46}$$

In the case where $f$ is not quadratic, $\alpha_k$ cannot be calculated by the formula (2.43). In this case, $\alpha_k$ is calculated by other methods. For example, the golden ratio number method or the dichotomy method is used. As will be seen later,$\alpha_k$ can be calculated by an inaccurate linear search of Armijo or Goldstein or Wolfe

**Non-quadratic conjugate Gradient algorithm**

**▶ step 1**

$Chooseany$x$_0 \in \mathbb{R}^n$ and $\epsilon > 0$

**▶ step 2**

$Ask$k = 0

$Call$g$_k = \nabla f(x_0)$. Ask $d_0 = -g_0$

**▶ step 3**

$Calculate\alpha_k$ using an exact or inaccurate linear search of Armijo or Goldstein or Wolfe or Strong Wolfe

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

**▶ step 4**

$If\|\nabla f(x_{k+1})\| < \epsilon$, Stop, $x^* = x_{k+1}$ .Otherwise go to Step 5

**▶ step 5**

$Calculate$g$_{k+1} = \nabla f(x_{k+1})$

Calculate $\beta_k$ by one of the following ways

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

Calculate

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

Put $k = k + 1$ and go to Step 3.

# Chapter 3

# Acceleration of the convergence of the gradient method by using the conjugate gradient

Consider the unconstrained optimization problem

$$(P) \quad \min\{f(x) : x \in R^n\} \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. The line search method usually takes the following iterative formula

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

for (3.1), where $x_k$ is the current iterate point, $\alpha_k > 0$ is a steplength and $d_k$ is a search direction. Different choices of $d_k$ and $\alpha_k$ will determine different line search methods [22, 35, 17]. We denote $f(x_k)$ by $f_k$ , $\nabla f(x_k)$ by $g_k$ , and $\nabla f(x_{k+1})$ by $g_{k+1}$ , respectively. $\|.\|$ denotes the Euclidian norm of vectors and define

$$y_k = g_{k+1} - g_k \; .$$

We all know that a method is called steepest descent method if we take

$$d_k = -g_k$$

as a search direction at every iteration, which has wide applications in solving large-scale minimization problems [47, 49, 25]. One drawback of the method is often yielding zigzag phenomena in solving practical problems, which makes the algorithm converge to an optimal solution very slowly, or even fail to converge [40, 41].

If we take

$$d_k = -H_k g_k$$

as a search direction at each iteration in the algorithm, where $H_k$ is an $n \times n$ matrix approximating $[\nabla^2 f(x_k)]^{-1}$, then the corresponding method is called the Newton-like method [40, 41, 50] such as the Newton method, the quasi-Newton method, variable metric method, etc. Many papers have proposed this method for optimization problems [ 17, 9].

However, the Newton-like method needs to store and compute matrix $H_k$ at each iteration and thus adds to the cost of storage and computation. Accordingly, this method is not suitable to solve large-scale optimization problems in many cases.

The steepest descent method is one of the simplest and the most fundamental minimization methods for unconstrained optimization. Since it uses the negative gradient as its descent direction, it is also called the gradient method.

For many problems, the steepest descent method is very slow. Although the method usually works well in the early steps, as a stationnary point is approached, it descends very slowly with zigzaguing phenomena. There are some ways to overcome these difficulties of zigzagging by defleting the gradient. Rather then moving along

$$d_k = -\nabla f(x_k) = -g_k,$$

we can move along

$$d_k = -D_k \nabla f(x_k),$$

or along

$$d_k = -g_k + h_k, \tag{3.3}$$

where $D_k$ is an appropriate matrix and $h_k$ is an appropriate vector. Due to its simplicity and its very low memory requirement, the conjugate gradient method is a powerful line search method for solving the large-scale optimization problems. In fact, the CG method is not among the fastest or most robust optimization algorithms for nonlinear problems available today, but it remains very popular for engineers and mathematicians who are interested in solving large problems [12, 7, 15, 6, 28, 58]. The conjugate gradient method is designed to solve unconstrained optimization problem (3.1). More explicitly, the conjugate gradient method is an algorithm for finding the nearest local minimum of a function of variables which presupposes that the gradient of the function can be computed.We consider only the case where the method is implemented without regular restarts. The iterative formula of the conjugate gradient method is given by (3.2), where $\alpha_k$ is a steplength which is computed by carrying out a line search, and $d_k$ is the search direction defined by

$$d_{k+1} = \begin{cases} -g_k & \text{si } k = 1 \qquad (3.4) \\ g_{k+1} + \beta_k d_k & \text{si } k \geq 2 \qquad (3.5) \end{cases}$$

where $\beta_k$ is a scalar, and $g_k$ denotes $g(x_k)$. Some well known formulas for $\beta_k$ are given as follows:

$$\beta_K^{HS} = \frac{g_{K+1}^T y_k}{d_k^T y_k}\,,\ \beta_K^{FR} = \frac{\|g_{K+1}\|^2}{\|g_k\|^2}\,,\ \beta_K^{PRP} = \frac{g_{K+1}^T y_k}{\|g_k\|^2}\,,\ \beta_K^{CD} = -\frac{\|g_{K+1}\|^2}{d_k^T g_k}$$

$$\beta_K^{LS} = -\frac{g_{K+1}^T y_k}{d_k^T g_k}\,,\ \beta_K^{DY} = \frac{\|g_{K+1}\|^2}{d_k^T y_k}\,,\ \beta_K^{HZ} = (y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k})^T \frac{g_{K+1}}{d_k^T y_k}$$

The above corresponding methods are known as Hestenes-Stiefel (HS) method [33], the Fletcher-Reeves (FR) method [29], the Polak-Ribiere-Polyak (PR) method (see [43, 8]), the Conjugate Descent method(CD) [29], the Liu-Storey (LS) method [37], the Dai-Yuan (DY) method [13], and Hager and Zhang (HZ) method [32], respectively.

In the convergence analysis and implementation of conjugate gradient methods, one often requires the inexact line search such as the Wolfe conditions or the strong Wolfe conditions. The Wolfe line search is to find $\alpha_k$ such that

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \tag{3.6}$$

$$d_k^T g(x_k + \alpha_k d_k) \geq \sigma d_k^T g_k \tag{3.7}$$

with $\delta < \sigma < 1$. The strong Wolfe line search is to find $\alpha_k$ such that

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \tag{3.8}$$

$$|d_k^T g(x_k + \alpha_k d_k)| \leq -\sigma d_k^T g_k \tag{3.9}$$

where $\delta < \sigma < 1$ are constants.

The convergence behavior of the above formulas with some line search conditions has been studied by many authors for many years.

Al-Baali [1] has proved the global convergence of the FR method for nonconvex functions with the strong Wolfe line search if the parameter $\sigma < \frac{1}{2}$. The PRP method with exact line search may cycle without approaching any stationary point, see Powellâs counter-example [44]. Although one would be satisfied with its global convergence properties, the FR method sometimes performs much worse than the PRP method in real computations. A similar case happen to the DY method and the HS method.

In next section, we will state the idea of the new method, then a new algorithm will be developed. Descent property and the global convergence will be established in Section 2. Section 3 is devoted to numerical experiments by implementing the algorithm to solve many large-scale benchmark test problems. The conclusions are presented in Section 4.

## 3.1 The new formula and the corresponding algorithm

In this section, we shall state the idea to propose a new conjugate gradient method and develop a new algorithm.

In this paper, based the modified strong Wolfe type line search, under some mild conditions, we give the Descent property and global convergence of the new $\beta_k$ which is known as $\beta_k^{BRB}$, where $BRB$ denotes **Belloufi**, **Rahali** and **Benzine**. Then we can define the following formulas $\beta_k$ to compute the search directions in (3.4) and (3.5).

$$\beta_k^{BRB} = \frac{\|g_{k+1}\|^2}{\|d_k\|^2} \tag{3.10}$$

With the constructed search direction, we find a stepsize by the modified strong Wolfe line search strategy:

**Modification of the strong Wolfe line search**

The step length is computed by performing a line search along $d_k$. In practice, a relevant choice is to compute $\alpha_k$ according to the realization of the modified strong Wolfe conditions, namely

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k \tag{3.11}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k \frac{\|d_k\|^2}{\|g_k\|^2} \tag{3.12}$$

The algorithm is given as follows:

**algorithm** Step 0: Given $x_1 \in \mathbb{R}^n$ , set $d_1 = -g_1, k := 1$ .

Step 1: If $\|g_k\| = 0$ then stop else go to Step 2.

Step 2: Set $x_{k+1} = x_k + \alpha_k d_k$ where $d_k$ is defined by (3.4)and (3.5),(3.10) and $\alpha_k$ is defined by (3.11),(3.12).

Step 3: Set $k := k + 1$ and go to Step 1.

## 3.2 Descent property and global convergence

The following theorem indicates that, in the inexact case, the search direction $d_k$ satisfies descent property.

**Theorem 3.1** *If an $\alpha_k$ is calculated wich satisfies modified strong Wolfe line search (3.11) and (3.12) with $\sigma \in ]0, \frac{1}{2}]$, $\forall k$ then for the new conjugate gradient method, the inequality*

$$-\sum_{j=0}^{k-1} \sigma^j \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -2 + \sum_{j=0}^{k-1} \sigma^j \tag{3.13}$$

*holds for all $k$, and hence the descent property*

$$g_k^T d_k < 0, \forall k \tag{3.14}$$

*holds, as long as $g_k \neq 0$.*

**Proof.** The proof is by induction.

when we take

$$\sigma g_k^T d_k \frac{\|d_k\|^2}{\|g_k\|^2} \leq g(x_k + \alpha_k d_k)^T d_k \leq -\sigma g_k^T d_k \frac{\|d_k\|^2}{\|g_k\|^2},$$

$$\sigma g_k^T d_k \frac{\|d_k\|^2}{\|g_k\|^2} \times \frac{1}{\|d_k\|^2} \leq g(x_k + \alpha_k d_k)^T d_k \times \frac{1}{\|d_k\|^2} \leq -\sigma g_k^T d_k \frac{\|d_k\|^2}{\|g_k\|^2} \times \frac{1}{\|d_k\|^2}$$

$$\sigma g_k^T d_k \frac{1}{\|g_k\|^2} \leq \frac{g(x_k + \alpha_k d_k)^t d_k}{\|d_k\|^2} \leq -\sigma g_k^T d_k \frac{1}{\|g_k\|^2}$$

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

For $k = 1$ Equations (3.13) and (3.14) is clearly satisfied. Now we suppose that (3.13) and (3.14) hold for any $k \geq 1$.

It follows from the definition (3.4),(3.5) and (3.10) of $d_{k+1}$ that

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

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

and hence from (3.12) and (3.14) that

$$-1+\sigma\frac{g_k^T d_k}{\|g_k\|^2} \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq 1-\sigma\frac{g_k^T d_k}{\|g_k\|^2} \tag{3.16}$$

Also, by induction assumption (3.13), we have

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

when We take the first part of the retracement (3.13)

$$-\sum_{j=0}^{k-1}\sigma^j \leq \frac{g_k^T d_k}{\|g_k\|^2}$$
$$-\sigma\sum_{j=0}^{k-1}\sigma^j \leq \sigma\frac{g_k^T d_k}{\|g_k\|^2}$$
$$-1-\sigma\sum_{j=0}^{k-1}\sigma^j \leq -1+\sigma\frac{g_k^T d_k}{\|g_k\|^2}$$
$$-1-\sigma\sum_{j=0}^{k-1}\sigma^j \leq -1+\sigma\frac{g_k^T d_k}{\|g_k\|^2} \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \text{............(1)}$$

and when We take again the first part of the retracement (3.13)

$$-\sum_{j=0}^{k-1}\sigma^j \leq \frac{g_k^T d_k}{\|g_k\|^2}$$
$$(-\sigma)\times\left(-\sum_{j=0}^{k-1}\sigma^j\right) \geq -\sigma\times\frac{g_k^T d_k}{\|g_k\|^2}$$
$$-\sigma\frac{g_k^T d_k}{\|g_k\|^2} \leq \sigma\sum_{j=0}^{k-1}\sigma^j$$

$$-1-\sigma\frac{g_k^T d_k}{\|g_k\|^2}\leq -1+\sigma\sum_{j=0}^{k-1}\sigma^j$$
$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2}\leq -1-\sigma\frac{g_k^T d_k}{\|g_k\|^2}\leq -1+\sigma\sum_{j=0}^{k-1}\sigma^j$$

we get the

$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2}\leq -1-\sigma\frac{g_k^T d_k}{\|g_k\|^2}\leq -2+\sum_{j=0}^{k-1}\sigma^j \text{ ............(2)}$$

from (1)and (2) we find :

$$-1+\sigma\frac{g_k^T d_k}{\|g_k\|^2}\leq -1-\sigma\sum_{j=0}^{k-1}\sigma^j\leq\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2}\leq -1-\sigma\frac{g_k^T d_k}{\|g_k\|^2}\leq -1+\sigma\sum_{j=0}^{k-1}\sigma^j$$

$$-\sum_{j=0}^{k}\sigma^j=-1-\sigma\sum_{j=0}^{k-1}\sigma^j\leq\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2}\leq -1+\sigma\sum_{j=0}^{k-1}\sigma^j=-2+\sum_{j=0}^{k}\sigma^j \tag{3.17}$$

so

$$-\sum_{j=0}^{k}\sigma^j\leq\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2}\leq -2+\sum_{j=0}^{k}\sigma^j\text{............(3)}$$

Then, (3.13) holds for $k+1$.

we multiply the second part of (3) in the $\|g_{k+1}\|^2$

$$\|g_{k+1}\|^2(-\sum_{j=0}^{k}\sigma^j)\leq\|g_{k+1}\|^2(\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2})\leq(-2+\sum_{j=0}^{k}\sigma^j)\|g_{k+1}\|^2$$

Since

$$g_{k+1}^T d_{k+1}\leq\|g_{k+1}\|^2\left(-2+\sum_{j=0}^{k}\sigma^j\right) \tag{3.18}$$

and

$$\sum_{j=0}^{k}\sigma^j<\sum_{j=0}^{\infty}\sigma^j=\frac{1}{1-\sigma} \tag{3.19}$$

where $\sigma\in]0,\frac{1}{2}]$ ,

$$0 < \sigma < \frac{1}{2}$$
$$0 > -\sigma > -\frac{1}{2}$$
$$1 > 1 - \sigma > 1 - \frac{1}{2}$$

it follows from $1 - \sigma > \frac{1}{2}$

$$\frac{1}{1-\sigma} < \frac{1}{\frac{1}{2}}$$
$$\frac{1}{1-\sigma} < 2$$
$$-2 + \frac{1}{1-\sigma} < 0$$

$-2 + \sum_{j=0}^{k} \sigma^j < -2 + \sum_{j=0}^{\infty} \sigma^j < 0$ that $-2 + \sum_{j=0}^{k} \sigma^j < 0$. Hence, from (3.18),

$$g_{k+1}^T d_{k+1} \leq (-2 + \sum_{j=0}^{k} \sigma^j) \|g_{k+1}\|^2 < 0$$
$$g_{k+1}^T d_{k+1} < 0$$

we obtain

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

We complete the proof by induction.

In order to establish the global convergence of the proposed method, we assume that the following assumption always holds, i.e. Assumption 1.1 : ■

**Assumption 1.1**

Let $f$ be twice continuously differentiable, and the level set $L = \{x \in \mathbb{R}^n | f(x) \leq f(x_1)\}$ be bounded

**Theorem 3.2** *Suppose that $x_1$ is a starting point for which Assumption 3.1 holds. Consider the New method (3.4),(3.5) and (3.10). If the steplength $\alpha_k$ is computed by the modified strong Wolfe line search (3.11) and (3.12) with $\delta < \sigma < \frac{1}{2}$ and if*

$$\frac{1}{\|d_{k-1}\|^4} \leq \frac{1}{\|g_{k-1}\|^4} \tag{3.20}$$

*then the method is globally convergent, i.e.,*

$$\liminf_{k\longrightarrow\infty}\|g_k\|=0 \tag{3.21}$$

**Proof.** It is shown in theorem 1 that the descent property (3.14) holds for $\sigma\in]0,\frac{1}{2}]$ ,
We take from (3.13) and multiply all its sides by $(-\sigma\|d_k\|^2)$
we find that :

$$\sigma\|d_k\|^2\sum_{j=0}^{k-1}\sigma^j\geq-\sigma\|d_k\|^2\frac{g_k^Td_k}{\|g_k\|^2}\geq\left(-2+\sum_{j=0}^{k-1}\sigma^j\right)(-\sigma\|d_k\|^2)$$

We get on with the (3.12):

$$|g_{k+1}^Td_k|\leq-\sigma g_k^Td_k\frac{\|d_k\|^2}{\|g_k\|^2}\leq\sigma\|d_k\|^2\sum_{j=0}^{k-1}\sigma^j$$

We take from (3.19) and multiply all its sides by $(\sigma\|d_k\|^2)$
we find that :

$$\sigma\|d_k\|^2\left(\sum_{j=0}^{k-1}\sigma^j\right)<\sigma\|d_k\|^2\left(\sum_{j=0}^{k}\sigma^j\right)<\sigma\|d_k\|^2\left(\sum_{j=0}^{\infty}\sigma^j\right)=\sigma\|d_k\|^2\left(\frac{1}{1-\sigma}\right)$$

so

$$|g_{k+1}^Td_k|\leq-\sigma g_k^Td_k\frac{\|d_k\|^2}{\|g_k\|^2}\leq\sigma\|d_k\|^2\sum_{j=0}^{k-1}\sigma^j\leq\frac{\sigma}{1-\sigma}\|d_k\|^2$$
$$|g_{k+1}^Td_k|\leq-\sigma g_k^Td_k\frac{\|d_k\|^2}{\|g_k\|^2}\leq\|d_k\|^2\sum_{j=0}^{k}\sigma^j\leq\frac{\sigma}{1-\sigma}\|d_k\|^2$$

for $k=k-1$ :

$$|g_k^Td_{k-1}|\leq-\sigma g_{k-1}^Td_{k-1}\frac{\|d_{k-1}\|^2}{\|g_{k-1}\|^2}\leq\|d_{k-1}\|^2\sum_{j=0}^{k-1}\sigma^j\leq\frac{\sigma}{1-\sigma}\|d_{k-1}\|^2$$

so from (3.12), (3.13), and (3.19) it follows that

$$|g_k^Td_{k-1}|\leq\sigma g_{k-1}^Td_{k-1}\frac{\|d_{k-1}\|^2}{\|g_{k-1}\|^2}\leq\|d_{k-1}\|^2\sigma\sum_{j=0}^{k-2}\sigma^j=\|d_{k-1}\|^2\sum_{j=0}^{k-1}\sigma^j\leq\frac{\sigma}{1-\sigma}\|d_{k-1}\|^2 \tag{3.22}$$

Thus from the definition of $d_k$ and using (3.10) and (3.22) we deduce that

$$\|d_k\|^2=\|g_k\|^2-2\beta_{k-1}g_k^Td_{k-1}+\beta_{k-1}^2\|d_{k-1}\|^2 \tag{3.23}$$

from (3.22) we have :

$$\begin{aligned}
|g_k^T d_{k-1}| &\le \frac{\sigma}{1-\sigma}\|d_{k-1}\|^2\\
-\frac{\sigma}{1-\sigma}\|d_{k-1}\|^2 &\le g_k^T d_{k-1} \le \frac{\sigma}{1-\sigma}\|d_{k-1}\|^2\\
\|g_k\|^2 - 2\beta_{k-1}g_k^T d_{k-1} + \beta_{k-1}^2\|d_{k-1}\|^2 &\le \|g_k\|^2 + 2\beta_{k-1}g_k^T d_{k-1} + \beta_{k-1}^2\|d_{k-1}\|^2\\
\|g_k\|^2 + \frac{2\sigma}{1-\sigma}\beta_k\|d_{k-1}\|^2 + \beta_{k-1}^2\|d_{k-1}\|^2 &= \left(\frac{1+\sigma}{1-\sigma}\right)\|g_k\|^2 + \frac{\|g_k\|^4}{\|d_{k-1}\|^4}\|d_{k-1}\|^2\\
\|g_k\|^2 + \left(\frac{2\sigma}{1-\sigma}\beta_k + \beta_{k-1}^2\right)\|d_{k-1}\|^2 &= \|g_k\|^2 + \left(\frac{2\sigma}{1-\sigma}\frac{\|g_{k+1}\|^2}{\|d_k\|^2} + \frac{\|g_k\|^4}{\|d_{k-1}\|^4}\right)\|d_{k-1}\|^2\\
&= \|g_k\|^2 + \frac{2\sigma}{1-\sigma}\frac{\|g_{k+1}\|^2}{\|d_k\|^2}\|d_{k-1}\|^2 + \frac{\|g_k\|^4}{\|d_{k-1}\|^4}\|d_{k-1}\|^2
\end{aligned}$$

for $k = k+1$ and $\beta_k = \beta_{k-1}$

$$\begin{aligned}
&= \frac{1-\sigma}{1-\sigma}\|g_k\|^2 + \frac{2\sigma}{1-\sigma}\|g_k\|^2 + \frac{\|g_k\|^4}{\|d_{k-1}\|^4}\|d_{k-1}\|^2\\
&= \frac{1-\sigma}{1-\sigma}\|g_k\|^2 + \frac{2\sigma}{1-\sigma}\|g_k\|^2 + \frac{\|g_k\|^4}{\|d_{k-1}\|^4}\|d_{k-1}\|^2\\
&= \frac{1-\sigma+2\sigma}{1-\sigma}\|g_k\|^2 + \frac{\|g_k\|^4}{\|d_{k-1}\|^4}\|d_{k-1}\|^2\\
&= \left(\frac{1+\sigma}{1-\sigma}\right)\|g_k\|^2 + \frac{\|g_k\|^4}{\|d_{k-1}\|^4}\|d_{k-1}\|^2
\end{aligned}$$

In (3.20) we have

$$\begin{aligned}
\frac{1}{\|d_{k-1}\|^4} &\le \frac{1}{\|g_{k-1}\|^4}\\
\frac{\|g_k\|^4}{\|d_{k-1}\|^4}\|d_{k-1}\|^2 &\le \frac{\|g_k\|^4}{\|g_{k-1}\|^4}\|d_{k-1}\|^2
\end{aligned}$$

$\sigma \in ]0, \frac{1}{2}[$

$$\begin{aligned}
\left(\frac{1+\sigma}{1-\sigma}\right)\|g_k\|^2 + \frac{\|g_k\|^4}{\|d_{k-1}\|^4}\|d_{k-1}\|^2 &\le \left(\frac{1+\sigma}{1-\sigma}\right)\|g_k\|^2 + \frac{\|g_k\|^4}{\|g_{k-1}\|^4}\|d_{k-1}\|^2\\
\left(\frac{1+\sigma}{1-\sigma}\right)\|g_k\|^2 + \frac{\|g_k\|^4}{\|d_{k-1}\|^2} &\le \left(\frac{1+\sigma}{1-\sigma}\right)\|g_k\|^2 + \frac{\|g_k\|^4}{\|g_{k-1}\|^4}\|d_{k-1}\|^2
\end{aligned}$$

So:

$$\begin{aligned}
&\le \|g_k\|^2 + \frac{2\sigma}{1-\sigma}\beta_k\|d_{k-1}\|^2 + \beta_{k-1}^2\|d_{k-1}\|^2\\
&= \frac{1+\sigma}{1-\sigma}\|g_k\|^2 + \frac{\|g_k\|^4}{\|d_{k-1}\|^4}\|d_{k-1}\|^2
\end{aligned}$$

$$\le \frac{1+\sigma}{1-\sigma}\|g_k\|^2 + \frac{\|g_k\|^4}{\|g_{k-1}\|^4}\|d_{k-1}\|^2 \tag{3.24}$$

where we used the facts that

$$\frac{1}{\|d_{k-1}\|^4} \leq \frac{1}{\|g_{k-1}\|^4} \tag{3.25}$$

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

So

$$\|d_k\|^2 \leq \left(\frac{1+\sigma}{1-\sigma}\right)\frac{\|g_k\|^4}{\|g_k\|^2} + \frac{\|g_k\|^4}{\|d_{k-1}\|^2} \leq \left(\frac{1+\sigma}{1-\sigma}\right)\frac{\|g_k\|^4}{\|g_k\|^2} + \frac{\|g_k\|^4}{\|g_{k-1}\|^4}\|d_{k-1}\|^2$$

By applying this relation repeatedly,it follows that

$$\|d_k\|^2 \leq \frac{1+\sigma}{1-\sigma}\|g_k\|^4 \sum_{j=1}^{k} \frac{1}{\|g_j\|^2} \tag{3.26}$$

Now we prove (3.21) by contradiction. It assumes that (3.21) does not hold, then there exists a constant $\epsilon > 0$ such that

$$\|g_k\| \geq \epsilon > 0 \tag{3.27}$$

holds for all $k$ sufficiently large. Since $g_k$ is bounded above on the level set $L$, it follows from (3.26) that

$$\|d_k\|^2 \leq c_1 k \tag{3.28}$$

where $c_1$ is a positive constant.From (3.13) and (3.19), we have

$$\sum_{j=0}^{k} \sigma^j < \sum_{j=0}^{\infty} \sigma^j = \frac{1}{1-\sigma}$$

$$1 + \sigma \sum_{j=0}^{k-1} \sigma^j = \sum_{j=0}^{k} \sigma^j < \sum_{j=0}^{\infty} \sigma^j = \frac{1}{1-\sigma}$$

$$-1 - \sigma \sum_{j=0}^{k-1} \sigma^j = -\sum_{j=0}^{k} \sigma^j > \frac{-1}{1-\sigma}$$

We take the first part

$$
\begin{aligned}
\sigma \sum_{j=0}^{k-1} \sigma^j &< \frac{1}{1-\sigma} \\
\sigma \sum_{j=0}^{k-1} \sigma^j &< \frac{1-1+\sigma}{1-\sigma} \\
\sum_{j=0}^{k-1} \sigma^j &< \frac{\sigma}{1-\sigma} \times \frac{1}{\sigma} \\
\sum_{j=0}^{k-1} \sigma^j < \frac{1}{1-\sigma} &\Rightarrow -\sum_{j=0}^{k-1} \sigma^j > \frac{-1}{1-\sigma}
\end{aligned}
$$

we take the second part of (3.19)

$$
\begin{aligned}
\frac{g_k^T d_k}{\|g_k\|^2} &\leq -2 + \sum_{j=0}^{k-1} \sigma^j \leq -2 + \frac{1}{1-\sigma} \\
&\leq \frac{-2+2\sigma+1}{1-\sigma}
\end{aligned}
$$

So

$$
\begin{aligned}
\frac{g_k^T d_k}{\|g_k\|^2} &\leq \frac{-1+2\sigma}{1-\sigma} \\
-\frac{g_k^T d_k}{\|g_k\|^2} &\geq \left(\frac{1-2\sigma}{1-\sigma}\right) \\
-\frac{g_k^T d_k}{\|g_k\|^2} \times \frac{\|g_k\|^2}{\|d_k\|^2} &\geq \left(\frac{1-2\sigma}{1-\sigma}\right) \frac{\|g_k\|^2}{\|d_k\|^2} \\
-\frac{g_k^T d_k}{\|d_k\|^2} &\geq \frac{(1-2\sigma)}{(1-\sigma)} \frac{\|g_k\|^2}{\|d_k\|^2}
\end{aligned}
$$

and by the condition from Zoutendijk.

$$
\cos\,\theta_k = -\frac{g_k^T d_k}{\|d_k\|^2}
$$

So

$$
\cos\,\theta_k \geq \left(\frac{1-2\sigma}{1-\sigma}\right) \frac{\|g_k\|}{\|d_k\|} \tag{3.29}
$$

When

$$
\begin{aligned}
-\sigma &> \frac{-1}{2} \\
1-\sigma &> 1 + \frac{-1}{2} \\
1-\sigma &> \frac{1}{2} \\
\frac{1}{1-\sigma} &> 2
\end{aligned}
$$

another side

$$-2\sigma > -1$$
$$1-2\sigma > 0$$
$$(1-2\sigma)\times\left(\frac{1}{1-\sigma}\right) > 2\times 0$$
$$\frac{1-2\sigma}{1-\sigma} > 0$$

and we have

$$\|d_k\|^2 \leq c_1 k$$
$$\frac{1}{\|d_k\|^2} \geq \frac{1}{c_1 k}$$
$$\|g_k\| \geq \epsilon > 0$$
$$\|g_k\|^2 \geq \epsilon^2$$
$$\frac{\|g_k\|^2}{\|d_k\|^2} \geq \frac{\epsilon^2}{c_1 k}$$

Since $\sigma < \frac{1}{2}$, substituting (3.28) and (3.27) into (3.29) gives

$$\sum_k \cos^2\theta_k \geq \left(\frac{1-2\sigma}{1-\sigma}\right)^2 \sum_k \frac{\|g_k\|^2}{\|d_k\|^2} \geq \frac{\epsilon^2}{c_1}\sum_k \frac{1}{k}$$

$$\sum_k \cos^2\theta_k \geq \left(\frac{1-2\sigma}{1-\sigma}\right)^2 \sum_k \frac{\|g_k\|^2}{\|d_k\|^2} \geq c_2\sum_k \frac{1}{k} \tag{3.30}$$

where $c_2$ is a positive constant. Therefore, the series $\sum_k \cos\,^2\theta_k$ is divergent. Let $M$ be an upper bound of $\|\nabla^2 f(x)\|$ on the level set $L$, then

$$g_{k+1}^T d_k = (g_k + a_k\nabla^2 f(x))^T d_k \leq g_k^T d_k + M a_k\|d_k\|^2$$

$$g_{k+1}^T d_k - g_k^T d_k \leq M a_k \|d_k\|^2$$
$$(g_{k+1}^T - g_k^T) d_k \leq M a_k \|d_k\|^2$$
$$\frac{1}{(g_{k+1}^T - g_k^T) d_k} \geq \frac{1}{M a_k \|d_k\|^2}$$

and

$$a_k \geq \frac{(g_{k+1}^T - g_k^T) d_k}{M\|d_k\|^2}$$
$$a_k \geq -\frac{(g_k^T - g_{k+1}^T) d_k}{M\|d_k\|^2}$$
$$a_k \geq -\frac{\left(1-\frac{g_{k+1}^T}{g_k^T}\right) g_k^T}{M\|d_k\|^2} d_k \geq -\frac{(1-\sigma)}{M\|d_k\|^2} g_k^T d_k$$

Thus by using (3.12) and (3.20) we obtain

$$a_k \geq \frac{(1-\sigma)}{M\|d_k\|^2} g_k^T d_k \tag{3.31}$$

from (3.11)

$$\begin{gathered}
f_{k+1} \leq f_k + \delta \alpha_k g_k^T d_k \\
f_{k+1} - f_k \leq \delta \alpha_k g_k^T d_k \\
-\frac{g_k^T d_k}{\|d_k\|^2} = \cos\theta_k \Rightarrow g_k^T d_k = -\|d_k\|^2 \cos\theta_k \\
f_{k+1} - f_k \leq -\frac{g_k^T d_k}{\|d_k\|^2 M} \delta(1-\sigma).g_k^T d_k \\
f_{k+1} - f_k \leq \cos\theta_k \frac{\delta(1-\sigma)}{M}.(-\|d_k\|^2 \cos\theta_k) \\
f_{k+1} - f_k \leq \cos\theta_k c_3(-\|d_k\|^2 \cos\theta_k) \\
f_{k+1} - f_k \leq -c_3\|d_k\|^2 \cos^2\theta_k
\end{gathered}$$

and

$$\begin{gathered}
\|d_k\|^2 \leq \|g_k\|^2 \\
f_{k+1} - f_k \leq -c_3\|d_k\|^2 \cos^2\theta_k \leq -c_3\|g_k\|^2 \cos^2\theta_k
\end{gathered}$$

So

$$f_{k+1} - f_k \leq -c_3\|g_k\|^2 \cos^2\theta_k$$

Substituting $a_k$ of (3.31) into (3.11) gives

$$f_{k+1} \leq f_k - c_3\|g_k\|^2 \cos^2\ \theta_k, \tag{3.32}$$

where $c_3 = \dfrac{(1-\sigma)\delta}{M} > 0$ . Since $f(x)$ is bounded below, $\sum\limits_k \|g_k\|^2 \cos^2\ \theta_k$ converges, which indicates that $\sum\limits_k \cos^2\ \theta_k$ converges by use of (3.27). This fact contradicts (3.30). We complete the proof. ■

## 3.3 Numerical results and discussions

In this section we report some numerical results obtained with a Fortran implementation of gradient algorithms and their accelerated variants. All codes are written in Fortran and compiled with f77 (default compiler settings) on a Workstation Intel(R) core(TM), i3@ 2.20GHz. We selected a number of 75 large-scale unconstrained optimization test functions in generalized or extended form [2]. For each test function we have considered ten numerical experiments with the number of variables $n = 1000, 2000, ..., 10000$. In the following we present the numerical performance of CG and $ACG$ codes corresponding to different formula for $\beta_k$ computation. All algorithms implement the Wolfe line search conditions with $\rho = 0.0001$ andr $\rho = 0.9$ , and the same stopping criterion $\|g_k\|_\infty \leq 10^{-10}$, where $\|.\|_\infty$ is the maximum absolute component of a vector.
The comparisons of algorithms are given in the following context. Let $f_i^{ALG1}$and $f_i^{ALG2}$ be the optimal value found by $ALG1$ and $ALG2$, for problem $i = 1, ..., 750$ , respectively. We say that, in the particular problem $i$, the performance of $ALG1$ was better than the performance of $ALG2$ if:

$$|f_i^{ALG1} - f_i^{ALG2}| < 10^{-3} \tag{3.33}$$

and the number of iterations, or the number of function-gradient evaluations, or the CPU time of $ALG1$ was less than the number of iterations, or the number of function-gradient evaluations, or the $CPU$ time corresponding to $ALG2$, respectively.
We compare the New method CGBRB with the steepest descent method, the CG DESCENT method, and PRP conjugate gradient method.
Figures 1â4 list the performance of the CGBRB, steepest descent, CG DESCNET and PRP conjugate gradient methods. Relative to CPU time, the number of iterations and the number of gradient evaluations, respectively, which were evaluated using the profiles of Dolan and More [7].
Clearly, Figures 1â4 present that our proposed method CGBRB exhibits the best overall performance since it illustrates the highest probability of being the optimal solver, followed by the steepest descent, CG DESCNET, and PRP conjugate gradient methods relative to all performance metrics .

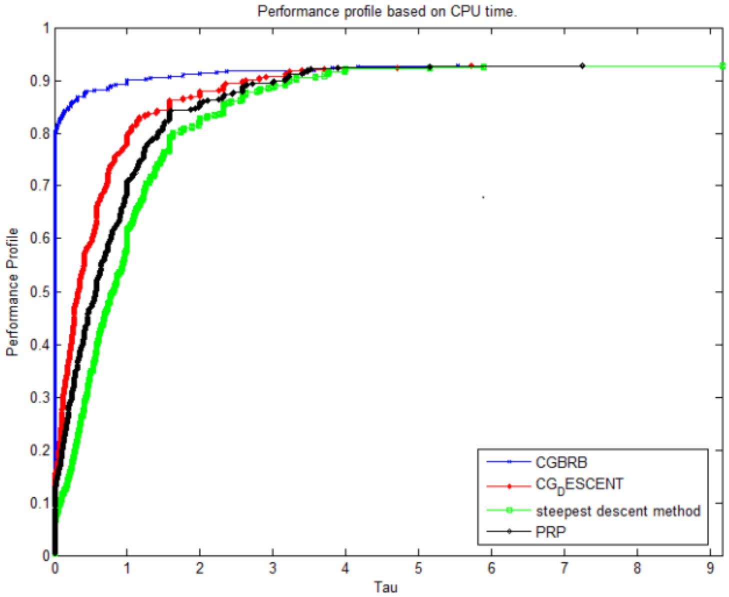


Figure 3.1: Performance based on CPU time.

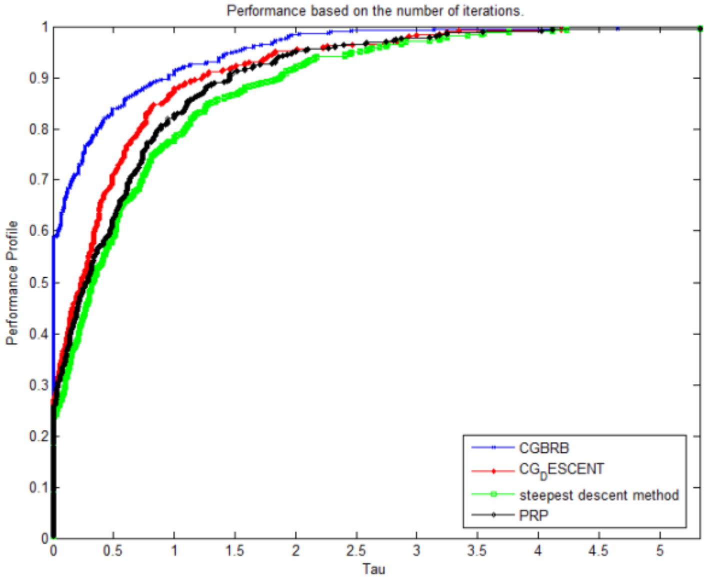


Figure 3.2: Performance based on the number of iterations.

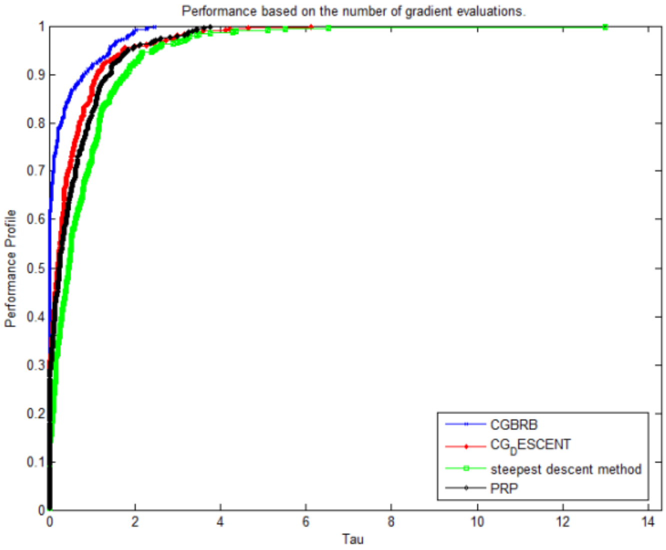


Figure 3.3: Performance based on the number of function evaluations.

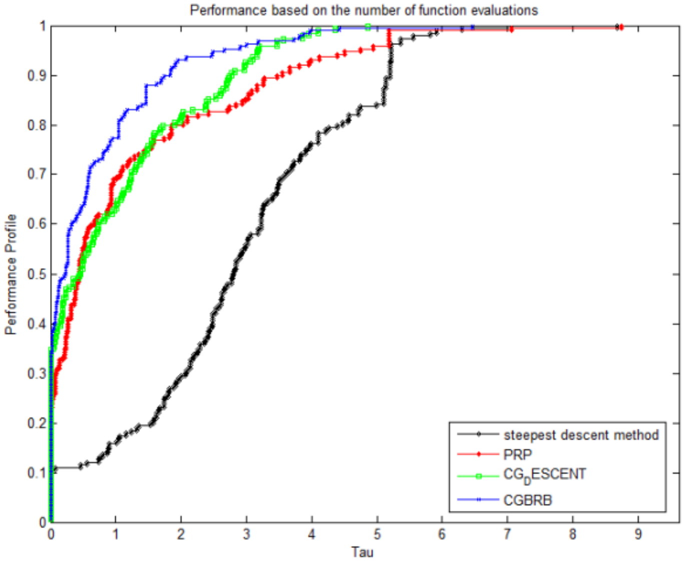


Figure 3.4: Performance based on the number of gradient evaluations.

## 3.4 Conclusion

We have presented a new conjugate gradient algorithm for solving unconstrained optimization problems. Under the modified strong Wolfe line search conditions we proved the descent property and global convergence of the algorithm. For the test problems, the comparison of the numerical results shows that the new algorithms is a good search direction at every iteration.

# Refrences

[1] M. Al-Baali, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, IMA J. Numer. Anal. 5 (1985) 121-124.

[2] K. Amini, P. Faramarzi, and N. Pirfalah. A modified hestenes-stiefel conjugate gradient method with an optimal property. Optimization Methods and Software, 34(4) :770-782, 2019.

[3] N. Andrei, An unconstrained optimization test functions collection, Advanced Modeling and Optimization, An Electronic International Journal 10 (2008) 147-161.

[4] S. Babaie-Kafaki. Two modified scaled nonlinear conjugate gradient methods. Journal of Computational and Applied Mathematics, 261 :172-182, 2014.

[5] S. Babaie-Kafaki and R. Ghanbari. A descent extension of the polka-ribière- polyak conjugate gradient method. Computers  Mathematics with Applications, 68(12) :2005-2011, 2014.

[6] S. Basri and M. Mamat. A new class of nonlinear conjugate gradient with global convergence properties. Materials Today : Proceedings, 5(10) :22029-22035, 2018.

[7] M. Belloufi, R. Benzine, Descent property and global convergence of a new search direction method for unconstrained optimization, Numerical Functional Analysis and Optimization , 36(2), 169-180, Taylor  Francis Group 2014.

[8] Sh. Sadigh Behzadi and A. Yildirim, Numerical solution of lr fuzzy hunter-saxeton equation by using homotopy analysis method, Journal of Applied Analysis and Computation, 2012, Vol.2, (1):1-10.

[9] R. Benouahboun, A. Mansouri, An interior point algorithm for convex quadratic programming with strict equilibrium. RAIRO Oper. Res. 39 (2005) 13-33.

[10] A. Cauchy. Méthode gènèrale pour la résolution des systemes d'èquations simultanèes. Comp. Rend. Sci. Paris, 25(1847) :536-538, 1847.

[11] M.Q. Chen, S. -P. Han, A parallel quasi-Newton method for partially separable large scale minimization, Annals of Operations Research, 1988, Volume 14, Issue 1 , pp 195-211

[12] Y. Dai, J. Han, G. Liu, D. Sun, H. Yin, and Y.-X. Yuan. Convergence properties of nonlinear conjugate gradient methods. SIAM Journal on Optimization, 10(2) :345-358, 2000.

[13] Y.H. Dai, Y. Yuan, An Efficient Hybrid Conjugate Gradient Method for Unconstrained Optimization, Annals of Operations Research, March 2001, Volume 103, Issue 1-4 , pp 33-47

[14] Z. Dai. Comments on a new class of nonlinear conjugate gradient coefficients with global convergence properties. Applied Mathematics and Computation, 276 :297-300, 2016.

[15] H. Degaichia, (2020). Optimisation Sans Contraintes.

[16] Sarra Delladji, Mohammed Belloufi, Badereddine Sellami, Behavior of the combination of PRP and HZ methods for unconstrained optimization, Numerical Algebra Control And Optimization, (2020), doi:10.3934/naco.2020032

[17] S. Deng, Coercivity properties and well-posedness in vector optimization. RAIRO Oper. Res. 37 (2003) 195- 208.

[18] J. E. Dennis, Jr., J. J. More, (1977). Quasi-Newton methods, motivation and theory. SIAM Review, 19, 46â89.

[19] D. Di Serafino, V. Ruggiero, G. Toraldo, and L. Zanni. On the steplength selection in gradient methods for unconstrained optimization. Applied Mathematics and Computation, 318 :176-195, 2018.

[20] E. D. Dolan and J. J. Morè. Benchmarking optimization software with performance profiles. Mathematical programming, 91(2) :201-213, 2002.

[21] X. L. Dong, H. W. Liu, and Y. B. He. New version of the three-term conjugate gradient method based on spectral scaling conjugacy condition that generates descent search direction. Applied Mathematics and Computation, 269 :606-617, 2015.

[22] X. L. Dong, H. W. Liu, Y. B. He, and X. M. Yang. A modified hestenes-stiefel conjugate gradient method with sufficient descent condition and conjugacy condition. Journal of Computational and Applied Mathematics, 281 :239-249, 2015.

[23] X. Dong, W.j. Li, Y.b. He, Some modified Yabe-Takano Conjugate gradient methods with Sufficient descent condition. RAIRO-Oper. Res. 2016.DOI: http://dx.doi.org/10.1051/ro/2016028.

[24] X. Du, P. Zhang, and W. Ma. Some modified conjugate gradient methods for unconstrained optimization. Journal of Computational and Applied Mathematics, 305 :92- 114, 2016.

[25] M. Fatemi. A new efficient conjugate gradient method for unconstrained optimization. Journal of Computational and Applied Mathematics, 300 :207-216, 2016.

[26] M.C. Ferris, A.J. Wathen, P. Armand, Limited memory solution of bound constrained convex quadratic problems arising in video games. RAIRO Oper. Res. 41 (2007) 19-34.

[27] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. The computer journal, 7(2) :149-154, 1964.

[28] R. Fletcher, Practical Method of Optimization, second ed., Unconstrained Optimization, vol. I, Wiley, New York, 1997.

[29] R. Fletcher. Practical method of optimization. unconstrained optimization, ed, 1997.

[30] R. Fletcher. Practical methods of optimization. 1987. John and Sons, Chichester, 1987.

[31] R. Fletcher. Practical methods of optimization. John Wiley  Sons, 2013.

[32] W. W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM Journal on Optimization 16 (1) :170-192, 2005.

[33] M. R. Hestenes, E. Stiefel, et al. Methods of conjugate gradients for solving linear systems. Journal of research of the National Bureau of Standards, 49(6) :409-436, 1952.

[34] M. R. Hestenes, E. Stiefel, Method of conjugate gradient for solving linear equations, J. Res. Nat. Bur. Stand. 49 (1952) 409â436.

[35] J. Jian, L. Han, and X. Jiang. A hybrid conjugate gradient method with descent property for unconstrained optimization. Applied Mathematical Modelling, 39(3- 4) :1281-1290, 2015.

[36] S. B. Kafaki, R. Ghanbari, A descent hybrid modification of the PolakâRibiereâPolyak conjugate gradient method. RAIRO-Oper. Res. 50 (2016) 567-574.

[37] Y. Liu and C. Storey. Efficient generalized conjugate gradient algorithms, part 1 : theory. Journal of optimization theory and applications, 69(1) :129-137, 1991.

[38] J. Liu and S. Li. New hybrid conjugate gradient method for unconstrained optimization. Applied Mathematics and Computation, 245 :36-43, 2014.

[39] J. Li, T. Zhou, and C. Wang. On global convergence of gradient descent algorithms for generalized phase retrieval problem. Journal of Computational and Applied Mathematics, 329 :202-222, 2018.

[40] Y. Liu, C. Storey, Efficient generalized conjugate gradient algorithms. Part 1: Theory, J. Optimiz. Theory Appl. 69 (1992) 129-137. A NEW CONJUGATE GRADIENT METHOD FOR ACCELERATION OF GRADIENT DESCENT ALGORITHMS 11 .

[41] J. Nocedal, Conjugate gradient methods and nonlinear optimization, in: L. Adams, J.L. Nazareth (Eds.), Linear and Nonlinear Conjugate Gradient Related Methods, SIAM, Philadelphia, PA, 1995, pp. 9-23.

[42] J. Nocedal, S.J. Wright, Numerical optimization, Springer Series in Operations Research, Springer, New York, 1999.

[43] B. T. Polyak. The conjugate gradient method in extremal problems. USSR Computational Mathematics and Mathematical Physics, 9(4) :94-112, 1969.

[44] M.J.D. Powell, Nonconvex minimization calculations and the conjugate gradient method, Lecture Notes in Math. 1066 (1984) 121-141.

[45] N.Rahali,these doctorat Noureddine.Rahali , http://dspace.univ-setif.dz:8888/jspui/handle/12345

[46] N. Rahali, Belloufi, M., Benzine, R. (2021). A new conjugate gradient method for acceleration of gradient descent algorithms. Moroccan Journal of Pure and Applied Analysis, 7(1), 1-11.

[47] G. Ribière and E. Polak. Note sur la convergence de directions conjugèes. Rev. Francaise Informat Recherche Opertionelle, 16 :35-43, 1969.

[48] M. A. Rincon, M. I. M. Copetti, Numerical analysis for a locally damped wave equation, Journal of Applied Analysis and Computation, 2013, Vol.3, (2):169-182.

[49] M. Rivaie, M. Mamat, L. W. June, and I. Mohd. A new class of nonlinear conjugate gradient coefficients with global convergence properties. Applied Mathematics and Computation, 218(22) :11323-11332, 2012.

[50] J. Schropp, A note on minimization problems and multistep methods. Numeric Mathematic, 78 (1997) 87-101.

[51] Z.J. Shi, X.S. Zhang, J. Shen, Convergence analysis of adaptive trust region methods. RAIRO Oper. Res. 41 (2007) 105-121.

[52] W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM Journal on optimization, 16(1) :170-192, 2005.

[53] L. Wang, W. Sun, R. J. de Sampaio, and J. Yuan. A barzilai and borwein scaling conjugate gradient method for unconstrained optimization problems. Applied Mathematics and Computation, 262 :136-144, 2015.

[54] Y.-J. Xie and C.-F. Ma. The scaling conjugate gradient iterative method for two types of linear matrix equations. Computers  Mathematics with Applications, 70(5) :1098-1113, 2015.

[55] C. Xu, J. Zhang, A Survey of Quasi-Newton Equations and Quasi-Newton Methods for Optimization, Annals of Operations Research,2001, Volume 103, Issue 1-4 , pp 213-234.

[56] Y. Yuan, W. Sun, et al. Theory and methods of optimization, 1999.

[57] C. Xu, J. Zhang, A Survey of Quasi-Newton Equations and Quasi-Newton Methods for Optimization, Annals of Operations Research,2001, Volume 103, Issue 1-4 , pp 213-234.

[58] N.H. Zhao, Y. Xia, W. Liu, P. J. Y. Wong and R. T. Wang, Existence of Almost Periodic Solutions of a Nonlinear System, Journal of Applied Analysis and Computation, 2013, Vol.3, (3):301-306.

[59] N.H. Zhou, and C. Wang. On global convergence of gradient descent algorithms for generalized phase retrieval problem. Journal of Computational and Applied Mathematics, 329 :202-222, 2018.