République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**UNIVERSITÉ MOHAMED KHIDER, BISKRA**

FACULTÉ des SCIENCES EXACTES et des SCIENCES de la NATURE et de la VIE

**DÉPARTEMENT DE MATHÉMATIQUES**

Mémoire présenté en vue de l'obtention du Diplôme **:**

**MASTER en Mathématiques**

Option : **Analyse**

Par

**Lekbiche Karima**

Titre :

# La méthode de gradient conjugué en optimisation et ses application

Membres du Comité d'Examen :

| | | | |
|---|---|---|---|
| Dr. | KABOUL Hanane | UMKB | Président |
| Dr. | BELLAGOUN A.Ghani | UMKB | Encadreur |
| Dr. | BENABA Fadhila | UMKB | Examinateur |

Juin 2018

## Dédicace

j'ai l'hounneur de dédier ce travail à :

Mon chér pére, qui a consacré sa vie pour mon éducation.

Ma chére maman pour toute sa tendresse, amour, et affection.

Mes chers soeurs : Dalila, Djamila et Wahiba.

Mes chére frères : Nadir, Oussama et Hamza.

Toutes mes amies.

Tous mes formateurs pour leur effort et leur amabilité et tous qui m'ont aidé de prés ou de loin pour la réalisation de ce travail.

# REMERCIEMENTS

Tout d'abord, je remercie DIEU le tout-puissant pour la volonté,la santé et la patience qu'il m'a donnée durant ma vie.

je tiens à remercier particuliément Mr BELLAGOUN A.Ghani pour l'aide et les consiels concernant ce travail.

je remarcie également Ms KABOUL Hanane et Ms BENABA Fadhila.

je remarcie également les membres de nos familles qui nous ont offert soutien moral et financier .

A tous ceux qui nous ont guidès avec gentillesse et efficacité.

# Table des matières

# Liste des tableaux

# Table des figures

# Introduction

Le but de ce travail est de montrer certains resultats concernant la théorie de l'optimisation continue sans contraintes et la notion de convergence, les méthodes et les algorithmes associés sont d'une importance capitale pour les applications, l'optimisation contrainte est baseé essentiellement sur la notion du gradient et du gradient conjugueé, la plupart des travaux consacrés à la minimisation des critéres utilisent le gradient mais avec des variantes. Historiquement les méthode du gradient conjugué sont parus des les anneés 40, òu le debut des calculateurs analogique ont encouragé le developpement de ces méthodes numiriques .

Pour les applications la méthode du gradient conjugué est utilisé particulierement dans le traitement des signaux et la restouration d'images donc dans le secteurs imagené médicale et autrs secteurs industriels.

Le travail est divisé en 3 chapitres

- Le premier chapitre est consacré aux géniralités et autres definitions concernant la méthode du gradient, nous essayons de doner la relation entre l'optimisation et la méthode de gradient, les resultats sont classique et pouvont être consulté dans la bibliographé.
- Le deuxieme chapitre est basé essentiellement sur la notion de convergence dans les méthode des gradients et de la vitésse de convergence qui joue un grand rôle pour optimiser les resultats
- Le troisième chapitre traite quelques application de la méthode du gradient conjugué avec des resultat numériques.

# Chapitre 1

# Introduction

Dans ce chapitre, nous considérons la classe des fonctions à valeurs réelles ou les méthodes du gradient sont utilisées pour optimiser les fonctions données. Dans notre discussion, nous employons des termes comme les courbes de niveau, vecteurs normaux et vecteurs tangentes.

On rappale qu'un ensemble de niveau d'une fonction $f : \mathbb{R}^n \to \mathbb{R}$ est l'ensemble de points $x$ satisfaisant $f(x) = c$ pour un certain $c$ constant. Ainsi, un point $x_0 \in \mathbb{R}^n$ est sur l'ensemble de niveau correspondant au niveau $c$ si $f(x_0) = c$. Le gradient de $f$ en $x_0$ noté $\nabla f(x_0)$.

La direction du taux d'accroissement maximum d'une fonction différentiable à valeurs réelles à un point est orthogonale à l'ensemble de niveau de la fonction en ce point. En d'autres termes, le gradient agit dans une telle direction qui pour un petit déplacement donné, la fonction $f$ augmentera plus dans la direction du gradient que dans toute autre direction. Pour prouver ce rapport, se rappeler que $\langle \nabla f(x), d\rangle$, $||d|| = 1$, est le taux d'accroissement de $f$ dans la direction $d$ au point $x$. Par l'inégalité de Cauchy-Schwarz.

$\langle \nabla f(x), d\rangle \leq \|\nabla f(x)\|$

parce que $||d|| = 1$. Mais si $d = \frac{\nabla f(x)}{\|\nabla f(x)\|}$ alors

$$\langle \nabla f(x), \frac{\nabla f(x)}{\|\nabla f(x)\|}\rangle = \|\nabla f(x)\|$$

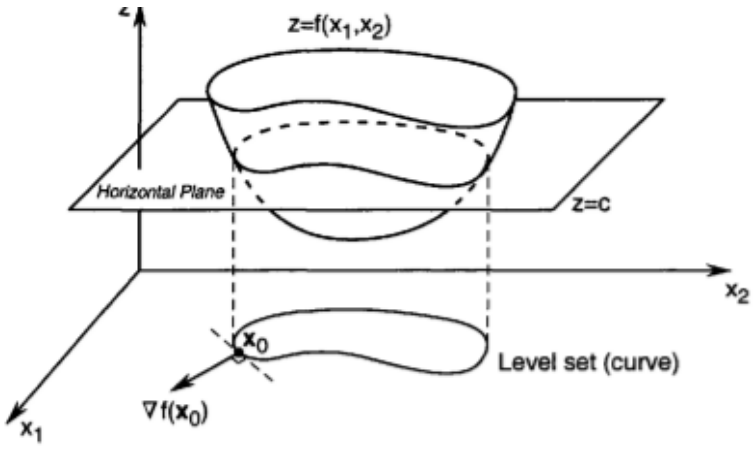


FIG. 1.1 – construction d'un ensemble de niveau correspondant au niveau $\alpha$ pour $f$

Ainsi, la direction dans laquelle $\nabla f(x)$ se dirige est la direction du taux maximum de d'augmentation de $f$ en $x$.

La direction dans la quelle $-\nabla f(x)$ se dirige est la direction de le taux maximum de diminution de $f$ en $x$. par conséquent, la direction du gradient négatif est une bonne direction à rechercher si nous voulons trouver un minimiseur de fonction.

Nous opérons comme suit. Soit $x_k$ un point de départ, et considérons le point $x_0 - \alpha\nabla f(x)$. Puis, par le théorème de Taylor nous obtenons

$$f(x_0 - \alpha\nabla f(x)) = f(x_0) - \alpha \left\|\nabla f(x)\right\|^2 + 0(\alpha)$$

Ainsi, si $\nabla f(x_0) \neq 0$ , puis pour $\alpha > 0$, suffisamment petit nous avons

$$f(x_0 - \alpha\nabla f(x_0)) < f(x_0)$$

Ceci signifie que le point $x_0 - \alpha\nabla f(x_0)$ est une amélioration au-dessus du point $x_0$ si nous recherchons un minimiseur.

Pour formuler un algorithme qui met en application l'idée ci-dessus, supposons que nous partons d'un point $x_k$. pour trouver le point $x_{k+1}$, nous commençons par $x_k$

et on se déplaçons avec la quantité $-\alpha_k\nabla f(x_k)$,où $\alpha_k$ est une grandeur scalaire positive

appelée le pas à l'étape $k$. Le procédé ci-dessus mène à l'algorithme itératif suivant :

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Ainsi la méthode est connue sous le nom l'algorithme de gradient de descente (ou tout simplement l'algorithme de gradient).

Le gradient varie en fonctions des itérations et tendant en norme vers zéro quant nous nous rapprochons du minimum. Nous avons l'option ou de prendre des mesures très petites et de réévaluer le gradient à chaque étape.

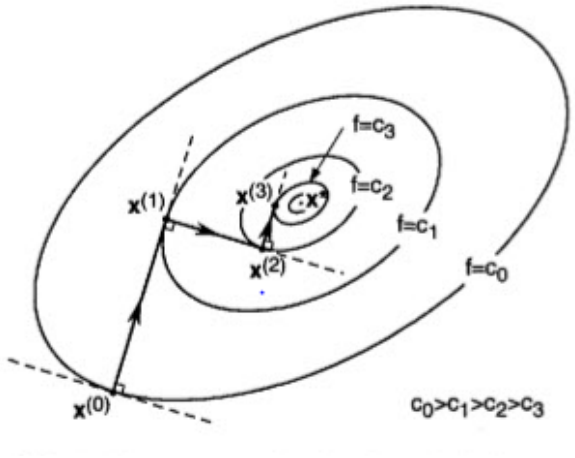


FIG. 1.2 – l'ordre topiyque resultant de la méthode de descent

La plus populaire est la méthode de la descente, que nous allons discuter après.

Les méthodes de gradient sont simples à mettre en application . Pour cette raison, elles sont employées couramment dans des applications pratiques. On peut trouver des applications de la méthode de la descente au calcul des contrôls optimaux.

## 1.1 La Méthode de Gradient (descent).

La méthode de la descente est un algorithme de gradient de taille $\alpha_k$ d'étape $k$ est choisi pour réaliser le taux de décroissance de la fonction objective à chaque étape. spécifique-

ment, $\alpha_k$ est choisi de tel façon pour réduire au minimum l'expression :

$$\Phi_k(\alpha) \triangleq f(x_k - \alpha \nabla f(x_k))$$

En d'autres termes $\alpha_k = \arg\min_{\alpha \geq 0} f(x_k - \alpha \nabla f(x_k))$, Pour récapituler, l'algorithme de la descente de la plus grande pente procède comme suit : à chaque étape, à partir de point $x_k$ nous déterminons une ligne de recherche dans la direction $-\nabla f(x_k)$ jusqu'à un minimiseur, $x_{k+1}$, est trouvé.

Un ordre typique résultant de la méthode de la descente (FIG 1.2)

Nous observons que la méthode de la descente se déplace orthogonalement d'une étape à l'autre , comme indiqué dans la proposition suivante **:**

Si $\{x_{(k)}\}_{k=0}^{\infty}$ est une suite décroissante in

duite par la descente pour une fonction donnée

$f : \mathbb{R}^n \to \mathbb{R}^n$ puis pour chaque $k$ le vecteur $x_{k+1}$ est orthogonal au vecteur $x_{k+2} - x_{k+1}$

De la formule itérative de la méthode de la descente on a :

$$\langle x_{k+1} - x_k, x_{k+2} - x_{k+1} \rangle = \alpha_k \alpha_{k+1} \langle \nabla f(x_k), \nabla f(x_{k+1}) \rangle$$

Montrons que :

$$\langle \nabla f(x_k), \nabla f(x_{k+1}) \rangle = 0$$

à cet effet, observons que $\alpha_k$ est une grandeur scalaire non négative qui réduit au minimum

$\Phi_k(\alpha) \triangleq f(x_k - \alpha \nabla f(x_k))$

d'ou

$$\begin{aligned} 0 &= \Phi_k'(\alpha_k) \\ &= \frac{d\Phi_k}{d\alpha}(\alpha_k) \\ &= \nabla f(x_k - \alpha_k \nabla f(x_k))^\top (-\nabla f(x_k)) \\ &= -\langle \nabla f(x_{k+1}, \nabla f(x_k) \rangle \end{aligned}$$

et la preuve est accomplie.

Notons qu' à chaque étape de l'algorithme de la descente , la valeur correspondante de la fonction $f$ diminue en valeur par rapport à l'étape précédente, comme indiqué ci-dessous.

**Proposition 1.1.1**

si$\{x_k\}_{k=0}^{\infty}$ est une suite induite par la méthode de la descente pour une fonction $f : R^n \to R$ et si $\nabla f(x_k) \neq 0$ , alors :

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

Preuve. Nous avons

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

de là où $\alpha_k > 0$ est le minimiseur de

$$\Phi_k(\alpha) = f(x_k - \alpha \nabla f(x_k))$$

pour tout $\alpha \geq 0$

$$\Phi_k(\alpha_k) \leq \Phi(\alpha)$$

Par la dérivée des fonctions composées :

$$\Phi'_k(0) = \frac{\partial \Phi_k}{\partial \alpha}(0) = -\nabla f(x_k - 0.\nabla f(x_k))^\top (\nabla f(x_k)) = -\left\|\nabla f(x_k)\right\|^2 < 0$$

parce que

$$\nabla f(x_k) \neq 0$$

Ainsi

$$\Phi'_k(0) < 0$$

et ceci implique qu'il existe $\tilde{\alpha} > 0$ tels que $\Phi_k(0) > \Phi_k(\alpha)$pour le tout un $\alpha \in (0, \tilde{\alpha}]$. Par conséquent,

$$f(x_{k+1}) = \Phi_k(\alpha_k) \leq \Phi_k(\alpha) < \Phi_k(0) = f(x_k)$$

et la preuve du rapport est accomplie.

Dans ce qui précède, nous avons montré que l'algorithme possède la propriété de descente :

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

si $\nabla f(x_k) \neq 0$, pour certains $k$ , nous avons $\nabla f(x_p) = 0$, $p > k$ puis

le point $x_k$ ; satisfait le l'hypothèse de minimisation. dans ce Cas $x_{k+1}$= $x_k$.

Nous pouvons la prendre comme base pour un critère (d'arrêt) pour l'algorithme.

La condition $\nabla f(x_{k+1}) = 0$, cependant, n'est pas directement appropriée en tant que pratique, parce que le calcul numérique du gradient est rarement identiquement égal à zéro.

En pratique l'arrê d'un critère est de vérifier si la norme $||\nabla f(x_k)|| = 0$ du gradient est plus petit qu'un seuil spécifié.

Alternativement, nous pouvons calculer la différence absolue $|f(x_{k+1}) - f(x_k)|$

entre les valeurs de la fonction objective pour chaque deux itérations successives, et si

la différence est inférieure à un certain seuil spécifié,nous nous arrêtons ; c'est-à-dire, quand

$$|f(x_{k+1}) - f(x_k)| < \varepsilon$$

Là où $\varepsilon > 0$ est un seuil préspécifié. Encore une autre alternative est de calculer la Norme $\|x_{k+1} - x_k\|$ de la différence entre deux solutions successives , et nous nous arrêtons si la norme $\|x_{k+1} - x_k\|$ est plus petit que $\varepsilon$.

Alternativement, nous pouvons vérifier des valeurs « relatives » des quantités ci-dessus ; par exemple

$$\frac{|f\ (x_{k+1}) - f\ (x_k)|}{|f\ (x_k)|} < \varepsilon$$

ou

$$\frac{\|x_{k+1} - x_k\|}{\|x_k\|} < \varepsilon$$

Les deux expressions ci-dessus sont aussi des critères d'arret relatifs mesurées indépendaments. Par exemple, la mesure de la fonction objective ne change pas le critère d'arrêt :

$$\frac{\left|f(x^{(k+1)}) - f(x^{(k)})\right|}{|f(x^{(k)})|} < \varepsilon$$

De même, le mesure de la variable de décision ne change pas critère d'arrêt

Pour éviter la division par des nombres très petits, nous pouvons modifier ces critères d'arrêt comme suit :

$$\frac{|f(x_{k+1}) - f(x_k)|}{\max(1, |f(x_k)|)} < \varepsilon$$

ou

$$\frac{\|x_{k+1} - x_k\|}{\|x_k\|} < \varepsilon$$

Notons que les critères d'arrêt ci-dessus sont appropriés à tous les algorithmes itératifs.

$$f(x_1, x_2, x_3) = (x_1 - 4)^4 + (x_2 - 3)^2 + 4(x_3 + 5)^4$$

de point initial $x_0 = [4, 2, -1]^\top$

nous trouvons

$$\nabla f(x) = \left[4(x_1 - 4)^3, 2(x_2 - 3), 16(x_3 + 5)^3\right]^\top$$

Par conséquent

$$\nabla f(x_0) = [0, -2, 1024]$$

pour calculer $x_1$ il faut de trouver $\alpha_{0,}$dabord

$$\begin{aligned}
\alpha_0 &= \arg\min_{\alpha \geq 0} f\ (x_0 - \alpha \nabla f(x_0) \\
&= \arg\min_{\alpha \geq 0} (0 - (2 + 2\alpha - 3)^2 + 4(-1 - 1024\alpha + 5)) \\
&= \arg\min_{\alpha \geq 0} \Phi_0(\alpha)
\end{aligned}$$

on utilise la méthode de secant on trouve

$$\alpha_0 = 3.967 \times 10^{-3}$$

Pour le but d'illustration, nous montrons une parcelle de terrain de$\Phi_0(\alpha)$ contre a $\alpha$ sur le schéma 1.3, obtenu utilison MATLAB.Ainsi,

$x_1 = x_0 - \alpha_0 \nabla f(x_0) = [0.000, 1.003, -1.3875]$

et

Nous procédons comme dans les itérations précédentes pour obtenir $\alpha_1 = 0.500$

$\alpha_2 = 16.29$. Voir schéma 1.4 ,1.5

avec

La valeur de $x_2$est

$$x_2 = [4.00, 3.000, -5,060]$$

La valeur de $x_3$ est

$$x_3 = [4.000, 3.00, -5.002]$$

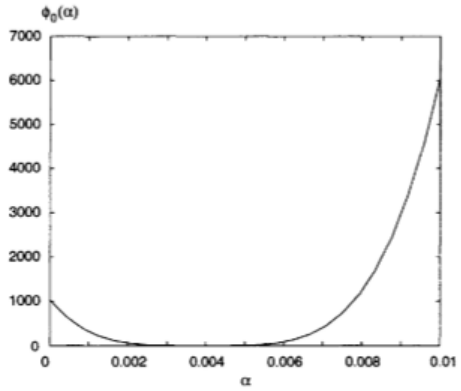


FIG. 1.3 – graphe de $\Phi_0(\alpha)$

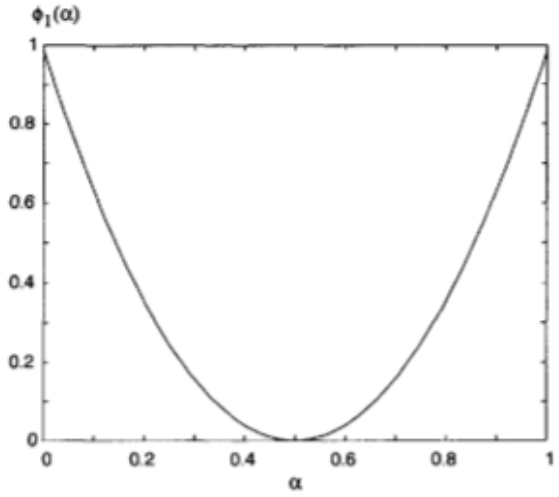


FIG. 1.4 – graphe de $\Phi_1(\alpha)$

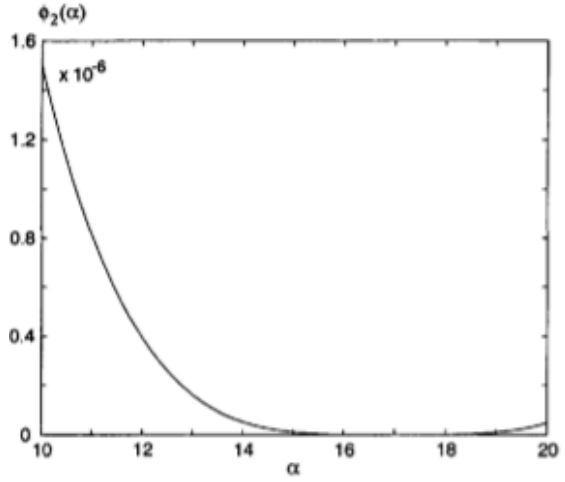


FIG. 1.5 – graphe de $\Phi_2(\alpha)$

Noter que le minimiseur de $f$ est $[4, 3, -5]^\top$ , et par conséquent il s'avère que nous sommes arrivés au minimum dans seulement trois itérations.

Les calculs numériques concernant cet exemple,

sont exécutés en utilisant le logociel Matlab. Les calculs ci-dessus ont été écrits explicitement, étape-par-étape.

## 1.2 Les méthodes de gradient conjugué

**Notation 1.2.1** *:*

**1.** Si $u_1, u_2, ..., u_p \in \mathbb{R}^n$On note$\mathcal{L}\ (u_1, u_2, ..., u_p) = \left\{ \sum_{i=0}^{p} \alpha_i u_i, \alpha_1 ... \alpha_p \in \mathbb{R} \right\}$l éspace vectoriel engendré par les vecteures $u_1, u_2, ..., u_p$, Cest un sous espace vectoriel de $\mathbb{R}^n$.

**2.** Si $a \in \mathbb{R}^n$et $M \subset \mathbb{R}^n$ alors $a + M$ désigne l'ensemble $\{a + x, x \in M\}$.

**Théorème 1.2.1** *Soit $\Omega \subset \mathbb{R}^n$un ouvert, $U \subset \Omega$ un ensemble convexe et $f : \Omega \longmapsto \mathbb{R}$ une fonction de classe $C^1$,soit $v^* \in U$ un minimum relatif de $f$ sur $U$.*

alors :

$$\langle \nabla f(u^*), u - u^* \rangle \geq 0 \ \forall \ u \in U$$

### 1.2.1 Le cas quadratique

Dans ce pragraphe on considère $f : \mathbb{R}^n \longmapsto \mathbb{R}$ une fonction quadratique

$$f(u) = \frac{1}{2} \langle Au, u \rangle - \langle b, u \rangle + c \ \forall u \in \mathbb{R}^n$$

avec $A \in M_n(\mathbb{R})$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$.On supose que la matrice $A$ est symétrique définie et positive.

**Idée de la méthode** : Rappellons que la méthode de gradient à pas optimal consiste à éccrire :

$$\begin{aligned} u^{(k+1)} &= u^{(k)} - \rho_k \nabla f(u^{(k)}) \\ &= \min_{\rho \in \mathbb{R}} f(u^{(k)} - \rho \nabla f(u^{(k)})) \end{aligned}$$

Ceci est équivalent à :

$$u^{(k+1)} \in u^{(k)} + \mathcal{L}(\nabla f(u^{(k)}))$$

est l'élement qui minimise $f$ sur $u^{(k)} + \mathcal{L}(\nabla f(u^{(k)}))$

Dans la suite on va procéder de la manière suivante :

On va noter pour tout $k \in \mathbb{N}$

$$G_k = \mathcal{L}(\nabla f(u^{(0)}), \nabla f(u^{(1)}), ...\nabla f(u^{(k)})) \subset \mathbb{R}^n$$

La méthode du gradient conjugué consiste à chercher

$$f(u^{(k+1)}) = \min_{\upsilon \in u^{(k)} + G_k} f(\upsilon) \tag{1.1}$$

(en supposant qu'un tel minimum existe).

On minimise donc sur un espace plus "grand" que dans la méthode de gradient à pas optimal, On s'attend alors à un "meilleur" minimum, Il reste à montrer que cette méthode est facile à implémenter et qu'elle donne le résultat attendu.

Comme $f$ est elliptique et que $u^{(k)} + G_k$ est un ensemble fermé et convexe (car c'est un espace affine), alors il existe une solution unique du problème de minimisation 1.1.

En plus, le Théorème 1.2.1 nous donne

$$\left\langle \nabla f(u^{(k+1)}), \upsilon - u^{(k+1)} \right\rangle \geq 0 \forall \upsilon \in u^{(k)} + G_k \tag{1.2}$$

soit maintenant $w \in G_k$ arbitraire remarquons qu'ona :

$u^{(k+1)} + w = u^{(k)} + \left[u^{(k+1)} - u^{(k)}\right] + w \in u^{(k)} + G_k$ car$u^{(k+1)} - u^{(k)} \in G_k$.

On peut alors prendre $\upsilon = u^{(k+1)} + w$ en 1.2 et aussi $\upsilon = u^{(k+1)} - w$ on obtient

$$\left\langle \nabla f(u^{(k+1)}), w \right\rangle = 0 \quad \forall w \in G_k$$

( c'est à dire :$\nabla f(u^{(k+1)})$est **orthogonal** sur tous les vecteurs de $G_k$).Ceci nous donne

$$\left\langle \nabla f(u^{(k+1)}), \nabla f(u^{(l)}) \right\rangle = 0 \ \forall l = 0, 1, ..k \tag{1.3}$$

(dans l'algorithme de gradient à pas optimale on avait $\left\langle \nabla f(u^{(k+1)}), \nabla f(u^{(l)}) \right\rangle = 0$).

**conséquences :**

- Si les vecteurs $\nabla f(u^{(0)}), \nabla f(u^{(1)}), ...\nabla f(u^{(k)})$sont tous $\neq$ 0alors ils sont linéairement indépendants (c'est une conséquence immédiate du résultat bien connu : si des vecteurs non nuls sont orthogonaux par rapport à un produit scalaire, alors ils sont indépendants)
- L'algorithme s'arrête en au plus $n$ itérations, car il existe $k \in \{0, 1, ...n\}$tel que

$$\nabla f(u^{(k)}) = 0$$

(sinon on aurait $n + 1$vecteurs non-nuls indépendants en $\mathbb{R}^n$ ce qui est impossible)

Alors $u^{(k)}$ est la solution rechérchée.

La question qui se pose maintenant est : **comment calculer $u^{(k+1)}$ à partir de $u^{(k)}$?**

Supposons qu'on a

$$\nabla f(u^{(i)}) \neq 0, \forall i = 0, 1, ...k$$

(sinon, l'algorithme s'arrêtait avant d'avoir à calcule $u^{(k+1)}$)

Nous avons l'expression suivante :

$$u^{(l+1)} = u^{(l)} + \triangle_l \ \ \forall l = 0, 1, ..k \tag{1.4}$$

avec $\triangle_l \in G_l$ que nous écrivons sous la forme :

$$\triangle_l = \sum_{i=0}^{l} \alpha_i^l \ \nabla f(u^{(i)}) \tag{1.5}$$

avec $\alpha_i^l \in \mathbb{R}$ des coefficients à trouver. Rappellons que

$$\nabla f(x) = Ax - b \quad \forall x \in \mathbb{R}^n$$

On utilisera souvent la formule suivante :

$$\nabla f(u^{(l+1)}) = \nabla f(u^{(l)}) + A\triangle_l \tag{1.6}$$

( car $\nabla f(u^{(l+1)}) = Au^{(l+1)} - b = A(u^{(l)} + \triangle_l) - b = Au^{(l)} - b + A\triangle_l$)

**Proposition 1.2.1** *On a*

1.

$$\triangle_l \neq 0 \ \forall \ l = 0, 1, ...k$$

2.

$$\langle A\triangle_l, \triangle_m \rangle = 0 \ \forall \ 0 \leq m < l \leq k$$

**Preuve.** En faisant le produit scalaire de l'égalité 1.6 par $\nabla f(u^{(l)})$ on obtient ■

$$\left\|\nabla f(u^{(l)})\right\|^2 + \left\langle A\triangle_l, \nabla f(u^{(l)})\right\rangle = 0$$

Comme $\left\|\nabla f(u^{(l)})\right\|^2 \neq 0$ on déduit $\left\langle A\triangle_l, \nabla f(u^{(l)})\right\rangle \neq 0$ danc $\triangle_l \neq 0$ ,ce qui finit la partie 1.

Pour montrer la partie 2. nous faisons le produit scalaire de$\nabla f(u^{(i)}), i < l$ avec 1.6 .

on obtient

$$\left\langle A\triangle_l, \nabla f(u^{(i)})\right\rangle \ \forall i = 0, 1, ...l-1 \ \forall i = 0, 1, ...m$$

En multipliant par $\alpha_i^l$ et en sommant pour$i$ de 0 à $m$ on obtient l'égalité de la partie 2.

Ceci nous amème à donner la définition suivante :

**Définition 1.2.1** *On dit que deux vecteurs $x$ et $y \in \mathbb{R}^n$ sont **conjugués** par raport à une matrice $B \in M_n(\mathbb{R})si$*

$$\langle Bx, y\rangle = 0$$

**Remarque 1.2.1** *: Si $B$ est une matrice symétrique définie et positive alors l'application*

$$(x, y) \in \mathbb{R}^n \times \mathbb{R}^n \to \langle Bx, y\rangle \in \mathbb{R}^n$$

est un produit scalaire en $\mathbb{R}^n$ (résultat admis !) qu'on appelle produit scalaire associé à la matrice $B$. (à noter que le produit scalaire habituel est un fait un produit scalaire associé à la matrice identité $I_n$) Alors la définition précédente nous dit que, dans le cas où B est symétrique définie et positive , deux vecteur sont conjugués par rapport à la matrice $B$ s'ils sont orthogonaux par rapport au produit scalaire associé à $B$ .
Comme les vecteurs $\triangle_0, \triangle_1, ...\triangle_k$ sont non-nuls et orthogonaux par rapport au produit scalaire associé à la matrice$A$.

**Proposition 1.2.2** *: les vecteurs $\triangle_0, \triangle_1, ..., \triangle_k$ sont indépendants. Il est facile de voir qu'on a*

$$\mathcal{L}(\triangle_0, \triangle_1, ...\triangle_1) = \mathcal{L}(\nabla f(u^{(0)}), \nabla f(u^{(1)}), ...\nabla f(u^{(1)})), \forall l = 0, 1, ...k$$

(car l'inclusion " $\subset$" est évidente de 1.5 et en plus les deux espaces ont la même dimension : $l + 1$
On déduit alors :

$$\alpha_l^l \neq 0, \forall\ l = 0, 1, ...k \tag{1.7}$$

(car sinon, on aurait $\triangle_l = \mathcal{L}(\nabla f(u^{(0)}), \nabla f(u^{(1)}), ...\nabla f(u^{(1)})) = \mathcal{L}(\triangle_0, \triangle_1, ...\triangle_1)$, ce qui contredirait l'indépendance des $\triangle_0, ...\triangle_1$)
On peut alors écrire :

$$\triangle_k = \rho_k d^{(k)}$$

avec

$$d^{(k)} = \nabla f(u^{(k)}) + \sum_{i=0}^{k-1} \lambda_i^k \nabla f(u^{(i)}) \tag{1.8}$$

où on a posé

$$\begin{cases} \rho_k = \alpha_k^k \\ \lambda_i = \frac{\alpha_i^k}{\alpha_k^k} \end{cases} \tag{1.9}$$

(remarquons que dans 1.8 la somme n'existe pas si $k = 0$)

Nous avons donc :

$$u^{(k+1)} = u^k + \rho_k d_k$$

avec $d_k$ donné par 1.8 et 1.9.

Il nous reste à calculer les $\rho_k$ et $d_k$.

comme $\langle A\triangle_k, \triangle_l \rangle = 0$ pour $l < k$ , on déduit que $\langle Ad_k, \triangle_l \rangle = 0$ ce qui nous donne en utilisant aussi 1.6 :

$0 = \langle d_k, A\triangle_l \rangle = \left\langle d_k, \nabla f(u^{(l+1)}) - \nabla f(u^{(l)}) \right\rangle$ pour $0 \leq l \leq k-1$.

A l'aide de 1.8 cela nous donne

$$\left\langle \nabla f(u^{(k)}) + \sum_{j=0}^{k-1} \lambda_j^k \nabla f(u^{(j)}), \nabla f(u^{(l+1)}) - \nabla f(u^{(l)}) \right\rangle = 0 \text{ pour } 0 \leq l \leq k-1 \tag{1.10}$$

En prennant $l = k-1$ dans cette égalité on trouve

$$\left\| \nabla f(u^{(k)}) \right\|^2 - \lambda_{k-1}^k \left\| \nabla f(u^{(k-1)}) \right\|^2 = 0$$

alors

$$\lambda_{k-1}^{k} = \frac{\left\|\nabla f(u^{(k)})\right\|^2}{\left\|\nabla f(u^{(k-1)})\right\|^2} \tag{1.11}$$

En prennant $l \leq k-2$ en1.11,on trouve la relation de reccurence

$$\lambda_{l+1}^{k} \left\|\nabla f(u^{(l+1)})\right\|^2 - \lambda_{l}^{k} \left\|\nabla f(u^{(l)})\right\|^2 = 0$$

ce qui donne

$$\lambda_{l}^{k} = \lambda_{l+1}^{k} \frac{\left\|\nabla f(u^{(l+1)})\right\|^2}{\left\|\nabla f(u^{(l)})\right\|^2}$$

On en déduit facilement, en utilisant aussi 1.11

$$\lambda_{i}^{k} = \frac{\left\|\nabla f(u^{(k)})\right\|^2}{\left\|\nabla f(u^{(i)})\right\|^2}, \qquad i = 0, 1, ...k-1$$

On obtient alors, à l'aide de 1.8 :

$$\begin{aligned} d_k &= \nabla f(u^{(k)}) + \sum_{i=0}^{k-1} \frac{\left\|\nabla f(u^{(k)})\right\|^2}{\left\|\nabla f(u^{(k)})\right\|^2} \nabla f(u^{(i)}) \\ &= \nabla f(u^{(k)}) \frac{\left\|\nabla f(u^{(k)})\right\|^2}{\left\|\nabla f(u^{(k-1)})\right\|^2} \left[\nabla f(u^{(k-1)}) \sum_{i=0}^{k-2} \frac{\nabla f(u^{(k-1)})}{\nabla f(u^{(i)})} \nabla f(u^{(i)})\right] \end{aligned}$$

Ceci nous donne la suite$\left\{d^{(k)}\right\}$par reccurence sous la forme

$$\begin{cases} d_0 = \nabla f(u^{(0)}) \\ d_k = \nabla f(u^{(k)}) + \frac{\left\|\nabla f(u^{(k)})\right\|^2}{\left\|\nabla f(u^{(k-1)})\right\|^2} d_{k-1} \end{cases} \tag{1.12}$$

Il reste à détérminer les $\rho_k$

Rappellons qu'on a la relation de reccurence :

$$u^{(k+1)} = u^{(k)} + \rho_k d_k$$

et que nous avons

$$f(u^{(k)} + \rho_k d_k) \leq f(u^{(k)} + y),\ \forall y \in G_k$$

ce qui nous donne

$$f(u^{(k)} + \rho_k d_k) \leq f(u^{(k)} + \rho d_k), \forall \rho \in \mathbb{R}$$

car $\rho d_k \in G_k$ .On en déduit que $\rho_K$ est un point de minimum sur $\mathbb{R}$ de l'application

$$\rho \in \mathbb{R} \rightarrow f(u^{(k)} + \rho d_k) \in \mathbb{R}$$

Danc $\rho_k$ est un point où la dérivée de cette application s'annule, ce qui donne

$$\left\langle \nabla f(u^{(k)} + \rho_k d_k), d_k \right\rangle = 0$$

c'est à dire

$$\left\langle Au^{(k)} + \rho_k A d_k - b, d_k \right\rangle = 0$$

On obtient alors

$$\rho_k = \frac{\left\langle \nabla f(u^{(k)}), d_k \right\rangle}{\langle Ad_k, d_k \rangle} \tag{1.13}$$

**Remarque 1.2.2** *:Proposition (2) nous dit que $\triangle_k \neq 0$,ce qui nous donne $d_k \neq 0$Ceci implique $\langle Ad_k, d_k \rangle > 0$,car la matrice A est symétrique définie positive ,l'algorithme des gradients conjugués (AGC) pour une fonction quadratique avec une matrice A symétrique définie positive est alors le suivant :*

1. **pas 1**

   On pose k=0,on choisit $u^{(0)} \in \mathbb{R}$ et on pose $d_0 = \nabla f(u^{(0)}) = Au^{(0)} - b$.

2. **pas 2**

   **si** $\nabla f(u^{(k)}) = 0$, STOP "La solution $u^*$est$u^{(k)}$.sinon va au **pas3.**

3. **pas 3**

   on pose

$$\rho_k = \frac{\left\langle \nabla f(u^{(K)}), d_k \right\rangle}{\langle Ad_k, d_k \rangle}$$
$$u^{(k+1)} = Au^{(k)} + \rho_k d_k$$
$$\beta_k = \frac{\left\| \nabla f(u^{(K+1)}) \right\|^2}{\left\| \nabla f(u^{(K)}) \right\|^2}$$
$$d_{k+1} = \nabla f(u^{(K+1)}) + \beta_k d_k$$

   faire $k = k + 1$

   retour au **pas 2.**

les cofficient $\beta_{k+1}$ étant choisis de manéaire que $d^{(k)}$ soit conjugué avec tout les diréctions précedents autrement dit :

$$d_{k+1}^\top A d_k = 0$$

en $u^{(k)}$ et de la dérection précédente $d_{k-1}$ c'est -à -dire :

$$d_{k+1} = -\nabla f(u^{(k+1)}) + \beta_k d_k$$

on déduire que :

$$(-\nabla f(u^{(k+1)}) + \beta_{k+1} d_k)^\top A d_k = 0$$

$$-\nabla^\top f(u^{(k+1)})Ad_k + \beta_{k+1}d_k^\top = 0$$

$$\beta_{k+1} = \frac{\nabla^\top f(u^{(k+1)})Ad_k}{d_k^\top Ad_k}$$

si on note $g_k = \nabla f(u^{(k)})$ donc :

$$\beta_{k+1} = \frac{g_{k+1}^\top Ad_k}{d_k^\top Ad_k}$$

**Défférentes formules de $\beta_k$ dans le cas quadratique**

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

si on note $y_{k-1} = g_k - g_{k-1}$ ,on 'obtient les varientes suivantes :

1. Gradient conjugué vaiente Histenes-Stiefel(HS)

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

2. Gradient conjugué vaiente Fletcher-Reeves(FR)

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

3. Gradient conjugué vaiente Daniel(D)

$$\beta_k^{D} = \frac{g_{k-1}^\top \nabla^2 f(u^{(k)})d_k}{d_k^\top \nabla^2 f(u^{(k)})d_k}$$

4. Gradient conjugué vaiente Polak-Ribiére-Polyak(PRP)

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

5. Gradient conjugué vaiente desent-Fletcher(CD)

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

6. Gradient conjugué vaiente Dai-Yuan

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

**Remarque 1.2.3** *Dans le cas quadratique ,on'a vu que :*

$$\beta_k^{HS} = \beta_k^{PRP} = \beta_k^{FR} = \beta_k^{CD} = \beta_k^{Dy}$$

**Notation 1.2.2** *soit le problème de minimisation suivant*

$$\Phi(x) = \frac{1}{2}x^\top Ax - b^\top x$$

*notons que le gradient de* $\Phi$ *est égale au résidu du systéme linéaire :*

$$\nabla\Phi(x) = Ax - b := r(x)$$

passons maintenant au théorème suivant , qui nous dit que les direction $d_0, ..., d_{n-1}$ sont effectivement $A$-conjugué, Dautre plus , il nous dit que les risidus $r_i$ sont deux à deux orthogonaux et que chaque $d_k$ et chaque risidus $r_i$ sont contenus dans le sous-espace de krylov de dégré $k$ pour $r_0$ , défini par $K(r_0; k) := span\{r_0, Ar_0, ..., A^k r_0\}$

**Théorème 1.2.2** *supposons que le k-iéme itéré, généré par la méthode du gradient conjugué n'est pas le point* $x^*$*, alors les quatre proprietés suivante sont vraies :*

$$r_k^\top r_i = 0, \ \textit{pour } i = 0, ..., k-1$$

$$span\{r_0, r_1, ..., r_k\} = span\{r_0, Ar_0, ..., A^k r_0\}$$

$$span\{d_0, d_1, ..., d_k\} = span\{r_0, Ar_0, ..., A^k r_0\}$$

$$d_k^\top A d_i = 0, \ \ pour\ i = 0, 1, ..., k-1$$

*voila pour quoi la séquence* $\{x_k\}$ *converge vers* $x^*$ *en au plus* $n$ *pas*

**algorithme 1**

$x_o$ soit fixé posons $r_0 \leftarrow Ax_0 - b$ , $d_0 \leftarrow -r_0$ , $k \leftarrow 0$; **while** $r_0 \neq 0$

$$\rho_k \leftarrow \frac{r_k^\top r_k}{d_k^\top A d_k};$$

$$x_{k+1} \leftarrow x_k + \rho_k A d_k;$$

$$r_{k+1} \leftarrow r_k + \rho_k A d_k;$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k};$$

$k \leftarrow k + 1$; **end(while)**

**les avantages de la méthode du gradient conjugué quadratique :**

1. la consomation mèmoire de l'algorithme est minime :on doit stoker les quatre vecteurs $u^{(k)}$, $g_k$ , $d_k$ , $Ad_k$ et les scalaires $\rho_k$ , $\beta_k$ .
2. l'algorithme du gradient conjugué linéaire est surtout utile pour-résoudre des grands systèmes creux, en effet il suffit de savoir appliquer la matrice $A$ à un vecteur.
3. la convergence peut être assez rapide : si $A$ admet seulement $r(r < n)$ valeurs propres distincts la convergence a lieu en au plus $r$ iteration.

### 1.2.2 Cas d'une fonction $f$ quelconque

On suppose ici $f : \mathbb{R} \to \mathbb{R}$ une fonction elliptique. Dans ce cas l'algorithme des gradients conjuguées est le suivant (c'est une "petite" modification de l'algorithme précédent) :

1. **pas 1**

   On pose $k = 0$, on choisit $u^{(0)} \in \mathbb{R}^n$ et on pose $d_0 = \nabla f(u^{(0)})$.

2. **pas 2**

   **si** $\nabla f(u^{(k)}) = 0$, STOP "La solution $u^*$est $u^{(k)}$. sinon va au **pas 3.**

3. **pas 3**

   on pose

$$u^{(k+1)} = u^{(k)} + \rho_k d_k$$

où $\rho_k \in \mathbb{R}$ est l'unique élément qui minimise la fonction

$$\rho \in \mathbb{R} \to f(u^{(k)} + \rho d_k) \in \mathbb{R}$$

   et ensuite

$$\beta_k = \frac{\left\|\nabla f(u^{(K+1)})\right\|^2}{\left\|\nabla f(u^{(K)})\right\|^2}$$
$$d_{k+1} = \nabla f(u^{(K+1)}) + \beta_k d_k$$

   faire $k = k + 1$ retour au **pas 2.**

Ceci est la variante dite de Fletcher-Reeves.

Il y a aussi la variante dite de Polack-Ribiere avec comme seul changement

$$\beta_k = \frac{\left\langle \nabla f(u^{(k+1)}) - \nabla f(u^{(K)}), \nabla f(u^{(K)}) \right\rangle}{\left\|\nabla f(u^{(K)})\right\|^2}$$

Cette dernière variante donne des meilleurs résultats en pratique.

**Remarque 1.2.4** *: Ces deux versions coincident dans le cas d'une fonction $f$ quadratique, car dans ce cas on a*

$$\left\langle \nabla f(u^{(k+1)}), \nabla f(u^{(K)}) \right\rangle = 0$$

# Chapitre 2

# La convergence

## 2.1 La convergence de la méthode du gradient

Nous disons qu'un algorithme itératif est globalement convergent si à partir d'un point initial arbitraire et admissible on génère une suite qui converge vers le point minimal de la fonction objective , si le point initial est suffisament prés de la solution, dans ce cas, nous disons que la convergence est locale.

Dans cette section , nous analysons les propriétés de convergence des méthodes de gradient de descent , en incluant la méthode de gradient à pas fixe.

Nous étudiérons les caréctéristiques importantes de convergence de la méthode de gradient d'un problème quadratique , soit :

tels que $Q = Q^\top > 0$. la solution $x^*$ est obtenue par résolution de système $Qx = b$ .

**Lemme 2.1.1** *l'algorithme iteratif*

$$x_{k+1} = x_k - \alpha_k g_k$$

*avec* $g_k = Qx_k - b$ *qui satisfait*

$v(x_{k+1}) = (1 - \gamma_k)v(x_k)$ *avec : si* $g_k = 0$ *alors* $\gamma_k = 1$ .

*et si* $g_k \neq 0$ *alors :*

$$\gamma_k = \alpha_k \frac{g_k^\top Q g_k}{g_k^\top Q^{-1} g_k} \left( 2 \frac{g_k^\top g_k}{g_k^\top Q g_k} - \alpha_k \right)$$

**Preuve.** la preuve est par calcul direct, notons que si $g_k = 0$, alors le resultat est trivial , le reste de la preuve pour $g_k \neq 0$, nous évaluons d'abord l'expression

$$\frac{v(x_k) - v(x_{k+1})}{v(x_k)}$$

pour faciliter des calcules, soit $y^{(k)} = x_k - x^*$, alors :

$$v(x_{k+1}) = \frac{1}{2} y^{(k)^\top} Q y^{(k)}.$$

par conséquent

$$\begin{aligned} v(x_{k+1}) &= \frac{1}{2}(x_{k+1} - x^*)^\top Q (x_{k+1} - x^*) \\ &= \frac{1}{2}(x_k - x^* - \alpha_k g_k)^\top Q (x_k - x^* - \alpha_k g_k) \\ &= \frac{1}{2} y^{(k)^\top} Q y^{(k)^\top} - \alpha_k g_k^\top Q y^{(k)} + \frac{1}{2} \alpha_k^2 g_k^\top Q g_k \end{aligned}$$

alors

$$\frac{v(x_k) - v(x_{k+1})}{v(x_k)} = \frac{2\alpha_k g_k^\top Q y^{(k)} - \alpha_k^2 g_k^\top Q g_k}{y^{(k)^\top} Q y^{(k)}}$$

puisque :

$$g_k = Q x_k - b = Q x_k - Q x^* = Q y^{(k)}.$$

nous avons :

$$y^{(k)^\top} Q y^{(k)} = g_k^\top Q^{-1} g_k$$

$$g_k^\top Q y^{(k)} = g_k^\top g_k$$

donc :

$$\frac{v(x_k) - v(x_{k+1})}{v(x)} = \alpha_k \frac{g_k^\top Q g_k}{g_k^\top Q^{-1} g^{(k)}} (2 \frac{g_k^\top g_k}{g_k^\top Q g_k} - \alpha_k) = \gamma_k \ .$$

Notons que $\gamma_k \leq 1 \ \ \forall\ k$, car $\gamma_k = 1 - \frac{v(x_{k+1})}{v(x)}$, et $v$ est une fonction non négative .

si $\gamma_k = 1$ pour un certaint $k$ , alors $v(x_{k+1}) = 0$

qui est équivalent à $x_{k+1} = x^*$. Dans ce cas, nous avons également cela pour tous $i \geq k+1$ $x^{(i)} = x^*$et $\gamma_i = 1$. Il s'avère que $\gamma_k = 1$ si seulement si oubien $g^{(k)} = 0$ ou $g^{(k)}$ est un vectur propre de $Q$

Nous sommes maintenant prêtes à énoncer et prouver notre théorème de convergence pour les méthodes de gradient , le théorème donne la condition nécessaire et suffisant pour la suite $\{x_k\}$ produite par la méthode de gradient qui converge vers $x^*$ c'est à dire $x_k \to x^*$, ou $\lim\limits_{k\to\infty} x_k = x^*$. ■

**Théorème 2.1.1** *soit $\{x_k\}$ la suite résultant d'un algorithme de gradient*

$$x^{(k+1)} = x_k - \alpha_k g_k$$

soit $\gamma_k$ définie par le lemme 2.1.1 et supposons que $\gamma_k > 0$ pour tous les $k$ , alors $x^{(k)}$converge vers $x^*$ pour n'importe quel le condition initiale $x^{(0)}$ si seulement si :

$$\sum_{k=0}^{\infty} \gamma_k = \infty$$

**Exemple 2.1.1** *nous montrons ,par un contre exemple , que l'hypothèse $\gamma_k > 0$ dans le théorème 2.1.1 est nécessaire pour la validation du thèorème .*

*En effet, pour chaque $k = 0, 1, 2, ...$choisise de telle maniére que $\gamma_{2k} = \frac{1}{2}$ et $\gamma_{2k+1} = -\frac{1}{2}$ (nous pouvons toujours faire ceci, par exemple $Q = I_n$)*

*du lemme 2.1.1 nous avons :*

$$\begin{aligned} v(x_{k+1}) &= (1 - \frac{1}{2})(1 + \frac{1}{2})v(x_{2k}) \\ &= \frac{3}{4}v(x_{2(k+1)}) \end{aligned}$$

*donc $v(x_{2k+1}) \to 0$ , puisque $v(x_{2k+1}) = \frac{3}{2}v(x_{2k})$ nous avons aussi que*

*$v(x_{2k+1}) \to 0$, alors $v(x_k) \to 0$ qui implique que $x_k \to 0$ (pour tous $x^{(0)}$)*

*d'une part , il est clair que :*

$$\sum_{i=0}^{k} \gamma_i \leq \frac{1}{2}$$

*pour tous $k$ , alors le résultat du thèorème ne tient pas si $\gamma_k \leq 0$ pour certain $k$ .*

nous pouvons maintenant étabir la convergenece dans le cas de l'algorithme de la méthode de descent à pas fixe , nous employons l'inégalité de Rayleigh pour tout $Q = Q^\top > 0$ , nous avons :

$$\lambda_{\min}(Q) \|x\|^2 \leq x^\top Q x \leq \lambda_{\max}(Q) \|x\|^2$$

la où $\lambda_{\min}(Q)$ est la valeur propre minimale de $Q$ et $\lambda_{\max}(Q)$ est la valeur propre maximale de $Q$ , pour $Q = Q^\top > 0$ nous avons aussi :

$$\lambda_{\min}(Q^{-1}) = \frac{1}{\lambda_{\max}(Q)}$$

$$\lambda_{\max}(Q^{-1}) = \frac{1}{\lambda_{\min}(Q)}$$

et

$$\lambda_{\min}(Q^{-1}) \|x\|^2 \leq x^\top Q^{-1} x \leq \lambda_{\max}(Q^{-1}) \|x\|^2$$

**Lemme 2.1.2** *soit $Q = Q^\top > 0$ une matrice ymetrique définie et positive de dimontion $n \times n$ pour n'importe quelle $x \in \mathbb{R}^n$ nous avons :*

$$\frac{\lambda_{\min}(Q)}{\lambda_{\max}(Q)} \leq \frac{(x^\top x)^2}{(x^\top Q x)(x^\top Q^{-1} x)} \leq \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)}$$

**Théorème 2.1.2** *dans l'algorithme de la descent nous avons $x^{(k)} \to x^*$ pour tous $x^{(0)}$.*

**Preuve.** si $g_k = 0$ pour certain $k$ ,puis $x^{(k)} = x^*$ supposons que $g_k \neq 0 \ \forall \ k$ , rappelons

que pour l'algorithme de discente à pas optimal

$$\alpha_k = \frac{g_k^\top g_k}{g_k^\top Q g_k}$$

la subsitution de l'éxpression ci dessus pour $\alpha_k$ dans la formule pour $\gamma_k$ rapporte

$$\gamma_k = \frac{(g_k^\top g_k)^2}{(g_k^\top Q g_k)(g_k^\top Q^{-1} g_k)}$$

Notons dans ce cas $\gamma_k > 0$ pour tous les $k$ , en autre ,par lemme 2.1.2 nous avons $\gamma_k \geq \frac{\lambda_{\min}(Q)}{\lambda_{\max}(Q)} \geq 0$ , donc $\sum_{k=0}^{\infty} \gamma_k = \infty$ (théorème1.2.1) nous conclusons que $x_k \to x^*$. ■
considérerons maintenan la méthode de gradient à pas fixe , ie que $\alpha_k = \alpha \in \mathbb{R}$ ,pour tout $k$ l'algorithme est de la forme :

$$x_{k+1} = x_k - \alpha g_k$$

l'algorithme est d'intérêt pratique en raison de sa simplicité , en particuler , l'algorithme n'exige pas d'une ligne recherche à chaque étape de détermination $\alpha_k$ , parce que la même étape $\alpha$ est empoloyeés à chaque étape , clairement la convergence de l'algorithme dépend du choix de $\alpha$ , et nous n'attendrions pas à ce que l'algorithme fonctionne pour $\alpha$ arbitraire , le théorème suivant donne des conditions nécessaires et suffisants sur $\alpha$ pour la convergence de l'algorithme.

**Théorème 2.1.3** *pour l'algorithme de gradient à pas fixe $x_k \to x^*$ pour tout $x^{(0)}$ si et seulement si :*

$$0 < \alpha < \frac{2}{\lambda_{\max}(Q)}$$

**Exemple 2.1.2** *soit la fonction $f$ est donée par :*

$$f(x) = x^\top \begin{bmatrix} 4 & 2\sqrt{2} \\ 0 & 5 \end{bmatrix} x + x^\top \begin{bmatrix} 3 \\ 6 \end{bmatrix} + 24$$

*l'objet dans ce exemple est de minimiser $f$ par l'utilisation d'algorithme de gradient à*

*pas fixe ,pour aplique le théorème 2.1.3, nous symetrisons d'abord la matrice de la form quadratique de $f$ pour obtenir :*

$$f(x) = \frac{1}{2}x^\top \begin{bmatrix} 8 & 2\sqrt{2} \\ 2\sqrt{2} & 10 \end{bmatrix} x + x^\top \begin{bmatrix} 3 \\ 6 \end{bmatrix} + 24$$

*les valeurs propre de la matrice sont 6 et 12, par le théorème 2.1.3 l'algorithme converges aux miniment pour tout $x^{(0)}$ si seulement si :*

$$0 < \alpha < \frac{1}{2}$$

**Taux de convergence**

nous tournons maintenan à la question des taux de convergent des alagorithme de gradient , en particulier nous concentrons sur l'algorithme de la descent ..., nous presontons d'abord le théorème suivant.

**Théorème 2.1.4** *dans la méthode de descent appliquée à la fonction quadratique à chaque pas $k$ nous avons :*

$$v(x_{k+1}) \leq (\frac{\lambda_{\max}(Q) - \lambda_{\min}(Q)}{\lambda_{\max}(Q)})v(x_k)$$

pour étudier plus loin les propriétiés de convergence d'une suite $\{x_k\}$ nous avons besoin la définition suivant

**Définition 2.1.1** *soit $\{x_k\}$ une suite qui converge aux $x^* c$ à $d$ :*

$$\lim_{k\to\infty} \|x_k - x^*\| = 0$$

*nous disons l'ordre de convergence est $p$ tels que $p \in \mathbb{R}$ si :*

$$0 < \lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} < \infty$$

et nous disons que l'ordre de convergence est $\infty$, si pour tout $p > 0$ :

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} = 0$$

l'ordre de convergence d'une suite est une mesure de son taux de convergence , plus l'ordre est haut , plus la vitesse de la convergence est rapide, l'ordre de convergence s'appelle par fois également le taux de convergence .

si $p = 1$ (convergence de premier ordre ) nous indiquons que la convergence est lineaire .

si $p = 2$(convergence de second ordre) nous disons que la convergence est quadratique.

**Exemple 2.1.3** *1. supposons que* $x_k = \frac{1}{k}$*, et que* $x_k \to 0$ *puis*

$$\frac{|x_{k+1}|}{|x_k|^p} = \frac{1/(k+1)}{1/k^p} = \frac{k^p}{k+1}$$

*si* $p < 1$ *, la suite converge à* $0$*.*

*si* $p > 1$ *, il devient à* $\infty$ *.*

*si* $p = 1$ *, la suite converge à* $1$ *, donc l'ordre de la convergence est* $1$ *(nous avons la convergence linéaire).*

l'ordre de convergence peut être interpréter utilise la notion du l'ordre 0 , se rappeler que $\alpha = 0(h)$ , s'il existe un constant $c$ tes que $|\alpha| < c(h)$ pour suffaisament petit $h$ .

**Théorème 2.1.5** *soit* $\{x_k\}$ *est une suite converge à* $x^*$ *si*

$$\|x_k - x^*\| = 0(\|x_k - x^*\|^p)$$

*l'ordre de convergence (s'il existe ) est mions que* $p$ *.*

par exemple , l'ordre de convergence est mions que 2 si :

$$\|x_{k+1} - x^*\| = 0(\|x_k - x^*\|^2)$$

**Preuve.** soit $s$ l'ordre de convergence de $\{x_k\}$ , supposons

$$\|x_{k+1} - x^*\| = 0(\|x_k - x^*\|^2).$$

puis il existe $c$ tels que pour $k$ suffaisament grand

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} < c \ .$$

donc

$$\begin{aligned}\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^s} &= \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} \|x_k - x^*\|^{p-s} \\ &\leq c \|x_k - x^*\|^{p-s}\end{aligned}$$

par passagr à la limite

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^s} \leq c \lim_{k\to\infty} \|x_k - x^*\|^{p-s}$$

puisque par définition $s$ est l'ordre de convergence

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^s} > 0$$

combinons les deux inégalité ci dessus nous obtenons

$$c \lim_{k\to\infty} \|x_k - x^*\|^{p-s} > 0$$

parce que $\lim_{k\to\infty} \|x_k - x^*\| = 0$ , nous concluons que $s \geq p$ , ie l'ordre de convergence est au moins $p$ ■

L'ordre de la convergence d'aucun suit  convergent ne peut pas être plus petit que 1

nous fournissons un exemple oú l'ordre de convergence d'un algorithme de gradient à pas fixe est plus petit que 1.

**Exemple 2.1.4** *considérons le problème de minimusation de la fonction* $f : \mathbb{R} \to \mathbb{R}$ *définie par* $f(x) = x^2 - \frac{x^3}{3}$
*nous employons l'algorithme*

$$x_{k+1} = x_k - \alpha f'(x^{(k)})$$

*avec* $\alpha = \frac{1}{2}$ *et la condition initiale* $x^{(0)} = 1$
*nous prouvons d'abord que l'algorithme converge à un minumum local de* $f$ *, nous avons* $f'(x) = 2x - x^2$ *,donc l'algorithme de gradient à pas fixe avec* $\alpha = \frac{1}{2}$ *est donée par :*

$$x_{k+1} = x_k - \alpha f'(x) = \frac{1}{2}(x_k)^2$$

*avec* $x_0 = 1$ *, nous pouvons derive l'expression* $x_k = (\frac{1}{2})^{2^k - 1}$ *alors l'algorithme converge vers* 0 *qu'est un minumum locale de* $f$ *, noter que*

$$\frac{|x_{k+1}|}{|x_k|^2} = \frac{1}{2}$$

*donc l'ordre de covergence est* 2

En conclusion, il y a des cas pour les quels l'ordre de convergence de l'algorithme de descent est égale à 1.

### 2.1.1 la convergence de la méthode de gradient conjuguè

grâce à la troisiéme propriété du théoréme 1.2.2 et au deuxième pas de l'algorithme 3, nous avons que

$$\begin{aligned} x_{k+1} &= x_0 + \rho_0 d_0 + ... + \rho_k d_k \\ &= x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + ... + \gamma_k A^k r_0 \end{aligned}$$

Définissons maintenant $P_k^*$ comme étant un polynôme de dégré $k$ avec les cofficients $\gamma_0, \gamma_1, ...\gamma_k$ remplaçons notre polynôme dans l'expressions ci-dessus :

$$x_{k+1} = x_0 + P_k^*(A) r_0$$

Nous allons maintenant que parmi toutes les méthodes possibles dans les $k$ premier pas sont restreints sur le sous espace de krylov $K(r_0; k)$ , l'algorithme 3 donne le meilleure résultat en termes de minimisation de la distance vers la solution , si cette distance est mesurée par $\|.\|_A$ , défini comme

$$\|Z\|_A = Z^\top A Z$$

D'aprés cette définition, on constate que :

$$\frac{1}{2}\|x - x^*\|_A^2 = \frac{1}{2}(x - x^*)^\top A (x - x^*)$$

par conséquence d'aprés la méthode de gradient conjugué $x_{k+1}$ minimise$\|x - x^*\|_A^2$sur l'ensemble $x_0 + span\{d_0, ..., d_k\}$, comme $x_{k+1} = x_0 + P_k^*(A) r_0$ , il en suit que $P_k^*$ est la solution de

$$\min_{P_k} \|x_0 + P_k(A) r_0 - x^*\|_A$$

de plus, nous avons que :

$$\begin{aligned} x_{k+1} - x^* &= x_0 + P_k^*(A)r_0 - x_0 \\ &= \left[I + P_K^*(A)A\right](x_0 - x^*) \end{aligned}$$

soient $0 < \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ les valeurs propres de $A$ et $v_1, ..., v_n$ les vecteurs propres orthonormés correspondants . Nous savons bien que ces vecteurs propres engendrent $\mathbb{R}^n$, donc $x_0 - x^* = \sum_{i=1}^{n} \Psi_i v_i$ et il est aussi facile à voir que les vecteurs propres de $A$ sont aussi des vecteurs propres de $P_k(A)$ pour tout polynôme , Nous pouvons donc écrire que

$$P_k(A)v_i = P_k(\lambda_i)v_i \quad , i = 1, 2, ..., n$$

et donc

$$x_{k+1} - x^* = \sum_{i=1}^{n} \left[1 + \lambda_i P_k^*(\lambda_i)\right] \Psi_i v_i$$

la norme définie ci dessus nous permet de dire que

$$\begin{aligned} \|x_{k+1} - x^*\|_A^2 &= \min_{P_k} \sum_{i=1}^{n} \lambda_i \left[1 + \lambda_i P_k(\lambda_i)\right]^2 \Psi_i^2 \\ &\leq \min_{P_k} \max_{1 \leq i \leq n} \left[1 +_i P_k(\lambda_i)\right]^2 \left( \sum_{j=1}^{n} \lambda_j \Psi_j^2 \right) \\ &= \min_{P_K} \max_{1 \leq i \leq n} \left[1 +_i P_k(\lambda_i)\right]^2 \|x_0 - x^*\|_A^2 \end{aligned}$$

sachant que $\|x_0 - x^*\|_A^2 = \sum_{j=1}^{n} \lambda_j \Psi_j^2$ . cette derniére expression nous permet de quantifier le taux de convergence de la méthode du gradient conjugué.

**Théorème 2.1.6** *si $A$ a seulement $r$ (avec $r < n$) valeurs propres distinctes , alors l'algorithme du gradient conjugué arrive à la soulution en au plus $r$ itrrations .*

**Preuve.** soint $\tau_1 < ... < \tau_r$ les valeurs distinctes des valeures propres $\lambda_1 ... \lambda_n$.

Définissons maintenant $Q_r(\lambda)$ par

$$Q_r(\lambda) = \frac{(-1)^r}{\tau_1\tau_2...\tau_r}(\lambda-\tau_1)...(\lambda-\tau_r).$$

Il est facile de voir que $Q_r(\lambda)-1$ est un polynôme de dégré $r$ avec une racine $\lambda = 0$. Definissons ensuite $\bar{P}_{r-1}$, un polynôme de dégré $r-1$ par

$$\bar{P}_{r-1}(\lambda) = \frac{(Q_r(\lambda)-1)}{\lambda}$$

En posont $r-1$ dons $\min\limits_{P_k}\max\limits_{1<i<n}[1+\lambda_i P_k(\lambda_i)]^2$nous obtenons

$$0 \leq \min_{P_k}\max_{1\leq i\leq n}\left[1+\lambda_i P_{r-1}(\lambda_i)\right]^2 \leq \max_{1\leq i\leq n}\left[1+\lambda_i P_{r-1}(\lambda_i)\right]^2 = \max_{1\leq i\leq n} Q_r(\lambda_i) = 0$$

Donc, pour $k = r-1$ on trouve que $\|x_r - x^*\|_A = 0$ alors $x_r = x^*$. ■

**Théorème 2.1.7** *si $A$ a des valeurs propres $\lambda_1 \leq \lambda_2... \leq \lambda_n$, nous avons que*

$$\|x_{k+1}-x^*\|_A^2 \leq \left(\frac{\lambda_{n-k}-\lambda_1}{\lambda_n+\lambda_1}\right)^2 \|x_0-x^*\|_A^2$$

on donne juste une idée comment ce resultat à été trouvé . On choisit un polynôme $\bar{P}_k$ de dégré $k$ tel que le polynôme $Q_{k+1}(\lambda) = 1+\lambda\bar{P}_k(\lambda)$ a comme racines les plus grandes valeurs propres $\lambda_n, \lambda_{n-1}, ..., \lambda_{n-k+1}$ et le point milieu entr $\lambda_1$et$\lambda_{n-k}$. On peut montre que la valeurs maximale atteinte par $Q_{k+1}$ sur $\lambda_1, ..., \lambda_{n-k+1}$ et $\frac{\lambda_{n-k}-\lambda_1}{\lambda_n+\lambda_1}$.
le théorème nous permet de prédires le comportement de la méthode du gradient conjugué . Supposons par exemple que la valeurs propres de $A$ consistent en $m$ grandes valeurs propres et les $n-m$ valeures prpres restantes sont situées autoure de 1 .
si nous définissons $\epsilon = \lambda_{n-m} - \lambda_1$ le théorème 2.1.7dit qu'aprés $m+1$ pas de la méthode de gradient conjugué , nous avons que

$$\|x_{m+1}-x^*\|_A \approx \|x_0-x^*\|_A$$

pour une petite valeur de $\epsilon$ aprés $m+1$ pas de la méthode va nous donner une bonne approximation de la solution. Mais attention,le théorème 2.1.7 nous donne une borne supéirieure.Il se peut que la méthode nous donne déja des bons résultats aprés les premiérs itérations . Il est généralement vrai que si les valeurs propres apparaissent en $r$ groupes, alors la méthode du gradient conjugué résoud approximativement le probléme aprés $r$ pas. une autre expression de convergence pour la méthode de gradient conjugué est celle basée sur le nombre de condition spectral. celui-ci est défini par

$$K = \|A\|_2 . \left\|A^{-1}\right\|_2 = \frac{\lambda_n}{\lambda_1}$$

On arrive ensuit à montrer que

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

cette expression est plus approximative que celle de l'erreure précedente. Mais en réalité, souvent, on n'a pas beaucoup d'information sur $A$ et ses valeurs propres. cette derniére métthode ne demande que les valeurs propres extrêmes ou des approximation de celles-ci.

**Préconditionnement**

pour accélérer la méthode du gradient conjugué, on peut transformer le systéme lineaire de telle façon à ce que les valeurs propres de la matrice $A$ mieux distribués, le processus présenté par la suite est appellé le préconditionement . Il s'agit de changer la variable $x$ en une variable $\hat{x}$ au moyen d'une matrice $C$ -non -singuliére : $\hat{x} = Cx$. Par conséquent $\phi$ est transformée en

$$\hat{\phi}(\hat{x}) = \frac{1}{2}\hat{x}^\top (C^{-\top} A C^{-1})\hat{x} - (C^{-\top} b)^\top \hat{x} \tag{2.1}$$

En utilisant l'algorithme 5 pour minimiser $\hat{\phi}(\hat{x})$ ou bien pour résoudre le système lineaire

$$(C^{-\top} A C^{-1})\hat{x} = C^{-\top} b$$

le taux de convergence dépond des valeurs propres de la matrice $C^{-\top}AC^{-1}$ et non de celles de matrice $A$ . On pourrait donc par exemple choisir $C$ de telle maniére que le nombre de condition de $A$ .Une autre idée serait de choisire $C$ tel que les valeurs propres de $C^{-\top}AC^{-1}$ sont regroupées en un petit nombre $r$ de groupes. Ainsi on trouve une bonne approximation aprés seulement $r$ iteration. L'algorithme ci-dessous nous n'est rien d'autres qu'une application de l'lgoritme 1 au probléme2.1 par rapport à $\hat{x}$. En suit , il transforme toutes les équations pour les exprimer par rapport à $x$. On remarque que l'algorithme 8 n'utilise pas explicitement $C$,mais plutôt la matrice $M = C\top C$, qui est symétrique et défini positive par construction.

**algorithme 2**

gradient conjugué préconditionné $x_0$ soit fixé et $M$ le préconditioneur de $A_i$ ; Posons $r_0 \leftarrow Ax_0 - b$ ; Résoudre $My_0 = r_0$ par ropport à $y_0$ ;Posons $P_0 \leftarrow -r_0$, $k \leftarrow 0$ ; while $r_0 \neq 0$

$$\rho_k \leftarrow \frac{r_k^\top y_k}{d_k^\top A d_k} ;$$

$$x_{k+1} \leftarrow x_k + \rho_k d_k ;$$

$$r_{k+1} \leftarrow r_k + \rho_k A d_k;$$

$$My_{k+1} \leftarrow r_{k+1};$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^\top y_{k+1}}{r_k^\top r_k};$$

$$d_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} d_k;$$

$$k \leftarrow k + 1;$$

end(while)

si dans l'algorithme 2 nous remplacent $M$ par $I$ on retombe sur l'algorithme 1 les proprietés de l'algorithme 1 se généralisent de façon intéressante, par exemple la premiére affirmation

du théorème 1.2.2 , à savoire $r_k^\top r_i = 0$ pour $i = 0, ..., k-1$ devient

$$r_k^\top M r_j = 0 \quad \forall i \neq j$$

pour l'ordinateur, la grande différence entre l'algorithme 8 et l'algorithme 5 est qu'il faut encore résoudre le systéme $My = r$ .

### 2.1.2 Préconditionneurs pratique

Comme si souvent en numérique, il n'existe pas une meilleure stratégie de préconditionnement valable pour tout type de matrices. Des préconditionneurs généraux ont bien été proposé mais leurs efficacité varie fortement d'un probléme à l'autre. Les stratégie les plus importantes de ce genre sont la SSOR (symétric succisseve overrelaxation), la méthode de cholesky incomplète et les préconditionneurs par bande .

# Chapitre 3

# Application

## 3.1 Le probléme de controle optimale

le but est réduire au minimise le côut fonctionnel

$$J = \varphi\left[x, \pi\right]_{t_r} + \int_{t_0}^{t_r} L(x, u, \pi, t)dt \tag{3.1}$$

avec l'équation diffirentielle :

$$\dot{x} = f(x, u, \pi, t), \qquad x(t_0) = 0 \tag{3.2}$$

contraintes de controle :

$$a_j \leq u_j(t) \leq b_j, \qquad j = 1; 2, ..., r \tag{3.3}$$

et avec les paramètres :

$$c_k \leq \pi_k \leq d_k, \qquad k = 1, 2, ..., q \tag{3.4}$$

$f$ une fonctins vectorielle non linéaire de dimension $n$ ,de direction le vecteur $x$ , $u$ le

vecteur d'entré de controle avec dimension $r$ et $\pi$ est un vecteur de pramètre de dimension $q$, $t_0$ est le temps initial donnè, $t_r$ est le temps final, on suppose qu'il est donné,$\varphi$ et $L$ sont des fonctions scalaires ,on suppose donné une controle $u$,a paramétre $\pi$,3.2 peut être résolu pour un unique $x = (u, \pi)$.Ainsé $J = J(u, \pi)$ est une fonction unique de $u$ et de $\pi$, on suppose que le gradient de $J$ est existe .

Notons $H$ la fonction hamiltonienne donné par :

$$H(x, \lambda, u, \pi, t) = \lambda^\top f(x, u, \pi, t) + L(x, u, \pi, t) \tag{3.5}$$

où $x(t)$ est une fonction de vecteur de dimension $n$, puis les états d'optimalité nécessaires sont pour le cas sans contraintes 3.3 , 3.4 comme suit (voir l'èxemple 3)

$$\dot{\lambda} = -\frac{\partial H}{\partial x} \tag{3.6}$$

$$\lambda(t_f) - \left(\frac{\partial \varphi}{\partial \pi}\right)_{t=t_r} = 0 \tag{3.7}$$

$$\frac{\partial H}{\partial u} = 0 \tag{3.8}$$

$$\left(\frac{\partial \varphi}{\partial \pi}\right)_{t=t_r} + \int_{t_0}^{t_r} \frac{\partial H}{\partial \pi} \partial t = 0 \tag{3.9}$$

supposons que $\frac{\partial H}{\partial u}$ est continu dans $u$ et $\frac{\partial H}{\partial \pi}$ sur $\pi$ .

Toutes les expressions sont évalués vers la solution optimale $x^*$,$u^*$ et $\pi^*$.Tous les vecteurs ,incluant aussi le gradient de diverses fonction, qui sont supposé être des vecteurs colonne. Les variation par gradient de $u$ et $\pi$ peut être comme suit :

$$g(t) = \frac{\partial H}{\partial u} \tag{3.10}$$

et

$$h = \left(\frac{\partial \varphi}{\partial \pi}\right)_{t=t_r} + \int_{t_0}^{t_r} \frac{\partial H}{\partial \pi} \partial t \tag{3.11}$$

## 3.2 l'algorithme de gradient conjugué

la méthode du gradient conjugué calcule la dérection de la recharche en $i$ -itération, la dérection de gradient conjugué de $u$ est déterminée par :

$$p^i(t) = -g^i(t) + \beta^i p^{i-1}(t) \tag{3.12}$$

tel que :

$$\beta^i = \frac{\int_{t_0}^{t_r} g^i(t) g^i(t) \partial t}{\int_{t_0}^{t_r} g^{i-1}(t) g^{i-1}(t) \partial t} \tag{3.13}$$

et

$$g^i(t) = \left.\frac{\partial H}{\partial u}\right|_i \tag{3.14}$$

à condition que $\int_{t_0}^{t_r} g^{i-1}(t) g^{i-1}(t) \partial t \neq 0$ et $\beta^0 = 0$ , la dérection de gradient conjugué de $\pi$ est déterminée par :

$$q^i = -h^i + \gamma^i q^{i-1} \tag{3.15}$$

avec

$$\gamma^i = \frac{h^i h^i}{h^{i-1} h^{i-1}} \tag{3.16}$$

et

$$h^i = \left(\frac{\partial \varphi}{\partial \pi}\right)_{t=t_r}\Bigg|_i + \int_{t_0}^{t_r} \left.\frac{\partial H}{\partial \pi}\right|_i \partial t \tag{3.17}$$

à condition que $h^{i-1}h^{i-1} \neq 0$ et $\gamma^0 = 0$ .

la nouvelle évaluation de $u(t)$ est alors

$$u^{i+1}(t) = u^i(t) + \alpha^i p^i(t) \tag{3.18}$$

et de $\pi$

$$\pi^{i+1} = \pi^i + \alpha^i q^i \tag{3.19}$$

tel que $\alpha^i$ est déterminé par la recherche unidimenssionelle afin de reduire au minimum :

$$J(u^{i+1}, \pi^{i+1}) = \min_{\alpha^i} J(u^i + \alpha^i p^i, \pi^i + \alpha^i q^i) \tag{3.20}$$

## 3.3 Traitement de contraintes

considérons le cas dans laquelle il y a une variable de controle $u(t)$ et un paramètre $\pi$ qui satisfaire la saturation de controle , les condition d'optimalité nécessaires 3.8 et 3.9 sont remplacer par :

$$H(x^*, \lambda, u^*, \pi^*, t) = \min_{u \in U} H(x^*, \lambda, u^*, \pi^*, t) \tag{3.21}$$

$$\left[ \left( \frac{\partial \varphi}{\partial \pi} \right)_{t=t_r} + \int_{t_0}^{t_r} \frac{\partial H}{\partial \pi} dt \right]^\top \delta\pi \geq 0 \tag{3.22}$$

tel que $x^*, u^*, \pi^*$ sont les solutions optimales, $u$ appelé la région admisible de controle voir 3.3 , $\delta\pi$ est n'importe quel changement faisable de paramètre .

$u^{i+1}(t)$ et $\pi^{i+1}$ sont tronqués selon les limites supèreures et infèrieires de la façon suivante :

$$u^{i+1}(t) = \begin{cases} u^i(t) + \alpha^i p^i(t) & \text{si } a \leq u^{i+1}(t) \leq b \\ a & \text{si } u^{i+1}(t) < a \\ b & \text{si } u^{i+1}(t) > b \end{cases} \tag{3.23}$$

$$\pi^{i+1} = \begin{cases} \pi^i + \alpha^i q^i & \text{si } c \leq \pi^{i+1} \leq d \\ c & \text{si } \pi^{i+1} < c \\ d & \text{si } \pi^{i+1} > c \end{cases}$$

## 3.4 Résumé de l'algorithme

**étape 1** : poser $i = 0$ et choisissons les évaluations initiaux $u^0$ $\pi^0$.

**étape 2 :** resoudre les équations d'état 3.2 avec $u = u^i$,$\pi = \pi^i$ et les équations adjoint 3.6 , 3.7 puis calculer $p^i$ et $q^i$ en utilisant 3.12 ,3.15 .

**étape 3 :**

$$u^{i+1}(t) = u^i(t) + \alpha^i p^i(t)$$

et

$$\pi^{i+1} = \pi^i + \alpha^i q^i$$

choisir $\alpha^i$ pour minimise 3.20 .

**étape 4 :**répéter les étapes 2 et 3 jusqu'à

$$\left|J(u^{i+1}, \pi^{i+1}) - J(u^i, \pi i)\right| < \delta \left|J(u^i, \pi^i)\right|$$

tel que $\delta$ est un nombre positif spicifique (ici $\delta = 10^{-4}$).

## 3.5 Exemples

**Exemple 3.5.1** *minimiser le côut :*

$$J = \frac{1}{2}\left[x_1^2\right]_{1.5} + \frac{1}{2}\int_0^{1.5} u^2 dt$$

*avec les contraintes différentiale* $\dot{x}_1 = x_2$ , $x_1(0) = \pi$

$$\dot{x_2} = -x_1 + u + x_2(1 - x_1^2) \quad , x_2(0) = 1$$

*pour obtenir l'état initial fixe comme dans 3.2 ,posons* $y_1 = x_1 - \pi$, $y_2 = x_2$ , *nous obtenons le problème suivant :*

$$J = \frac{1}{2}\left[(y_1 + \pi)^2\right]_{1.5} + \frac{1}{2}\int_0^{1.5} u^2 dt$$

$\dot{y}_1 = y_2$.

$$\dot{y_2} = -(y_1 + \pi) + u + y_2\left[1 - (y_1 + \pi)^2\right]$$

$y_2(0) = 1$.

*les évaluation initiales ont été choisiès :*

$u = 0$, $\pi = 0$. *l'intervalle de l'intégration été divisé en 50 étapes . l'algorithme a convergé en 3 itération au* $J = 0.30323$, $\pi = 1.0392$.

*La solution convergée pour l'exemple est donné dans le tableau suivant :*

*par comparisation de ces resultats nous observons une convergence plus rapide de l'algorithme proposé pour la solution de ce type de probléme.*

*avec controle et paramètre contraintes* $u \geq -0.4$ , $\pi \leq 1$ *été atteient en 3 itérations au* $J = 0.31655$ , $\pi = 1$.*la solution convergée pour cet exemple avec des contraintes est donnée dans le tableau suivent :*

| $t$ | $y_1$ | $y_2$ | $u$ |
|---|---|---|---|
| 0.00 | $0.00000E-01$ | $1.00000E+00$ | $-6.26581E-01$ |
| 0.15 | $1.28989E-01$ | $7.12482E-01$ | $-6.10498E-01$ |
| 0.30 | $2.12591E-01$ | $4.01632E-01$ | $-5.81324E-01$ |
| 0.45 | $2.50107E-01$ | $1.02659E-01$ | $-5.39093E-01$ |
| 0.60 | $2.44980E-01$ | $-1.65173E-01$ | $-4.83889E-01$ |
| 0.75 | $2.02314E-0.1$ | $-3.98320E-01$ | $-4.16408E-01$ |
| 0.90 | $1.26911E-01$ | $-6.03270E-01$ | $-3.38439E-01$ |
| 1.05 | $2.22014E-02$ | $-7.91139E-01$ | $-2.53152E-01$ |
| 1.20 | $-1.10193E-01$ | $-9.74541E-01$ | $-1.64951E-01$ |
| 1.35 | $-2.70563E-01$ | $-1.16624E+00$ | $-7.89282E-02$ |
| 1.50 | $-4.61071E-01$ | $-1.37820E+00$ | $0.00000E+00$ |

TAB. 3.1 – Tableau 1

| $t$ | $y_1$ | $y_2$ | $u$ |
|---|---|---|---|
| 0.00 | $0.00000E+00$ | $1.00000E+00$ | $-4.00000E-01$ |
| 0.15 | $1.32744E-01$ | $7.61194E-01$ | $-4.00000E-01$ |
| 0.30 | $2.26583E-01$ | $4.86913E-01$ | $-4.00000E-01$ |
| 0.45 | $2.78648E-01$ | $2.08859E-01$ | $-4.00000-01$ |
| 0.60 | $2.90137E-01$ | $-5.17519E-02$ | $-4.00000E-01$ |
| 0.75 | $2.64383E-01$ | $-2.87520E-01$ | $-4.00000E-01$ |
| 0.90 | $2.04958E-01$ | $-5.01941E-01$ | $-4.00000E-01$ |
| 1.05 | $1.14440E-01$ | $-7.02937E-01$ | $-3.46228E-01$ |
| 1.20 | $-5.43584E-03$ | $-8.94921E-01$ | $-2.62924E-01$ |
| 1.35 | $-1.54115E-01$ | $-1.08830E-01$ | $-1.38924E-01$ |
| 1.50 | $-3.32345E-01$ | $-1.29047+00$ | $0.00000E+00$ |

TAB. 3.2 – Tableau 2

| $t$ | $x$ | $u$ |
|---|---|---|
| 0.0 | $0.00000E+00$ | $8.29442E-01$ |
| 0.1 | $8.922603E-02$ | $8.06543E-01$ |
| 0.2 | $1.92482E-01$ | $7.79291E-01$ |
| 0.3 | $3.12204E-01$ | $7.46683E-01$ |
| 0.4 | $4.51148E-01$ | $7.07574E-01$ |
| 0.5 | $6.11884E-01$ | $6.60717E-01$ |
| 0.6 | $7.96141E-01$ | $6.05053E-01$ |
| 0.7 | $1.00365E-00$ | $5.40226E-01$ |
| 0.8 | $1.23073E-00$ | $4.67572E-01$ |
| 0.9 | $1.46941E-00$ | $3.91381E-01$ |
| 1 | $1.70911E-00$ | $3.19592E-01$ |

TAB. 3.3 – Tableau 3

**Exemple 3.5.2** *minimisation du côut fonctionnel*

$$J = \int_0^1 (x+\pi)^2 u^2 dt - 2\ln(x(1)+\pi)$$

*avec*

$$\dot{x} = (x+\pi)^2 + u^2 \ , \ \ x(0) = 0$$

*et paramètre contraint* $\pi \leq 1$.

*Les évaluation initiales ont été choisies* $u = 1$, $\pi = 0$ *et l'intervalle de l'intégration a été divisé en 100 pas . L'algorithme a convergé en 6 itération avec* $J = -1.0033$ *et* $\pi = 1$, *la solution de l'exemple est donné dans le tableau 3 :*

la resultat est dans concordance trés bonne avec la solution analytique $J = -1$, $\pi = 1$, $x = e^t - 1$ et $u = e^{-t}$.

## 3.6 Application du gradient conjugué a la résolution d'un système linéaire :$Ax = b$

**objectifs**

On note $M_n(\mathbb{R})$ l'ensemble des matrices carrées d'ordre $n$. Soit $A \in M_n(\mathbb{R})$ une matrice inversible et $b \in \mathbb{R}^n$, ona comme objectife de résoudre le système linéaire $Ax = b$, c'est à dire de trouver $x \in \mathbb{R}^n$ solution de :

$$Ax = b \tag{3.24}$$

Comme $A$ est inversible, il existe unique vecteur $x \in \mathbb{R}^n$ solution de 3.24.

,Nous utilisons la méthode du gradient conjugué pour calcul de ce vecture $x$.

**Exemple 3.6.1** *résolution d'un système linéaire $Ax = b$, dans ce exemple on multiplyant un matrix $A \in M_n(\mathbb{R})$ par un vecture $\grave{x} \in \mathbb{R}^n$ pour trouve un vecture $b \in \mathbb{R}^n$ ensuite on construire un système linéaire $Ax = b$,l'objectif est de trouvé $x^*$ la solution de ce système avec la méthode du gradient conjugué ensuite comparé entre ce dernier et $\grave{x}$ avec $n = 100$*

.

$$A = \begin{bmatrix} 0.81 & 0.16 & ... & 0.94 \\ 0.90 & 0.79 & ... & 0.66 \\ \vdots & \vdots & \vdots & \vdots \\ 0.33 & 0.79 & ... & 0.46 \end{bmatrix}, b = \begin{bmatrix} 26.90 \\ 25.58 \\ \vdots \\ 24.36 \end{bmatrix}$$

*pour résoudre ce système on applique le programme de resolution d'un système linéaire par la méthode du gradient conjugué [cgs] voir Annex.*

*on va donné la valeur de résidu tol=0.000001*

| $\grave{x}$ | $x^*$ |
|---|---|
| 0.1538 | 0.1538 |
| 0.9618 | 0.9618 |
| $\vdots$ | $\vdots$ |
| 0.2088 | 0.2088 |

## 3.7 Application du gradient conjugué au traitement d'image

**Définition 3.7.1** *La technique de traitement d'image concerne la manipulation et l'analyse d'image, elle est divisée en trois parties : restauration, numérisation et compression, description et reconnaissance de formes. Pour la restauration d'image bruitée nous utiliserons la méthode du gradient conjugué pour trouver itérativement l'image originale.*

**Problème**

supposons qu'on a une image originales filtrée, par une matrice $A$ .

comment fait-ont pour trouver l'image originale ?

La solution de ce problème est basée sur la résolution d'un système $AX = B$ tel que :

– $A$ : la matrice filtre (BR bruit)

– $X$ : l'image originale à débruiter.

– $B$ : l'image bruitée

l'objectif est de trouver la matrice $X = A^{-1}B$ , mais le problème reside dans le calul de l'inverse de $A$ car elle est de grande taille. :

1. problème du calcul d'inverse
2. la méthode de gradient conjugué est appliquée sur des vecteurs $\min \|AX - B\|^2 = f(x) \Longleftrightarrow \nabla f(x) = 0$. or $X$ est une matrice

donc pour appliqué la méthode de gradient conjugué nous tronsformons l'équation matricielle $AX = B$ en une équation vectorielle $Ax = b$.

c'est à dire, transformer une image matricielle à une image vectorielle ($x = (X :)$)

pour redimensionner l'équation on utilise le produit de **Kronecker.**

Alors le problème est de trouver l'image vectorielle bruitée (voir Annex)

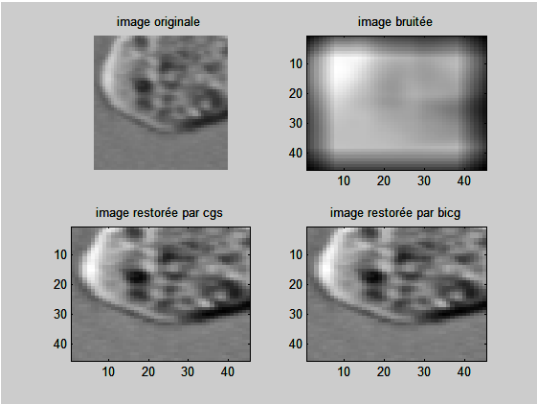
image originale
image bruitée
image restorée par cgs
image restorée par bicg
10
20
30
40

# Conclusion

Ce travail nous a permis d'éclaircir l'utilité et l'importance des methodes du gradient et particuierment les methodes du gradient conjuguées, nous avons montré que les méthodes utilisées sont toutes convergentes pour des critères simples à savoir par exemples les valeurs propres des matrices où la convexité des fonctions critères. Les applications sont nombreu, nous avons pris des exemples qui montrent la limite de chaque méthode, pour le traitement d'image, nous nons somme limités à des images à faible pixels à cause des grands calculs que notre micro ne supporte pas.

Comme conclusion, la métode du gradient conjugué est l'une des bases des méthode de calcul numerique en traitement d'images, d'autres méthode sont plus prefarmantes que celles exposeés ici.

# Bibliographie

[1] Danial kouth , (2009). Optimisation numérique (chapitr 5).University de Friboug.

[2] Edwing . P. Chong, Stanislaw.H .ŻAK. An Introduction to optimization , Second Édition. copy right (2001).

[3] Jorge Nocedal, stephen.J. Numerical Optimization . Editors : Peter Glynn, Stephn M.Robinson

[4] Mohammed Belloufi,(2013/2014).Dévllopement récents de la métode du gradient conjugué.Thèse doctorat en MathématiquesUniversité Badji Mokhtar Annaba

[5] Laghbeche Hadjer, Lahreche Messaouda, (2008).Optimisation sans contrainte, Mémoire de fin d'etudes pour l'obtention du diplôme de license Université M.K Biskra

# Annexe A : Logiciel *R*

```
function x = conjgrad(A,b,tol)
% CONJGRAD Conjugate Gradient Method.
% X = CONJGRAD(A,B) attemps to solve the system of linear equations A*X=B
% for X. The N-by-N coefficient matrix A must be symmetric and the right
% hand side column vector B must have length N.
%
% X = CONJGRAD(A,B,TOL) specifies the tolerance of the method. The
% default is 1e-10.%
% By Yi Cao at Cranfield University, 18 December 2008
% Updated on 6 Feb 2014.
%
if nargin<3tol=1e-10;
end
x = b;
r = b - A*x;
if norm(r) < tol
return
end
y = -r;
z = A*y;
```

```
s = y'*z;
t = (r'*y)/s;
x = x + t*y;
for k = 1 :numel(b);
r = r - t*z;
if( norm(r) < tol )
return;
end
B = (r'*z)/s;
y = -r + B*y;
z = A*y;
s = y'*z;
t = (r'*y)/s;
x = x + t*y;
end
end
```

command :

```
% Example (highlight lines between %{ and %}, then press F9) :
%{
n = 60;
m = 80;
A = randn(n,m);
A = A * A';
b = randn(n,1);
x = conjgrad(A,b);
norm(A*x-b);
%}
```

C = rand(4,4) ;image(C,'CDataMapping','scaled')

axis image

# Annexe B : Abréviations et Notations