وزارة التعليم العالي والبحث العلمي

Université Badji Mokhtar
Annaba

Badji Mokhtar University
Annaba

جامعة باجي مختار

عنابة

Faculté des Sciences
Département de Mathématiques.

THESE

Présentée en vue de l'obtention du diplôme de
Doctorat en Mathématiques
Option : EDP

APPLICATION DE LA TECHNIQUE DE SYMETRIE DE POWELL AUX METHODES DU GRADIENT CONJUGUE AVEC LA GENERALISATION DE CONDITION DE CONJUGAISON

Par:
BENRABIA NOUREDDINE

Sous la direction de

PROF : YAMINA LASKRI

Devant le jury

PRESIDENT :	Rebbani faouzia	Prof	EPST Annaba
EXAMINATRICE :	Diaba Fatma	Prof	Université Annaba
EXAMINATRICE :	Amiar Rachida	MCA	Université Annaba
EXAMINATEUR :	Nisse Lamine	Prof	Université Annaba
EXAMINATEUR :	Bousstila nadjib	Prof	Université Guelma

Année : 2015/2016

Table des matières

Remerciements

Tout d'abord, mes remerciements s'adressent a la personne qui m'a proposé le sujet de thèse et qui m'a encadré tout au long de ces années d'étude (2012/2016) le professeur Laskri yamina.

Je tiens à la remercier pour son enthousiasme, disponibilité et son conseils. Je garderai en mémoire la confiance et la sympathie qu'elle m'a témoignée au cours de ces trois années de thèse.

Je tiens également à exprimer mes remerciements aux membres du jury, qui ont accepté d'évaluer mon travail de thèse.

Merci à Mme Rebbani faouzia, professeur à l'EPST annaba d'avoir accepté de présider le jury de cette thèse.

Merci aussi à les professeurs Diaba Fatma de l'Université annaba , Bousstila nadjib de l'Université de 8 mai 45 Guelma et Nisse Lamine de l'Université annaba d'avoir accepté d'être les rapporteurs de ce manuscrit.

Merci également à le docteur Amiar Rachida de l'Université annaba pour avoir accepté de faire partie de mon jury de thèse.

Je remercie le professeur Al-baali de m'avoir accueilli dans son laboratoire où j'ai pu réaliser mes travaux de recherche dans des conditions excellentes.

Finalement, j'adresse un grand merci à toute ma famille qui a toujours été présente lorsque j'en ai eu besoin, en particulier merci à ma femme.

Résumé

A travers cette thèse on se propose d'exposer la nouvelle méthode du gradient conjugué en appliquant la technique de symétrie de Powell et en utilisant la recherche linéaire de Wolfe.

La convergence globale de la méthode est ainsi prise en compte en utilisant l'analyse spectrale de la matrice d'itération de gradient conjugué et la condition de Zoutendjik.

Lorsqu'on se base sur ces derniers, un algorithme de descente puissant est développé.

Les résultats numériques montrent bien que cet algorithme est concurrentiel en comparaison avec les algorithmes de PRP^{+}(Powell [1984]) et FR. En conclusion un bref commentaire concernant la nouvelle méthode est donné, en indiquant la direction d'éventuelles futures recherches.

Abstract

A new conjugate gradient method is proposed by applying Powell's symmetrical technique to conjugate gradient methods in this paper. Using Wolfe line searches, the global convergence of the method is analyzed by using the spectral analysis of the conjugate gradient iteration matrix and Zoutendijk's condition. Base on this, one concrete descent algorithms is developed. 86s numerical experiments are presented to verify his performance and the numerical results show that the algorithm is competitive compared with the PRP^{+}(Powell [1984])and FR algorithms. Finally, a brief discussion of the new proposed method is given.

الملخص

نتناول في هاته الرساله طريقه التدرج المترافق الطيفي الجديدة من خلال تطبيق تقنيه متناظرة باول مع البحث الخطي التقريبي لوولف ، وذلك من أجل إيجاد القيمه الأصغر لداله بعدة متغيرات. وقد عرضنا التقارب الاجمالي لهذه الطريقه باستخدام التحليل الطيفي لمصفوفه التكرار للتدرج المترافق الطيفي وحاله زوتنديجك. على هذا الأساس، وضعت خوارزميه ملموسه . واستخدمت ٨٦ تجربه للتحقق من أدائها والنتائج العدديه تشير إلى أن الخوارزميه تنافسيه مقارنه مع خوارزميات (باول ﴿١٩٨٤﴾) وفليتشر ريفز. وأخيرا، مناقشه موجزة عن طريقه المقترح الجديد نبين فيها الخطوط المستقبليه للبحث.

Les mots-clés

- Recherche linéaire.
- Condition de Zoutendjik.
- Méthodes du gradient conjugué.
- Condition de conjugaison.
- Matrice d'itération du gradient conjugué.
- Technique de symétrie.
- Analyse spectrale.
- Profile de performance .
- Convergence globale.

Introduction

Optimiser : rendre optimal, donner à quelque chose les meilleures conditions d'utilisation, de fonctionnement ou de rendement au regard de certaines circonstances. (Déf. du LAROUSSE).

Les gens optimisent :
Des compagnies aériennes établissent les horaires des équipages et des avions afin de minimiser le coût. Des investisseurs cherchent a constituer des portefeuilles boursiers qui évitent des risques excessifs tout en donnant un bon rendement. Des fabricants visent le maximum d'efficacité dans la conception et l'opération de leurs processus de production.
La nature optimise :
Des systèmes physiques tendent vers un état d'énergie minimale. Des molécules dans un système chimique réagissent entre eux jusqu'à ce que l'énergie potentielle totale de leurs électrons soit minimisée. Des rayons de lumière suivent des trajectoires qui minimisent leur temps de trajet. L'optimisation est un outil important dans la théorie des décisions et dans l'analyse des systèmes physiques. Afin de l'utiliser, nous devons d'abord identifier un objectif. Ceci est une mesure quantitative de la performance du système. Cet objectif pourrait être : bénéfice, temps, énergie potentielle, ou toute autre quantité ou combinaison des quantités qui peut être représentée par un seul nombre. L'objectif dépend de certaines caractéristiques du système, appelées des variables ou des inconnues. Notre but est de trouver des valeurs de variables qui optimisent l'objectif. Souvent les variables sont restreintes, ou contraintes, d'une certaine manière. Par exemple, la densité ou l'énergie ou le taux d'intérêt ne peuvent pas être négatifs. Le processus de l'identification de l'objectif, variables et contraintes pour un problème donné est appelé la modélisation. La construction d'un modèle approprié est la première étape. Parfois c'est l'étape la plus importante du processus d'optimisation. Si le modèle est simpliste, il ne donnera pas d'informations utiles concernant le problème pratique. Par contre, si le modèle est trop complexe, il pourra devenir trop difficile à résoudre. Une fois que le modèle a été formulé, un algorithme d'optimisation peut être utilisé afin de trouver sa solution. Ceci est fait, normalement, par un ordinateur. Il

n'existe pas un algorithme universel d'optimisation. Plutôt, il existe de nombreux algorithmes, chacun étant adapté à un type particulier de problème d'optimisation. Il est souvent de la responsabilité de l'utilisateur/programmeur/modeleur de choisir un algorithme approprié pour l'application spécifique. Ce choix est très important puisqu'il déterminera si le problème est résolu rapidement ou pas et effectivement, même si la solution est dèjà trouvée. Après l'application d'un algorithme d'optimisation au problème, on doit être capable de reconnaître si l'on a réussi à trouver une solution. Dans certains cas, il existe des expressions mathématiques, les conditions d'optimalité, pour vérifier si l'ensemble des variables obtenu est bien la solution du problème. Si les conditions d'optimalité ne sont pas vérifiées, ils peuvent donner des indications sur la manière d'améliorer notre estimation actuelle. Finalement, le modèle peut être amélioré par l'application de techniques comme l'analyse de sensibilité qui révèle la sensibilité de la solution aux changements dans le modèle et dans les données.
En résumé, voici les étapes principales :

- $\triangle$ identifier un objectif et des variables ;
- $\triangle$ construire un modèle approprié ;
- $\triangle$ choisir un algorithme d'optimisation ;
- $\triangle$ vérifier les conditions d'optimalité ;
- $\triangle$ faire éventuellement une analyse de sensibilité.

Nous divisons les problèmes d'optimisation en plusieurs familles et catégories (voir la Figure 1)

- $\triangledown$ optimisation continue ;
- $\triangledown$ optimisation discrète ;
- $\triangledown$ optimisation stochastique.

Parmi les problèmes pratiques en optimisation continue, nous pouvons citer les suivants :

- $\triangleleft$ trouver le trajectoire optimal pour un avion, une navette spatiale ou le bras d'un robot ;
- $\triangleleft$ identifier les propriétés sismiques de l'écorce terrestre en ajustant un modèle à un ensemble d'enregistrements pris sur le terrain (en fait, un problème inverse) ;

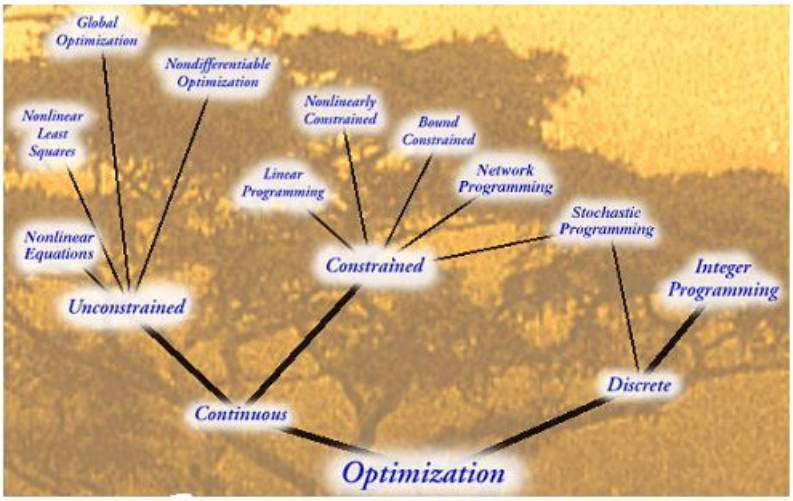

FIGURE 0.1: L'arbre d'optimisation

◁ composer une portefeuille d'investissements afin de maximiser le retour tout en gardant un niveau de risque acceptable ;

◁ contrôler un processus chimique ou un dispositif mécanique afin d'optimiser la performance ;

◁ calculer la forme optimale d'une voiture, un voilier ou un avion.

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

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

-Taylor, Newton , Lagrange, qui ont élaboré les bases des développements limités.

- Cauchy fut le premier à mettre en œuvre une méthode d'optimisation, méthode du pas de descente, pour la résolution de problèmes sans contraintes.

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

Parmi les plus anciennes méthodes utilisées pour résoudre cet problème du type , on peut citer la méthode du Gradient conjugué. Cette méthode est surtout

utilisée pour les problèmes de grande taille. Cette méthode a été découverte en 1952 par Hestenes et Steifel , pour la minimisation de fonctions quadratiques strictement convexes.

Plusieurs mathématiciens ont étendu cette méthode pour le cas non linéaire. Ceci a été réalisé pour la première fois, en 1964 par Fletcher et Reeves (méthode de Fletcher-Reeves) puis en 1969 par Polak, Ribière et Polyak (méthode de Polak-Ribière-Polyak).

Une autre variante a été étudiée en 1987 par Fletcher (Méthode de la descente conjuguée).

Récemment LIU, D., XU, G(2009) ont proposé une nouvelle méthode du gradient conjugué en appliquant la technique de symétrie de Powell et en utilisant la recherche linéaire de Wolfe. La convergence globale de la méthode est ainsi prise en compte en utilisant la condition de conjugaison :

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

l'analyse spectrale de la matrice d'itération de gradient conjugué et de la condition de Zoutendjik.

La question naturelle qui se pose est la suivante :

Peut on obtenir des résultats de cette méthode en utilisant la nouvelle condition de conjugaison comme suit :

$$d_{k+1}^T y_k = -\sigma g_{k+1}^T s_k,$$

là où σ est un scalaire positif ?

Ce qui est le but de cette thèse.

Plan de thèse

Cette thèse comporte quatre chapitres.

Dans le **chapitre premier (01)**, on rappelle les concepts mathématiques à appliquer au domaine d'optimisation.

Dans le **chapitre deux (02)**, on aborde les problèmes de minimisation sans contraintes et leurs algorithmes à travers notamment l'aspect général, les conditions d'optimalité , les méthodes à directions de descente et la recherche linéaire.

A travers le **chapitre trois (03)**, sont donnés les principes liés aux différentes méthodes itératives d'optimisation sans contraintes avec les exemples requis pour la démonstration. Ainsi on peut citer les méthodes du gradient conjugué linéaire et non linéaire.

Le **chapitre quatre (04)** est au cœur de cette thèse car il traite essentiellement de l'application de la technique de symétrie de Powell à la méthode du gradient conjugué. On y trouvera ladite application suivie de l'algorithme qui en est déduit, autrement dit, l'algorithme GDSHS1(descente symétrique de Hestenes-Stiefel) . Ce dernier a été soumis aux expériences numériques adéquates. En fin de ce dernier chapitre, une discussion finale est présentée.

Chapitre 1

Optimisation sans contraintes

1.1 Rappels de pré-requis Mathématiques

Nous allons ici rappeler brièvement les définitions de quelques notions importantes auxquelles nous ferons appel par la suite, ainsi que quelques propriétés. Pour d'évidentes raisons de place, cette thèse ne peut prétendre présenter les démonstrations de tous les théorèmes mentionnés.

1.1.1 Espace normé

L'espace euclidien $\mathbb{R}^n$

Soit $\mathbb{R}^n$ L'espace vectoriel réel de dimension $n \in \mathbb{N}$. On notera sa base canonique $e_1, ..., e_n$. On munit $\mathbb{R}^n$ du produit scalaire usuel < . , . > et de la norme associée $\|.\|_2$ (ou $\|.\|$ lorsqu'il n'y a pas d'ambiguïté). C'est 'a dire que si dans la base canonique de $\mathbb{R}^n$ les vecteurs $U, V \in \mathbb{R}^n$ s'écrivent $U = (u_1, u_2, ..., u_n)$ et $V = (v_1, v_2, ..., v_n)$, alors :

$$< U, V >= U^T V = u_1 v_1 + u_2 v_2 + ... + u_n v_n$$

$$\|U\| =< U, U >^{1/2}= \sqrt{u_1^2 + u_2^2 + ... + u_n^2}$$

On parle alors de l'espace euclidien $(\mathbb{R}^n, < ., . >)$ de dimension n. On y vérifie l'inégalité de Cauchy-Schwartz :

$$|\langle U, V \rangle| \leq \|U\| \|V\|$$

Normes de $\mathbb{R}^n$

Une norme $\|.\|$ sur $\mathbb{R}^n$ est une application de $\mathbb{R}^n$ dans $\mathbb{R}^+$ vérifiant :

$$\forall X, Y \in \mathbb{R}^n, \forall \lambda \in \mathbb{R} :$$

(séparation) $\quad \|X\| = 0 \Rightarrow X = 0,$

(homogénéité) $\quad \|\lambda X\| = |\lambda| \|X\|,$

(sous-additivité) $\quad \|X + Y\| \leq \|X\| + \|Y\|.$

Lorsque l'on munit $\mathbb{R}^n$ d'une norme $\|.\|$, on parle de l'espace normé $(\mathbb{R}^n, \|.\|)$. Voici plusieurs exemples de normes sur $\mathbb{R}^n$:

- La norme 1 : $\|X\|_1 = \sum_{i=1}^n |x_i|$,

- La norme 2 : $\|X\|_2 = (\sum_{i=1}^n x_i^2)^{\frac{1}{2}}$,

- La norme p : $\|X\|_p = (\sum_{i=1}^n x_i^p)^{\frac{1}{p}}$,

- La norme ∞ : $\|X\|_\infty = max\{|x_1|, ...|x_n|\}$.

Sur $\mathbb{R}^n$ toutes les normes sont équivalentes, c'est 'a dire si $\|.\|_a$ et $\|.\|_b$ désignent deux normes de $\mathbb{R}^n$, il existe c_1 , $c_2 > 0$ tels que $\forall X \in \mathbb{R}^n$:

$$\|.\|_a \leq c_1 \|.\|_b \qquad \text{et} \qquad \|.\|_b \leq c_2 \|.\|_a.$$

L'espace normé$(\mathbb{R}^n, \|.\|)$ est un espace complet : toute suite de Cauchy y est convergente.

Normes matricielles

Definition 1 *Soit $\|.\|$ une norme quelconque sur $\mathbb{R}^n$, on appelle norme matricielle subordonnée de la matrice carrée A d'ordre n*

$$\|A\| = \max_{X\in\mathbb{R}^n, \|X\|\neq 0} \frac{\|AX\|}{\|X\|} = \max_{X\in\mathbb{R}^n, \|X\|=1} \|AX\|.$$

Proposition 2 *(a) Une norme matricielle subordonnée vérifie les propriétés suivantes :*

$$\|A\| = 0 \Leftrightarrow A = 0 \qquad \textit{pour tout } \boldsymbol{A},$$

$$\|\alpha A\| = |\alpha|\|A\| \qquad \textit{pour tout } \boldsymbol{A}\textit{ , } \alpha \in \mathbb{R}\textit{ ,}$$

$$\|A + B\| \leq \|A\| + \|B\| \qquad \textit{pour tout } \boldsymbol{A}\textit{ ,}\boldsymbol{B}\textit{ ,}$$

$$\|AB\| \leq \|A\|\|B\| \qquad \textit{pour tout } \boldsymbol{A}\textit{ ,}\boldsymbol{B}\textit{ .}$$

(b) Soit $A = (a_{ij})$ une matrice carrée réelle, pour les normes vectorielles usuelles on a les normes matricielles subordonnée :

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{j=n} |a_{ij}|,$$

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{i=n} |a_{ij}|,$$

$$\|A\|_2 = \sqrt{\rho\left(A^T A\right)} = \|A^T\|_2,$$

ou

$$\rho\left(A^T A\right)$$

est le rayon spectral de la matrice symétrique $A^T A$ *.*

(c) Si O et P sont des matrices orthogonales(i.e.$O^T = O^{-1}$ *et* $P^T = P^{-1}$ *), alors :*

$$\|OAP\|_2 = \|A\|_2.$$

Si $AA^T = A^T A$ *(i.e. A est normale), alors :*

$$\|A\|_2 = \rho\left(A\right).$$

Definition 3 *Soit* S *un sous-espace vectoriel de* $\mathbb{R}^n$ *. L'orthogonal de* S *est* $S^{\perp}$*défini par :*

$$S^{\perp} = \left\{Y : X^T Y = 0\right\}, \quad \forall X \in S.$$

1.1.2 Valeurs propres et vecteurs propres

Definition 4 *Le nombre complexe* λ *est valeur propre de la matrice carrée* A *si la matrice* $(A - \lambda I)$ *est singulière, autrement dit s'il existe un vecteur* X *non nul tel que :*

$$AX = \lambda X$$

Proposition 5 *Toute matrice carrée* A *d'ordre* n *a* n *valeurs propres complexes, qui sont les racines du polynome caractèristique* $\det(A - \lambda I)$*. On a :*

$$\det(A) = \lambda_1 \cdot \lambda_2 \cdot ... \cdot \lambda_n,$$

et

$$tr(A) = \lambda_1 + \lambda_2 + ... + \lambda_n.$$

Definition 6 • *La multiplicité algébrique d'une valeur propre est sa multiplicité dans le polynome caractéristique.*

• *Une valeur propre est simple si sa multiplicité algébrique est 1. Sinon, c'est une valeur propre multiple.*

• *La multiplicité géométrique d'une valeur propre est la dimension du sous-espace vectoriel engendré par les vecteurs propres associés.*

• *Si la multiplicité géométrique est strictement inférieure à la multiplicité algébrique, la valeur propre est défective.*

• *Toute valeur propre simple est non défective.*

Definition 7 *Une matrice est diagonalisable si et seulement si toutes ses valeurs propres sont non défectives. De manière équivalente, une matrice est diagonalisable si et seulement s'il existe une matrice inversible complexe V et une matrice diagonale complexe $D = diag(\lambda_i)$ telle que :*

$$AV = VD, \quad A = VDV^{-1}.$$

V est la base des vecteurs propres et les éléments de D sont les valeurs propres.

1.1.3 Matrices symétriques

Definition 8 *La transposée d'une matrice $A \in M_{m,n}(\mathbb{k})$ est la matrice notée $A^T \in M_{n,m}(\mathbb{k})$ obtenue en inter changeant les lignes et les colonnes. Soit $A = (a_{ij})$ et $B = A^T = (b_{ij})$, on a donc $b_{ij} = a_{ji}$.*

Definition 9 *Une matrice carrée A est symétrique si :*
$A = A^T$

Proposition 10 *On suppose ici que $\mathbb{k}$ est un anneau commutatif.*
L'application « transposition » est linéaire :

$$(A + B)^\mathsf{T} = A^\mathsf{T} + B^\mathsf{T}, \quad (\alpha \cdot A)^\mathsf{T} = \alpha \cdot A^\mathsf{T}.$$

La transposée de A^T est A. Autrement dit, l'application « transposition »

$$.^T : M_{n,n}(\mathbb{k}) \to M_{n,n}(\mathbb{k})$$

est une involution. Elle est par conséquent (non seulement linéaire, mais aussi) bijective. C'est donc un isomorphisme d'espaces vectoriels. La transposée du produit de deux matrices est égale au produit des transposées de ces deux matrices, mais dans l'ordre inverse :

$$(AB)^T = B^T \cdot A^T.$$

En effet,

$$[(AB)^T]_{i,j} = (AB)_{j,i} = \sum_k A_{j,k}B_{k,i} = \sum_k B_{k,i}A_{j,k} = \sum_k (B^T)_{i,k}(A^T)_{k,j} = [(B^T)(A^T)]_{i,j}.$$

Si une matrice carrée A est inversible, alors sa transposée l'est aussi, et la transposée de l'inverse de A est égale à l'inverse de sa transposée :

$$\left(A^{-1}\right)^T = \left(A^T\right)^{-1}.$$

Si A désigne une matrice carrée de taille n et B sa transposée, alors A et B ont la même diagonale principale (et par conséquent la même trace) :

$$\forall i \in \{1, \ldots, n\}, \qquad b_{i,i} = a_{i,i}.$$

En particulier, toute matrice diagonale est symétrique, c'est-à-dire égale à sa transposée. Plus généralement, deux matrices carrées transposées l'une de l'autre ont même polynôme caractéristique donc mêmes valeurs propres, comptées avec leurs multiplicités (en particulier, non seulement même trace mais aussi même déterminant), et même polynôme minimal. Mieux : sur un corps, elles sont semblables. Cela se montre en remarquant qu'elles ont les mêmes invariants de similitude.

Theorem 11 *Toute matrice symétrique a des valeurs propres réelles et des vecteurs propres réels. Elle est diagonalisable. De plus, les vecteurs propres forment une base orthonormée*

Matrices semblables

En mathématiques, deux matrices carrées A et B sont dites semblables s'il existe une matrice inversible P telle que :

$$A = PBP^{-1}.$$

La similarité est une relation d'équivalence.

Deux matrices sont semblables si et seulement si elles constituent deux matrices représentatives du même endomorphisme dans deux bases (éventuellement) différentes. Il ne faut pas confondre la notion de matrices semblables avec celle de matrices équivalentes. En revanche, si deux matrices sont semblables, alors elles sont équivalentes. Un moyen de déterminer si deux matrices sont semblables est de les réduire, c'est-à-dire de les ramener à une forme type : diagonale, forme réduite de Jordan.

Remark 12 *Les applications de l'espace des matrices carrées dont le résultat est identique pour une matrice et une matrice qui lui est semblable sont appelés des invariants de similitude.*

Par exemple, la trace d'une matrice en est un, puisqu'avec les notations précédentes :

$$\text{tr } B = \text{tr } \big((P^{-1}A)P\big) = \text{tr } \big(P(P^{-1}A)\big) = \text{tr } A$$

De même, le rang, le déterminant, les valeurs propres, le polynôme caractéristique et le polynôme minimal sont des invariants de similitudes, mais ils ne forment pas un système complet, c'est-à-dire qu'ils ne suffisent pas toujours à détecter la non-similitude de deux matrices.

Définie positivité

En algèbre linéaire, la notion de matrice définie positive est analogue à celle de nombre réel strictement positif : une matrice définie positive est une matrice positive inversible.

Definition 13 (Matrice symétrique réelle définie positive) *Soit M une matrice symétrique réelle d'ordre n. Elle est dite définie positive si elle est positive et inversible, autrement dit si elle vérifie l'une des quatre propriétés équivalentes suivantes :*

1. Pour toute matrice colonne non nulle $\boldsymbol{x}$ à n éléments réels, on a $\boldsymbol{x}^{\mathsf{T}} M \boldsymbol{x} > 0$

(autrement dit, la forme quadratique définie par M est strictement positive pour $\mathrm{x} \neq 0$).

2.Toutes les valeurs propres de M (qui sont nécessairement réelles) sont strictement positives.

3.La forme bilinéaire symétrique $\mathbb{R}^n\mathbb{R}^n \to \mathbb{R}, \quad (\mathrm{x}, \mathrm{y}) \mapsto \mathrm{x}^{\mathsf{T}} M \mathrm{y}$ est un produit scalaire sur $\mathbb{R}^n$.

4.Il existe une matrice $N \in \mathcal{M}_n(\mathbb{R})$ inversible telle que $M = N^T N$ (autrement dit : M est congruente à la matrice identité).

Remark 14 (Matrice symétrique réelle définie négative) *Une matrice symétrique réelle est dite définie négative si son opposée (symétrique elle aussi) est définie positive.*

Intérêt des matrices définies positives

♠ Les problèmes de résolution de systèmes linéaires les plus faciles à traiter numériquement sont ceux dont les matrices sont symétriques définies positives.

♠ Toute matrice symétrique réelle positive est limite d'une suite de matrices symétriques réelles définies positives, ce qui est à la base de nombreux raisonnements par densité.

1.1.4 Différentiabilité

En analyse fonctionnelle et vectorielle, on appelle différentielle d'ordre 1 d'une fonction en un point a (ou dérivée de cette fonction au point a) la partie linéaire de l'accroissement de cette fonction entre a et a+h lorsque h tend vers 0. Elle généralise aux fonctions de plusieurs variables la notion de nombre dérivé d'une fonction d'une variable réelle, et permet ainsi d'étendre celle de développements limités. Cette différentielle n'existe pas toujours, et une fonction possédant une différentielle est appelée une fonction différentiable. On peut ensuite calculer des différentielles d'ordre supérieur à 1.

On utilise la notation différentielle avec beaucoup d'efficacité dans le cadre du calcul d'approximations et du calcul de dérivées. Elle facilite la formule de la dérivée de la composée. Elle se révèle très pratique dans le changement de variable en calcul intégral.

Différentielle au sens de Fréchet

Soient E un espace vectoriel normé et F un espace vectoriel topologique séparé, et f une application de E dans F. Soit a un point de E.

On dit que f est différentiable en a (au sens de Fréchet) si et seulement s'il existe une application linéaire continue L de E dans F telle que :

$$\forall h \in E \quad f(a+h) = f(a) + L(h) + o\left(\|h\|\right),$$

ou, de manière équivalente

$$\lim_{h \to 0} \frac{f(a+h) - f(a) - L(h)}{\|h\|} = 0.$$

Une telle application linéaire L est alors unique. Elle est appelée différentielle de f en a et se note $L = \mathrm{d}f(a)$. De plus, sa continuité assure la continuité en a de f.

Dérivée directionnelle

Soient E un espace vectoriel normé, U un ouvert de E, f une fonction définie sur U à valeurs dans un espace vectoriel normé F (ou plus généralement, un espace vectoriel topologique séparé), u un point de U et h un vecteur de E.

La dérivée de f au point u suivant le vecteur h est, si elle existe, la dérivée en 0 de la fonction de la variable réelle $t \to f(u+th)$:

$$D_h f(u) = \lim_{\substack{t \to 0 \\ t \neq 0}} \frac{f(u+th) - f(u)}{t}.$$

Si h est le vecteur nul, cette dérivée directionnelle existe toujours et a une valeur nulle. On pourra supposer que h n'est pas le vecteur nul dans ce qui suit.

Lorsque l'espace E est de dimension finie n et muni d'une base, la fonction f peut être vue comme une fonction de n variables réelles, et le calcul des dérivées directionnelles suivant les vecteurs de base correspond au calcul des dérivées partielles de f :

$$D_{e_i} f(u) = \frac{\partial f}{\partial x_i}(u).$$

Si on remplace h par un vecteur colinéaire αh, le calcul de dérivée est identique à la multiplication par le facteur α près :

$$D_{\alpha h} f(u) = \alpha D_h f(u).$$

Ainsi, lors-qu'existe en un point une dérivée suivant un vecteur, il en existe une suivant tout vecteur de même direction, mais la valeur de cette dérivée directionnelle dépend du choix du vecteur. On parlera de dérivée directionnelle de f au point u dans la direction de h lorsque le vecteur h est unitaire.

En revanche il n'y a pas de raison a priori d'observer un résultat particulier lorsqu'on somme deux vecteurs h et h' .

Différentielle au sens de Gâteaux

Soient E et F deux espaces vectoriels normés et $f : E \to F$ une fonction. On dit que f est Gâteaux différentiable en $x \in E$ si

la dérivée directionnelle $f'_D(x; d)$ existe quel que soit $d \in E$, l'application

$$D_G f(x) : d \in E \mapsto f'_D(x; d) \in F$$

est linéaire continue. On dit que f est continûment Gâteaux différentiable en $x \in E$ si f est Gâteaux différentiable dans un voisinage V de x et

$$D_G f : V \to \mathcal{L}(E, F)$$

est continue en x ; on a noté $\mathcal{L}(E, F)$ l'ensemble des opérateurs linéaires continus de E dans F, muni de sa norme canonique.

Gradient

En physique et en analyse vectorielle, le gradient est une grandeur vectorielle indiquant la façon dont une grandeur physique varie dans l'espace. En mathématiques, le gradient est un vecteur représentant la variation d'une fonction par rapport à la variation de ses différents paramètres.

Il est courant, selon la façon de noter des vecteurs, d'écrire le gradient d'une fonction f ainsi :

$$\overrightarrow{\operatorname{grad}} f \quad \text{ou} \quad \vec{\nabla} f$$

Souvent, en typographie, on préfère mettre un caractère en gras pour afficher son caractère vectoriel :

$$\mathbf{grad}\, f \quad \text{ou} \quad \nabla f.$$

Le gradient est d'une importance capitale en physique, où il fut d'abord employé. Utilisé en théorie des variations, il est aussi fondamental dans le domaine de l'optimisation ou de la résolution d'équations aux dérivées partielles. Il peut être intéressant d'en voir certains exemples avant d'en donner une définition plus mathématique.

Hessien

En mathématiques, la matrice hessienne (ou simplement la hessienne) d'une fonction numérique f est la matrice carrée, notée $H(f)$, de ses dérivées partielles secondes.

Plus précisément, étant donnée une fonction f à valeurs réelles

$$f : \mathbb{R}^n \to \mathbb{R} : f(x_1, x_2, ..., x_n),$$

et en supposant que toutes les dérivées partielles secondes de f existent, le coefficient d'indice i, j de la matrice hessienne de f vaut

$H_{ij}(f) = \frac{\partial^2 f}{\partial x_i \partial x_j}$ ou, en d'autres termes,

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

On appelle hessien (ou discriminant hessien) le déterminant de cette matrice.

Le terme « hessien » a été introduit par James Joseph Sylvester, en hommage au mathématicien allemand Ludwig Otto Hesse.

Soit notamment f une fonction de classe $\mathcal{C}^2$ définie sur un ouvert U de l'espace E, à valeurs réelles. Sa matrice hessienne est bien définie et en vertu du théorème de Schwarz, elle est symétrique.

On appelle forme hessienne la forme quadratique associée à la matrice hessienne.

Formule de Taylor pour les fonctions de plusieurs variables

Il existe des formule pour des fonctions n fois différentiables en un point a d'un ouvert Ω de $\mathbb{R}^p$ et à valeurs dans$\mathbb{R}$ (ou même à valeurs dans $\mathbb{R}^q$). Cependant, les coefficients multinomiaux(En mathématiques, la formule du multinôme de Newton est une relation donnant le développement d'une puissance entière n d'une somme d'un nombre fini m de termes sous forme d'une somme de produits de puissances de ces termes affectés de coefficients, lesquels sont appelés des coefficients multinomiaux) qui interviennent rendent l'expression assez lourde.

Considérons une boule ouverte B de $\mathbb{R}^p$ (généralisation de l'intervalle I) centrée en a et f une fonction à valeurs réelles définie sur l'adhérence B, possédant des dérivées partielles d'ordre n + 1 continues en chaque point. Alors pour tout $x \in B$:

$$f(x) = \sum_{|\alpha|=0}^{n} \frac{1}{\alpha!} \frac{\partial^\alpha f(a)}{\partial x^\alpha} (x-a)^\alpha + \sum_{|\alpha|=n+1} R_\alpha(x)(x-a)^\alpha$$

où les sommes portent sur les multi-indices α, et où le reste vérifie l'inégalité

$$|R_\alpha(x)| \leq \sup_{y \in \bar{B}} \left| \frac{1}{\alpha!} \frac{\partial^\alpha f(y)}{\partial x^\alpha} \right|$$

pour tous les α tels que $|\alpha| = n + 1$.

En particulier, pour une fonction f deux fois différentiable en

$$a \in \Omega \subset \mathbb{R}^n,$$

et à valeurs réelles, on peut écrire, pour tout $x \in \Omega$:

$$f(x) = f(a) + \nabla f(a) \cdot (x-a) + \frac{1}{2}(x-a)^T \mathbb{H}(a)(x-a) + o(\|x-a\|^2)$$

où ∇f est le gradient de f et $\mathbb{H}(a)$ est sa matrice hessienne évaluée en a.

1.1.5 Convergence des Suites

Une suite de nombres réels est dite convergente lorsqu'elle admet un nombre réel comme limite.

Cette notion a été généralisée, en topologie : une suite définie sur un espace topologique E converge vers un élément l de E si tout ouvert de E contenant l contient tous les éléments de la suite à partir d'un certain rang.

Une suite qui n'est pas convergente est une suite qui n'admet pas de limite ou dont la limite n'est pas dans l'ensemble étudié (par exemple, une suite de réels qui a pour limite $+\infty$ n'est pas convergente).

La démonstration de la convergence d'une suite, ainsi que la recherche et l'amélioration d'algorithmes de calcul d'une valeur approchée de la limite trouvent une place centrale dans la résolution de problèmes, en conjuguant les raisonnements de topologie, d'analyse numérique, d'algorithmique et des techniques de calcul et d'informatique.

Limite d'une suite complexe

On dit qu'une suite converge vers un complexe ℓ si

$$\forall \epsilon > 0, \exists N \in \mathbb{N}, \forall n \in \mathbb{N} : n \geq N \Rightarrow |u_n - \ell| < \epsilon$$

On remarque qu'il s'agit de la même définition que dans $\mathbb{R}$, au détail près qu'il ne s'agit plus de valeur absolue mais de module.

On écrit alors

$$\lim_{n \to +\infty} u_n = \ell$$

ou plus simplement, quand il n'y a pas ambiguïté, $\lim u = \ell$. On retrouve pour les suites complexes convergentes, les mêmes propriétés que pour les suites réelles, exceptées celles liées à la relation d'ordre : la limite est unique, une suite convergente est de module borné, toute suite de Cauchy converge (en effet, $\mathbb{C}$ est aussi complet), les différentes opérations comme somme, produit, quotient se transmettent bien à la limite.

Limite d'une suite dans un espace vectoriel normé

Dans un espace vectoriel normé, on dit qu'une suite (u_n) converge vers ℓ si

$$\forall \epsilon > 0, \exists N \in \mathbb{N}, \forall n \in \mathbb{N} : n \geq N \Rightarrow \|u_n - \ell\| < \epsilon$$

C'est une généralisation de la limite d'une suite complexe, la norme usuelle dans le plan complexe étant le module.

On écrit alors

$$\lim_{n \to +\infty} u_n = \ell$$

ou plus simplement, quand il n'y a pas ambiguïté,

$$\lim u = \ell.$$

L'unicité de la limite est conservée ainsi que la transmission à la limite de la somme et de la multiplication par un scalaire. Ce n'est que dans un espace vectoriel normé complet que l'on pourra affirmer que toute suite de Cauchy converge.

Limite d'une suite dans un espace métrique

Dans un espace métrique, on dit qu'une suite (u_n) converge vers ℓ si

$$\forall \epsilon > 0, \exists N \in \mathbb{N}, \forall n \in \mathbb{N} : n \geq N \Rightarrow (d(u_n, \ell) < \epsilon).$$

On remarque qu'il s'agit de la même définition que dans $\mathbb{R}$, au détail près qu'il ne s'agit plus de valeur absolue d'une différence mais de distance.

On écrit alors

$$\lim_{n \to +\infty} u_n = \ell$$

ou plus simplement, quand il n'y a pas ambiguïté,

$$\lim u = \ell.$$

Seule l'unicité de la limite est conservée. Il faudra être dans un espace métrique complet pour pouvoir dire que toute suite de Cauchy converge. Si une opération existe sur l'espace en question, il faudra qu'elle soit continue pour se transmettre à la limite.

Limite d'une suite dans un espace topologique

Toutes les définitions précédentes se rejoignent dans la définition de la convergence dans un espace topologique.

Soit E un espace muni d'une topologie T.

On dit que la suite

$$(u_n)_{n \in \mathbb{N}} \in E^{\mathbb{N}}$$

converge vers $\ell \in E$ si, pour tout ouvert O de T contenant l'élément ℓ , il existe un entier naturel N tel que tous les u_n pour $n \geq N$ appartiennent à O.

Il suffit que l'espace soit séparé pour pouvoir affirmer que la limite est unique.

Valeurs d'adhérence

Definition 15 *Soit $(u_n)_{n \in \mathbb{N}}$ une suite à valeurs dans un espace métrique E.*

Si

$$\mathbb{N} \to \mathbb{N}, n \to \sigma(n)$$

est une fonction strictement croissante (une telle fonction s'appelle une ex tractrice), on dit que la suite

$$(u_{\sigma(n)})_{n \in \mathbb{N}}$$

est une suite extraite (ou sous-suite) de la suite

$$(u_n)_{n \in \mathbb{N}}.$$

Grosso modo, c'est la suite

$$(u_n)_{n\in\mathbb{N}}$$

pour laquelle on n'a gardé que certains termes (une infinité quand même).

On dit que la valeur ℓ est une valeur d'adhérence de la suite

$$(u_n)_{n\in\mathbb{N}}$$

s'il existe une suite extraite de $(u_n)_{n\in\mathbb{N}}$ qui converge vers ℓ.

Pour se faire une idée, une valeur d'adhérence est un élément « près duquel la suite passe souvent », c'est-à-dire qu'aussi loin qu'on aille, on trouvera toujours un terme de la suite près de cet élément.

Propriétés

Propriété 1 Si une suite

$$(u_n)_{n\in\mathbb{N}}$$

à valeurs dans E converge vers $l \in E$, l est l'unique valeur d'adhérence de

$$(u_n)_{n\in\mathbb{N}}$$

, c'est-à-dire que toutes les suites extraites $(u_{\sigma(n)})_{n\in\mathbb{N}}$ convergent vers l.

Dans le cas où E est un espace compact, on dispose même d'une réciproque. Elle s'applique par exemple à toute suite à valeurs dans un segment de $\mathbb{R}$ (autrement dit à toute suite réelle bornée), ou encore à toute suite réelle, en prenant comme compact la droite réelle achevée (dans ce cas, $+\infty$ et $-\infty$ ne sont pas exclus a priori de l'inventaire des valeurs d'adhérence de la suite)

Propriété 2 Si une suite est à valeurs dans un espace compact E, alors elle admet au moins une valeur d'adhérence dans E, et elle converge si et seulement si elle n'en admet qu'une.

Propriété 3 Une suite

$$(u_n)_{n\in\mathbb{N}}$$

à valeurs dans E converge vers $l \in E$si et seulement si :

- $(u_{2n})_{n\in\mathbb{N}}$ converge vers l et
- $(u_{2n+1})_{n\in\mathbb{N}}$ converge vers l.

On voit d'ailleurs bien comment généraliser ce résultat : il suffit que les images des ex tractrices considérées recouvrent entièrement $\mathbb{N}$ (par exemple, ici, $n \mapsto 2n$ et $n \mapsto 2n+1$), c'est-à-dire que les ensembles (infinis) d'indices des suites extraites utilisées aient pour réunion l'ensemble des naturels.

Théorème de Bolzano-Weierstrass Toujours lorsque E est un espace métrique, on dispose du puissant théorème de Bolzano-Weierstrass :

Un espace métrique E est compact si (et seulement si) il est séquentielle-ment compact, c'est-à-dire si toute suite à valeurs dans E admet au moins une valeur d'adhérence dans E.

1.1.6 Extrema d'une fonction

Le maximum d'une fonction f définie sur un ensemble E et à valeurs dans un ensemble F ordonné est le maximum de l'ensemble des valeurs prises par f (de la partie $f(E)$ de F). Ainsi m est le maximum de f s'il existe un élément a de E tel que $f(a) = m$ et tel que pour tout élément x de E, $f(x) \leq f(a)$; l'élément a (qui n'est pas nécessairement unique) est appelé point de maximum de f.

Dans le cas où l'espace de départ de f est muni d'une structure topologique (par exemple si f est une fonction d'une ou plusieurs variables réelles à valeurs réelles), on distingue deux types d'extrema : les extrema globaux, qui correspondent à la définition précédente, et les extrema locaux.

Extremum local d'une fonction Soient une fonction f définie sur un espace topologique E et a un point de E. On dit que f atteint en a un maximum local s'il existe un voisinage V de a tel que pour tout élément x de V, on ait $f(x) \leq f(a)$.

On dit alors que $f(a)$ est un « maximum local » de f sur E et que a est un point de maximum local de f.

Lorsqu'il existe un voisinage V de a tel que pour tout élément x de V différent de a, on ait $f(x) < f(a)$, on dit que f atteint en a un maximum local strict.

Théorèmes topologiques d'existence d'extrema globaux Soit une fonction $f : D \to \mathbb{R}$, où D est un espace topologique. Par exemple, D peut être une partie de $\mathbb{R}$ (cas d'une fonction d'une variable réelle), ou d'un espace $\mathbb{R}^k$, avec k un entier naturel (cas d'une fonction de k variables réelles).

L'existence d'extrema globaux est assurée dès lors que la fonction f est continue et que la partie D est compacte : en effet, l'image $f(D)$ est alors une partie compacte de l'espace d'arrivée $\mathbb{R}$; en tant que partie bornée de $\mathbb{R}$, elle admet une borne supérieure, et cette borne supérieure est dans $f(D)$ puisque cette partie est fermée.

En dimension $k = 1$, c'est en particulier le cas si I est un intervalle fermé borné, c'est-à-dire de la forme [a, b] (voir Théorème des bornes). En dimension supérieure k, c'est en particulier le cas si D est une boule fermée (de la forme

$$D = B(A, r) = \{X \in \mathbb{R}_k \mid \|X - A\| \leq r\},$$

ou $\| \ \|$ désigne une norme sur $\mathbb{R}^k$).

Méthodes issues du calcul différentiel pour la recherche d'extrema locaux Soit une fonction

$$f : D \to \mathbb{R},$$

où D est une partie ouverte de $\mathbb{R}^k$; par exemple, dans le cas d'une variable réelle, D peut être un intervalle ouvert de la forme $]a, b[$ (avec a et b des nombres réels, ou $a = -\infty$, ou $b = +\infty$).

Si la fonction f atteint un extremum local en un point a où elle est différentiable, alors toutes ses dérivées partielles s'annulent en a ; en particulier, dans le cas d'une fonction d'une seule variable, le nombre dérivé de f en a est nul.

Pour cette raison, l'étude des extrema passe souvent par la recherche des points d'annulation de la dérivée, appelés points critiques de f. Un point critique n'est pas nécessairement un point d'extremum, comme le montre l'exemple de la fonction

$$f : \mathbb{R} \to \mathbb{R},\ x \mapsto x^3$$

au point 0. On peut, cependant, sous certaines hypothèses supplémentaires, affirmer qu'un point critique est un point d'extremum.

Cas d'une fonction d'une variable Condition suffisante pour un extremum local (voir figure 1.1) : Si f est dérivable sur I, et si a est un point intérieur à I où la dérivée de f s'annule en changeant de signe, alors f atteint un extremum local en a. Plus précisément, en supposant

$$f'(a) = 0.$$

S'il existe α réel strictement positif tel que

$$[a - \alpha,\, a + \alpha] \subset I \quad \text{et} \quad f' \geq 0 \quad \text{sur} \quad [a - \alpha,\, a], f' \leq 0 \quad \text{sur} \quad [a,\, a + \alpha],$$

alors f atteint un maximum local en a. S'il existe α réel strictement positif tel que

$$[a - \alpha,\, a + \alpha] \subset I \quad \text{et} \quad f' \leq 0 \quad \text{sur} \quad [a - \alpha,\, a], f' \geq 0 \quad \text{sur} \quad [a,\, a + \alpha],$$

alors f atteint un minimum local en a.

Cas des fonctions de plusieurs variables Condition suffisante pour un extremum local :

On suppose ici que A est un ouvert, et que f est deux fois dérivable en un point a de A. La (matrice) hessienne de f en a est notée $\nabla^2 f(a)$; elle est symétrique réelle.

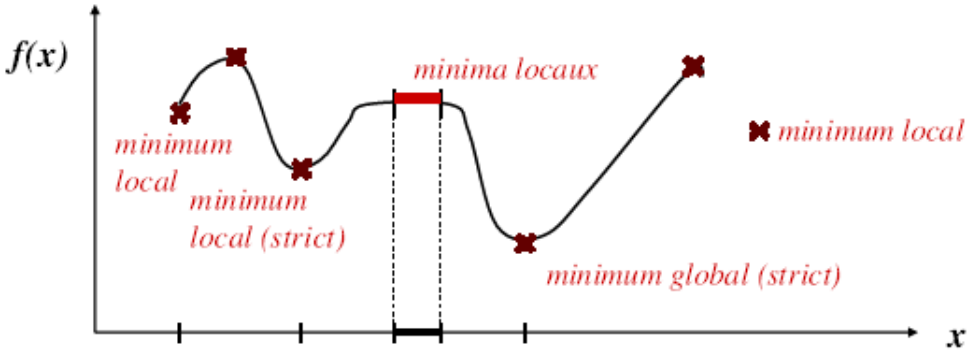

FIGURE 1.1: Extrema d'une fonction d'une variable

• Si $\nabla f(a) = 0$ et si $\nabla^2 f(a)$ est définie négative, alors f atteint un maximum local strict en a.

• Si $\nabla f(a) = 0$ et si $\nabla^2 f(a)$ est définie positive, alors f atteint un minimum local strict en a.

Rappel : par définition, la hessienne de f en a est la matrice carrée d'ordre n ayant

$$\frac{\partial^2 f}{\partial x_i \, \partial x_j}(a),$$

pour élément en ligne i et colonne j. Comme f est deux fois dérivable en a, il résulte du théorème de Schwarz sur les dérivées partielles d'ordre 2 que la hessienne en a est symétrique.

1.2 Définition d'un problème d'optimisation

• Minimiser la fonction $f(x)$ avec $x \in \mathbb{R}^n$ sous les contraintes :

$c_i(x) = 0, i = 1, ...m;$ (contraintes d'égalité)

$d_j(x) \geq 0, j = 1, ...p;$ (contraintes d'inégalité)

$f(x)$ est appelée fonction de coût, fonction objectif, ou critère d'optimisation, c-a-d minimiser $f(x)$ avec

$$x \in \Omega \subset \mathbb{R}^n.$$

Donc de point de vue mathématique, l'optimisation est la minimisation ou la maximisation d'une fonction soumise à des contraintes sur ses variables.

ou Ω est l'ensemble admissible :

$$\Omega = \{x : c_i(x) = 0 \quad \text{et} \quad d_j(x) \geq 0\}$$

1.2.1 Classification des problèmes d'optimisation

• Suivant les propriétés de la fonction de coût :

– Fonction d'une seule variable
– Fonction linéaire
– Somme de carrés de fonctions linéaires
– Fonction quadratique
– Fonction convexe
– Somme de carrés de fonctions non linéaires
– Fonction non linéaire continûment dérivable
– Fonction non linéaire non dérivable

• Suivant les propriétés des contraintes :

– Pas de contraintes
– Simples bornes
– Fonctions linéaires
– Fonctions convexes
– Fonctions non linéaires continûment dérivables
– Fonctions non linéaires non dérivables

1.2.2 Terminologie

• **Programmation linéaire :**

Un programme linéaire est un problème dans lequel on est amené à maximiser (ou minimiser) une application linéaire, appelée fonction d'objectif ou fonction économique , sur un ensemble d'équations et/ou d'inéquations linéaires, dites contraintes . Autrement dit, la programmation linéaire est une branche des mathématiques qui a pour but de résoudre des problèmes d'optimisation linéaire de type :

$$f(x) = C^T x$$

$$\Omega = \{x : A_1 x = b_1 \quad \text{et} \quad A_2 x \geq b_2\}$$

Les coefficients C , A_1 et A_2 sont des matrices réelles fixées et les b_1,b_2 sont des vecteurs réels donnés. Les contraintes d'inégalités éventuelles sont toutes larges et non strictes. Il se peut qu'une contrainte d'inégalité soit de type "$\geq$". En multipliant chaque inégalité de type " $\geq$ " par (-1), on peut supposer que toutes les contraintes d'inégalité sont de type " $\leq$ ".

• **Programmation quadratique :** Un problème d'optimisation quadratique est un problème d'optimisation dans lequel on minimise (ou maximise) une fonction quadratique sur un polyèdre convexe. Les contraintes peuvent donc être décrites par des fonctions linéaires (on devrait dire affines). Autrement dit, la pro-

grammation quadratique est une branche des mathématiques qui a pour but de résoudre des problèmes d'optimisation linéaire de type :

$$f(x) = (1/2)x^T Ax + b^T x + c : A = A^T > o$$

$$\Omega = \{x : C_1 x = d_1 \quad \text{et} \quad C_2 x \geq d_2\}.$$

L'optimisation linéaire peut être vue comme un cas particulier de l'optimisation quadratique.

• **Programmation convexe :** L'optimisation convexe est une sous-discipline de l'optimisation mathématique, dans laquelle le critère à minimiser $f(x)$ est convexe et l'ensemble admissible Ω est convexe. Ces problèmes sont plus simples à analyser et à résoudre que les problèmes d'optimisation non convexes.

• **Programmation non linéaire :**

En optimisation, vue comme branche des mathématiques, l'optimisation non linéaire (en anglais : nonlinear programming – NLP) s'occupe principalement des problèmes d'optimisation dont les données, i.e., les fonctions et ensembles définissant ces problèmes, sont non linéaires (bien sûr), mais sont aussi différentiables autant de fois que nécessaire pour l'établissement des outils théoriques, comme les conditions d'optimalité, ou pour la bonne marche des algorithmes de résolution qui y sont introduits et analysés. Cette sous-discipline de l'optimisation, à la frontière mal définie et dont l'introduction est un peu artificielle, a aussi son existence liée à la communauté de chercheurs qui se sont spécialisés sur ces sujets et au type de résultats qui ont pu être obtenus.

Elle complimente l'optimisation non lisse (ou non différentiable), elle aussi liée à une communauté de chercheurs spécialisés. Ces deux disciplines se rassemblent pour former ce que l'on appelle l'optimisation continue, qui jouxte, quant à elle, d'autres sous-disciplines telles que l'optimisation combinatoire (ou discrète), l'optimisation stochastique, etc.

Chapitre 2

Les problèmes de minimisation sans contraintes et leurs algorithmes

2.1 Aspect général des algorithmes

Les algorithmes d'optimisation sont itératifs. Ils commencent avec un jeu (guess)initial des valeurs optimales des variables et puis ils génèrent une suite d'estimés améliorées jusqu'à l'obtention d'une solution. La stratégie employée afin de procéder d'une itérée à la suivante est ce qui distingue un algorithme d'un autre. La plupart des stratégies utilisent les valeurs de la fonction objectif f , les contraintes et possiblement les première et deuxième dérivées de ces fonctions. Certains algorithmes accumulent l'information obtenue aux itérations précédentes, tandis que d'autres utilisent que de l'information locale du point actuel. Tout algorithme de bonne qualité devrait posséder les propriétés suivantes :

• Robustesse : L'algorithme doit marcher bien sur une gamme de problèmes dans sa classe, pour tous les choix raisonnables de valeurs initiales.

• Eficacité : L'algorithme ne doit pas nécessiter du temps ou du stockage excessif sur l'ordinateur.

• Précision : L'algorithme doit être capable d'identifier une solution avec précision, sans être trop sensible aux erreurs dans les données ou erreurs d'arrondi pendant le calcul.

Ces exigences peuvent être en conflit. Par exemple, une méthode qui converge rapidement peut demander trop de place de stockage pour des grands problèmes. De l'autre coté, une méthode robuste peut être trop lente. Des compromis entre taux de convergence et besoins de stockage, entre robustesse et vitesse, et ainsi de suite, sont des problèmes au centre de l'optimisation numérique. Ils recevront de la considération soigneuse dans ce cours. La théorie mathématique d'optimisation est utilisée aussi bien pour caractériser des points optimaux, que pour fournir une

base pour les algorithmes. Il est impossible d'avoir une bonne compréhension d'optimisation numérique sans une prise ferme sur la théorie sous-jacente. Ainsi, ce cours va donner un traitement solide des conditions d'optimalité, aussi bien que l'analyse de convergence qui révèle les forces et les faiblesses des algorithmes les plus importants.

2.1.1 Algorithme à directions de descente

Definition 16 *Un algorithme à directions de descente est un algorithme d'optimisation différentiable (l'optimisation dont il est question ici est une branche des mathématiques), destiné à minimiser une fonction réelle différentiable définie sur un espace euclidien (par exemple, $\mathbb{R}^n$, l'espace des n-uplets de nombres réels, muni d'un produit scalaire) ou, plus généralement, sur un espace hilbertien. L'algorithme est itératif et procède donc par améliorations successives. Au point courant, un déplacement est effectué le long d'une direction de descente, de manière à faire décroître la fonction. Le déplacement le long de cette direction est déterminé par la technique numérique connue sous le nom de recherche linéaire.*

Cette approche algorithmique peut être vue comme une technique de globalisation, c'est-à-dire une méthode permettant d'obtenir la convergence des itérés (sous certaines conditions) quel que soit l'itéré initial choisi. Elle s'apparente ainsi aux algorithmes à régions de confiance ; ces dernières améliorent légèrement (mais parfois de manière décisive) leurs résultats de convergence mais sont plus compliquées à mettre en œuvre, ce qui limite parfois leur application.

Les algorithmes à directions de descente s'étendent aux problèmes avec contraintes simples (pourvu que la projection sur l'ensemble admissible soit aisée, peu coûteuse en temps de calcul) ou pour des problèmes avec contraintes fonctionnelles non linéaires, par l'intermédiaire de fonctions de mérite. Elles sont aussi utilisées en optimisation non lisse.

Principes de l'algorithme

Cadre Le cadre est le suivant. On cherche un point x_* qui minimise une fonction différentiable

$$x \in \mathbb{E} \mapsto f(x) \in \mathbb{R}$$

définie sur un espace hilbertien $\mathbb{E}$, dont on note $\langle \cdot, \cdot \rangle$ le produit scalaire et $\| \cdot \|$ la norme associée. On note $f'(x)$ et $\nabla f(x)$ la dérivée et le gradient de f en x, si bien que

$$\forall\, d \in \mathbb{E} : \qquad f'(x) \cdot d = \langle \nabla f(x), d \rangle.$$

Énoncé Les algorithmes à directions de descente cherchent un minimum de f en générant une suite de points $(x_k)_{k\geqslant 1}$, appelés itérés, qui approchent de mieux en mieux un minimiseur x_* du critère f, si tout va bien. Cette suite est construite en se fondant sur deux constructions :

calcul d'une direction de descente $d_k \in \mathbb{E}$, détermination d'un pas α_k, qui est un nombre réel strictement positif, le long de la direction de descente de telle sorte que le nouvel itéré donne au critère une valeur inférieure à celle qu'il a en l'itéré courant ; le nouvel itéré est de la forme suivante :

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

cette opération de détermination du pas s'appelle la recherche linéaire. Ces deux opérations seront décrites ci-dessous, mais on peut dès à présent résumer l'algorithme. Il s'agit d'un schéma algorithmique, car beaucoup d'aspects de celui-ci ne sont pas spécifiés avec précision.

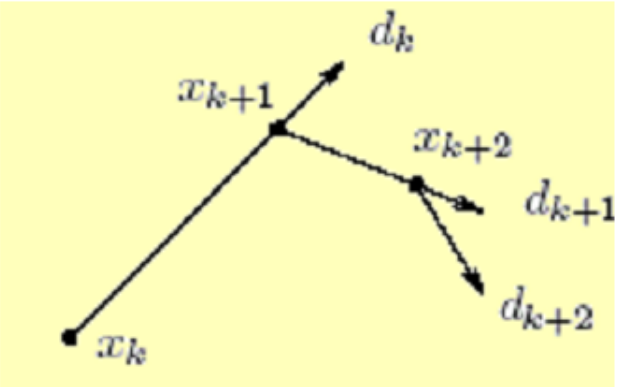

FIGURE 2.1: Algorithmes à directions de descente

Schéma On se donne un point/itéré initial $x_1 \in \mathbb{E}$ et un seuil de tolérance $\varepsilon \geqslant 0$. Un algorithme à directions de descente définit une suite d'itérés $x_1, x_2, \ldots \in \mathbb{E}$, jusqu'à ce qu'un test d'arrêt soit satisfait. Il passe de x_k à x_{k+1}(voir figure 2.1) par les étapes suivantes.

1) Simulation : calcul de $\nabla f(x_k)$ au moins.

2) Test d'arrêt : si$\|\nabla f(x_k)\| \leqslant \varepsilon$, arrêt.

3) Direction : calcul d'une direction de descente $d_k \in \mathbb{E}$.

4) Recherche linéaire : déterminer un pas $\alpha_k > 0$ le long de d_k. Nouvel itéré :

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

À propos du test d'arrêt En pratique, il faudra prendre $\varepsilon > 0$ dans le test d'arrêt de l'étape 2 ; la valeur nulle de cette tolérance a été admise uniquement pour simplifier l'expression des résultats de convergence ci-dessous.

Dans les problèmes sans contrainte, il est normal que le test d'arrêt porte sur la petitesse du gradient (ε est généralement pris petit). C'est en effet ce que suggère la condition nécessaire d'optimalité du premier ordre $\nabla f(x_*) = 0$. Comme x_k n'est jamais exactement égal à x_*, ce test ne pourra fonctionner que si $\nabla f(x)$ est faible en norme pour x voisin de x_*, ce qui revient pratiquement à supposer que f est de classe C^1.

Par ailleurs, un tel test d'arrêt suggère qu'un algorithme à directions de descente ne peut pas trouver mieux qu'un point stationnaire de f. C'est en effet souvent le cas, mais ce point faible est rarement rédhibitoire en pratique. On peut noter qu'il existe une version élaborée des méthodes à régions de confiance qui permet de trouver un minimum local, évitant ainsi les points stationnaires qui n'ont pas cette propriété de minimalité locale.

On est parfois tenté d'arrêter l'algorithme si le critère f ne décroît presque plus. Ceci n'est pas sans risque et il vaut mieux ne pas utiliser un tel test d'arrêt, car une faible variation du critère peut se produire loin d'une solution. En effet, au premier ordre, $f(x_{k+1}) \simeq f(x_k)$ revient à $\alpha_k \langle \nabla f(x_k), d_k \rangle \simeq 0$, ce qui peut arriver si le pas α_k est petit (c'est en général très suspect) ou si la direction de descente fait avec l'opposé du gradient un angle proche de 90 degrés, une situation qui se rencontre fréquemment (si l'algorithme est bien conçu, cela traduit un mauvais conditionnement du problème).

Même si le test d'arrêt de l'étape 2 est suggéré par la théorie, on peut s'interroger sur sa pertinence, du point de vue suivant : peut-on préciser dans quelle mesure le fait d'avoir un petit gradient implique que l'itéré est proche d'un point stationnaire de f ? Le cas où f est quadratique strictement convexe est instructif : $f(x) = \frac{1}{2} x^\top A x - b^\top x, \qquad \text{avec} \quad A \succ 0$. Minimiser f revient alors à déterminer l'unique solution x_* du système linéaire $Ax = b$. Par ailleurs, le gradient de f (pour le produit scalaire euclidien) est le résidu du système linéaire : $\nabla f(x) = Ax - b$. Or on sait bien que, si le conditionnement de A est élevé, on peut très bien avoir $\|Ax - b\|_2$ petit et une erreur $\|x - x_*\|_2$ importante. Le test d'arrêt portant sur la petitesse du gradient doit donc être interprété avec précaution.

2.1.2 Choix d'une direction de descente

En optimisation différentiable, qui est une discipline d'analyse numérique en mathématiques étudiant en particulier les algorithmes minimisant des fonctions différentiables sur des ensembles, une direction de descente est une direction le long de laquelle la fonction à minimiser a une dérivée directionnelle strictement négative. Ces directions sont utilisées par les méthodes à directions de descente.

C'est le long de ces directions qu'un déplacement est effectué afin de trouver l'itéré suivant, en lequel la fonction à minimiser prend une valeur inférieure à celle qu'elle a en l'itéré courant. Des directions de descente peuvent être calculées par de nombreuses techniques dont les plus classiques sont présentées ci-dessous.

Definition 17 *Soient $\mathbb{E}$ un espace vectoriel et*

$$f : \mathbb{E} \to \mathbb{R}$$

une fonction réelle définie sur $\mathbb{E}$, admettant des dérivées directionnelles au point $x \in \mathbb{E}$ considéré. On note

$$f'(x; d) := \lim_{t \downarrow 0} \frac{f(x + td) - f(x)}{t} \in \mathbb{R}$$

la dérivée directionnelle de f en $x \in \mathbb{E}$ dans la direction $d \in \mathbb{E}$. La notation $t \downarrow 0$ signifie que le réel t tend vers zéro dans $\mathbb{R}$ par des valeurs strictement positives.

La notion de direction de descente est surtout utilisée en optimisation numérique.

Direction de descente Une direction de descente de f en x est un vecteur $d \in \mathbb{E}$ tel que

$$f'(x; d) < 0.$$

On en déduit que

$$\forall\, \alpha > 0 \quad \text{petit :} \qquad f(x + \alpha d) < f(x),$$

si bien que f décroît en x dans la direction d. Cette propriété justifie le nom donné à cette direction. Certains auteurs utilisent cette dernière propriété comme définition d'une direction de descente ; cependant, comme cette propriété n'implique pas que la dérivée directionnelle soit strictement négative, elle n'est pas suffisamment forte pour les algorithmes à directions de descente.

Exemples de direction de descente Il existe de nombreuses méthodes permettant de calculer une direction de descente. Les principales sont présentées dans cette paragraphe ; chacune avec ses propres caractéristiques. Un algorithme qui utilise une telle direction hérite d'elle son nom. Ainsi l'algorithme du gradient est l'algorithme à directions de descente qui utilise la direction du gradient, l'algorithme du gradient conjugué utilise la direction du gradient conjugué, etc.

On suppose dans cette paragraphe que $\mathbb{E}$ est un espace hilbertien, dont le produit scalaire est noté $\langle \cdot, \cdot \rangle$ et la norme associée $\| \cdot \|$, et que

$$f : \mathbb{E} \to \mathbb{R}$$

est au moins une fois différentiable au point x considéré. On note $f'(x)$ sa dérivée, qui est une application linéaire continue de $\mathbb{E}$ dans $\mathbb{R}$, et

$$f'(x) \cdot d = f'(x; d)$$

la valeur en $d \in \mathbb{E}$ de cette dérivée. Par le théorème de Riesz-Fréchet, il existe alors un vecteur $\nabla f(x) \in \mathbb{E}$, appelé le gradient de f en x, défini par :

$$\forall\, d \in \mathbb{E} : \qquad f'(x) \cdot d = \langle \nabla f(x), d \rangle.$$

Direction du gradient La direction du gradient d est, en réalité, l'opposé du gradient :

$$d = -\nabla f(x).$$

Il s'agit bien d'une direction de descente si

$$f'(x) \neq 0$$

puisqu'alors

$$f'(x) \cdot (-\nabla f(x)) = \langle \nabla f(x), -\nabla f(x) \rangle = -\|\nabla f(x)\|^2 < 0.$$

Remark 18 *L'algorithme du gradient, qui utilise les directions du gradient comme directions de descente, est lent et il vaut mieux l'éviter, c'est pour ça qu'on a fait quelques remède a cet algorithme tel que :*

► *Au lieu de prendre comme direction de descente, la direction :*
$d_k = -\nabla f(x_k)$
on prend des directions de la forme :
$d_k = -D.\nabla f(x_k)$*, ou* D *est une matrice choisie convenablement.*
► *Un autre choix pourrait s'opérait de la façon suivante :*
$d_k = -\nabla f(x_k) + u_k$ *ou* u_k *est vecteur approprié.*

Direction de Newton On suppose ici que la fonction f à minimiser est deux fois différentiable en x et on désigne par $\nabla^2 f(x)$ son hessien en x, lequel est l'unique opérateur linéaire auto-adjoint

$$\nabla^2 f(x) : \mathbb{E} \to \mathbb{E}$$

vérifiant

$$\forall\, (h, k) \in \mathbb{E}^2 : \qquad f''(x) \cdot (h, k) = \langle \nabla^2 f(x) h, k \rangle.$$

La direction de Newton est définie en un point x en lequel le hessien de f est inversible par

$$d = -\left(\nabla^2 f(x)\right)^{-1} \nabla f(x).$$

Cette direction est une direction de descente si

$\triangleright$ $f'(x) \neq 0$,

$\triangleright$ $\nabla^2 f(x)$ est définie positive.

En effet

$$f'(x) \cdot d = \langle \nabla f(x), d \rangle = -\langle \nabla f(x), \nabla^2 f(x)^{-1} \nabla f(x) \rangle \leqslant -\lambda_{\max}^{-1} \|\nabla f(x)\|^2 < 0,$$

où $\lambda_{\max}$ désigne la plus grande valeur propre de $\nabla^2 f(x)$.

La seconde condition assurant le caractère descendant de la direction de Newton sera vérifiée dans le voisinage d'une solution vérifiant les conditions suffisantes d'optimalité du deuxième ordre.

Direction de quasi-Newton Les algorithmes de quasi-Newton en optimisation définissent une direction de descente en prenant une approximation convenable du hessien du critère au moyen d'un opérateur M auto-adjoint :

$$M \sim \nabla^2 f(x).$$

Une direction de quasi-Newton est donc de la forme

$$d = -M^{-1} \nabla f(x)$$

. Comme pour la direction de Newton, cette direction est une direction de descente si

$\triangleright$ $f'(x) \neq 0$,

$\triangleright$ M est définie positive.

En effet

$f'(x) \cdot d = \langle \nabla f(x), d \rangle = -\langle \nabla f(x), M^{-1} \nabla f(x) \rangle \leqslant -\lambda_{\max}^{-1} \|\nabla f(x)\|^2 < 0$,

où $\lambda_{\max}$ désigne la plus grande valeur propre de M.

Direction de Gauss-Newton La direction de Gauss-Newton est utilisée pour résoudre les problèmes de moindres-carrés dont le critère est de la forme

$$f(x) := \frac{1}{2} \|F(x)\|^2,$$

où $F : \mathbb{E} \to \mathbb{F}$ ($\mathbb{F}$ est un espace hilbertien dont le produit scalaire est aussi noté $\langle \cdot, \cdot \rangle$et $\| \cdot \|$ est la norme associée). On calcule aisément

$$\begin{aligned} f'(x) \cdot h &= \langle F(x), F'(x) \cdot h \rangle \\ &= \langle F'(x)^* F(x), h \rangle. \\ f''(x) \cdot (h,k) &= \langle F'(x) \cdot h, F'(x) \cdot k \rangle + \langle F(x), F''(x) \cdot (h,k) \rangle \\ &= \langle \left(F'(x)^* F'(x) \right) h, k \rangle + \langle F(x), F''(x) \cdot (h,k) \rangle. \end{aligned}$$

La direction de Gauss-Newton s'obtient en ne gardant du hessien de f que son premier terme dans l'expression ci-dessus, de manière à éviter le calcul des dérivées secondes de F. C'est en réalité une solution arbitraire d de l'équation normale

$$\left(F'(x)^* F'(x) \right) d = -F'(x)^* F(x).$$

On reconnaît dans le membre de droite l'opposé du gradient de f. Cette équation linéaire a en fait une solution unique si et seulement si $F'(x)$ est injective. Les directions de Gauss-Newton sont aussi les solutions du problème de moindres-carrés linéaire suivant

$$\min_{x \in \mathbb{E}} \frac{1}{2} \|F(x) + F'(x) \cdot d\|^2.$$

Une direction de Gauss-Newton d est une direction de descente de f en x si

$$f'(x) \neq 0.$$

En effet

$$f'(x) \cdot d = \langle \nabla f(x), d \rangle = -\langle \left(F'(x)^* F'(x) \right) d, d \rangle = -\|F'(x) d\|^2 < 0.$$

L'inégalité stricte vient du fait que si

$$F'(x) d = 0$$

, alors

$$\nabla f(x) = F'(x)^* F(x)$$

est nul par l'équation normale, ce que nous avons supposé ne pas avoir lieu.

2.1.3 Règles de recherche linéaire

En optimisation mathématique, la recherche linéaire est l'une des deux approches classiques permettant de forcer la convergence des algorithmes de calcul d'un minimum x_* d'une fonction

$$f : \mathbb{R}^n \to \mathbb{R},$$

lorsque le premier itéré est éloigné d'un tel minimum. L'autre méthode est celle des régions de confiance.

Plusieurs règles permettant de déterminer la valeur du paramètre α_k existent. Elles consistent, pour la plupart, à trouver la valeur qui minimise la fonction-coût

$$h_k(\alpha) = f(x_k + \alpha d_k)$$

. Considérant que d_k est une direction de descente, on obtient

$$h'_k(0) = f(x_k) \cdot d_k < 0,$$

ce qui permet de déterminer le comportement de h_k en fonction des valeurs de α. Il convient toutefois d'être prudent :

Objectifs à atteindre

Le premier objectif : Consiste à faire décroître f suffisamment. Cela se traduit le plus souvent par la réalisation d'une inégalité de la forme

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \quad \text{"un terme négatif "} \tag{2.1}$$

Le terme négatif, disons ν_k, joue un rôle-clé dans la convergence de l'algorithme utilisant cette recherche linéaire. L'argument est le suivant : Si $f(x_k)$ est minorée (il existe une constante C telle que

$$f(x_k) \geq C \quad \forall k$$

, alors ce terme négatif tend nécessairement vers zéro :

$$\nu_k \to 0$$

. C'est souvent à partir de la convergence vers zéro de cette suite que l'on parvient à montrer que le gradient lui-même doit tendre vers zéro. Le terme négatif devra prendre une forme bien particulière si on veut pouvoir en tirer de l'information. En particulier, il ne suffit pas d'imposer :

$$f(x_k + \alpha_k d_k) < f(x_k).$$

Le second objectif : Consiste d'empêcher le pas $\alpha_k > 0$ d'être trop petit, trop proche de zéro. Le premier objectif n'est en effet pas suffisant car l'inégalité (2.1) est en général satisfaite par des pas $\alpha_k > 0$ arbitrairement petit. Or ceci peut entraîner une "fausse convergence", c'est-à-dire la convergence des itérés vers un point non stationnaire.

Recherches linéaires exactes et inexactes Il existe deux grandes classes de méthodes qui s'intéressent à l'optimisation unidimensionnelle :

Recherches linéaires exactes Comme on cherche à minimiser f, il semble naturel de chercher a minimiser le critère le long de d_k et donc de déterminer le pas α_k comme solution du problème

$$\min h_k(\alpha), \quad \alpha > 0.$$

C'est ce que l'on appelle la règle de Cauchy et le pas déterminé par cette règle est appelé pas de Cauchy ou pas optimal. Dans certains cas, on préférera le plus petit point stationnaire de h_k qui fait décroître cette fonction :

$$\alpha_k = \inf\{\alpha \geq 0 : h_k'(\alpha) < h_k(0)\}.$$

On parle alors de règle de Curry et le pas déterminé par cette règle est appelé pas de Curry. De manière un peu imprécise, ces deux règles sont parfois qualifiées de recherche linéaire exacte(voir figure 2.2)

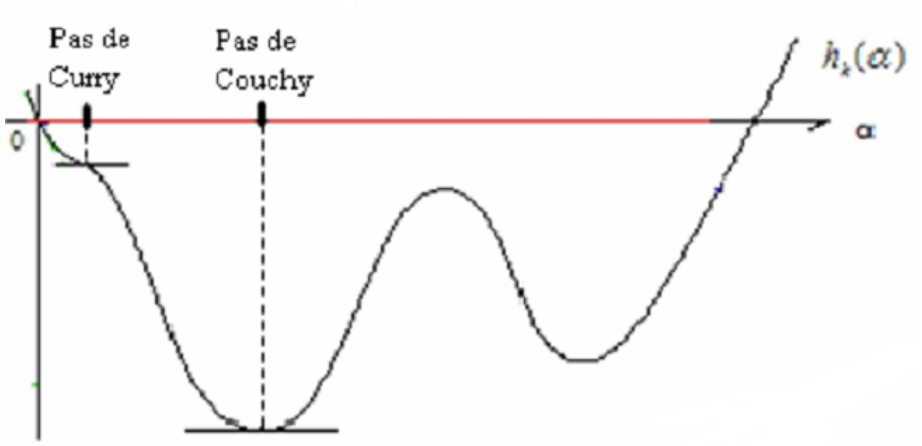

FIGURE 2.2: Règle de Cauchy et de Curry

Remark 19 *Ces deux règles ne sont utilisées que dans des cas particuliers, par exemple lorsque h_k est quadratique. Le mot exact prend sa signification dans le fait que si f est quadratique la solution de la recherche linéaire s'obtient de façon exacte et dans un nombre fini d'itérations.*

Remark 20 *Dans la plupart des algorithmes d'optimisation modernes, on ne fait jamais de recherche linéaire exacte, car trouver α_k signifie qu'il va falloir calculer un grand nombre de fois la fonction h_k et cela peut être dissuasif du point de vue du temps de calcul. En pratique, on recherche plutôt une valeur de α^*qui assure une décroissance suffisante de f. Cela conduit à la notion d'intervalle de sécurité.*

Definition 21 (Intervalle de sécurité) *On dit que $[\alpha_g, \alpha_d]$ est un intervalle de sécurité s'il permet de classer les valeurs de α de la façon suivante :*

- *Si $\alpha < \alpha_g$ alors α est considéré trop petit,*
- *Si $\alpha_d \geq \alpha \geq \alpha_g$ alors α est satisfaisant,*
- *Si $\alpha > \alpha_d$ alors est considéré trop grand. Le problème est de traduire de façon numérique sur h_k les trois conditions précédentes, ainsi que de trouver un algorithme permettant de déterminer α_g et α_d.*

Recherches linéaires inexactes Au lieu de demander que α_k minimise h_k, on préfère imposer des conditions moins restrictives, plus facilement vérifiées, qui permettent toute fois de contribuer à la convergence des algorithmes. En particulier, il n'y aura plus un unique pas (ou quelques pas) vérifiant ces conditions mais tout un intervalle de pas (ou plusieurs intervalles), ce qui rendra d'ailleurs leur recherche plus aisée. C'est ce que l'on fait avec les règles d'Armijo, de Goldstein et de Wolfe.

Règle d'Armijo La règle d'Armijo se base sur le choix d'un paramètre

$$0 < \rho < 1$$

et détermine une valeur approchée de α_k par la condition :

$$h_k(\alpha) \leqslant h_k(0) + \rho\alpha h_k'(0).$$

Le risque de cette méthode est de favoriser les valeurs trop petites, aussi, elle est rarement utilisée seule.

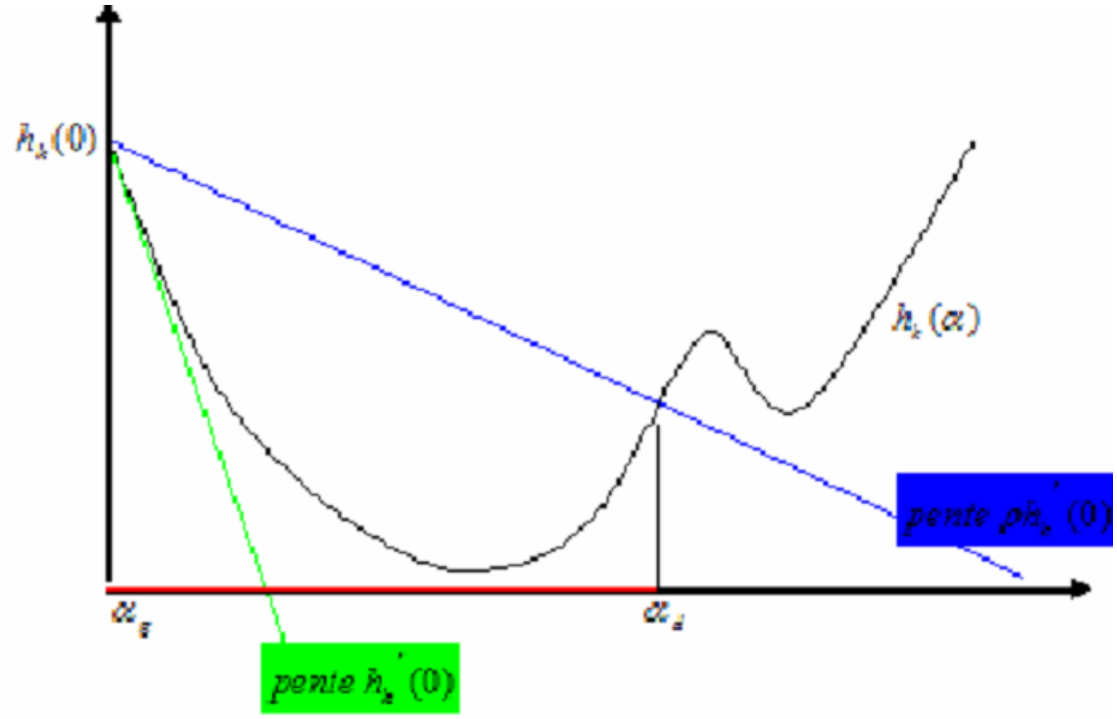

FIGURE 2.3: Règle d'Armijo

Règle de Goldstein Goldstein propose en 1967 une méthode basée sur le choix cette fois-ci de deux paramètres

$$0 < \rho < \delta < 1$$

et détermine les valeurs approchées de α_k par deux conditions :

$$\begin{cases} h_k(\alpha) & \leqslant h_k(0) + \rho\alpha h_k'(0), \\ h_k(\alpha) & \geqslant h_k(0) + \delta\alpha h_k'(0). \end{cases}$$

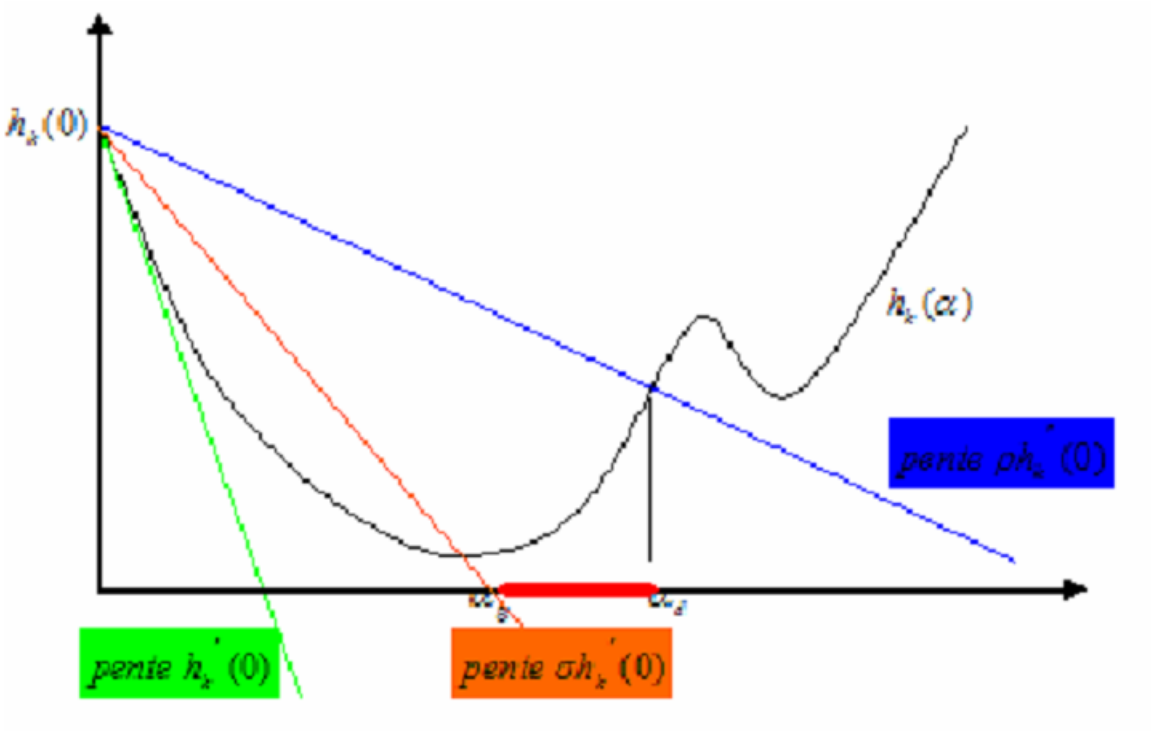

FIGURE 2.4: Règle de Goldstein

Règle de Wolfe Wolfe propose en 1969 une méthode basée sur le choix de deux paramètres

$$0 < \rho < \delta < 1$$

et détermine les valeurs approchées de α_k par deux conditions :

$$h_k(\alpha) \leqslant h_k(0) + \rho\alpha h_k'(0), \tag{2.2}$$

$$h_k'(\alpha) \geqslant \delta h_k'(0). \tag{2.3}$$

Deux valeurs usuelles des paramètres sont $\rho = 0,1$ et $\delta = 0,7$.

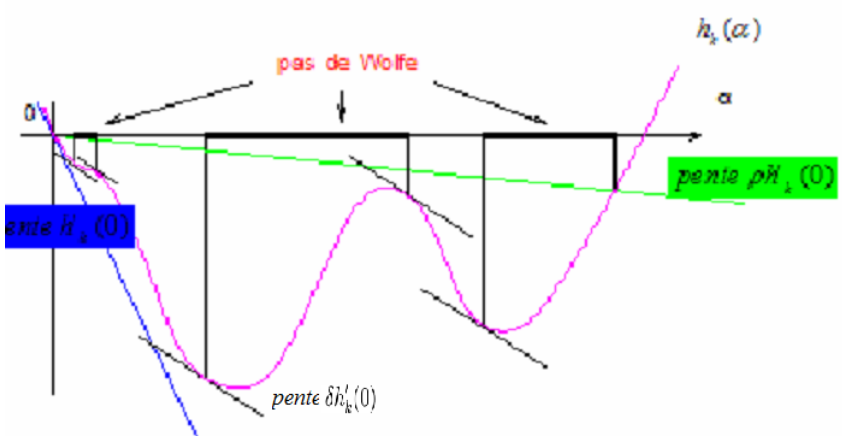

FIGURE 2.5: Règle de Wolfe

Remark 22 *La règle de Wolfe fait appel au calcul de h'_k, elle est donc en théorie plus couteuse que la règle de Goldstein. Cependant dans de nombreuses applications, le calcul du gradient $\nabla f(x)$ représente un faible cout additionnel en comparaison du cout d'évaluation de $f(x)$, c'est pourquoi cette règle est très utilisée.*

Remark 23 *Pour certains algorithmes (par exemple le gradient conjugué non linéaire) il est parfois nécessaire d'avoir une condition plus restrictive que (2.3).Pour cela la deuxième condition (2.3) est remplacée par :*

$$|h'_k(\alpha)| \leq -\delta h'_k(0). \tag{2.4}$$

On aurra donc les conditions de Wolfe fortes [1971] :((2.2)et (2.4))

2.2 Convergence et vitesse de convergence d'un algorithme à directions de descente

2.2.1 Convergence

Étudier la convergence d'un algorithme, c'est étudier la convergence de la suite des itérés générés par l'algorithme. Un algorithme de descente selon le modèle précédent, est dit convergent si la suite de ses itérés

$$(x_k)_{k\in\mathbb{N}}$$

converge vers un point limite x_*, solution du problème :

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

De plus, la convergence est dite locale si elle n'a lieu que pour des points initiaux x_0 dans un voisinage de x_*. Sinon elle est dite globale.

En pratique, le but d'un algorithme d'optimisation ne sera que de trouver un point critique (i.e. un point vérifiant la condition d'optimalité du premier ordre :

$$\nabla f(x_*) = 0).$$

On introduit alors la notion de convergence globale d'un algorithme d'optimisation :

Definition 24 *Soit un algorithme itératif qui génère une suite*

$$(x_k)_{k\in\mathbb{N}}$$

dans $\mathbb{R}^n$ *afin de résoudre le problème :*

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

où

$$f : \mathbb{R}^n \to \mathbb{R}$$

est une application de classe C^1*. L'algorithme est dit globalement convergent si quel que soit le point initial* $x_0 \in \mathbb{R}^n$ *,*

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

Cette propriété garantit que le critère d'arrêt :

$$\|\nabla f(x_k)\| \leq \epsilon$$

sera satisfait à partir d'un certain rang quelle que soit la précision $\epsilon > 0$ *demandée.*

Remark 25 *Attention, la notion de convergence que l'on a vu ne suppose pas que l'algorithme converge vers un minimum, même un minimum local.*

2.2.2 Vitesse de convergence

Il est bien entendu très important de garantir la convergence d'un algorithme sous certaines hypothèses, mais la vitesse de convergence et la complexité sont également des facteurs à prendre en compte lors de la conception ou de l'utilisation d'un algorithme ; en effet, on a tout intérêt à ce que la méthode choisie soit à la fois rapide, précise et stable. Pour cela, on introduit les notions de vitesse (ou taux) de convergence qui mesurent l'évolution de l'erreur commise

$$\|x_k - x_*\|.$$

Definition 26 *Soit*

$$(x_k)_{k\in\mathbb{N}},$$

une suite d'itérer générées par un algorithme convergent donné. On note x_* *la limite de la suite*

$$(x_k)_{k\in\mathbb{N}},$$

et on suppose :

$$\forall k \in \mathbb{N} \quad x_k \neq x_*,$$

(sinon l'algorithme convergerait en un nombre fini d'itérations). La convergence de l'algorithme est dite :

- *linéaire si l'erreur*

$$e_k = \|x_k - x_*\|$$

décroît linéairement i.e. s'il existe

$$\tau \in]0, 1[,$$

tel que :

$$\lim_{k\to+\infty} \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} = \tau,$$

- *super linéaire si :*

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

- *d'ordre* p *s'il existe* $\tau \geq 0$ *tel que :*

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

En particulier, si $p = 2$*, la convergence est dite quadratique (grosso modo a partir d'un certain rang, le nombre de chiffres significatifs exacts double à chaque itération).*

Bien entendu, on a intérêt à ce que la convergence d'un algorithme soit la plus élevée possible afin de converger vers la solution en un minimum d'itérations pour une précision donnée.

2.3 Contribution de la recherche linéaire inexacte à la convergence des algorithmes à directions de descente

On va étudier la contribution de la recherche linéaire inexacte à la convergence des algorithmes à directions de descente. Ce n'est qu'une contribution, parce que

la recherche linéaire ne peut à elle seule assurer la convergence des itérés. On comprend bien que le choix de la direction de descente joue aussi un rôle. Cela se traduit par une condition, dite de Zoutendijk[17], dont on peut tirer quelques informations qualitatives intéressantes.

Definition 27 (La condition de Zoutendijk) *On dit qu'une règle de recherche linéaire satisfait la condition de Zoutendijk s'il existe une constante $C > 0$ telle que pour tout indice $k \geq 1$ on ait*

$$f(x_{k+1}) \leq f(x_k) - C\|\nabla f(x_k)\| cos^2\theta_k, \tag{2.5}$$

où θ_k est l'angle que fait d_k avec $-\nabla f(x_k)$, défini par

$$cos\theta_k = ((-[\nabla f(x_k)]^T d_k)/(\|\nabla f(x_k)\|\|d_k\|)).$$

Voici comment on se sert de la condition de Zoutendijk.

Proposition 28 *Si la suite (x_k) générée par un algorithme d'optimisation vérifie la condition de Zoutendijk (2.5) et si la suite $(f(x_k))_{k\geq 1}$ est minorée, alors*

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

Proof. En sommant les inégalités (2.5), on a

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

La série est donc convergente puisqu'il existe une constant C' telle que pour tout k, $f(x_k) \geq C'$. ■

La proposition suivante précise les circonstances dans les quelles la condition de Zoutendijk (2.5) est vérifie avec la règle de Wolfe.

Proposition 29 *Soit*

$$f : \mathbb{R}^n \to \mathbb{R}$$

une fonction continument différentiable dans un voisinage de

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

On considère un algorithme à directions de descente d_k, qui génère une suite (x_k) en utilisant la recherche linéaire de Wolfe (2.2)-(2.3). Alors il existe une constante $C > 0$ telle que, pour tout $k \geq 1$, la condition de Zoutendijk (2.5) est vérifiée.

Proof. D'après (2.3)

$$\begin{aligned}
\nabla^T f(x_k + \alpha_k d_k) d_k \geq \delta \nabla^T f(x_k) d_k \Rightarrow \\
(\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k))^T d_k \geq (\delta - 1) \nabla^T f(x_k) d_k \Rightarrow \\
-(1-\delta)\nabla^T f(x_k) d_k = (1-\delta)|\nabla^T f(x_k) d_k| \Leftrightarrow \\
(1-\delta)|\nabla^T f(x_k) d_k| \leq (\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k))^T d_k.
\end{aligned}$$

Et du fait que f est continument différentiable :

$$\begin{aligned}
(1-\delta)|\nabla^T f(x_k) d_k| &= (1-\delta)\|\nabla^T f(x_k)\|\|d_k\| cos\theta_k \\
&\leq \|\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k)\|\|d_k\| \Rightarrow \\
(1-\delta)\|\nabla^T f(x_k)\| cos\theta_k &\leq L\alpha_k \|d_k\| \Rightarrow \\
\alpha_k \|d_k\| &\leq ((1-\delta)/L)\|\nabla^T f(x_k)\| cos\theta_k.
\end{aligned}$$

en utilisant (2.2), on aura :

$$\begin{aligned}
f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k \nabla^T f(x_k) d_k \Rightarrow \\
f(x_k + \alpha d_k) \leq f(x_k) + |\rho\alpha \nabla^T f(x_k) d_k| \Rightarrow \\
f(x_k + \alpha d_k) \leq f(x_k) - \rho\alpha \nabla^T f(x_k) d_k \Rightarrow \\
f(x_k + \alpha d_k) \leq f(x_k) - \rho\alpha \|\nabla^T f(x_k)\|\|d_k\| cos\theta_k \Rightarrow \\
f(x_k + \alpha d_k) \leq f(x_k) - ((\rho(1-\delta))/L)\|\nabla^T f(x_k)\|^2 cos^2\theta_k.
\end{aligned}$$

On en déduit (2.5) ■

Lemma 30 *Supposons que :*

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

est continûment différentiable. Soit d une direction de descente au point x_k, et supposons que f est bornée inférieurement le long du rayon

$$\{x_k + \alpha d_k, \quad \alpha > 0\}.$$

Ensuite, si

$$0 < \rho < \delta < 1,$$

alors l'ensemble des pas vérifiant la règle de Wolfe (faible et forte) est non vide.

Proof. Supposent que :

$$\begin{aligned}
h_k(\alpha) = f(x_k + \alpha d_k), \\
\psi_\rho(\alpha) = f(x_k) + \rho\alpha \nabla^T f(x_k) d_k.
\end{aligned}$$

Le développement de Taylor-Yong en $\alpha = 0$ de h_k est :

$$h_k(\alpha) = f(x_k + \alpha d_k) = f(x_k) + \rho\alpha\nabla^T f(x_k)d_k + \alpha\xi(\alpha),$$

avec

$$\lim_{\alpha\to 0} \xi(\alpha) = 0,$$

et comme

$$0 < \rho < 1$$

et

$$h_k^{'}(0) = \nabla^T f(x_k)d_k < 0,$$

on déduit :

$$f(x_k) + \alpha\nabla^T f(x_k)d_k < f(x_k) + \rho\alpha\nabla^T f(x_k)d_k, \quad \alpha > 0.$$

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

$$h_k(\alpha) < \psi_\rho(\alpha).$$

De ce qui précède et du fait que h_k est bornée inférieurement,et

$$\lim_{\alpha\to+\infty} \psi_\rho(\alpha) = -\infty,$$

on déduit que la fonction $\psi_\rho(\alpha) - h_k(\alpha)$ a la propriété :

$\begin{cases} \psi_\rho(\alpha) - h_k(\alpha) & > 0 \quad \text{pour } \alpha \text{ assez petit} \\ \psi_\rho(\alpha) - h_k(\alpha) & < 0 \quad \text{pour } \alpha \text{ assez grand} \end{cases}$ donc s'annule au moins une fois pour $\alpha > 0$.

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

$$h_k(\bar{\alpha}) = \psi_\rho(\bar{\alpha}) \quad \text{et} \quad h_k(\alpha) < \psi_\rho(\alpha) \quad \text{pour} \quad 0 < \alpha < \bar{\alpha}. \qquad (*)$$

La formule des accroissements finis fournit alors un nombre

$$\hat{\alpha}, \quad 0 < \hat{\alpha} < \bar{\alpha},$$

tel que :

$$\begin{aligned} h_k(\bar{\alpha}) - h_k(0) &= \bar{\alpha}h_k^{'}(\hat{\alpha}) \\ &= \bar{\alpha}\nabla^T f(x_k + \hat{\alpha}d_k)d_k \\ &\Rightarrow \rho\bar{\alpha}\nabla^T f(x_k)d_k = \bar{\alpha}\nabla^T f(x_k + \hat{\alpha}d_k)d_k \\ &\Rightarrow \nabla^T f(x_k + \hat{\alpha}d_k)d_k = \rho\nabla^T f(x_k)d_k \geq \delta\nabla^T f(x_k)d_k \end{aligned}$$

car

$$0 < \rho < \delta < 1 \quad \text{et} \quad \nabla^T f(x_k) d_k < 0.$$

Donc $\hat{\alpha}$ satisfait :

$$h_k(\hat{\alpha}) \leqslant h_k(0) + \rho \hat{\alpha} h'_k(0),$$

et

$$h'_k(\hat{\alpha}) \geqslant \delta h'_k(0),$$

en effet, $\hat{\alpha}$ satisfait $(*)$ n'est autre que :

$$f(x_k + \hat{\alpha} d_k) < f(x_k) + \rho \hat{\alpha} \nabla^T f(x_k) d_k.$$

Ce qu'il fallait démontrer. ■

Remark 31 *En pratique, on utilise des algorithmes spécifiques pour trouver un pas de Wolf.*

Chapitre 3

La méthode du gradient conjugué

3.1 La méthode du gradient conjugué linéaire

En analyse numérique, la méthode du gradient conjugué est un algorithme pour résoudre des systèmes d'équations linéaires dont la matrice est symétrique définie positive. Cette méthode, imaginée en 1950 simultanément par Cornelius Lanczos et Magnus Hestenes, est une méthode itérative qui converge en un nombre fini d'itérations (au plus égal à la dimension du système linéaire)[14]. Toutefois, son grand intérêt pratique du point de vue du temps de calcul vient de ce qu'une initialisation astucieuse (dite « pré conditionnement ») permet d'aboutir en seulement quelques passages à une estimation très proche de la solution exacte, c'est pourquoi en pratique on se borne à un nombre d'itérations bien inférieur au nombre d'inconnues.

3.1.1 La méthode du gradient conjugué linéaire vue comme une méthode directe

On rappelle que deux vecteurs non nuls u et v sont conjugués par rapport à A si

$$u^{\mathrm{T}}\mathbf{A}v = 0.$$

Sachant que A est symétrique définie positive, on en déduit un produit scalaire

$$\langle u, v\rangle_{\mathbf{A}} := \langle \mathbf{A}u, v\rangle = \langle u, \mathbf{A}^{\mathrm{T}}v\rangle = \langle u, \mathbf{A}v\rangle = u^{\mathrm{T}}\mathbf{A}v.$$

Deux vecteurs sont conjugués s'ils sont donc orthogonaux pour ce produit scalaire.

La conjugaison est une relation symétrique : si u est conjugué à v pour A, alors v est conjugué à u.

Supposons que (d_k) est une suite de n directions conjuguées deux à deux. Alors les d_k forment une base de $\mathbb{R}^n$, ainsi la solution x_* de $Ax = b$ dans cette base :

$$x_* = \sum_{i=1}^{n} \alpha_i d_i$$

Les coefficients sont donnés par

$$b = \mathbf{A}x_* = \sum_{i=1}^{n} \alpha_i \mathbf{A} d_i.$$

$$d_k^{\mathrm{T}} b = d_k^{\mathrm{T}} \mathbf{A} x_* = \sum_{i=1}^{n} \alpha_i d_k^{\mathrm{T}} \mathbf{A} d_i = \alpha_k d_k^{\mathrm{T}} \mathbf{A} d_k.$$

(car $\forall i \neq k, d_i, d_k$ sont conjugués deux à deux)

$$\begin{aligned} \alpha_k &= \frac{d_k^{\mathrm{T}} b}{d_k^{\mathrm{T}} \mathbf{A} d_k} \\ &= \frac{\langle d_k, b \rangle}{\langle d_k, d_k \rangle_{\mathbf{A}}} \\ &= \frac{\langle d_k, b \rangle}{\|d_k\|_{\mathbf{A}}^2}. \end{aligned}$$

On a ainsi l'idée directrice de la méthode pour résoudre le système

$$Ax = b$$

: trouver une suite de n directions conjuguées, et calculer les coefficients α_k.

3.1.2 La méthode du gradient conjugué linéaire vue comme une méthode itérative

En choisissant correctement les directions conjuguées d_k, il n'est pas nécessaire de toutes les déterminer pour obtenir une bonne approximation de la solution x_*. Il est ainsi possible de considérer la méthode du gradient conjugué comme une méthode itérative. Ce choix permet ainsi de considérer la résolution de systèmes de très grande taille, où le calcul de l'ensemble des directions aurait été très long.

On considère ainsi un premier vecteur x_0, qu'on pourra supposer nul (sinon, il faut considérer le système $Az = b - Ax_0$). L'algorithme va consister, partant de x_0, à se « rapprocher » de la solution x_* inconnue, ce qui suppose la définition d'une

métrique. Cette métrique vient du fait que la solution x_* est l'unique minimiseur de la forme quadratique :

$$f(x) = \frac{1}{2}x^T \mathbf{A}x - b^T x, \quad x \in \mathbb{R}^n.$$

Ainsi, si $f(x)$ diminue après une itération, alors on s'approche de x_*.

Ceci suggère donc de prendre la première direction d_1 comme l'opposé du gradient de f à $x = x_0$. Le gradient vaut

$$Ax_0 - b = -b,$$

d'après notre première hypothèse. Les vecteurs suivants de la base seront ainsi conjugués au gradient, d'où le nom « méthode du gradient conjugué ».

Soit r_k le résidu à la k^{em} itération :

$$r_k = b - \mathbf{A}x_k. \tag{3.1}$$

Notons que r_k est l'opposé du gradient de f en $x = x_k$, ainsi, l'algorithme du gradient indique d'évoluer dans la direction r_k. On rappelle que les directions d_k sont conjuguées deux à deux. On veut aussi que la direction suivante soit construite à partir du résidu courant et des directions précédemment construites, ce qui est une hypothèse raisonnable en pratique.

La contrainte de conjugaison est une contrainte d'orthonormalisation, aussi le problème partage des similitudes avec le procédé de Gram-Schmidt.

On a ainsi

$$d_{k+1} = r_k - \sum_{i \leq k} \frac{d_i^{\mathrm{T}} \mathbf{A} r_k}{d_i^{\mathrm{T}} \mathbf{A} d_i} d_i$$

Suivant cette direction, le point suivant est donné par

$$x_{k+1} = x_k + \alpha_{k+1} d_{k+1}$$

avec

$$\alpha_{k+1} = \frac{d_{k+1}^{\mathrm{T}} b}{d_{k+1}^{\mathrm{T}} \mathbf{A} d_{k+1}} = \frac{d_{k+1}^{\mathrm{T}} (r_k + \mathbf{A}x_k)}{d_{k+1}^{\mathrm{T}} \mathbf{A} d_{k+1}} = \frac{d_{k+1}^{\mathrm{T}} r_k}{d_{k+1}^{\mathrm{T}} \mathbf{A} d_{k+1}}, \tag{3.2}$$

la dernière égalité venant du fait que d_{k+1} et x_k sont conjugués.

3.1.3 Algorithme de La méthode du gradient conjugué linéaire

Cette méthode est basée sur la génération (très économique) de directions conjuguées. Si l'on note

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

Cet algorithme consiste à générer une suite d'itérés (x_k) sous la forme :

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

L'idée de la méthode est :

a- construire itérativement des directions $d_0, ..., d_k$ mutuellement conjuguées : La méthode du gradient conjugué est obtenue en appliquant la procédure de Gram-Schmidt aux gradients

$$\{\nabla f(x_0), ..., \nabla f(x_{n-1})\}.$$

En outre, nous avons que :

$$\nabla f(x) = Ax - b \quad \text{et} \quad \nabla^2 f(x) = A$$

notons que la méthode se termine si

$$\nabla f(x) = 0.$$

La particularité intéressante de la méthode du gradient conjugué est que le membre de droite de l'équation donnant la valeur de d_{k+1} dans la procédure de Gram-Schmidt peut être grandement simplifié. Notons que la méthode du gradient conjugué est inspirée de celle du gradient (plus profonde pente). A chaque étape $k$la direction d_k est obtenue comme combinaison linéaire du gradient en x_k et de la direction précédente d_{k-1} c-a-d :

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

les coefficients β_{k+1}étant choisis de telle manière que d_k soit conjuguée avec toutes les directions précédentes. autrement dit :

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

on en déduit :

$$d_{k+1}^T A d_k = 0 \Rightarrow (-\nabla f(x_{k+1})) + \beta_{k+1} d_k)^T A d_k = 0 \Rightarrow$$

$$-\nabla^T f(x_{k+1}) A d_k + \beta_{k+1} d_k^T A d_k = 0 \Rightarrow$$

$$\boxed{\beta_{k+1} = \frac{\nabla^T f(x_{k+1}) A d_k}{d_k^T A d_k} = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}.}$$

b-déterminer le pas α_k : en particulier, une façon de choisir α_k peut être de résoudre le problème d'optimisation (à une seule variable)

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

on en déduit :

$$\boxed{\alpha_k = \frac{-d_k^T g_k}{d_k^T A d_k}.}$$

Le pas α_k obtenu ainsi s'appelle le pas optimal.

3.1.4 Convergence de la méthode de gradient conjugué linéaire

Nous avons vu que la méthode de gradient conjugué terminera en au plus n itérations (supposant des calculs numériques exactes.) Ce qui est plus remarquable, est que si les valeurs propres de A ont une distribution favorable, l'algorithme trouvera la solution en moins de n itérations.

Theorem 32 *Si A a seulement r valeurs propres distinctes, alors l'itération de GC terminera à la solution en au plus r itérations. C'est à dire, si A est proche de la matrice d'identité, $A = I + B$ avec $rank(B) = r$ alors la performance de l'algorithme est améliorée. Ce fait inspire des méthodes de pré-conditionnement.*

Proof. Soient

$$\tau_1 < ... < \tau_n,$$

les valeurs distinctes des valeurs 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 à voir que $Q_r(\lambda) - 1$ est un polynôme de degré r avec une racine $\lambda = 0$.

Définissons ensuite $\tilde{P}_{r-1}$, un polynôme de degré $(r-1)$ par :

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

En posant $k = r - 1$ dans

$$\min_{\tilde{P}_k} \max_{1 \le i \le n} \left[1 + \lambda_i \tilde{P}_k(\lambda_i)\right]_2,$$

nous obtenons :

$$0 \leq \min_{\tilde{P}_k} \max_{1 \leq i \leq n} \left[1 + \lambda_i \tilde{P}_k(\lambda_i)\right]_2 \leq \max_{1 \leq i \leq n} \left[1 + \lambda_i \tilde{P}_k(\lambda_i)\right]_2 = \max_{1 \leq i \leq n} Q_r(\lambda_i) = 0.$$

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

Theorem 33 *Si A a des valeurs propres*

$$\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n,$$

on a :

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

Remark 34 *Le théorème 3.1.2 nous permet de prédire le comportement de la méthode du gradient conjugué. Supposons par exemple que les valeurs propres de A consistent en m grandes valeurs propres et les $n - m$ valeurs propres restantes sont situées autour de 1.*

Si nous définissons

$$\epsilon = \lambda_{n-m} - \lambda_1,$$

le théorème 3.1.2 nous dit qu'après $m+1$ pas de la méthode du gradient conjugué, nous avons :

$$\|x_{m+1} - x_*\|_A \approx \epsilon \|x_0 - x_*\|_A.$$

Pour une petite valeur de ϵ après $m+1$ pas, la méthode va nous donner une bonne approximation de la solution. Mais attention, le théorème 3.1.2 nous donne une borne supérieure. Il se peut que la méthode nous donne déjà des bons résultats après les premières itérations. Il est généralement vrai que si les valeurs propres apparaissent en r groupes, alors la méthode du gradient conjugué résout approximativement le problème après r pas. Une autre expression de convergence pour la méthode du gradient conjugué est celle basée sur le nombre de condition spectral. Celui-ci est défini par :

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

On arrive ensuite à montrer que :

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

Cette expression est plus approximative et donne souvent d'importantes surestimations de l'erreur. Mais en réalité, souvent, on n'a pas beaucoup d'informations sur A et ses valeurs propres. Cette dernière méthode ne demande que les valeurs propres extrêmes ou des approximations de celles-ci.

3.1.5 Le gradient conjugué linéaire pré conditionné

Dès que le nombre n de variables dépasse quelques dizaines ou centaines (cela dépend du conditionnement de A), l'algorithme du GC peut avoir des difficultés à minimiser la fonction quadratique et, du fait des erreurs d'arrondi, il peut demander beaucoup plus de n itérations pour avoir une solution acceptable. Pour remédier à cette situation, il faut essayer d'améliorer le conditionnement du problème ou encore de"pré conditionner" l'algorithme. Nous présentons ci-après l'approche qui utilise un changement de variables. Si on fait le changement de variables

$$\tilde{x} = Lx$$

au moyen d'une matrice L d'ordre n inversible, la fonction

$$f : \mathbb{R}^n \to \mathbb{R},$$

définie par :

$$\tilde{f} = f \circ L^{-1}$$

est telle que $\tilde{f}(\tilde{x}) = f(x)$. Les fonctions $\tilde{f}$ et f atteignent donc la même valeur minimale et ce aux points x_* et

$$\tilde{x}_* = Lx_*,$$

respectivement. L'algorithme du GC pré conditionné s'obtient en appliquant l'algorithme du GC standard dans l'espace des $\tilde{x}$ et en traduisant l'algorithme résultant dans l'espace des x. L'algorithme est ainsi modifié, amélioré si le changement de variables est bien choisi ; on dit aussi qu'il est sensible à un changement de variables. Le gradient et le hessien de $\tilde{f}$ en $\tilde{x} = Lx$ s'écrivent :

$$\tilde{g} \equiv \nabla \tilde{f}(\tilde{x}) = L^{-T} \nabla f(x) \quad \text{et} \quad \tilde{A} \equiv \nabla^2 \tilde{f} = L^{-T} A L^{-1}.$$

Soient $(\tilde{x}_k)$ la suite générée par le GC dans l'espace des $\tilde{x}$ et

$$x_k := L^{-1} \tilde{x}_k.$$

On note

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

et

$$\tilde{g}_k = \nabla \tilde{f}(x_k),$$

si bien que

$$\tilde{g}_k = L^{-T} g_k.$$

La direction de la méthode du gradient conjugué dans l'espace des $\tilde{x}$ s'écrit :

$$\tilde{d}_k = -\tilde{g}_k + \tilde{\beta}_k \tilde{d}_{k-1} = -L^{-T} g_k + \frac{\| - L^{-T} g_k \|^2}{\| - L^{-T} g_{k-1} \|^2} \tilde{d}_{k-1}.$$

Ramenée à l'espace des x, cela donne pour

$$d_k = L^{-1} \tilde{d}_k,$$

la formule :

$$d_k = \begin{cases} -P g_1 & \text{si} \quad k = 1, \\ -P g_k + \dfrac{g_k^T P g_k}{g_{k-1}^T P g_{k-1}} d_{k-1} & \text{si} \quad k \geq 2, \end{cases} \tag{3.3}$$

où :

$$\boxed{P = L^{-1} L^{-T}}.$$

On montre que g_k est encore orthogonal à d_{k-1}, si bien que le pas optimal le long de d_k est donné par la formule :

$$\alpha_k = \frac{g_k^T P g_k}{d_k^T A d_k}. \tag{3.4}$$

Si on remplace la direction d_k et le pas α_k de la méthode du gradient conjugué par les valeurs données par les formules (3.3) et (3.4), on obtient la méthode du gradient conjugué pré conditionné. Le choix de L, donc de P, est gouverné par le souhait d'avoir $\tilde{A}$ proche de la matrice unité, ce qui a pour effet d'accélérer la convergence de l'algorithme dans l'espace des $\tilde{x}$, donc aussi dans l'espace des x. On voit qu'il est souhaitable de prendre L proche de $A^{\frac{1}{2}}$, ou encore

$$\boxed{P \simeq A^{-1}}.$$

Remark 35 *La technique du pré conditionnement peut aussi être envisagée pour les algorithmes GCNL. Le choix de l'inverse du Hessien comme pré conditionneur entraine pour les algorithmes GCNL un taux de convergence super linéaire lorsque le pas optimal est utilisé, ce qui correspond au taux de convergence de la méthode de Newton [Al-Baali et Fletcher, 1996].*

Lors de l'utilisation d'un pré conditionnement il est important de distinguer deux notions :

– Le taux de convergence quantifie l'efficacité d'un algorithme en terme de nombre d'itérations.

– La vitesse de convergence expérimentale est une mesure globale d'efficacité, en temps de calcul.

Pré conditionneurs pratiques Comme si souvent en numérique, il n'existe pas une meilleure stratégie de pré conditionnement valable pour tout type de matrices. Des pré conditionneurs généraux ont bien été proposés, mais leur efficacité varie fortement d'un problème à l'autre. Les stratégies les plus importantes de ce genre sont la SSOR (symmetric successive overrelaxation), la méthode de Cholesky incomplète et les pré conditionneurs par bande.

3.1.6 Les avantages de la méthode du gradient conjugué linéaire

1- la consommation mémoire de l'algorithme est minime : on doit stocker les quatre vecteurs x_k, g_k, d_k, Ad_k(bien sur x_{k+1} prend la place de x_k au niveau de son calcul avec des remarques analogues pour g_{k+1}, d_{k+1}, Ad_{k+1}) et les scalaires α_k, β_{k+1}.

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 itérations.

3.2 La méthode du gradient conjugué non linéaire

Afin de minimiser une fonction non linéaire, la méthode de gradient conjugué doit être modifiée. D'abord nous faisons une recherche linéaire afin de calculer le pas $\alpha_k$qui minimise la fonction non-linéaire f le long de d_k. Deuxièmement, le résidu r doit être remplacé par le gradient de f. Nous obtenons ainsi un algorithme très efficace pour l'optimisation non-linéaire[15]. Si f est strictement convexe et si α_k est le minimiseur exacte, alors cet algorithme est exactement le gradient conjugué linéaire [26] . Le paramètre α_k doit vérifier certaines conditions afin d'assurer que la direction d_{k+1} soit une direction de descente. Si la recherche linéaire est exacte, nous voyons que :

$$\nabla f(x_k)^T d_k < 0$$

Mais si la recherche n'est pas exacte (parce que la recherche exacte est trop coûteuse), il faut imposer d'autres conditions .

L'idée de la méthode est de construire itérative-ment des directions $d_0, ..., d_k$ mutuellement conjuguées. A chaque étape k la direction d_k est obtenue comme combinaison linéaire du gradient en x_k et de la direction précédente d_{k-1}, les

coefficients étant choisis de telle manière que d_k soit conjuguée avec toutes les directions précédentes. On s'intéresse ici à la minimisation d'une fonction

$$f : \mathbb{R}^n \to \mathbb{R},$$

non nécessairement quadratique :

$$\boxed{\min_{x\in\mathbb{R}^n} f(x) \qquad \text{(P)}}$$

et on cherche à étendre la méthode du gradient conjugué a ce problème. Il y a plusieurs manières de le faire et peu de critères permettant de dire laquelle est la meilleure. Une extension possible consiste simplement a reprendre les formules utilisées dans le cas quadratique[13]. On se propose donc d'étudier les méthodes oú la direction d_k est définie par la formule de récurrence suivante :

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

ou $g_k = \nabla f(x_k)$, $\beta_k \in \mathbb{R}$ et x_k est générée par la formule :

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

le pas $\alpha_k \in \mathbb{R}$ étant déterminé par une recherche linéaire.

Remark 36 *Ces méthodes sont des extensions de la méthode du gradient conju-*

guée si β_k prend l'une des valeurs :

$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$	*Proposé par Hestenes et Stiefel[1952].*
$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$	*(NCG), proposé par Fletcher et reeves[1964].*
$\beta_k^{D} = \frac{g_{k+1}^T \nabla^2 f(x_k) d_k}{d_k^T \nabla^2 f(x_k) d_k}$	*(NCG), proposé par Daniel[1967].*
$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{g_k^T g_k}$	*(NCG), proposé par Polak, Ribiere et Polyak[1969].*
$\beta_k^{PRP^+} = \max\{0, \frac{g_{k+1}^T y_k}{g_k^T g_k}\}$	*(NCG), proposé par Powell[1984].*
$\beta_k^{CD} = \frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}$	*(NCG), proposé par Flétcher[1987].*
$\beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}$	*(NCG), proposé par Lieu et Story[1991].*
$\beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{d_k^T g_k}$	*(NCG), proposé par Dai et Yuan[1999]*
$\beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T g_k}, \quad t > 0$	*(NCG), proposé par Dai et Lieu[2001].*
$\beta_k^{YT} = \frac{g_{k+1}^T (z_k - t s_k)}{d_k^T z_k}$	*(NCG), proposé par Yabe et Takano[2004].*

la òu :

$$z_k = y_k + \frac{\rho v_k}{s_k^T u_k} u_k, \quad \rho \geq 0 \quad \textit{et} \quad u_k \in [0,1],$$

et

$$v_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k.$$

Remark 37 *Pour que les méthodes ainsi définies soient utilisables, il faut répondre aux deux questions suivantes :*

- *Les directions d_k sont-elles des directions de descente de f ?*
- *Les méthodes ainsi définies sont-elles convergentes ?*

En ce qui concerne la première question remarquons que, quel que soit

$$\beta_k \in \mathbb{R},$$

d_k est une direction de descente si on fait de la recherche linéaire exacte, c'est-a-dire si le pas α_{k-1} est un point stationnaire de :

$$\alpha \to f(x_{k-1} + \alpha d_{k-1}).$$

Proof. En effet, dans ce cas

$$g_k^T d_{k-1} = 0,$$

et on trouve lorsque

$$g_k \neq 0 :$$

$$d_k^T g_k = (-g_k + \beta_k d_{k-1})^T g_k = -\|g_k\|^2 + \beta_k d_{k-1}^T g_k = -\|g_k\|^2 < 0.$$

■

Cependant, il est fortement déconseillé de faire de la recherche linéaire exacte lorsque f n'est pas quadratique : le coût de détermination de α est excessif.

Remark 38 *Nous avons noté que la méthode CG, peut être considérée comme un algorithme de minimisation pour la fonction quadratique convexe. Il est naturel de se demander si nous pouvons adapter l'approche de minimiser des fonctions non linéaires[12]. En fait, comme nous le montrons dans ce que suit , les variantes du gradient conjugué non linéaires sont bien étudiés et se sont révélés être très efficace dans la pratique[16].*

3.2.1 La méthode de Fletcher-Reeves

Fletcher et Reeves ont montré comment étendre la méthode du gradient conjugué à des fonctions non-linéaires en faisant deux changements simples . Le premier, à la place de la formule (3.2) pour la longueur de l'étape α_k (qui minimise h_k le long de la direction de la recherche d_k), nous avons besoin d'effectuer une recherche linéaire qui identifie un minimum approximatif de la fonction f non linéaire le long d_k . Deuxièmement, le résidu r_k dans (3.1) doit être remplacé par le gradient de l'objectif f non linéaire[18]. Ces modifications donnent lieu à l'algorithme suivant pour l'optimisation non linéaire.

Algorithm 39 (FR)

Given x_0;

Evaluate $f_0 = f(x_0), \quad \nabla f_0 = \nabla f(x_0)$;
Set $d_0 \longleftarrow \nabla f_0, \quad k \longleftarrow 0$;
w h i l e $\nabla f_k \neq 0$
Compute α_k *and set* $x_{k+1} = x_{_k} + \alpha_k d_k$;
Evaluate ∇f_{k+1};

$$\beta_{k+1}^{FR} \longleftarrow \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k};$$

$$d_{k+1} \longleftarrow -\nabla f_{k+1} + \beta_{k+1}^{FR} d_k;$$

$$k \longleftarrow k + 1;$$

end(while)

Remark 40 *Si nous choisissons f comme une fonction quadratique fortement convexe et α_k le minimiseur exacte, cette algorithme se réduit à la méthode du gradient conjugué linéaire, et il est attrayant pour les grands problèmes d'optimisation non linéaire parce que à chaque itération il exige seulement l'évaluation de la fonction objectif et de son gradient. Aucune opération de matrice sont nécessaires pour le calcul de l'étape, et on a quelques vecteurs de stockage sont nécessaires.*

Pour rendre la spécification de cet algorithme complète, nous devons être plus précis sur le choix de α_k. En raison que le d_k peut manquer d'être une direction de descente à moins que α_k satisfait à certaines conditions. Donc on a :

$$\nabla f_k^T d_k = -\|\nabla f_k\|^2 + \beta_k^{FR} \nabla f_k^T d_{k-1} \tag{3.5}$$

Si la recherche linéaire est exacte, de sorte que α_{k-1} est un minimiseur local de f le long du direction d_k , nous avons cela :

$$\nabla f_k^T d_{k-1} = 0,$$

alors d_k est une direction de descente .

Si la recherche linéaire n'est pas exact, le terme a droite dans (3.5) peut dominer l premier terme, et nous pouvons avoir

$$\nabla f_k^T d_k > 0$$

qui implique que le d_k est réellement n'est pas une direction de descente . Heureusement, nous pouvons éviter cette situation en exigeant α_k satisfaire les conditions de Wolfeforte.

Comportement de la méthode Fletcher-Reeves

Nous étudions maintenant l'algorithme Fletcher-Reeves .

Le résultat suivant donne des conditions sur la recherche linéaire en vertu de laquelle toutes les directions de recherche sont des directions de descente[23] .Il suppose que l'ensemble

$$\mathcal{L} := \{x | f(x) \leq f(x_0)\}$$

est bornée ; où x_0 est le point de départ de l'itération et que f est deux fois continûment différentiable, de sorte que nous avons du lemme 3.1 qu'il existe une longueur de pas satisfaisant aux conditions de Wolfe forte.

3.2.2 La méthode de Polak-Ribière Polyak

Cette méthode a été proposée en 1969 par Polak et Ribière et Polyak . Il y a beaucoup de variantes entre cette méthode et la méthode de Fletcher-Reeves, principalement dans le choix du paramètre β_k, proposée par Polak et Ribiere Polyak, ce paramètre est définit comme suit :

$$\beta_{k+1}^{PRP} = \frac{\nabla f_{k+1}^T(\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}.$$

Remark 41 *Une fois appliqué les deux méthodes aux fonctions non-linéaires avec la recherche linéaire inexacte, le comportement des deux algorithmes diffère nettement. Les expériences numérique indique que le P.R.P tend à être le plus robuste et efficace que F.R*

Remark 42 *Un fait étonnant sur l'algorithme P.R.P est que les conditions fortes de Wolfe ne peux pas garantir que le d_k est toujours une direction de descente.*

Modification de la méthode PRP en agissant sur le coefficient β_k

Dai a suggéré la modification suivante dans le paramètre de mise à jour pour la méthode PRP :

$$\beta_{k+1}^{PRP^+} = \max\{\beta_{k+1}^{PRP}, \quad 0\}.$$

Gilbert et Nocedal ont prouvé la convergence de la méthode PRP^+.

3.2.3 Convergence de la méthode de gradient conjugué non linéaire

À la différence de la méthode de gradient conjugué linéaire , dont les propriétés de convergence sont bien comprises et qui est connue pour être optimale comme décrit ci-dessus, les méthodes de gradient conjugué non-linéaire possèdent des propriétés de convergence parfois bizarre et étonnantes . Nous présentons maintenant quelques uns de résultats connus pour Fletcher-Reeves et de Polak-Ribiére employant la recherche linéaire pratique. Pour ça , nous faisons les hypothèses (non limitatifs) suivants sur la fonction objectif.

Hypothèses 3

(i) L'ensemble

$$\mathcal{L} := \{x | f(x) \leq f(x_0)\},$$

est bornée ; où x_0 est le point de départ de l'itération.

(ii) Dans un voisinage $\mathcal{N}$ ouvert de $\mathcal{L}$, la fonction objectif f est lipschitzien continûment différentiables.

Ces hypothèses impliquent qu'il y a une constante$L$ tels que :

$$\|\nabla f(x)\| \leq L, \quad \forall x \in \mathcal{L}.$$

Notre principal outil d'analyse dans cette section est le théorème théorème de Zoutendijk . Il affirme que, sous les Hypothèses précédentes , si la suite (x_k) générée par un algorithme d'optimisation vérifie la condition de Zoutendijk , alors :

$$\sum_{k\geq 1} \|\nabla f_k\|^2 cos^2\theta_k < \infty \tag{3.6}$$

Nous pouvons utiliser ce résultat[22] pour prouver la convergence globale pour les algorithmes qui sont périodiquement renouvelées par la mise de $\beta_k = 0$. Si

$$k_1, k_2, ...$$

désignent les itérations sur lequel se redémarre l'algorithme alors d'après (3.6) on a :

$$\sum_{k_1,k_2...} \|\nabla f_k\|^2 < \infty. \tag{3.7}$$

Si nous permettons pas plus de n itérations entre les redémarrages, la suite

$$(k_j)_{j\geq 1}$$

est infinie, et de (3,7) nous avons :

$$\lim_{k\to\infty} \|\nabla f_{k_j}\| = 0.$$

Autrement dit, une sous-suite de gradients approche de zéro, ou de manière équivalente,

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0.$$

Ce résultat vaut également pour les versions renouvelées de tous les algorithmes décrits dans ce chapitre. Il est plus intéressant, cependant, pour étudier la convergence globale des méthodes de gradient conjuguées, parce que pour les grands problèmes (disent $n \geq 1000$) nous nous attendons de trouver une solution dans beaucoup moins de n itérations[21].

Nous pouvons prouver un résultat de convergence globale pour la méthode de Fletcher-Reeves. Bien que nous ne pouvons pas montrer que la limite de la suite des gradients (∇f_k) est égal à zéro, le résultat suivant montre que cette suite ne soit pas délimitée loin de zéro

Theorem 43 (AL-Baali) *Supposons que les (Hypothèses 3) sont satisfaites , et que l'algorithme de Fletcher-Reeves est mis en œuvre avec une recherche linéaire qui satisfait les les conditions de Wolfe fortes , avec*

$$0 < \rho < \delta < \frac{1}{2}.$$

Alors :

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0.$$

Remark 44 *Le premier théorème démontrant la descente d'une méthode du gradient conjugué non linéaire a été établi par Albaali (1985) pour la méthode de Fletcher-Reeves avec la recherche forte de Wolfe avec*

$$\delta < \frac{1}{2}.$$

Gilbert et Nocedal [1992] ont généralisé ce résultat pour tout algorithme du gradient conjugué dont

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

Aucun théorème n'a était établi afin de démontrer la satisfaction de la propriété de descente pour la méthode de Polak-Ribière-Polyak non linéaire. Grippo etLucidi [1997] et P.Armand [2005] ont suggéré des modifications dans le choix de α_k afin d'établir le résultat de la convergence, ainsi la propriété de descente. Fletcher [1987] a démontré que la méthode de la descente conjuguée est une méthode de descente si le pas α_k est déterminé par la règle forte de Wolfe avec

$$\delta < \frac{1}{2}$$

. Dai et Yuan [1996] ont démontré que cette méthode avec la règle de Wolfe relaxée, génère des directions de descente suffisante à chaque itération $k \geq 1$. Dai etYuan [1998] ont démontré qu'à chaque itération $k \geq 1$, la direction recherchée par la méthode de Dai-Yuan avec la recherche de Wolfe faible , est de descente si la fonction objectif f est strictement convexe.

3.2.4 Condition de conjugaison

Si la direction initiale d_0 est choisie de tel sorte

$$d_0 = -g_0,$$

et la fonction objective f à minimiser est une fonction quadratique convexe c.a.d :

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c, \tag{3.8}$$

et la recherche linéaire exact est employé c.a.d :

$$\alpha_k = arg \min_{\alpha>0} f(x_k + \alpha d_k), \tag{3.9}$$

alors la condition de conjugaison est :

$$d_i^T A d_j = 0 \tag{3.10}$$

est satisfaite pour tous $i \neq j$ cette relation (3.10) est la condition originale de conjugaison employé par Hestenes et Stiefel pour dériver les algorithmes de gradient conjugué, principalement pour résoudre les systèmes des équations linéaires définie-positives.

On employant (3.8),(3.9)et (3.10)on montre que x_{k+1} est le minimum de la fonction quadratique (3.8) est dans le sous-espace :

$$x_k + span\{g_1, g_2, ...g_k\}$$

et les gradients $g_1, g_2, ...g_k$ sont mutuellement orthogonaux à moins que $g_k = 0$.

Il suit de cela que pour des fonctions quadratiques et convexes la solution sera trouvée a n itérations au plus [27] .

Powell a prouvé que si la direction initiale d_0 n'est pas $-g_0$ les algorithmes de gradient conjugué ne se termine pas dans un nombre fini d'itérations pour la fonction (3.8).

Il est bien connu que l'algorithme du gradient conjugué converge au moins linéairement[24] .

Une limite supérieure pour le taux de convergence des algorithmes de gradient conjugué a été donnée par Yuan .

Dénotons :

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

Pour une fonction f non-linéaire deux fois différentiable,on a par le théorème de valeur moyenne, il existe certains

$$\nu \in (0.1)$$

tels que :

$$d_{k+1}^T y_k = \alpha_k d_{k+1}^T \nabla^2 f(x_k + \nu\alpha_k d_k) d_k. \tag{3.11}$$

Par conséquent, il semble raisonnable de remplacer (3.10)par la condition suivante :

$$d_{k+1}^T y_k = 0. \tag{3.12}$$

Pour accélérer l'algorithme de gradient conjugué Perry (voir également Shanno) a prolongé la condition de conjugaison en incorporant l'information du second degré.

Il a employé l'état sécant

$$H_{k+1}y_k = s_k,$$

où H_k est une approximation symétrique de l'inverse de la matrice hessienne.

Puisque pour la méthode quasi-newtonienne[11] la direction d_{k+1} est calculée par

$$d_{k+1} = -H_{k+1}g_{k+1},$$

donc :

$$d_{k+1}^T y_k = (-H_{k+1}g_{k+1})\, y_k = -g_{k+1}^T\, (H_{k+1}y_k) = -g_{k+1}^T s_k,$$

de ce fait on obtient une nouvelle condition de conjugaison.

Récemment, Dai et Liao ont prolongée cette condition ils ont donné la nouvelle condition de conjugaison comme suit :

$$d_{k+1}^T y_k = -\sigma g_{k+1}^T s_k, \tag{3.13}$$

là où σ est un scalaire positif.

Chapitre 4

APPLICATION DE LA TECHNIQUE DE SYMETRIE DE POWELL AUX METHODES DU GRADIENT CONJUGUE AVEC LA GENERALISATION DE CONDITION DE CONJUGAISON

4.1 Introduction

Les méthodes du gradient conjugué non linéaire(NCG) sont utilisées pour résoudre les problèmes d'optimisation non linéaires sans contraintes et spécialement les problèmes de grande taille :

$$\min_{x\in\mathbb{R}^n} f(x). \tag{4.1}$$

Les méthodes du gradient conjugué classique avec recherche linéaire sont définies comme suit :

$$x_{k+1} = x_k + \alpha_k d_k, \tag{4.2}$$

α_k est la longueur d'étape d'une recherche linéaire et d_k donné par

$$\begin{cases} d_0 &= -g_0, \\ d_{k+1} &= -g_{k+1} + \beta_k d_k, \quad \forall k > 0, \end{cases} \tag{4.3}$$

$g_k = g(x_k) = \bigtriangledown f(x_k)$ et β_k est une grandeur scalaire. Afin de garantir la convergence globale de (NCG) , les proprietés suivantes sont importantes et néces-

saires :,[1]

$$d_{k+1}^T g_{k+1} < 0 \qquad \text{(propriété de descente)} \tag{4.4}$$

$$d_{k+1}^T g_{k+1} \leq -c_0 \mid\mid g_{k+1} \mid\mid^2, \quad c_0 > 0 \quad \text{(propriété de descente suffisante)}. \tag{4.5}$$

Cependant, à la différence des méthodes quasi-newtoniennes en général, les méthodes(NCG) n'assurent pas la descente ou la propriété de descente suffisante pour la recherche linéaire inexacte. Beaucoup d'efforts ont été consacrés à l'étude de la propriété de descente. En 1978, Shanno a proposé les méthodes quasi-newtoniennes à mémoire limitée et des algorithmes de gradient conjugué basés sur les idées de Perry , l'equation quasi-newton et la technique de variable métrique d'auto-graduation. En 1990, Touati-Ahmed et Storey ont combiné l'algorithme de Fletcher-Reeves et l'algorithme de Polak-Ribière et ont avancé plusieurs algorithmes hybrides de gradient conjugué. En 1991, Liu et Storey ont développé la méthode de gradient conjugué généralisé (appelée l'algorithme de LS). Récemment, Hager et Zhang ont proposé l'algorithme de CG.-DESCENTE avec c = 7/8 dans (1.5). Wei et Al ont présenté quelques nouvelles formules pour β_k dans (1.3). Andrei a présenté les algorithmes du gradient conjugué mesurés, basé sur les méthodes spectrales du gradient conjugué et l'idée de Shanno . En 2008, Andrei a suggéré un nouvel algorithme hybride du gradient conjugué. En appliquant la technique de symétrie pour des méthodes du gradient conjugué, on se propose de présenter ici une méthode symétrique du gradient conjugué, satisfaisant la propriété (1.5) pour n'importe quelle recherche linéaire ; et cette idée peut également être appliquée à d'autres algorithmes du gradient conjugué.[2]

4.2 Application de la technique de symétrie de Powell au méthode du gradient conjugué

Selon la notation de Perry [3], on s'intéresse dans cette section à la forme de β_k de Hestenes-Stiefel suivante : [4]

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

les d_{k+1}, seront donnés par :

$$d_{k+1} = -D_{k+1} g_{k+1}, \tag{4.6}$$

with,

$$D_{k+1} = (I - \frac{d_k y_k^T}{d_k^T y_k}) = (I - \frac{s_k y_k^T}{s_k^T y_k}), \tag{4.7}$$

$s_k = x_{k+1} - x_k = \alpha_k d_k$, $y_k = g_{k+1} - g_k$ et la matrice D_{k+1} s'appelle *la matrice d'itération du gradient conjugué*. La matrice D_{k+1} peut ne pas être symétrique définie positive , donc d_{k+1}peut ne pas satisfaire la propriété de descente . Pour garantir la propriété de descente, la technique de symétrie [2] appliquée à D_{k+1} donne :

Premièrement, la suite $\{C_k\}$ est présentée comme suit :

$$\begin{cases} C_1 &= I - \dfrac{dy^T}{d^T y}, \\ \\ C_2 &= \dfrac{C_1 + C_1^T}{2} \end{cases} \qquad and \qquad \begin{cases} C_{2k+1} &= C_{2k} - \dfrac{dy^T}{d^T y} C_{2k}, \\ \\ C_{2k+2} &= \dfrac{C_{2k+1} + C_{2k+1}^T}{2}. \end{cases}$$

Posons :$E_k = C_{2k}$, $u_k = E_k y$ et $u_0 = E_0 y = Iy$, donc,

$$E_{k+1} = E_k - \frac{1}{2}\frac{du_k^T + u_k d^T}{d^T y}$$

and

$$u_{k+1} = E_k y - \frac{1}{2}\frac{du_k^T y + u_k d^T y}{d^T y} = (\frac{dy^T}{d^T y})u_k = Du_k = D^{k+1}u_0,$$

$D = \dfrac{1}{2}(\dfrac{dy^T}{d^T y})$. Évidemment, les valeurs propres de D sont 0 et $\dfrac{1}{2}$ (n - 1 multiplicité), donc :

$$\sum_{k=0}^{\infty} u_k = \sum_{k=0}^{\infty} D^k u_0 = (I - D)^{-1} u_0 = 2(I - \frac{1}{2}\frac{dy^T}{d^T y})y = 2Iy - \frac{dy^T}{d^T y}y.$$

Alors,

$$\sum_{k=0}^{\infty}(E_{k+1} - E_k) = -\frac{1}{2}\sum_{k=0}^{\infty}\frac{du_k^T + u_k d^T}{d^T y} = -\frac{1}{d^T y}(yd^T + dy^T - \frac{yy^T}{d^T y}dd^T). \quad (4.8)$$

Donc $\lim_{x\to\infty} C_k$ existe, notons cette limite par C. De (4.8), il s'ensuit :

$$C = \lim_{x\to\infty} C_k = \lim_{x\to\infty} E_k = E_0 + \sum_{k=0}^{\infty}(E_{k+1} - E_k)$$

$$= I - \frac{yd^T + dy^T}{d^T y} + \frac{y^T y}{(d^T y)^2}dd^T = \left(I - \frac{dy^T}{d^T y}\right)\left(I - \frac{yd^T}{d^T y}\right).$$

Alors, la matrice C_1 peut être symétrisée par la matrice C. Puis, du procédé de symétrisation ci-dessus, on conclut que la matrice d'itération D_{k+1} du gradient conjugué peut être symetrisée par $\bar{D}_{k+1}$ comme suit :

$$\bar{D}_{k+1} = I - \frac{y_k d_k^T + d_k y_k^T}{d_k^T y_k} + \frac{y_k^T y_k}{(d_k^T y_k)^2} d_k d_k^T = \left(I - \frac{d_k y_k^T}{d_k^T y_k}\right)\left(I - \frac{y_k d_k^T}{d_k^T y_k}\right). \quad (4.9)$$

Ainsi, les directions du gradient conjugué (4.3) sont re-écrites :

$$\begin{cases} d_0 &= -g_0, \\ d_{k+1} &= -\bar{D}_{k+1} g_{k+1}, \quad \forall k > 0. \end{cases} \quad (4.10)$$

Donc d_{k+1} est nommé *la direction symétrique du gradient conjugué* et$\bar{D}_{k+1}$ est nommée *la matrice d'itération symétrique du gradient conjugué (SHS matrix)* . Si $\bar{D}_{k+1}$ est mis à jour avec la matrice de rang 1, comme suit :

$$\hat{D}_{k+1} = \bar{D}_{k+1} + \frac{s_k \varsigma_k^T}{y_k^T \varsigma_k}, \qquad \forall \varsigma_k \in \mathbb{R}^n$$

alors $\hat{D}_{k+1}$ satisfait l'équation quasi-Newton , $\hat{D}_{k+1} y_k = s_k$,et avec la recherche linéaire exact , $d_{k+1} = -\hat{D}_{k+1} g_{k+1}$ satisfait la condition :

$$y_k^T d_{k+1} = 0, \quad \forall k \geq 0 \quad (\text{conjugacy}). \quad (4.11)$$

If $\varsigma_k = s_k$, d'où

$$d_{k+1}^{mBFGS} = -\left(\bar{D}_{k+1} + \frac{s_k s_k^T}{y_k^T s_k}\right) g_{k+1},$$

qui est juste la formule de la direction de recherche $m - BFGS$.

Dans cette thèse , (4.11) est remplacée par[5]

$$y_k^T d_{k+1} = -\sigma s_k^T g_{k+1}, \quad laconditiondeconjugaisongnralise \quad (4.12)$$

Maintenant on suppose que $\bar{D}_{k+1}$ dans (4.10) être mis à jour par une matrice de rang un, donnée par

$$\bar{P}_{k+1} = \bar{D}_{k+1} + uv^T,$$

ou u et v deux vecteurs de $\mathbb{R}^n$ tel que (4.12) détient.
Par (4.9), (4.12) et $d_{k+1} = -\bar{P}_{k+1} g_{k+1}$ on a :

$$\begin{aligned} y_k^T d_{k+1} &= -y_k^T \left(\bar{D}_{k+1} g_{k+1} + uv^T g_{k+1}\right) \\ &= -y_k^T \left(I - \frac{d_k y_k^T}{d_k^T y_k}\right)\left(I - \frac{y_k d_k^T}{d_k^T y_k}\right) g_{k+1} - y_k^T uv^T g_{k+1} \\ &= -\left(y_k^T - y_k^T\right)\left[\left(I - \frac{y_k d_k^T}{d_k^T y_k}\right) g_{k+1}\right] - y_k^T uv^T g_{k+1} \\ &= -y_k^T uv^T g_{k+1} \\ &= -\left(y_k^T u\right) v^T g_{k+1}, \end{aligned}$$

so,

$$-\left(y_k^T u\right) v^T g_{k+1} = -\sigma s_k^T g_{k+1} \Rightarrow \left(\sigma s_k - v y_k^T u\right)^T g_{k+1} = 0.$$

Ainsi, nous pouvons sélectionner v tel que : $v = \dfrac{\sigma s_k}{y_k^T u}$. Par conséquent

$$\bar{P}_{k+1} = \bar{D}_{k+1} + \sigma \frac{u s_k^T}{y_k^T u}, \tag{4.13}$$

u est un vecteur de $\mathbb{R}^n$ tel que $y_k^T u \neq 0$. La matrice $\bar{P}_{k+1}$ est nommée *la matrice de Hestenes-Stiefel symétrique*. Ainsi, nous pouvons introduire une nouvelle direction comme suit :

$$\begin{cases} d_0 &= -g_0, \\ d_{k+1} &= -\bar{P}_{k+1} g_{k+1}, \quad \forall k > 0, \end{cases} \tag{4.14}$$

$\bar{P}_{k+1}$ est définie par (13) avec $u = u_k$, i.e.,

$$d_{k+1} = -\bar{D}_{k+1} g_{k+1} - \sigma \frac{u_k s_k^T}{y_k^T u_k} g_{k+1}. \tag{4.15}$$

Dans cette thèse, nous prenons $u_k = y_k$ et $\sigma = c\dfrac{y_k^T y_k}{s_k^T y_k} \quad , \quad c > 0$. Alors

$$\bar{P}_{k+1} = \bar{D}_{k+1} + c \frac{y_k s_k^T}{s_k^T y_k},$$

et,

$$d_{k+1} = -\bar{D}_{k+1} g_{k+1} - c \frac{d_k^T g_{k+1}}{d_k^T y_k} y_k. \tag{4.16}$$

On note le schéma itératif (4.2) et (4.14) avec d_{k+1} calculé par (4.16) par **GDSHS**.

4.3 La propriété de descente suffisante et algorithme de descente

Dans cette section, nous considérons la propriété de descente suffisante de la méthode**GDSHS** , elle est,

$$d_{k+1}^T g_{k+1} \leq -c_0 \mid\mid g_{k+1} \mid\mid^2, \quad c_0 > 0.$$

Selon le théorème (2.1) dans [2] on a $\bar{D}_{k+1}$ est une matrice semi-définie positive et ses valeurs propres sont 0, 1(n - 2 multiplicity) et λ_{max}^{k+1} , respectivement, λ_{max}^{k+1} est la valeur propre maximale :

$$\lambda_{max}^{k+1} = \frac{\mid\mid y_k \mid\mid^2 \mid\mid d_k \mid\mid^2}{(d_k^T y_k)^2}.$$

Par (4,16), on obtient ce qui suit :

$$d_{k+1}^T g_{k+1} = -g_{k+1}^T \bar{D}_{k+1} g_{k+1} - c\frac{y_k^T g_{k+1}}{d_k^T y_k} d_k^T g_{k+1},$$

depuis ,

$$g_{k+1}^T \bar{D}_{k+1} g_{k+1} \geq 0,$$

donc,

$$d_{k+1}^T g_{k+1} \leq -c\frac{y_k^T g_{k+1}}{d_k^T y_k} d_k^T g_{k+1} \frac{d_k^T y_k}{d_k^T y_k} \quad \Rightarrow$$

$$d_{k+1}^T g_{k+1} \leq -c\frac{((g_{k+1}^T y_k) d_k)^T ((d_k^T y_k) g_{k+1})}{(d_k^T y_k)^2}.$$

De l'inégalité suivante,

$$u^T v \leq \frac{1}{2}(a \mid\mid u \mid\mid^2 + \frac{1}{a} \mid\mid v \mid\mid^2), \quad \forall a > 0,$$

on peut en déduire que,

$$\begin{aligned} d_{k+1}^T g_{k+1} &\leq -\frac{c}{2(d_k^T y_k)^2}(a \mid\mid (g_{k+1}^T y_k) d_k \mid\mid^2 + \frac{1}{a} \mid\mid (d_k^T y_k) g_{k+1} \mid\mid^2) \\ &\leq -\frac{c}{2(d_k^T y_k)^2}(a(g_{k+1}^T y_k)^2 \mid\mid d_k \mid\mid^2 + \frac{1}{a}(d_k^T y_k)^2 \mid\mid g_{k+1} \mid\mid^2) \\ &\leq -\frac{ca \mid\mid g_{k+1} \mid\mid^2 \mid\mid y_k \mid\mid^2 \mid\mid d_k \mid\mid^2}{2(d_k^T y_k)^2} - \frac{c}{2a} \mid\mid g_{k+1} \mid\mid^2 . \end{aligned}$$

Alors,

$$d_{k+1}^T g_{k+1} \leq -(ca\frac{\mid\mid y_k \mid\mid^2 \mid\mid d_k \mid\mid^2}{2(d_k^T y_k)^2} + \frac{c}{2a}) \mid\mid g_{k+1} \mid\mid^2 .$$

Ainsi, (4.5) est vraie pour

$$c_0 = ca\frac{\mid\mid y_k \mid\mid^2 \mid\mid d_k \mid\mid^2}{2(d_k^T y_k)^2} + \frac{c}{2a}.$$

De discussion ci-dessus, nous avons,

$$d_{k+1} = -\bar{D}_{k+1} g_{k+1} - c\frac{d_k^T g_{k+1}}{d_k^T y_k} y_k.$$

Ainsi, les directions du gradient conjugué (4.14) sont réécrites en tant que :

$$\begin{cases} d_0 &= -g_0, \\ d_{k+1} &= -\bar{P}_{k+1} g_{k+1} = -v_{k+1} + \beta_k d_k - c\zeta_k y_k, \end{cases} \tag{4.17}$$

où,

$$v_{k+1} = t_k g_{k+1} + (1 - t_k)g_k = g_{k+1} - (1 - t_k)y_k = g_k + t_k y_k,$$

$$\beta_k = \frac{v_{k+1}^T y_k}{d_k^T y_k} = t_k \frac{g_{k+1}^T y_k}{d_k^T y_k} + (1 - t_k)\frac{g_k^T y_k}{d_k^T y_k},$$

$$\zeta_k = \frac{d_k^T g_{k+1}}{d_k^T y_k},$$

et,

$$t_k = \frac{-d_k^T g_k}{d_k^T y_k}.$$

Ainsi, nous pouvons obtenir *the Generalized Descent Symmetrical Hestenes-Stiefel algorithm*, noté **GDSHS**, comme suit :

Algorithm 3.1

step 1. Give an initial point x_0 and $\varepsilon \geq 0$. Set $k = 0$.

step 2. Calculate $g_0 = g(x_0)$. If $|| \ g_k \ || \leq \varepsilon$, then stop ; otherwise let $d_0 = -g_0$ and continue with **step 3.**.

step 3. Calculate steplength α_k with Wolfe line searches :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta_1 \alpha_k d_k^T g_k \tag{4.18}$$

and,

$$d_k^T g(x_k + \alpha_k d_k) \geq \delta_2 d_k^T g_k, \tag{4.19}$$

where δ_1 and δ_2 are positive constants such that

$$0 < \delta_1 < \delta_2 < 1.$$

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

step 5. Calculate $g_{k+1} = g(x_{k+1})$.

step 6. If $|| \ g_{k+1} \ || \leq \varepsilon$, then stop.

step 7. Calculate the direction d_{k+1} via (4.17). Set $k = k + 1$, then go to **step 3.**.

Pour la démonstration numérique dans Sect. 5, un algorithme de béton est le suivant :

GDSHS1 : la direction d_{k+1} est calculée par (4.17) et $c = 1$.

4.4 La convergence de l'algorithme GDSHS

Dans cette section, pour analyser la convergence de l'algorithme **GDSHS**, first, nous présentons les hypothèses suivantes sur la fonction objective $f(x)$.

H1. f est minorée dans $\mathbb{R}^n$ et f est continument différentiable dans un domaine $\aleph$ de l'ensemble de niveau $\Gamma \stackrel{\text{def}}{=} \{x : f(x) \leq f(x_0)\}$, ou x_0 est le point de départ de l'itération.

H2. Le gradient de f est Lipschitz continu dans $\aleph$, autrement dit, il existe une constante $L > 0$ tel que :

$$|| \nabla f(\bar{x}) - \nabla f(x) ||\leq L || \bar{x} - x ||, \qquad \forall \bar{x}, x \in \aleph.$$

Ensuite, nous introduisons la condition théorème spectral de la convergence globale pour une fonction objectif satisfaisant H1 et H2, qui génère le théorème 4.1 dans [2].

Theorem 4.4.1 *Supposons que la direction de recherche linéaire d'une méthodedu gradient conjugué non linéaire satisfait*

$$\begin{cases} d_0 &= -g_0, \\ d_k &= -\bar{P}_k g_k, \quad \forall k > 0. \end{cases} \tag{4.20}$$

*Soit la fonction objectif $f(x)$ satisfont **H1** et **H2**. Pour un procédé du gradient conjugué non linéaire((4,2) et (4,20)), qui satisfait à la condition de descente suffisante (4.5), si sa recherche linéaire satisfait les conditions de Wolfe (4.18) et (4.19), et [7]*

$$\sum_{k=0}^{\infty} \Lambda_k = +\infty, \quad (\textit{the spectral condition}) \tag{4.21}$$

tel que Λ_k est la valeur propre maximale de $\bar{P}_k^T \bar{P}_k$, alors :

$$\lim_{x \to \infty} inf \; || \; g_k \; || = 0. \tag{4.22}$$

Par ailleurs, si $\Lambda_k \leq \tilde{\Lambda}$,ou $\tilde{\Lambda}$ est une constante positive, alors

$$\lim_{x \to \infty} || \; g_k \; || = 0 \tag{4.23}$$

Proof. Supposons que,

$$g_k \neq 0, \quad \forall k \geq 0$$

et,

$$\lim_{x \to \infty} inf \; || \; g_k \; || \neq 0,$$

alors il existe $\gamma > 0$ tel que,

$$|| \; g_k \; || > \gamma, \quad \forall k \geq 0,$$

et la condition de descente suffisante (4,5) implique que,

$$d_k \neq 0, \quad \forall k \geq 0.$$

A partir de (4,20) et le fait que $\bar{P}_k^T \bar{P}_k$ est symétrique et semi-définie positive, il résulte que :

$$|| \, d_k \, ||^2 = g_k^T \bar{P}_k^T \bar{P} g_k = || \, \bar{P}_k^T \bar{P} \, || \, || \, g_k \, ||^2 \leq \Lambda_k \, || \, g_k \, ||^2 \, .$$

Ainsi, à partir de (4,5) et l'inégalité ci-dessus, on peut en déduire que

$$\begin{aligned} \cos^2 \theta_k &= \frac{\left(-d_k^T g_k\right)^2}{|| \, d_k \, ||^2 || \, g_k \, ||^2} \\ &\geq c_0^2 \frac{|| \, g_k \, ||^2}{|| \, d_k \, ||^2} \\ &\geq \frac{c_0^2}{\Lambda_k}, \end{aligned}$$

où θ_k est l'angle entre d_k et $-g_k$. Ainsi,

$$\sum_{k \geq 0} || \, g_k \, ||^2 \cos^2 \theta_k \geq \gamma^2 \sum_{k \geq 0} \frac{c_0^2}{\Lambda_k} = \infty,$$

ce qui contredit à la condition de Zoutendijk dans[17],

$$\sum_{k \geq 0} || \, g_k \, ||^2 \cos^2 \theta_k = \sum_{k \geq 0} \frac{(g_k^T d_k)^2}{|| \, d_k \, ||^2} < \infty \quad \text{(the Zoutendijk's condition)}. \tag{4.24}$$

Par conséquent, (4.21) implique que $g_k = 0$ pour certains $k > 0$, ou (4,22) détient.

Si $\Lambda_k \leq \tilde{\Lambda}$ suffisamment grande k, alors $\cos^2 \theta_k \geq \dfrac{c_0^2}{\tilde{\Lambda}_k} > 0$, qui, conjointement avec (4.24) implique (4,23). ■

4.5 Expérience numérique

Dans cette section, nous comparons les performances de la nouvelle méthode du gradient conjugué GDSHS avec le paramètre $c = 1$, noté GDSHS1, aux méthodes FR standard et PRP^+ . Les problèmes d'essai sont les 86 problèmes sans contrainte, chaque fonction de test est effectué a une expérience avec le nombre de variables 2, 10, 100, 1000, 2000,. . ., 3500, respectivement. Les points de départ

utilisés sont ceux donnés dans [Andrei, N. 2008]. Le critère de fin de tous les algorithmes est que $\mid \ midg_k \ \mid\mid < 10^{-7}$. Les tests sont effectués sur *PC, Pentium Dual-core T4400@2.20GHz CPU, RAM 2.0GB, Mobil Intel 4 Series Express Chipset Family*, en utilisant des codes MATLAB. Nous avons adopté Dolan et Moré [18] profils de performance pour comparer les performances entre les méthodes éprouvées. Figure 4.1, 4.2 et 4.3 sont le profil de performance mesurée par le temps CPU (ou temps de traitement), le nombre d'itérations et de la 2-norme du gradient de la fonction objective au point minimale approximative, respectivement.

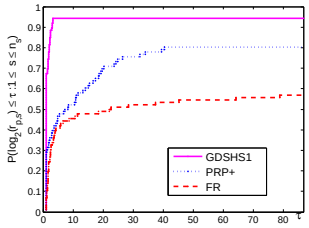

FIGURE 4.1: profil de performance par le temps (CPU)

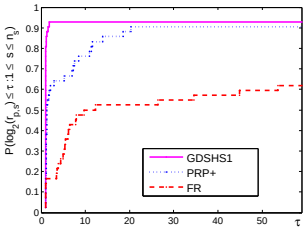

FIGURE 4.2: profil de performance par nombre d'itérations

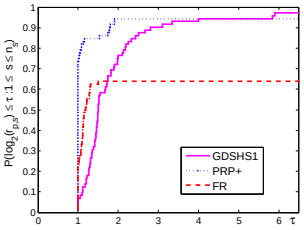

FIGURE 4.3: Profil de performance en 2-norme du gradient de la fonction objective

4.6 Remarques finales

Dans cette thèse, nous avons proposé une nouvelle méthode du gradient conjugué (Algorithme 3.1) qui produit toujours une direction de recherche de descente et Sect. 4, nous avons prouvé la propriété globale de convergence de notre méthode en utilisant l'analyse spectrale du gradient conjugué et de la condition de Zoutendijk. Certains résultats numériques ont été rapportés. Ces résultats montrent l'efficacité de notre méthode si nous choisissons un bon paramètre c. En perspective, nous proposons d'étudier une valeur optimale pour elle. Le choix théorique de ce paramètre est actuellement à l'étude. Le profil de performance pour notre algorithme de gradient conjugué ((4.2) et (4.20)), mis en œuvre avec notre nouvel algorithme , était plus élevé que ceux de la FR et PRP^+ méthodes pour un ensemble de test constitué de 86 problèmes dans [19].

Bibliographie

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

[2] LIU, D., XU, G., *Applying Powells symmetrical technique to conjugate gradient methods*, Computational Optimization and Applications, DOI 10. 1007/s10589-009-9302-1. (2009)

[3] PERRY, A., *A modified conjugate gradient algorithm*, Oper. Res. Tech. Notes 26(6), 1073–1078 (1978)

[4] HESTENES, M. R. , STIEFEL, E., *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand. 49(6), 409-439 (1952)

[5] DAI, Y. H., LIAO, L. Z., *An efficient hybrid conjugate gradient method for unconstrained optimization* , Ann. Oper. Res., 103 (2001) 33-47.

[6] BARZILAI, J. , BORWEIN, J. M., *Two point step size gradient methods*, IMA J. Numer. Anal. 8, 141-148 (1988)

[7] BIRGIN, E. G. , MARTÃNEZ, J. M., *A spectral conjugate gradient method for unconstrained optimization. Appl*, Math. Optim. 43, 117-128 (2001)

[8] BUCKLEY, A., *Extending the relationship between the conjugate gradient and BFGS algorithms*, Math. Program. 15, 343-348 (1978)

[9] GILBERT, J. C. , NOCEDAL, J., *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim. 2(1), 21-42 (1992)

[10] HAGER, W. W. , ZHANG, H. C., *A new conjugate gradient method with guaranteed descent and an efficientline search*, SIAM J. Optim. 16, 170-192 (2005)

[11] LIU, Y. , STOREY, C., *Efficient generalized conjugate gradient algorithms*, part 1, theory. J. Optim. Theory Appl. 69, 129-137 (1991)

[12] NOCEDAL, J. , WRIGHT, S. J., *Numerical Optimization*, 2nd edn. Springer, Berlin (2006)

[13] HAGER, W. W., ZHANG, H., *A survey of nonlinear conjugate gradient methods*, Pac. J. Optim. 2, 35-5 (2006b)

[14] HESTENES, M. R., STIEFEL, E., *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49(6), 409-439 (1952)

[15] WEI, Z. X. , LI, G. Y. , QI, L. Q., *New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems*, Appl. Math. Comput. 179, 407-430 (2006)

[16] YU, G. H. , ZHAO, Y. L. , WEI, Z. X., *A descent nonlinear conjugate gradient method for large-scale un-constrained optimization*, Appl. Math. Comput. 187, 636-643 (2007)

[17] ZOUTENDIJK, G., *Nonlinear Programming, Computational Methods*, In J. Abadie (Eds.), Integer and Nonlinear Programming (pp. 37-86). Amsterdam : North-Holland. (1970)

[18] DOLAN, E. D. , MORÃ©, J. J., *Benchmarking optimization software with performance profiles*, Math. Program. Ser. A 91, 201-213 (2002)

[19] ANDREI, N., *An Unconstrained Optimization Test Functions Collection*, Advanced Modeling and Optimization, Volume 10, Number 1, 2008

[20] ANDREI, N., *An unconstrained optimization test functions collection*, Advanced Modeling and Optimization, 10 (2008), pp. 147-161 .

[21] E. BIRGIN, J.M. MARTÍNEZ, *A spectral conjugate gradient method for unconstrained optimization* , Applied Math. and Optimization, 43, pp.117-128, 2001 .

[22] Y.H. DAI, L.Z. LIAO, *New conjugacy conditions and related nonlinear conjugate gradient methods* , Applied Mathematical Optimization, 43 (2001), pp. 87-101 .

[23] R., FLETCHER, REEVES, *Function minimization by conjugate gradients* , Comput. J., 7 (1964), pp.149-154 .

[24] R., GLOWINSKI, *Numerical Methods for Nonlinear Variational Problems* SpringerVerlag, Berlin, 1984 .

[25] J., GOODMAN, R., KOHN, L., *Reyna, Numerical study of a relaxed variational problem from optimal design.* , Comput. Methods Appl. Mech. Engrg., 57, 1986, pp.107-127 .

[26] SHANNO, D.F., *Conjugate gradient methods with inexact searches* , Math. Oper. Res. 3, 244–256 (1978) .

[27] TOUATI-AHMED, D., STOREY, C., *Efficient hybrid conjugate gradient techniques* , J. Optim. Theory Appl.64(2), 379–397 (1990) .