

Table des matières

0.1	Liste Figure	1
	Introduction	2
1	Recherche Linéaire Exacte	3
1.1	Méthodes de descente	3
1.2	Recherche linéaire exacte	4
1.2.1	Méthode de la section dorée	5
1.2.2	Méthode de dichotomie	9
2	La Recherche Linéaire Inexacte	13
2.1	La recherche linéaire inexacte	13
2.1.1	But de la recherche linéaire	13
2.1.2	Intervalle de sécurité	13
2.1.3	La recherche linéaire inexacte d'Armijo	14
2.1.4	Les conditions de wolf	16
2.1.5	Méthode de Goldstein	19
3	Tests Numériques	22
4	Conclusion	24

0.1 Liste Figure

Figure 1.1 : Une fonction unimodale.....	5
Figure 1.2 : L'évaluation de la fonction objectif en deux points intermédiaires.....	5
Figure 1.3 : Le cas où $f(a_1) < f(b_1)$	6
Figure 1.4 : Trouver la valeur de ρ après une seule nouvelle évaluation de f	7
Figure 1.5 : Réduction de l'intervalle d'incertitude.....	10
Figure 1.6 : Construction de la recherche dichotomique.....	12
Figure 2.1 : Règle d'Armijo.....	15
Figure 2.2 : La condition suffisante diminuer.....	16
Figure 2.3 : La condition de la courbure.....	17
Figure 2.4 : Le pas de déplacement satisfaisant aux condition de Wolf.....	18
Figure 2.5 : Les condition de Goldestien.....	20

Introduction

Dans le calcul du pas de déplacement dans les méthodes de descente, nous sommes confrontés à un compromis, nous voudrions d'une part choisir un pas qui donnera une réduction substantielle de f , mais d'autre part nous voudrions éviter de passer trop de temps pour réaliser ce choix. L'approche idéale serait de trouver le minimum exact de la fonction unidimensionnelle φ définie par :

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

pour $\alpha \geq 0$ où f est la fonction objective à minimiser. Malheureusement, il est difficile, en général, d'identifier en un temps raisonnable cette valeur, cela demande un trop grand nombre d'évaluations de la fonction objectif f et éventuellement son gradient ∇f . Un algorithme de recherche linéaire inexacte engendre une suite de valeurs de candidats pour α , il s'arrête en acceptant une bonne valeur de α lorsque certaines conditions seront satisfaites. La recherche linéaire se fait en deux étapes : Une phase de "bracketing" qui trouve un intervalle contenant le pas de déplacement souhaité et une phase d'interpolation ou bisection qui calcule le meilleur pas de déplacement dans cet intervalle. Ce mémoire est organisé comme suit : dans le premier chapitre nous donnons deux méthodes exactes unidimensionnelles à savoir la méthode section d'orée et dichotomie. Le deuxième chapitre est consacré aux méthodes de recherche linéaire inexacte. Nous étudions la méthode d'Armijo, Goldstein et Wolfe. Nous passerons au dernier chapitre une comparaison numérique entre les deux approches de recherche linéaire.

Recherche Linéaire Exacte

Considérons le problème d'optimisation sans contraintes suivant :

$$\begin{cases} \min f(x) \\ x \in \mathbb{R}^n \end{cases}, \quad (1.1)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction différentiable. Résoudre ce problème numériquement revient en général à générer une suite $(x_k)_{k \in \mathbb{N}}$ qui devrait converger vers la solution optimale x^* du problème (1.1). Les méthodes de descente sont fondées sur la notion de direction de descente et le pas de déplacement.

1.1 Méthodes de descente

Définition 1.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$. On dit que le vecteur d est une direction de descente en x , s'il existe $\bar{\alpha}$ tel que :

$$f(x + \alpha d) < f(x), \forall \alpha \in]0, \bar{\alpha}[.$$

Le scalaire α est appelé le pas de déplacement.

De telles directions sont intéressantes en optimisation, car pour faire décroître f , il suffit de faire un déplacement le long de d . Les méthodes à directions de descente utilisent cette idée pour minimiser la fonction objectif dans (1.1). L'itéré principal d'une telle méthode est formulé comme suit :

$$x_{k+1} = x_k + \alpha_k d_k, \quad \alpha_k > 0.$$

Tout en assurant la propriété :

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

Le vecteur d_k est la direction de descente en x_k alors que α_k est le pas de la méthode. La proposition suivante donne une condition suffisante assurant le fait que d_k soit une direction de descente.

Proposition 1.1 Soit $d \in \mathbb{R}^n$ vérifiant

$$\langle d, \nabla f(x) \rangle < 0. \quad (1.2)$$

Alors d est une direction de descente en x .

Preuve. On a pour $\alpha > 0$

$$f(x + \alpha d_k) = f(x) + \alpha \nabla f(x)^T d_k + \alpha \varepsilon(\alpha) ,$$

d'où

$$\frac{f(x + \alpha d_k) - f(x)}{\alpha} = \nabla f(x)^T d_k + \varepsilon(\alpha) \text{ (i.e } \exists \bar{\alpha} > 0, \forall \alpha \in]0, \bar{\alpha}[\text{)} .$$

On voit bien que pour α suffisamment petit on aura :

$$f(x + \alpha d_k) - f(x) < 0 .$$

□

La direction de descente est en général sous la forme

$$d_k = -B_k^{-1} \nabla f_k ,$$

où B_k est une matrice symétrique et non singulière. Dans la méthode du Gradient B_k est simplement la matrice identité, tandis que dans la méthode de Newton B_k est la matrice Hessienne $\nabla^2 f(x_k)$. Dans les méthodes quasi-Newtonienne B_k est une approximation de la matrice Hessienne. Si d_k vérifie (1.2) et la matrice B_k est définie positive, alors

$$d_k^T \nabla f_k = -\nabla f_k^T B_k^{-1} \nabla f_k < 0 ,$$

d'où $d_k = -B_k^{-1} \nabla f_k$ est une direction de descente.

L'autre aspect des méthodes à direction de descente est la détermination du pas α_k , c'est ce qu'on appelle " La recherche linéaire". Le choix d'un α_k acceptable a un impact majeur sur l'efficacité des méthodes de descente.

1.2 Recherche linéaire exacte

Comme on cherche à minimiser f , il semble naturel de chercher le pas α_k comme solution du problème unidimensionnel suivant :

$$\begin{cases} \min \varphi_k(\alpha) = f(x_k + \alpha d_k) \\ \alpha > 0 \end{cases} . \quad (1.3)$$

De la différentiabilité de f , le pas optimal α_k peut être caractérisé par :

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

Dans le cas où f est une fonction quadratique i.e : $f(x) = \frac{1}{2} x^T H x + x^T p$, avec $H \in \mathbb{R}^{n \times n}$ symétrique et définie positive, un calcul simple montre que la solution du problème (1.3) est donnée explicitement par

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

Pour une fonction quelconque, il est nécessaire d'utiliser une méthode d'optimisation unidimensionnel. pour calculer α_k . On va présenter deux méthodes unidimensionnelles .

1.2.1 Méthode de la section dorée

On s'intéresse à minimiser la fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ sur un intervalle fermé $[a_0, b_0]$. La seule hypothèse que doit vérifier f est qu'elle soit unimodale, ce qui signifie que f n'admet qu'un seul minimum local. Un exemple d'une telle fonction est représenté par (la figure 1.1).

Les méthodes que nous discutons sont basées sur l'évaluation de la fonction objectif à différents points dans l'intervalle $[a_0, b_0]$. Nous choisissons ces points de telle sorte qu'une approximation du minimum de f peut être obtenu en un nombre réduit d'évaluations de f .

Considérons la fonction f sur $[a_0, b_0]$. Si nous évaluons f en un seul point de l'intervalle, nous ne pouvons pas réduire ce dernier en sachant où le minimum est situé. Nous évaluons, cependant, la fonction f en deux points intermédiaires, comme illustre à (la figure 1.2), les deux points soit choisi de telle sorte que la réduction dans l'intervalle $[a_0, b_0]$ soit symétrique, dans le sens où

$$a_1 - a_0 = b_0 - b_1 = \rho (b_0 - a_0),$$

avec

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

Nous évaluons en suite f aux points intermédiaires. Si $f(a_1) < f(b_1)$, le minimum doit se situer dans $[a_0, b_1]$. D'autre part, si $f(a_1) \geq f(b_1)$, le minimum est situé dans $[a_1, b_0]$ (voir la figure 1.3)

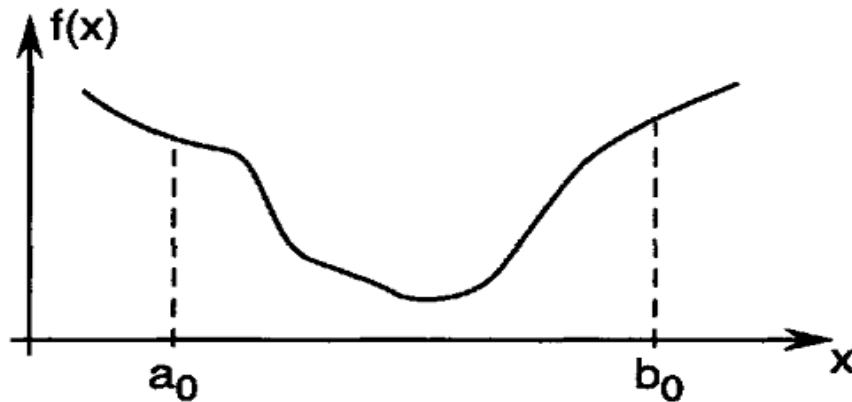


Figure 1.1 Une fonction unimodale.

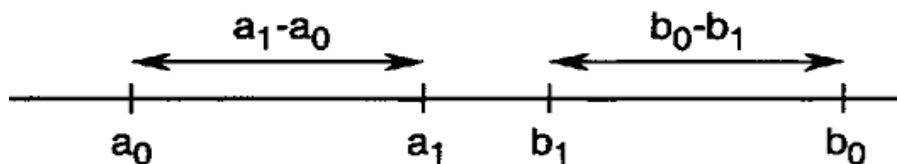


Figure 1.2 L'évaluation de la fonction objectif en deux points intermédiaires .

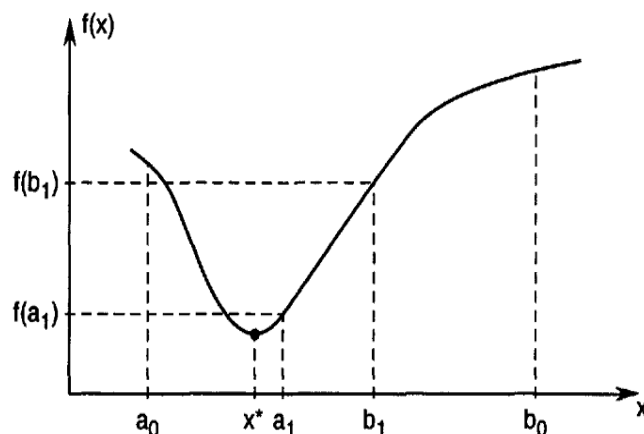


Figure 1.3 Le cas où $f(a_1) < f(b_1)$; le minimum $x^* \in [a_0, b_1]$.

Une fois l'intervalle est réduit nous pouvons répéter le processus et trouver de manière similaire deux nouveaux point a_2 et b_2 , en utilisant la même valeur de $\rho < \frac{1}{2}$.

L'objectif est aussi de réduire au minimum le nombre des évaluations de la fonction objectif tout en réduisant l'intervalle d'incertitude. Supposons, par exemple, que $f(a_1) < f(b_1)$ (comme dans la figure 1.3), alors nous savons que $x^* \in [a_0, b_1]$, a_1 est déjà dans l'intervalle d'incertitude et $f(a_1)$ est déjà connue, nous affectons à b_2 la valeur a_1 . Ainsi, une seule nouvelle évaluation de f à a_2 serait nécessaire.

Pour trouver la valeur de ρ que les résultats dans une seule nouvelle évaluations de f , le rôle de ρ est impotant dans l'algorithme, pour trouver la valeur de ρ donnant une seule évaluation de f (voir la figure 1.4), sans restreindre la généralité nous supposons que $[a_0, b_0]$ admet une longueur unitaire. Alors, afin d'avoir une seule évaluation de f , il est suffisant de choisir ρ comme suit

$$\rho(b_1 - a_0) = b_1 - b_2.$$

Comme $b_1 - a_0 = 1 - \rho$ et $b_1 - b_2 = 1 - 2\rho$, nous aurons

$$\rho(1 - \rho) = 1 - 2\rho.$$

La fonction quadratique en ρ peut être écrite sous la forme

$$\rho^2 - 3\rho + 1 = 0.$$

Ainsi, les solutions sont

$$\rho_1 = \frac{3 + \sqrt{5}}{2}, \rho_2 = \frac{3 - \sqrt{5}}{2}.$$

Puisque nous exigeons que $\rho < \frac{1}{2}$, nous prenons

$$\rho = \frac{3 - \sqrt{5}}{2} \simeq 0.382.$$

Observons que

$$1 - \rho = \frac{\sqrt{5} - 1}{2},$$

et

$$\frac{\rho}{1 - \rho} = \frac{3 - \sqrt{5}}{\sqrt{5} - 1} = \frac{\sqrt{5} - 1}{2} = \frac{1 - \rho}{1},$$

ce qui veut dire

$$\frac{\rho}{1 - \rho} = \frac{1 - \rho}{1}.$$

Cette règle a été appelée en section d'or par les anciens géomètres Grecs. La longueur dans l'intervalle à chaque itération est réduite de rapport égale à $1 - \rho \simeq 0,61803$.

Par conséquent, N étapes de la méthode de section d'or les intervalles par le facteur [3]

$$(1 - \rho)^N \simeq (0,61803)^N.$$

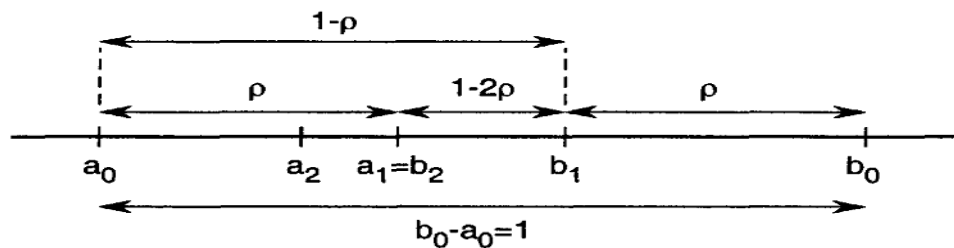


Figure 1.4 : Trouver la valeur de ρ après une seule nouvelle évaluation de f

Algorithme de section d'or :

Initialisation : $a_0, b_0, \varepsilon, \rho, k$ (itération)

$$a_1 \leftarrow a_0 + \rho(b_0 - a_0)$$

$$b_1 \leftarrow a_0 + (1 - \rho)(b_0 - a_0)$$

$$f_1 \leftarrow f(a_1), f_2 \leftarrow f(b_1)$$

Tantque : $|b_0 - a_0| < \varepsilon$

Si $(f_1 < f_2)$ alors

$$b_0 \leftarrow b_1, b_1 \leftarrow a_1$$

$$a_1 \leftarrow a_0 + b_0 - b_1$$

$$f_2 \leftarrow f_1, f_1 \leftarrow f(a_1)$$

Si non

$$a_0 \leftarrow a_1, a_1 \leftarrow b_1$$

$$b_1 \leftarrow a_0 + b_0 - a_1$$

$$f_1 \leftarrow f_2$$

Finsi

Fintanque

Exemple 1.1 utilisant la méthode de la Section d'or pour trouver la valeur de x qui minimise

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

dans l'intervalle $[0, 2]$, avec une précision de $0,3$.

Après les étapes N l'intervalle $[0, 2]$ est réduite par $(0.61803)^N$. Ainsi, nous choisissons N afin que

$$(0.61803)^N \leq \frac{0.3}{2}.$$

Quatre itérations seront nécessaires.

Itération 1. Nous évaluons f aux deux points intermédiaires a_1 et b_1 , Nous aurons

$$\begin{aligned} a_1 &= a_0 + \rho(b_0 - a_0) = 0.7639, \\ b_1 &= a_0 + \rho(1 - \rho)(b_0 - a_0) = 1.236, \end{aligned}$$

où $\rho = \frac{(3-\sqrt{5})}{2}$. Nous calculons

$$\begin{aligned} f(a_1) &= -24.36, \\ f(b_1) &= -18.96. \end{aligned}$$

Ainsi, $f(a_1) < f(b_1)$ et donc l'intervalle d'incertitude est réduit à

$$[a_0, b_1] = [0, 1.236].$$

Itération 2. Nous choisissons $b_2 = a_1$, et f doit être évaluée au nouveau point

$$a_2 = a_0 + \rho(b_1 - a_0) = 0.4721.$$

Nous aurons

$$\begin{aligned} f(a_2) &= -21.10, \\ f(b_2) &= f(a_1) = -24.36. \end{aligned}$$

Maintenant, $f(b_2) < f(a_2)$. Alors l'intervalle d'incertitude est réduit à

$$[a_2, b_1] = [0.4721, 1.236].$$

Itération 3. $a_3 = b_2$, et on calcul b_3

$$b_3 = a_2 + (1 - \rho)(b_1 - a_2) = 0.9443$$

Nous aurons

$$\begin{aligned} f(a_3) &= f(b_2) = -24.36, \\ f(b_3) &= -23.59. \end{aligned}$$

Alors $f(b_3) > f(a_3)$. Par conséquent, l'intervalle d'incertitude est

$$[a_2, b_3] = [0.4721, 0.9443].$$

Itération 4. $b_4 = a_3$, et

$$a_4 = a_2 + \rho(b_3 - a_2) = 0.6526.$$

Nous aurons

$$\begin{aligned} f(a_4) &= -23.84, \\ f(b_4) &= f(a_3) = -24.36. \end{aligned}$$

Par conséquent, $f(a_4) > f(b_4)$. Ainsi, la valeur de x qui minimise f se trouve dans l'intervalle

$$[a_4, b_3] = [0.6526, 0.9443]$$

Notez que $b_3 - a_4 = 0.292 < 0.3$.

1.2.2 Méthode de dichotomie

Considérons une fonction unimodale ayant un minimum dans $[x_L, x_U]$. Cet intervalle est dit l'intervalle d'incertitude. Le minimum x^* de f peut être situé en réduisant progressivement l'intervalle d'incertitude jusqu'à l'optimum à une précision prêt.

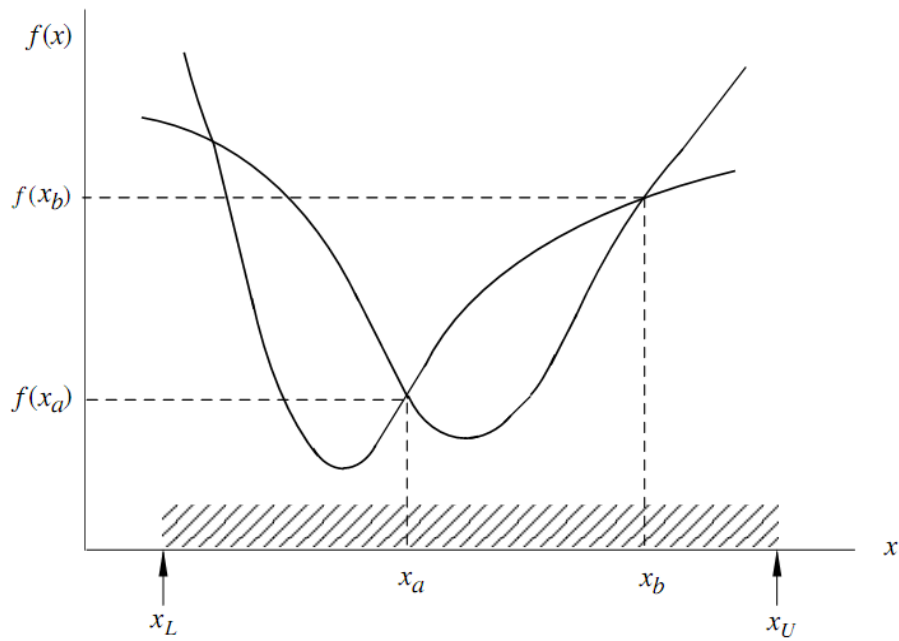
Si la valeur de $f(x)$ est connu en un seul point x_a dans l'intervalle $x_L < x_a < x_U$, le point x^* est également susceptible d'être dans l'intervalle x_L à x_a ou x_a à x_U comme représenté sur (la Figure 1.5 (a)) Par conséquent, les informations disponibles ne sont pas suffisantes pour permettre la réduction de l'intervalle d'incertitude. Cependant, si la valeur de $f(x)$ est connue en deux points : x_a et x_b , une réduction immédiate est possible. Trois possibilités peuvent survenir, à savoir :

- (a) $f(x_a) < f(x_b)$
- (b) $f(x_a) > f(x_b)$
- (c) $f(x_a) = f(x_b)$

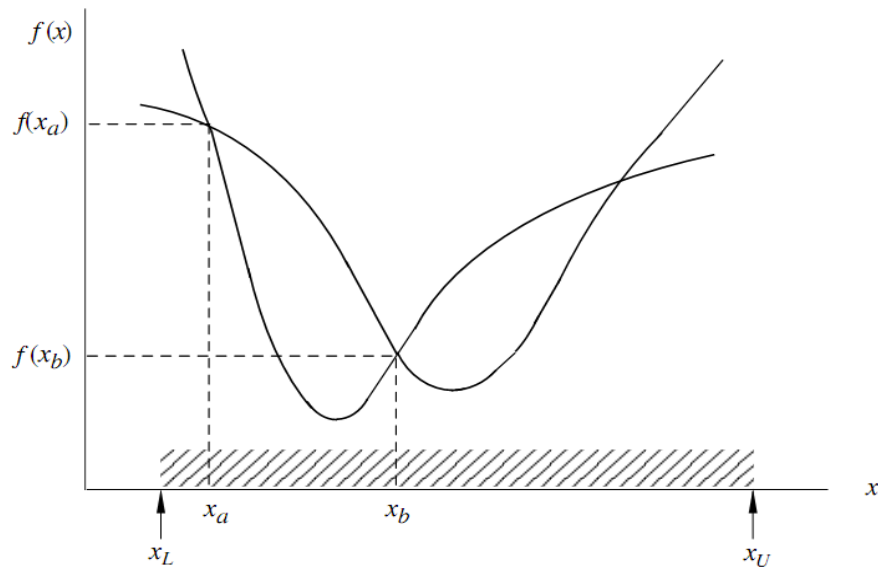
Dans le cas (a), x^* peut être l'intervalle $x_L < x^* < x_a$ ou $x_a < x^* < x_b$, c'est à dire : $x_L < x^* < x_b$ (la Figure. 1.5 a). La possibilité $x_b < x^* < x_U$ est définitivement écartée car cela impliquerait que $f(x)$ possède deux minimum.

De même, pour le cas (b), nous devons avoir $x_a < x^* < x_U$ (la figure.1.5b). Pour le cas (c), nous devons avoir $x_a < x^* < x_b$, est que, les deux inégalités $x_L < x^* < x_b$, $x_a < x^* < x_U$

doivent être satisfaites comme dans (la Figure 1.5c).

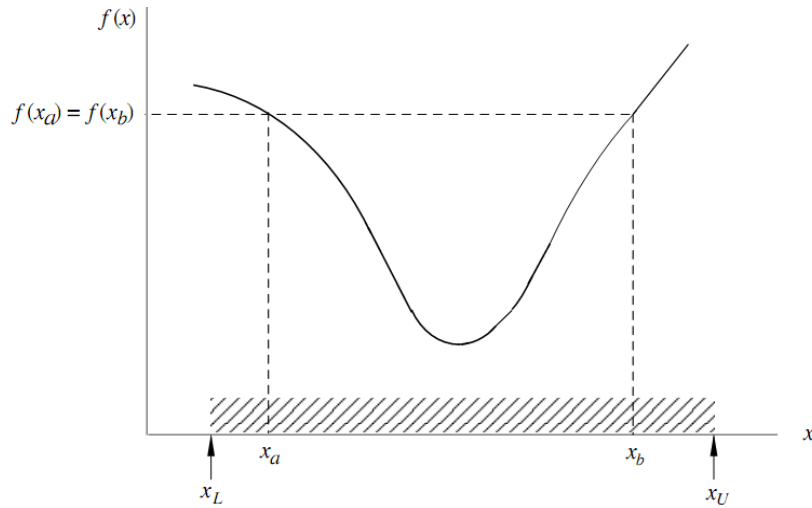


(a)



(b)

Figure 1.5 .Réduction de l'intervalle d'incertitude : (a) cas (a), $f(x_a) < f(x_b)$, (b) cas (b), $f(x_a) > f(x_b)$.



(c)

Figure 1.5 Réduction de l'intervalle d'incertitude : (c) cas (c), $f(x_a) = f(x_b)$.

Une stratégie simple pour réduire l'intervalle d'incertitude est la recherche dichotomique. Dans ce procédé, $f(x)$ est évaluée à deux points $x_a = x_1 - \varepsilon/2$ et $x_b = x_1 + \varepsilon/2$ où ε est un petit nombre positif. Puis, si $f(x_a) < f(x_b)$ alors on a l'intervalle x_L à $x_1 + \varepsilon/2$ et si $f(x_a) > f(x_b)$ alors on prendra l'intervalle $x_1 - \varepsilon/2$ à x_U , et si $f(x_a) = f(x_b)$ on pourra prendre l'un des deux. Si nous supposons que $x_1 - x_L = x_U - x_1$, c'est à dire, $x_1 = (x_L + x_U)/2$, l'intervalle d'incertitude est immédiatement réduit de moitié. La même procédure peut être répétée pour l'intervalle réduit, (i.e) $f(x)$ peut être évaluée en $x_2 - \varepsilon/2$ et $x_2 + \varepsilon/2$ où x_2 est situé au centre de l'intervalle réduite, et ainsi de suite. Par exemple, si le recherche dichotomique est appliquée à la fonction de (la Fig. 1.6), l'intervalle d'incertitude sera réduit à partir de $0 < x^* < 1$ à $9/16 + \varepsilon/2 < x^* < 5/8 - \varepsilon/2$ dans quatre itérations. Chaque itération réduit l'intervalle d'incertitude de moitié et donc après K itérations la longueur de l'intervalle d'incertitude se réduit à

$$I_K = \left(\frac{1}{2}\right)^K I_0$$

où $I_0 = x_U - x_L$. Par exemple après 7 itérations l'intervalle d'incertitude sera réduit à moins de 1% de l'intervalle initial. L'effort de calcul correspondant serait de 14 évaluations de la fonction objectif puisque deux évaluations sont nécessaires pour chaque itération.[1].

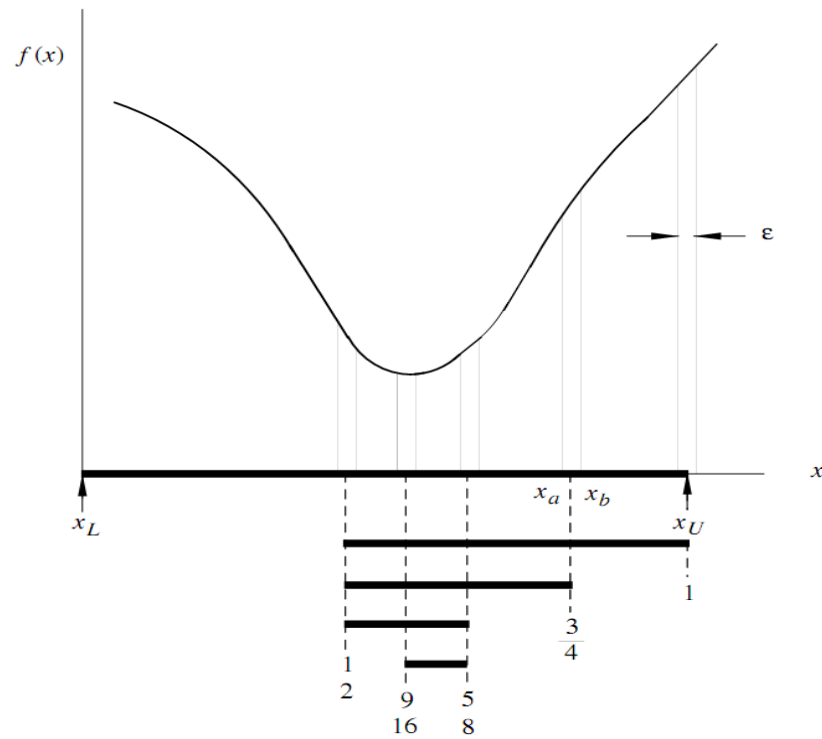


Figure 1.6 Construction de la recherche dichotomique.

Algorithme de Dichotomie :

Initialisation : $a_0, b_0, \varepsilon, tol, k \leftarrow 1$ (itération)

$a_1 \leftarrow a_0, b_1 \leftarrow b_0$

Tantque $|b(k) - a(k)| < tol$

$$x_k \leftarrow \frac{a_k + b_k}{2}$$

$$c_k \leftarrow x_k - \frac{\varepsilon}{2}$$

$$d_k \leftarrow x_k + \frac{\varepsilon}{2}$$

Si

$f(c_k) \leq f(d_k)$ alors

$$a_{k+1} \leftarrow a_k$$

$$b_{k+1} \leftarrow d_k, \text{ Sinon}$$

$$a_{k+1} \leftarrow c_k$$

$$b_{k+1} \leftarrow b_k$$

Finsi

Fintanque

$k = k + 1.$

La Recherche Linéaire Inexacte

2.1 La recherche linéaire inexacte

2.1.1 But de la recherche linéaire

On a vu que dans le cas non-quadratique la méthode de descente :

$$x_{k+1} = x_k + \alpha_k d_k, \alpha_k > 0,$$

nécessitent la recherche d'une valeur optimale $\alpha_k > 0$ vérifiant :

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

On définit comme précédemment la fonction $\varphi(\alpha) = f(x_k + \alpha d_k)$. Rappelons que si f est différentiable, le pas optimal $\bar{\alpha}$ peut être caractérisé par

$$\begin{aligned} \varphi'(\bar{\alpha}) &= 0, \\ \varphi(\bar{\alpha}) &\leq \varphi(\alpha), \text{ pour } 0 \leq \alpha \leq \bar{\alpha}, \end{aligned}$$

autrement dit, $\bar{\alpha}$ est un minimum local de φ qui assure de plus la décroissance de f . En fait, dans la plupart des algorithmes d'optimisation modernes, on ne fait jamais de recherche linéaire exacte, car trouver $\bar{\alpha}$ signifie qu'il va falloir calculer un grand nombre de fois la fonction φ , et cela peut être dissuasif du point de vue du temps de calcul. En pratique, on cherche plutôt une valeur de α qui assure une décroissance suffisante de f . Cela conduit à la notion d'intervalle de sécurité[5].

2.1.2 Intervalle de sécurité

Définition 2.1 *On dit que $[a, b]$ est un intervalle de sécurité s'il permet de classer les valeurs de α de la façon suivante :*

- i Si $\alpha < a$ alors α est considéré trop petit .
- ii Si $a \leq \alpha \leq b$ alors α est satisfaisant .

iii Si $\alpha > b$ alors α est considéré trop grand .

Le problème est de traduire de façon numérique sur φ les trois conditions précédentes, ainsi que de trouver un algorithme permettant de déterminer a et b . L'idée est de partir d'un intervalle suffisamment grand pour contenir $[a, b]$, et d'appliquer une bonne stratégie pour itérativement réduire cet intervalle[5].

Algorithme de base

Initialement, on part de $[c, d]$ contenant $I = [a, b]$, par exemple en prenant $c = 0$ et d tel que $\varphi(d) > \varphi(0)$ (une telle valeur de d existe avec un minimum d'hypothèses, par exemple f coercive). On fait, ensuite, les itérations suivantes :

- 1 On choisit α dans l'intervalle $[c, d]$.
- 2 Si α est trop petit on prend $c = \alpha$ et on retourne en 1.
- 3 Si α est trop grand on prend $d = \alpha$ et on retourne en 1.
- 4 Si α convient on s'arrête.

Il faut maintenant préciser quelles sont les relations sur φ qui vont nous permettre de caractériser les valeurs de α convenables, ainsi que les techniques utilisées pour réduire l'intervalle de sécurité.

2.1.3 La recherche linéaire inexacte d'Armijo

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $x \in \mathbb{R}^n$, $d \in \mathbb{R}^n$ telles que $\nabla f(x_k)^T d_k < 0$: Définissons la fonction $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ avec $\varphi(\alpha) = f(x_k + \alpha d_k)$, $\alpha > 0$.

Notons que :

$$\varphi'(\alpha) = (\nabla f(x_k + \alpha d_k))^T d_k$$

$$\varphi'(0) = (\nabla f(x_k))^T d_k < 0$$

$$\varphi(0) = f(x_k)$$

Posons

$$\varphi_1(\alpha) = \varphi(0) + \varphi'(0)\alpha.$$

Donc

$$\varphi_1(\alpha) = f(x_k) + \nabla f^T(x_k)d_k\alpha.$$

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

$$\hat{\varphi}(\alpha) = f(x_k) + c\alpha\varphi'(0) = f(x_k) + c\alpha\nabla f^T(x_k)d_k; \quad 0 < c < 1 \quad (2.1)$$

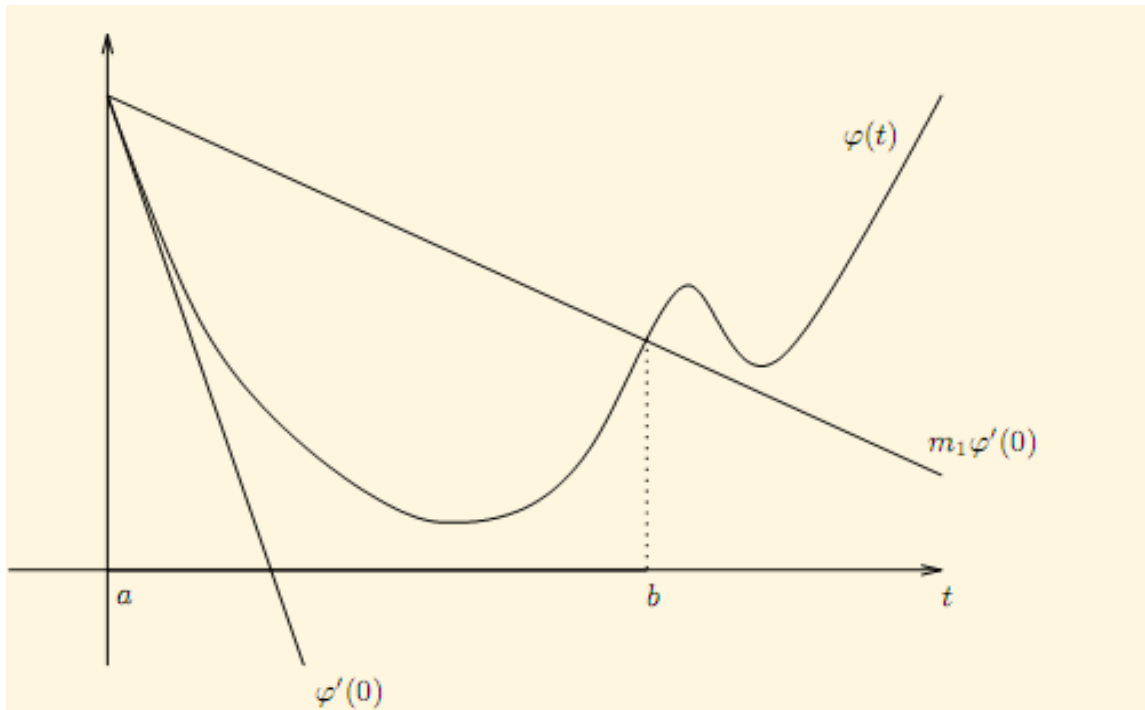


Fig2.1 Règle d'Armijo.

On cherche $\bar{\alpha}_k$ tel que

$$\varphi(\bar{\alpha}_k) \leq \hat{\varphi}(\bar{\alpha}_k).$$

Remarque 2.1 La condition $\varphi(\bar{\alpha}_k) \leq \hat{\varphi}(\bar{\alpha}_k)$ implique la décroissance de la fonction f . En effet

$$\begin{aligned} \varphi(\bar{\alpha}_k) &\leq \hat{\varphi}(\bar{\alpha}_k), \\ f(x_k + \bar{\alpha}_k d_k) &\leq f(x_k) + c \bar{\alpha}_k \nabla f^T(x_k) d_k < f(x_k), \end{aligned}$$

car la direction d est une direction de descente[4].

Algorithme (Règle d'Armijo)

- 1 Initialisation : $\alpha_{g,1} = \alpha_{d,1} = 0$, $c \in]0, 1[$, posons $k = 1$ et aller à 2.
- 2 Itération principale
 - si $\varphi(\alpha_k) \leq \varphi(0) + c \varphi'_k(0) \alpha_k$; STOP ($\alpha^* = \alpha_k$).
 - si $\varphi(\alpha_k) > \varphi(0) + c \varphi'_k(0) \alpha_k$, alors
 - $\alpha_{d,k+1} = \alpha_k$, $\alpha_{g,k+1} = \alpha_{g,k}$ et aller à 3.
- 3 Déterminer $\alpha_{k+1} \in]\alpha_{g,k+1}, \alpha_{d,k+1}[$
Remplacer k par $k + 1$ et aller à 2.

2.1.4 Les conditions de wolf

Une condition de recherche linéaire inexacte exige que α_k soit calculer de façon à diminuer suffisamment la fonction objectif f , donnée par :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T d_k, \quad (2.2)$$

avec $c_1 \in]0, 1[$. Autrement dit, la réduction de f devrait être proportionnelle à la fois du pas de déplacement α_k et la dérivée directionnelle $\nabla f_k^T d_k$. L'inégalité (2.2) est parfois appelée la condition d'Armijo.

La condition suffisante de diminution est illustrée dans (la Figure 2.2).

On note, $l(\alpha) = f(x_k) + c_1 \alpha \nabla f_k^T d_k$. La fonction $l(\cdot)$ est une fonction linéaire à pente négative $c_1 \nabla f_k^T d_k$, mais comme $c_1 \in]0, 1[$, $l(\alpha)$ se trouve au-dessus du graphe de φ pour les petites valeurs positives de α . La condition suffisante de diminution indique que α est acceptable seulement si $\varphi(\alpha) \leq l(\alpha)$. Les intervalles sur lesquels cette condition est satisfaite sont indiquées dans (la Figure 2.2). En pratique, c_1 est choisi pour être assez faible, par exemple $c_1 = 10^{-4}$. La condition suffisante de diminution ne suffit pas pour assurer que l'algorithme fait des progrès raisonnable : comme nous le voyons dans (la Figure 2.2), ses progrès sont assurés pour toutes les valeurs suffisamment petites de α . Pour exclure des mesures trop petites, nous introduisons une deuxième condition, la condition de courbure, qui est nécessaire pour α_k satisfaite

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f_k^T d_k,$$

pour certains constant $c_2 \in]c_1, 1[$, où c_1 est la constante de (2.2). Et $\nabla f(x_k + \alpha_k d_k)^T d_k$ est tout simplement la dérivée $\varphi'(\alpha_k)$, donc la condition de la courbure assure que la pente de la fonction φ en α_k est supérieure à c_2 dans la pente initiale $\varphi'(0)$. C'est logique car si la pente

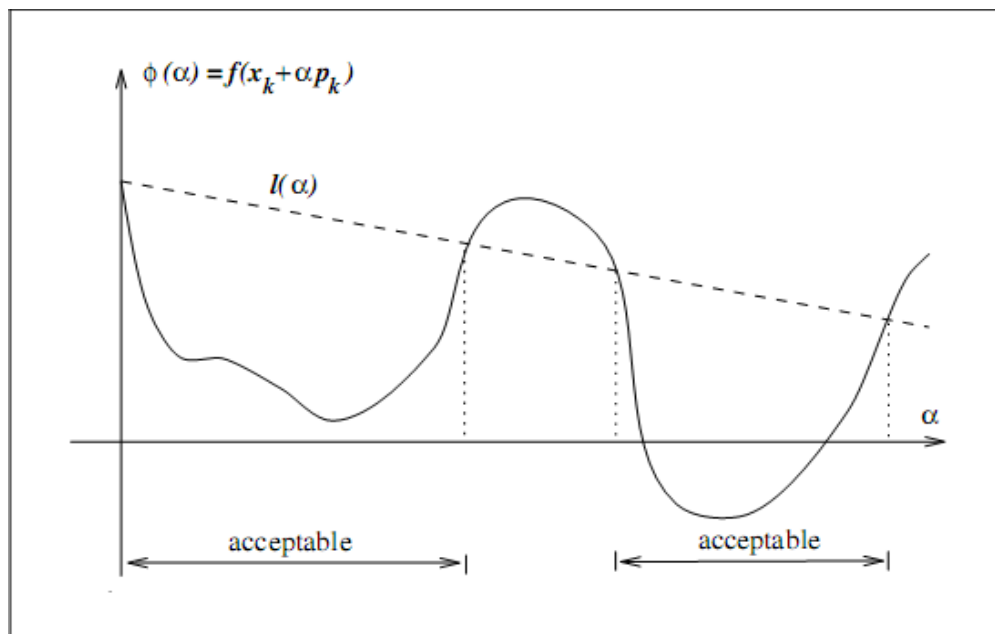


Figure 2.2 la condition suffisante diminuer.

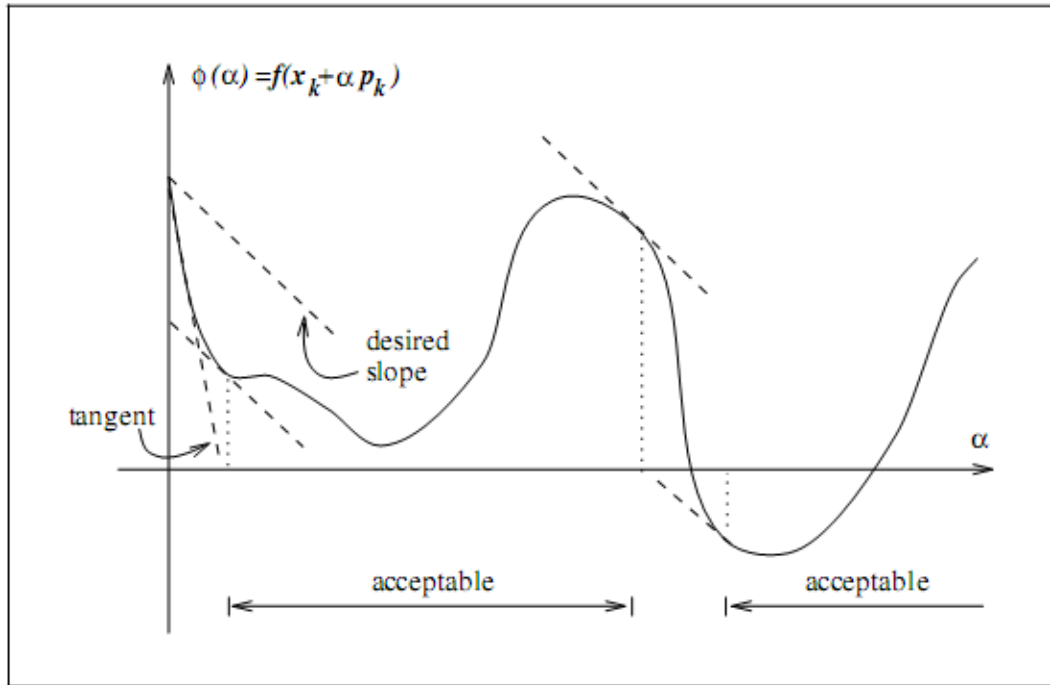


Figure 2.3 la condition de la courbure.

$\varphi'(\alpha)$ est fortement négatif, nous avons une indication que nous pouvons réduire f significativement en déplaçant le long de la direction choisie.

D'autre part, si $\varphi'(\alpha_k)$ est seulement négative ou même positif, c'est un signe que nous ne pouvons atteindre beaucoup plus de diminution de f dans cette direction. Donc, il est logique de mettre fin à la recherche linéaire.

La condition de la courbure est illustrée à (la Figure 2.3). Les valeurs typiques de c_2 sont 0,9 lorsque la recherche direction d_k est choisie par la méthode de Newton ou méthode de quasi-Newton et 0.1 lorsque d_k provient dans la méthode du gradient conjugué non linéaire. La diminution suffisante combinée à la condition de la courbure sont connues comme les conditions de Wolfe données par :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T d_k \quad (2.3)$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f_k^T d_k \quad (2.4)$$

avec $0 < c_1 < c_2 < 1$. Nous illustrons leur (Figure 2.4).

Le pas de déplacement peut satisfaire les conditions de Wolfe sans être particulièrement proche du minimum de φ , que nous montrons dans (Figure 2.4). Cependant, Nous pouvons modifier la condition de courbure pour forcer α_k à se trouver au moins aux voisinage du minimum local ou le point fixe de φ . Les fortes conditions de Wolfe exigent α_k pour satisfaire

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T d_k \quad (2.5)$$

$$\left| \nabla f(x_k + \alpha_k d_k)^T d_k \right| \leq c_2 \left| \nabla f_k^T d_k \right|, \quad (2.6)$$

avec $0 < c_1 < c_2 < 1$. La seule différence avec les conditions de Wolfe c'est de permettre à la dérivées $\varphi'(\alpha_k)$ d'être trop positive. Par conséquent, nous excluons les points qui sont loin d'être les points fixes de φ .

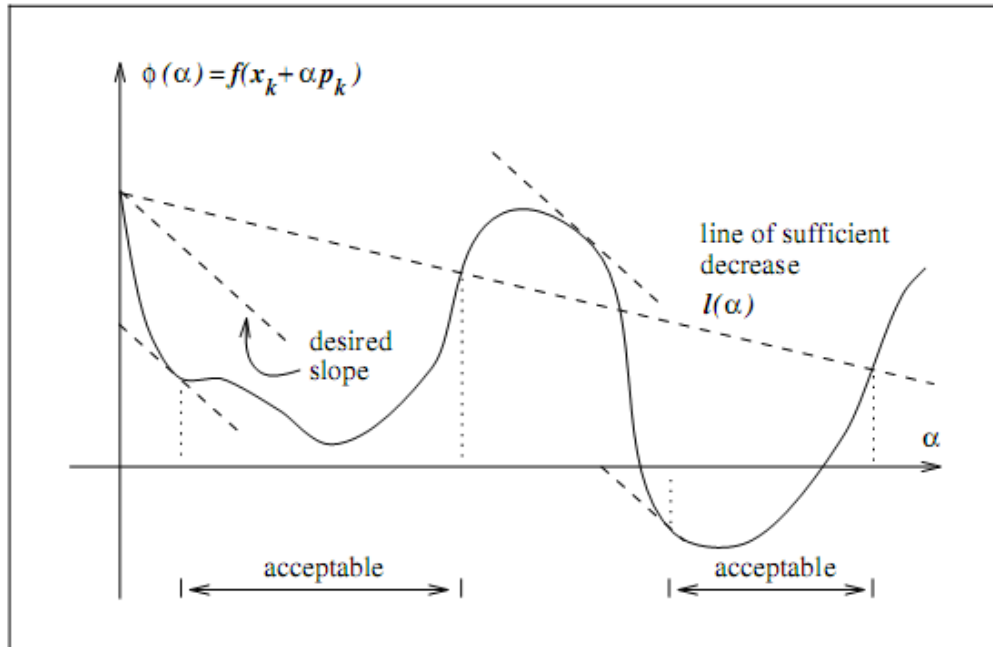


Figure 2.4 le pas de déplacement satisfaisant aux conditions de Wolfe.

Il n'est pas difficile de prouver l'existence d'un pas de déplacement qui satisfait les conditions de Wolfe pour chaque fonction f qui est continue et bornée ci-dessous[2].

Lemme 2.1 *Supposons que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est continue et différentiable. Soit d_k une direction de descente à x_k et supposons que f est bornée, ci-dessous, l'ensemble $\{x_k + \alpha d_k | \alpha > 0\}$. Ensuite, si $0 < c_1 < c_2 < 1$, il existe des intervalles de pas de déplacement satisfaisant aux conditions de Wolfe (2.3) et (2.4) et les conditions de la forte Wolfe (2.5) et (2.6) [2].*

Théorème 2.1 [2] *Nous rappelons maintenant le théorème de valeur moyenne pour les fonctions unidimensionnelles.*

à tant donné une fonction différentiable $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ et deux nombres réels α_0 et α_1 qui satisfont $\alpha_1 > \alpha_0$, que nous avons

$$\varphi(\alpha_1) = \varphi(\alpha_0) + \varphi'(\xi)(\alpha_1 - \alpha_0) \quad (2.7)$$

pour certains $\xi \in (\alpha_0, \alpha_1)$. Une extension de ce résultat par une fonction à variables multiples $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est que pour tout vecteur p nous avons

$$f(x + p) = f(x) + \nabla f(x + \alpha p)^T p. \quad (2.8)$$

Preuve. Notez que $\varphi(\alpha) = f(x_k + \alpha d_k)$ est bornée, ci dessous, pour tout $\alpha > 0$. Depuis $0 < c_1 < 1$, posons $l(\alpha) = f(x_k) + c_1 \alpha \nabla f_k^T d_k$ est illimitée ci dessous et doivent par conséquent se croisent le graphe de φ au moins une fois, laisser $\alpha' > 0$ étant la valeur plus petite intersection de α , c'est à dire

$$f(x_k + \alpha' d_k) = f(x_k) + c_1 \alpha' \nabla f_k^T d_k \quad (2.9)$$

La condition suffisante de diminution (2.3) détient clairement pour toutes les pas de déplacement inférieur à α .

Par le théoreme de la valeur moyenne voir (2.8), il existe $\alpha'' \in (0, \alpha')$ tel que

$$f(x_k + \alpha' d_k) - f(x_k) = c_1 \alpha' \nabla f_k \left(x_k + \alpha'' d_k \right)^T d_k \quad (2.10)$$

En Combinant (2.9) et (2.10), nous obtenons

$$\nabla f_k \left(x_k + \alpha'' d_k \right)^T = c_1 \nabla f_k^T d_k > c_2 \nabla f_k^T d_k \quad (2.11)$$

depuis $c_1 < c_2$ et $\nabla f_k^T d_k < 0$. Donc, α est conforme aux conditions de wolfe (2.3), (2.4), les inégalités tenir strictement dans les deux((2.3), (2.4)),((2.5),(2.6))

Ainsi, par notre hypothèse :

$$\nabla f_k = \nabla^2 f(x^*) (x_k - x^*) + o(\|x_k - x^*\|).$$

il y a un intervalle autour de α pour laquelle les conditions de wolfe tenir, en autre, depuis le terme dans la partie gauche de (2.11) est négatif, les forte conditions de wolfe (2.9) tenir dans le même intervalle[2]. \square

Algorithme de Wolf :

1 Initialisation : $\alpha = 1$ (par exemple), $\alpha^- = \alpha^+ = 0$, on se donne $0 < c_1 < c_2$.

2 Si $\varphi_k(\alpha) \leq \varphi_k(0) + c_1 \alpha \varphi'_k(0)$ et $\varphi'_k(\alpha) \geq c_2 \varphi'_k(0)$; stop, $\alpha_k = \alpha$.

3 Sinon,

si $\varphi_k(\alpha) > \varphi_k(0) + c_1 \alpha \varphi'_k(0)$, on pose $\alpha^+ = \alpha$.

si $\varphi_k(\alpha) \leq \varphi_k(0) + c_1 \alpha \varphi'_k(0)$ et $\varphi'_k(\alpha) < c_2 \varphi'_k(0)$; on pose $\alpha^- = \alpha$ et on.va à 4.

4 Choisi d'un nouveau α :

si $\alpha^+ = 0$, on cherche $\alpha > \alpha^-$ (par exemple $\alpha = 2\alpha^-$).

si $\alpha^+ > 0$, on cherche $\alpha \in]\alpha^-, \alpha^+[$ (par exemple $\alpha = \frac{\alpha^- + \alpha^+}{2}$).

Retourne à 2[6].

2.1.5 Méthode de Goldstein

Les conditions de Goldstein assurent que le pas de déplacement α réalise une diminution suffisante mais il n'est pas trop petit. Ces conditions peuvent être données par :

$$f(x_k) + (1 - c) \alpha_k \nabla f_k^T d_k \leq f(x_k + \alpha_k d_k) \leq f(x_k) + c \alpha_k \nabla f_k^T d_k, \quad (2.12)$$

avec $0 < c < 1/2$. La première inégalité est introduite pour contrôler le pas de déplacement ; voir (figure2.5) et on peut exclure tous les minimums de φ . Les conditions de Goldstein sont utilisées souvent dans les méthodes de type-Newton, mais ne sont pas bien adaptées pour les méthodes quasi-Newtonniennes qui assure une approximation de Hessienne qui est définie positive[2].

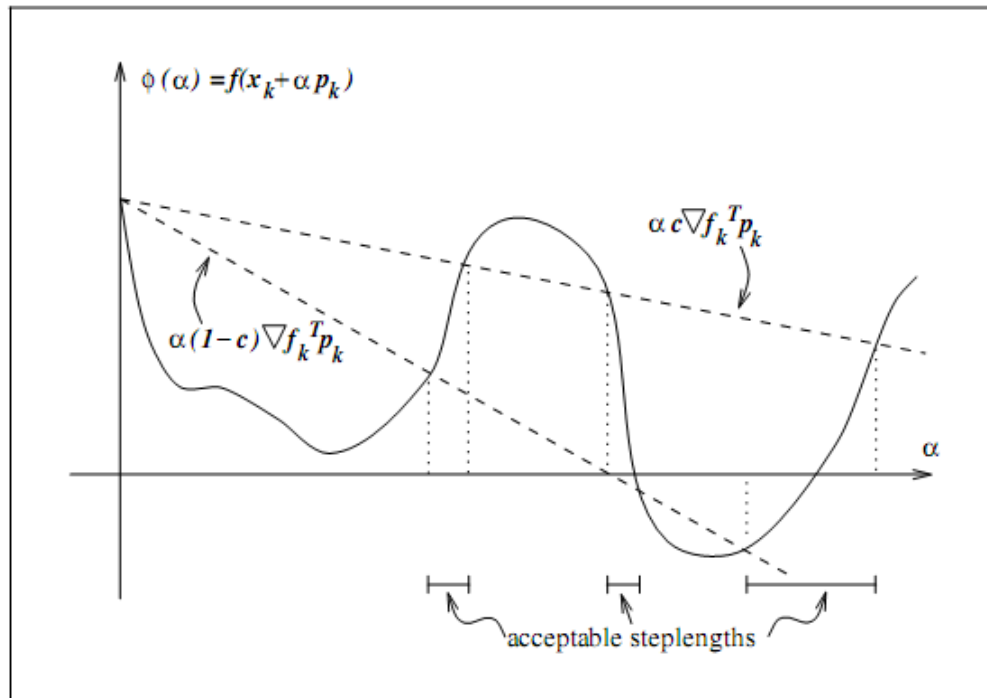


Fig2.5 Les conditions de Goldstein.

Diminution suffisante et retour en arrière

Nous avons mentionné que la deuxième inégalité de (2.2), comme condition de diminution suffisante, ne suffit pas pour assurer que l'algorithme fasse une progression raisonnable dans la direction de recherche donnée. Toutefois, si l'algorithme de recherche linéaire choisit comme candidat le pas de déplacement de manière appropriée en utilisant une approche dite retour en arrière, et il suffit d'utiliser l'état de diminution suffisante pour mettre fin à la procédure de recherche linéaire. Il est basé sur retour en arrière comme suit[2].

Algorithme (retour en arrière de recherche linéaire) Choisissez $\bar{\alpha} > 0, \rho \in (0, 1), c \in (0, 1)$; Définir $\alpha \rightarrow \bar{\alpha}$;
 répéter jusqu'à $f(x_k + \alpha d_k) \leq f(x_k) + c\alpha \nabla f_k^T d_k$
 $\alpha \leftarrow \rho\alpha$;
 fin (répéter)
 Terminer avec $\alpha_k = \alpha$.

Dans cette procédure, le pas de déplacement initial $\bar{\alpha}$ est choisi pour être égal à 1 dans les méthodes de Newton et quasi-Newton, mais peut avoir différentes valeurs dans d'autres algorithmes comme le gradient à pas optimal ou gradient conjugué. Le pas de déplacement

acceptable sera trouvé après un nombre fini d'itération, parce que α_k sera devenu petit pour que la condition de diminution suffisante soit vérifiée. Dans la pratique, le facteur de contraction ρ varie souvent dans chaque itération de la recherche linéaire. Par exemple, il peut être choisi par interpolation. Nous devons assurer qu'à chaque itération nous avons $\rho \in [\rho_{lo}, \rho_{hi}]$, pour certaines constantes fixées $0 < \rho_{lo} < \rho_{hi} < 1$.

L'approche du retour en arrière garantit que le choix de pas de déplacement α_k a une valeur fixée (le choix initial $\bar{\alpha}$), ou bien qu'il est assez petit pour satisfaire la condition de la diminution suffisante, mais pas trop petit. La dernière affirmation tient parce que la valeur α_k est acceptée dans un facteur ρ de l'itération précédente de valeur α_k / ρ , qui a été rejeté parce qu'il ne vérifie pas la condition de la diminution suffisante, donc le pas de déplacement est trop grand.

Cette simple stratégie pour terminer la recherche linéaire est bien adaptée pour les méthodes de Newton, mais moins appropriée pour quasi-Newton et les méthodes de gradient conjugués[2].

Algorithme 3.2 de Goldstein

1. **1** Initialisation : Choisir $\alpha_0 \in [0, 10^{100}]$ et $\rho \in]0; 1[$. Poser $a_0 = 0$, $b_0 = 10^{100}$, poser $k = 0$ et aller à 2.
- 2 **2** Test Goldstein1
 Itération k , on a $[a_k, b_k]$ et α_k , calculez $\varphi(\alpha_k)$.
 Si $\varphi(\alpha_k) \leq \varphi_k(0) + \rho \alpha_k \varphi'_k(0)$, et allez à 3
 Sinon
 Poser $b_{k+1} = \alpha_k$, $a_{k+1} = a_k$, et allez à 4.
- 3 **3** Test Goldstein 02
 Si $\varphi(\alpha_k) \geq \varphi_k(0) + (1 - \rho) \alpha_k \varphi'_k(0)$ stop. $\alpha^* = \alpha_k$.
 Sinon Poser $a_{k+1} = \alpha_k$, $b_{k+1} = b_k$ et allez à 4.
- 4 **4** Poser $\alpha_{k+1} = \frac{a_{k+1} + b_{k+1}}{2}$
 Poser $k = k + 1$ et allez 2.

Tests Numériques

On définit dans ce chapitre un algorithme qui accélère la convergence du gradient a pas optimal avec la recherche linéaire inexacte d'armijo et la recherche linéaire exacte (section d'or).

Nous avons choisit la fonction de Resonbrock comme fonction teste sur matlab qui définie par :

$$f(x) = \sum_{i=1}^{n/2} [(x_{2i-1} - 1)^2 + 100(x_{2i-1}^2 - x_{2i})^2] .$$

Et nous avons associé un nombre de variable croissant n=2,4,6, l'algorithme s'arrête quant $\|\nabla f(x_k)\| < 10^{-6}$.

Pour $n = 2$

x^0	(2, 2)			(-2, 2)			(-10, 10)		
Méthodes	La solution	Le temps	Nbr d'itér	La solution	Le temps	Nbr d'itér	La solution	Le temps	Nbr d'itér
Armijo	(1,1)	4.9444	14502	(1,1)	3.8307	11390	(1,1)	14.4132	38981
Section d'or	(1,1)	20.0138	18947	(1,1)	6.9035	6768	(1,1)	27.0215	26115

Pour $n = 4$

x^0	(2, 2, 2, 2)			(-2, 2, -2, 2)			(-10, 10, -10, 10)		
Méthodes	La solution	Le temps	Nbr d'itér	La solution	Le temps	Nbr d'itér	La solution	Le temps	Nbr d'itér
Armijo	(1, 1, 1, 1)	5.5020	14933	(1, 1, 1, 1)	4.1740	11821	(1, 1, 1, 1)	15.2290	39387
Section d'or	(1, 1, 1, 1)	20.5967	19034	(1, 1, 1, 1)	14.7317	14269	(1, 1, 1, 1)	27.9616	26951

Pour $n = 6$

x^0	$(2, 2, 2, 2, 2, 2)$			$(-2, 2, -2, 2, -2, 2)$			$(-10, 10, -10, 10, -10, 10)$		
Méthodes	La solution	Le temps	Nbr d'itér	La solution	Le temps	Nbr d'itér	La solution	Le temps	Nbr d'itér
Armijo	(1,1,1,1,1,1)	5.6429	15162	(1,1,1,1,1,1)	4.4161	12074	(1,1,1,1,1,1)	15.8501	39641
Section d'or	(1,1,1,1,1,1)	12.1038	11226	(1,1,1,1,1,1)	6.8050	6490	(1,1,1,1,1,1)	26.3113	25578

Nbr d'itér : Le nombre d'itération .

Nous Comparons à travers les tests numériques effectués que si le nombre d'itérations avec méthode d'Armijo par fois dépasse méthode section d'or, le temps CPU dans la méthode d'Armijo est toujours inférieur que section d'or.

Ce procédé montre clairement que la recherche linéaire inexacte d'Armijo performante que la recherche linéaire exacte (section d'or).

Conclusion

Les tests numérique réalisés montrons que les méthodes de recherche linéaire inexacte sont d'un apport énorme dans la résolution des problèmes d'optimisation sans contraintes. Les tests montrent que même si le nombre d'itérations avec une méthode inexacte parfois dépasse le cas exacte, le temps CPU est en général meilleurs. ceci explique sont utilisation comme subroutines dans les algorithmes de descentes les plus utilisés pratique.

Bibliographie

- [1] A. Antoniou and W. S. Lu : *Practical optimization*, Algorithms and Engineering Applications Andreas Antoniou, Computer Engineering university of Victoria. Canada.(2007).
- [2] J. Nocedal and S. J. Wright : *Numerical Optimization*, Springer Series in Operation Research and Financial Engineering. (2006).
- [3] E. K. P Chong and S. H. Zak : *An Introduction To Optomization*, *Wiley-Interscience Series In Discrete Mathematics And Optimisation*. New York, (2001).
- [4] H. Degaichia : *L'Epsilon Steepest Descent Algorithme Associé à La Recherche Linéaire Inexacte de Wolfe*. thèse de doctorat (2014).
- [5] S. Mottelet : *Optimisation non -linéaire*, Université de technologie de Compiègne (2003).
- [6] M.Bergounioux : *Optimisation et Controle et des Systèmes Linéaires*. Université d'Orléans, Dunod, Paris.(2001).