Chapitre 3

Recherche linéaire

Chapitre 3

Recherche linéaire

Dans ce chapitre, nous nous intéressons uniquement au calcul du pas optimal t^* qui est solution du problème

$$\min_{t>0} \theta(t). \tag{P'}$$

Avec

- $\theta(t)$, la fonction test qui représente $f(x+td)$, définie pour $t \geq 0$,
- x, le point de départ de la recherche linéaire,
- d, la direction de la recherche linéaire.

Pour la suite, d est supposée une direction de descente et de ce fait, on aura toujours

$$\theta'(0) < 0.$$

3.1 Principe général de résolution

Un algorithme de recherche linéaire résolvant le problème (P′) génère une suite de valeurs $\left\{t^0, t^1, ..., t^i, ..\right\}_{i\in\mathbb{N}}$ est un test à trois sorties qui étant donné un $t^0 > 0$ (une valeur initial) répond si

a) t^i est satisfaisant donc c'est une solution,

b) t^i est trop grand,

c) t^i est trop petit.

On notera t^* un pas optimal, t_- un pas trop petit et t_+ un pas trop grand. Commençons par un exemple simple, mais on verra ensuite qu'il servira de modèle pour des algorithmes plus développés.

Algorithme 3.1. *Soit $t^0 > 0$ une valeur initial, $t_+^0 = t_-^0 = 0$,*

a) *$\theta'(t^i) = 0$ (on peut envisagé de stopper la recherche linéaire t^i à l'air de minimiser θ), donc $t^* = t^i$.*

b) *$\theta'(t^i) > 0$ (alors le minimum de θ devrait être plus petit que t^i), donc $t_+^{i+1} = t^i$, choisir t^{i+1}, remplacer i par $i+1$ et aller en a).*

c) *$\theta'(t^i) < 0$ (alors le minimum de θ devrait être plus grand que t^i), donc $t_-^{i+1} = t^i$, choisir t^{i+1}, remplacer i par $i+1$ et aller en a).*

Remarque 3.1.

- *Pour le choix du pas initial, si f est de classe C^2, à hessien défini positif, alors au voisinage de zéro on a*

$$\theta(t) = \theta(0) + t\theta'(0) + \frac{1}{2}t^2\theta''(0).$$

Une solution optimale de θ au voisinage de zéro implique $\theta'(t) = 0$, donc

$$\theta'(t) = \theta'(0) + t\theta''(0) = 0.$$

Et on peut prendre

$$t^0 = -\frac{\theta'(0)}{\theta''(0)} = -\frac{g(x)^\top d}{d^\top H(x)d} > 0. \tag{3.1}$$

- *S'il existe un $N \in \mathbb{N}$ telle que $t_+^N \neq 0$ alors, il est claire que $\{t_+^i\}_{i\in\mathbb{N}}$ serait une suite décroissante, $\{t_-^i\}_{i\in N}$ est une suite croissante et $t_-^i < t_+^i, \forall i \geq N$*

Dans ces conditions et pour que l'algorithme de recherche linéaire s'arrête après un nombre fini d'itérations, le test doit satisfaire les propriétés suivantes.

3.1.1 Propriétés

1) Les ensembles suivants constituent une partition de $\mathbb{R}_+$

$$\left\{t \in \mathbb{R}_+ : \theta'(t) = 0\right\}, \left\{t \in \mathbb{R}_+ : \theta'(t) > 0\right\}, \left\{t \in \mathbb{R}_+ : \theta'(t) < 0\right\}.$$

2) $\exists N \in \mathbb{N}$ telle que $t_+^N \neq 0$ $(\theta'(t_+^N) > 0)$, sinon $t_+^i \to +\infty$.

3) Tout intervalle $[t_-^i, t_+^i]$ contient un t^* tell que $\theta'(t^*) = 0$ pour éviter le cas : $\left|t_-^i - t_+^i\right| \to 0$.

3.1.2 Remarques sur le calcul de t^{i+1}

1) La recherche linéaire est caractérisé par le choix de t^{i+1}, pour cela considérons les deux cas suivants :

- Cas $t_+^{i+1} = 0$ (extrapolation)
  - On peut prendre $t^{i+1} = \alpha t_-^{i+1}$ avec α fixe par exemple $\alpha = 10$
- Cas $t_+^{i+1} \neq 0$ (interpolation)
  - Donc, $t^{i+1} \in]t_-^{i+1}, t_+^{i+1}[$, on peut procéder à une simple interpolation par dichotomie, c'est à dire

    $$t^{i+1} = \frac{t_-^{i+1} + t_+^{i+1}}{2}$$

    Ce qui nous donne un intervalle de confiance (où d'incertitude) divisé par deux pour chaque itération.

2) De coutume, il est préférable d'orienter le choix de t^{i+1}. Pour cela l'idée générale est de construire un polynôme d'interpolation à partir des points déjà testés puis minimiser ce dernier.

Théorème 3.1. *La recherche linéaire définie par l'algorithme 3.1 et satisfaisant les propriétés 3.1.1 se termine après un nombre fini d'itérations.*

Démonstration. En utilisant le fait que si la recherche linéaire est infinie, $\left|t_-^i - t_+^i\right|$ tendrait vers 0 et on aurait une contradiction avec la propriété 3) de 3.1.1. □

Remarque 3.2. *Pour assurer la convergence de l'algorithme 3.1 dans le cas par exemple d'une interpolation par un polynôme, une fois calculé, le pas t^{i+1} doit être ramené à l'intérieur de $[t_-^{i+1}, t_+^{i+1}]$ en remplaçant successivement t^{i+1} par*

$$\min\left\{t^{i+1}, t_+^{i+1} - \alpha(t_+^{i+1} - t_-^{i+1})\right\},$$

puis par

$$\max\left\{t^{i+1}, t_-^{i+1} + \alpha(t_+^{i+1} - t_-^{i+1})\right\},$$

où $\alpha = \frac{1}{10}$ (par exemple).

La question du choix de t^{i+1} étant régler, nous allons se pencher sur les tests a), b) et c) de l'algorithme 3.1.

3.2 Recherche linéaire exacte

Historiquement (Cauchy 1847 [7]), On cherché une valeur t^* telle que

$$\theta(t^*) < \theta(0) \text{ et } \theta'(t^*) = 0. \tag{3.2}$$

Sous ces conditions, on peut dresser l'algorithme suivant

Algorithme 3.2.

Etape 0 (Initialisation) :

$i = 0$, $t^0 > 0$, $t_-^0 = 0$, $t_+^0 = 0$, $\varepsilon > 0$ *(suffisament petit),*

Etape 1 :

si $|\theta'(t^i)| \leq \varepsilon$, *stop* $t^* = t^i$

si $\theta(t^i) \geq \theta(0)$ *où* $\theta'(t^i) > \varepsilon$, *alors*

$t_+^{i+1} = t^i$, $t_-^{i+1} = t_-^i$, *on va à l'étape 2*

si $\theta(t^i) < \theta(0)$ *et* $\theta'(t^i) < -\varepsilon$, *alors*

$t_-^{i+1} = t^i$, $t_+^{i+1} = t_+^i$, *on va à l'étape 2*

Etape 2 :

si $t_+^{i+1} = 0$ *déterminer* $t^{i+1} \in]t_-^{i+1}, +\infty[$

si $t_+^{i+1} \neq 0$ *déterminer* $t^{i+1} \in]t_-^{i+1}, t_+^{i+1}[$

remplacer i *par* $i+1$ *et aller à l'étape 1*

On peut noter que ce test est plus raffiné que l'algorithme 3.2 (ce dernier n'assurait même pas la décroissance de θ).

Théorème 3.2. *Supposant que* θ *est de classe* $C^1, \theta'(0) < 0$ *et* $\exists t^* \in \mathbb{R}_+$ *tel que,* $\theta(t) \geq \theta(t^*)$, $\forall t \in \mathbb{R}_+$ *alors l'algorithme 3.2 est fini.*

Démonstration.

Puisque il existe $t^* \in \mathbb{R}_+$ tel que $\theta(t) \geq \theta(t^*)$, pour tout $t \in \mathbb{R}_+$ alors $\exists N \in \mathbb{N}^*$, $t_+^N \neq 0$. De même, on va montrer qu'il existe $N \in \mathbb{N}^*$ tel que $t^* \in]t_-^N, t_+^N[$. Par l'absurde, supposons que $\{t_-^i\}$ et $\{t_+^i\}$ sont des suites adjacentes avec

$$\lim t_+^i = \lim t_-^i = t^*$$

θ est continue et $\theta(t_+^i) < \theta(0)$ donc,

$$\theta(t_-^i) \to \theta(t^*) \leq \theta(0)$$

la même chose pour θ' donne

$$\theta'(t^*) \leq -\varepsilon \leq -\frac{\varepsilon}{2} < 0.$$

Le développement de Taylor pour $t^i_+ - t^*$ petit :

$$\theta(t^i_+) = \theta(t^*) + \theta'(t^*)(t^i_+ - t^*) \leq \theta(t^*) - (\varepsilon/2)(t^i_+ - t^*) < \theta(t^*) < \theta(0)$$

et pour que b) soit vérifié il faut que $\theta'(t^i_+) > \varepsilon$, donc

$$\begin{cases} \theta'(t^*) \leq -\varepsilon \\ \theta'(t^*) \geq \varepsilon \end{cases}$$

d'où une contradiction avec la continuité de θ'. □

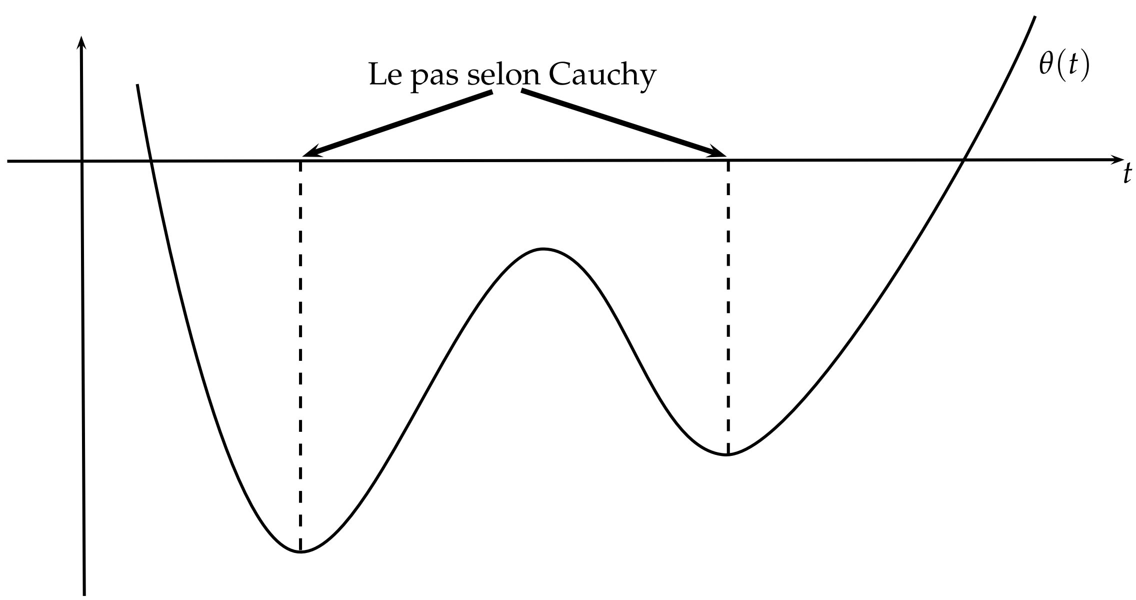

Figure 3.1 – Recherche d'un pas optimal.

3.3 Recherche linéaire inexacte

Jusqu'à maintenant on s'efforce d'identifier un minimum local de $\theta(t)$ pour $t > 0$. Cela conduit à un long algorithme, actuellement il est reconnu que ceci est pratiquement inutile. On préfère utiliser une recherche linéaire dite inexacte ou économique, où l'objectif est de

1) Faire décroitre θ suffisamment.

2) Empêcher le pas optimal t^* d'être trop petit.

Le premier objectif se traduit le plus souvent en forçant la réalisation d'une inégalité de la forme

$$\theta(t^*) \leq \theta(0) + \nu,$$

où ν est un terme négatif.

Le second objectif est motivé par le fait que la décroissance est assurée pour des valeurs proche de zéro (vue la continuité de θ').

3.3.1 La recherche linéaire de Goldestein

Cette méthode est commode lorsqu'on veut pas calculer θ'. On choisit deux coefficients $0 < m_1 < m_2 < 1$, le pas t^* "optimal" doit satisfaire aux conditions

$$\begin{cases} \theta(t^*) \leq m_1\theta'(0)t^* + \theta(0), \\ \theta(t^*) \geq m_2\theta'(0)t^* + \theta(0), \end{cases} \tag{3.3}$$

dites conditions de Goldestein [15].

Algorithme 3.3.

Etape 0 (Initialisation) : $i = 0,\ t^0 > 0,\ t_-^0 = 0,\ t_+^0 = 0,\ 0 < m_1 < m_2 < 1$

Etape 1 : *si* $m_2\theta'(0)t^i + \theta(0) \leq \theta(t^i) \leq m_1\theta'(0)t^i + \theta(0)$, *stop.* $t^* = t^i$

si $\theta(t^i) > m_1\theta'(0)\ t^i + \theta(0)$, *alors*

$t_+^{i+1} = t^i,\ t_-^{i+1} = t_-^i$, *on va à l'étape 2*

si $\theta(t^i) < m_2\theta'(0)t^i + \theta(0)$, *alors*

$t_-^{i+1} = t^i,\ t_+^{i+1} = t_+^i$, *on va à l'étape 2*

Etape 2 : *si* $t_{+}^{i+1}=0$ *déterminer* $t^{i+1}\in]t_{-}^{i+1},+\infty[$

si $t_{+}^{i+1}\neq 0$ *déterminer* $t^{i+1}\in]t_{-}^{i+1},t_{+}^{i+1}[$

remplacer i *par* $i+1$ *et aller à l'étape 1.*

Théorème 3.3. *Supposons que* θ *est de classe* $C^1, \theta'(0)<0$ *et* $\exists t^*\in \mathbb{R}_+$ *tel que,* $\theta(t)\geq\theta(t^*)$*,* $\forall t\in\mathbb{R}_+$ *alors la méthode de Goldestein est finie.*

Démonstration.

Puisque $\exists t^*\in\mathbb{R}_+$ tel que $\theta(t)\geq\theta(t^*), \forall t\in\mathbb{R}_+$ alors $\exists N\in\mathbb{N}$, $t_{+}^{N}\neq 0$ et de la remarque 3.1, $\left\{t_{-}^{i}\right\}$ et $\left\{t_{+}^{i}\right\}$ sont des suites adjacentes avec $lim\ t_{+}^{i}=lim\ t_{-}^{i}=t^*$, θ est continue et $\theta(t_{+}^{i})<\theta(0)$ donc, de b) on a

$$\theta(t_{+}^{i})>m_1t_{+}^{i}\theta'(0)+\theta(0)\Rightarrow\theta(t^*)\geq m_1t^*\theta'(0)+\theta(0),$$

$$\theta(t_{-}^{i})<m_2t_{+}^{i}\theta'(0)+\theta(0)\Rightarrow\theta(t^*)\leq m_2t^*\theta'(0)+\theta(0).$$

Ce qui donne $m_2\leq m_1$ et puisque $\theta'(0)<0\Rightarrow$ une contradiction avec les hypothèses.

□

Dans ce test on demande à t^*soit de façon à ce que

$$\theta_1(t^*)\leq\theta(t^*)\leq\theta_2(t^*),\ t>0$$

où

$$\theta_1(t)=m_2\theta'(0)t+\theta(0)$$

$$\theta_2(t)=m_1\theta'(0)t+\theta(0)$$

Ce qui est schématisé par la figure 3.2

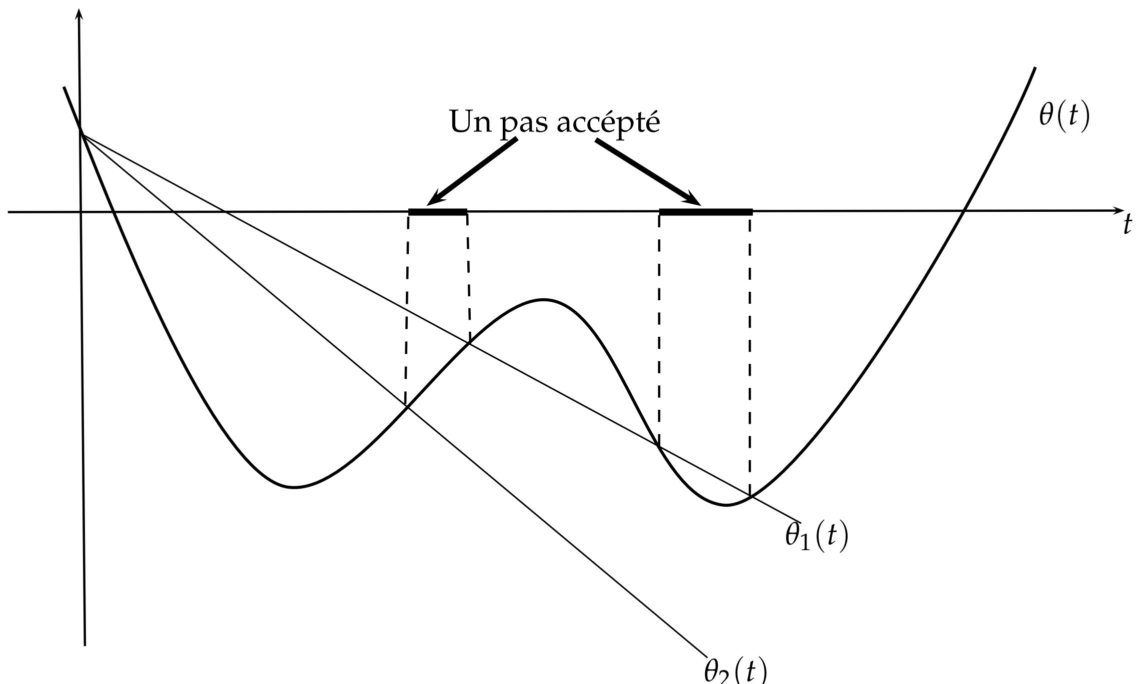

Figure 3.2 – La sélection du pas par la règle de Goldestein.

3.3.2 La recherche linéaire d'Armijo

Cette méthode n'exige pas le calcul de θ'. On choisit $m \in]0,1[$, et on cherche à réaliser une condition dite d'Armijo (ou condition de décroissance linéaire [1]), qui est

$$\theta(t^*) \leq m\theta'(0)t^* + \theta(0). \tag{3.4}$$

Algorithme 3.4.

Etape 0 (Initialisation) : $i = 0, t^0 > 0, 0 < m < 1$

Etape 1 : *si* $\theta(t^i) \leq m\theta'(0)t^i + \theta(0)$, *stop* $t^* = t^i$

si $\theta(t^i) > m\theta'(0)t^i + \theta(0)$, *alors*

$t_+^{i+1} = t^i$, *on va à l'étape 2*

Etape 2 : *Déterminer* $t^{i+1} \in]0, t_+^{i+1}[$

remplacer i par $i+1$ et aller à l'étape 1.

Théorème 3.4. *Supposons que θ est de classe $C^1, \theta'(0) < 0$ et $\exists t^* \in \mathbb{R}_+$, tel que $\theta(t) \geq \theta(t^*)$, $\forall t \in \mathbb{R}_+$ alors la méthode d'Armijo est finie.*

Démonstration.

Pareille à celle du théorème 3.4

□

Remarque 3.3. *L'astuce de cette méthode est de prendre un t^0 raisonnablement grand, par exemple $t^0 = 1$, pour le nouveau pas t^{i+1} on utilise souvent la technique de rebroussement "backtraking" qui consiste à prendre $t^i = \alpha^i$, où $\alpha \in]0,1[$ dans ce cas le pas choisis est : $t^* = \alpha^{\underline{i}}$ et $\underline{i}$ est le plus petit entier tel que l'on ait (3.4). Ce qui nous donne une suite $\{1, \alpha^1, \alpha^2, ...\}$ décroissante d'où le nom de rebroussement.*

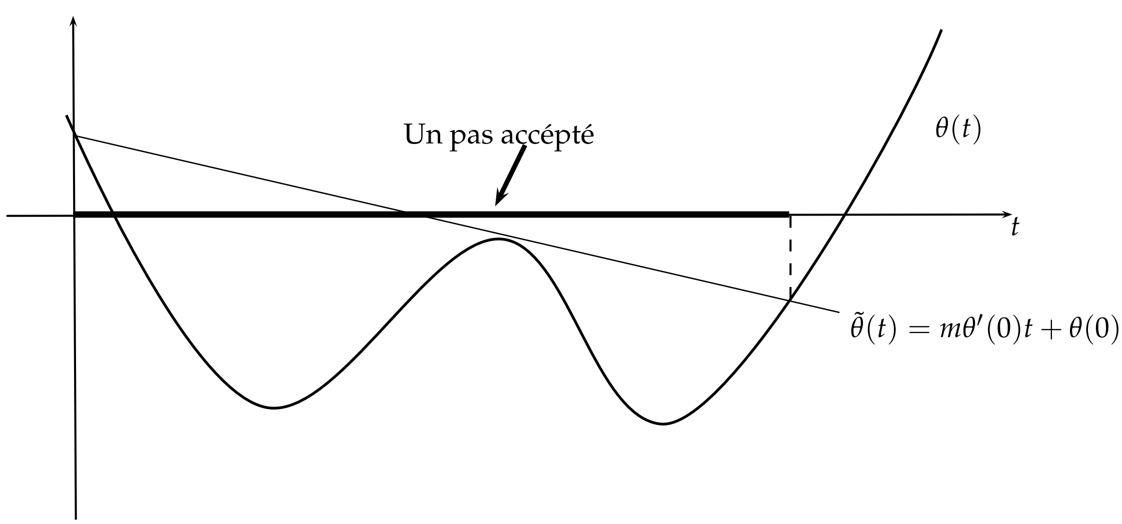

Figure 3.3 – La sélection du pas selon la règle d'Armijo.

3.3.3 La recherche linéaire de Wolfe

C'est la méthode qui semble actuellement la plus efficace, on choisit $0 < m_1 < m_2 < 1$ et le test est une double inégalités, appelées conditions de Wolfe [41]

$$\theta(t^*) \leq m_1 t^* \theta'(0) + \theta(0), \tag{3.5}$$

$$\theta'(t^*) \geq m_2 \theta'(0). \tag{3.6}$$

Algorithme 3.5.

Etape 0 (Initialisation) : $i = 0, t^0 > 0, 0 < m_1 < m_2 < 1,\ t_-^0 = t_+^0 = 0.$

Etape 1 : *si* $\theta(t^i) \leq m_1\theta'(0)t^i + \theta(0)$ *et* $\theta'(t^i) \geq m_2\theta'(0)$, *stop* $t^* = t^i$

si $\theta(t^i) > m_1\theta'(0)t^i + \theta(0)$, *alors*

$t_+^{i+1} = t^i, t_-^{i+1} = t_-^i$ *on va à l'étape 2*

si $\theta(t^i) \leq m_1\theta'(0)t^i + \theta(0)$ *et* $\theta'(t^i) < m_2\theta'(0)$, *alors*

$t_-^{i+1} = t^i, t_+^{i+1} = t_+^i$ *on va à l'étape 2*

Etape 2 : *si* $t_+^{i+1} = 0$ *déterminer* $t^{i+1} \in]t_-^{i+1}, +\infty[$

si $t_+^{i+1} \neq 0$ *déterminer* $t^{i+1} \in]t_-^{i+1}, t_+^{i+1}[$

remplacer i *par* $i+1$ *et aller à l'étape 1.*

Remarque 3.4. *Cette méthode respecte bien le principe d'une recherche linéaire inexacte. D'une part, on assure la décroissance de* θ *puisque*

$$\frac{\theta(t^*) - \theta(0)}{t^*} \leq m_1\theta'(0) < 0 \Rightarrow \theta(t^*) < \theta(0).$$

Et d'autre part du fait que $\theta'(t^*) \geq m_2\theta'(0)$, *la dérivée a suffisamment augmenté (donc* x_{k+1} *ne sera pas trop proche de* x_k*).*

Théorème 3.5. *Supposant que : θ est de classe C^1, bornée inférieurement et $\theta'(0) < 0$. Alors la recherche linéaire de Wolfe est finie.*

Démonstration.

$\exists N \in \mathbb{N}$ tel que $t^i_+ \neq 0$ sinon $t^i \to +\infty$, avec

$$\theta(t^i) \leq \theta(0) + m_1 t^i \theta'(0).$$

Donc $\theta(t^i)$ tendrait vers $-\infty$ d'où une contradiction avec les hypothèses

Les suites $\{t^i_-\}$ et $\{t^i_+\}$ sont des suites adjacentes avec

$$\lim t^i_+ = \lim t^i_- = t^*$$

$$\theta(t^i_+) > m_1 t^i_+ \theta'(0) + \theta(0) \Rightarrow \theta(t^*) \geq m_1 t^* \theta'(0) + \theta(0)$$

$$\theta(t^i_-) \leq m_1 t^i_+ \theta'(0) + \theta(0) \Rightarrow \theta(t^*) \leq m_1 t^* \theta'(0) + \theta(0)$$

Ce qui donne

$$\theta(t^*) = m_1 t^* \theta'(0) + \theta(0)$$

Et puisque $\theta(t^i_-) < m_2 \theta'(0)$ alors

$$\theta(t^*) \leq m_2 \theta'(0)(t)$$

D'autre part pour tout t^i_+

$$\begin{aligned} \theta(t^i_+) &> \theta(0) + m_1 \theta'(0)(t^* + t^i_+ - t^*) = \\ &= \theta(0) + m_1 \theta'(0)(t^i_+ - t^*) + m_1 \theta'(0) t^*. \end{aligned}$$

Donc

$$\theta(t^i_+) > \theta(t^*) + m_1 \theta'(0)(t^i_+ - t^*),$$

ce qui entraine $[\theta(t^i_+) - \theta(t^*)]/(t^i_+ - t^*) \geq m_1 \theta'(0)$. Passant à la limite, on obtient

$$\theta'(t^*) \geq m_1 \theta'(0) > m_2 \theta'(0).$$

Ce qui contredit contredit les hypothèses sur m_1 et m_2. □

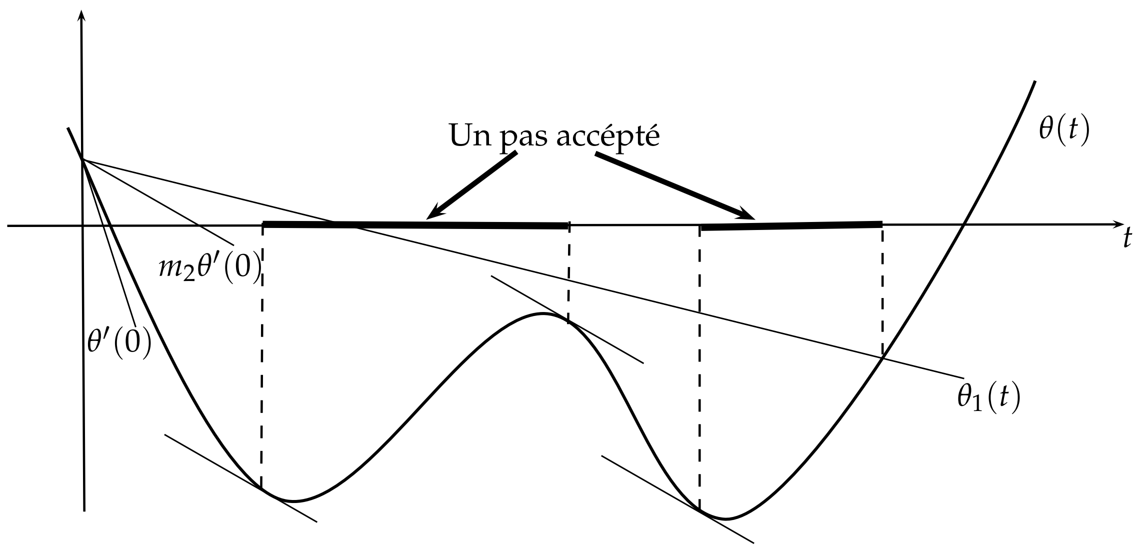

Figure 3.4 – La sélection du pas selon la règle de Wolfe.

3.4 La convergence des méthodes avec recherche linéaire

Pour démontrer la convergence de certain algorithme il est parfois nécessaire d'exiger des conditions plus stricte que (3.6), dans la règle de Wolfe forte [6, 28], (3.6) est remplacé par

$$\left|\theta'(t^*)\right| \leq m_2 \left|\theta'(0)\right|. \tag{3.7}$$

Le théorème suivant, montre qu'il convient de choisir le coefficient de descente (m pour Armijo, m_1 pour Goldestein et Wolfe) inférieur à $\frac{1}{2}$, et dans la règle de Goldstein il faudrait mieux choisir $m_2 > \frac{1}{2}$.

Théorème 3.6. *Supposons θ quadratique avec un minimum t^*. Alors*

$$\theta(t^*) = \theta(0) + \frac{1}{2}\theta'(0)t^*.$$

C'est-à-dire si θ est quadratique, le pas optimale $t^* = 1$ sera refusé par le test de descente si $m_1 > \frac{1}{2}$. On doit donc prendre

$$0 < m_1 < \frac{1}{2} < m_2 < 1. \tag{3.9}$$

Remarque 3.5.

- *Nous avons définie plusieurs méthodes de recherche linéaire, ce qui permet de calculer x_{k+1} à partir de x_k suivant le schéma itératif déjà défini, donc*

$$x_{k+1} = x_k + t_k d_k,$$

où d_k est une direction et t_k un pas choisit par une recherche linéaire. On peut se demander si la suite ainsi générée converge vers une solution du problème (P). Bien entendu que cette propriété ne peut pas avoir indépendamment du choix de d_k, la recherche linéaire sera impuissante si d_k est "trop orthogonale" à $g(x_k)$. Ainsi on défini l'angle φ_k entre $-g(x_k)$ et d_k par

$$c_k = \cos(\varphi_k) = -\frac{g(x_k)^\top d_k}{\|g(x_k)\| \, \|d_k\|} \tag{3.10}$$

et d'après le théorème 2.1 d_k serait une direction de descente si et seulement si $c_k > 0$.

- *Dans le chapitre 1, on a vu un théorème essentiel de convergence (théorème 1.1), ce théorème couvre un domaine très large d'algorithmes. Sachant que notre but est l'étude des méthodes utilisant essentiellement des directions de descentes, avec des recherches linéaires qui sont dans la plus part du temps inexactes, on utilise souvent le théorème suivant.*

Théorème 3.7 (Zoutendijk)**.** *Supposons que g (le gradient de f est Lipchitzien sur la tranche $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. Soit un algorithme dans laquel la direction est telle que $\sum c_k^2$ diverge, et en utilisant la règle de Wolfe. Alors*

- *Soit $\theta(t_k) \to -\infty$.*
- *Soit $\liminf \|g(x_k)\| = 0$.*

Démonstration.

Le premier cas est vrai si f n'est pas bornée inférieurement donc $t_k \to +\infty$ et $\theta(t_k) \to -\infty$. Sinon la règle de Wolfe se termine après un nombre finie d'itération (voir théorème 3.5).

Donc on a $f(x_k) < f(x_0)$ et $g(x_k)$ est lipschitzien dans $\mathcal{L}$, $\forall k \in \mathbb{N}$ et soit t_k le pas optimal donné par Wolfe à l'itération k. De (3.4)

$$t_k\theta'(0) = t_k g(x)^\top d_k = -t_k c_k \|d_k\| \|g(x_k)\| = -c_k \|x_{k+1} - x_k\| \|g(x_k)\| .$$

Et le test de descente entraîne directement

$$\theta(t_k) \leq \theta(0) + m_1 t_k \theta'(0) \Leftrightarrow f(x_{k+1}) \leq f(x_k) - m_1 c_k \|x_{k+1} - x_k\| \|g(x_k)\| .$$

Donc

$$f(x_k) - f(x_{k+1}) \geq m_1 c_k \|x_{k+1} - x_k\| \|g(x_k)\| . \tag{*}$$

La règle de Wolfe, donne

$$\theta'(t_k) \geq m_2 \theta'(0),$$

donc

$$\begin{aligned} g(x_{k+1})^\top d_k &\geq m_2 g(x_k)^\top d_k \Rightarrow \\ (g(x_{k+1}) - g(x_k))^\top d_k &\geq (1 - m_2) g(x_k)^\top d_k = (1 - m_2) c_k \|g(x_k)\| \|d_k\| . \end{aligned}$$

La formule de *Cauchy-Schwarz* et le fait que $g(x_k)$ est lipchitzien sur $\mathcal{L}$ entraine l'existence d'une constante L (dépendant de x_0) telle que

$$(g(x_{k+1}) - g(x_k))^\top d_k \leq \|g(x_{k+1}) - g(x_k)\| \|d_k\| \leq L \|d_k\|^2 ,$$

donc

$$(1 - m_2) c_k \|g(x_k)\| \leq L \|d_k\| . \tag{**}$$

On déduit des deux arguments (*) et (**) que (avec $r = \dfrac{m_1(1-m_2)}{L}$)

$$rc_k^2 \|g(x_k)\|^2 \leq f(x_k) - f(x_{k+1})$$

donc

$$r\sum_{i=0}^{i=k} c_i^2 \|g(x_i)\|^2 \leq f(x_0) - f(x_{k+1}).$$

Si f est bornée inférieurement (et puisque $\{f(x_k)\} \searrow$), alors

$$\sum c_i^2 \|g(x_i)\|^2 < \infty.$$

Et si de plus on suppose que $\|g(x_k)\| \geq c > 0, \forall i \in \mathbb{N}$ implique

$$\sum c_i^2 < \infty,$$

contrairement aux hypothèses. □

Remarque 3.6. .

- *On démontre dans [28] que le théorème 3.7 reste valable si on utilise une recherche linéaire vérifiant ce qu'on appelle condition de Zoutendijk : Il existe une constante $M > 0$ telle que pour tout $k \geq 1$ on ait*
$$f(x_{k+1}) \leq f(x_k) - M\|g(x_k)\|^2 c_k. \tag{3.11}$$
- *On note que les recherches linéaires exactes ainsi que la recherche linéaire du type Armijo et Goldestein vérifie la condition (3.11).*

Exemple 3.1. *Nous allons tester l'algorithme "steepest descent" avec différentes recherches linéaires, comme fonction test on prend : la vallée - banane - de Rosenbrock. Il sagit d'une fonction à deux* *variables du quatrième degré qui se présente comme une vallée en U dont le fond assez plat est incurvé suivant une parabole.*

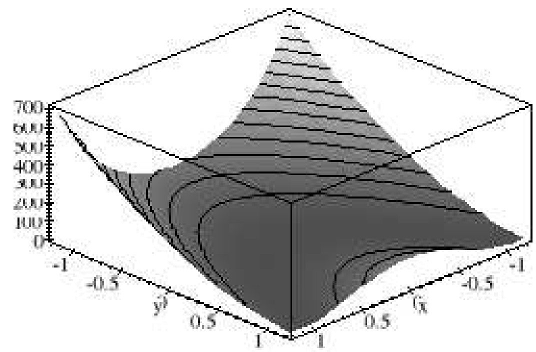

Figure 3.5 – La fonction de Rosenbrock ($f(x,y) = 100 \cdot (x^2 - y)^2 + (x-1)^2$)

La figure 3.6 situe le point solution $(x,y) = (1,1)$ *du problème de minimisation de la fonction de Rosenbrock, le point de départ qui sera systématiquement utilisé* $(x,y) = (-1.2,1)$, *ainsi que le acheminement typique de l'algorithme Steepest descent avec une recherche linéaire.*

Les résultats obtenus sont illustrés dans les tableaux 3.1, 3.2,3.3 et 3.4.

k	t_k	x_k	$f(x_k)$	$\|g(x_k)\|$
0	–	$(-1.20, 1.000)$	24.200	232.867
10	4.777×10^{-03}	$(-1.009, 1.019)$	4.037	2.928
100	3.811×10^{-03}	$(-0.771, 0.596)$	3.138	3.092
200	8.999×10^{-03}	$(0.147, 0.018)$	0.727	1.642
500	1.634×10^{-03}	$(0.807, 0.650)$	3.726×10^{-02}	0.220
1000	1.243×10^{-03}	$(0.944, 0.891)$	3.095×10^{-03}	5.547×10^{-02}
1500	1.161×10^{-03}	$(0.981, 0.962)$	3.590×10^{-04}	1.826×10^{-02}
2000	1.136×10^{-03}	$(0.993, 0.986)$	4.538×10^{-05}	6.4229×10^{-03}

Tableau 3.1 – Steepest descent avec recherche linéaire exacte (dichotomie).

k	t_k	x_k	$f(x_k)$	$\|g(x_k)\|$
0	–	$(-1.200, 1.000)$	24.200	232.867
10	2.499×10^{-03}	$(-1.016, 1.035)$	4.067	3.239
100	4.999×10^{-03}	$(0.360, 0.123)$	0.413	1.399
200	2.499×10^{-03}	$(0.683, 0.466)$	0.100	0.409
500	2.499×10^{-03}	$(0.930, 0.865)$	4.841×10^{-03}	0.124
1000	1.250×10^{-03}	$(0.976, 0.953)$	5.501×10^{-04}	2.304×10^{-02}
1500	2.499×10^{-03}	$(0.988, 0.977)$	1.225×10^{-04}	1.390×10^{-02}
2000	4.999×10^{-03}	$(0.994, 0.989)$	2.838×10^{-05}	1.037×10^{-02}

Tableau 3.2 – Steepest descent avec recherche linéaire d'Armijo.

k	t_k	x_k	$f(x_k)$	$\|g(x_k)\|$
0	–	$(-1.200, 1.000)$	24.200	232.867
10	9.024×10^{-03}	$(-0.939, 0.878)$	3.765	5.702
100	8.573×10^{-04}	$(-0.483, 0.248)$	2.221	2.956
200	8.573×10^{-04}	$(0.386, 0.149)$	0.376	1.221
500	8.573×10^{-04}	$(0.737, 0.543)$	6.894×10^{-02}	0.428
1000	9.024×10^{-03}	$(0.906, 0.821)$	8.749×10^{-03}	0.142
1500	8.573×10^{-04}	$(0.989, 0.978)$	1.117×10^{-04}	1.356×10^{-02}
2000	8.573×10^{-04}	$(0.998, 0.996)$	2.716×10^{-06}	1.478×10^{-03}

Tableau 3.3 – Steepest descent avec recherche linéaire de Goldestein.

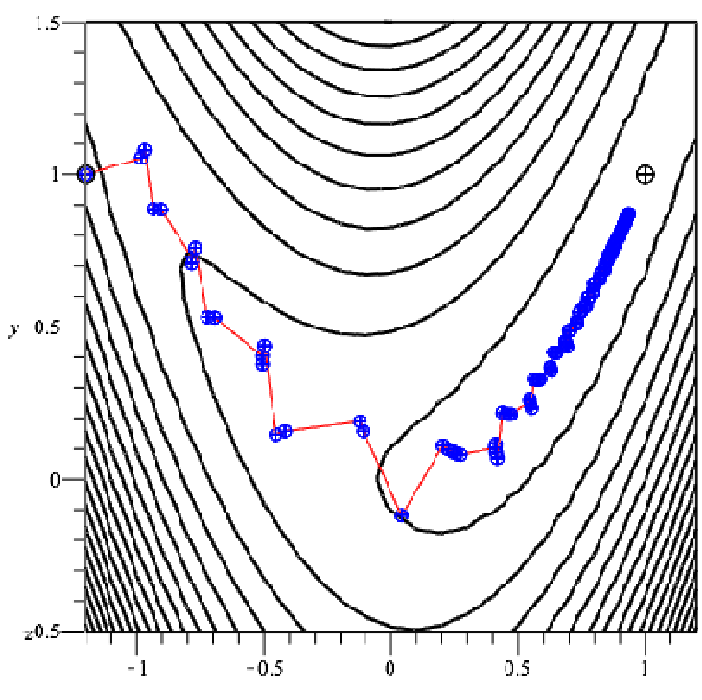

Figure 3.6 – Acheminement typique de l'algorithme Steepest descent avec une recherche linéaire.

k	t_k	x_k	$f(x_k)$	$\|g(x_k)\|$
0	$-$	$(-1.200, 1.000)$	24.200	232.867
10	2.499×10^{-03}	$(-1.016, 1.03)$	4.067	3.239
100	4.999×10^{-03}	$(0.593, 0.347)$	0.166	0.762
200	2.499×10^{-03}	$(0.716, 0.511)$	8.059×10^{-02}	0.455
500	2.000×10^{-02}	$(0.927, 0.859)$	5.380×10^{-03}	0.332
1000	9.999×10^{-03}	$(0.975, 0.951)$	5.919×10^{-04}	8.178×10^{-02}
1500	9.999×10^{-03}	$(0.988, 0.977)$	1.279×10^{-04}	1.259×10^{-02}
2000	1.250×10^{-03}	$(0.994, 0.989)$	2.887×10^{-05}	6.108×10^{-03}

Tableau 3.4 – Steepest descent avec recherche linéaire de Wolfe.

Exercices

Exercice 3.1. *Montrer que si θ est quadratique avec un minimum t^*. Alors*

$$\theta(t^*) = \theta(0) + \frac{1}{2}\theta'(0)t^*.$$

Exercice 3.2. *Soit $f : \mathbb{R}^2 \to \mathbb{R}$, une fonction différentiable. On suppose que $x^* \in \mathbb{R}$ est un minimum local le long de n'importe quelle ligne qui passe par x^*, autrement, $\alpha = 0$ est un minimum local de la fonction définie par $\theta(\alpha) = f(x^* + \alpha d),\ \forall d \in \mathbb{R}^n$.*

1) Montrer que $g(x^*) = 0$.

2) Est-ce que x^* est un minimum local pour f ?

Exercice 3.3. *Soit le problème de minimisation de la fonction $f : \mathbb{R} \longrightarrow \mathbb{R}$ donnée par*

$$f(x) = x^2 - \frac{x^3}{3}.$$

Supposons qu'on utilise un l'algorithme

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

avec un pas $\alpha = 1/2$ et une condition initiale $x_0 = 1$.

1) Montrer que cet algorithme converge vers un minimum local de f.

2) Trouver la vitesse de convergence de cet algorithme.

Exercice 3.4. *Nous voulons minimiser la fonction définie par*

$$f(x\,,\,y) = 10\,x^2 + 5\,x\,y + 10\,(y-3)^2.$$

Trouver le minimum de cette fonction en utilisant la méthode de la plus profonde pente (steepest descent), avec une recherche linéaire de Cauchy et partant du point initial $(x_0, y_0) = (10\,,\,15)$ (prenez une tolérance de l'ordre de 10^{-3}).

Exercice 3.5. *Pour la fonction définie par*

$$f(x, y) = x^2 + 2\,y^2 + 4\,x + 4\,y.$$

1) Montrer que la méthode "steepest desent" appliquée avec le point initial $(x_0, y_0) = (0, 0)$ génère une suite $\{x_k\}$ telle que

$$x_{k+1} = \left(\frac{2}{3^k} - 2\,,\,\left(-\frac{1}{3}\right)^k - 1\right)^\top.$$

2) Déduiser le minimum de f.

Exercice 3.6. *Montrer que si f est une fonction quadratique convexe, $f(x) = \dfrac{1}{2}x^\top Q\,x + b^\top x + c$. Alors la solution du problème de recherche linéaire*

$$\min_{t>0}\, f(x_k + td_k),$$

peut être calculée de manière analytique et est donnée par

$$t_k = -\frac{g(x_k)^\top d_k}{d_k^\top Q d_k}.$$

Bibliographie

[1] L. Armijo. Minimization of functions having Lipschitz continuous first derivatives. Pacific Journal of Mathematics 16 (1966) 1-3

[2] M. S. Bazara. H.D. Sherali, C.M Shetty. Nonlinear programming. J.W&S (1993)

[3] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal and C. A. Sagastizábal. Numerical optimization : theoretical and practical aspects. Springer Science & Business Media (2006).

[4] C. G. Broyden. Quasi-Newton method and their application to function minimization, Mathematics of Computation, 21,(1967).

[5] A. Bukley, A.Lenir. QN-like variable storage conjugate gradients. Math. Prog. 27,(1983)155-175.

[6] R. Byrd and J. Nocedal. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization, SIAM Journal on Numerical Analysis, 26 (1989) 727-739.

[7] A. Cauchy. Méthodes générales pour la résolution des systèmes d'équations simultanées. Compte rendus Acad. Sc. Paris, Tome 25 (1847) 536-538.

[8] F. Chatelin. Eigenvalues of Matrices, J.Wiley & S (1993).

[9] W. C. Davidon. Variable metric methods for minimization, AEC Research Development, Report ANL-5990 (1959).

[10] J. E. Denis, Jr., J.J. Moré. A characterisation of superlinear convergence and its application to quasi-Newton methods, Mathematics of computation, 28 (1974) 549-560.

[11] J. E. Dennis, H. Wolkowitcz. Sizing and Least Change Secant Methods, CORR Report 93-11 (1993).

[12] R. Fletcher. A new approch to variable metric algorithms, Computer Journal, 13,(1970)317-322.

[13] R. Fletcher, M.J.D. Powell. A Rapidly convergent descent method for minimisation, Computer Journal 6,(1963)163-168.

[14] R. Fletcher. An overview of Unconstrained Optimization in Algorithms for Continuous Optimization, The State of the Art, E. Spedicato, ed., Kluwer academic Publishers, Boston, (1994)109-143.

[15] A. A. Goldestein, J.F. Price. An effective algorithm for minimization. Num. Math. 10,(1967)184-189.

[16] D. Goldfarb. A falmily of metric variablee methods derived by variation means, mathematics Of computation 24,(1970)23-26.

[17] A. Griewank, Ph.L. Toint. Local convergence analysis of partitioned quasi-Newton udaptes, Numeric Mathematics 39,(1982)429-448.

[18] T.G. Kolda. Limited-Memory matrix methods with application, Applied Mathematics program, university of Maryland (1998).

[19] A. S.Lewis. Convex analysis on the hermitian matrices, Technical report CORR 97-09, University of Waterloo (1994).

[20] D. H. Li, M. Fukushima. A global convergence of BFGS methods for nonconvex unconstrained optimization problems, SIAM Journal on Numerical Analysis.7, (1999).

[21] D. H. Li, M. Fukushima. A modified BFGS method and its global convergence in noncovex minimization, Technical Repport 98003, Département of Applied Mathematics and Physics, Kyoto University, (1998).

[22] D. G. Luenberger. Linear and Nonlinear Programming, 2nd edition, Addison-Wesley (1984).

[23] P. E. Gill, W. Murray. Conjugate gradient method for large-scale nonlinear optimization, Technical report SOL 79-15, Dept. of Operation Res. Stanford University, (1979).

[24] J. J. Moré, B.S. Garbow, K.E. Hillstrome. Testing unconstrained optimization softwear, ACM Trans. Math. Softwear, 7, (1981)17-41.

[25] L. . Conjugate gradient methods less dependent on conjugacy. SIAM Review 28,(1986)501-511.

[26] Y. E. Nestrov. On the a approach to the construction of optimal methods of minimization of smooth convex function, Economica I Matem. 24, (1988)504-517.

[27] J. Nocedal. Udapting quasi-Newton matrice with limited storage, Mathemating of computation 35, (1980)773-782.

[28] J. Nocedal. Theory of algorithms for unconstrained optimization, acta Numerica, (1999)199-242.

[29] J. Nocedal, J. W. Stephen. Numerical optimization. Springer (1999).

[30] D. P. O'Leary. A discrete Newton algorithm for minimizing a function of many variable, Math. Prog. 23, (1982)20-30.

[31] S. S. Oren, E. Spedicato, Optimal condition of self-scaling variable metric algorithms, Math. Prg., 10, (1976)70-90.

[32] J. M. Perry. A class of conjugate gradient with two-step variable-metric memory, Discussion paper 269, C.M.S.E.M.S. Northwestern University (Evanston, IL, 1977)

[33] J. D. Pearson. Variable metric methods of minimization. Computer Journal, Vol 12, (1971)21-36.

[34] M. J. D. Powell. On the convergence of the variable metric algorithm, Journal of of the Institute of Mathematics and its Applications, (1971)21-36.

[35] M. J. D. Powell. Some properties of the variable metric algorithm for minimization without exact line searches, in Nonlinear Programming, SIAM-AMS Proceeding, Vol. IX, R.W. Cottle, and C.E. Lemke, eds., SIAM, (1976)35-72.

[36] L. Schwartz. Analyse, topologie générale et analyse fonctionnelle, Hermann (1970).

[37] M. Sibony, J.C. Mardon. Analyse Numérique. Systèmes linéaires et non linéaires, HERMAN (1982).

[38] D. F. Schanno. Conditionning of quasi-Newton methods for function minimization, Mathematics of computation, 24,(1970)641-656.

[39] D. F. Schanno. On the convergence of new conjugate gradient algorithm, SIAM Journal on numerical analysis, 15,(1978a)1247-1257.

[40] T. Steihaug. The conjugate gradient method and trust regions in large-scale optimization, SIAM J. Num. Anal. 20, (1983)626-637.

[41] P. Wolfe. Convergence for ascent methods. SIAM Rev. 11,226-235, 1968.

[42] W. I. Zangwill. Nonlinear programming : a unified approach, Prentice Hall, Englewood Cliffs, RI, 1969.

[43] G. Zoutendijk. Non linear programming, computation programming. In J. Abadie (ed), integer and non linear programming. North Holland (1970)37-86.