Chapitre 3

Méthodes de Quasi-Newton

3.1 Introduction

La méthode de Newton permet de construire un algorithme permettant de résoudre le système d'équations non-linéaires : $g(x) = 0$ où $g : \mathbb{R}^n \to \mathbb{R}^n$ est diférentiable : on se donne $x_0 \in \mathbb{R}^n$ et on fait les itérations

$$x_{k+1} = x_k - (g'(x_k))^{-1} g(x_k) \tag{3.1}$$

où $g'(x)$ est la dérivée (ou jacobienne) de g au point x. L'application de cette méthode au problème d'optimisation $\min_{x \in \mathbb{R}^n} f(x)$ consiste à l'utiliser pour résoudre le système d'optimalité du problème précédent, c'est à dire que l'on pose $g(x) = \nabla f(x)$ dans (3.1) : on obtient les itérations

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k) \tag{3.2}$$

Comme on a vu au chapitre 3, que la méthode de Newton est intéressante car sa convergence est quadratique au voisinage de la solution mais la convergence n'est assurée que si x_0 est suffisamment proche de $\overline{x}$, ce qui en limite l'intérêt. Pour résoudre le problème de convergence locale de la méthode de Newton, on peut penser à lui ajouter une phase

de recherche linéaire, dans la direction

$$d_k = (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

Cela est possible uniquement si d_k est une direction de descente en x_k, soit$-\nabla^\top f(x_k) d_k = -\nabla^\top f(x_k)(\nabla^2 f(x_k))^{-1} \nabla f(x_k) < 0$.

Ce qui sera le cas si $\nabla^2 f(x_k)$ est une matrice définie positive, ce qui n'est pas garanti.Le principe des méthodes que nous allons voir maintenant consiste à remplacer le Hessien $\nabla^2 f(x_k)$ par par une approximation H_k (si possible définie positive), construite au cours des itérations.

Le couplage de la méthode de Newton avec la recherche linéaire de Wolfe a permis de construire une méthode globalement convergente. Les méthodes de quasi-Newton ont été développées pour pallier autres inconvénients de la méthode de Newton : en particulier le problème du calcul de la matrice hessienne qui n'est pas toujours possible ou conseille. Ces méthodes se concentrent donc sur la construction itérative de matrices H_k approchant la hessienne, ou de matrices S_k approchant l'inverse de la hessienne.

Equation de sécante et approximation Comment calculer une approximation H_{k+1} de la matrice hessienne $\nabla^2 f(x_{k+1})$ connaissant x_k et x_{k+1}, $\nabla f(x_k)$ et $\nabla f(x_{k+1})$?

Ecrivons le développement limité de ∇f au voisinage de x_{k+1}et appliqué en x_k :

$$\nabla f(x_k) = \nabla f(x_{k+1}) + \nabla^2 f(x_{k+1})(x_k - x_{k+1}) + o(x_k - x_{k+1}).$$

D'où

$$\nabla^2 f(x_{k+1})(x_k - x_{k+1}) \simeq \nabla f(x_{k+1}) - \nabla f(x_k).$$

On construit une approximation H_{k+1}de $\nabla^2 f(x_{k+1})$comme solution de l'équation :

$$\nabla f(x_{k+1}) - \nabla f(x_k) = H_{k+1}(x_{k+1} - x_k), \tag{3.3}$$

appelée **équation de sécante** ou **équation de quasi-Newton**. De façon similaire, on

peut construire une approximation B_{k+1} de $\nabla^2 f(x_{k+1})^{-1}$comme solution de l'équation :

$$S_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k, \tag{3.4}$$

Dans les deux cas, les équations de quasi-Newton forment un système sous-determiné à n équations et n^2 inconnues. Il existe donc une infinité de matrices H_{k+1} pouvant convenir.

3.2 Principes d'obtention des formules de mise a jour

On s'intéresse à la résolution du problème d'optimisation non-linéaire sans contraintes $\min_{x\in\mathbb{R}^n} f(x)$, en utilisant des algorithmes à direction de descente $x_{k+1} = x_k + t_k d_k$ où le pas t_k est calculé par une recherche linéaire effectuée le long d'une direction de descente d_k. Dans les méthodes de quasi-Newton cette direction est donnée par la formule

$$d_k = -H_k^{-1}\nabla f(x_k). \tag{3.0}$$

Si $H_k = \underset{0}{\nabla^2 f(x_k)}$ on retrouve l'algorithme de Newton. Pour certain problèmes, on cherche à éviter le calcul du hessien ainsi que son inverse car

- Ce calcul demande trop de temps à l'exécution,

-On ne dispose pas de dérivées secondes, et leur calcul est très coûteux,

-La fonction objective n'est pas deux fois dérivable.

Pricinpe : Supposons que H_k est connue et construisons son successeur H_{k+1} qui doit être une approximation de la matrice $\nabla^2 f(x_{k+1})$. Ceci est motive par le désir de donner a l'algorithme une convergence locale rapide .Dans toute la suite, nous noterons

$$\delta_k = x_{k+1} - x_k \quad \text{et} \quad \gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k). \tag{3.5}$$

La formule de taylor avec reste integrale donne

$$\gamma_k = (\int_0^1 \nabla^2 f(x_k - t\delta_k)dt)\delta_k. \tag{3.6}$$

Pour que H_{k+1} soit proche de $\nabla^2 f(x_{k+1})$ it est naturel qu'elle doit vérifier l'équation (3.6) (au moins sa valeur moyenne entre $x_k$et x_{k+1}), c'est-à-dire l'équation dite de "quasi-Newton"

$$\gamma_k = H_{k+1}\delta_k. \tag{3.7}$$

Si $n = 1$: Avec $H_0 = 1$,on retrouve la formule de la secante (regula-falsi)

Si $n > 1$: L'équation matricielle (3.7) à $n \times n$ inconnues et elle ne fournie que n équations linéaires,

• Comme le hessien et symétrique, on impose la symétrie pour H_{k+1} (ce qui est logique pour que $h_{k+1} \approx \nabla^2 f(x_{k+1})$), donc it reste $n^2 - \frac{n^2-n}{2} = \frac{n^2+n}{2}$ inconnues, ce qui laisse le choix pour une infinit2 de matrices "quasi-Newton"

• En se donnant une matrice initiale H_0 (par exemple $H_0 = I$, la matrice identité), et on construit une suite de matrices $\{H_k\}$, par la relation

$$H_{k+1} = H_k + C_k,$$

3.2.1 Formules de Correction de rang un (SR1)

Puisque la matrice hessienne est symétrique it est souhaitable que l'approximation qu'on construit le soit également. Ceci est possible si on utilise une formule de mise à jour de la forme

$$H_{k+1} = H_k + a_k u_k u_k^\top, \tag{3.8}$$

avec u_k un vecteur de $\mathbb{R}^n$ et a_k une constante. On note que la matrice $u_k u_k^\top$ est symétrique puisque $(uu^\top)^\top = uu^\top$ et elle est de rang un car elle est le produit de deux matrices de rang un. Il est facile de calculer a_k et u_k explicitement.

La matrice (3.8) doit vérifier l'équation de quasi-Newton (3.7), donc

$$\gamma_k = (H_k + a_k u_k u_k^\top)\delta_k \Rightarrow \gamma_k - H_k\delta_k = a_k u_k (u_k^\top \delta_k).$$

• Si $\gamma_k - H_k\delta_k = 0$, alors $H_{k+1} = H_k$

• Si $\gamma_k - H_k\delta_k \neq 0$, en prenant $a_k = \frac{1}{u_k^\top \delta_k}$, on a $u_k = \gamma_k - H_k\delta_k$, on remplace dans (3.8) pour trouver

$$H_{k+1} = H_k + \frac{(\gamma_k - H_k\delta_k)(\gamma_k - H_k\delta_k)^\top}{(\gamma_k - H_k\delta_k)^\top \delta_k}, \tag{3.9}$$

qui est la formule de mise à jour SR1 (Symetric Rank 1 [3]).

Remarque 3.1 *Supposons qu'au lieu de B_{k+1}, on veut calculer $S_{k+1} = (H_{k+1})^{-1}$, destinée a approcher $H^{-1}(x_{k+1})$, alors S_{k+1} doit vérifier l'équation "quasi-Newton" dite "duale"*

$$S_{k+1}\gamma_k = \delta_k$$

En intervertissant γ_k et δ_k dans le raisonnement qui précède, on trouve

$$S_{k+1} = S_k + \frac{(\delta_k - S_k\gamma_k)(\delta_k - S_k\gamma_k)^\top}{(\delta_k - S_k\gamma_k)_k^\top \gamma_k}, \tag{3.10}$$

qui est la formule duale de (3.9)

Etape 0 (Initialisation) : x_0 et $\varepsilon > 0$ donnée, $S_0 = I, k = 0$

Etape 1 : Test d'arrêt : calculer $\nabla f(x_K)$, si $\| \nabla f(x_K) \| < \varepsilon$, alors stop sinon

Etape 2 : Calcul de la direction d_k :

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

Etape 3 : Recherche linéaire ; trouver t_k solution du problème $\underset{t>0}{min}\ f(x_k + td_k)$

Etape 4 : Si la recherche linéaire réussie ; $x_{k+1} = x_k + td_k$

Calculer S_{k+1} d 'apres la formule (3.10)

Etape 5 : Remplacer k par $k+1$ et aller en 1.

Proposition 3.1 *Si f est une fonction quadratique de hessien symetrique Q et si $\delta_1, \delta_2, ..., \delta_n$ (définie en* (3.5) *et générée par l'algorithme* (3.0)*, sont linéairement indépendant, alors,*

$$H_{n+1} = Q.$$

Démonstration.

Démontrons par récurrence sur k, que si l'équation (3.7) est vérifiée, alors

$$\gamma_i = H_{k+1}\delta_i \quad \forall i \leq k. \tag{3.11}$$

Pour $k = 0$ la relation (3.11) est vraie d'après la définition de H_2

Supposons que (3.11) est vraie pour $k \succeq 0$ et démontrons quelle reste vraie pour $k+1$, d'après (3.9) on a

$$H_{k+1}\delta_i = H_k\delta_i + \frac{(\gamma_k - H_k\delta_k)(\gamma_k - H_k\delta_k)^\top \delta_i}{(\gamma_k - H_k\delta_k)^\top \delta_k}$$

Grâce a l'hypothèse de récurrence, on obtient

$$(\gamma_k - H_k\delta_k)^\top \delta_i = \gamma_k^\top \delta_i - {\delta_k}^\top H_k\delta_i = \gamma_k^\top \delta_i - {\delta_k}^\top \gamma_i$$

et puisque f est quadratique alors, $Q\delta_i = \gamma_i$,donc

$$(\gamma_k - H_k\delta_k)^\top \delta_i = \gamma_k^\top \delta_i - \delta_k{}^\top Q\delta_i = \gamma_k^\top \delta_i - (Q\delta_k{}^\top)\delta_i = \gamma_k^\top \delta_i - \gamma_k^\top \delta_i = 0,$$

ce qui donne

$$H_{k+1}\delta_i = H_k\delta_i + 0 = \gamma_i \ ,\forall i \leq k-1.$$

Et d'apres (3.7), on peut conclure(3.11), et qui peut être réécrite (avec $k = n$) comme suite :

$$H_{n+1}\Delta = \Gamma,$$

où,$\Delta = [\delta_0, \delta_1,..., \delta_n]$ et $\Gamma = [\gamma_0, \gamma_1, ..., \gamma_n]$.On vérifie aussi

$$Q\Delta = \Gamma,$$

donc

$$Q\Delta = H_{n+1}\Delta \Rightarrow Q = H_{n+1}.$$

Puisque f est quadratique de matrice Q, c'est-à-dire

$$f(x) = \frac{1}{2}x^\top Qx - b^\top x + c, b \in \mathbb{R}^n, c \in \mathbb{R},$$

la condition nécessaire d'optimalité est

$$Qx^* - b = 0 \Rightarrow x^* = Q^{-1}b = S_{n+1}b.$$