1 Directions de descente

Soit f une fonction à minimiser. On note :

- $x^{(k)}$ le point courant,
- $g^{(k)}$ le gradient de f en $x^{(k)}$.

On cherche une direction $d^{(k)}$ dans laquelle la fonction f décroisse à partir du point $x^{(k)}$:

$$\exists \alpha^{(k)} > 0 \ , \quad f(x^{(k)} + \alpha^{(k)} d^{(k)}) < f(x^{(k)}) \ ,$$

inégalité induite par la relation : ${g^{(k)}}^\top d^{(k)} < 0$. Le pas $\alpha^{(k)}$ est en général déterminé par recherche linéaire.

1.1 Formule de Polak-Ribière

Dans les méthodes de type **gradient conjugué**, la direction de descente $d^{(k)}$ au point $x^{(k)}$ est de la forme :

$$d^{(k)} = \begin{cases} -g^{(1)} & \text{si } k = 1 \\ -g^{(k)} + \beta^{(k)} d^{(k-1)} & \text{sinon} \end{cases} \ . \tag{1}$$

Dans la mise en œuvre de Polak-Ribière, le coefficient $\beta^{(k)}$ est donné par la formule :

$$\beta^{(k)} = \frac{{g^{(k)}}^\top \left(g^{(k)} - g^{(k-1)}\right)}{\left\| g^{(k-1)} \right\|^2} \ . \tag{2}$$

1.2 Formule de BFGS inverse

Dans les méthodes de type **quasi-Newton**, la direction de descente est de la forme :

$$d^{(k)} = -W^{(k)} g^{(k)} \ , \tag{3}$$

où $W^{(k)}$ représente une approximation de l'*inverse* de la matrice hessienne $H^{(k)}$ de f au point $x^{(k)}$. Cette approximation est remise à jour à chaque itération de l'algorithme.

Avec les notations suivantes :

- $\delta_x^{(k)} = x^{(k)} - x^{(k-1)}$,
- $\delta_g^{(k)} = g^{(k)} - g^{(k-1)}$,

la formule de remise à jour de BFGS (Broyden-Flechter-Goldfarb-Shanno) est donnée par :

$$W^{(k)} = \left(I - \frac{\delta_x^{(k)} {\delta_g^{(k)}}^\top}{{\delta_g^{(k)}}^\top \delta_x^{(k)}} \right) W^{(k-1)} \left(I - \frac{\delta_g^{(k)} {\delta_x^{(k)}}^\top}{{\delta_g^{(k)}}^\top \delta_x^{(k)}} \right) + \frac{\delta_x^{(k)} {\delta_x^{(k)}}^\top}{{\delta_g^{(k)}}^\top \delta_x^{(k)}} \ . \tag{4}$$

A défaut de meilleure idée, la matrice $W^{(1)}$ initialisant cette formule est prise égale à l'identité.

Remarque 1. Lorsque le pas $\alpha^{(k)}$ est déterminé par la règle de Wolfe, la relation (6b) implique :

$$ {\delta_g^{(k)}}^\top \delta_x^{(k)} \geq \alpha^{(k)} (\omega_2 - 1) {g^{(k)}}^\top d^{(k)} > 0 \ . \tag{5}$$

Si l'on choisit une matrice initiale $W^{(1)}$ définie positive, la formule de mise à jour (4) génère alors des matrice $W^{(k)}$ qui sont aussi définies positives. C'est pourquoi on utilise *toujours* la règle de Wolfe pour effectuer la recherche linéaire dans l'algorithme de BFGS.

2 Recherche linéaire par la règle de Wolfe

2.1 Principes généraux

Soit f une fonction à minimiser. On note :

- $x^{(k)}$ le point courant,
- $g^{(k)}$ le gradient de f en $x^{(k)}$,
- $d^{(k)}$ la direction de descente,
- $\alpha^{(k)}$ le pas de descente.

La règle de Wolfe a pour but de déterminer un pas $\alpha^{(k)}$ vérifiant les deux conditions suivantes :

1. la fonction f doit décroître de manière significative :

$$f(x^{(k)} + \alpha^{(k)} d^{(k)}) \leq f(x^{(k)}) + \omega_1 \alpha^{(k)} {g^{(k)}}^\top d^{(k)} \ , \tag{6a}$$

2. le pas $\alpha^{(k)}$ doit être suffisamment grand :

$$\left(\nabla f(x^{(k)} + \alpha^{(k)} d^{(k)})\right)^\top d^{(k)} \geq \omega_2 {g^{(k)}}^\top d^{(k)} \ , \tag{6b}$$

avec $0 < \omega_1 \ < \omega_2 < 1$ (typiquement, $\omega_1 = 0.1$ et $\omega_2 = 0.9$).

Remarque 2. Il existe une variante appelée **règle de Wolfe forte**, convenant mieux aux algorithmes de type *gradient conjugué non linéaire*, qui consiste à vérifier d'une part la condition (6a) et d'autre part la condition suivante, plus restrictive que (6b) :

$$| \, [\nabla f(x^{(k)} + \alpha^{(k)} d^{(k)})]^\top d^{(k)} \, | \leq \omega_2 | \, [g^{(k)}]^\top d^{(k)} \, | \ .$$

2.2 Algorithme de résolution

La procédure itérative suivante, inspirée d'une mise en œuvre due à *Flechter & Lemaréchal*, permet de manière particulièrement simple de calculer un pas $\alpha^{(k)}$ vérifiant les deux conditions de Wolfe.

L'initialisation consiste à poser $\underline{\alpha} = 0$ et $\overline{\alpha} = +\infty$ et à se donner un pas $\alpha^{(k)} \in]\underline{\alpha}, \overline{\alpha}[$. Puis, on effectue le test suivant pour modifier la valeur de $\alpha^{(k)}$ jusqu'à obtenir un pas vérifiant les conditions de Wolfe :

- si $\alpha^{(k)}$ ne vérifie pas la condition (6a) :
  - on *diminue* la borne supérieure : $\overline{\alpha} = \alpha^{(k)}$,
  - on choisit un nouveau pas : $\alpha^{(k)} = \frac{1}{2}(\underline{\alpha} + \overline{\alpha})$,
- sinon,
  - ⋆ si $\alpha^{(k)}$ ne vérifie pas la condition (6b) :
    - on *augmente* la borne inférieure : $\underline{\alpha} = \alpha^{(k)}$,
    - on choisit un nouveau pas :
      - . $\alpha^{(k)} = 2\underline{\alpha}$ si $\overline{\alpha} = +\infty$,
      - . $\alpha^{(k)} = \frac{1}{2}(\underline{\alpha} + \overline{\alpha})$ sinon,
  - ⋆ sinon, le pas $\alpha^{(k)}$ vérifie la règle de Wolfe.

On notera que les conditions (6a) et (6b) ne sont pas traitées de manière symétrique dans cette procédure. En effet, la seconde condition de Wolfe n'est testée que si la première condition est satisfaite. Par contre, la modification de la borne inférieure $\underline{\alpha}$ dans le cas où la seconde condition n'est pas vérifiée oblige à refaire le test de la première condition.

2.3 Mise en œuvre

2.3.1 Terminaison de l'algorithme

D'un point de vue pratique, et bien que l'on puisse montrer, sous des conditions raisonnables, que l'algorithme de Flechter-Lemaréchal fournit un pas $\alpha^{(k)}$ satisfaisant les deux conditions de Wolfe en un nombre fini d'étapes, il peut arriver que cet algorithme nécessite dans certains cas un très grand nombre d'étapes pour obtenir une valeur $\alpha^{(k)}$ correcte. C'est pourquoi il est souhaitable, lors de sa programmation, de contrôler le nombre maximal d'essais de pas de l'algorithme. De plus, il faut définir un test d'arrêt pour cet algorithme. Cependant, ce test ne doit pas porter sur la longueur de l'intervalle de recherche en α, mais plutôt sur le déplacement en x correspondant. Ceci revient donc à se donner une *résolution en x*, car cette dernière grandeur a une interprétation physique qui devrait permettre à l'utilisateur du code de connaître l'ordre de grandeur de ses variations. Notant $x^{(k,\ell)} = x^{(k)} + \alpha^{(\ell)} d^{(k)}$ le point proposé à la ℓ-ème étape de l'algorithme de Flechter-Lemaréchal, le test d'arrêt portera donc sur la quantité :

$$\|x^{(k,\ell+1)} - x^{(k,\ell)}\| = |\alpha^{(\ell+1)} - \alpha^{(\ell)}| \|d^{(k)}\| \ .$$

2.3.2 Pas initial de l'algorithme

Il reste enfin à déterminer la valeur initiale $\alpha^{(k,0)}$ avec laquelle on initialise cette procédure. Dans le cas d'une utilisation dans le cadre d'un algorithme de type Newton ou quasi-Newton, le pas unité est tout indiqué. Pour d'autres méthodes comme le gradient ou le gradient conjugué, il n'existe pas de pas naturel "évident". Une possibilité consiste à déterminer ce pas en ce basant sur la *décroissance attendue* $\Delta^{(k)}$ de la fonction f. Pour cela, on considère une approximation quadratique de la fonction $\alpha \mapsto f(x^{(k)} + \alpha\, d^{(k)})$, à savoir :

$$\varphi^{(k)}(\alpha) = a_0^{(k)} + a_1^{(k)} \alpha + \frac{1}{2} a_2^{(k)} \alpha^2 \ .$$

Les deux premiers coefficients de cette approximation sont facilement disponibles :

$$\begin{aligned} a_0^{(k)} &= f(x^{(k)}) \ , \\ a_1^{(k)} &= {g^{(k)}}^\top \, d^{(k)} \ . \end{aligned}$$

Plutôt que d'obtenir le dernier coefficient $a_2^{(k)}$ à l'aide de la matrice hessienne de f, on le détermine en imposant que la décroissance maximale de $\varphi^{(k)}$ soit égale à la *décroissance attendue* $\Delta^{(k)}$ de f :

$$\Delta^{(k)} = \varphi^{(k)}(0) - \min_{\alpha \geq 0} \varphi^{(k)}(\alpha) \ ,$$

d'où :

$$a_2^{(k)} = \frac{{a_1^{(k)}}^2}{2\Delta^{(k)}} \ .$$

On en déduit alors le pas initial $\alpha^{(k,0)}$, qui est celui donnant la décroissance maximale de l'approximation quadratique ainsi définie. On obtient :

$$\alpha^{(k,0)} = -\frac{a_1^{(k)}}{a_2^{(k)}} = -\frac{2\Delta^{(k)}}{a_1^{(k)}} = -\frac{2\Delta^{(k)}}{{g^{(k)}}^\top \, d^{(k)}} \ .$$

Ce pas est appelé *pas de Flechter*.

Remarque 3. La détermination du pas initial est donc reportée sur la détermination de la décroissance attendue $\Delta^{(k)}$. Cette dernière valeur peut provenir de considérations directes sur le problème d'optimisation que l'on doit résoudre : par exemple, si l'on dispose d'une bonne approximation d'un minorant de la fonction f, la décroissance attendue pourra être prise égale à une (petite) fraction de l'écart entre $f(x^{(k)})$ et ce minorant.

P. Carpentier, 21 avril 2013.