Optimisation Non linèaire

Par

Professeur Abdellatif El Afia

Chapitre 3

Optimisation sans contraint

- Fonction à plusieurs variables
  - Méthodes itératives de résolution
  - Recherche linéaire approchée.

Optimisation sans contraint : Fonction à plusieurs variables

- $f: \mathbb{R}^n \to \mathbb{R} \Longrightarrow x = (x_1, \dots, x_n)^T$
- $\nabla f(x) vecteur\ d'ordre\ n\ \to \nabla^2 f(x) matrice\ d'ordre\ n \times n$
- $x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} f(x)$
- $\{x^k\}$ tel que
  - Calcule de d^k
  - $x^{k+1} = x^k + \alpha_k d^k \Longrightarrow \alpha_k = \underset{\alpha > 0}{\operatorname{argmin}} \varphi(\alpha)$ Recherche linéaire: $\varphi(\alpha) = f(x^k + \alpha d^k)$(Chapitre2)
- Objectif 1: Type des d^k
  - Gradient $\Leftrightarrow d^k = -\nabla f(x^k)$
  - Gradient Conjugué $\Leftrightarrow \boldsymbol{x^{k+1} = x^k + \alpha_k d^k}$ **avec** $\boldsymbol{\left(d^i\right)^T Q d^j = 0 \quad \forall\, i \neq j \Leftrightarrow x^* = \sum_{k=0}^{n-1} \alpha_k\, d^k}$
  - Newton $\Leftrightarrow d^k = -(\nabla^2 f(x))^{-1} \nabla f(x^k)$
  - Quasi-Newton $\Leftrightarrow d^k = -H_k \nabla f(x^k)$

- Objectif 2: la recherche linéaire approchée

Méthodes itératives de résolution

- **Méthode de gradient**
- **Gradient conjugué**
- **Algorithme de Newton**
- **Algorithme Quasi-Newton**

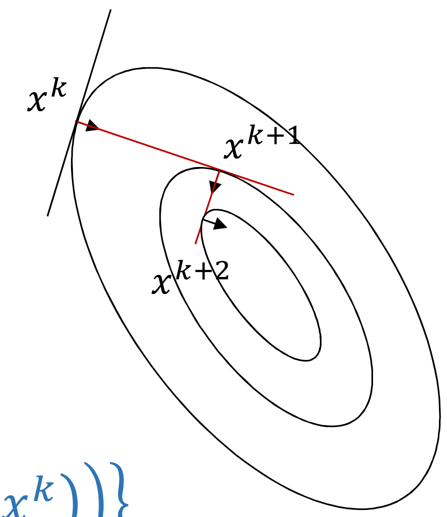
Méthode de gradient

1. **Algorithme**
2. **Exemple**
3. **Convergence**
4. **Déplacement**

Méthode de gradient: Algorithme $d^k = -\nabla f(x^k)$

- Entrée :
  - x^0 : Réel (point initial)
  - δ : Réel (la tolérance)
  - f : Fonction ($f: \mathbb{R}^n \to \mathbb{R}^1$ et $f \in C^1$)
- Sortie :
  - x^* : Réel (point de sortie)
- Début : $k = 0$
- WHILE($\|\nabla f(x^k)\| > \delta$ or $\|x^k - x^{k-1}\| > \delta$) {
  - Déterminer α_k tel que $\alpha_k = \underset{\alpha \geq 0}{argmin}\left\{f\left(x^k - \alpha \nabla f(x^k)\right)\right\}$
  - $x^{k+1} = x^k - \alpha_k \nabla f(x^k)$
  - $k = k + 1$

  }
- $x^* = x^k$
- RETURN x^*
- Fin



Méthode de gradient: Exemple

$$\mathrm{f} = \frac{1}{2}(x_1^2 + x_2^2) \rightarrow \nabla f(x_1, x_2) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \nabla^2 f(x_1, x_2) = I$$

Itération 1 :

Etape 0 : $k = 0,\ x^0 = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix},\ \delta = 10^{-2}$

Etape 1 : $\nabla f(x^0) = \nabla f(x_1^0, x_2^0) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \rightarrow \|\nabla f(x^0)\| = \sqrt{4+1} = \sqrt{5} > 10^{-2}$

Etape 2 : α_0 solution optimal de $\min\limits_{\alpha \geq 0} f(x^0 - \alpha \nabla f(x^0))$

- $\varphi(\alpha) = f\left(x^0 - \alpha \nabla f(x^0)\right) = \frac{1}{2}\left((2-2\alpha)^2 + (1-\alpha)^2\right) \Rightarrow \varphi'(\alpha) = -3 + 3\alpha = 0 \Rightarrow \alpha_0 = 1$
- $x^1 = x^0 - \alpha_0 \nabla f(x^0) = \begin{pmatrix} x_1^0 - 2\alpha_0 \\ x_2^0 - \alpha_0 \end{pmatrix} = \begin{pmatrix} 2 - 2 \times 1 \\ 1 - 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Itération 2 :

Etape 1: $\nabla f(x^1) = \nabla f(x_1^1, x_2^1) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow \|\nabla f(x^1)\| = 0 < 10^{-2} \rightarrow$ arrête

$$x^* = \begin{pmatrix} x_1^* \\ x_2^* \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Méthode de gradient: Convergence

Lemme : Soit $f \in C^1$. Tout point limite x^* d'une sous-suite convergente de la suite générée par la méthode du gradient est tel que $\nabla f(x^*) = 0$.

Théorème :

Si $f \in \mathrm{C}^1$ et tell que $\lim_{\|x\| \to +\infty} f(x) = +\infty$ alors, pour tout point de départ x_0 , la méthode de la plus fort pente converge vers un point stationnaire de f.

Théorème :

Soit une fonction $f: \mathbb{R}^n \to \mathbb{R}$ convexe et différentiable, tel que :

$$\forall x, y \in \mathbb{R}^n, \qquad \text{on a } \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

(i.e. ∇f est L-Lipchitzienne avec constante $L > 0$)

Alors l'algorithme de gradient descendant avec **un pas fixe** $\alpha \leq 1/L$ satisfait à la relation suivante : $\|f(x^k) - f(x^*)\| \leq \frac{\|x^0 - x^*\|^2}{2\alpha_k}$

- **L'ordre de convergence** de la méthode du gradient est linéaire.
- Ainsi pour arriver à $\|f(x^k) - f(x^*)\| \leq \varepsilon$, on a besoin de $O(1/\varepsilon)$ itérations

Méthode de gradient: Convergence

Preuve de lemme:

Considérons la sous-suite $\{x^{k_j}\}$ de $\{x^k\}$ telle que $\{x^{k_j}\} \to x^*$

Par conséquent, se référant aux suites

- $\{x^k\} = \{\dots, x^{k_j}, x^{k_j+1}, \dots, x^{k_j+\tau}, \dots\}$
- $\{x^{k_j}\} = \{\dots, x^{k_j}, \quad , x^{k_{j+1}}, \dots\}$ où $x^{k_{j+1}} = x^{k_j+\tau}$.

Ainsi, se référant à la suite $\{x^k\}$

- $f\left(x^{k_{j+1}}\right) = f\left(x^{k_j+\tau}\right) \leq \cdots \leq f\left(x^{k_j+1}\right)$
- $f\left(x^{k_j+1}\right) = \min_{\alpha \geq 0} \left\{ f\left(x^{k_j} - \alpha \nabla f\left(x^{k_j}\right)\right)\right\}$

Donc

- $f\left(x^{k_{j+1}}\right) \leq f\left(x^{k_j} - \alpha \nabla f\left(x^{k_j}\right)\right) \quad \forall \alpha \geq 0$
- $f\left(x^{k_{j+1}}\right) \leq f\left(x^{k_j+1}\right) \quad$ (puisque $k_{j+1} \geq k_j + 1$)
- $f\left(x^{k_j+1}\right) = \min_{\alpha \geq 0} \left\{ f\left(x^{k_j} - \alpha \nabla f\left(x^{k_j}\right)\right)\right\}$,

Ainsi $f\left(x^{k_{j+1}}\right) \leq f\left(x^{k_j} - \alpha \nabla f\left(x^{k_j}\right)\right) \quad \forall \alpha \geq 0$

Méthode de gradient: Convergence

Puisque $f \in C^1$, f est continue sur $\mathbb{R}^n$ et $\lim_{j\to\infty} f\left(x^{k_{j+1}}\right) = f(x^*)$

$$\lim_{j\to\infty}\left[f\left(x^{k_j} - \alpha\nabla f\left(x^{k_j}\right)\right)\right] = f\left(x^* - \alpha\nabla f(x^*)\right) \quad \forall\alpha \geq 0$$

Donc

$$f(x^*) \leq f\left(x^* - \alpha\nabla f(x^*)\right) \quad \forall\alpha \geq 0 \quad (3.2)$$

Par conséquent $\nabla f(x^*) = 0$ car autrement si $\nabla f(x^*) \neq 0$, alors $-\nabla f(x^*)$ **est une direction de descente** et se référant au lemme 3.2 nous obtenons une contradiction à la relation (3.2)

Remarque: Au début de son application, la méthode permet de faire diminuer la valeur de la fonction économique relativement rapidement, mais son efficacité à ce chapitre diminue au cours des itérations.

Méthode de gradient: Déplacement en zigzag

Le lemme suivant montre que la méthode de descente se déplace en zigzag puisque les directions utilisées à deux itérations successives sont à angle droit.

Lemme:

Si $x^{k+1} = x^k - \alpha_k \nabla f\left(x^k\right)$ où α_k est tel que $f\left(x^k - \alpha_k \nabla f\left(x^k\right)\right) = \min_{\alpha \geq 0}\{f(x^k - \alpha \nabla f\left(x^k\right)\}$

Alors $\nabla f\left(x^{k+1}\right)^T \nabla f\left(x^k\right) = 0$

Preuve

Si $\nabla f\left(x^k\right) = 0$ alors la preuve est complétée

Si $\nabla f\left(x^k\right) \neq 0$, considérons la fonction $\varphi : \mathbb{R}^+ \to \mathbb{R} \quad \varphi(\alpha) = f\left(x^k - \alpha \nabla f\left(x^k\right)\right)$

Par dérivation en chaine,

$$\varphi'(\alpha) = -\nabla f\left(x^k\right)^T \nabla f\left(x^k - \alpha \nabla f\left(x^k\right)\right)$$

Puisque α_k est un minimum local de $\varphi(\alpha)$, $\varphi'(\alpha_k) = 0$

Et alors

$$0 = \varphi'(\alpha_k) = -\nabla f\left(x^k\right)^T \nabla f\left(x^k - \alpha_k \nabla f\left(x^k\right)\right) = -\nabla f\left(x^k\right)^T \nabla f\left(x^{k+1}\right)$$

Méthode de gradient: Déplacement en zigzag

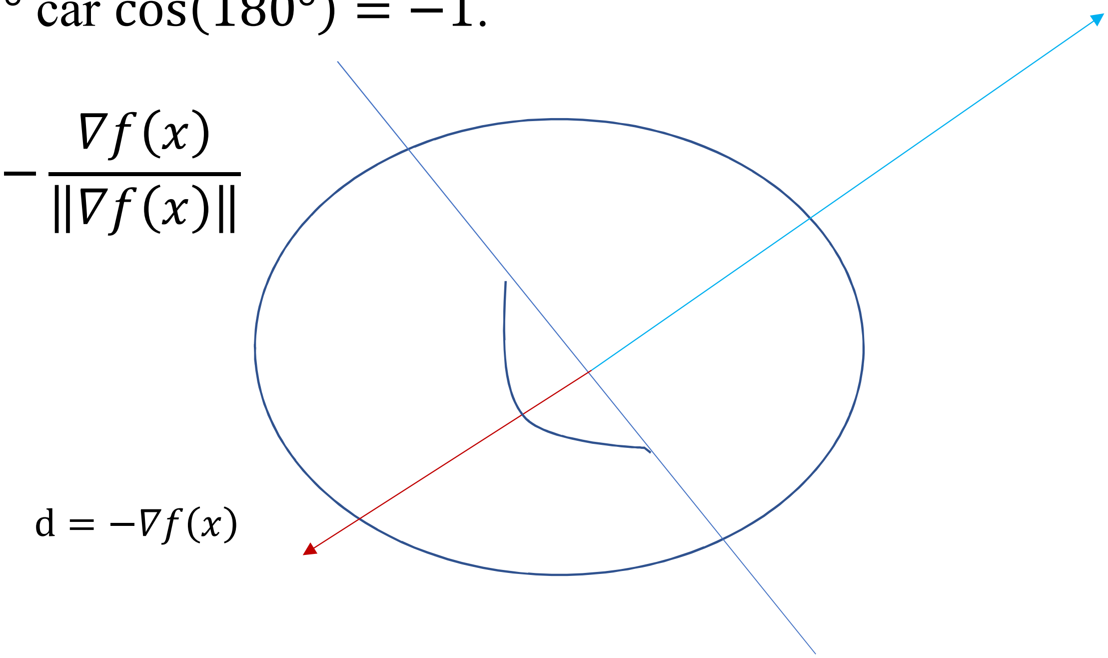
Remarque:
La méthode du gradient est la méthode **la plus rapide marginalement** (« steepest descent method »)

$$\nabla f(x)^T d = \|\nabla f(x)\| \|d\| \cos(\theta)$$

où θ est l'angle entre d et $\nabla f(x)$.ainsi, parmi les directions d de norme 1 ($\|d\| = 1$) ; celle minimisant $\nabla f(x)^T d$ a un angle de 180° car $\cos(180°) = -1$.
Ainsi, cette direction d correspond à

$$d = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$$

- Si $\nabla f(x)^T d < 0$ alors $\cos(\theta) < 0$
- $\Longrightarrow \theta \in \left]\frac{\pi}{2} + 2k\pi, \frac{3\pi}{2} + 2k\pi\right[$
- $(\|d\| = 1)$



Méthode de gradient: Cas quadratique $Q = Q^T,\ d^k \neq 0$

Soit Q est supposée définie positive. $f : \mathbb{R}^n \to \mathbb{R} \in C^2$

$$f(x) = \frac{1}{2}x^T Q x - b^T x \Rightarrow \nabla f(x) = Qx - b \Longrightarrow \nabla^2 f(x) = Q$$

La suite générée par la méthode du gradient est donnée par

$$\boldsymbol{x^{k+1} = x^k - \alpha_k \nabla f\left(x^k\right) = x^k + \alpha_k d^k}$$

Dons ce cas précis, ou peut déterminer explicitement α_k. En effet :

$$\begin{aligned}
\nabla f\left(x^{k+1}\right)^T \nabla f\left(x^k\right) &= -\left(Qx^{k+1} - b\right)^T d^k \\
&= -\left(Q\left(x^k + \alpha_k d^k\right) - b\right)^T d^k \\
&= -\left(Qx^k - b + \alpha_k Q d^k\right)^T d^k \\
&= -\left(\nabla f\left(x^k\right) + \alpha_k Q d^k\right)^T d^k \\
&= -\left(-d_k + \alpha_k Q d^k\right)^T d^k \\
&= -\left(-d^k\right)^T d_k - \left(\alpha_k Q d^k\right)^T d^k \\
&= (d^k)^T d_k - \alpha_k (d^k)^T Q d^k = 0 \Rightarrow \alpha_k = \frac{(d^k)^T d^k}{(d^k)^T Q d^k} > 0
\end{aligned}$$

Méthode de gradient: Cas quadratique $Q = Q^T,\ d^k \neq 0$

Vitesse de convergence

Pour faciliter l'étude de ce vitesse de convergence on va définir la norme associée à la matrice Q par $\|x\|_Q = \sqrt{x^T Q x}$

Et on introduit la fonction utilitaire $E(x) = \frac{1}{2}(x - x^*)^T Q(x - x^*) = \frac{1}{2}\|x - x^*\|_Q^2$

Où x^* est l'unique minimum de f c.à.d. $Qx^* = b$. Alors $E(x) = f(x) + \frac{1}{2}{x^*}^T Q x^*$

On remarque que E et f ne différent que par une constante. Ainsi $\min_{x \in \mathbb{R}^n} f(x) \Leftrightarrow \min_{x \in \mathbb{R}^n} E(x)$

Q étant définie positive alors toutes ses valeurs propres sont strictement positive.

Si l'on désigne par $\lambda_n > \lambda_1 > 0$ la plus petite et la plus grande valeur propre de Q alors on a théorème suivant : $\beta = \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 < 1$

Théorème :

Pour tout point de départ x^0 ; la méthode du gradient converge vers l'unique minimum x^* de f et à chaque itération k, on a $E\left(x^{k+1}\right) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 E\left(x^k\right)$

Méthode de gradient: Cas quadratique $Q = Q^T,\ d^k \neq 0$

<u>Remarques</u>

- la plus petite et la plus grande valeur propre de Q apparaissent dans le résultat car elles sont une indication du degré ‘’d’aplatissement ‘’ des courbes de niveau de f
- **si $\boldsymbol{\lambda_1 = \lambda_n}$ alors ces courbe sont sphériques et la méthode du gradient converge en une seule itération quel que soit le point de départ**
- le théorème dit que la méthode du gradient converge linéairement avec un rapport de convergence $\beta < 1$
- En posant $r = \frac{\lambda_n}{\lambda_1}$ on aura :

$$\left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 = \left(\frac{r-1}{r+1}\right)^2$$

  - r est appelé **la condition menber** de Q
  - Ainsi**, plus r est grand et plus la convergence est lente**, en faisant un changement de variable, il est possible d’améliorer cette vitesse.

Gradient Conjugué

1. **Méthode fondamentale des directions conjuguées**
2. **Principe**
3. **Algorithme**
4. **Exemple**
5. **convergence**

Gradient Conjugué: Méthode fondamentale des directions conjuguées

Type de méthodes présenté dans le cadre de la programmation quadratique :

$$f(x) = \frac{1}{2}x^T\boldsymbol{Q}x - b^Tx \Longrightarrow \nabla f(x) = Qx - b, \nabla^2 f(x) = Q$$

Où Q est une matrice symétrique $\boldsymbol{n} \times \boldsymbol{n}$ **définie positive**.

Définition :

- Deux vecteurs d^1 et d^2 sont $\boldsymbol{Q}$-orthogonaux ou conjugués par rapport à Q si ${d^1}^T Qd^2 = 0$
- Un ensemble fini de vecteurs distincts non nuls $\{d^0, d^1, \ldots, d^k\}$ est dit conjugué par rapport à une matrice symétrique réelle Q, si

$$\left(d^i\right)^T Qd^j = 0 \quad \forall\, i \neq j$$

Théorème 1 : Indépendance linéaire des vecteurs conjugués

Si les vecteurs non nuls $d^0, d^1, \ldots, d^{\mathrm{n}-1}$ forment un ensemble conjugué par rapport à une matrice définie positive Q, alors ils sont linéairement indépendants(forme une base).

Gradient Conjugué: Méthode fondamentale des directions conjuguées

$$x \in \mathbb{R}^2, \qquad {d^1}^T Q d^2 = 0$$

- Si $Q = I$
  - $Q = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
  - $d^1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, d^2 = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$
  - ${d^1}^T Q d^2 = (2 \quad 1) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \end{pmatrix} = (2 \quad 1) \begin{pmatrix} 1 \\ -2 \end{pmatrix} = 2 - 2 = 0$
  - d^1 et d^2 sont orthogonuax
- Si $Q = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$
  - $d^1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, d^2 = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \Longrightarrow {d^1}^T Q d^2 = (2 \quad 1) \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \end{pmatrix} = (2 \quad 1) \begin{pmatrix} -3 \\ 0 \end{pmatrix} = -6 \neq 0$
  - $d^1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, d^2 = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \Longrightarrow (0 \quad 1) \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \end{pmatrix} = (0 \quad 1) \begin{pmatrix} -3 \\ 0 \end{pmatrix} = 0$
  - ${d^1}^T d^2 = (0 \quad 1) \begin{pmatrix} 1 \\ -2 \end{pmatrix} = -2 \neq 0$

Gradient Conjugué: Principe

Soit $\{d^0, d^1, \ldots, d^{\mathrm{n}-1}\}$ ensemble conjugué

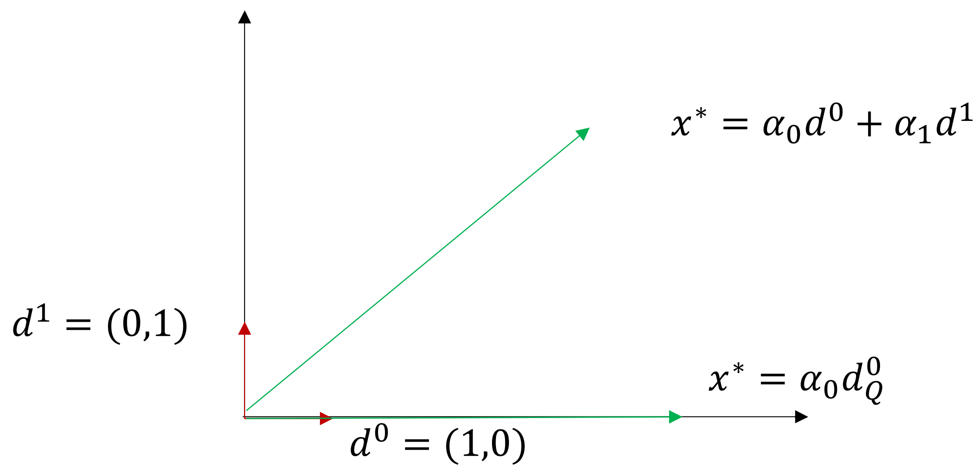
Le calcul de x^* peut être considéré comme un calcul itératif dans lequel n ajustements successifs de $\alpha_k d^k$sont effectués à un point initial x^0. Par ailleurs, la séquence générée par la relation récursive

- $\boldsymbol{x^{k+1} = x^k + \alpha_k d^k}$ **avec** $\boldsymbol{\left(d^i\right)^T Q d^j = 0 \quad \forall\, i \neq j}$
- $\boldsymbol{\alpha_k = \dfrac{\left(d^k\right)^T d_k}{\left(d^k\right)^T Q d_k}}$
- **Et** x^k**converge quand** $\boldsymbol{k = n - 1}$ **à** $\boldsymbol{x^n = x^* = \sum_{k=0}^{n-1} \alpha_k\, d^k}$

- Gradient Conjugué
  - $x^0, \ \ d^0 = -\nabla f(x^0)$
  - $\boldsymbol{x^{k+1} = x^k + \alpha_k d^k}$
  - $\alpha_k = \dfrac{\left(d^k\right)^T d_k}{\left(d^k\right)^T Q d_k}$
  - Et x^kconverge quand $k = n - 1$ à $x_n = x^* = \sum_{k=0}^{n-1} \alpha_k\, d^k$

Gradient Conjugué:

$\boldsymbol{Q = I}$



Gradient Conjugué: Algorithme ${d^k}^T Q d^k > 0$

- **Entrée :**
  - x^0 : Réel (point initial)
  - $f(x) = \frac{1}{2} x^T Q x - b^T x$: Fonction ($f: \mathbb{R}^n \to \mathbb{R}^1$ et $f \in C^2$, Q définie positive , ${d^k}^T Q d^k > 0$
- **Sortie :**x^*: Réel (point de sortie)
- **Début :**
  - $d^0 = -\nabla f(x^0), \quad k = 0$
- **FOR ($k = 0 : n - 1$)**
  - $\alpha_k = \frac{(d^k)^T d^k}{{d^k}^T Q d^k} \Longrightarrow x^{k+1} = x^k + \alpha_k d^k$
  - $\beta_k = \frac{\nabla f(x^{k+1})^T Q d^k}{{d^k}^T Q d^k} \Longrightarrow d^{k+1} = -\nabla f\left(x^{k+1}\right) + \beta_k d^k$
  - $k = k + 1$

ENDFOR

- RETURN $x^* = x^n$
- Fin

Gradient Conjugué: Exemple1: $b = 0, \mathrm{n} = 2$

Soit le problème : $\min_{x \in \mathbb{R}^n} f(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2) = \frac{1}{2} x^T Q x$

On a : $\nabla f(x_1, x_2) = Qx = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ et $\nabla^2 f(x_1, x_2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = Q$

Considérer la solution initiale $x^0 = (x_1^0, x_2^0)^T = (2,1)^T = \binom{2}{1}$

Itération 1:

- $x^0 = \binom{2}{1}$, $d^0 = -\nabla f(x^0) = \begin{pmatrix} -2 \\ -1 \end{pmatrix}$

Itération 2:

- $\alpha_0 = \frac{(d^0)^T d^0}{d^{0^T} Q d^0} = \frac{(-2,-1)\binom{-2}{-1}}{(-2,-1)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\binom{-2}{-1}} = \frac{5}{(-2,-1)\binom{-2}{-1}} = \frac{5}{5} = 1$
- $x^1 = x^0 + \alpha_0 d^0 = \binom{2}{1} + 1 \times \begin{pmatrix} -2 \\ -1 \end{pmatrix} = \binom{0}{0} \Longrightarrow \nabla f(x^1) = \nabla f(x_1^1, x_2^1) = \binom{0}{0}$
- $\beta_0 = \frac{\nabla f(x^1)^T Q d^0}{d^{0^T} Q d^0} == \frac{(0,0)\binom{-2}{-1}}{(-2,-1)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\binom{-2}{-1}} = \frac{0}{5} = 0$
- $d^1 = -\nabla f(x^1) + \beta_0 d^0 = -\binom{0}{0} + 0 \times \begin{pmatrix} -2 \\ -1 \end{pmatrix} = \binom{0}{0}$

Gradient Conjugué: Exemple2

Soit la fonction à minimiser :

$$f(x) = 4x_1^2 + 4x_2^2 - 4x_1x_2 - 12x_2 = \frac{1}{2}(x_1, x_2)\begin{pmatrix} 8 & -4 \\ -4 & 8 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - (0,12)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

On a :

- $\nabla f(x) = [8x_1 - 4x_2, -4x_1 + 8x_2 - 12]^T = \begin{pmatrix} 8 & -4 \\ -4 & 8 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} 0 \\ 12 \end{pmatrix}$
- $\nabla^2 f(x) = \begin{pmatrix} 8 & -4 \\ -4 & 8 \end{pmatrix} = Q \; et \; b = \begin{pmatrix} 0 \\ 12 \end{pmatrix}$

Itération 1:

- $x^0 = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix} \Longrightarrow d^0 = -\nabla f(x^0) = -\left(\begin{pmatrix} 8 & -4 \\ -4 & 8 \end{pmatrix}\begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 12 \end{pmatrix}\right) = -\left(\begin{pmatrix} -8 \\ 10 \end{pmatrix} - \begin{pmatrix} 0 \\ 12 \end{pmatrix}\right) = \begin{pmatrix} 8 \\ 2 \end{pmatrix}$

Itération 2:

- $\alpha_0 = \frac{(d^0)^T d^0}{d^{0^T} Q d^0} = \frac{(8,2)\begin{pmatrix} 8 \\ 2 \end{pmatrix}}{(8,2)\begin{pmatrix} 8 & -4 \\ -4 & 8 \end{pmatrix}\begin{pmatrix} 8 \\ 2 \end{pmatrix}} = \frac{68}{(8,2)\begin{pmatrix} 56 \\ -16 \end{pmatrix}} = \frac{68}{416} = \frac{17}{104}$
- $x^1 = x^0 + \alpha_0 d^0 = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix} + \frac{17}{104}\begin{pmatrix} 8 \\ 2 \end{pmatrix} = \begin{pmatrix} \frac{-11}{26} \\ \frac{69}{52} \end{pmatrix}$
- $\beta_0 = \frac{\nabla f(x^1)^T Q d^0}{d^{0^T} Q d^0} \Longrightarrow d^1 = -\nabla f(x^1) + \beta_0 d^0$

Gradient Conjugué: Convergence

Théorème : (Théorème des directions conjuguées)
Supposons que la matrice Q est symétrique définie positive.
Soit $\{d^k\}_{k=0}^{n-1}$ **un ensemble de $\boldsymbol{n}$ vecteurs non-nuls qui sont $\boldsymbol{Q}$-orthogonaux deux à deux.**
Etant donné un point $x^0 \in \mathbb{R}^n$ quelconque, la suite $\{x^k\}$ générée comme suit :

$$x^{k+1} = x^k + \alpha_k d^k, \qquad k = 0, \dots, n-1$$

Où

$$\alpha_k = -\frac{\nabla f\left(x^k\right)^T d^k}{{d^k}^T \nabla^2 f(x^k) d^k} = -\frac{\left(Qx^k - b\right)^T d^k}{{d^k}^T Q d^k}$$

Converge vers l'unique minimum $\boldsymbol{x}^*$ de $\boldsymbol{f(x) = \frac{1}{2} x^T Q x - b^T x}$ après $\boldsymbol{n}$ itérations $(\mathbf{i.e.}, \boldsymbol{x^* = x^n})$

Algorithme de Newton

1. **Principe**
2. **Algorithme**
3. **Exemple**
4. **convergence**

Algorithme de Newton: Principe

La méthode de Newton est une processus itératif pour construire une suite de points $\{x^k\}$ en partant d'un point x^0 dans le voisinage d'une solution optimale x^*.
Pour obtenir x^{k+1}, considérons l'approximation quadratique de f.

$$Q_k(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k)$$

x^{k+1} est le points où le gradient de l'approximation $Q_k(x)$ s'annule :

$$\nabla Q_k(x^{k+1}) = \nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = 0 \Longrightarrow x^{k+1} = x^k - [\nabla^2 f(x_k)]^{-1} \nabla f(x^k)$$

Il faut noter que le Hessien $\nabla^2 f(x^k)$ doit être une matrice non-singulière.

La méthode newton est itérative où la direction d^k au point x^k est $d^k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$

- d^k est bien **une direction de descente** si le Hessien $\nabla^2 f(x^k)$ est une matrice positive définie

et si $\nabla f(x^k) \neq 0$: $\nabla f(x^k)^T d^k = -\nabla f(x^k)^T [\nabla^2 f(x^k)]^{-1} \nabla f(x^k) < 0$

- Pas optimal α_k défini comme suivant:

$$f(x^{k+1}) = f\left(x^k - \alpha_k [\nabla^2 f(x^k)]^{-1} \nabla f(x^k)\right) = \min_{\alpha \geq 0} \left\{ f\left(x^k - \alpha [\nabla^2 f(x^k)]^{-1} \nabla f(x^k)\right) \right\}$$

Algorithme de Newton: Principe

- Pour contrer la difficulté de choisir x^0 dans le voisinage de x^* où $\nabla f(x^*) = 0$
- Si le Hessien $\nabla^2 f\left(x^k\right)$ n'est pas une matrice positive définie, considérons la direction d^k suivante au point x^k :

$$d^k = -\left[\varepsilon_k I + \nabla^2 f\left(x^k\right)\right]^{-1} \nabla f\left(x^k\right)$$

- Où I est la matrice identité et où $[\varepsilon_k I + \nabla^2 f(x_k)]$ a toute ses valeurs propres positives (i.e., est définie positive).
- $\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \Longrightarrow \nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial^2 x_n} \end{pmatrix}$
- Si $f \in C^2$ $alors$ $\forall i \neq j$ $\frac{\partial^2 f}{\partial x_j \partial x_i} = \frac{\partial^2 f}{\partial x_i \partial x_j}$

Algorithme de Newton:(Algorithme)

Entrée :

- x^0**:** Réel (point initial)
- δ : Réel (la tolérance)
- f: Fonction ($f: \mathbb{R}^n \to \mathbb{R}^1$ et $f \in C^1$)

Sortie :

- x^*: Réel (point de sortie)

Début :

- $k = 0$
- $d^0 = -[\nabla^2 f(x^0)]^{-1} \nabla f(x^0)$

- **WHILE** ($\|d^k\| > \delta$) {
  - Déterminer α_k tel que $\alpha_k = \underset{\alpha \geq 0}{argmin} \{f(x^k - \alpha d^{k+1})\}$ le pas optimal
  - $x^{k+1} = x^k - \alpha_k d^k$
    - IF ($\nabla^2 f(x^{k+1})$ est définie positive) $d^{k+1} = -[\nabla^2 f(x^{k+1})]^{-1} \nabla f(x^{k+1})$

      ELSE $d^k = -[\varepsilon_k I + \nabla^2 f(x^k)]^{-1} \nabla f(x^k)$
  - $k = k + 1$ }
- **RETURN** $x^* = x^k$

Fin

Algorithme de Newton: Exemple

Entrée :

- $\min_{x\in\mathbb{R}^n} f(x_1,x_2) = \frac{1}{2}(x_1^2 + x_2^2), \delta = 10^{-2}$
- $\nabla f(x_1,x_2) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $\nabla^2 f(x_1,x_2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- $x^0 = (x_1^0, x_2^0)^T = \binom{2}{1} \Longrightarrow \nabla f(x^0) = \binom{2}{1}$

Itération 1:

- $d^0 = -[\nabla^2 f(x^0)]^{-1} \nabla f(x^0) = -\nabla f(x^0) = -\binom{2}{1}$
- $\alpha_0 = \underset{\alpha>0}{\operatorname{argmin}}\, \varphi(\alpha)$
- $\varphi(\alpha) = f(x^0 + \alpha d^0) = \frac{1}{2}((2-2\alpha)^2 + (1-\alpha)^2) \Longrightarrow \varphi'(\alpha) - 5 + 5\alpha = 0 \Longrightarrow \alpha_0 = 1$
- $x^1 = x^0 + \alpha_0 d^0 = \binom{2}{1} - \alpha_0 \binom{2}{1} = \binom{2-2\alpha_0}{1-\alpha_0} \Longrightarrow x^1 = \binom{2-2\alpha_0}{1-\alpha_0} = \binom{0}{0}$

Itération 2:

- $d^1 = -[\nabla^2 f(x^1)]^{-1} \nabla f(x^1) = -\nabla f(x^1) = \binom{0}{0} \rightarrow \|d^1\| = 0 < \delta$ Stop

Sortie : $x^* = \binom{0}{0}$

Algorithme de Newton: Convergence

Théorème de Convergence d'ordre 2 :

Soit $f \in C^3$, sur $\mathbb{R}^n$, et supposons que dans le minimum local x^* , le Hessien $H(x^*)$ Est défini positive, alors si x^0 est choisi suffisamment près de x^*. Les points généré par la méthode de newton converge vers x^*. Avec **l'ordre de convergence au moins 2.**

Théorème de Convergence de méthode de newton modifié (cas quadratique)

Soit x^* un minimum unique de la fonction, définissons $E(x) = \frac{1}{2}(x - x^*)Q(x - x^*)$ alors pour algorithmes de la forme $x^{k+1} = x^k - \alpha_k S^k \nabla f\left(x^k\right)^T$

- Avec $S_k = [\nabla^2 f(x)]^{-1}$ dans le cas de Newton pure
- on a pour chaque itération k :

$$E\left(x^{k+1}\right) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 E\left(x^k\right)$$

- Où λ_1 et λ_n sont respectivement le petit et la plus grande valeur propre de la matrice $S_k Q$.

Méthode de Quasi-Newton

1. **Principe**
2. **Algorithme général**
3. **Variances**
4. **Davidon–Fletcher–Powell (DFP)**

Méthode de Quasi-Newton : Principe

L'idée fondamentale derrière la plupart des méthodes de quasi-Newton est d'essayer de construire un approximation H^k **l'inverse de Hessien** $\boldsymbol{\nabla^2 f\left(x^k\right)^{-1}}$, en utilisant l'information recueillie au cours du processus de descente.

- $\left(\boldsymbol{\nabla^2 f\left(x^k\right) + \varepsilon}\right)^{\boldsymbol{-1}} \approx H^k \Leftrightarrow \boldsymbol{B^k} \approx \boldsymbol{\nabla^2 f}\left(x^k\right) + \boldsymbol{\varepsilon} \Leftrightarrow \left(\boldsymbol{B^k}\right)^{\boldsymbol{-1}} = H^k$

Pour obtenir x^{k+1}, considérons l'approximation quadratique de f.

- $Q_k(x) = f\left(x^k\right) + \nabla f\left(x^k\right)^T\left(x - x^k\right) + \frac{1}{2}\left(x - x^k\right)^T \boldsymbol{B^k}\left(x - x^k\right)$
- x^{k+1} est le points où le gradient de l'approximation $Q_k(x)$ s'annule :
- $\nabla Q_k\left(x^{k+1}\right) = \nabla f\left(x^k\right) + \boldsymbol{B^k}\left(x^{k+1} - x^k\right) = 0 \Longrightarrow \boldsymbol{x^{k+1} = x^k} - H^k \boldsymbol{\nabla f\left(x^k\right)}$

Soit f une fonction dans $\mathbb{R}^n$ de C^2 si pour deux points x^{k+1},x^k on défini :

- $g_{k+1} = \nabla f\left(x^{k+1}\right)$ et $g_k = \nabla f\left(x^k\right)$
- $s^k = x^{k+1} - x^k = \alpha_k d^k$
- $y^k = g_{k+1} - g_k$

alors :

$$\boldsymbol{g_{k+1} - g_k \approx B^{k+1} s^k \Leftrightarrow s^k \approx H^{k+1}\left(g_{k+1} - g_k\right)}$$

Méthode de Quasi-Newton :Algorithme Général

Entrée :

- x^0**:** Réel (point initial)
- δ : Réel (la tolérance)
- f : Fonction ($f : \mathbb{R}^n \to \mathbb{R}^1$ et $f \in C^1$)

Sortie : x^* : Réel (point de sortie)

Début : $k = 0,\ \ g^0 = \nabla f(x^0)^T,\ \ H^0 = I$

WHILE ($\|g^k\| > \delta$) {

$$d^k = -H^k g^k$$

Déterminer $\alpha_k = \underset{\alpha \geq 0}{\operatorname{argmin}}\{f(x^k - \alpha d^k)\}$

$$x^{k+1} = x^k - \alpha_k d^k,\ \ g^{k+1} = \nabla f(x^{k+1})^T$$

Déterminer H^{k+1}

$k = k + 1$}

ENDWHILE

$x^* = x^k$, RETURN x^*

Fin

Méthode de Quasi-Newton :Variances

Limite de Méthode Newton:

- Il exige que le Hessien soit définit positive, ce qui est n'est pas toujours vrai pour chaque itération.
- Eviter de calculer le Hessien $\nabla^2 f\left(x^k\right)^{-1}$ et son inverse à chaque itération.

Solution:

Nous utilisons plutôt une matrice H^k définie positive qui est modifiée d'une itération à l'autre, il existe certains algorithmes pour calculer H^k:

- **Symmetric rank-one:** SR1
- **Davidon–Fletcher–Powell:** DFP , La matrice est H^{k+1} est déterminer comme suivant
  - $H^{k+1} = H^k + A^k + B^k$: $A^k = \frac{\alpha_k d^k {d^k}^T}{{d^k}^T y^k}, B^k = \frac{-H^k y^k \left(H^k y^k\right)^T}{{y^k}^T H^k y^k},\ y^k = \nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)$
- **Broyden–Fletcher–Goldfarb–Shanno** BFGS

Méthode de Quasi-Newton : Davidon–Fletcher–Powell

Exemple

$$\min_{x\in\mathbb{R}^2} f(x_1,x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

- $\nabla f(x) = \begin{pmatrix} 1+4x_1+2x_2 \\ -1+2x_1+2x_2 \end{pmatrix}$

Entrée:

- $x^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, H^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \delta = 0.01,\ \nabla f(x^0) = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$
- $k = 0$

While $\nabla f(x^0) = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \Longrightarrow \|\nabla f(x^0)\| = \sqrt{2} > \delta$

Itération 1 :

- Calcule de d^0 : $d^0 = -H^0\nabla f(x^0) = -\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$
- Calcule de pas optimal α_0
  - $\varphi(\alpha) = f(x^0 + \alpha d^0) = f\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix} + \alpha\begin{pmatrix} -1 \\ 1 \end{pmatrix}\right) \Longrightarrow \varphi(\alpha) = f(-\alpha,\alpha) = \alpha^2 - 2\alpha \Longrightarrow \varphi'(\alpha_0) = 2\alpha - 2 = 0 \Longrightarrow \alpha_0 = 1,$
- Calcule de x^1
  - $x^1 = x^0 + \alpha_0 d^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 1\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$
- Calcule de $\nabla f(x^1)$

Méthode de Quasi-Newton : Davidon–Fletcher–Powell

- Calcule de H^1 :
  - $y^0 = \nabla f(x^1) - \nabla f(x^0) = \begin{pmatrix}-1\\-1\end{pmatrix} - \begin{pmatrix}1\\-1\end{pmatrix} = \begin{pmatrix}-2\\0\end{pmatrix}$
  - $d^{0^T} y^0 = (-1,1)\begin{pmatrix}-2\\0\end{pmatrix} = 2,\ d^0 d^{0^T} = \begin{pmatrix}-1\\1\end{pmatrix}(-1,1) = \begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}$
  - $H^0 y^0 = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}-2\\0\end{pmatrix} = \begin{pmatrix}-2\\0\end{pmatrix} \Longrightarrow (H^0 y^0)^T = (-2,0)$
  - $y^{0^T} H^0 y^0 = (-2,0)\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}-2\\0\end{pmatrix} = (-2\ 0)\begin{pmatrix}-2\\0\end{pmatrix} = 4$
  - $A^0 = \alpha_0 \dfrac{d^0 d^{0^T}}{d^{0^T} y^0} = 1\left(\frac{1}{2}\right)\begin{pmatrix}1 & -1\\-1 & 1\end{pmatrix} = \begin{pmatrix}\frac{1}{2} & -\frac{1}{2}\\-\frac{1}{2} & \frac{1}{2}\end{pmatrix}$
  - $B^0 = -\dfrac{(H^0 y^0)(H^0 y^0)^T}{y^{0^T} H^0 y^0} = -\dfrac{\begin{pmatrix}-2\\0\end{pmatrix}(-2,0)}{4} = -\frac{1}{4}\begin{pmatrix}4 & 0\\0 & 0\end{pmatrix} = -\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}$
  - $H^1 = H^0 + A^0 + B^0 = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} + \begin{pmatrix}\frac{1}{2} & -\frac{1}{2}\\-\frac{1}{2} & \frac{1}{2}\end{pmatrix} + \begin{pmatrix}-1 & 0\\0 & 0\end{pmatrix} \Longrightarrow H^1 = \begin{pmatrix}0.5 & -0.5\\-0.5 & 1.5\end{pmatrix}$

Méthode de Quasi-Newton : Davidon–Fletcher–Powell

Itération 2 : $\|\nabla f_1\| = 1.4142 > \delta$

- Calcule de d^1
  - $d^1 = -H^1 \nabla f(x^1) = -\begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 1.5 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- Calcule de pas optimal α_1
  - $\varphi(\alpha) = f(x^1 + \alpha d^1) = f\left(\begin{pmatrix} -1 \\ 1 \end{pmatrix} + \alpha \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right) = f\left(\begin{pmatrix} -1 \\ 1+\alpha \end{pmatrix}\right)$
  - $\varphi(\alpha) = -1 - (1+\alpha) + 2(-1)^2 + 2(-1)(1+\alpha) + (1+\alpha)^2$
  - $\varphi(\alpha) = \alpha^2 - \alpha - 1 \Longrightarrow \varphi'(\alpha_1) = 2\alpha - 1 = 0 \Longrightarrow \alpha_1 = \frac{1}{2}$
- Calcule de x^2
  - $x^2 = x^1 + \alpha_1 d^1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix}$
- $\nabla f(x^2) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Longrightarrow \|\nabla f(x^2)\| = 0 < \delta \Longrightarrow x^* = x^2 = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix}$

Méthode de Quasi-Newton : Davidon–Fletcher–Powell

la variance DFP est une variance typique lequel est de rang 2, i.e., H^{k+1} est obtenu par addition de deux matrices symétrique à H^k, chacune de premier rang :

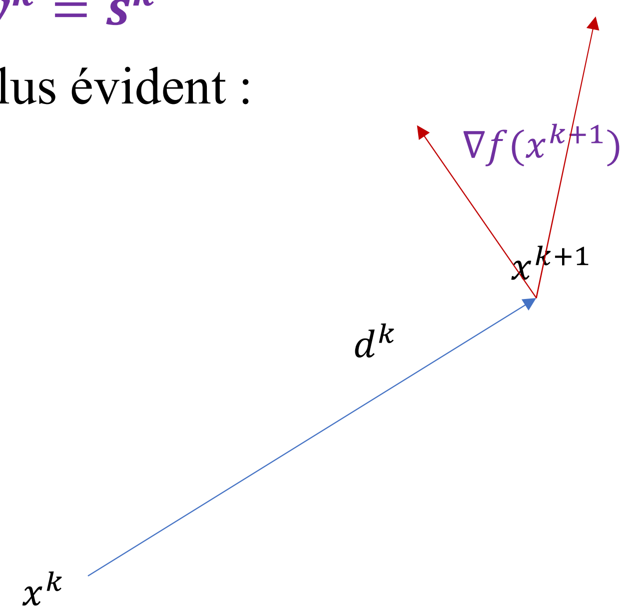
$$H^{k+1} = H^k + auu^T + bvv^T = H^k + A^k + B^k$$

Où , $u, v \in \mathbb{R}^n$, a et b sont des scalaire à déterminer, par l'équation de quasi-Newton :

$$\boldsymbol{H^k y^k + auu^T y^k + bvv^T y^k = s^k}$$

Clairement, u et v sont non uniques, mais la solution la plus évident :

- $u = s^k, v = H^k y^k$
  - $s^k = x^{k+1} - x^k = \alpha_k d^k \rightarrow s^k {s^k}^T = (\alpha_k)^2 d^k {d^k}^T$
  - $\Longrightarrow a = \frac{1}{u^T y^k} = \frac{1}{{s^k}^T y^k} = \frac{1}{(\alpha_k d^k)^T y^k},$
  - $\Longrightarrow b = -\frac{1}{v^T y^k} = -\frac{1}{{y^k}^T H^k y^k}$
  - $\Longrightarrow H^{k+1} = H^k + \frac{s^k {s^k}^T}{{s^k}^T y^k} - \frac{H^k y^k {y^k}^T H^k}{{y^k}^T H^k y^k}$



Méthode de Quasi-Newton : Davidon–Fletcher–Powell

Lemme 1 : $\left(\mathbf{i.e.}, \boldsymbol{\nabla f}\left(\boldsymbol{x^k}\right)^{\boldsymbol{T}} \boldsymbol{d^k} < \mathbf{0}\right)$

Si H^k est définie positive et $\nabla f\left(x^k\right) \neq 0$, alors d^k est une direction de descente à x^k.

Preuve:

Par définition $d^k = -H^k \nabla f\left(x^k\right)$ donc $\nabla f\left(x^k\right)^T d^k = -\nabla f\left(x^k\right)^T H^k \nabla f\left(x^k\right) < 0$

Sous les hypothèses que H^k est définie positive et $\nabla f\left(x^k\right) \neq 0$

Lemme 2 : (d^k **est perpendiculaire au** $\boldsymbol{\nabla f}(\boldsymbol{x^{k+1}})$ **)**

Si H^k est définie positive, alors ${d^k}^T \nabla f\left(x^{k+1}\right) = 0$

Preuve:

Si $d^k = 0$, la preuve est complétée

Si $d^k \neq 0$, pour $\alpha \geq 0$ définissons les fonctions: $\varphi(\alpha) = f\left(x^k + \alpha d^k\right)$ et $g(\alpha) = x^k + \alpha d^k$

Puisque $\varphi: \mathbb{R} \to \mathbb{R}$, par dérivation en chaîne on a $\varphi'(\alpha) = \left(fog(\alpha)\right)' = {d^k}^T \nabla f\left(x^k + \alpha d^k\right)$

Puisque α_k est un minimum local de $\varphi(\alpha)$, alors

- $\varphi'(\alpha_k) = 0 \Longrightarrow \varphi'(\alpha_k) = {d^k}^T \nabla f\left(x^k + \alpha_k d^k\right) = {d^k}^T \nabla f\left(x^{k+1}\right) \Longrightarrow {d^k}^T \nabla f\left(x^{k+1}\right) = 0$

Méthode de Quasi-Newton : Davidon–Fletcher–Powell

Lemme3 :

Soit H^k est définie positive alors la formule DFP produit H^{k+1} définit positive si et seulement si $\boldsymbol{(d^k)^T y^k > 0}$.

Preuve

La preuve par induction sur k pour $k = 0, H^0 = I$ est définie positive

Supposons que H^k est définie positive. Alors

- $x^k H^{k+1} x = x^T H^k x + x^T A^k x + x^T B^k x = x^T H^k x + \frac{\alpha_k^2 x^T d^k {d^k}^T x}{\alpha_k {d^k}^T y^k} - \frac{x^T H^k y^k (H^k y^k)^T x}{{y^k}^T H^k y^k}$
- $\Longrightarrow x^k H^{k+1} x = \frac{x^T H^k x {y^k}^T H^k y^k - x^T H^k y^k {y^k}^T {H^k}^T x}{{y^k}^T H^k y^k} + \frac{\left(\alpha_k {d^k}^T x\right)^2}{\alpha_k {d^k}^T y^k}$

Mais puisque H^k est définie positive, il découle de l'inégalité de Schwartz que $x^T H^{k+1} x \geq \frac{\left(\alpha_k {d^k}^T x\right)^2}{\alpha_k {d^k}^T y^k}$

Rappelons que $y^k = \nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)$ en utilisant les lemmes 2 et 1 On obtient:

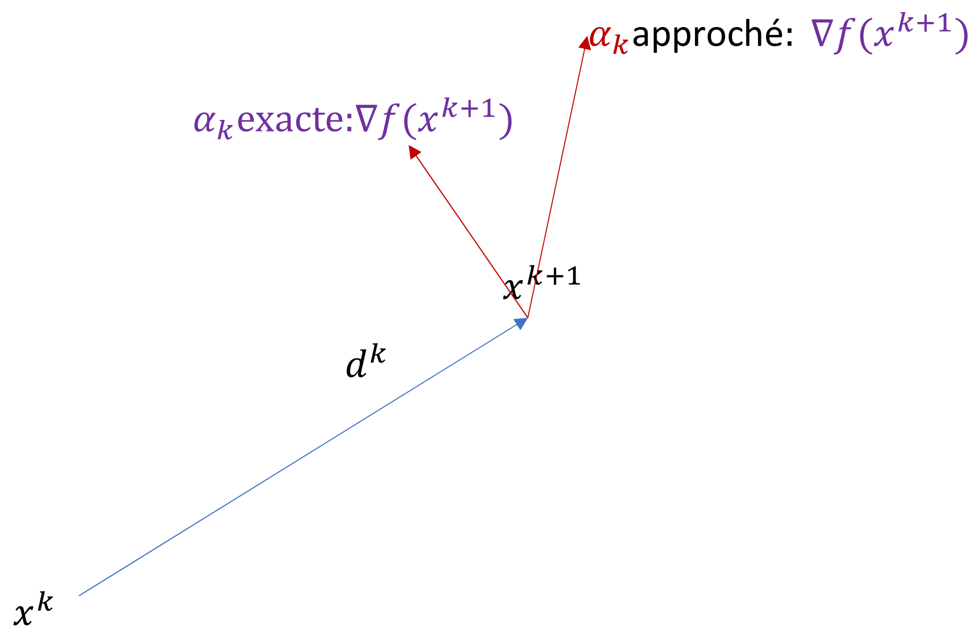
$$\alpha_k {d^k}^T y^k = \alpha_k {d^k}^T \left[\nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)\right] = -\alpha_k {d^k}^T \nabla f\left(x^k\right) > 0$$

Recherche linéaire du pas approchée

1. **Introduction**
2. **Règles sur le pas approchée:**
    1. **Armijo**
    2. **Goldstein**
    3. **Wolfe**

Recherche linéaire du pas approchée: Introduction

- $\alpha_k \in [I, S]$



Recherche linéaire du pas approchée: Introduction

Dans plusieurs méthodes itératives, pour identifier x^{k+1} il faut résoudre le problème unidimensionnelle

$$\min_{x} \varphi(\alpha) = \min_{\alpha>0} f\left(x^k + \alpha d^k\right) = f\left(x^k + \alpha_k d^k\right) = \varphi(\alpha_k)$$

Afin de déterminer le pas optimal

- Au lieu d'effectuer la chercher du pas optimaux qui pourrait être couteuse, parfois il est préférable de trouver juste un pas qui au moins **assure la convergence**.
- Soit l'approximation linéaire de φ à 0 est obtenue à partir du développement de Taylor d'ordre 1 :

$$\varphi(\alpha) \approx \bar{\varphi}(\alpha) = \varphi(0) + \varphi'(0)\alpha$$

Tel que :

$\bar{\varphi}(0) = \varphi(0)$ et $\bar{\varphi}'(\alpha) = \varphi'(0)$

Recherche linéaire du pas approchée: Introduction

Pour déterminer une approximation acceptable du pas optimal α_k, nous considérons l'approximation suivant de φ

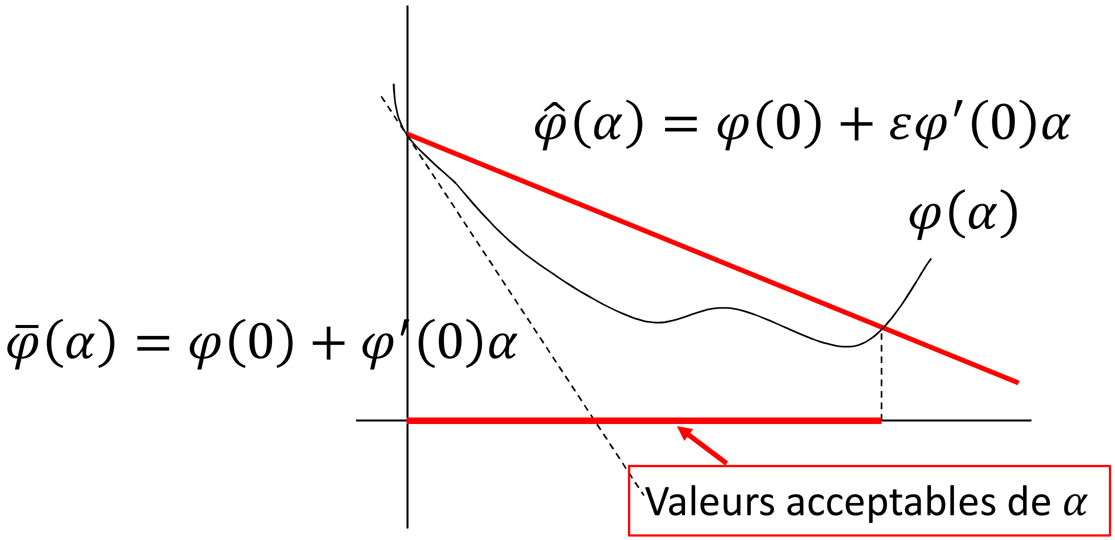
$$\hat{\varphi}(\alpha) = \varphi(0) + \varepsilon\varphi'(0)\alpha$$

Pour une valeur fixée de ε entre 0 et 1 ($\varepsilon = 0.2$, par exemple)

Une valeur α est acceptable tout en n'étant pas trop grande,

Si $\varphi(\alpha)$ n'excède pas l'approximation $\hat{\varphi}(\alpha)$, i.e.

$$\varphi(\alpha) \leq \hat{\varphi}(\alpha) = \varphi(0) + \varepsilon\varphi'(0)\alpha \qquad (3.3)$$



Recherche linéaire du pas approchée: Règle d'Armijo

Pour s'assurer que la valeur de α choisie n'est pas trop petite α doit satisfaire un des tests additionnels suivants :

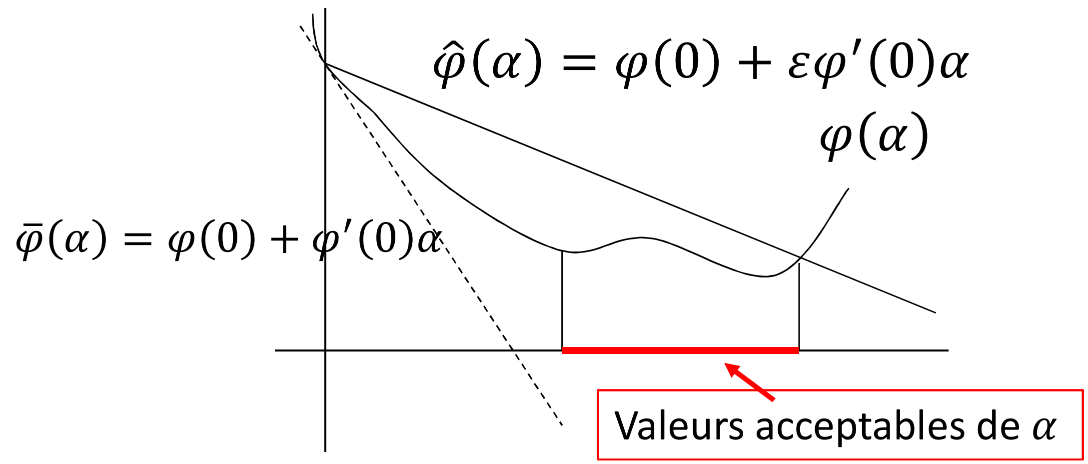
Règle d'Armijo :

la valeur α est telle que

$$\varphi(\eta\alpha) > \hat{\varphi}(\eta\alpha) = \varphi(0) + \varepsilon\varphi'(0)\eta\alpha \quad (3.4)$$

Où $\eta > 1$ ($\eta = 2$, par exemple)

Ainsi, si α est multiplié par un facteur égal à η alors la relation (3.3) n'est plus vérifiée.

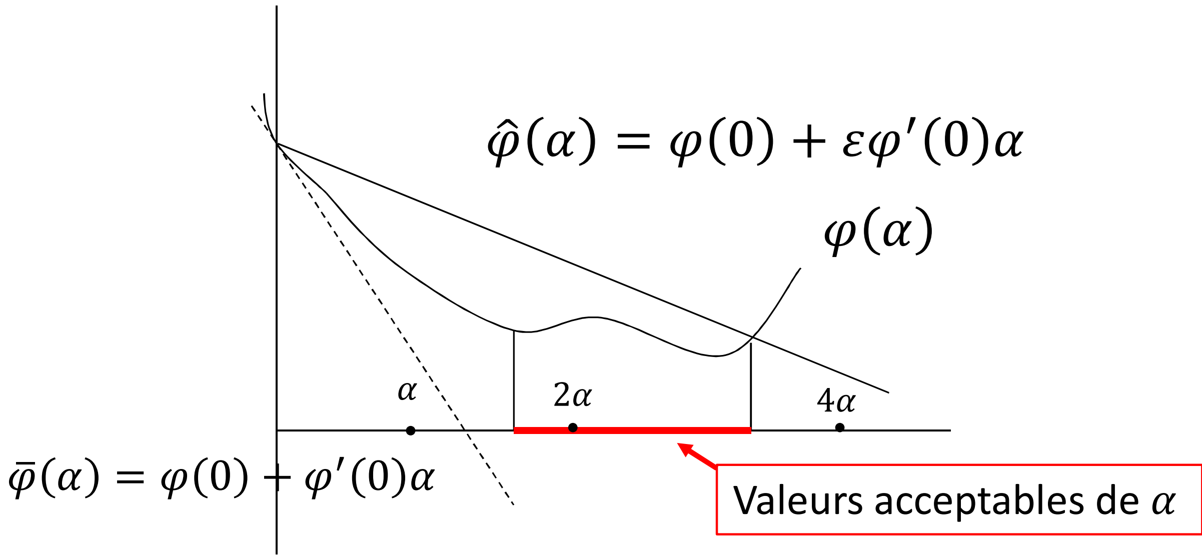


Recherche linéaire du pas approchée: Règle d'Armijo

$$\varphi(2\alpha) > \hat{\varphi}(2\alpha) = \varphi(0) + \varepsilon\varphi'(0)2\alpha$$

Algorithme de la règle d'Armijo :

- Choisir une certaine valeur de α
- Si (3.3) est satisfaite, multiplier à répétition par η jusqu'à ce que (3.3) ne soit plus satisfaite.
- Alors la valeur précédente parmi la suite des valeurs obtenues est celle du pas retenu.

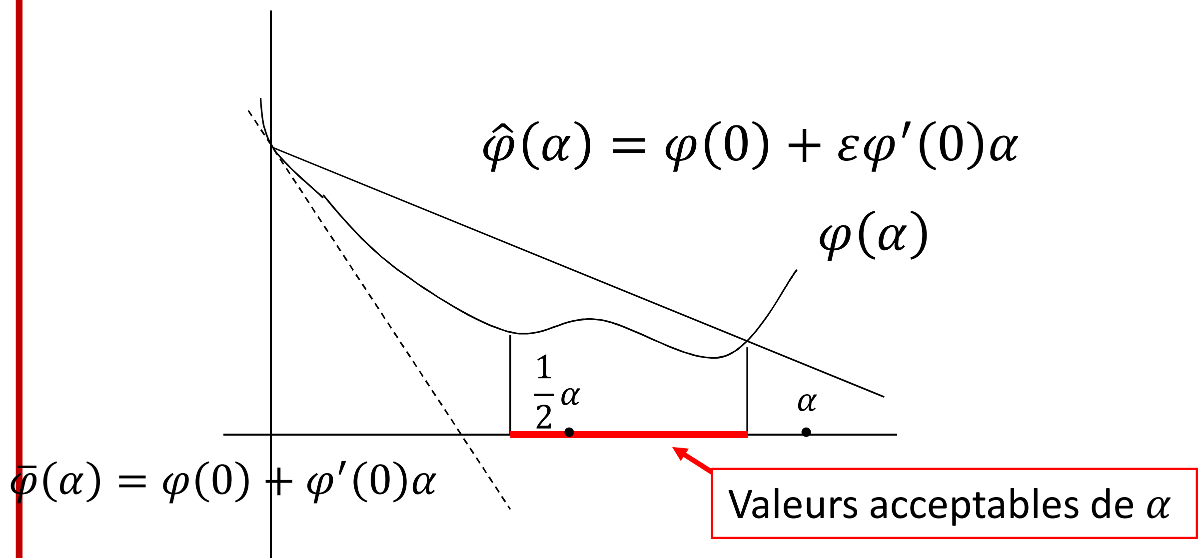


Recherche linéaire du pas approchée: Règle d'Armijo

Algorithme de la règle d'Armijo :

- Si (3.3) n'est pas satisfaite, diviser à répétition par η jusqu'à ce que (3.3) soit vérifiée.

- Cette dernière valeur satisfaisant (3.3) est celle du pas retenu puisque la multiplier par η entraine que (3.3) n'est plus vérifiée.



Recherche linéaire du pas approchée: Règle d'Armijo

Algorithme de la règle d'Armijo

L'essentiel de la règle d'Armijo c'est que le pas α est accepté si et seulement si il satisfait au (3) et (4).

$$\varphi(\alpha) \leq \hat{\varphi}(\alpha) = \varphi(0) + \varepsilon\varphi'(0)\alpha \qquad (3)$$
$$\varphi(\eta\alpha) > \hat{\varphi}(\eta\alpha) = \varphi(0) + \varepsilon\varphi'(0)\eta\alpha \qquad (4)$$

Algorithme:

Entrée: $\alpha_0 > 0, \varepsilon \in (0,1), \eta > 1$

Mettre $\alpha = \alpha_0$

- Si α vérifier (3)
  - Effectuer **Parcours en avant jusqu'à** α vérifier (4)
- Sinon (**α ne vérifier pas (3) $\Rightarrow$ α vérifier (4)**)
  - Effectuer **Parcours en arrière avant jusqu'à** α vérifier (3)

Recherche linéaire du pas approchée: Règle d'Armijo

- **Parcours en avant: sous hypothèse que 4 n'est pas encore vérifié**
  1. Entrée: $\alpha_0 > 0, \varepsilon \in (0,1), \eta > 1$
  2. Mettre $\alpha = \alpha_0$
  3. Si $\varphi(\eta\alpha) > \hat{\varphi}(\eta\alpha) = \varphi(0) + \varepsilon\varphi'(0)\eta\alpha$ ((4) est satisfait) retourner α et arrêter. Sinon Substitue $\alpha = \eta\alpha$ et retourne à 2

- **Parcours en arrière: sous hypothèse que 3 n'est pas encore vérifié**
  1. Entrée: $\alpha_0 > 0, \varepsilon, \eta > 1$
  2. Mettre $\alpha = \alpha_0$
  3. Si $\varphi(\alpha) \leq \varepsilon\alpha\varphi'(0) + \varphi(0)$ ((3) est satisfaite) retourner α et arrêter. Sinon Substitue $\alpha = \frac{\alpha}{\eta}$ et retourne à 2

Recherche linéaire du pas approchée: Règle de Goldstein

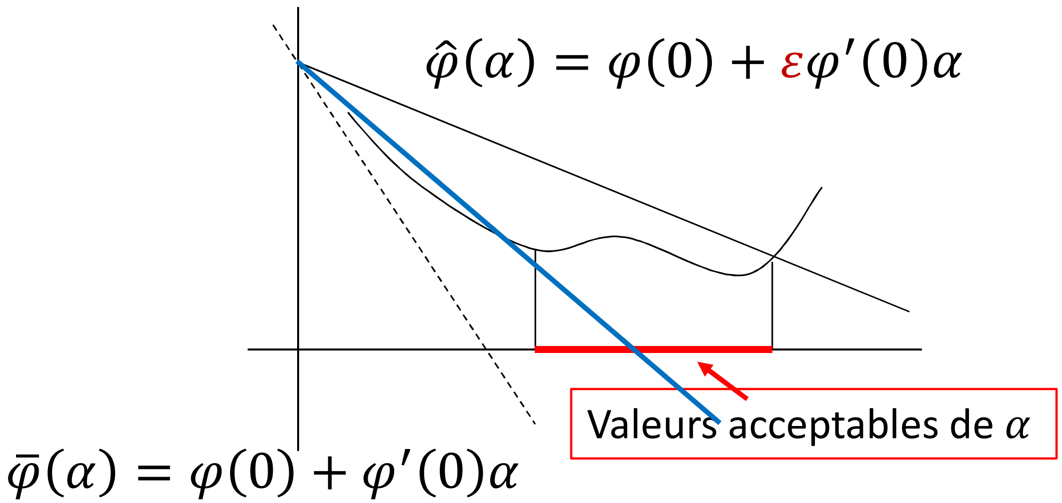
Pour s'assurer que la valeur de α choisie n'est pas trop petite α doit satisfaire un des tests additionnels suivants :

Règle de Goldstein :

la valeur α est telle que

$$\varphi(\alpha) > \varphi(0) + (1-\rho)\varphi'(0)\alpha \quad (3.5)$$

où $0 < \rho < \frac{1}{2}$



Recherche linéaire du pas approchée: Règle de Goldstein

Algorithme Goldstein

Etape 1 : Prendre $\boldsymbol{\alpha_0}$ dans $[\mathbf{0}, +\infty[$ où $[\mathbf{0}, \boldsymbol{\alpha_{max}}[$

- Calculer $\boldsymbol{\varphi}(\mathbf{0})$, $\boldsymbol{\varphi}'(\mathbf{0})$. Prendre $\boldsymbol{\rho} \in (\mathbf{0}, \frac{\mathbf{1}}{\mathbf{2}})$, $\boldsymbol{t} > \mathbf{1}$,
- Mettre $\boldsymbol{\alpha_0} = \mathbf{0}$, $\boldsymbol{b_0} = +\infty$ ou $\boldsymbol{\alpha_{max}}$ $k = 0$

Etape 2 : Vérifier la règle $\mathbf{3}$: calcule $\boldsymbol{\varphi}(\boldsymbol{\alpha_k})$

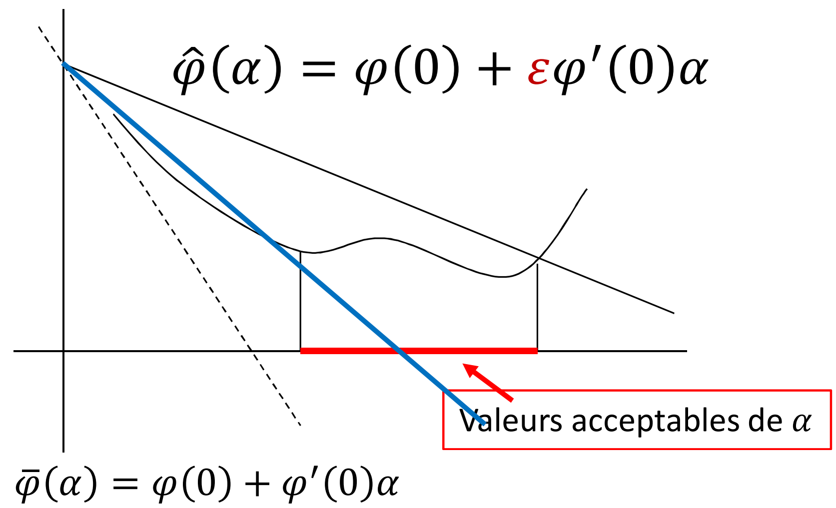
- Si $\boldsymbol{\varphi}(\boldsymbol{\alpha_k}) \leq \boldsymbol{\varphi}(\mathbf{0}) + \boldsymbol{\rho}\boldsymbol{\alpha_k}\boldsymbol{\varphi}'(\mathbf{0})$ Passer à l'étape 3
- Sinon mettre $\boldsymbol{a_{k+1}} = \boldsymbol{a_k}$ et $\boldsymbol{b_{k+1}} = \boldsymbol{\alpha_k}$ passer à l'étape 4

Etape 3 : Vérifier la règle 4 :

- Si $\boldsymbol{\varphi}(\boldsymbol{\alpha_k}) \geq \boldsymbol{\varphi}(\mathbf{0}) + (\mathbf{1} - \boldsymbol{\rho})\boldsymbol{\alpha_k}\boldsymbol{\varphi}'(\mathbf{0})$ arrête, et renvoyer $\boldsymbol{\alpha_k}$
- sinon, mettre $\boldsymbol{a_{k+1}} = \boldsymbol{\alpha_k}$, $\boldsymbol{b_{k+1}} = \boldsymbol{b_k}$
- Si $\boldsymbol{b_{k+1}} < +\infty$, passer à l'étape 4
- sinon mettre $\boldsymbol{\alpha_{k+1}} = \boldsymbol{t}\boldsymbol{\alpha_k}$ et $\boldsymbol{k} = \boldsymbol{k} + \mathbf{1}$, Et passer à l'étape 2.

Etape 4 : Choisir un nouveau point mettre $\boldsymbol{\alpha_{k+1}} = \frac{\boldsymbol{a_{k+1}} + \boldsymbol{b_{k+1}}}{\mathbf{2}}$

- Et $\boldsymbol{k} = \boldsymbol{k} + \mathbf{1}$ et passer à l'étape 2.



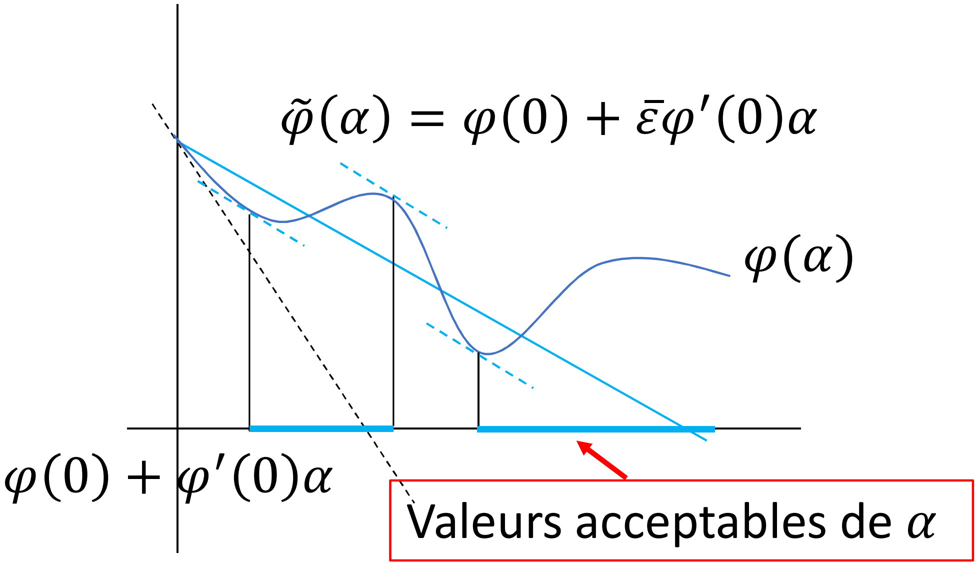
Recherche linéaire du pas approchée: Règle de Wolfe

Pour s'assurer que la valeur de α choisie n'est pas trop petite α doit satisfaire un des tests additionnels suivants :

Règle de Wolfe :

Une valeur α est acceptable tout en préservant la convergence,

Si $\varphi'(\alpha)$ est plus grande que $\bar{\varepsilon}\varphi'(0)$, i.e.



$$\varphi'(\alpha) \geq \bar{\varepsilon}\varphi'(0) \quad \text{où} \quad 0 < \bar{\varepsilon} < \frac{1}{2} \qquad (3.4)$$

Recherche linéaire du pas approchée: Convergence

Théorème :

- Soit α_k dans algorithme 3 qui est défini par règle de Goldstein ou Wolfe,
- soit s^k tel que à $\theta_k \leq \frac{\pi}{2} - \mu$

Si ∇f existe et uniformément continue sur l'ensemble $\Omega = \{x \mid f(x) \leq f(x_0)\}$ alors $\nabla f\left(x^k\right) = 0$ Pour certains k, ou $f\left(x^k\right) \rightarrow -\infty$, ou $\nabla f\left(x^k\right) \rightarrow 0$.