Cours 3 : Optimisation sans contraintes
Méthodes de Newton

Optimisation déterministe

EISTI
Ingénieurs $2^{\text{ème}}$ année GI
2010-2011

Nisrine Fortin Camdavant

1. Méthode de Newton
   - Algorithme de Newton dans $\mathbb{R}$
   - Algorithme de Newton dans $\mathbb{R}^n$

2. Les méthodes de Quasi-Newton

Problème d'optimisation **sans contraintes**

$$(\mathcal{P}) \quad \begin{cases} \min J(x) \\ x \in \mathbb{R}^n \end{cases}$$

J une fonction de $\mathbb{R}^n$ dans $\mathbb{R} \cup \{+\infty\}$.

- **Méthodes de descente** : $x_{k+1} = x_k + \rho_k d_k$
  - On choisit le vecteur d_k une direction de descente et ρ_k le pas de descente.
  - Le choix de d_k et ρ_k se fait de sorte que $J(x_{k+1}) < J(x_k)$.
- Méthodes de descente :
  1. point initial $\mathbf{x}_0$
  2. pour $k \geq 1$ croissant
     - 2.1 choisir une **direction de descente** $d_k \neq 0$ ($\langle d_k, \nabla J(x_k)\rangle < 0$)
     - 2.2 choisir un **pas de descente** $\rho_k > 0$
     - 2.3 poser $\mathbf{x}_{k+1} = \mathbf{x}_k + \rho_k \mathbf{d}_k$
     - 2.4 tester la convergence

La méthode de Newton n'est pas une méthode d'optimisation à proprement parler. C'est en réalité une méthode utilisée pour résoudre des équations non linéaires de la forme $F(x) = 0$ où F est une fonction de $\mathbb{R}^n$ dans $\mathbb{R}^n$.

- Méthode de recherche d'un zéro d'une fonction f ($f \in \mathcal{C}^1(\mathbb{R})$) :
  - approximation linéaire de f :

$$f(x+h) = f(x) + hf^{'}(x) + o(h)$$

  - au premier ordre, $f(x+h) = 0$ conduit à $h = -\dfrac{f(x)}{f'(x)}$
  - algorithme : $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$
  - convergence quadratique
- Application à la minimisation de f ($f \in \mathcal{C}^2(\mathbb{R})$) :
  - on cherche à résoudre l'équation d'Euler $f^{'}(x) = 0$
  - algorithme

$$x_{n+1} = x_n - \frac{f^{'}(x_n)}{f''(x_n)}$$

Cette méthode est aussi appelée **méthode de la tangente**. Chaque itéré x_{k+1} est obtenu à partir du précédent en traçant la tangente à la courbe de f au point $(x_k, f(x_k))$ et en prenant son intersection avec l'axe des abscisses.

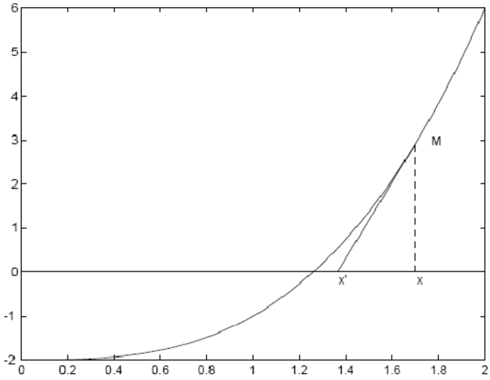

Figure1. Interprétation géométrique de la méthode de Newton

Algorithme de Newton dans $\mathbb{R}$

1. Initialisation

   $k = 0$: choix de $x_0 \in \mathbb{R}$ dans un voisinage de x^*.

2. Itération k

   $$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} ;$$

3. Critère d'arrêt

   Si $\|x_{k+1} - x_k\| < \epsilon$, STOP

   Sinon, on pose $k = k + 1$ et on retourne à 2.

- Il faut assurer la convergence de la suite $(x_k)_k$ vers la solution x^*,
- il faut que cette suite soit bien définie, c'est à dire montrer que $f^{'}(x_k) \neq 0$ à l'étape 2.

- On cherche un zéro de ∇J, or

$$\nabla J(x+h) = \nabla J(x) + \nabla^2 J(x) h + o\left(\|h\|\right)$$

donc au premier ordre

$$h = -\nabla^2 J(x)^{-1} \nabla J(x)$$

- correspond aussi au minimum de l'approximation au deuxième ordre

$$J(x+h) = J(x) + h^t \nabla J(x) + \frac{1}{2} h^t \nabla^2 J(x) h + o\left(\|h\|^2\right)$$

- si $\nabla^2 J(x)$ est définie positive, h est une direction de descente

$$\langle \nabla J(x), -\nabla^2 J(x)^{-1} \nabla J(x) \rangle < 0$$

Méthode de Newton en optimisation

Algorithme de Newton

1. Initialisation : choix d'un point initial x_0.
2. Pour $k \geq 1$ croissant
   - 2.1 calculer $d_k = -\nabla^2 J(x_k)^{-1} \nabla J(x_k)$
   - 2.2 tester la convergence par

$$\langle \nabla J(x_k), \nabla^2 J(x_k)^{-1} \nabla J(x_k) \rangle < \epsilon$$

   - 2.3 poser $x_{k+1} = x_k + d_k$

Méthode de Newton

- L'étape 2.2 de la méthode revient à résoudre le système linéaire suivant :

$$\nabla^2 J(x_k) d_k = \nabla J(x_k)$$

puis à poser $x_{k+1} = x_k - d_k$

- Dans l'algorithme de Newton pour l'optimisation sans contraintes, on détermine une direction d_k par la formule suivante :

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

Cette direction est appelée direction de Newton.

- Il faut que le hessien de J en l'itéré courant soit inversible pour que cette définition ait un sens.

Méthode de Newton

- **Inconvénient majeur :**
  - sensibilité au choix du point de départ : la convergence est locale.
  - choisir le point de départ x_0 "assez près" de x^*. Puisqu'on ne connait pas x^*, en pratique on essaie de s'approcher de x^* par une méthode de type gradient par exemple, puis on applique la méthode de Newton.
- **Avantage :**
  - grande rapidité.
  - La convergence est quadratique, c'est à dire que l'erreur $e_k = \|x_{k+1} - x_k\|$ est élevée au carré à chaque itération. Concrètement, si elle vaut 10^{-2} à l'étape k elle vaudra 10^{-4} à l'étape $k+1$ et 10^{-8} à l'étape $k+2$.

Les méthodes de Quasi-Newton : principe

Les méthodes du **Quasi-Newton** consistent à imiter l'algorithme de Newton, mais sans calculer le Hessien de J ni son inverse. Au lieu de procéder à l'itération $x_{k+1} = x_k - \nabla^2 J(x_k)^{-1} \nabla J(x_k)$, on calcule

$$x_{k+1} = x_k - \alpha_k S_k \nabla J(x_k),$$

avec S_k une approximation symétrique de $\nabla^2 J(x_k)^{-1}$ et α_k un paramètre positif fourni par une recherche linéaire le long de la direction

$$d_k = -\alpha_k S_k \nabla J(x_k)$$

Les méthodes dites quasi-Newton gardent la rapidité de la méthode de Newton, evitent le calcul (coûteux) de la matrice $\nabla^2 J(x_k)$ à chaque itération et sont plus robustes par rapport au point de départ. La méthode BFGS (pour Broyden-Fletcher-Goldfarb-Shanno) propose un calcul de H_{k+1} (approximation de $\nabla^2 J(x_k)^{-1}$) en fonction de H_k grâce à des formules algébriques.

Les méthodes de Quasi-Newton

On note $\delta_k = x_{k+1} - x_k$, $y_k = \nabla J(x_{k+1}) - \nabla J(x_k)$ et la condition de Quasi-Newton s'écrit donc $S_{k+1} y_k = \delta_k$.

Algorithme de B.F.G.C

- On choisit une matrice S_0, symétrique définie positive, et un point x_0
- à l'étape k, la recherche linéaire se fait dans la direction $d_k = -S_k \nabla J(x_k)$ et produit α_k. D'où

$$x_{k+1} = x_k + \alpha_k d_k, \quad \delta_k = x_{k+1} - x_k = \alpha_k d_k,$$

avec $y_k = \nabla J(x_{k+1}) - \nabla J(x_k)$, on remet à jour la matrice S_k par

$$S_{k+1} = S_k + \left(1 + \frac{y_k^t S_k y_k}{\delta_k^t y_k}\right) \frac{\delta_k \delta_k^t}{\delta_k^t y_k} - \frac{\delta_k y_k^t S_k + S_k y_k \delta_k^t}{\delta_k^t y_k}$$

Bibliographie :

- Introduction à l'optimisation, Jean Christophe Culioli, ellipses,1994.
- Introduction à l'analyse numérique matricielle et à l'optimisation PG. CIARLET MASSON 1990
- Inverse Problem Theory and Methods for Model Parameter Estimation Albert Tarantola SIAM 2005.
- Convexité et optimisation Guy Cohen Ecole Nationale des Ponts et Chaussées et INRIA 2000.
- Optimisation sans contraintes M. Bergounioux http ://www.univ-orleans.fr/mapmo/membres/maitine/MASTER2/FAP/optimFAP.pdf
- Introduction à l'analyse numérique http ://www.math.uvsq.fr/jaisson/ENSAE cours45.pdf
- Algorithmes d'optimisation non-linéaire sans contrainte http ://www.math.u-bordeaux1.fr/ bergmann/PDF/These/annexeC.pdf