# Cours 2 : Optimisation sans contraintes
# Méthode du gradient conjugué

Optimisation déterministe

EISTI
Ingénieurs 2ème année GI
2010-2011

Nisrine Fortin Camdavant

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 $\rho_k$ le pas de descente.
  - Le choix de $d_k$ et $\rho_k$ 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

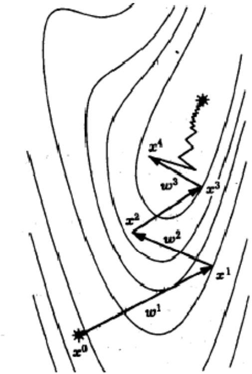


Fig.1 Principe des méthodes de descente

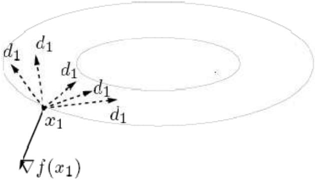


Fig.2 Direction de descente $\langle \nabla f(x_1), d_1 \rangle < 0$

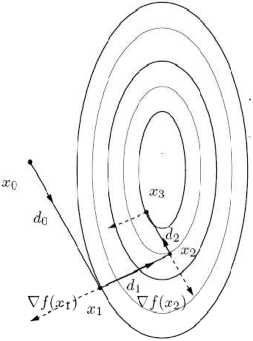


Fig.3 Méthodes de descente à pas optimal

## Méthodes du Gradient : $x_{k+1} = x_k + \rho_k d_k$

- La direction de descente : $d_k = -\nabla J(x_k)$.
- Le pas de descente : $\rho_k$ est choisi de sorte que $J(x_{k+1}) < J(x_k)$.

### Variantes de la méthode :

- Méthode du gradient à pas constant : on choisit $\rho_k$ constant.
- Méthode du gradient à pas optimal :
  - $\rho_k$ est choisi comme minimum de la fonction

$$f(\rho) = J(x_k + \rho d_k).$$

  - $\rho_k$ est choisi de façon à diminuer suffisamment la valeur de la fonction $J$. On utilisera la règle d'Armijo.

## Méthodes du Gradient

### Algorithme du Gradient

1. Initialisation

   $k = 0$ : choix de $x_0$ et de $\epsilon > 0$

2. Itération $k$

   $x_{k+1} = x_k - \rho_k \nabla J(x_k)$ ;

3. Critère d'arrêt

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

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

Dans tout ce qui suit, $\epsilon$ est un réel positif (petit) donné qui représente la précision désirée.

Lorsqu'on dispose d'informations supplémentaires sur la fonctionnelle $J$ (constante de Lipchitz $L$ de la dérivée, constante de forte convexité $a$,...) on peut choisir $\rho = \frac{a}{L^2}$.

Algorithme du Gradient à pas fixe

1. Initialisation

   $k = 0$ : choix de $x_0$, de $\rho > 0$ et de $\epsilon > 0$

2. Itération $k$

   $x_{k+1} = x_k - \rho \nabla J(x_k)$ ;

3. Critère d'arrêt

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

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

Déterminer le pas qui *minimise* la fonction $J$ le long de la direction de descente $d_k$. Le pas $\rho_k$ est donc solution du problème :

$$\rho_k = \arg\min_{\rho>0} J(x_k + \rho d_k)$$

`Algorithme du Gradient à pas optimal`

1. `Initialisation`

   $k = 0$ : `choix de` $x_0$ `et de` $\epsilon > 0$
2. `Itération` $k$

   $\rho_k = \arg\min_{\rho>0} J(x_k - \rho\nabla(J(x_k)))$ ;

   $x_{k+1} = x_k - \rho_k \nabla J(x_k)$ ;
3. `Critère d'arrêt`

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

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

# Principes

Nous supposons qu'à l'itération $k$, la direction de descente $d_k$ est calculée.
Notations : $q(t) = J(x_k + td_k)$. Nous supposons que nous savons calculer une approximation numérique de $q(t)$ et de $q'(t) = \langle \nabla J(x_k + td_k), d_k \rangle$ pour chaque $t$, ces calculs pouvant être couteux. Il s'agit alors de trouver $t$ qui minimise $q$ (on minimise $J$ dans la direction $d_k$ ) ou du moins un $t$ convenable. Une recherche linéaire consiste alors à tester $t$ pour savoir s'il est trop grand ou trop petit

Choisir un pas qui *diminue* suffisamment la valeur de la fonction : délimiter un intervalle convenable pour $\rho$ et choisir le plus grand $\rho_k$ possible sur cet intervalle. Conditions d'Armijo-Goldstein

$$J(x_k + \rho_k d_k) \leq J(x_k) + \rho_k \beta_1 \langle \nabla J(x_k), d_k \rangle, \qquad \beta_1 \in ]0, 1[$$

## Algorithme d'Armijo

Initialisation : choix d'un pas $\rho_k^1 > 0$ et d'un paramètre $\tau \in ]0, 1[$, $i = 1$.

1. Test : le pas $\rho_k^i$ est accepté s'il vérifie

$$J\left(x_k + \rho_k^i d_k\right) \leq J\left(x_k\right) + \omega_1 \rho_k^i \langle \nabla J(x_k), d_k \rangle,$$

   Sinon :

2. On choisit $\rho_k^{i+1} \in \left[\tau \rho_k^i, (1-\tau)\rho_k^i\right]$
3. On pose $i = i + 1$ et $\rho_k = \rho_k^i$, puis on retourne à l'étape 1.

Le paramètre $\tau$ est habituellement choisi égal à $10^{-2}$. Le pas obtenu par cet algorithme est appelé pas d'Armijo. La constante $\omega_1$ est choisie arbitrairement et est habituellement prise très petite (de l'ordre de $10^{-4}$) afin que la condition d'Armijo soit plus aisément satisfaite. Mais il est souvent préférable de choisir $\omega_1 < 0, 5$.

## Algorithme du Gradient avec la règle d'Armijo

1. Initialisation

   $k = 0$ : choix de $x_0$ et de $\epsilon > 0$

2. Itération $k$

   $\rho_k$ calculé avec la règle d'Armijo

   $x_{k+1} = x_k - \rho_k \nabla J(x_k)$ ;

3. Critère d'arrêt

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

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

- Principe des méthodes de descentes : choisir des directions de descente.
- Possibilités de revenir en arrière et de prendre des directions déjà explorées et "mauvaises".
- **le gradient conjugué** construit une suite de directions que l'on garde en mémoire : on cherche $x_{k+1}$ tel que

$$J(x_{k+1}) = \min_{v \in G_k} J\left(x_k + v\right)$$

$$G_k = Vect\left(\nabla J(x_0), \ldots, \nabla J(x_k)\right)$$

Les directions de descente sont donc des combinaisons linéaires de tous les gradients précedents.

$$x_{k+1} = x_k + \rho_k p_k, \quad p_k = \sum_{i=0}^{k} \alpha_i \nabla J(x_i)$$

## Fonctions quadratiques

$$J(x) = \frac{1}{2}\langle Ax, x\rangle + \langle b, x\rangle, \qquad x_{k+1} = x_k + \rho_k p_k$$

$A$ est une matrice carrée symétrique définie positive de taille $n$.

## Algorithme du Gradient conjugué

1. Initialisation

   $k = 0$ : choix de $x_0$ et de $\epsilon > 0$

   Poser $p_0 = -\nabla J(x_0) = -Ax_0 - b$

2. Itération $k$

   Tant que $\|\nabla J(x_k)\| > \epsilon$, faire

   $$\rho_k = \frac{\|\nabla J(x_k)\|^2}{\langle Ap_k, p_k\rangle}$$

   $$x_{k+1} = x_k + \rho_k p_k$$

   $$p_{k+1} = -\nabla J(x_{k+1}) + \frac{\|\nabla J(x_{k+1})\|^2}{\|\nabla J(x_k)\|^2} p_k$$

   Poser $k = k + 1$

   Fin tant que

   $x_{min} = x_k$

## Algorithme du gradient conjugué de Polak-Ribière

1. $k = 0$ : choix de $x_0$ et de $\epsilon > 0$
2. Poser $p_0 = -\nabla J(x_0)$
3. Poser $k = 0$
4. Tant que $\|\nabla J(x_k)\| > \epsilon$, faire $\quad \rho_k = \dfrac{\|\nabla J(x_k)\|^2}{\langle H(x_{k+1}) p_k, p_k \rangle}$
5. $x_{k+1} = x_k + \rho_k p_k$
6. Si $k+1 = n$, poser $x_0 = x_{k+1}$, $\nabla J(x_0) = \nabla J(x_{k+1})$ passer à (3) sinon passer à (9)
7. Calculer $\beta_k = \dfrac{\langle \nabla J(x_{k+1}), \nabla J(x_{k+1}) - \nabla J(x_k) \rangle}{\|\nabla J(x_k)\|^2}$
8. Calculer $p_{k+1} = -\nabla J(x_{k+1}) + \beta_k p_k$
9. Poser $k = k + 1$

   Fin tant que
10. $x_{min} = x_k$