

MAT 2410: Optimisation

Robert Guénette

Département de Mathématiques et de Statistique

Chapitre 3 Optimisation différentiable sans contrainte

Références

Condition d'optimalité

Condition du premier ordre

Condition du deuxième ordre

Méthodes numériques

Généralité

Méthode de descente

Méthode de Newton

Méthode quasi-Newton

Méthode du gradient conjugué

Méthodes de recherche linéaire

Méthodes directes

Références:

- Notes de cours: chapitre 3, section 3.1.
- Livre de M. Delfour: chapitre 3.

Condition nécessaire d'optimalité du premier ordre

Théorème:

Soit $f : U \rightarrow \mathbb{R}$ une fonction numérique de classe C^1 définie sur un ensemble ouvert $U \subset \mathbb{R}^n$. Si $x \in U$ est un point minimisant (minimum local ou global), alors

$$\nabla f(x) = 0.$$

Preuve: Fixons $v \in \mathbb{R}^n$. La fonction $t \rightarrow f(x + tv)$ atteint un minimum en $t = 0$. Par conséquent

$$\frac{d}{dt} f(x + tv)|_{t=0} = (\nabla f(x), v) = 0, \quad v \in \mathbb{R}^n,$$

d'où le résultat.

Condition nécessaire et suffisante d'optimalité

Théorème:

Soit $f : U \rightarrow \mathbb{R}$ une fonction numérique convexe de classe C^1 définie sur un ensemble ouvert convexe $U \subset \mathbb{R}^n$.

- ▶ $x \in U$ est un point minimisant de f si et seulement si $\nabla f(x) = 0$.

Preuve: Il suffit de montrer l'implication \Leftarrow .

f est convexe si

$$f(y) \geq f(x) + (\nabla f(x), y - x) \quad \forall y \in U$$

Mais $\nabla f(x) = 0 \implies f(y) \geq f(x) \quad \forall y \in U$, d'où le résultat.

Condition nécessaire d'optimalité du deuxième ordre

Théorème:

Soit $f : U \rightarrow \mathbb{R}$ une fonction numérique de classe C^2 définie sur un ensemble ouvert $U \subset \mathbb{R}^n$. $x \in U$ est un point minimisant de f , alors

- ▶ $\nabla f(x) = 0$
- ▶ $H_f(x) \geq 0$ où H_f est la matrice hessienne.

Preuve: Montrons que $H_f(x) \geq 0$. Par Taylor, on a

$$f(y) = f(x) + (\nabla f(x), y - x) + \frac{1}{2} (H(y - x), (y - x))$$

où $H = H(x + \bar{t}(y - x))$ pour $0 < \bar{t} < 1$. Ceci est valide dans un voisinage V du point x . Mais

$$\nabla f(x) = 0 \implies f(y) - f(x) = \frac{1}{2} (H(y - x), (y - x)).$$

Par un argument de continuité et $y \rightarrow x$, on obtient que

$$(H_f(x)(y - x), (y - x)) \geq 0$$

Pour tout $v \in \mathbb{R}^n$, il est toujours possible de choisir $r \in \mathbb{R}$ de sorte que $y = x + rv \in V$. En posant $y - x = rv$, on obtient le résultat.

Condition suffisante d'optimalité du deuxième ordre

Théorème:

Soit $f : U \rightarrow \mathbb{R}$ une fonction numérique de classe C^2 définie sur un ensemble ouvert $U \subset \mathbb{R}^n$.

(a) Si le point $x \in U$ vérifie

$$\nabla f(x) = 0 \quad \text{et} \quad H_f(x) > 0,$$

où H_f est la matrice hessienne au point x , alors x est un minimum local dans U .

(b) S'il existe un voisinage V du point $x \in U$ vérifiant

$$\nabla f(x) = 0 \quad \text{et} \quad H_f(y) \geq 0 \quad \forall y \in V,$$

alors x est un minimum global dans V et local dans U .

Preuve de (a)

Si $H_f(x) > 0$, on aura que localement, il existe un voisinage V vérifiant $x \in V \subset U$ tel que $\implies H_f(y) > 0 \quad \forall y \in V$.

Par Taylor, on a

$$f(y) = f(x) + (\nabla f(x), y - x) + \frac{1}{2} (H(y - x), (y - x))$$

où $H = H(x + \bar{t}(y - x))$ pour $0 < \bar{t} < 1$.

Mais $\nabla f(x) = 0$ et que $H = H(x + \bar{t}(y - x)) > 0$.

Ceci implique

$$f(y) = f(x) + \frac{1}{2} (H(y - x), (y - x)) \geq f(x) \quad \forall y \in V.$$

Donc x est un minimum local.

Preuve de (b)

Pour $y \in V$, on a

$$f(y) = f(x) + (\nabla f(x), y - x) + \frac{1}{2} (H(y - x), (y - x))$$

Mais $\nabla f(x) = 0$ et $H_f(y) \geq 0 \quad \forall y \in V \implies H \geq 0$ Donc

$$f(y) = f(x) + (\nabla f(x), y - x) + \frac{1}{2} (H(y - x), (y - x))$$

$$f(y) = f(x) + \frac{1}{2} (H(y - x), (y - x))$$

$$f(y) \geq f(x)$$

d'où le résultat.

Méthodes numériques de minimisation

On considère le problème de minimisation $f(\bar{x}) = \min_{x \in U} f(x)$ où $f : U \rightarrow \mathbb{R}$ est une fonction numérique définie sur un ensemble ouvert $U \subset \mathbb{R}^n$.

On cherche à produire une suite ayant les propriétés suivantes:

- ▶ $f(x_{k+1}) \leq f(x_k)$
- ▶ $x_k \rightarrow \bar{x}$ où \bar{x} est un point minimisant (local ou global).

Les méthodes sont regroupées en deux familles.

a) Les méthodes de descentes

$$\begin{cases} x_0 \in \mathbb{R}^n, \\ x_{k+1} = x_k + \rho_k d_k. \end{cases}$$

où d_k est la direction de descente et $\rho_k > 0$.

b) Les méthodes de type Newton. On résout le système non linéaire

$$\nabla f(x) = 0.$$

Méthode du gradient

Dans la méthode du gradient, on choisit

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

comme direction de descente car

$$\begin{aligned} f(x_{k+1}) &= f(x_k - \rho_k \nabla f(x_k)) \\ &= f(x_k) - \rho_k (\nabla f(x_k), \nabla f(x_k)) + \frac{1}{2} \rho_k^2 (H \nabla f(x_k), \nabla f(x_k)) \\ &\simeq f(x_k) - \rho_k \|\nabla f(x_k)\|^2 \leq f(x_k) \end{aligned}$$

Algorithme du gradient:

- ▶ $x_0 \in \mathbb{R}^n$ donné
- ▶ $x_{k+1} = x_k - \rho_k \nabla f(x_k)$ où $\rho_k > 0$.

Si $\rho_k = \rho$, l'algorithme est dit à pas constant.

Critère d'arrêt

On peut utiliser un (ou une combinaison) des critères suivants pour arrêter les itérations d'un algorithme de descente.

- ▶ Etant donné que $\nabla f(x_k) \rightarrow \nabla f(\bar{x}) = 0$, on pose

$$\|\nabla f(x_k)\| < \epsilon_1.$$

- ▶ On a $x_k \rightarrow \bar{x}$, on peut aussi prendre

$$\|x_{k+1} - x_k\| < \epsilon_2.$$

- ▶ Finalement, on a $f(x_k) \rightarrow f(\bar{x})$, on peut prendre

$$\|f(x_{k+1}) - f(x_k)\| < \epsilon_3.$$

Etude de la convergence

Voici un résultat de convergence de la méthode du gradient.

Théorème: Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction convexe et coercive de classe C^2 admettant un seul minimum. De plus, on suppose qu'il existe une constante $M > 0$ telle que

$$(H_f(x) v, v) \leq M \|v\|^2 \quad \forall x, v \in \mathbb{R}^n.$$

Si on choisit les ρ_k dans l'intervalle

$$0 < \beta_1 < \rho_k < \beta_2 < 2/M,$$

alors l'algorithme du gradient converge.

Exemple: on prend $f(x) = \frac{1}{2} (Ax, x) - (b, x) + c$ avec $A > 0$. Vérifions les hypothèses du théorème.

On sait que $H_f(x) = A$. Aussi f admet un seul minimum. De plus

$$(A v, v) \leq M \|v\|^2 \quad \forall v \in \mathbb{R}^n$$

où $M > 0$ est la plus grande valeur propre de A . Donc la méthode du gradient converge et s'écrit

- ▶ $x_0 \in \mathbb{R}^n$ donné,
- ▶ $x_{k+1} = x_k + \rho_k (b - A x_k)$.

Méthode du gradient à pas optimaux

Dans la pratique, la méthode du gradient à pas constant converge très lentement. Pour améliorer la convergence il est préférable de choisir les ρ_k de manière optimale

- ▶ $x_0 \in \mathbb{R}^n$ donné
- ▶ $x_{k+1} = x_k - \rho_k \nabla f(x_k)$
où $\rho_k > 0$ est choisi de sorte que

$$\min_{\rho} f(x_k - \rho \nabla f(x_k))$$

Note: la condition d'optimalité de ce problème de minimisation s'écrit

$$0 = \frac{d}{d\rho} f(x_k - \rho \nabla f(x_k))|_{\rho=\rho_k} = (\nabla f(x_{k+1}), \nabla f(x_k))$$

c'est-à-dire

$$\nabla f(x_{k+1}) \perp \nabla f(x_k)$$

Calcul du pas optimal

En général, il n'est pas facile de calculer exactement la valeur ρ qui minimise $\min_{\rho} f(x_k - \rho \nabla f(x_k))$. Par contre, pour la fonction

$$f(x) = \frac{1}{2} (Ax, x) - (b, x) + c$$

avec $A > 0$, la situation est plus facile.

Résidu: il est défini par $r = -\nabla f(x) = b - Ax$.

La méthode du gradient s'écrit:

$$\begin{aligned} x_{k+1} &= x_k - \rho_k \nabla f(x_k) = x_k + \rho_k r_k \\ \implies Ax_{k+1} - b &= Ax_k - b + \rho_k Ar_k \\ \implies r_{k+1} &= r_k - \rho_k Ar_k \end{aligned}$$

La condition d'optimalité pour ρ_k est

$$\nabla f(x_{k+1}) \perp \nabla f(x_k) \iff (r_{k+1}, r_k) = 0 \iff (r_k - \rho_k Ar_k, r_k) = 0$$

ce qui fournit la valeur $\rho_k = \frac{\|r_k\|^2}{(Ar_k, r_k)}$.

Méthode de Newton

Soit $f : U \rightarrow \mathbb{R}$ une fonction numérique de classe C^2 définie sur un ensemble ouvert $U \subset \mathbb{R}^n$.

Une condition nécessaire (et suffisante si f est convexe) pour qu'un point $\bar{x} \in U$ soit un minimum (local ou global) est

$$\nabla f(\bar{x}) = 0$$

Par conséquent, le point \bar{x} doit vérifier le système de n équations à n variables

$$F(x) = \nabla f(x) = 0.$$

Une approche très populaire pour résoudre $F(x) = 0$ est la méthode de Newton.

Méthode de Newton (suite)

Soit $F : U \rightarrow \mathbb{R}^n$ une application à valeur vectorielle de classe C^2 définie sur un ensemble ouvert $U \subset \mathbb{R}^n$. On notera $F = (F_1, F_2, \dots, F_n)$.

$$F(x) = 0 \iff \begin{cases} F_1(x_1, x_2, x_3, \dots, x_n) = 0 \\ F_2(x_1, x_2, x_3, \dots, x_n) = 0 \\ F_3(x_1, x_2, x_3, \dots, x_n) = 0 \\ \vdots = \vdots \\ F_n(x_1, x_2, x_3, \dots, x_n) = 0 \end{cases}$$

Résidu: le résidu est défini par $R(x) = -F(x)$.

Matrice jacobienne: la matrice jacobienne dF (aussi noté par J) est définie par

$$dF(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \cdots & \frac{\partial F_n}{\partial x_n} \end{pmatrix}$$

Méthode de Newton (suite)

La méthode de Newton est basée sur l'approximation linéaire de F autour d'un point x_0

$$F(x_0 + \Delta x) \approx F(x_0) + dF(x_0) \Delta x$$

On désire calculer la correction Δx de sorte que

$$0 = F(x_0 + \Delta x) \approx F(x_0) + dF(x_0) \Delta x = 0 \implies \Delta x = -[dF(x_0)]^{-1} F(x_0)$$

Etant donné l'approximation, il est nécessaire d'itérer ce qui conduit à l'algorithme suivant.

Méthode de Newton

1. Etant donné une approximation initiale x_0 ,
2. Résoudre le système linéaire: $dF(x_k) \Delta x = -F(x_k) = R(x_k)$,
3. Mettre à jour la solution: $x_{k+1} = x_k + \Delta x$,
4. Si $\frac{\|\Delta x\|}{\|x_{k+1}\|} < \epsilon_1$ et/ou $\|F(x_{k+1})\| < \epsilon_2$, la convergence est atteinte.

Calcul du minimum par la méthode de Newton

Soit $f : U \rightarrow \mathbb{R}$ une fonction numérique de classe C^2 définie sur un ensemble ouvert $U \subset \mathbb{R}^n$. Il s'agit de résoudre

$$f(\bar{x}) = \min_{x \in U} f(x)$$

On pose $F(x) = \nabla f(x)$. La matrice jacobienne est

$$dF(x) = \frac{\partial F_i}{\partial x_j} = \frac{\partial}{\partial x_j} \frac{\partial f}{\partial x_i} = \frac{\partial^2 f}{\partial x_i \partial x_j} = H_f(x)$$

qui est toujours symétrique.

L'algorithme de Newton s'écrit

$$\begin{cases} x_0 \in \mathbb{R}^n, \\ x_{k+1} = x_k - [H_f(x_k)]^{-1} \nabla f(x_k). \end{cases}$$

Remarque:

- ▶ Très sensible au choix du point initial.
- ▶ La convergence est en général quadratique.

Lien entre Newton et la méthode de descente

Faisons l'hypothèse que la matrice hessienne vérifie au point minimisant \bar{x} , la condition

$$H_f(\bar{x}) > 0 \implies H_f(x) > 0 \quad \forall x \text{ près de } \bar{x}.$$

En particulier, on aura pour les x_k près de \bar{x}

$$H_f(x_k) > 0 \implies [H_f(x_k)]^{-1} > 0.$$

On développe f par Taylor autour du point x_k

$$f(x_{k+1}) = f(x_k) + (\nabla f(x_k), x_{k+1} - x_k) + \frac{1}{2} (H(x_{k+1} - x_k), x_{k+1} - x_k).$$

Mais l'algorithme de Newton fournit la valeur $x_{k+1} = x_k - M_k^{-1} \nabla f(x_k)$
où $M_k^{-1} = [H_f(x_k)]^{-1} > 0$.

En négligeant le terme d'ordre 2, on obtient

$$\begin{aligned} f(x_{k+1}) &= f(x_k) - (M_k^{-1} \nabla f(x_k), \nabla f(x_k)) + \frac{1}{2} (H(x_{k+1} - x_k), x_{k+1} - x_k), \\ &\approx f(x_k) - (M_k^{-1} \nabla f(x_k), \nabla f(x_k)), \\ &\leq f(x_k) \quad \text{car } M_k^{-1} > 0. \end{aligned}$$

Méthode quasi-Newton

La méthode de Newton pose plusieurs difficultés.

Approximation de la matrice hessienne

En premier, il y a la nécessité de calculer la matrice hessienne. Pour certains types de problèmes, cela peut devenir problématique. Dans ce cas, on peut avoir recours à une approximation de la matrice hessienne. Pour cela, on utilise la formule de différences finies

$$H(x) e_j \approx \frac{\nabla f(x + h e_j) - \nabla f(x)}{h}$$

où e_j est le j ième vecteur de la base canonique de \mathbb{R}^n . $h > 0$ est une petite valeur de l'ordre de 10^{-8} . On notera que $H(x) e_j$ représente la j ième colonne de la matrice hessienne.

Méthode quasi-Newton (suite)

Méthode de quasi-Newton modifiée

Cette approche est basée sur l'observation que si M_k est une matrice symétrique et définie-positive, alors

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

est une direction de descente. En effet, on applique de nouveau Taylor

$$f(x_{k+1}) = f(x_k + d_k) = f(x_k) + (\nabla f(x_k), d_k) + \frac{1}{2} (H d_k, d_k).$$

En négligeant le terme d'ordre 2, on obtient

$$\begin{aligned} f(x_{k+1}) &= f(x_k) - (M_k^{-1} \nabla f(x_k), \nabla f(x_k)) + \frac{1}{2} (H d_k, d_k), \\ &\approx f(x_k) - (M_k^{-1} \nabla f(x_k), \nabla f(x_k)), \\ &\leq f(x_k) \quad \text{car } M_k^{-1} > 0. \end{aligned}$$

Méthode quasi-Newton (suite)

Le choix de newton comme direction de descente $d_k = -H(x_k)^{-1} \nabla f(x_k)$ ne fonctionne que si la matrice est $H(x_k) > 0$. Supposons qu'à une certaine itération, la matrice $H(x_k)$ n'est pas définie-positive. Elle admet les valeurs propres

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$$

Si $\lambda_1 < \epsilon$ pour une petite valeur $\epsilon > 0$, on peut translater la matrice $H(x_k)$ de sorte que les valeurs propres soient toujours plus grande que $\epsilon > 0$. Il suffit de poser

$$M_k = H(x_k) + (\epsilon - \lambda_1) I.$$

Méthode de Newton modifiée

1. Etant donné une approximation initiale x_0 ,
2. Calculer la première valeur propre λ_1 de $H(x_k)$. Si $\lambda_1 < \epsilon$, poser $M_k = H(x_k) + (\epsilon - \lambda_1) I$, sinon $M_k = H(x_k)$.
3. Calculer la direction de descente: $M_k d_k = -\nabla f(x_k)$,
4. Mettre à jour la solution: $x_{k+1} = x_k + d_k$.

Méthode quasi-Newton (suite)

Finalement, il est souvent nécessaire de garantir que $f(x_{k+1}) \leq f(x_k)$. Ceci peut être fait en modifiant la mise à jour de la solution

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

où $\rho_k > 0$ est choisi de sorte que

$$\min_{\rho} f(x_k + \rho d_k)$$

Dans la pratique, il est préférable de faire plutôt une recherche linéaire à partir de la valeur $\rho = 1$. Afin de garantir la convergence quadratique, il est important de terminer les itérations avec le choix de $\rho_k = 1$. Les algorithmes de recherche linéaire seront présentés plus loin.

Directions conjuguées

Pour l'algorithme du gradient à pas optimal, les directions de descentes $d_k = -\nabla f(x_k)$ vérifie la propriété

$$d_{k+1} \perp d_k.$$

Pour des courbes de niveaux très aplaties, la convergence de l'algorithme du gradient peut être très lente à cause du mouvement en zig-zag. Par conséquent, nous allons définir d'autres directions de descentes qui respectent mieux la géométrie du problème.

En premier, nous allons considérer le cas quadratique $\min f(x)$ avec $f(x) = \frac{1}{2} (Ax, x) - (b, x)$ où A est symétrique et définie-positive. C'est-à-dire, nous allons traiter de la résolution itérative du système linéaire

$$Ax = b.$$

Algorithme du gradient conjugué

Directions conjuguées Un ensemble de vecteurs $\{d_0, d_1, d_2, \dots, d_k\}$ est dit A-conjugué si

$$(A d_i, d_j) = 0 \quad \forall i \neq j$$

Autrement dit, les $\{d_i\}$ sont perpendiculaires entre eux par rapport au produit scalaire induit par la matrice A : $\langle u, v \rangle = (Au, v)$.

Le but de l'algorithme du gradient conjugué est de construire deux suites de vecteurs: les itérés $\{x_0, x_1, x_2, \dots, x_k\}$ et les directions de descentes $\{d_0, d_1, d_2, \dots, d_k\}$ qui vérifient les propriétés suivantes. On note le vecteur résidu: $r_k = b - Ax_k$.

- ▶ la suite des résidus $\{r_0, r_1, r_2, \dots, r_k\}$ forme un système orthogonal (au sens usuel)
- ▶ la suite des directions de descentes $\{d_0, d_1, d_2, \dots, d_k\}$ forme un système A-conjuguées.

Conséquence: l'algorithme du gradient conjugué converge en au plus n itérations, i.e. fournit la solution exacte de $Ax = b$.

Algorithme du gradient conjugué (suite)

L'objectif est de construire les itérés x_k et les directions conjuguées d_k .

Voici les étapes de l'algorithme du gradient conjugué

- ▶ Mise à jour de x_k :

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

- ▶ Mise à jour du résidu r_k :

$$r_{k+1} = r_k - \rho_k A d_k$$

- ▶ Mise à jour des directions conjuguées d_k :

$$d_{k+1} = r_{k+1} + \beta_k d_k$$

L'algorithme démarre avec le choix $d_0 = r_0 = b - Ax_0$ pour un certain point initial x_0 (par exemple $x_0 = 0$.)

Calcul des coefficients ρ_k

Il suffit d'exiger que $r_{k+1} \perp r_k$.

$$0 = (r_{k+1}, r_k) = (r_k - \rho_k A d_k, r_k) \implies \rho_k = \frac{(r_k, r_k)}{(A d_k, r_k)}$$

Or $r_k = d_k - \beta_{k-1} d_{k-1} \implies$

$(A d_k, r_k) = (A d_k, d_k - \beta_{k-1} d_{k-1}) = (A d_k, d_k)$ car les d_k sont A-conjugué.

Ceci fournit la valeur

$$\rho_k = \frac{\|r_k\|^2}{(A d_k, d_k)}$$

Calcul des coefficients β_k

Il suffit d'exiger que $d_{k+1} \perp_A d_k$.

$$0 = (d_{k+1}, A d_k) = (r_{k+1} + \beta_k d_k, A d_k) \implies \beta_k = -\frac{(r_{k+1}, A d_k)}{(A d_k, d_k)}$$

Mais $r_{k+1} = r_k - \rho_k A d_k \implies -A d_k = \frac{r_{k+1} - r_k}{\rho_k}$. Ceci fournit la valeur

$$\beta_k = -\frac{(r_{k+1}, A d_k)}{(A d_k, d_k)} = \frac{(r_{k+1}, \frac{r_{k+1} - r_k}{\rho_k})}{(A d_k, d_k)} = \frac{(r_{k+1}, r_{k+1})}{\rho_k (A d_k, d_k)}$$

car $(r_{k+1}, r_k) = 0$. De plus, $\rho_k = \frac{\|r_k\|^2}{(A d_k, d_k)}$

Ceci fournit la valeur finale

$$\beta_k = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}.$$

Méthode du gradient conjugué

Voici l'algorithme final:

- ▶ Evaluer le résidu initial $r_0 = b - Ax_0$ et poser $d_0 = r_0$.
- ▶ Pour $k = 0, \dots$, jusqu'à convergence, faire:
 - ▶ calculer $\rho_k = \frac{\|r_k\|^2}{(Ad_k, d_k)}$
 - ▶ $x_{k+1} = x_k + \rho_k d_k$
 - ▶ $r_{k+1} = r_k - \rho_k A d_k$
 - ▶ $\beta_k = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$
 - ▶ $d_{k+1} = r_{k+1} + \beta_k d_k$
- ▶ Fin de la boucle sur k

Remarques: Le critère d'arrêt est généralement de la forme $\|r_k\| < \epsilon$ ou encore $\frac{\|r_k\|}{\|r_0\|} < \epsilon$. Souvent, on prend $x_0 = 0$ comme point de départ.

Généralisation au cas non linéaire

Nous allons généralisé l'algorithme du gradient conjugué pour des problèmes $\min f(x)$ où f est non quadratique (cas non linéaire).

Les principales étapes de l'algorithme demeurent les mêmes.

On pose $r_k = -\nabla f(x_k)$.

- ▶ Mise à jour de x_k :

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

- ▶ Mise à jour du résidu r_k :

$$r_{k+1} = -\nabla f(x_{k+1})$$

- ▶ Mise à jour des directions conjuguées d_k :

$$d_{k+1} = r_{k+1} + \beta_k d_k$$

Calcul des coefficients: cas non linéaire

- ▶ Calcul des coefficients ρ_k : on calcule la valeur optimale qui réalise le minimum de

$$\min_{\rho} f(x_k + \rho d_k),$$

généralement fait par un algorithme de recherche linéaire.

- ▶ Calcul des coefficients β_k : deux choix sont possibles

- ▶ $\beta_k = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$ (Fletcher-Reeves)

- ▶ $\beta_k = \frac{(r_{k+1} - r_k, r_{k+1})}{\|r_k\|^2}$ (Polak-Ribière)

Souvent, ce dernier choix conduit à de meilleurs résultats.

Remarque: dans le cas non linéaire, on perd la propriété que l'algorithme converge en au plus n itérations.

Méthodes de recherche linéaire

Dans plusieurs algorithmes présentés jusqu'à présent, il est nécessaire de minimiser

$$\min_{\rho > 0} f(x + \rho d),$$

suivant une direction de descente d .

Posons $g(\rho) = f(x + \rho d)$. La condition d'optimalité du minimum est

$$g'(\rho) = 0$$

que l'on peut résoudre soit par la méthode de Newton ou celle de la sécante.

Recherche linéaire: méthode de Newton

$$\begin{cases} \rho_0 \text{ donné,} \\ \rho_{k+1} = \rho_k - \frac{g'(\rho_k)}{g''(\rho_k)} \end{cases}$$

où $g'(\rho) = (\nabla f(x + \rho d), d)$ et $g''(\rho) = (H_f(x + \rho d)d, d)$

Remarques:

- ▶ Converge rapidement
- ▶ On peut choisir $\rho_0 = 0$
- ▶ Si f convexe, $g''(\rho) > 0$
- ▶ Nécessite le calcul de la matrice hessienne
- ▶ Peut diverger si ρ_{opt} trop loin de $\rho_0 = 0$
- ▶ Peut converger vers une valeur $\rho < 0$

Recherche linéaire: méthode de la sécante

$$\begin{cases} \rho_0, \rho_1 \text{ donnés,} \\ \rho_{k+1} = \rho_k - g'(\rho_k) \frac{\rho_k - \rho_{k-1}}{g'(\rho_k) - g'(\rho_{k-1})} \end{cases}$$

où $g'(\rho) = (\nabla f(x + \rho d), d)$. On notera que

$$g''(\rho_k) \approx \frac{g'(\rho_k) - g'(\rho_{k-1})}{\rho_k - \rho_{k-1}}$$

Remarques:

- ▶ Converge rapidement
- ▶ Besoin de 2 valeurs: $\rho_0 = 0$ mais $\rho_1 = ?$
- ▶ N'exige pas le calcul de la matrice hessienne
- ▶ Peut diverger si ρ_{opt} trop loin de $\rho_0 = 0$
- ▶ Peut converger vers une valeur $\rho < 0$

Méthode de recherche linéaire approchés

En général, il n'est pas nécessaire de calculer précisément la valeur de ρ qui minimise

$$\min_{\rho > 0} f(x + \rho d).$$

On peut se contenter d'une valeur très approximative. En fait, il suffit de trouver une valeur ρ qui diminue de manière significative $f(x + \rho d) < f(x)$.

Posons $g(\rho) = f(x + \rho d)$. On choisit une valeur $0 < \beta < 1$ plus près de 0 (par exemple: $\beta = 0.1$).

On dira que la valeur ρ diminue de manière significative f si

$$g(\rho) \leq g(0) + \beta \rho g'(0) \iff f(x + \rho d) \leq f(x) + \beta \rho (\nabla f(x + \rho d), d)$$

Méthode d'Armijo

- ▶ ρ donné. ($\rho = 1$)
- ▶ On vérifie que $g(\rho) \leq g(0) + \beta \rho g'(0)$
- ▶ Sinon, on diminue le pas: $\rho \leftarrow \tau \rho$
où $0 < \tau < 1$ pas trop petit, par exemple $\tau = 0.5$.

Remarque: cette méthode connue sous le nom de Armijo Backtracking Line Search, est idéal pour les méthodes de minimisation de type Newton car la valeur $\rho = 1$ joue un rôle particulier.

Conditions de Wolfe

Pour les méthodes de type gradient (conjugué), la méthode d'Armijo n'est pas suffisante. La condition

$$g(\rho) \leq g(0) + \beta \rho g'(0)$$

va fournir une borne supérieure ρ_{max} pour laquelle la condition est satisfaite. Mais on a aucune borne inférieure. Donc on peut être très loin du minimum de $\min_{\rho > 0} f(x + \rho d)$. Pour cela, on introduit une seconde condition qui exige que la dérivée $g'(\rho)$ soit près de 0. Soit $0 < \beta_2 < 1$ pas trop petit, par exemple $\beta_2 = 0.9$.

Condition faible de Wolfe

$$\frac{g'(\rho)}{g'(0)} \leq \beta_2 \iff -g'(\rho) \leq \beta_2 (-g'(0))$$

car $g'(0) < 0$. Cette condition permet d'accepter toutes les valeurs où $g'(\rho) > 0$.

Condition forte de Wolfe: $|g'(\rho)| \leq \beta_2 |g'(0)|$

Cette condition plus forte, limite grandement les valeurs acceptables de ρ .

Méthode directe basée sur le nombre d'Or

- ▶ Connue sous le nom de Golden section minimization.
- ▶ Permet le calcul d'un minimum (local) d'une fonction continue $f : [a, b] \rightarrow \mathbb{R}$.
- ▶ N'utilise pas la dérivée.
- ▶ Peut s'appliquer aux fonctions non dérivables

Principe de la méthode: soient $x_1 < x_2$ dans l'intervalle $[a, b]$

- ▶ Si $f(x_1) \leq f(x_2)$, alors il y a un minimum (local) dans l'intervalle $[a, x_2]$.
- ▶ Si $f(x_1) \geq f(x_2)$, alors il y a un minimum (local) dans l'intervalle $[x_1, b]$.

Méthode directe (suite)

A partir du principe de base, il s'agit de construire une suite de sous-intervalles $[a_k, b_k]$ de l'intervalle initial $[a, b]$ contenant le minimum local \bar{x} . On pose $a_0 = a$ et $b_0 = b$.

- ▶ $a_k \leq \bar{x} \leq b_k \quad \forall k,$
- ▶ $b_k - a_k \rightarrow 0.$

Pour les deux évaluations $x_1 < x_2$ dans chacun des sous-intervalles $[a_k, b_k]$, on pourrait prendre ceux situés au tiers et au deux-tiers de l'intervalle.

Par exemple: pour $[a, b] = [0, 1]$, on aurait $x_1 = 1/3$ et $x_2 = 2/3$. Si $f(x_1) \leq f(x_2)$, les prochaines évaluations de l'intervalle $[0, 2/3]$ seraient $x_1 = 2/9$ et $x_2 = 4/9$. Aucun de ces points ne correspondent aux points x_1, x_2 de l'itération précédente.

Méthode directe (suite)

Par un meilleur choix des points x_1, x_2 , il est possible de réutiliser un de ces points à l'itération suivante. Pour cela, on pose $x_1 = (1 - \lambda)a_k + \lambda b_k$ et $x_2 = \lambda a_k + (1 - \lambda)b_k$ avec $0 < \lambda < 1$. Faisons l'hypothèse que l'intervalle $[a_k, x_2]$ est choisi. Il s'agira de déterminer λ de sorte que le nouveau x_2 est égal à l'ancien x_1

$$x_1 = \lambda a_k + (1 - \lambda)x_2$$

En substituant, on obtient

$$\lambda = (1 - \lambda)^2 \implies \lambda = \frac{3 - \sqrt{5}}{2} \approx 0.382 > 0$$

On observe que $1 - \lambda = \frac{-1 + \sqrt{5}}{2} \approx 0.618$ est le nombre d'Or.

Autrement dit, le découpage de l'intervalle $[a_k, b_k]$ doit se faire suivant les rapports 0.382 et 0.618.

Méthode directe (suite)

Il reste à préciser le critère d'arrêt.

Pour cela, il faut évaluer $b_{k+1} - a_{k+1}$ en fonction de $b_k - a_k$. De nouveau, faisons l'hypothèse que l'intervalle $[a_k, x_2]$ est choisi. On aura que $a_{k+1} = a_k$ et $b_{k+1} = x_2$. On obtient

$$b_{k+1} - a_{k+1} = [\lambda a_k + (1 - \lambda)b_k] - a_k = (1 - \lambda)(b_k - a_k),$$

c'est-à-dire

$$b_k - a_k = (1 - \lambda)^k (b - a) \approx (0.618)^k (b - a).$$

Si \bar{x} dénote le minimum cherché, on aura

$$\bar{x} - a_k \leq b_k - a_k = (1 - \lambda)^k (b - a) < \epsilon$$

Ceci permet de calculer le nombre d'itérations k nécessaires pour obtenir la précision désirée.