Chapitre 3

Problèmes quadratiques
Méthodes de Newton locales
Méthodes de descente de gradient
Méthodes quasi-Newton

I
III

Méthodes d'optimisation sans contrainte III

I) Problèmes quadratiques

A) Méthode directe

$\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$ $\quad Q \in \mathbb{R}^{n \times n}$ matrice symétrique définie positive

Considérons le problème d'optimisation suivant :
(Cf cours Chap 1 T26 et Chap 2 T4)

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$$

Unique minimum global ⇔ $= -\mathbf{b}$ ⇒ Résolution classique (méthode d'algèbre linéaire)

I
II
III

Configuration initiale

Matrice $Q \in \mathbb{R}^{n\times n}$ matrice symétrique définie positive
Vecteur $\mathbf{b} \in \mathbb{R}^n$

↓

Calculer factorisation de Cholesky de Q
$Q = LL^T$

↓

Calculer y* tq : $Ly = -b$

↓

Calculer x* tq : $L^T x = y$

↓

STOP x* solution exacte x*

remarque

Cette résolution typique nécessite que f soit quadratique au sens strict; ce qui n'est pas toujours le cas …

I) Problèmes quadratiques

B) Méthode des gradients conjugués

Hestenes et Stiefel (1952)

Directions conjuguées

$Q \in \mathbb{R}^{n\times n}$ matrice définie positive

Les vecteurs $\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_k$ non nuls de $\mathbb{R}^n$ sont dits Q-conjugués si

$$\mathbf{d}_i^T Q \mathbf{d}_j = 0 \quad \forall i, j \text{ tq } i \neq j$$

⇔ $\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_k$ linéairement indépendants

⇔ Nombre maximal de direction conjuguées = n

remarque

Si Q=I les directions conjuguées sont orthogonales

I) Problèmes quadratiques

B) Méthode des gradients conjugués

Principe de la méthode

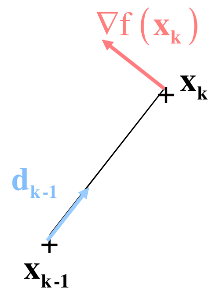
Utilisation de n directions Q-conjuguées $\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_n$ dans un algorithme basée sur la récurrence suivante :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad \text{avec} \quad \alpha_k = \arg\min_{\alpha} f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)$$

$\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_{n+1}$: les itérés générés par une méthode de directions conjuguées

Propriétés:

- pour tout k=1,...,n, le pas α_k est défini par $\alpha_k = -\dfrac{\mathbf{d}_k^T(Q\mathbf{x}_k + \mathbf{b})}{\mathbf{d}_k^T Q \mathbf{d}_k} = -\dfrac{\mathbf{d}_k^T \nabla f(\mathbf{x}_k)}{\mathbf{d}_k^T Q \mathbf{d}_k}$
- pour tout k=1,...,n, $\nabla f(\mathbf{x}_k)$ est orthogonal à $\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_{k-1}$
  $$\nabla f(\mathbf{x}_k)^T \mathbf{d}_i = 0 \quad \text{pour} \quad i = 1, \ldots, k-1$$
- $\nabla f(\mathbf{x}_{n+1}) = 0$
- soit k tq $\nabla f(\mathbf{x}_k) = 0$ ⇨ $\nabla f(\mathbf{x}_i) = 0$ pour $i = k, \ldots, n+1$



Identification du min global en **au plus n itérations**
Résolution successives dans des sous espaces vectoriels de dimension croissante engendrés par $\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_i \quad (i \leq n)$

Détermination des directions conjugués (Analogie avec l'orthogonalité)

Vecteurs linéairement indépendants

Procédure d'orthogonalisation **de Gram-Schmidt** (Cf littérature associée)

Vecteurs Q-conjugués $\mathbf{d}_i$ engendrant le même espace vectoriel

Choix des **vecteurs gradients** non nul $-\nabla f(\mathbf{x}_i)$

Méthode des **gradients conjugués**

$$\mathbf{d}_i = -\nabla f(\mathbf{x}_i) + \sum_{k=1}^{i-1} \frac{\mathbf{d}_k^T Q \nabla f(\mathbf{x}_i)}{\mathbf{d}_k^T Q \mathbf{d}_k} \mathbf{d}_k$$

remarque

Les vecteurs gradients sont non seulement indépendant mais … tous orthogonaux:

$$\nabla f(\mathbf{x}_i)^T \nabla f(\mathbf{x}_k) = 0 \text{ pour } k = 1, \dots, i-1$$

Expression de $\mathbf{d}_i$ en fonction de $\mathbf{d}_{i-1}$

$$\mathbf{d}_i = -\nabla f(\mathbf{x}_i) + \beta_i \mathbf{d}_{i-1} \quad \text{avec} \quad \beta_i = \frac{\nabla f(\mathbf{x}_i)^T \nabla f(\mathbf{x}_i)}{\nabla f(\mathbf{x}_{i-1})^T \nabla f(\mathbf{x}_{i-1})} = \frac{(Q\mathbf{x}_i + \mathbf{b})^T (Q\mathbf{x}_i + \mathbf{b})}{(Q\mathbf{x}_{i-1} + \mathbf{b})^T (Q\mathbf{x}_{i-1} + \mathbf{b})}$$

Algorithme

remarque

Le critère d'arrêt peut également être: k=n+1

Configuration initiale

Vecteur $\mathbf{b} \in \mathbb{R}^n$

Matrice $Q \in \mathbb{R}^{n\times n}$ symétrique définie positive

1ère approximation $\mathbf{x}_1 \in \mathbb{R}^n$ de la solution

Initialisation : k=0

$$\mathbf{d}_1 = -Q\mathbf{x}_1 - b = -\nabla f(\mathbf{x}_1)$$

Calculer le pas

$$\alpha_k = -\frac{\mathbf{d}_k^T (Q\mathbf{x}_k + b)}{\mathbf{d}_k^T Q \mathbf{d}_k}$$

$$\mathbf{x}_{k+1} = x_k + \alpha_k \mathbf{d}_k$$

Calculer β_{k+1}

$$\beta_{k+1} = \frac{(Q\mathbf{x}_{k+1} + \mathbf{b})^T (Q\mathbf{x}_{k+1} + \mathbf{b})}{(Q\mathbf{x}_k + \mathbf{b})^T (Q\mathbf{x}_k + \mathbf{b})}$$

Calculer $\mathbf{d}_{k+1}$ solution de

$$\mathbf{d}_{k+1} = -Q\mathbf{x}_{k+1} - b + \beta_{k+1} \mathbf{d}_k$$

k=k+1

$$\left\| \nabla f(x_k) \right\| = 0$$

OUI

NON

STOP $\mathbf{x}^* = x_k$

Approximation de la solution x*

II) Méthode de Newton locale

A) Méthode de Newton linéaire

Utilisation de l'algorithme de Newton *(cf chap 2 T12)* → Résoudre les conditions nécessaires d'optimalité → Trouver une approximation de la solution de l'équation $\nabla f(\mathbf{x}) = 0$

Algorithme

Configuration initiale

Gradient de f $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$

Hessien de f $\nabla^2 f : \mathbb{R}^n \to \mathbb{R}^{n \times n}$

1ère approximation $\mathbf{x}_0 \in \mathbb{R}^n$ de la solution

Précision demandée: $\varepsilon \in \mathbb{R}^n, \varepsilon > 0$

Initialisation : k=0

k=k+1

Calculer $\mathbf{d}_{k+1}$ solution de $\nabla^2 f(\mathbf{x}_k)\mathbf{d}_k = -\nabla f(x_k)$

$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$

$\|\nabla f(x_k)\| \leq \varepsilon$

NON

OUI

STOP $\mathbf{x} \simeq x_k$

Approximation de la solution x*

remarque

Ici, la fonction f du chap 2 est $\nabla f(\mathbf{x})$

I

II

III

II) Méthode de Newton locale

A) Méthode de Newton linéaire

Avantages / Inconvénients

- Méthode converge q-quadratiquement dans les conditions favorables *(cf chap 2 T10)*
- Peut diverger si le point de départ est trop éloigné de la solution
- **Méthode non définie si** $\nabla^2 f(x_k)$ **non inversible** → **Solution** : rendre inversible cette matrice
- Aucun mécanisme permettant de discerner les minima, des maxima et points selles…

I

II

III

Globalement, ne peut être utilisée telle quelle dans les problème d'optimisation

Cas favorables → Mise à profit de sa grande efficacité et rapidité de convergence lorsque le cas est adéquat ⟺ **Méthode de Newton quadratique (cf T9)**

Autres cas… → Utilisation de méthodes alternatives ⟺ **Méthode de Newton avec recherche linéaire (cf T20)**

II) Méthode de Newton locale

B) Méthode de Newton quadratique

Idée de base de la méthode de Newton → Remplacer une fonction non linéaire par un modèle plus simple

Par exemple linéaire *(cf Chap 2 T6)*

Modèle quadratique

Application de la formule de Taylor sur une fonction non linéaire

Fonction à plusieurs variables Soit une fonction $f:\mathbb{R}^n \to \mathbb{R}$ deux fois différentiable *(cf Chap 2 T6)*

Modèle quadratique de f en $\hat{\mathbf{x}}$ est noté $m_{\hat{x}}(\mathbf{x}):\mathbb{R}^n \to \mathbb{R}$ défini par :

$$m_{\hat{x}}(\mathbf{x}) = f(\hat{\mathbf{x}}) + (\mathbf{x}-\hat{\mathbf{x}})^T \nabla f(\hat{\mathbf{x}}) + \frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^T \nabla^2 f(\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})$$

En posant : $d = \mathbf{x}-\hat{\mathbf{x}}$

$$m_{\hat{x}}(\hat{\mathbf{x}}+\mathbf{d}) = f(\hat{\mathbf{x}}) + \mathbf{d}^T\nabla f(\hat{\mathbf{x}}) + \frac{1}{2}\mathbf{d}^T\nabla^2 f(\hat{\mathbf{x}})\mathbf{d}$$

remarque

On retrouve bien la définition d'une forme quadratique (cf chap1 T26) avec $Q = \nabla^2 f(\hat{\mathbf{x}}),\ b = \nabla f(\hat{\mathbf{x}})$ et $c = f(\hat{\mathbf{x}})$

II) Méthode de Newton locale

B) Méthode de Newton quadratique

Principe

Modélisation quadratique de la fonction objectif en x_k ⟺ Condition favorable pour l'efficacité de la méthode de Newton

$\nabla^2 f(x)$ doit forcément être définie positive pour pouvoir appliquer la méthode

Algorithme

Configuration initiale

Gradient de f $\nabla f:\mathbb{R}^n \to \mathbb{R}^n$

Hessien de f $\nabla^2 f:\mathbb{R}^n \to \mathbb{R}^{n\times n}$

1ère approximation $\mathbf{x_0} \in \mathbb{R}^n$ de la solution

Précision demandée: $\varepsilon \in \mathbb{R}^n, \varepsilon > 0$

Initialisation : k=0

k=k+1

Construire le modèle quadratique

$$m_{x_k}(\mathbf{x_k}+\mathbf{d}) = f(\mathbf{x_k}) + \mathbf{d}^T\nabla f(\mathbf{x_k}) + \frac{1}{2}\mathbf{d}^T\nabla^2 f(\mathbf{x_k})\mathbf{d}$$

Calculer d_k

$$d_k = \arg\min_d m_{x_k}(x_k + d)$$

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

$$\|\nabla f(x_k)\| \le \varepsilon$$

NON

OUI

STOP $\mathbf{x} \simeq x_k$

Approximation de la solution x*

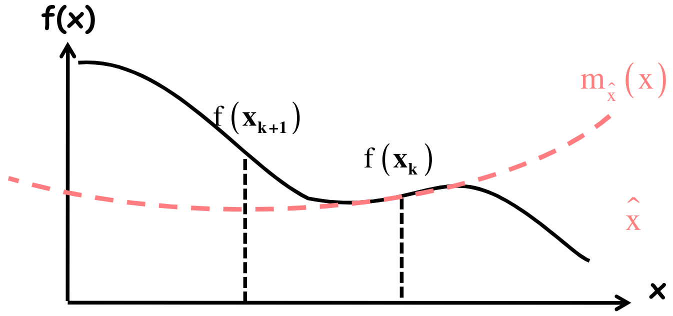
remarque

La minimisation de $m_{x_k}(x)$ *est assurée soit par méthode directe (cf T 2) soit par gradient conjugués (cf T 3)*

II) Méthode de Newton locale

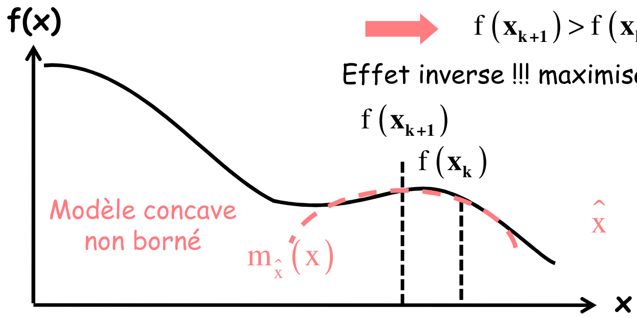
B) Méthode de Newton quadratique

Limitation



$f(\mathbf{x}_{k+1}) > f(\mathbf{x}_k)$

Nouveau point généré est moins bon que le précédent !!!



$f(\mathbf{x}_{k+1}) > f(\mathbf{x}_k)$

Effet inverse !!! maximisation

Méthode efficace si la fonction est par nature très proche d'une forme quadratique et convexe

I
II
III

II) Méthode de Newton locale

C) Points particuliers

Soit une fonction $f : \mathbb{R}^n \to \mathbb{R}$ deux fois différentiable et soit $\mathbf{x}_k \in \mathbb{R}^n$

Point de Newton

Point qui minimise le modèle quadratique de la fonction en $\mathbf{x}_k$

remarque *Point obtenu lors d'une itération de la méthode de Newton locale*

si $\nabla^2 f(\mathbf{x}_k)$ est définie positive, le point de Newton de f en $\mathbf{x}_k$ est le point :

$\mathbf{x}_N = \mathbf{x}_k + \mathbf{d}_N$ où $\mathbf{d}_N$ est la solution du système de Newton suivant: $\nabla^2 f(\boldsymbol{x}_k)\mathbf{d}_N = -\nabla f(\boldsymbol{x}_k)$

Point de Cauchy

Point qui minimise le modèle quadratique dans la direction de la plus forte pente

Le point de Cauchy de f en $\mathbf{x}_k$ est le point: $\mathbf{x}_C = \mathbf{x}_k - \alpha_C \nabla f(\mathbf{x}_k)$ où $\alpha_C = \arg\min_{\alpha \in \mathbb{R}_0^+} m_{x_k}(\boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k))$

Si f est convexe dans la direction du gradient, on a (*cf T9*):

$$\alpha_C = \frac{\nabla f(\mathbf{x}_k)^T \nabla f(\mathbf{x}_k)}{\nabla f(\mathbf{x}_k)^T \nabla^2 f(\mathbf{x}_k) \nabla f(\mathbf{x}_k)}$$

I
II
III

III) Méthode de la plus forte pente

A) Principe de la méthode

Idée de base des méthodes de descente

Suivre la direction de la plus forte pente donnée par la direction opposée au gradient *(Cf chap 1 T19)*

Configuration initiale

Gradient de f: $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$

1ère approximation $\mathbf{x_0} \in \mathbb{R}^n$ de la solution

Précision demandée: $\varepsilon \in \mathbb{R}^n, \varepsilon > 0$

Initialisation : k=0

$k = k+1$

Calculer $\mathbf{d}_k$

$$\mathbf{d}_k = -\nabla f(x_k)$$

Calculer α_k

$$\alpha_k = \arg\min_{\alpha \in \mathbb{R}_0^+} f(x_k + \alpha d_k)$$

$$\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k \mathbf{d_k}$$

$$\|\nabla f(x_k)\| \le \varepsilon$$

NON

OUI

STOP $\mathbf{x} \simeq x_k$

Approximation de la solution x^*

Méthode lente (même pour des cas simples)

I
II
III

III) Méthode de la plus forte pente

B) Préconditionnement

Réduction du conditionnement *(cf Chap 1 T27)* ⟺ Amélioration des performances de l'algorithme des plus fortes pentes

Modification de la direction de descente

Soit H_k une matrice symétrique définie positive telle que : $\mathrm{H}_k = \mathrm{L}_k \mathrm{L}_k^T$

définissant le **changement de variable** pour l'itération k: $\mathbf{x'} = \mathrm{L}_k^T \mathbf{x}$

remarque

Ici, H_k n'est pas forcement la matrice Hessienne mais une matrice permettant la réduction de κ_2

Ecriture de la méthode des plus fortes pentes

- sur les variables **x'**: $\mathbf{x'_{k+1}} = \mathbf{x'_k} - \alpha_k \nabla \tilde{f}(\mathbf{x'_k})$
- sur les variables **x**: $\mathbf{x_{k+1}} = \mathbf{x_k} - \alpha_k \mathrm{H}_k^{-1} \nabla f(\mathbf{x_k})$

on pose $\mathrm{D}_k = \mathrm{H}_k^{-1}$ ⟹ $\mathbf{x_{k+1}} = \mathbf{x_k} - \alpha_k \mathrm{D}_k \nabla f(\mathbf{x_k})$

Direction de descente à chaque itération:

$$\mathbf{d_k} = -\mathrm{D}_k \nabla f(\mathbf{x_k})$$

remarque

D_k permet de préconditionner différemment à chaque itération k

I
II
III

III) Méthode de la plus forte pente

B) Préconditionnement

Algorithme avec préconditionnement

Configuration initiale

Fonction f différentiable: $f:\mathbb{R}^n \to \mathbb{R}$

Gradient de f: $\nabla f:\mathbb{R}^n \to \mathbb{R}^n$

Famille de préconditionneurs D_k définie positive pour tout k

$1^{ère}$ approximation $\mathbf{x}_0 \in \mathbb{R}^n$ de la solution

Précision demandée: $\varepsilon \in \mathbb{R}^n, \varepsilon > 0$

Initialisation : k=0

$k = k+1$

Calculer d_k

$$\mathbf{d}_k = -D_k \nabla f(\mathbf{x}_k)$$

Calculer α_k

$$\alpha_k = \arg\min_{\alpha \in \mathbb{R}_0^+} f(x_k + \alpha d_k)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$$

$$\|\nabla f(x_k)\| \le \varepsilon$$

NON

OUI

STOP $\mathbf{x} \simeq x_k$

Approximation de la solution x*

Algorithme général

- Pas d'indication sur la manière de générer D_k
- Optimisation unidimensionnel sur α_k non trivial !!!

I II III

III) Méthode de la plus forte pente

B) Préconditionnement

Exemple

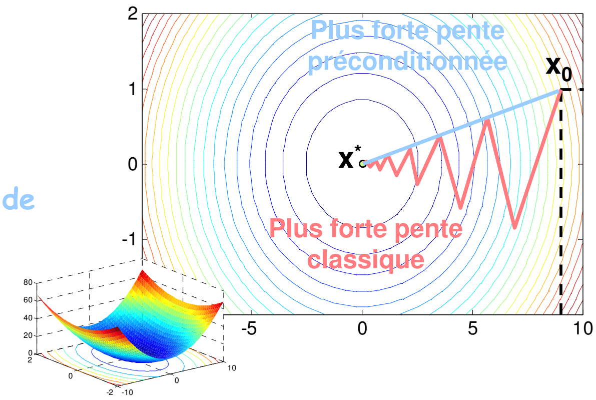
Minimum de la fonction $f(x) = \frac{1}{2}x_1 + \frac{9}{2}x_2$

Descente en « zig zag » caractéristique de la méthode

Lenteur désastreuse !!!

Préconditionnement par factorisation de Cholesky de la matrice Hessienne

Le minimum est atteind au bout …d'une seule itération !!!



Amélioration significative lorsque la fonction objectif est préconditionnée

I II III

III) Méthode de la plus forte pente

C) Choix du pas

compromis

α_k **petit**		α_k **élevé**
Assurer la validité de la direction de plus forte pente donnée par le gradient	$\Leftrightarrow$	Atteindre l'optimum le plus rapidement possible

I
II
III

→ α_k **pas trop grand**

Eviter la disproportion entre la longueur du pas et la diminution de la fonction objectif résultante

Notion de « **Diminution suffisante** » $\Leftrightarrow$ Diminution jugée suffisante si proportionnelle au pas

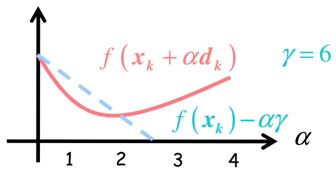
$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) \geq \alpha_k \gamma$$

$$\gamma = -\beta_1 \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k$$
$$0 < \beta_1 < 1$$

Choix de γ en fonction de la pente en $\mathbf{x_k}$ dans la direction $\mathbf{d_k}$

1ère condition de Wolfe (CW1)

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) - \alpha_k \gamma$$



Diminution jugée suffisante pour $\alpha < 2$

III) Méthode de la plus forte pente

C) Choix du pas

→ α_k **pas trop petit**

Eviter la dégénérescence des pas α_k vers 0

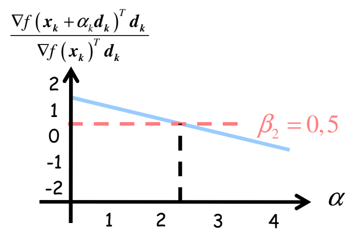
Notion de « **progrès suffisant** » $\Leftrightarrow$ Progrès suffisant si dérivée directionnelle au point $\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$ augmente par rapport à celle évaluée point $\boldsymbol{x}_k$

I
II
III

$$\nabla f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k)^T \boldsymbol{d}_k \geq \beta_2 \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k \quad \text{avec} \quad \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k < 0$$

$$\frac{\nabla f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k)^T \boldsymbol{d}_k}{\nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k} \leq \beta_2$$
$$0 < \beta_2 < 1$$

2nde condition de Wolfe (CW2)



Progrès jugé suffisante pour $\alpha \geq 2,4$

remarque

Ces conditions portent également de nom de condition d'Armijo-Goldstein

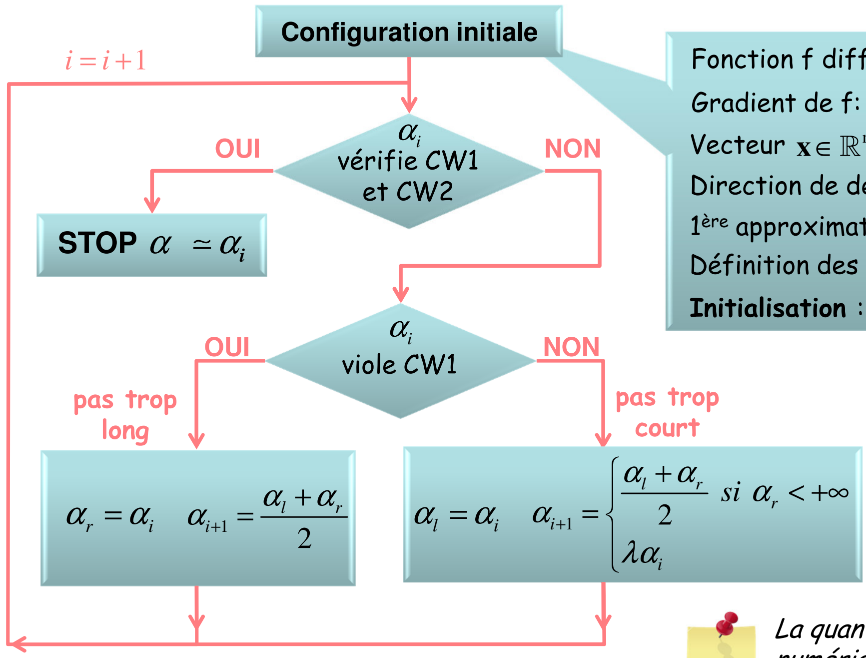
La validité des conditions de Wolfe impose que: $0 < \beta_1 < \beta_2 < 1$

III) Méthode de la plus forte pente

C) Choix du pas

Algorithme de Fletcher et Lemaréchal (FL)

Recherche linéaire du pas qui vérifie les conditions de Wolfe



Fonction f différentiable: $f : \mathbb{R}^n \to \mathbb{R}$
Gradient de f: $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$
Vecteur $\mathbf{x} \in \mathbb{R}^n$
Direction de descente **d** tq $\nabla f(x_k)^T d_k < 0$
1ère approximation $\alpha_0 > 0$ de la solution
Définition des paramètres : β_1, β_2 et $\lambda > 1$
Initialisation : i=0, $\alpha_l = 0$, $\alpha_r = +\infty$

Exemple $\lambda = 20$
$\alpha_0 = 10^{-3}, \beta_1 = 10^{-3}, \beta_2 = 0.99$

remarque
La quantité $+\infty$ est représenté numériquement par un nb suffisamment grand

IV) Méthode de Newton avec recherche linéaire

A) Principe général

Combinaison de 2 méthodes

Méthode de Newton
(cf T7)

Méthode de plus forte pente préconditionnée
(cf T15)

Itération k

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \nabla^2 f(\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k)$$

Itération k

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k D_k \nabla f(\mathbf{x}_k)$$

D_k défini positive

Souvent en échec

Converge rapidement

fonctionnelle

lente

IV) Méthode de Newton avec recherche linéaire

A) Principe général

Application de l'itération de la méthode des plus fortes pentes préconditionnée

- Si $\begin{cases} \nabla^2 f(\boldsymbol{x}_k) \text{ est défini positive} \\ \alpha_k = 1 \text{ vérifie les CW} \end{cases}$ → Itérations équivalentes

$$\boxed{D_k = \nabla^2 f(\boldsymbol{x}_k)^{-1}}$$

- Si $\alpha_k = 1$ ne vérifie les CW → Trouver un autre pas

Application de l'algorithme de Fletcher et Lemaréchal (FL)

- Si $\nabla^2 f(\boldsymbol{x}_k)$ n'est pas défini positive → Trouver un autre préconditionneur

I
II
III

Plusieurs méthodes…

Exemple

D_k diagonale avec pour élément (i,i) $\max\left(\varepsilon, \left(\frac{\partial^2 f}{\partial x_i^2}(\boldsymbol{x}_k)\right)^{-1}\right)$

avec $\varepsilon > 0$ pour garantir l'aspect définie positive

IV) Méthode de Newton avec recherche linéaire

B) Méthode complète

- **Principe**

Application de l'itération de la méthode des plus fortes pentes préconditionnée → **Choix du préconditionneur**

Modifier la matrice Hessienne afin de la rendre définie positive

$$\boxed{D_k = \left(\nabla^2 f(\boldsymbol{x}_k) + \tau I\right)^{-1}}$$

remarque: *Il existe toujours un τ tel que D_k soit définie positive*

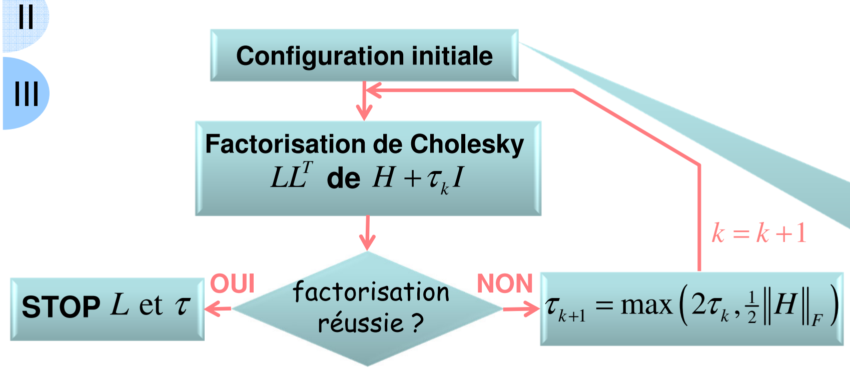
I
II
III

- **Recherche de τ par une méthode simple**

Permettant également la factorisation de Cholesky modifiée de $\nabla^2 f(\boldsymbol{x}_k) + \tau I$

Norme de Frobénius

$A \in \mathbb{R}^{m\times n} \quad \|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2}$



Matrice symétrique $H \in \mathbb{R}^{n\times n}$
Initialisation : k=0,
si $\min h_{ii} > 0$ alors $\tau_0 = 0$
sinon $\tau_0 = \|H\|_F$

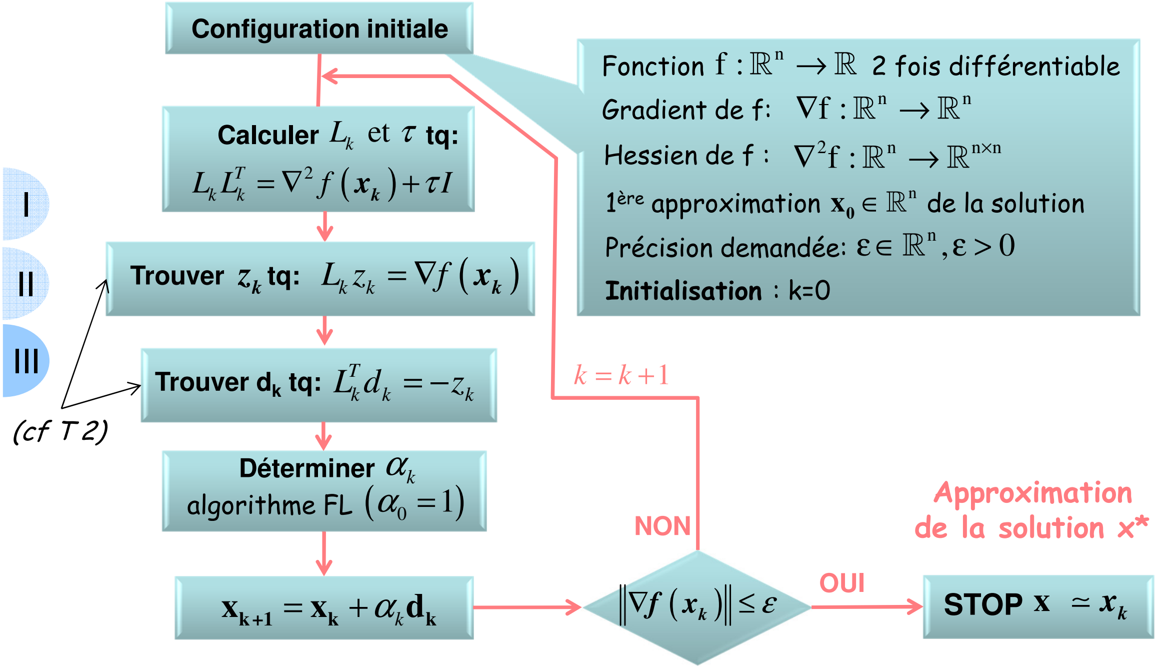
IV) Méthode de Newton avec recherche linéaire

B) Méthode complète

Algorithme

remarque

Lorsque cette méthode est appliquée sur un problème de moindres carrés, elle porte le nom de ***Levenberg Marquardt***



I
II
III

IV) Méthode de Newton avec recherche linéaire

C) Convergence

Convergence de la méthode de Newton

Dépendance du point de départ

Plus la fonction est non linéaire plus x_0 doit être proche de la solution *(cf chapitre 2 T11)*

Intérêt de la méthode de Newton avec recherche linéaire

Convergence quelque soit le point de départ

I
II
III

Convergence globale

Soit un algorithme itératif qui génère une suite $(x_k)_k$ dans $\mathbb{R}^n$, afin de résoudre le problème de minimisation sans contrainte :

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$$

avec $f : \mathbb{R}^n \to \mathbb{R}$ une fonction continûment différentiable.

L'algorithme est dit globalement convergent si:

$$\lim_{k \to \infty} \|\nabla f(\mathbf{x_k})\| = 0 \quad \forall x_0 \in \mathbb{R}^n$$

IV) Méthode de Newton avec recherche linéaire

C) Convergence

remarque *Attention à ne pas confondre « convergence globale » avec « minimum global »*

Convergence globale
Pour un algorithme utilisant méthode de Newton avec recherche linéaire

Directions de descente de plus en plus orthogonales à la direction du gradient
(non démontré ici)

I
II
III

Pour garantir : $\lim\limits_{k} \nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$

$$\cos\theta_k = \frac{\nabla f(\mathbf{x}_k)^T \mathbf{d}_k}{\|\nabla f(\mathbf{x}_k)\| \|\mathbf{d}_k\|}$$

ne peut pas tendre vers 0 !!!

$\cos\theta_k$ est borné par une constante strictement positive $\Leftrightarrow$ Les directions d_k ne dégénèrent pas en devenant asymptotiquement orthogonales au gradient

On dit que la suite $(\mathbf{d}_k)_k$ doit être en **relation gradient** avec $(\mathbf{x}_k)_k$ pour assurer **la convergence globale**

V) Méthodes de Quasi-Newton

A) Principe

Même principe que pour la résolution numérique des équations non linéaires
(cf chapitre 2 T16)

Méthode de Newton
(locale et avec recherche linéaire)

Calcul de matrice Hessienne

En pratique:

- Calcul analytique et implémentation fastidieux
  - Sujet à erreurs
- Consomme beaucoup de temps de calcul
  - Pénalise l'efficacité de la méthode

I
II
III

Inspiration directe des méthodes sécantes appliquées au modèle quadratique

$H_k \mathbf{d}_{k-1} = \mathbf{y}_{k-1}$ avec

$\mathbf{y}_{k-1} = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})$

Mise à jour de Broyden de la matrice Hessienne
(cf chapitre 2 T20)

$\Leftrightarrow$

Problème !!!

$\nabla^2 f(\mathbf{x}_k)$ pas forcément symétrique ni définie positive

Condition nécessaire pour calculer d_k
(cf T10)

V) Méthodes de Quasi-Newton

B) Méthode BFGS

Modification de la mise à jour de Broyden

BUT: obtenir $H_k = A_k A_k^T$ **symétrique positive** vérifiant l'équation sécante $\Rightarrow A_k A_k^T \mathbf{d}_{k\text{-}1} = \mathbf{y}_{k\text{-}1}$

remarque *L'équation sécante définie positive possède une solution ssi* $\mathbf{d}_{k\text{-}1}^T \mathbf{y}_{k\text{-}1} > 0$

I
II
III

Soit une fonction $f : \mathbb{R}^n \to \mathbb{R}$ différentiable et 2 itérés $\mathbf{x}_k$ et $\mathbf{x}_{k\text{-}1}$ tq $\mathbf{d}_{k\text{-}1}^T \mathbf{y}_{k\text{-}1} > 0$

avec $\mathbf{d}_{k\text{-}1} = \mathbf{x}_k - \mathbf{x}_{k\text{-}1}$ et $\mathbf{y}_{k\text{-}1} = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k\text{-}1})$

Soit une matrice symétrique définie positive $H_{k\text{-}1} \in \mathbb{R}^{n\times n}$

La mise a jour de C.G. Broyden, D. Goldfarb et D.F. Shanno (BFGS) est définie par:
$$H_k = H_{k\text{-}1} + \frac{y_{k\text{-}1} y_{k-1}^T}{y_{k-1}^T d_{k\text{-}1}} - \frac{H_{k\text{-}1} d_{k\text{-}1} d_{k-1}^T H_{k\text{-}1}}{d_{k-1}^T H_{k\text{-}1} d_{k\text{-}1}}$$

V) Méthodes de Quasi-Newton

B) Méthode BFGS

Algorithme

Reprendre algorithme de Newton avec recherche linéaire *(Cf T23)*

I
II
III

Remplacement de la matrice Hessienne par son approximation BFGS $\nabla^2 f(\mathbf{x}_k) \Leftrightarrow H_k$

H_k forcément définie positive $\Leftrightarrow$ Simplification de l'algorithme pour la recherche de d_k $H_k \mathbf{d}_k = -\nabla f(\mathbf{x}_k)$

Résolution du système sur d_k fastidieux et couteux en calcul $\Leftrightarrow$ Calcul analytique de H_k^{-1}

$$\mathbf{d}_k = -H_k^{-1} \nabla f(\mathbf{x}_k) \quad \text{avec} \quad H_k^{-1} = \left(I - \frac{d_{k\text{-}1} y_{k-1}^T}{d_{k-1}^T y_{k\text{-}1}} \right) H_{k\text{-}1}^{-1} \left(I - \frac{y_{k\text{-}1} d_{k-1}^T}{d_{k-1}^T y_{k\text{-}1}} \right) + \frac{d_{k\text{-}1} d_{k-1}^T}{d_{k-1}^T y_{k\text{-}1}}$$

Démonstration en appliquant la formule de Sherman-Morrision-Woodbury à H_k

V) Méthodes de Quasi-Newton

B) Méthode BFGS

Configuration initiale

Fonction $f : \mathbb{R}^n \to \mathbb{R}$ continûment différentiable

Gradient de f: $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$

1ère approximation $\mathbf{x_0} \in \mathbb{R}^n$ de la solution

1ère approximation $H_0^{-1} \in \mathbb{R}^{n \times n}$ de l'inverse du Hessien symétrique définie positive $H_0^{-1} = I$

Précision demandée: $\varepsilon \in \mathbb{R}^n, \varepsilon > 0$

Initialisation : k=0

Calculer d_k

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

Déterminer α_k

algorithme FL $(\alpha_0 = 1)$

$\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k \mathbf{d_k}$

$k = k + 1$

Mettre à jour H_k^{-1}

$y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$

$d_{k-1} = \alpha_{k-1} d_{k-1} = x_k - x_{k-1}$

$$H_k^{-1} = \left(I - \frac{d_{k-1} y_{k-1}^T}{d_{k-1}^T y_{k-1}} \right) H_{k-1}^{-1} \left(I - \frac{y_{k-1} d_{k-1}^T}{d_{k-1}^T y_{k-1}} \right) + \frac{d_{k-1} d_{k-1}^T}{d_{k-1}^T y_{k-1}}$$

$\|\nabla f(x_k)\| \leq \varepsilon$

NON

OUI

STOP $\mathbf{x} \simeq x_k$

Approximation de la solution x*

V) Méthodes de Quasi-Newton

C) Autres méthodes …

Autres formules de mise à jour pour la matrice Hessienne → Autres méthodes de Quasi-Newton

- BFGS = mise à jour de rang 2 ⇔ $(H_k - H_{k-1})$ Matrice de rang 2

  → Mise à jour symétrique de rang 1 (SR1)

  $H_k = H_{k-1} \pm vv^T$ où $v \in \mathbb{R}^n$

  remarque: *SR1 ne génère pas forcément une matrice définie positive même si H_{k-1} l'est!!!*

Rang d'une matrice A

rang(A) est la dim du sous espace $\mathrm{Im}(A) = \{ \boldsymbol{y} \in \mathbb{R}^m \mid \exists \boldsymbol{x} \in \mathbb{R}^n \ tq\ \boldsymbol{y} = A\boldsymbol{x} \}$

$A \in \mathbb{R}^{m \times n}$ Nb de valeurs singulières non nulles de A

- Mise à jour de Davidon étudiée par Fletcher et Powell (DFP)
- Etc…