

Chapitre 5.

Méthodes de descente en optimisation différentiable sans
contrainte

Généralités sur les algorithmes de descente

Principe général

Un algorithme de descente va chercher à générer une suite $(x_k)_{k \in \mathbb{N}}$ d'itérés vérifiant:

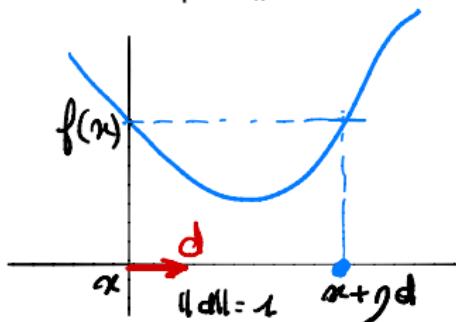
$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + s_k d_k$$

vérifiant:

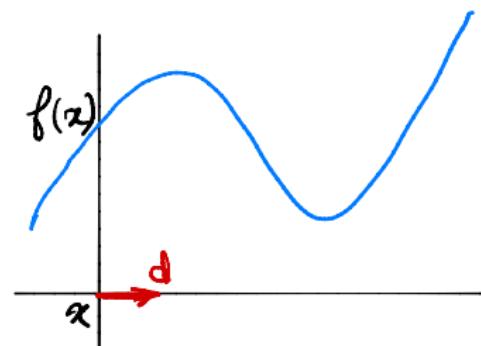
$$\forall k \in \mathbb{N}, \quad f(x_{k+1}) \leq f(x_k).$$

Un tel algorithme est complètement déterminé par:

- le choix de la direction $d_k \in \mathbb{R}^n$
- le choix du pas $s_k > 0$.



d est une direction de descente de f en x



d ne l'est pas ici

Généralités sur les algorithmes de descente

Choix de la direction d_k

Soit $x \in \mathbb{R}^n$ l'itéré courant. On s'intéresse aux directions qui vont permettre de faire diminuer la valeur de f i.e. aux directions $d \in \mathbb{R}^n$ telles que:

$$\exists \eta > 0, \forall t \in]0, \eta], f(x + td) < f(x).$$

i.e. telles que $\varphi : t \mapsto f(x + td)$ soit décroissante en $t = 0$ i.e.:

$$\varphi'(0) \leq 0$$

ce qui s'écrit aussi:

Le vecteur $d \in \mathbb{R}^n$ est une direction de descente pour f au point x si $\langle \nabla f(x), d \rangle < 0$.

Exercices:

- ① Soit $f : x \mapsto \frac{1}{2}x_1^2 + 2x_2^2$. Les directions suivantes sont-elles des directions de descente de f au point $(1, 1)$?

$$d_1 = -\nabla f(1, 1), \quad d_2 = (1, -3), \quad d_3 = (+1, +1).$$

- ② Montrer que la direction $d = -\nabla f(x)$ est toujours une direction de descente de f au point x .

- ③ A quelle condition (suffisante), la direction $d = -H_f(x)^{-1}\nabla f(x)$ est-elle une direction de descente de f au point x ?

$$CS : H_f(x) \succ 0$$

La recherche linéaire ou comment choisir un pas de descente

Soit $x \in \mathbb{R}^n$ un point non critique i.e.:

$$\nabla f(x) \neq 0,$$

et d une direction de descente de f en x :

$$\langle \nabla f(x), d \rangle < 0.$$

Etape de recherche linéaire

On cherche un pas $s > 0$ tel que:

$$f(x + sd) < f(x).$$

Reformulation

Soit: $\varphi : s \in \mathbb{R} \mapsto f(x + sd)$. La fonction φ est dérivable sur \mathbb{R} , de dérivée:

$$\varphi'(s) = \langle \nabla f(x + sd), d \rangle \text{ et: } \varphi'(0) = \langle \nabla f(x), d \rangle < 0.$$

On cherche un pas $s > 0$ tel que:

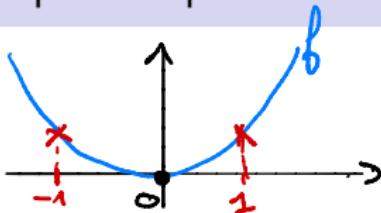
$$\varphi(s) < \varphi(0).$$

Stratégies de recherche linéaire sur un exemple simple

Soit:

$$f'(x) = \alpha$$

$$f : x \in \mathbb{R} \mapsto \frac{1}{2}x^2$$



- Direction de descente normalisée: $d_k = -f'(x_k)/|f'(x_k)| = -\text{sgn}(x_k)$.
(direction du gradient normalisé)

Pas fixe: $s_k = s > 0$.

$$x_1 = x_0 + s d_0 = 1 + 2 \times (-1) = -1$$

$$x_2 = x_1 + s d_1 = -1 + 2 \times (+1) = 1$$

- $x_0 = 1, s = 2$.

- Calcul des itérés:

$$\forall k, \quad x_{2k} = 1 \quad \text{et} \quad x_{2k+1} = -1.$$

- L'algorithme ne converge pas !

Pas optimal: s_k solution de: $\min_{t>0} f(x_k - t \text{sgn}(x_k))$.

- $\varphi_k(t) = f(x_k - t \text{sgn}(x_k)) = \frac{1}{2}(|x_k| - t)^2$, d'où: $s_k = |x_k|$.

- L'algorithme converge en une itération !

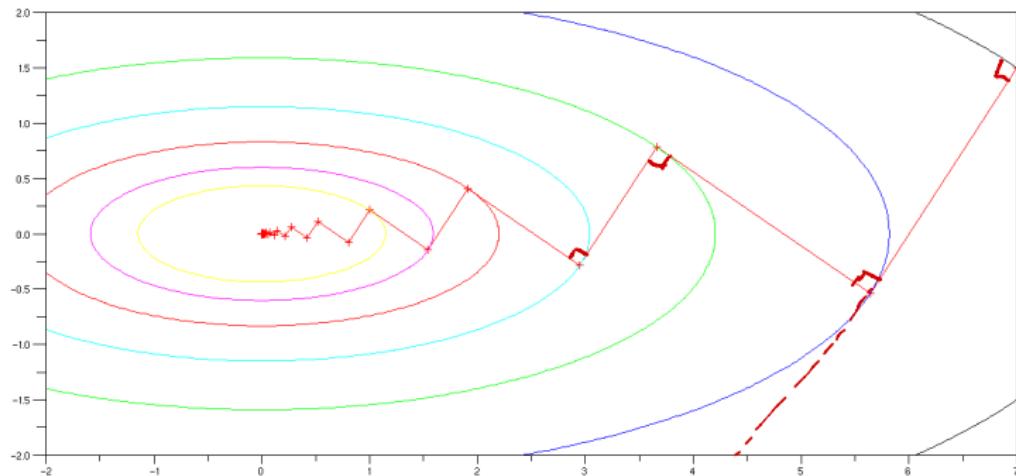
$$x_0, x_1 = x_0 - |x_0| \times \text{sgn}(x_0) = x_0 - x_0 = 0$$

= 0 près minimum de f

Rappels sur l'algorithme de gradient à pas optimal

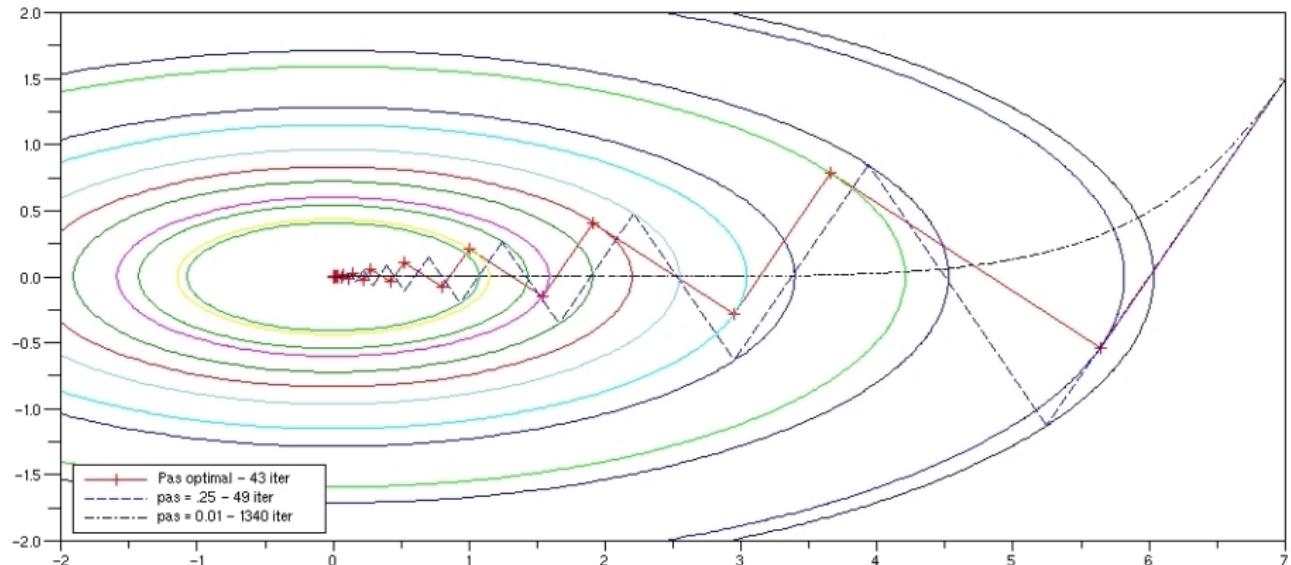
Soit:

$$\min_{(x,y) \in \mathbb{R}^2} f(x,y) = \frac{1}{2}x^2 + \frac{7}{2}y^2, \quad d_k = -\nabla f(X_k) = \begin{pmatrix} -x_k \\ -7y_k \end{pmatrix}.$$



- Deux directions de descente successives sont orthogonales.
- Convergence de l'algorithme de gradient à pas optimal en 43 itérations.

Pas fixe vs pas optimal



pas	0.325	0.25	0.125	0.05	0.01
Nb d'itérations	DV	49	101	263	1340

Condition d'Armijo ou comment éviter les pas trop grands

Retour à l'exemple:

$$f : x \in \mathbb{R} \mapsto \frac{1}{2}x^2.$$

On choisit: $x_0 = 2$.

- Direction de descente normalisée: $d_k = -f'(x_k)/|f'(x_k)| = -\text{sgn}(x_k)$.
- Pas de descente: $s_k = 2 + \frac{3}{2^{k+1}} \xrightarrow{k \rightarrow \infty} 2$, $\forall k, x_k > 2$.

$$x_{k+1} = x_k - s_k \text{sgn}(x_k)$$

$$x_0 = 2, \quad x_1 = 2 - s_0 = 2 - 2 - \frac{3}{2^1} = -\frac{3}{2} < 0$$

$$x_2 = -\frac{3}{2} + s_1 = -\frac{3}{2} + 2 + \frac{3}{2^2} = \frac{5}{2^2} > 0$$

$$x_3 = \frac{5}{4} - \left(2 + \frac{3}{2^3}\right) = \frac{5-8}{4} - \frac{3}{2^3}$$

$$= -\frac{3}{4} - \frac{3}{8} = -\frac{9}{2^3}$$

$$x_k = (-1)^k \left(1 + \frac{1}{2^k}\right) \quad (\text{à vérifier par récurrence})$$

Condition d'Armijo ou comment éviter les pas trop grands

Retour à l'exemple:

$$f : x \in \mathbb{R} \mapsto \frac{1}{2}x^2.$$

On choisit: $x_0 = 2$.

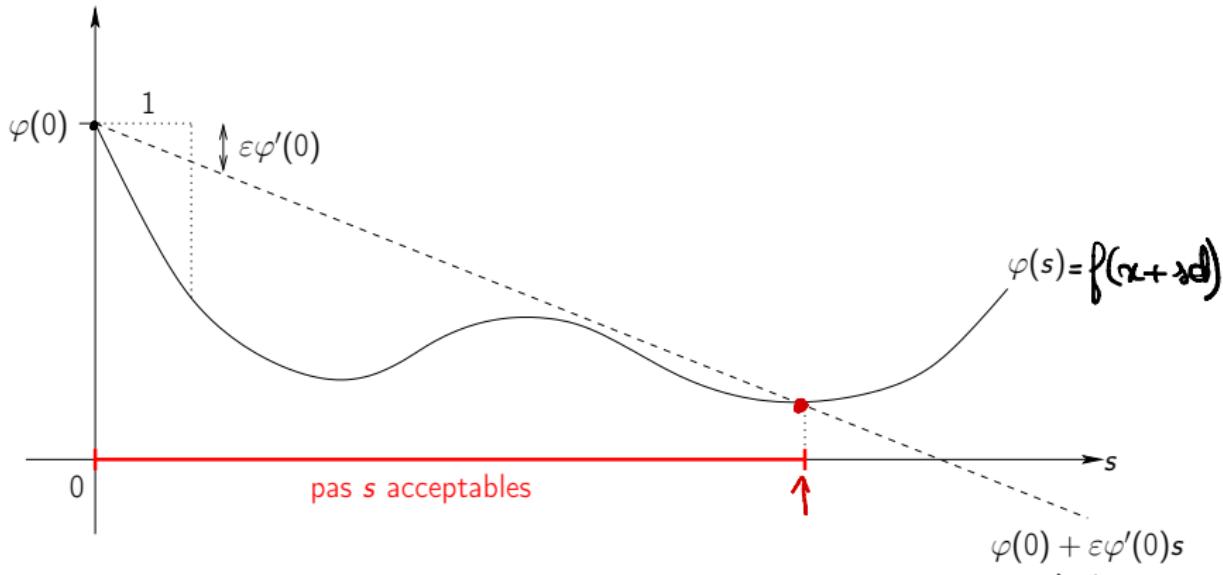
- Direction de descente normalisée: $d_k = -f'(x_k)/|f'(x_k)| = -\text{sgn}(x_k)$.
- Pas de descente: $s_k = 2 + \frac{3}{2^{k+1}}$.

$$x_k = (-1)^k \left(1 + \frac{1}{2^k} \right)$$

$$|x_k - x^*| = |x_k| = 1 + \frac{1}{2^k} \xrightarrow{k \rightarrow +\infty} 1$$

L'algorithme ne converge pas : il reste bloqué en 1 et -1.

Condition d'Armijo ou comment éviter les pas trop grands



Condition de Armijo

$$f(x + sd) \leq f(x) + s\varepsilon \langle \nabla f(x), d \rangle, \quad 0 < \varepsilon < 1$$

ou encore:

$$\varphi(s) \leq \varphi(0) + s\varepsilon\varphi'(0)$$

≤ 0 car d direction de descente de f

en x .

Il existe tout un intervalle de pas satisfaisant la condition d'Armijo.

$$\left| \begin{array}{l} x_0 \in \mathbb{R} \\ x_{k+1} = x_k + s_k d_k \end{array} \right.$$

avec

de direction de recherche de f en x_k
 $s_k > 0$ pas calculé par recherche linéaire
 (CLS)

LS \rightarrow pas fine: $s_k = s$, $\forall k$
 pas optimal: x_k sol de $\min_{s>0} f(x_k + s d_k)$

pas de Wolfe $\left\{ \begin{array}{l} \text{Armijo} \quad (s_k \text{ pas trop grand}) \\ 2^{\text{e}} \text{ crit.} \quad (s_k \text{ pas trop petit}) \end{array} \right.$

Enjeux: prouver la cr globale des algos (quelque soit x_0)

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$$

et si on peut la cr des itérés vers un pt critique de f .

Conditions de Wolfe ou comment éviter les pas trop petits

On part de: $x_0 = 2$.

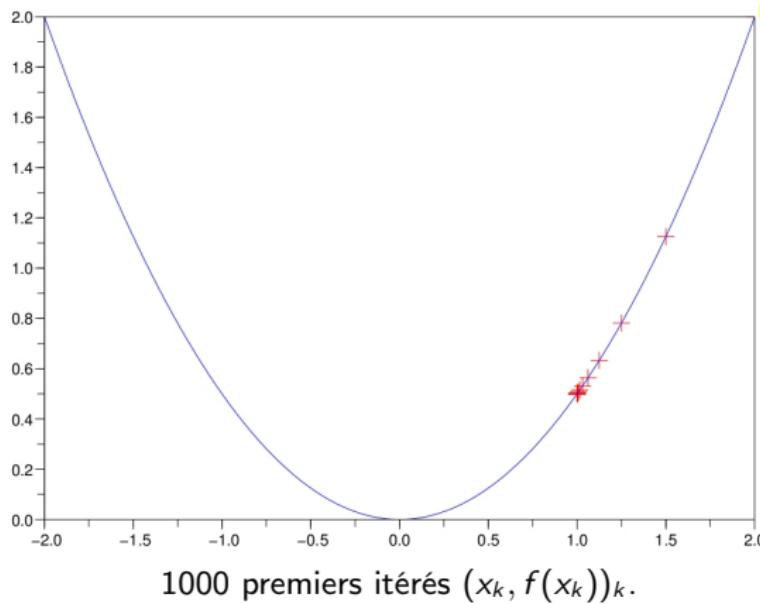
- Direction de descente normalisée: $d_k = -f'(x_k)/|f'(x_k)| = -\text{sgn}(x_k)$.

- Pas de descente: $s_k = \underbrace{\frac{1}{2^{k+1}}}_{\text{tend vers } 0} \xrightarrow{k \rightarrow +\infty} 0$.

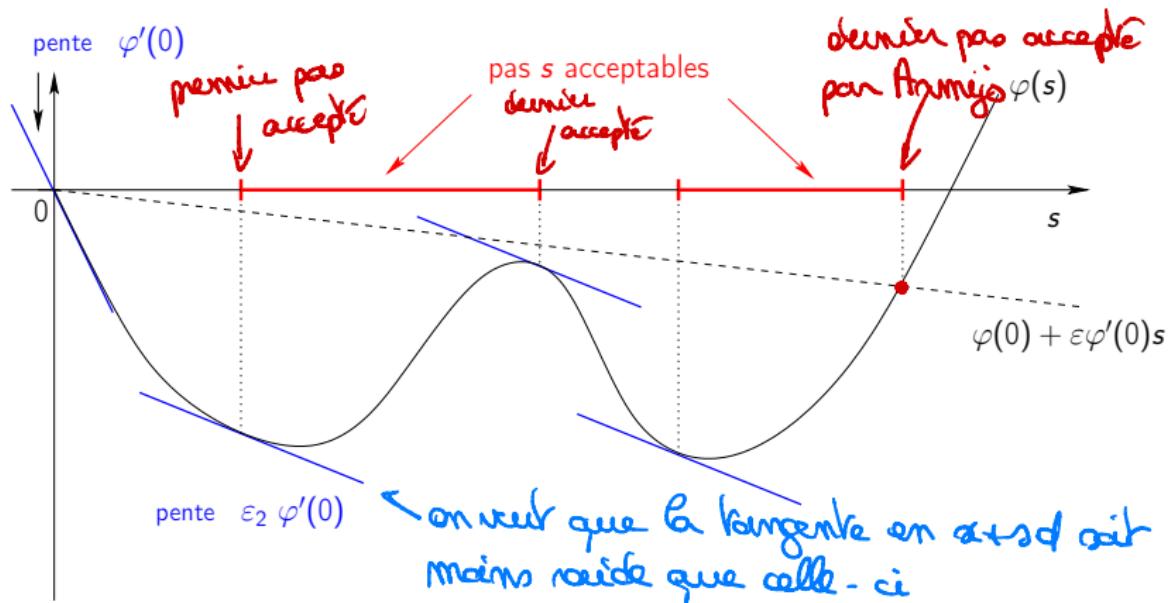
$$x_{k+1} = x_k - \frac{1}{2^{k+1}} \text{sgn}(x_k)$$

$$x_k = 1 + \frac{1}{2^k}$$

\downarrow
 $k \rightarrow +\infty$



Conditions de Wolfe ou comment éviter les pas trop petits



Conditions de Wolfe.

$$\begin{cases} f(x + sd) \leq f(x) + \varepsilon_1 s \langle \nabla f(x), d \rangle & (\text{Armijo}) \\ \nabla f(x + sd)^T d \geq \varepsilon_2 \langle \nabla f(x), d \rangle \end{cases}$$

$\leftarrow 0$

avec:

$$0 < \varepsilon_1 < \varepsilon_2 < 1.$$

$$\varepsilon_1 = 10^{-4}, \varepsilon_2 = 0.99$$

Algorithme de recherche linéaire de Wolfe

Fletcher 1980 - Lemaréchal 1981

➊ $k := 0; s_- = 0; s_+ = +\infty;$

➋ Tant que s_k ne vérifie pas les deux conditions de Wolfe,

➌ Si s_k ne vérifie pas la cd d'Armijo, alors le pas est trop long et:

$$s_+ = s_k \quad \text{et} \quad s_{k+1} = \frac{s_- + s_+}{2}.$$

➍ Si s_k vérifie la cd d'Armijo mais pas la seconde de Wolfe, alors le pas est trop court et:

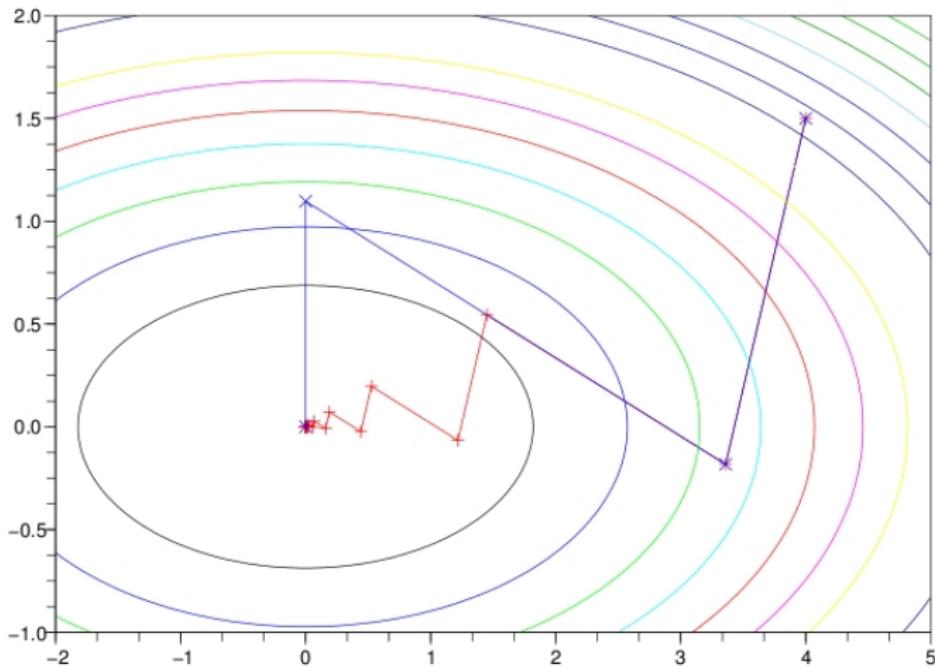
$$s_- = s_k \quad \text{et} \quad s_{k+1} = \begin{cases} \frac{s_- + s_+}{2} & \text{si } s_+ < +\infty \\ 2s_k & \text{sinon.} \end{cases}$$

➎ $k := k + 1;$

➏ Retourner s_k .

Descentes de gradient: pas optimal vs pas de Wolfe

Minimiser $\frac{1}{2}x^2 + \frac{7}{2}y^2$, $(x_0, y_0) = (4, 1.5)$.



Plus profonde descente: 23 itérations / Gradient avec pas de Wolfe: 3 itérations.

Cv globale des algorithmes de descente avec pas de Wolfe

Soit $(x_k)_{k \in \mathbb{N}}$ une suite d'itérés générés par un algorithme de descente avec pas de Wolfe i.e.:

$$x_{k+1} = x_k + s_k d_k$$

avec:

- d_k direction de descente de f en x_k : $\langle \nabla f(x_k), d_k \rangle < 0$.
- s_k pas vérifiant les deux conditions de Wolfe

On veut démontrer la convergence globale de l'algorithme i.e. montrer que quel que soit le point initial x_0 , on a:

$$\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

Ce résultat ne donne pas la cv des itérés mais signifie que tout point d'accumulation de la suite $(x_k)_k$ est un point critique de f .

= valeur d'adhérence.

pt d'accumulation = limite d'une sous-suite qui cv
ex: Le pts d'accumulation de $(-1)^k$ sont -1 et 1

Si une sous-suite $(x_{\varphi(k)})_k$ cv vers \bar{x} alors $\nabla f(\bar{x}) = 0$.

Cv globale des algorithmes de descente avec pas de Wolfe

Principe général de démonstration

- Montrer une inégalité du type:

$$f(x_k) - f(x_{k+1}) \geq c \|\nabla f(x_k)\|^2$$

où $c > 0$ est une constante réelle.

- En sommant ces inégalités pour k variant de 0 à $N - 1$, on obtient:

$$\forall N \in \mathbb{N}, f(x_0) - f(x_N) \geq c \underbrace{\sum_{n=0}^{N-1} \|\nabla f(x_k)\|^2}_{\text{somme partielle d'une série à termes positifs}}.$$

- Si f est bornée inférieurement, alors nécessairement $f(x_0) - f(x_N)$ est majorée et donc la somme partielle est majorée, et donc la série $\sum \|\nabla f(x_k)\|^2$ converge, ce qui implique:

$$\lim_{k \rightarrow +\infty} \nabla f(x_k) = 0.$$

Si f admet un minimum global alors $\forall x, f(x) \geq f(x^*) = f^*$

$$\forall N, \sum_{n=0}^{N-1} \|\nabla f(x_k)\|^2 \leq f(x_0) - f(x_N) \leq f(x_0) - f^*$$

Cv globale des algorithmes de descente avec pas de Wolfe

Théorème de Zoutendijk

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ supposée différentiable, de gradient Lipschitz et bornée inférieurement.
Soit A un algorithme générant des itérés définis par:

$$x_{k+1} = x_k + s_k d_k,$$

où d_k est une direction de descente de f en x_k et $s_k > 0$ un pas vérifiant les conditions de Wolfe. Alors:

$$\sum \cos(\theta_k)^2 \|\nabla f(x_k)\|^2 \quad \text{converge.}$$

f à gradient-Lipschitz si $\forall (x, y), \|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$

θ_k angle entre la direction d_k et $-\nabla f(x_k)$

$$\cos(\theta_k) = \frac{\langle -\nabla f(x_k), d_k \rangle}{\|\nabla f(x_k)\| \cdot \|d_k\|}$$

Rmq: Si: $\exists c > 0, \forall k, \cos(\theta_k)^2 \geq c$ alors on a également
 $\sum \|\nabla f(x_k)\|^2 \leq C$ i.e. Cv globale de l'algorithme

Cv globale des algorithmes de descente avec pas de Wolfe

Théorème de Zoutendijk - Démonstration à travailler

D'après la seconde condition de Wolfe, on a: $\forall k, \langle \nabla f(x_{k+1}), d_k \rangle \geq \varepsilon_2 \langle \nabla f(x_k), d_k \rangle$, d'où:

$$\forall k, \langle \nabla f(x_{k+1}) - \nabla f(x_k), d_k \rangle \geq (\varepsilon_2 - 1) \langle \nabla f(x_k), d_k \rangle.$$

De plus:

$$\begin{aligned} \langle \nabla f(x_{k+1}) - \nabla f(x_k), d_k \rangle &\leq \|\nabla f(x_{k+1}) - \nabla f(x_k)\| \|d_k\| && \text{(Cauchy-Schwartz)} \\ &\leq L \|x_{k+1} - x_k\| \|d_k\| && \text{(\nabla f L-Lipschitz)} \\ &\leq L s_k \|d_k\|^2. \end{aligned}$$

D'où en combinant les deux inégalités précédentes, on obtient:

$$\forall k, 0 \leq (\varepsilon_2 - 1) \langle \nabla f(x_k), d_k \rangle \leq L s_k \|d_k\|^2,$$

soit un pas s_k vérifiant:

$$s_k \geq \frac{\varepsilon_2 - 1}{L} \frac{\langle \nabla f(x_k), d_k \rangle}{\|d_k\|^2} > 0. \quad (1)$$

Sachant que f est supposée bornée inférieurement, on en déduit:

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq -\varepsilon_1 s_k \langle \nabla f(x_k), d_k \rangle && \text{(condition de Armijo)} \\ &\geq \varepsilon_1 \frac{1 - \varepsilon_2}{L} \frac{\langle \nabla f(x_k), d_k \rangle^2}{\|d_k\|^2} && \text{(d'après (1))} \\ &\geq \varepsilon_1 \frac{1 - \varepsilon_2}{L} \cos(\theta_k)^2 \|\nabla f(x_k)\|^2. \end{aligned}$$

Comme $\sum f(x_k) - f(x_{k+1})$ converge, on en déduit que $\sum \cos(\theta_k)^2 \|\nabla f(x_k)\|^2$ converge.

Cv globale des algorithmes de type gradient

$$x_{k+1} = x_k - s_k \nabla f(x_k).$$

① CS pour qu'un algorithme de gradient soit un algorithme de descente i.e. pour que

$$\forall k, f(x_{k+1}) < f(x_k).$$

Supposons f de classe C^1 à gradient L -Lipschitz.

lemme de
descende

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|^2 \\ &\leq f(x_k) - s_k \|\nabla f(x_k)\|^2 + \frac{L}{2} s_k^2 \|\nabla f(x_k)\|^2 \\ &\leq f(x_k) + \underbrace{s_k \left(\frac{L}{2} s_k - 1 \right)}_{>0} \underbrace{\|\nabla f(x_k)\|^2}_{>0} \underset{\text{↑ chain rule puisque } \frac{L}{2}s_k - 1 < 0}{\leq} f(x_k) \end{aligned}$$

CS pour qu'un algo de gradient soit un algo de descente

$$\forall k, s_k < \frac{2}{L}.$$

Cv globale des algorithmes de type gradient

Résultats de CV globale

Supposons que le pas vérifie la condition: $\forall k, s_k < \frac{2}{L}$.

① Algorithme de gradient à pas fixe: $\forall k, s_k = s$.

$$f(x_{k+1}) \leq f(x_k) + s \left(\frac{L}{2}s - 1 \right) \|\nabla f(x_k)\|^2$$

soit:

$$\overbrace{s \left(1 - \frac{L}{2}s \right)}^{>0} \|\nabla f(x_k)\|^2 \leq f(x_k) - f(x_{k+1}).$$

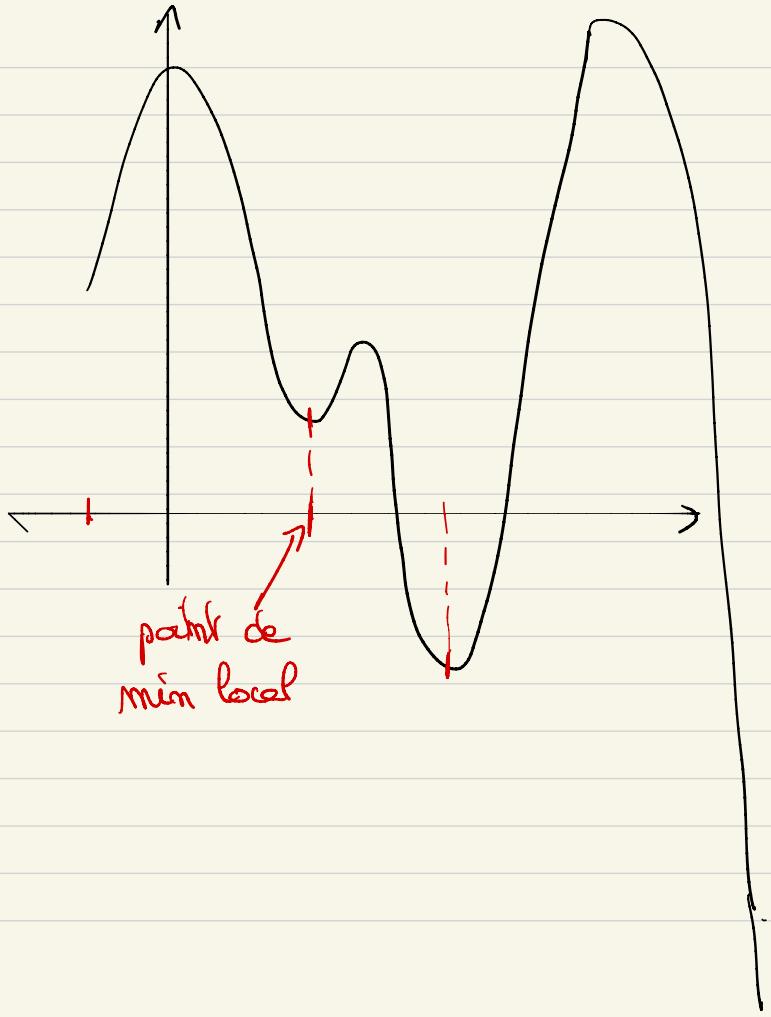
Par un argument de série et en supposant que f est bornée inférieurement, on en déduit que $\sum \|\nabla f(x_k)\|^2$ converge et donc la convergence globale de l'algorithme.

Tout point d'accumulation des $(x_k)_{k \in \mathbb{N}}$ est un point critique de f .

On a de plus $\lim_{k \rightarrow +\infty} \|x_{k+1} - x_k\| = 0$, d'où (démonstration classique mais non triviale) la convergence des itérés $(x_k)_k$ vers un point critique de f .

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

quel que soit le point initial !



Cv globale des algorithmes de type gradient

Résultats de CV globale

② Algorithme de gradient à pas optimal.

$$s_k \left(1 - \frac{L}{2}s_k\right) \|\nabla f(x_k)\|^2 \leq f(x_k) - f(x_{k+1}).$$

Le pas optimal est par définition le pas qui assure que $f(x_k) - f(x_{k+1})$ est le plus grand possible. Nécessairement la décroissance obtenue avec le s_k optimal est meilleure que celle qu'on obtiendrait avec $s_k = \frac{1}{L}$ soit:

$$\frac{1}{2L} \|\nabla f(x_k)\|^2 \leq f(x_k) - f(x_{k+1}).$$

A nouveau on obtient ainsi la convergence globale de l'algorithme et la convergence des itérés $(x_k)_k$ vers un point critique de f .

Cv globale des algorithmes de type gradient

Résultats de CV globale

3 Algorithme de gradient avec pas de Wolfe.

D'après le théorème de Zoutendijk, la série $\sum \cos(\theta_k)^2 \|\nabla f(x_k)\|^2$ converge. Or:

$$\begin{aligned}\cos(\theta_k) &= \frac{\langle -\nabla f(x_k), d_k \rangle}{\|\nabla f(x_k)\| \|d_k\|} \quad \text{ici } d_k = -\nabla f(x_k) \\ &= \frac{\langle -\nabla f(x_k), -\nabla f(x_k) \rangle}{\|\nabla f(x_k)\| \|\nabla f(x_k)\|} = 1.\end{aligned}$$

Ainsi, la série $\sum \|\nabla f(x_k)\|^2$ converge et on a convergence globale de l'algorithme et convergence des itérés $(x_k)_k$ vers un point critique de f .

Méthode de Newton

Point de vue de l'Analyse Numérique

On cherche les points critiques de f ce qui conduit à résoudre le système non linéaire:

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

~ on recherche les pts critiques

Une itération de la méthode de Newton appliquée à la résolution de $F(x) = 0$ avec $F = \nabla f$ s'écrit:

$$x_{k+1} = x_k - J_F(x_k)^{-1} F(x_k)$$

à condition que la matrice jacobienne $J_F(x_k)$ soit inversible, i.e.:

$$x_{k+1} = x_k - H_f(x_k)^{-1} \nabla f(x_k),$$

à condition que la matrice hessienne $H_f(x_k)$ soit inversible.

Algorithme de Newton:

- $x_0 \in \mathbb{R}^n$ quelconque.
- Tant que $\|\nabla f(x_k)\| > \varepsilon$,
 - ▶ d_k solution du système:

$$H_f(x_k)d = -\nabla f(x_k).$$

- ▶ $x_{k+1} = x_k + d_k.$

(pas finie = 1)

Rappels sur la méthode de Newton

Point de vue de l'optimisation

Soit x_k l'itéré courant, au lieu de minimiser $f(x_k + d)$, on va chercher à résoudre:

$$(P) \quad \min_{d \in \mathbb{R}^n} q_k(d) := f(x_k) + \langle \nabla f(x_k), d \rangle + \frac{1}{2} \langle H_f(x_k)d, d \rangle.$$

C'est un problème d'optimisation quadratique sans contrainte:

- Points critiques:

$$\nabla q_k(d_k) = 0 \Leftrightarrow H_f(x_k)d_k = -\nabla f(x_k).$$

- d_k est un point de minimum global de f ssi $H_f(x_k)$ est définie positive.

Algorithme de Newton:

- $x_0 \in \mathbb{R}^n$ quelconque.
- Tant que $\|\nabla f(x_k)\| > \varepsilon$,
 - d_k solution du système:
$$H_f(x_k)d = -\nabla f(x_k).$$
 - $x_{k+1} = x_k + d_k.$

La méthode de Newton: une méthode de descente ?

Algorithme de Newton:

- $x_0 \in \mathbb{R}^n$ quelconque.
- Tant que $\|\nabla f(x_k)\| > \varepsilon$,
 - ▶ d_k solution du système:
$$H_f(x_k)d = -\nabla f(x_k).$$
 - ▶ $x_{k+1} = x_k + d_k.$

Supposons que la matrice $H_f(x_k)$ est inversible à chaque itération: la méthode de Newton est un algorithme de:

- direction de recherche:

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

C'est un algorithme de descente ssi: $\langle \nabla f(x_k), d_k \rangle < 0$

i.e. $\forall k \in \mathbb{N}, \langle H_f(x_k)^{-1}\nabla f(x_k), \nabla f(x_k) \rangle > 0.$

Il suffit donc que la matrice hessienne $H_f(x_k)$ soit définie positive à chaque itération.

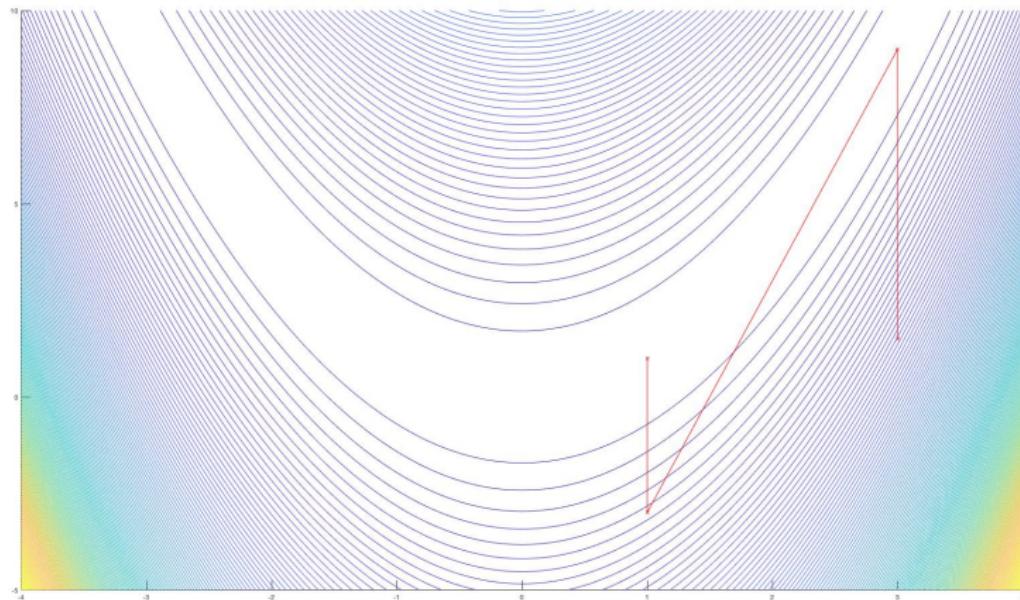
- pas fixe égal à 1.

Illustration de l'algorithme de Newton en optimisation

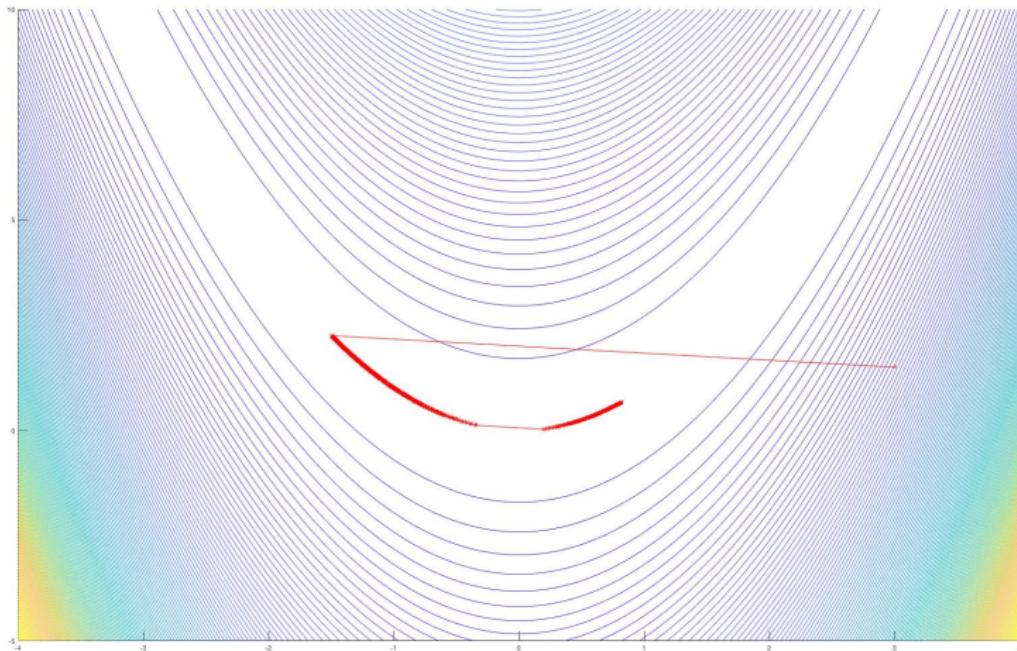
Soit:

$$\min_{(x,y) \in \mathbb{R}^2} f(x, y) = (1 - x)^2 + 100(y - x^2)^2.$$

f est non convexe et admet un unique point de minimum (global) en $(1, 1)$.



Comparaison avec la méthode du gradient à pas optimal

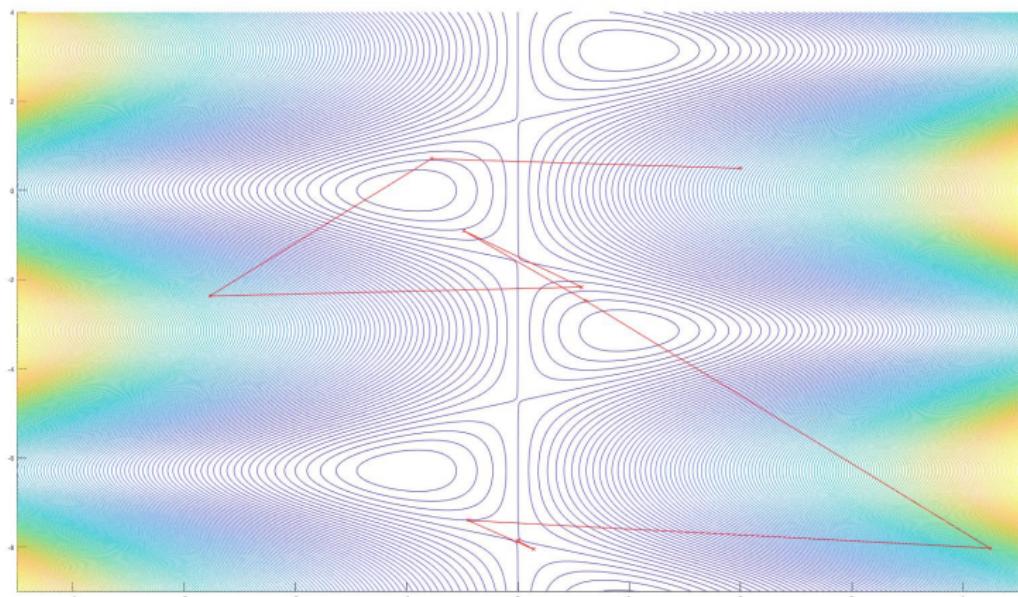


L'algorithme du gradient à pas optimal a été stoppé à 1000 itérations alors que la méthode de Newton converge en 5 itérations !

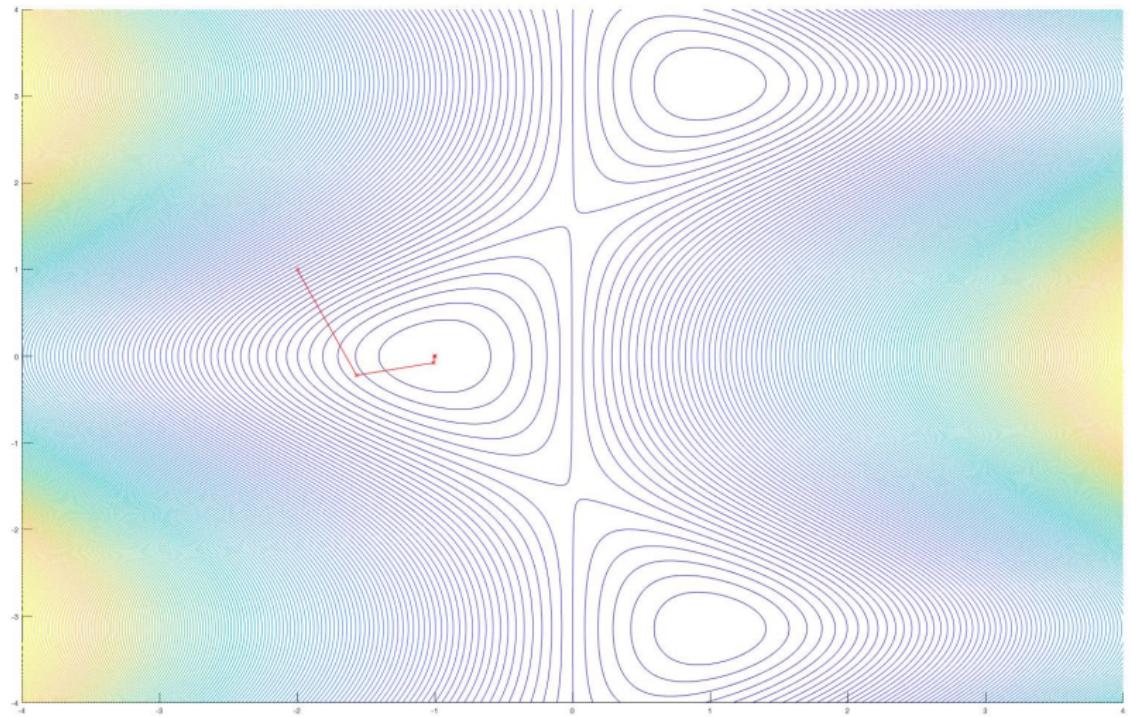
L'algorithme de Newton sur un autre exemple

$$g(x, y) = \frac{1}{2}x^2 + x \cos y, \quad ((-1)^{k+1}, k\pi) \quad \text{points de min local de } g$$

$$(0, \frac{\pi}{2} + k\pi), \quad \text{points col de } g$$



Convergence locale de l'algorithme de Newton



Comment améliorer la méthode de Newton ?

Enjeux:

- Assurer la convergence globale de l'algorithme. *ie à point initial.*
- Se débarrasser de la matrice hessienne: calcul coûteux, pas toujours disponible, éventuellement mal conditionnée.
- Bénéficier, quand on le peut, de la rapidité de convergence de l'algorithme de Newton classique.

Comment améliorer la méthode de Newton ?

Enjeux:

- Assurer la convergence globale de l'algorithme.
 - ↪ Algorithme de Newton avec une recherche linéaire un peu plus sophistiquée (recherche linéaire de Wolfe).
- Se débarrasser de la matrice hessienne: calcul couteux, pas toujours disponible, éventuellement mal conditionnée.
- Bénéficier, quand on le peut, de la rapidité de convergence de l'algorithme de Newton classique.

Comment améliorer la méthode de Newton ?

Enjeux:

- Assurer la convergence globale de l'algorithme.
 - ↪ Algorithme de Newton avec une recherche linéaire un peu plus sophistiquée (recherche linéaire de Wolfe).
- Se débarrasser de la matrice hessienne: calcul couteux, pas toujours disponible, éventuellement mal conditionnée.
 - ↪ Algorithmes de quasi-Newton (approximation de la hessienne en n'utilisant que les informations sur le gradient & recherche linéaire de Wolfe).
- Bénéficier, quand on le peut, de la rapidité de convergence de l'algorithme de Newton classique.

Méthode de Newton avec recherche linéaire

But: assurer la convergence globale de l'algorithme.

Deux ingrédients:

- Garantir que d_k soit une direction de descente
- Recherche linéaire de Armijo / Wolfe.

- $x_0 \in \mathbb{R}^n$ quelconque.
- Tant que $\|\nabla f(x_k)\| > \varepsilon$,

▶ Calculer:

$$H_k = \begin{cases} H_f(x_k) & \text{si } H_f(x_k) \succ 0 \\ H_f(x_k) + \alpha I_n & \text{sinon,} \end{cases}, \quad \text{avec: } \alpha > -\lambda_{\min}(H_f(x_k)).$$

↳ α être choisi de sorte que: $\forall d \in \text{Sp}(H_f(x_k)), d + \alpha d \succ 0$

▶ d_k unique solution du système:

ie tq $H_k d_k, H_k d_k \succ 0$

$$H_k d = -\nabla f(x_k).$$

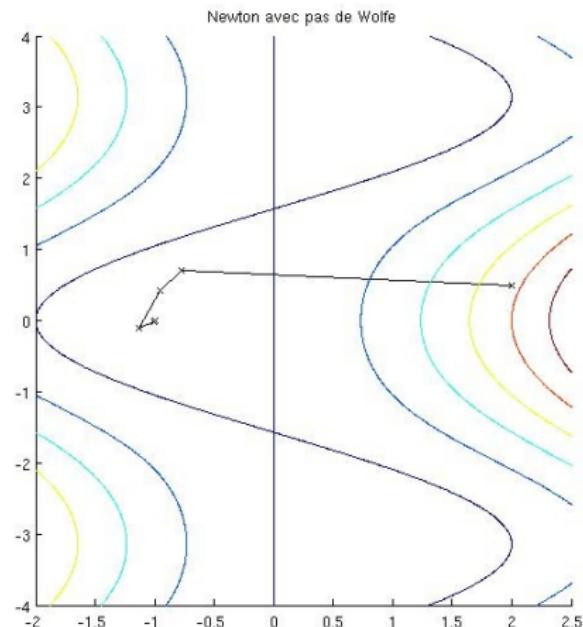
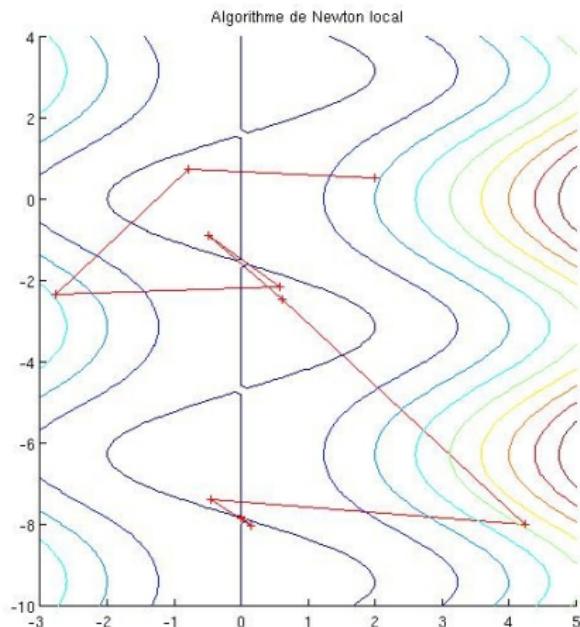
▶ Calculer un pas s_k à l'aide d'une recherche linéaire de Wolfe.

$$x_{k+1} = x_k + s_k d_k.$$

ici d_k est toujours une direction de descente de f en x_k !!

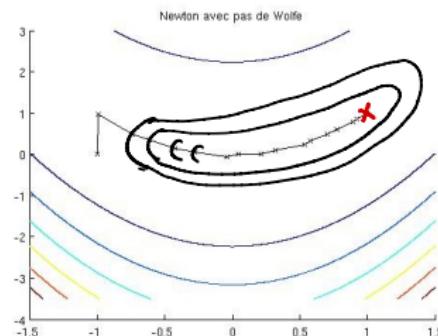
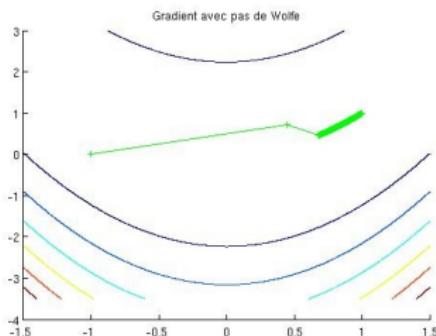
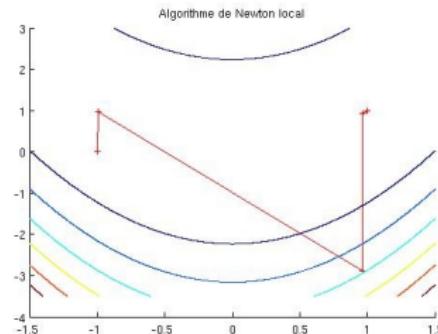
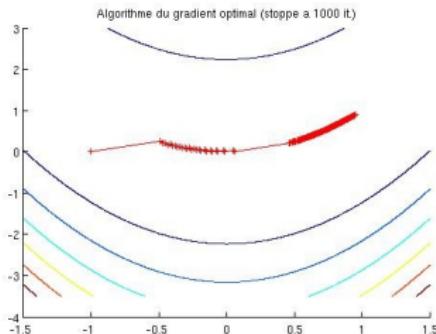
Algorithme de Newton avec pas de Wolfe

Minimiser $f(x, y) := \frac{1}{2}x^2 + x \cos(y)$, $(x_0, y_0) = (2, 0.5)$.



Algorithmes de gradient et de Newton avec pas de Wolfe

Minimiser $f(x, y) := 100(y - x^2)^2 + (1 - x)^2$, $(x_0, y_0) = (-1, 0)$



CV globale de l'algorithme de Newton avec LS

Supposons que f est de classe C^1 à gradient L -Lipschitz admettant au moins un minimiseur.

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k - s_k H_k^{-1} \nabla f(x_k)$$

où s_k est un pas de Wolfe et $H_k \succ 0$ par construction. On démontre alors:

L'algorithme de Newton avec pas de Wolfe converge globalement à condition qu'il existe $M > 0$ tel que:

$$\forall k \in \mathbb{N}, \kappa_2(H_k) = \left\| H_k \right\| \left\| H_k^{-1} \right\| \leq M.$$

Démo: d'après le th de Zoutendijk, on a: $\sum \cos^2(\theta_k) \|\nabla f(x_k)\|^2 \rightarrow 0$

où θ_k est l'angle entre la direction d_k et $-\nabla f(x_k)$

$$\cos(\theta_k) = \frac{\langle -\nabla f(x_k), d_k \rangle}{\|\nabla f(x_k)\| \cdot \|d_k\|} = \frac{\langle H_k^{-1} \nabla f(x_k), \nabla f(x_k) \rangle}{\|H_k^{-1} \nabla f(x_k)\| \cdot \|\nabla f(x_k)\|}$$

$$\forall k, \quad \langle H_k^{-1} \nabla f(x_k), \nabla f(x_k) \rangle \geq \lambda_{\min}(H_k^{-1}) \|\nabla f(x_k)\|^2$$

Rappel: A sym. réelle $\xrightarrow[m]{}$ A diagonalisable obs une b.o.m de rep (v_i)

$$\forall x, \exists \alpha_i \in \mathbb{R} \text{ tq } x = \sum_{i=1}^n \alpha_i v_i : \langle Ax, x \rangle = \sum_{i=1}^n \lambda_i \alpha_i^2 \geq \lambda_{\min}(A) \sum \alpha_i^2 \geq \lambda_{\min}(A) \|x\|^2$$

CV globale de l'algorithme de Newton avec LS

Supposons que f est de classe C^1 à gradient L -Lipschitz admettant au moins un minimiseur.

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k - s_k H_k^{-1} \nabla f(x_k)$$

où s_k est un pas de Wolfe et $H_k \succ 0$ par construction. On démontre alors:

L'algorithme de Newton avec pas de Wolfe converge globalement à condition qu'il existe $M > 0$ tel que:

$$\forall k \in \mathbb{N}, \kappa_2(H_k) = \|H_k\| \|H_k^{-1}\| \leq M.$$

On a donc : $\forall k, \quad \langle H_k^{-1} \nabla f(x_k), \nabla f(x_k) \rangle \geq \frac{1}{\lambda_{\min}(H_k)} \|\nabla f(x_k)\|^2$

$$\geq \frac{\|\nabla f(x_k)\|^2}{\|H_k\|}$$

Or $\|H_k^{-1} \nabla f(x_k)\| \leq \lambda_{\max}(H_k^{-1}) \|\nabla f(x_k)\| = \|H_k^{-1}\| \cdot \|\nabla f(x_k)\|$

Donc : $\forall k, \cos(\theta_k) \geq \frac{\|\nabla f(x_k)\|^2}{\|I + R_k\|} \times \frac{1}{\|H_k^{-1}\| \|\nabla f(x_k)\|^2}$

$$\geq \frac{1}{\|H_k\| \cdot \|H_k^{-1}\|} = \frac{1}{\kappa_2(H_k)}$$

On voudrait : $\exists \alpha > 0 \text{ tq } \cos(\theta_k) \geq \alpha \text{ ie } (\text{cs}) \kappa_2(H_k) \text{ reste borné}$

CV globale de l'algorithme de Newton avec LS

Supposons que f est de classe C^1 à gradient L -Lipschitz admettant au moins un minimiseur.

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k - s_k H_k^{-1} \nabla f(x_k)$$

où s_k est un pas de Wolfe et $H_k > 0$ par construction. On démontre alors:

L'algorithme de Newton avec pas de Wolfe converge globalement à condition qu'il existe $M > 0$ tel que:

$$\forall k \in \mathbb{N}, \kappa_2(H_k) = \|H_k\| \|H_k^{-1}\| \leq M.$$

On suppose $\exists n \geq 0, \forall k, \kappa_2(H_k) \leq M$. On a alors

$$\forall k \quad 0 \leq \frac{1}{M^2} \|\nabla f(x_k)\|^2 \leq \underbrace{\kappa_2(H_k)^2 \|\nabla f(x_k)\|^2}_{\text{terme général d'une suite CV}}$$

Donc $\sum \|\nabla f(x_k)\|^2$ CV ie $\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0, \forall x_k$

$$\|A\|_2 \stackrel{\text{def}}{=} \sup_{\alpha \neq 0} \frac{\|A\alpha\|_2}{\|\alpha\|_2} \stackrel{?}{=} \lambda_{\max}(A)$$

$$\frac{\|A\alpha\|_2}{\|\alpha\|_2} = \frac{\|\lambda_{\max}(A)\alpha\|_2}{\|\alpha\|_2} = \lambda_{\max}(A)$$

$\forall \alpha, \|A\alpha\|_2 \leq \lambda_{\max}(A) \|\alpha\|_2$. d'où $\|A\|_2 \leq \lambda_{\max}(A)$

$$\|Ax\|_2 \leq \|A\|_2 \|x\|_2$$

Propriété des normes subordonnées.

A symétrique réelle donc diagonalisable dans une base de $\text{rep}(v_i)$

$$\forall x, \exists x_i, x = \sum_{i=1}^m x_i v_i$$

$$\begin{aligned}\|Ax\|_2^2 &= \left\| \sum_{i=1}^m x_i \underbrace{Av_i}_{\lambda_i v_i} \right\|_2^2 = \left\| \sum_{i=1}^m \lambda_i x_i v_i \right\|_2^2 \\ &= \sum_{i=1}^m \lambda_i^2 x_i^2 \leq \lambda_{\max}(A)^2 \sum_{i=1}^m x_i^2 \\ &\leq \lambda_{\max}(A)^2 \|x\|_2^2\end{aligned}$$

ici ça marche car $A \succeq 0$.

Algorithmes de quasi-Newton

Principe général

Connaissant x_{k-1} et x_k ,

- On construit une approximation H_k de $H_f(x_k)$ comme solution de:

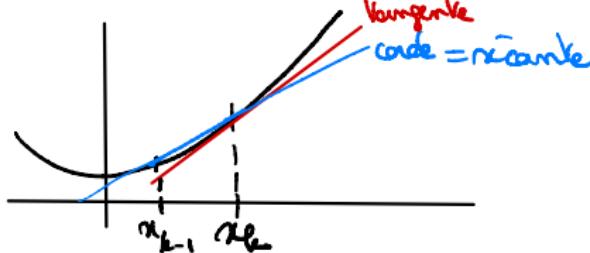
$$H_k(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1}) \quad (\text{Eq. de sécante})$$

système de n équations à n^2 inconnues as une infinité de sol.

- Recherche linéaire: choix du pas s_k .

- $x_{k+1} = x_k - s_k H_k^{-1} \nabla f(x_k)$.

Illustration en dim 1:



$$g : \mathbb{R} \rightarrow \mathbb{R}$$

$$g'(x) = \lim_{\Delta x \rightarrow 0} \frac{g(x+\Delta x) - g(x)}{\Delta x} \approx g'(x) \approx \frac{g(x+\Delta x) - g(x)}{\Delta x}$$

Dév de Taylor de ∇f

$$\nabla f(x) = \nabla f(x_k) + H_p(x_k)(x - x_k) \rightarrow (||x - x_k||)$$

On prend $x = x_{k-1}$

$$\nabla f(x_{k-1}) - \nabla f(x_k) = H_p(x_k)(x_{k-1} - x_k) \rightarrow (||x_{k-1} - x_k||)$$

On cherche H_p tq: $\rightarrow (||x_{k-1} - x_k||)$

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

Algorithmes de quasi-Newton

Principe général

Connaissant x_{k-1} et x_k ,

- ① On construit une approximation H_k de $H_f(x_k)$ comme solution de:

$$H_k(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1}) \quad (\text{Eq. de sécante})$$

- ② Recherche linéaire: choix du pas s_k .

- ③ $x_{k+1} = x_k - s_k H_k^{-1} \nabla f(x_k)$.

OU on remarque que: $x_k - x_{k-1} = H_k^{-1} (\nabla f(x_k) - \nabla f(x_{k-1}))$



Connaissant x_{k-1} et x_k ,

- ① On construit une approximation B_k de $H_f(x_k)^{-1}$ comme solution de:

$$x_k - x_{k-1} = B_k (\nabla f(x_k) - \nabla f(x_{k-1})) \quad (\text{Eq. de sécante})$$

- ② Recherche linéaire: choix du pas s_k .

- ③ $x_{k+1} = x_k - s_k B_k \nabla f(x_k)$.

Algorithmes de quasi-Newton

Idée générale de ces méthodes

Comment choisir H_k ? On a qui il existe une infinité de matrices H_k :

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

Broyden 1965

Pour calculer H_k , on va résoudre :

Exercice à faire

$$\min \frac{1}{2} \| H - H_{k-1} \|^2$$

$$\text{soit } H_k(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1})$$

On trouve :

$$H_k = H_{k-1} + \frac{(y_{k-1} - H_{k-1}\sigma_{k-1})\sigma_{k-1}^T}{\sigma_{k-1}^T \sigma_{k-1}}$$

$$\text{avec } \sigma_{k-1} = x_k - x_{k-1}, \quad y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$$

H_k n'est pas nécessairement symétrique.

Algorithmes de quasi-Newton

Idée générale de ces méthodes

On va impoer la symétrie de H_k ! (puisque $H_k(x_k)$ l'est)

$$\min_H \frac{1}{2} \|H - H_k\|^2$$

avec $\begin{cases} \text{Eq de récurrence} \\ H^T = H \end{cases}$

pour \neq normes.

Algorithmes de quasi-Newton

Les méthodes les plus connues/utilisées

En posant:

$$\sigma_k = x_{k+1} - x_k \quad \text{et} \quad y_k = \nabla f(x_k) - \nabla f(x_{k-1}),$$

on a:

Méthode DFP (Davidson, Fletcher, Powell. 1959-63)

$$H_{k+1} = H_k + \frac{y_k y_k^\top}{y_k^\top \sigma_k} - \frac{H_k \sigma_k \sigma_k^\top H_k}{\sigma_k^\top H_k \sigma_k}.$$

Méthode BFGS (Broyden, Fletcher, Goldfarb, Shannon. 1969-70)

$$B_{k+1} = \left(I - \frac{\sigma_k y_k^\top}{y_k^\top \sigma_k} \right)^\top B_k \left(I - \frac{\sigma_k y_k^\top}{y_k^\top \sigma_k} \right) + \frac{\sigma_k \sigma_k^\top}{y_k^\top \sigma_k}.$$

Les deux formules sont duales l'une de l'autre !

Exercice : vérifiez que $H_{k+1} \sigma_k = y_k$ (eq. de réconstr.)

Algorithmes de quasi-Newton

Algorithme BFGS

x_0 point initial et H_0 matrice symétrique définie positive arbitraires

- ① $k := 0$
- ② Tant que "test d'arrêt" non satisfait,

① Choix de la direction:

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

- ② Recherche linéaire de Wolfe pour calculer un pas $s_k > 0$.

→ Pour maintenir la définie positivité de B_k et garantir que d_k est une direction de descente de f en x_k .

- ③ $x_{k+1} = x_k - s_k B_k \nabla f(x_k)$.

- ④ Mises à jour:

- ★ de la matrice B_k en B_{k+1} par la formule BFGS.
- ★ $k := k + 1$;

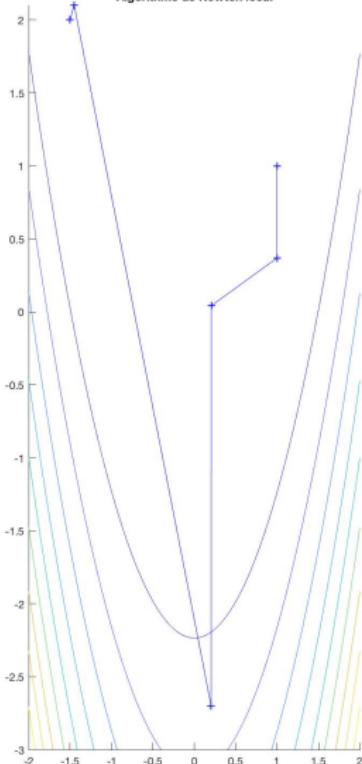
- ⑤ Retourner x_k .

Exercice: vérifier que l'algorithme ci-dessus est bien un algorithme de descente.

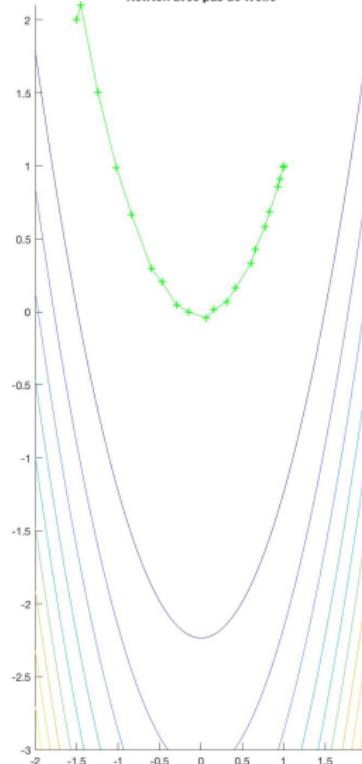
Algorithmes de quasi-Newton

Tests numériques

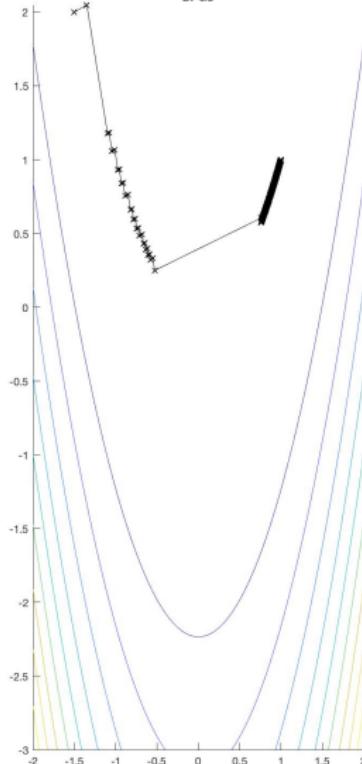
Algorithme de Newton local



Newton avec pas de Wolfe



BFGS



Algorithmes de quasi-Newton

I Propriétés et cv de l'algorithme BFGS