Chapitre 2:

Optimisation sans contraintes

I.DAMERGI FAIZ

Plan

- Existence et unicité d’une solution
- Conditions d’optimalité
- Algorithmes d’optimisation sans contrainte
    - Méthode de descente
    - Méthodes de gradient
    - Méthode de Newton

Un problème d'optimisation sans contrainte est de type

(P) $$\min_{x \in X} f(x)$$

où *f* une fonction $f : X \to \mathbb{R}$, au moins différentiable.

Sans perte de généralité, nous supposons que $X = \mathbb{R}^n$).

1-Existence et unicité d’un minimum

Proposition : Si f est continue vérifiant $\lim_{\|x\| \to +\infty} f(x) = +\infty$.

Alors f admet au moins un minimum c.à.d le problème de minimisation (P) admet au moins une solution.

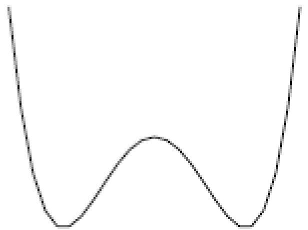
Remarque:

Une fonction f continue et minorée sur $\mathbb{R}^n$ admet une borne inférieure, mais pas forcément de minimum si elle n'est pas infinie à l'infini.

exemple: La fonction $x \to e^{-x}$ est positive, mais elle n'a pas de minimum sur $\mathbb{R}$.

Remarque:

Ce résultat donne l'existence d'un minimum mais pas l'unicité.
En effet la fonction $x \to x^4 - 2x^2$ *admet deux minima atteints en x = -1 et x = 1.*



La fonction $x \mapsto x^4 - 2x$

Proposition:

Si la fonction $f : \mathbb{R}^n \to \mathbb{R}$ est strictement convexe et admet un minimum sur $\mathbb{R}^n$, alors il est unique.

2-Condition d’optimalité

Définition: *(point critique)*
On dit que x est un point critique de f si $\nabla f(x) = 0$.

Théorème 1 :*(condition nécessaire d’optimalité CNO)*

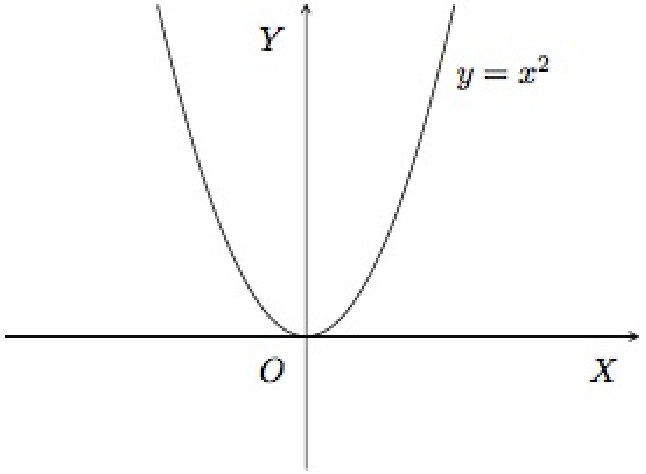
Si f admet un minimum local x, alors $\nabla f(x) = 0$ *(x est un point critique).*

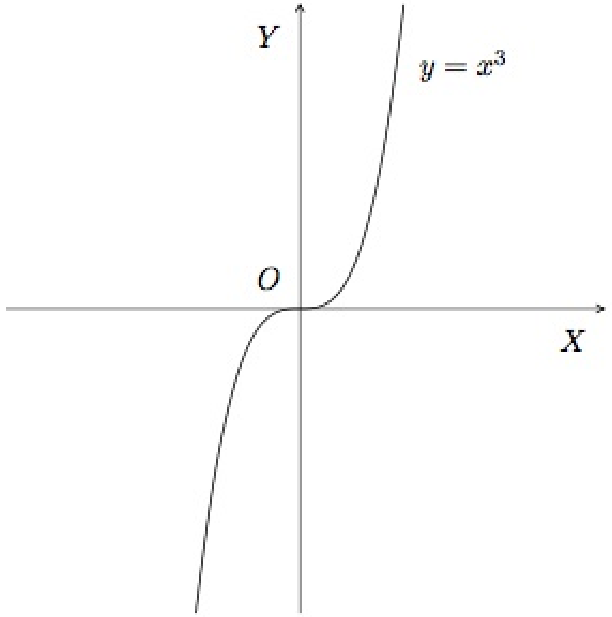
De plus, si f est deux fois différentiable dans un voisinage de x alors $\nabla^2 f(x)$ *est une matrice semi définie positive*

Remarque :

Les conditions du Théorème1 donnent des conditions nécessaires non suffisantes. Tout point critique n'est pas nécessairement un extremum. En effet

$f(x) = x^2$; $x = 0$ *est un point critique qui est aussi un minimum local.*
$f(x) = x^3$; $x = 0$ *est un point critique qui n'est minimum local ni global.*





Théorème 2 : *(condition suffisante d'optimalité CSO)*

Si $\nabla f(x) = 0$ *et* $\nabla^2 f(x)$ *est symétrique définie positive, alors x est un minimum local de f.*

Remarque :

La condition du Théorème 2 est suffisante non nécessaire. En effet,
$f(x) = x^4$; $x = 0$ *est un point critique de f mais* $\nabla^2 f(x)$ *en ce point n'est pas définie positive (elle est nulle).*

Théorème : *(condition suffisante d'optimalité globale CSG)*

Soit x est un point critique de f.

i) Si f est convexe, alors x est un point minimum global de f.
ii) Si f est strictement convexe, alors x est l'unique point de minimum global de f.

Proposition:

Soit $f : \mathbb{R}^2 \to \mathbb{R}$, $\boldsymbol{C}^2$ telle que $\nabla f(\bar{x}, \bar{y}) = 0$. Posons

$$\nabla^2 f(\bar{x}, \bar{y}) = \begin{pmatrix} r & s \\ s & t \end{pmatrix},$$

alors

- Si $rt - s^2 > 0$, $r < 0$ => f admet un maximum local en $(\bar{x}, \bar{y})$
- Si $rt - s^2 > 0$, $r > 0$ => f admet un minimum local en $(\bar{x}, \bar{y})$
- Si $rt - s^2 < 0$ => f n'admet pas d'extremum en $(\bar{x}, \bar{y})$
- Si $rt - s^2 = 0$ => on ne peut pas conclure

Remarque:

Si $rt - s^2 < 0$, $les\ vp\ sont\ de\ signes\ opposés$ => $\nabla^2 f(\bar{x}, \bar{y})$ n'est ni définie positive ni définie négative. Avec le théorème 2, on ne peut rien conclure car il nous donne une condition suffisante non nécessaire.

Par contre si $rt - s^2 < 0$ alors les v p de A sont de signes opposés et ≠ 0 => A n'est ni semi définie positive ni semi définie négative. D'après la négation de la CNO d'ordre 2, $(\bar{x}, \bar{y})$ n'est pas un extremum

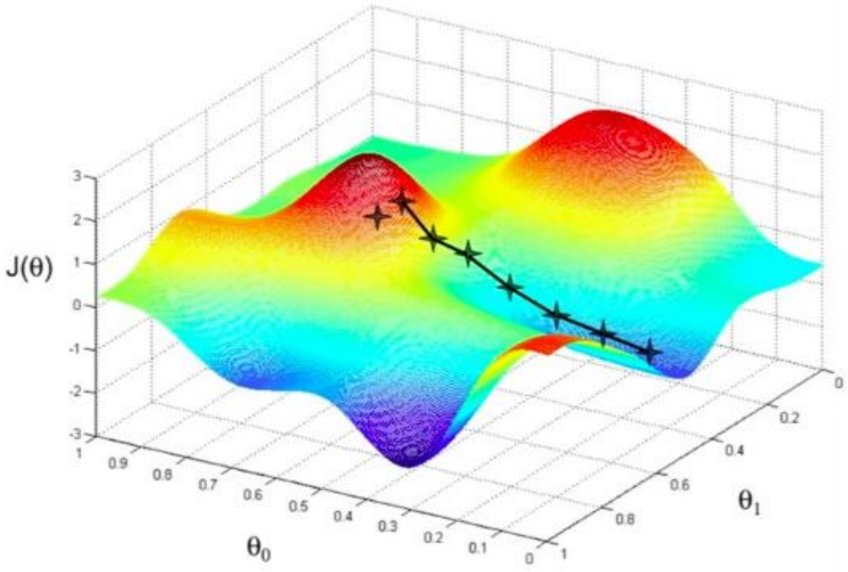
3-Algorithmes d'optimisation sans contrainte

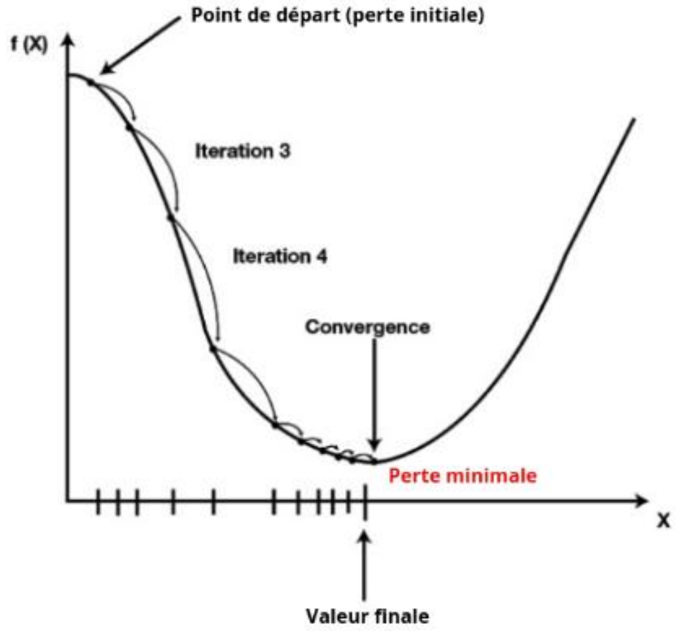
- On s'intéresse aux algorithmes de calcul de minimum et plus particulièrement aux algorithmes de descente:

  Partant d'un point x_0 arbitraire choisi, un algorithme de descente va chercher à construire une suite itérés $(x_k)_{k\in\mathbb{N}}$ vérifiant $\forall\, k \in \mathbb{N}$

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

  et qui converge vers la solution optimale





3-1- Méthodes de descente

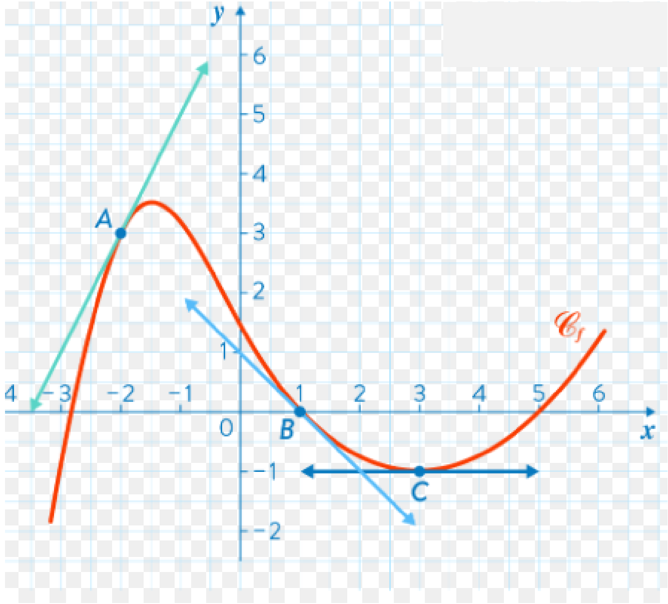
Définition *: (Direction de descente)*

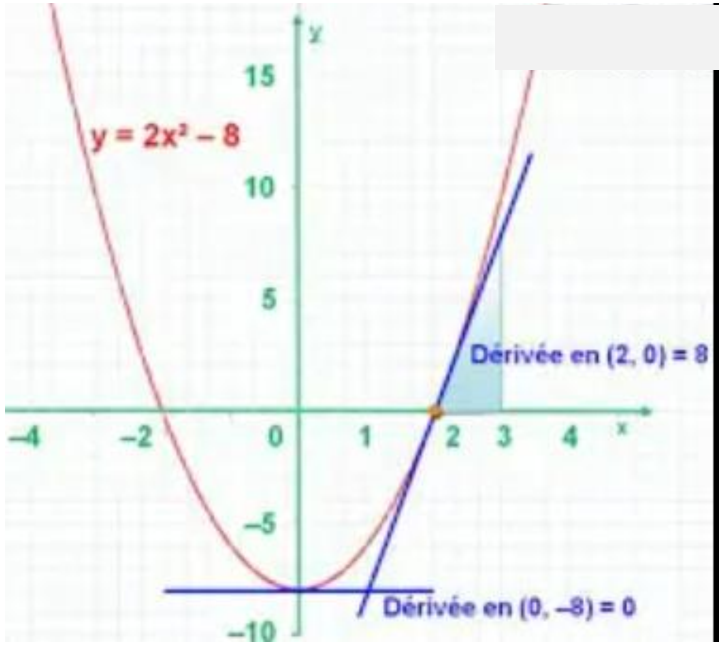
Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $x \in \mathbb{R}^n$. *On dit que le vecteur* $d \in \mathbb{R}^n \backslash \{0\}$ *est une direction de descente pour* f *à partir de* x, *si* $\exists \eta > 0 \quad tq \quad \forall t \in [0, \eta] \quad f(x + td) \leq f(x)$

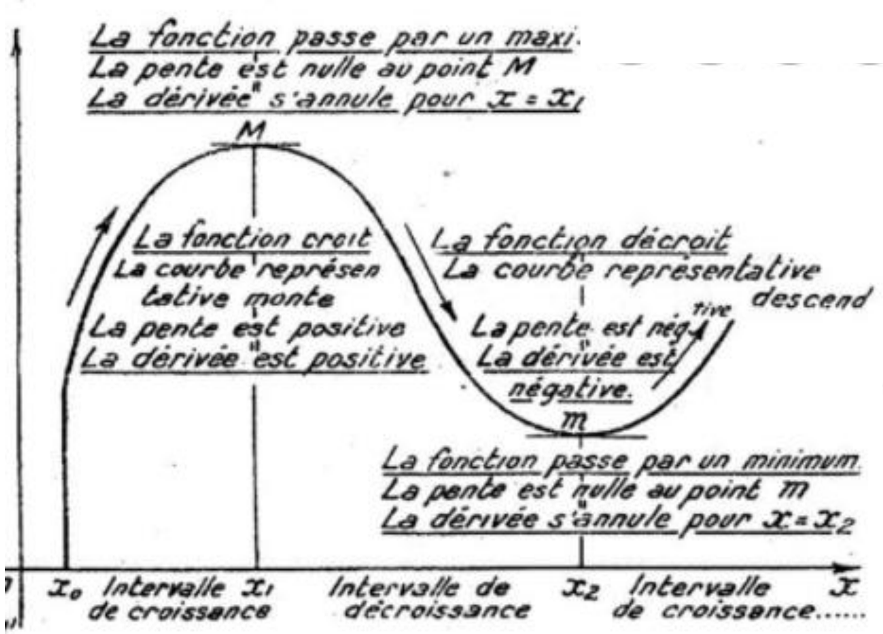
Proposition :

Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $x \in \mathbb{R}^n$. *Le vecteur* $d \in \mathbb{R}^n$ *est une direction de descente pour* f *à partir de* x *si et seulement si* $\nabla f(x)^t d < 0$

- Lorsqu’elle existe la dérivée directionnelle donne des informations sur la pente de la fonction dans la direction d (tout comme la dérivée donne des informations sur la pente des fonctions à une seule variable)







- Si df(x,d)=∇f(x)^t . d >0 alors f est croissante dans la direction d

- Si df(x,d)=∇f(x)^t . d <0 alors f est décroissante dans la direction d. et dans ce cas d est bien une direction de descente

- Ces méthodes itératives sont basées sur le choix à l'étape *k*

  - d'une direction d_k

  - d'un scalaire t_k

- On choisi un "test d'arrêt" convenable, par exemple $||\nabla f(x_k)|| \leq \varepsilon$. (test d'optimalité)

où ε est la précision demandée.

Ainsi, une méthode de descente construit une suite d'itétés (x_k) suivant l'algorithme suivant

Algorithme de descente

(1) *initialisation : choisir* x_0

(2) ***tant que*** *"test d'arrêt" non satisfait*

(a) choisir une direction de descente d_k *telle que* $\nabla f(x_k)^t d_k < 0$

(b) recherche linéaire : choisir un pas $t_k > 0$ *tel que* $f(x_k + t_k d_k) < f(x_k)$

(c) Mise à jour $x_{k+1} = x_k + t_k d_k$; $k = k + 1$

fin tant que

Critère d’arrêt :

- Soit x* la solution optimale càd le minimum local de f à optimiser
- Théoriquement, l’algorithme construit une suite de points $x_0, x_1, \ldots x_k, \ldots$. qui converge vers x*
- En pratique, on choisi un test d’arrêt pour garantir que l’algorithme s’arrête après un nb fini d’itérations et que le dernier point calculé soit suffisamment proche de x* (une approximation)
- le test d’optimalité $\|\nabla f(x_k)\| \le \varepsilon.$ n’est pas toujours satisfait . On fait appel à d’autres critères:

Stagnation de la solution : $\|x_{k+1}-x_k\|<\|x_k\|$

Stagnation de la valeur courante de f: $|f(x_{k+1})-f(x_k)|<|f(x_k)|$

Nombre d’itération dépassant un seuil fixé à l’avance : **itermax**>k

Remarque

La condition d'optimalité $||\nabla f(x_k)|| \leq \varepsilon$. garanti la convergence de l'algorithme mais ne suppose pas que l'algorithme converge vers un minimum

Exemple :

soit f(x,y)=x^2-y^2-y^4 , qui admet $(0,\pm 1/\sqrt{2}\,)$ $des\ \ minumus\ globaux$

En partant de x_0=(1,0), l'algorithme de descente converge vers (0,0) qui est bien de gradient nul mais pas un point minimum.
Pour cela, on doit voir aussi la notion de vitesse de convergence qui mesure l'évolution de l'erreur commise $||x_k - x^*||$

Un algorithme de descente est complètement déterminé par les stratégies de choix des directions de descente successives et du pas effectué à chaque itération dans la direction choisie.

Il existe deux stratégies de choix de direction de descente

- stratégie de Cauchy : $d_k = -\nabla f(x_k)$, conduisant aux algorithmes de gradient.
- stratégie de Newton : $d_k = \nabla^2 f(x_k)^{-1} \nabla f(x_k)$, conduisant aux algorithmes de Newton.

3-2-Méthodes de gradient

Parmi toutes les directions de descente existant en un point $x \in \mathbb{R}^n$ donné, la direction où la pente est la plus forte est celle du gradient $d = -\nabla f(x)$

En effet, on écrit le développement de taylor de $f(x_{k+1})$ au voisinage de x_k

$$
\begin{aligned}
f(x_{k+1}) &= f(x_k - t\,\nabla f(x_k)) \\
&= f(x_k) + \langle \nabla f(x_k), -t\,\nabla f(x_k) \rangle + o(t) \\
&= f(x_k) - t \langle \nabla f(x_k), \nabla f(x_k) \rangle + o(t) \\
&= f(x_k) - t\, \| \nabla f(x_k) \|^2 + o(t)
\end{aligned}
$$

Pour $t>0$ « assez petit » et $\nabla f(x_k) \neq 0$, la quantité $f(x_{k+1}) - f(x_k) < 0$

Ainsi, la direction $-\nabla f(x)$ est la direction de plus forte descente de f au point x.

a- Algorithme de gradient à pas fixe

L'idée est d'imposer une fois pour toute le pas de descente $t > 0$ (fixé)

(1) initialisation : choisir x_0

(2) tant que "test d'arrêt" non satisfait

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

(b) Mise à jour $x_{k+1} = x_k - t\nabla f(x_k)\,;\, k = k+1$

fin tant que

b-Algorithme de gradient à pas optimal

L'idée est de calculer à chaque itération le pas qui minimise la fonction dans la direction de descente donnée par le gradient

(1) initialisation : choisir $x0$

(2) tant que "test d'arrêt" non satisfait

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

(b) choisir $t_k \geq 0$ solution de $\min_{t>0} f(x_k + td_k)$

(c) Mise à jour $x_{k+1} = x_k + t_k d_k\,;\, k = k+1$

fin tant que

Remarque :

Ces deux algorithmes de gradient à pas fixe ou optimal se caractérisent par:

- *La non-garantie de cv pour l'algorithme de gradient à pas fixe. Dans la pratique, on prend un pas « assez petit » pour garantir la cv. Dans ce cas, elle peut être très lente.*

- *la lenteur de la méthode de gradient à pas optimal est due au comportement en zigzag des itérés lorsqu'on se rapproche de la solution.*

En effet, à l'itération k + 1, l'algorithme à pas optimal minimise $\varphi : t \rightarrow f(xk - t\nabla f(x_k))$

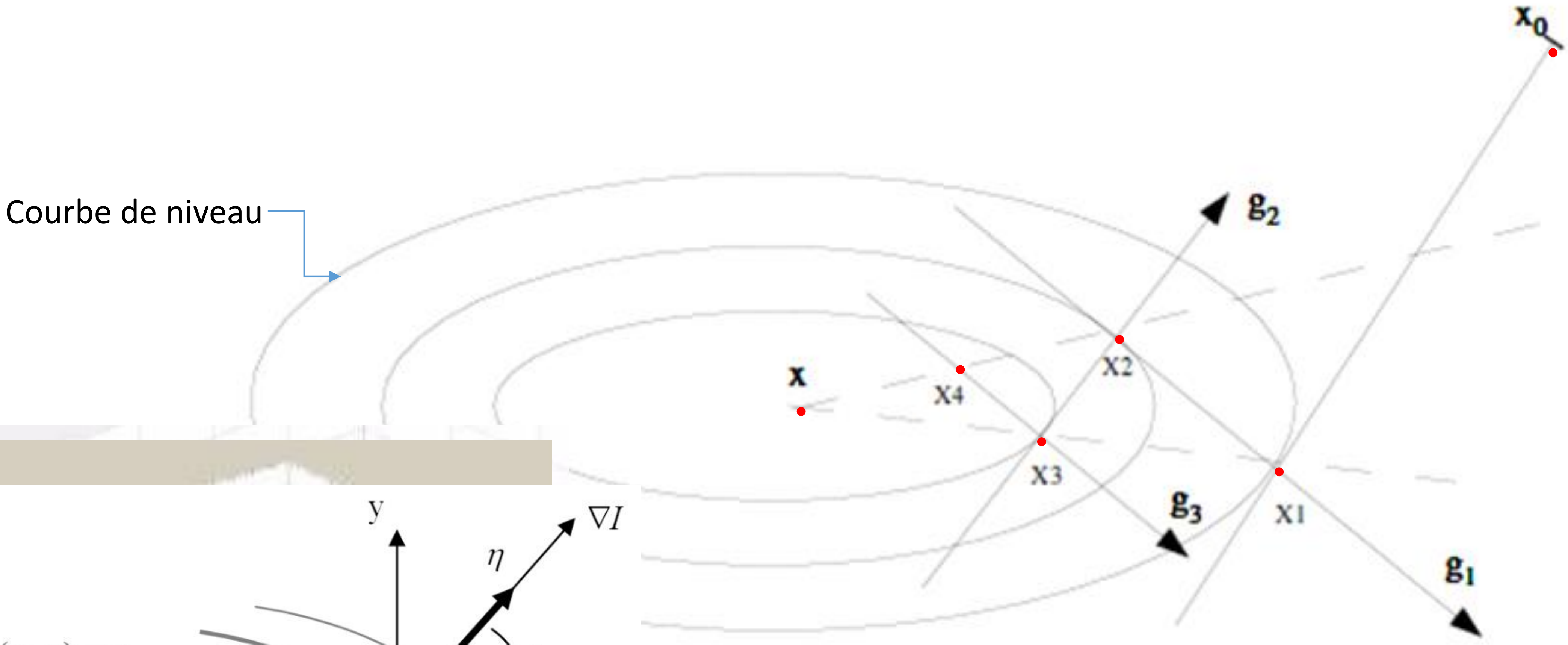
f étant supposé différentiable, la fonction φ est dérivable sur $\mathbb{R}$ de dérivée:

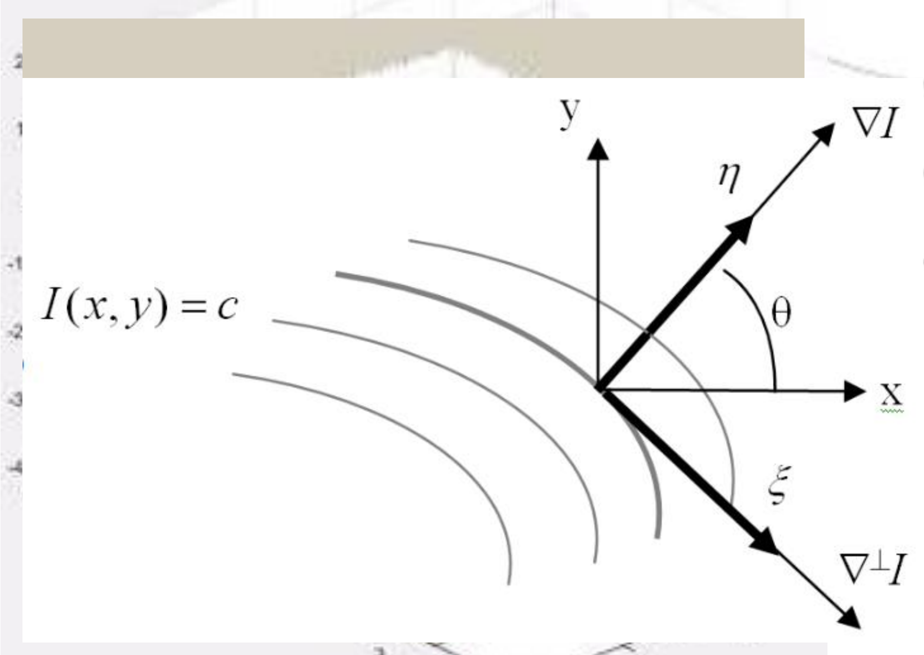
$$\begin{aligned}\varphi'(t) &= -< \nabla f(x_k), \nabla f(x_k - t\nabla f(x_k)) > . \\ &= - \ < \nabla f(x_k), \nabla f(x_{k+1}) >\end{aligned}$$

Soit t_k le pas optimal calculé, donc il vérifie : $\varphi'(t_k) = 0$.

Deux directions de descente successives sont orthogonales. Ce que traduisent les zigzags des itérés.

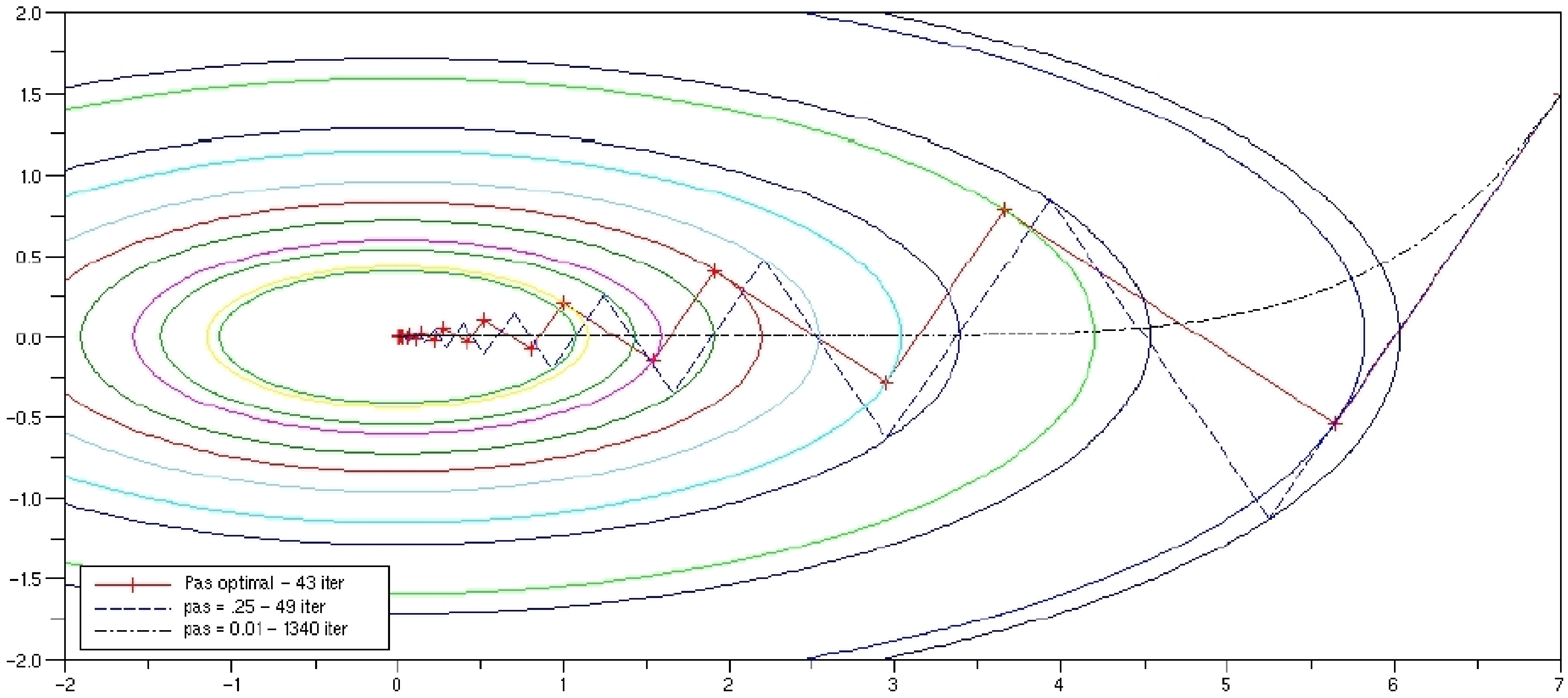
Étude graphique de la méthode du gradient





Exemple: Minimiser $f(x,y) = \frac{1}{2}x^2 + \frac{7}{2}y^2$

en utilisant les algorithmes de gradient à pas fixe (t=0,25) et à pas optimal, à partir de x_0=(7;1,5)



itérations de gradient pas fixe et pas optimal à 10^{-5} près à partir de x_0 =(7;1,5)

k	$f(x_k, y_k)$	$\|\nabla f(x_k, y_k)\|_2$	s_k	x_k	y_k
0	32.375	10.547512	–	7	1.5
1	16.925373	7.9786973	0.1940299	5.641791	-0.5373134
2	8.8484403	6.5973298	0.3513514	3.6595401	0.7841872
3	4.6258889	3.5448339	0.1940299	2.9494801	-0.2809029
4	2.4183752	3.4490276	0.3513514	1.9131763	0.4099663
5	1.2643059	1.8532089	0.1940299	1.541963	-0.1468536
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
40	$1.751e{-}10$	2.9343653×10^{-5}	0.3513514	1.63×10^{-5}	0.35×10^{-5}
41	$9.155e - 11$	1.5725775×10^{-5}	0.1940299	1.31×10^{-5}	-0.12×10^{-5}
42	$4.786e - 11$	1.536522×10^{-5}	0.3513514	0.85×10^{-5}	0.18×10^{-5}
43	$2.502e - 11$	0.8292768×10^{-5}	0.1940299	0.69×10^{-5}	0.07×10^{-5}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
76	$1.268e - 20$	0.2523886×10^{-9}	0.3513514	0.14×10^{-9}	0.03×10^{-9}
77	$6.630e - 21$	0.1303840×10^{-9}	0.1940299	0.11×10^{-9}	-0.01×10^{-9}
78	$3.466e - 21$	0.1303840×10^{-9}	0.3513514	0.72×10^{-10}	0.16×10^{-10}
79	$1.812e - 21$	$0.6989278 \times 10^{-10}$	0.1940299	0.58×10^{-10}	-0.05×10^{-10}

TABLE 2.1 – Itérations de la méthode de plus profonde descente. Le critère d'optimalité est satisfait en 43 itérations pour une précision $\varepsilon = 10^{-5}$ et en 79 itérations si $\varepsilon = 10^{-10}$.

C- Algorithme du gradient conjugué

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

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

ce qui rend la convergence de l'algorithme du gradient à pas optimal très lente pour des courbes de niveaux très applaties. Par conséquent, nous allons définir d'autres directions de descentes qui respectent mieux la géométrie du problème.

- Considérons en premier lieu le cas d'une fonctionnelle quadratique :

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

où A est symétrique définie positive et $(u, v) = u^t v$ le produit scalaire dans $\mathbf{R}^n$.

- Il est clair que $\nabla f(x) = Ax - b$ ce qui implique que le minimum de cette fonctionnelle est atteint pour le point $\overline{x}$ qui annule le grade vérifiant $A\overline{x} = b$. De ce fait minimiser f revient à résoudre le système linéaire $Ax = b$.

Définition *: (directions conjuguées)*

Un ensemble de vecteurs $\{d_0, d_1, d_2, ..., d_k\}$ *est dit A-conjuguée si*

$$(Ad_i, d_j) = 0 \qquad i \neq j$$

Autrement dit, les d_i *sont perpendiculaires entre eux par rapport au produit scalaire induit par la matrice* A *:* $< u, v >_A = (Au, v)$.

Principe:

l'algorithme du gradient conjugué construit deux suites de vecteurs : les itérés $\{x_0; x_1; x_2, ..., x_k\}$ et les directions de descentes $\{d_0; d_1; ... ; d_k\}$ qui vérifient les propriétés suivantes:.

- la suite des gradients $\{\nabla f(x_0); \nabla f(x_1); ...; \nabla f(x_k)\}$ forme un système orthogonal

(ce qui n'est pas le cas dans la méthode de gradient où seulement deux gradient consécutifs sont orthogonaux).

- la suite des directions de descentes $\{d_0; d_1; ... ; d_k\}$ forme un système A-conjuguées

- Grâce à ces propriétés l'algorithme du gradient conjugué converge en au plus n itérations.

Construction des itérés *xk* et des directions conjuguées *dk* :

On note $\boldsymbol{r_k = -\nabla f(x_k) = b - A\, x_k}$ le vecteur résidu

- le calcul de $\boldsymbol{x_{k+1}}$ est suivant le procédé itératif

$$\boldsymbol{x_0} \quad \text{donné}, \quad \boldsymbol{x_{k+1} = x_k + t_k\, d_k}$$

où $\boldsymbol{t_k}$ est le pas de descente qui réalise le minimum de f selon la direction $\boldsymbol{x_k + t_k\, d_k}$

- Mise à jour du résidu $\boldsymbol{r_k}$

$$\begin{aligned} \boldsymbol{r_{k+1}} &= \boldsymbol{b - A\, x_{k+1}} \\ &= \boldsymbol{b - A\,(x_k + t_k\, d_k)} \\ &= \boldsymbol{r_k - t_k\, A\, d_k} \end{aligned}$$

- la direction conjuguée $\boldsymbol{d_{k+1}}$ est combinaison linéaire entre la direction $\boldsymbol{d_k}$ et le résidu $\boldsymbol{r_k}$:

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

où $\boldsymbol{\beta_k}$ est choisi tel que l'*A*-conjugaison entre $\boldsymbol{d_k}$ et $\boldsymbol{d_{k+1}}$ soit vérifiée.

Calcul des coefficients t_k et β_k :

- Calcul de t_k : on pose $\varphi(t) = f(x_k + td_k)$ et on a $\varphi'(t) = <\nabla f(x_k + td_k); d_k>$.

  Or $t_k = \min_{t>0} \varphi(t)$.

  Ce qui implique que
  $$\begin{aligned}\varphi'(t_k) &= <\nabla f(x_{k+1}); d_k> \\ &= -<r_{k+1}; d_k> \\ &= -<r_k - t_k A d_k; d_k> = 0\end{aligned}$$

  On obtient ainsi que
  $$t_k = \frac{<r_k; d_k>}{<Ad_k; d_k>} = \frac{<b - Ax_k; d_k>}{<Ad_k; d_k>}$$

- Calcul de β_k : on veut que $d_{k+1} \perp_A d_k$.

  On a alors $0 = <Ad_k, d_{k+1}> = <Ad_k, r_{k+1} + \beta_k d_k>$

  ce qui implique que $$\beta_k = -\frac{<Ad_k, r_{k+1}>}{<Ad_k, d_k>}$$

L'algorithme du gradient conjugué est généralisé pour des fonctions non quadratiques (cas non linéaire) :

Algorithme : Gradient conjugué (Fletcher-Reeves 1964)

Données : f de classe C^2, $x_0 \in \mathrm{IR}^n$, $\varepsilon > 0$

(1) initialisation : $d_0 = -\nabla f(x_0)$; $k = 0$

(2) **tant que** $\|\nabla f(x_k)\| > \varepsilon$ **faire**

(a) Choisir $t_k > 0$ solution de $\min_{t>0} f(x_k + t d_k)$

(b) Mise à jour $x_{k+1} = x_k + t_k d_k$

(c) $\beta_k = \dfrac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$

nouvelle direction : $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$

(d) $k = k + 1$

fin tant que

Remarque :

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

3-3-Méthode de Newton

- La méthode de Newton pour la résolution de l'équation non linéaire dans IR de type $F(x) = 0$

$$
\begin{cases} x_0 \quad \text{donné} \\ x_{k+1} = x_k - \dfrac{F(x_k)}{F'(x_k)} \end{cases}
$$

- En optimisation non linéaire sans contraintes, on cherche les solutions de l'équation $\nabla f(x) = 0$ (les points critiques de la fonction f à minimiser)
- Par analogie on a $F(x)= \nabla f(x)$ et donc on retrouve

Si f est de classe C^2 et $\nabla^2 f(x)$ est inversible, une itération de l'algorithme de Newton s'écrit

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

où $d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$ est appelée direction de Newton.

Remarque: *La méthode ne doit être jamais appliquée en utilisant l'inversion de la matrice Hessienne.*

Algorithme de Newton

(1) initialisation : choisir x_0 proche de x

(2) **tant que** "test d'arrêt" non satisfait **faire**

(a) calculer la direction d_k solution du système

$$[\nabla^2 f(x_k)]\, d_k = -\nabla f(x_k)$$

(b) $x_{k+1} = x_k + d_k$; $k = k + 1$

fin tant que

Remarque : *La méthode de Newton est un algorithme de descente à pas fixe égal à 1.*