Optimisation

Nicolas Pécheux
nicolas.pecheux@limsi.fr

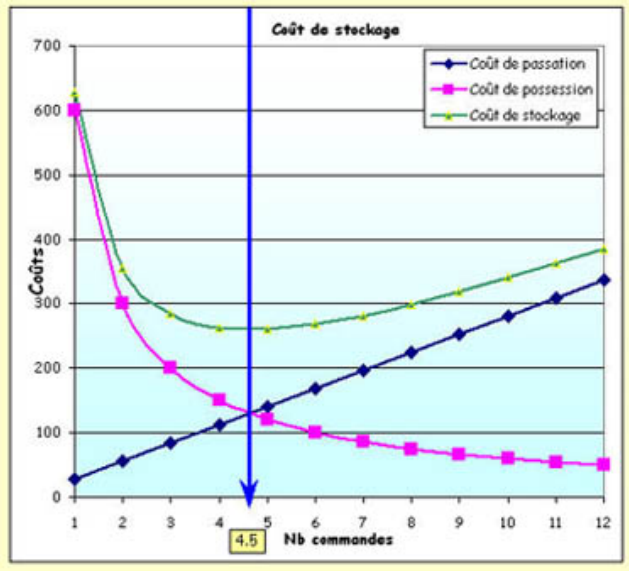
Université Paris Sud — LIMSI

27 mars 2013

Première partie I

Qu'est ce que l'optimisation ?

Introduction



N	Coût de passation	Coût de possession	Coût de stockage
1	28 €	600.00 €	628 €
2	56 €	300.00 €	356 €
3	84 €	200.00 €	284 €
4	112 €	150.00 €	262 €
5	140 €	120.00 €	260 €
6	168 €	100.00 €	268 €
7	196 €	85.71 €	282 €
8	224 €	75.00 €	299 €
9	252 €	66.67 €	319 €
10	280 €	60.00 €	340 €
11	308 €	54.55 €	363 €
12	336 €	50.00 €	386 €

- But : minimiser une fonction (sous contraintes)

$$\begin{array}{ll} \underset{x}{\text{minimiser}} & f_0(x) \\ \text{s.c.} & x \in \mathcal{C} \end{array}$$

- Exemples ?

Exemple d'applications

Économie

- Minimiser le risque d'un placement financier
- Maximiser le profil...

Ingénierie

- Minimiser la taille de circuits électroniques
- Minimiser la quantité de réactifs d'une équation chimique

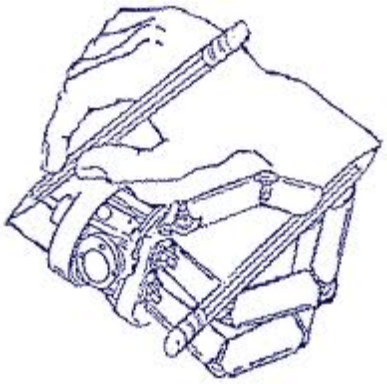
Apprentissage

- Minimiser le nombre d'erreur d'une méthode

Et (presque) tout le reste !

Deuxième partie II

Formalisation



But $\underset{x}{\text{minimiser}} \quad f_0(x)$

$$\text{s.c.} \quad x \in \mathcal{C}$$

- f_0 fonction de coût (ou de perte)
- $\mathcal{C}$ est l'ensemble des contraintes

- **Solution optimale** : x^* a la plus petit valeur de f_0 parmi les points qui satisfont les contraintes

rem : $\Leftrightarrow \max_x -f_0(x)$

Combinatoire vs *numérique*

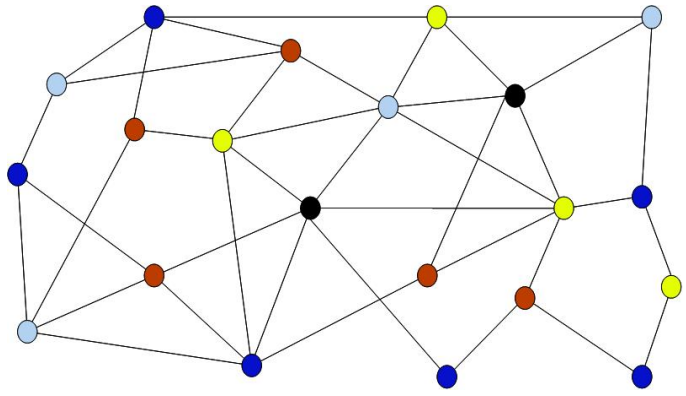
Optimisation combinatoire

- $x \in \mathcal{S}$ discret (e.g. $\mathbb{N}$)

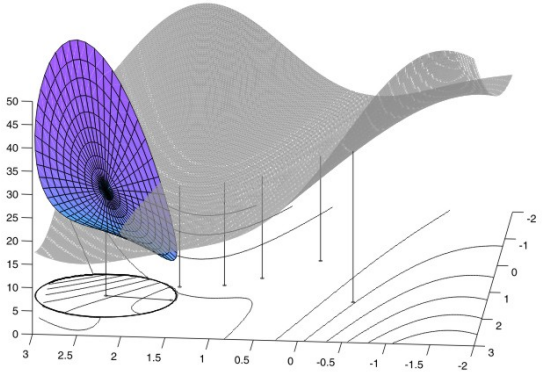
Exemples :

- Plus court chemin dans un graph
- K-moyennes !

Souvent très compliqué (NP-dur)



Numérique (mathématique)



- $x \in \mathbb{R}^n$

$$\begin{aligned} \underset{x}{\text{minimiser}} \quad & f_0(x) \\ \text{s.c.} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{aligned}$$

Linéaire vs *non linéaire*

Programmation linéaire

$$\begin{aligned} \underset{x}{\text{minimiser}} \quad & c^T x \\ \text{s.c.} \quad & a_i^T x \leq b_i, \quad i = 1, \ldots, m \\ & c_j^T x = d_j, \quad j = 1, \ldots, p \end{aligned}$$

- Pas de solution analytique
- Mais méthodes efficaces pour la résolution
- Faciles à reconnaitre

$\Rightarrow$ Technologie (sauf si n est très grand)

Linéaire vs *non linéaire*

Non-linéaire

- Très difficile à résoudre
- En général besoin de compromis (approximation ou très longue attente)

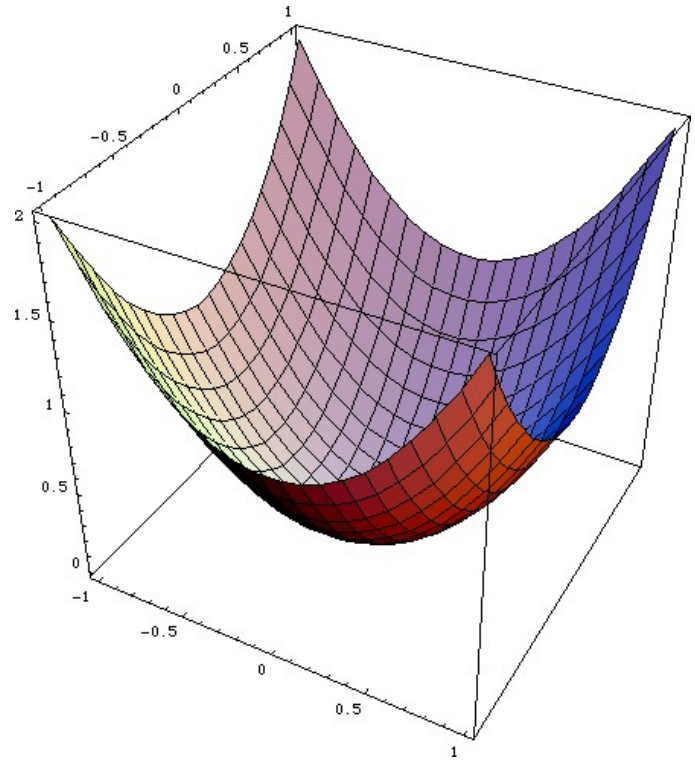
Sauf...

Convexe vs *non convexe*

> *In fact the great watershed in optimization isn't between linearity and nonlinearity, ut convexity and nonconvexity.*
>
> *[Rockafellar 93]*

Optimisation convexe

$$\begin{aligned} \underset{x}{\text{minimiser}} \quad & f_0(x) \\ \text{s.c.} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & c_j^T x = d_j, \quad j = 1, \ldots, p \end{aligned}$$



avec f_i convexes, $i = 0, \ldots, m$
(et donc C convexe)

- Rappel : $f : \mathbb{R}^n \to \mathbb{R}$ est convexe ssi
  $\forall x, y \in \mathbb{R}^n$, $\alpha, \beta \in \mathbb{R}^+$, $\alpha + \beta = 1$

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y)$$

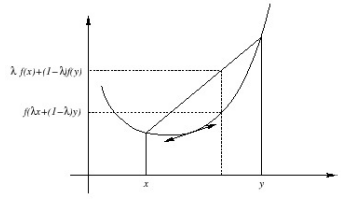
Convexe vs *non convexe*

Problèmes convexes

- Sorte de généralisation de la linéarité
- Pas de solution analytique
- Mais méthodes efficaces pour la résolution
- De très nombreux problèmes peuvent s'y ramener
- Presque une technologie

Propriétés

- minimum local = minimum global
- la "courbe" est au dessus de ses "tangentes"
- stricte convexité → encore meilleures propriétés

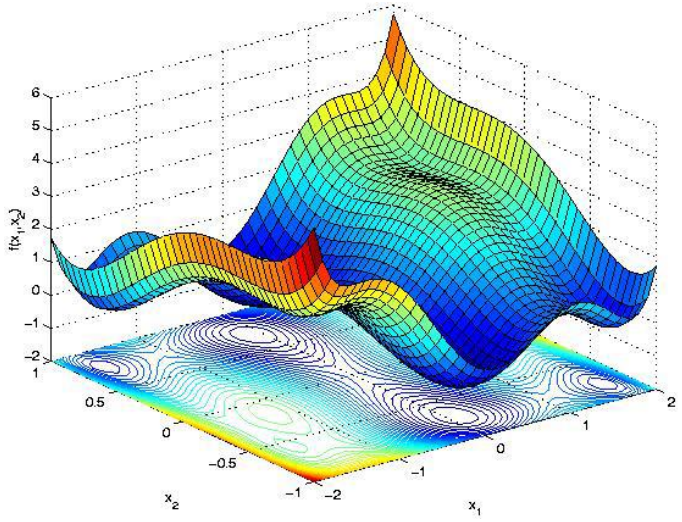


Mais...

- Difficiles à reconnaitre
- Techniques pour mettre un problème sous forme convexe

Convexe vs *non convexe*

Problèmes non convexes



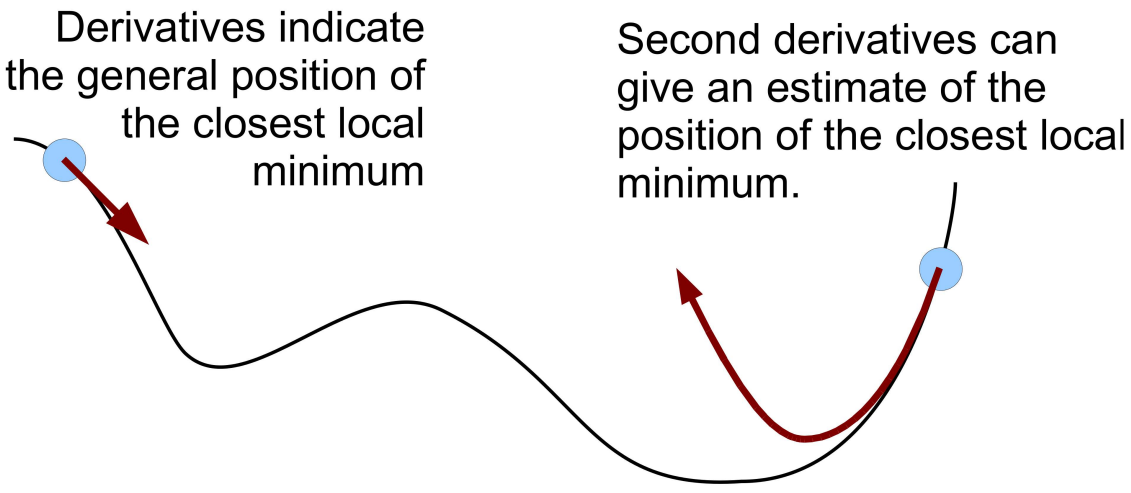
- Minimums locaux, point selles, plateaux, ravins, etc.
- Bons et mauvais minimums locaux
- Résultats dépendent de détails subtils

- Compromis : approximation/temps de calcul
- Optimisation convexe peut jouer un rôle : initialisation, approximation, bornes, sous-problèmes convexes

Exemples

- Apprentissage non supervisées (e.g. K-moyennes !)
- Réseaux de neurones

Différentiable vs non différentiable



- Sans régularité, pas d'information locale
- Les dérivées successives peuvent ne pas exister
- Les dérivées successives peuvent être trop longues à calculer

Condition d'Euler

Si f_0 est C_1, alors $\nabla f_0(x^*) = 0$

Si f_0 est convexe, alors $\nabla f_0(x) = 0 \Rightarrow x = x^*$

Rappel : gradient et hessienne

Idée principale : approximation locale d'une fonction (lisse)

1er ordre : par une fonction linéaire (gradient)

$$f(x+h) \approx f(x) + \nabla f(x)^T h \quad \text{où} \quad \nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)$$

$f(x+h) \geq f(x) + \nabla f(x)^T h$ dans le cas convexe

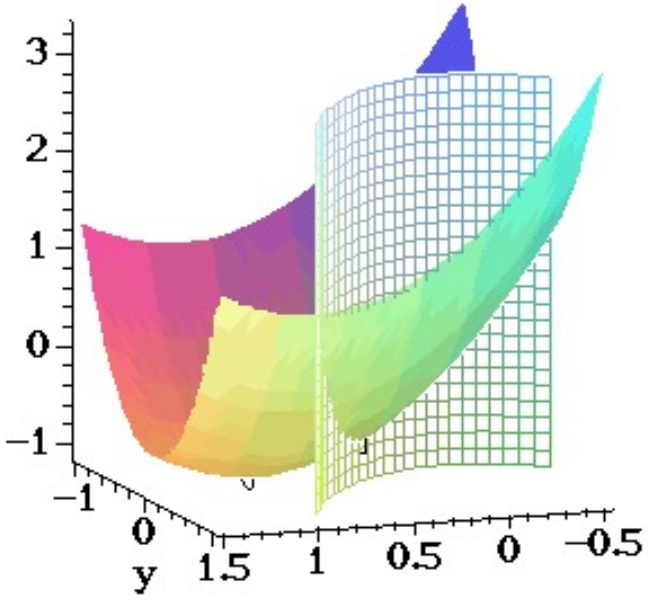
2e ordre : par une fonction quadratique (hessienne)

$$f(x+h) \approx f(x) + \nabla f(x)^T h + \frac{1}{2} h^T \nabla^2 f(x) h \quad \text{où} \quad \nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

Avec contraintes vs sans contraintes

Avec contraintes

- On n'a plus nécessairement $\nabla f_0(x^*) = 0$
- Méthodes de résolutions assez différentes
- Utilisent souvent sous-problème sans contraintes



- Mots clef : théorie lagrangienne, conditions KKT, méthode des points intérieurs, projection
- Pas aujourd'hui

Troisième partie III

Méthodes

Méthodes de descentes - intuition

Comment atteindre le minimum ? Comment le reconnaître ?
$\rightarrow$ Fonction convexes : ouf !

Attention :
On ne “voit” pas la surface ! On ne dispose que d’information locale. Comment faire ?

Méthodes de descentes

Idée

Partir de quelque part, choisir une direction de descente Δx et l'emprunter (un peu). Puis recommencer !

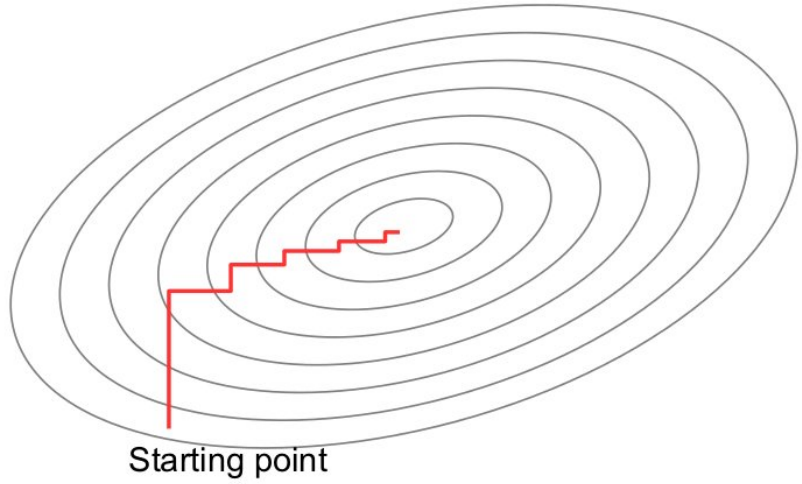
Require : a starting point $x \in$ **dom** f

1 : **repeat**
2 : Determine a descent direction Δx
3 : *Line search*. Choose a step size $t > 0$
4 : *Update*. $x := x + t\Delta x$
5 : **until** stopping criterion is satisfied.

$\rightarrow$ méthodes itératives

Descente par coordonnées

On choisit successivement comme directions les vecteurs de la base (i.e. coordonnée par coordonnée)



- Tendance au zig-zag
- Pas besoin de calculer le gradient
- On peut choisir les coordonnées avec de l'aléa
- On peut grouper certaines coordonnées
- On peut changer de base avant (e.g. ACP)

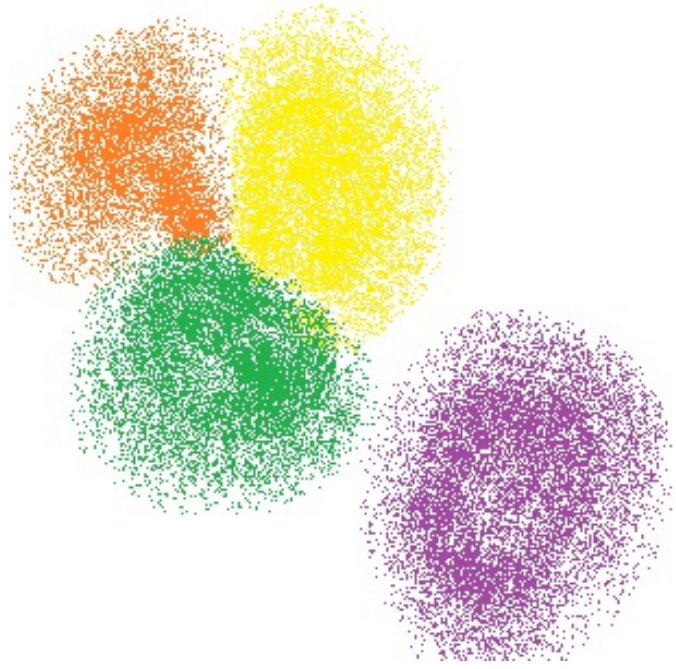
Exemple ?

Rappel : K-moyennes

But : minimiser la qualité d'un partitionnement

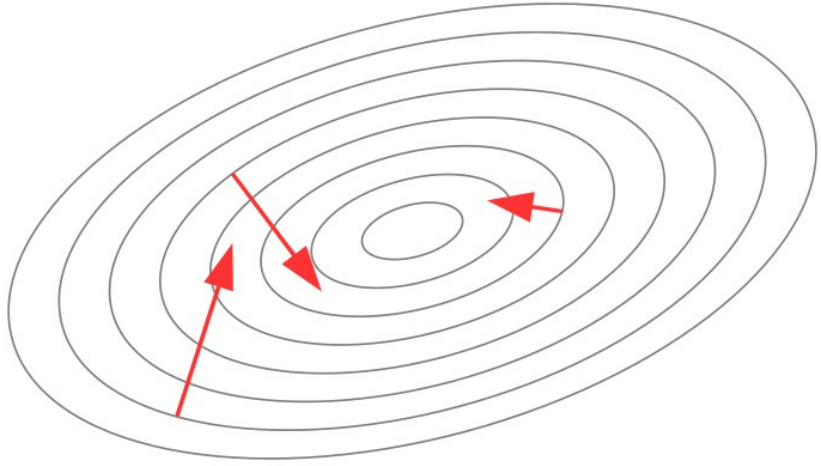
$$\mathcal{E}(\mu, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k \|x_i - \mu_k\|^2$$

- μ_k est la moyenne du cluster k
- $z_i^k = 1$ si x_i est dans le cluster k et 0 sinon
- Exercice : montrer en quoi c'est une descente de gradient par coordonnées
- Question : est-ce que c'est *vraiment* une descente de gradient ?



Descente de gradient

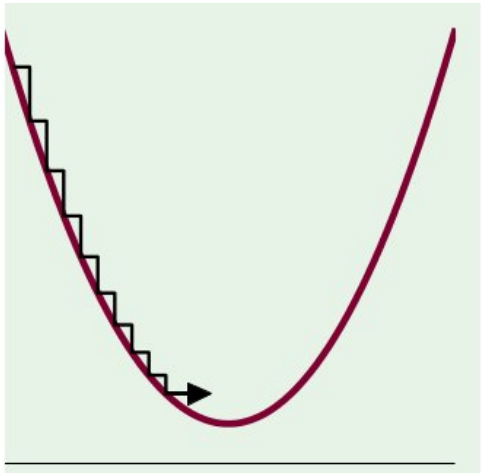
Idée naturelle : choisir la direction de plus forte pente

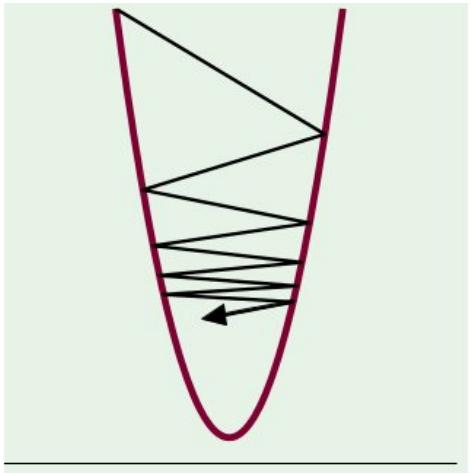


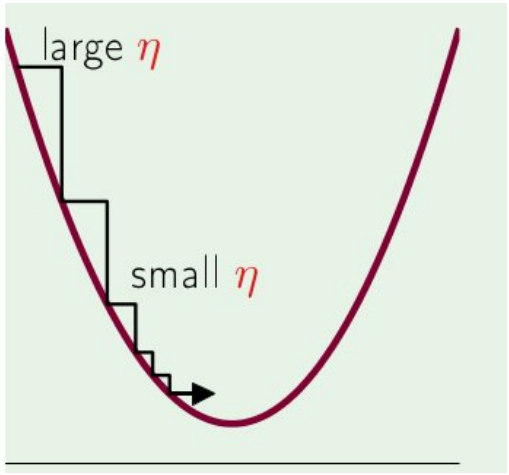
- Celle-ci est donné par $\delta x = -\nabla f(x)$
- On peut prendre comme critère d'arrêt $\|\nabla f(x)\|_2 \leq \epsilon$

Choix du pas

- Compromis : petits pas (approximation valable localement) mais alors on avance tout doucement
- Remarque : Δx n'est pas forcément normalisé (e.g. gradient)





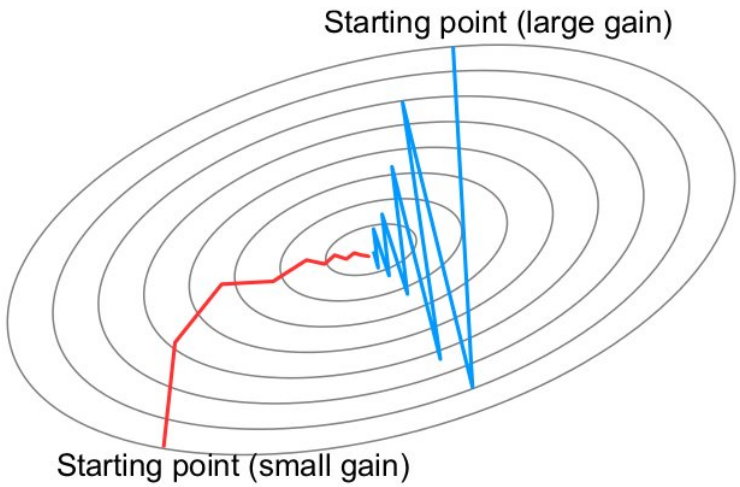


- Gradient à pas constant
- Gradient à pas optimal
- De toutes façon, c'est juste *un* pas, pourquoi s'embêter ?
  $\Rightarrow$ prendre pas approché (*line search*)
- De très nombreuses méthodes : *Backtracking*, méthode de Brent, méthode de Wolfe

Analyse

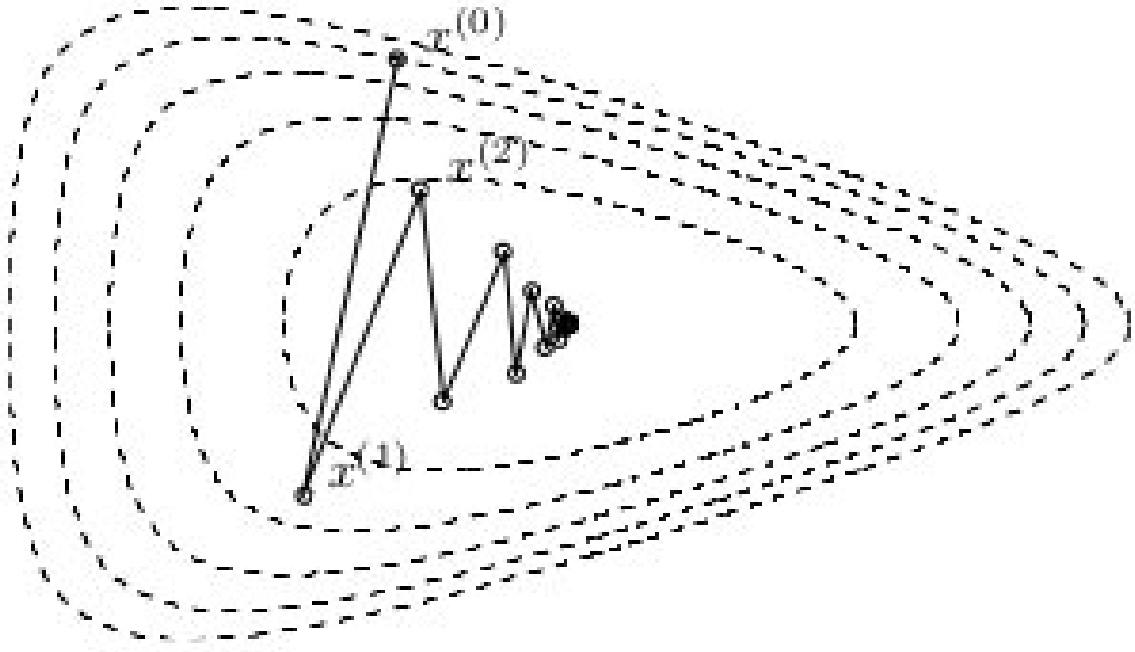
- Les méthode de gradient convergent "lentement" $\rightarrow$ zig-zag
- Vitesse dépend beaucoup de l'anisotropie des courbes de niveau

$$\textbf{cond}(C) = \frac{L_{max}^2}{L_{min}^2} = \frac{\lambda_{min}}{\lambda_{max}}$$



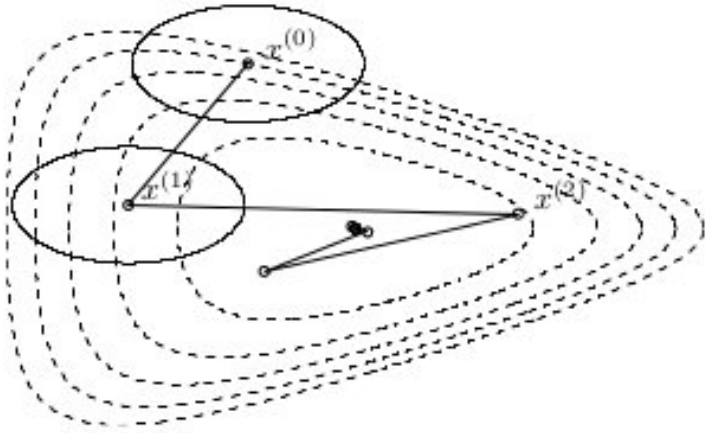
- Résultat de convergence (gradient à pas variable) si $\eta \leq 1m/M$
- C.f. TP

Solution : changer la géométrie ?

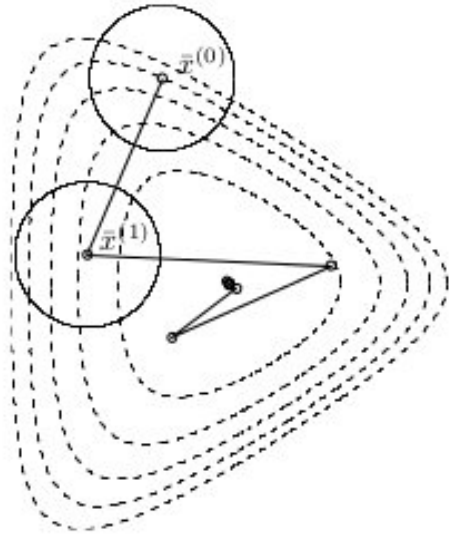


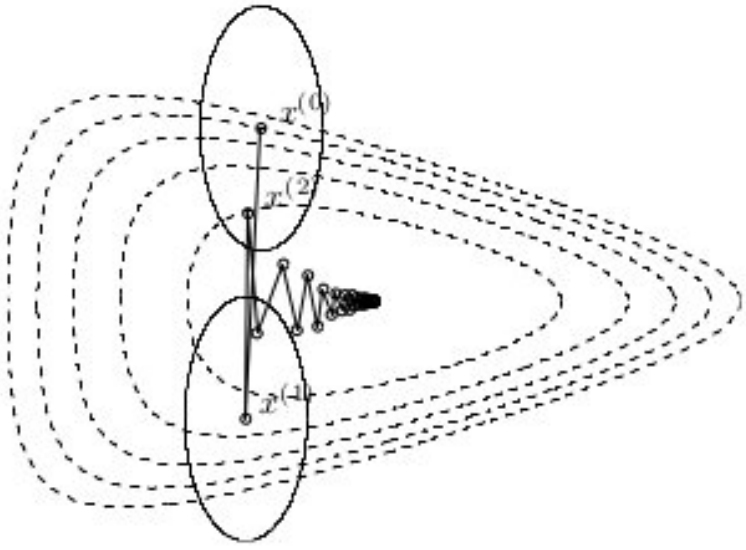
- i.e. faire un changement de base (matrices de passage)

Mais quelle géométrie prendre ?

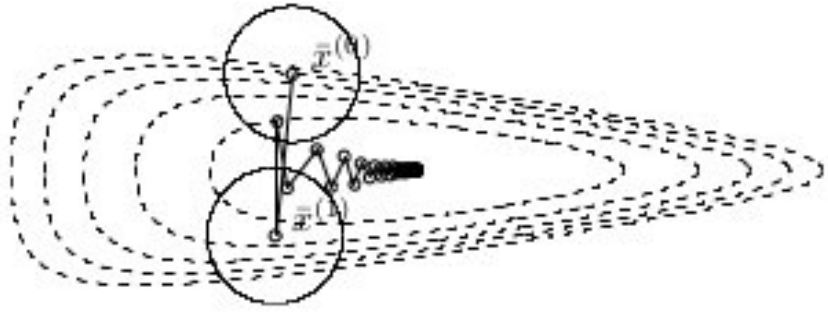


$\Rightarrow$

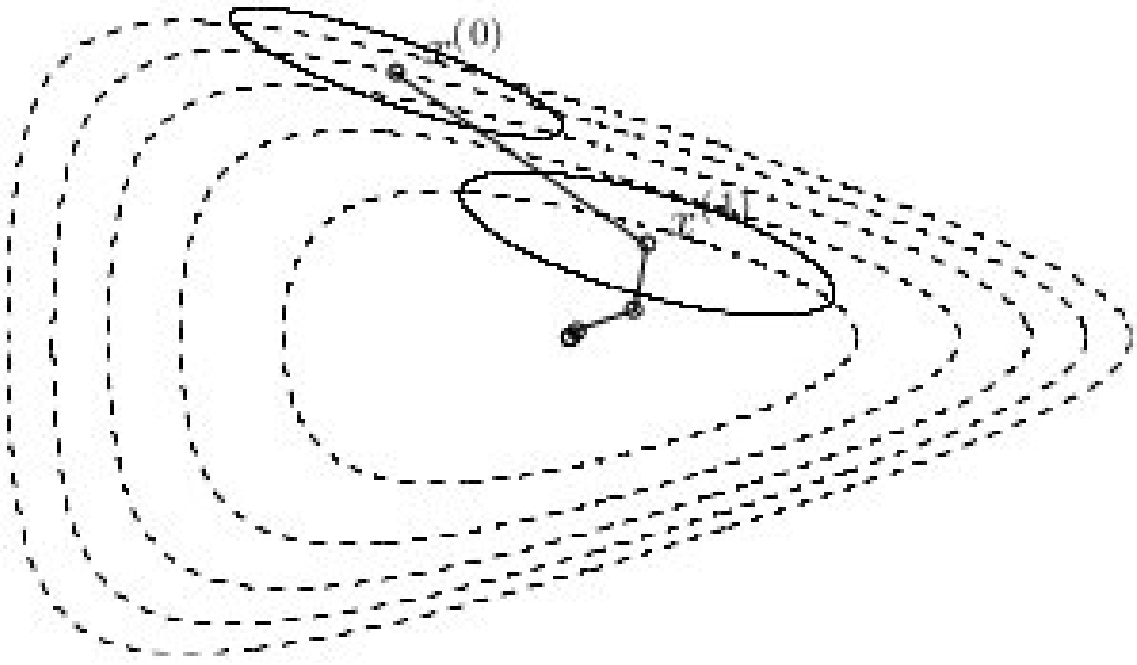




$\Rightarrow$

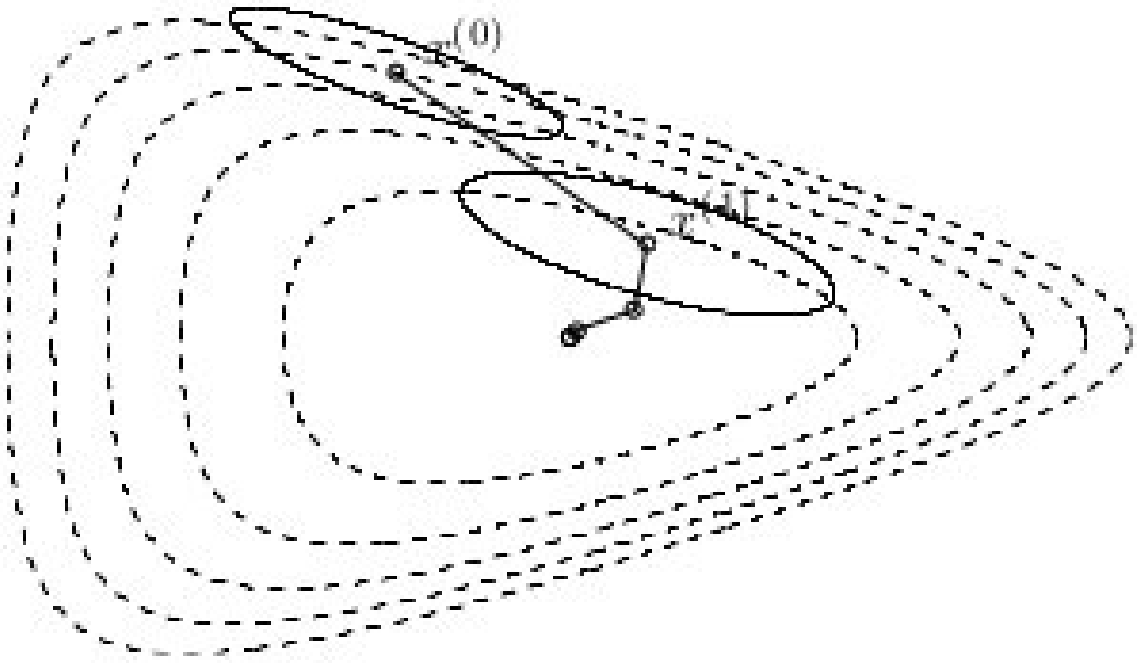


La “meilleure” géométrie est donnée par la hessienne



- Correspond au changement de coordonnées $u' = P^{1/2}u$ où $P = \nabla^2 f(x)$
- On retrouve l’idée d’approximation au deuxième ordre
- C’est...

La “meilleure” géométrie est donnée par la hessienne



- Correspond au changement de coordonnées $u' = P^{1/2}u$ où $P = \nabla^2 f(x)$
- On retrouve l’idée d’approximation au deuxième ordre
- C’est la méthode de Newton

La méthode de Newton

$$\Delta x_{newton} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

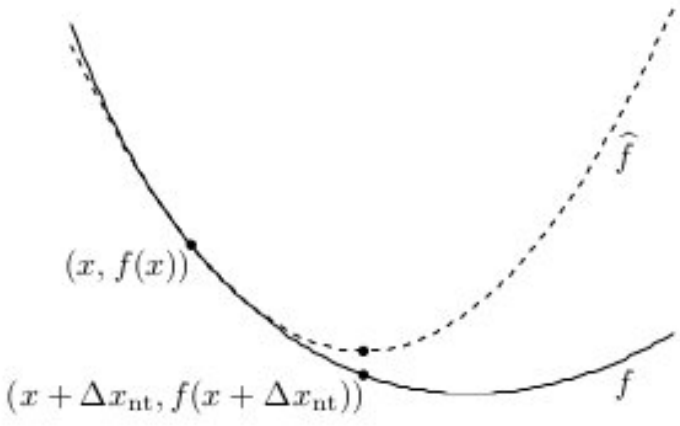
Exercice :

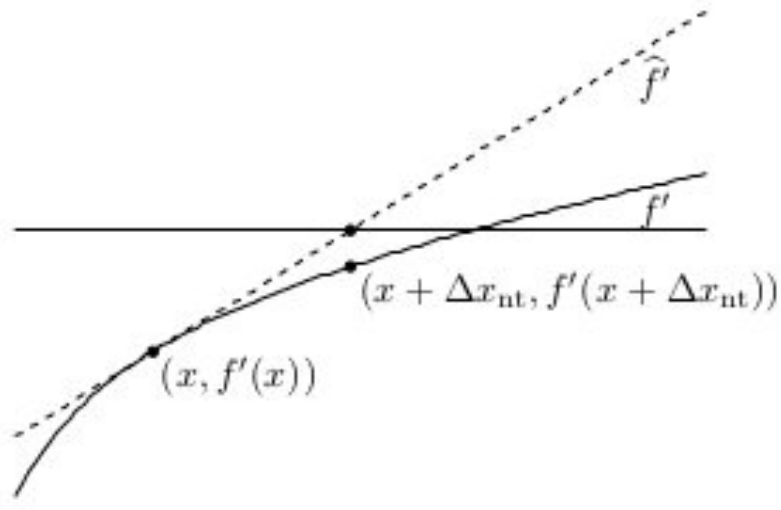
- $x + \Delta x_{newton}$ minimise l'approximation au second ordre :

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

- $x + \Delta x_{newton}$ est solution de l'équation d'Euler linéarisée :

$$\hat{f}(x+h) = 0$$





La méthode de Newton

- Convergence très rapide (quelques itérations)
- Exacte pour une fonction quadratique
- Très bonne propriétés si $\nabla^2 f$ est strictement convexe

Mais...

- Problèmes si $\nabla^2 f$ n'est pas inversible !
- Calculer $\nabla^2 f$ est très cher (souvent impossible)

Méthodes quasi-Newton

- Évitent les inconvénients de la méthode de Newton
- Mais au final se comportent à peu près de la même manière

- Méthode du gradient conjugé

Quatrième partie IV

Optimisation et apprentissage

Rappel - Fonction de perte

Données : $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{1 \leq i \leq n}$

Fonction de perte :

$$L(\mathbf{x}, y; \mathbf{w}) = \ell(h_{\mathbf{w}}(\mathbf{x}_i), y_i)$$

où

- $h_{\mathbf{w}}$ est un classifieurs paramétré par $\mathbf{w}$
- $\ell(y_{pred}, y_{vrai})$ le coût associé à la prédiction de y_{pred} sachant que la bonne étiquette est y_{vrai}

Risque régularisé :

$$f(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{x}_i, y_i; \mathbf{w}) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

- λ est le coefficient de régularization

Rappel - Régression linéaire

- $\ell(y_{pred}, y_{vrai}) = (y_{pred} - y_{vrai})^2$

$$f(\mathbf{w}) = ??$$

Rappel - Régression linéaire

- $\ell(y_{pred}, y_{vrai}) = (y_{pred} - y_{vrai})^2$

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i - y_i)^2 = \|X\mathbf{w} - Y\|_2^2$$

Rappel - Régression linéaire

- $\ell(y_{pred}, y_{vrai}) = (y_{pred} - y_{vrai})^2$

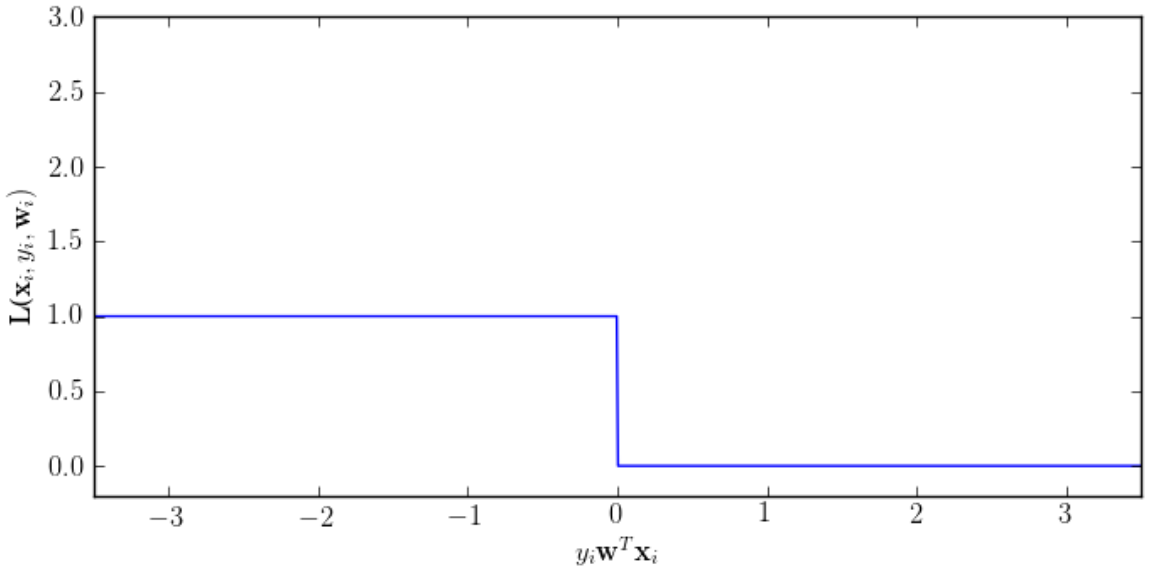
$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i - y_i)^2 = \|X\mathbf{w} - Y\|_2^2$$

- La solution est $w^* = (X^TX)^{-1}X^TY$
- Solution analytique (très rare !)
- Pas besoin des algorithmes précédents
- Souvent une sous-routine de nombreuses méthodes
- C'est en gros une technologie

Classification binaire

- $\ell(y_{pred}, y_{vrai}) = \delta_{\{y_{pred} - y_{vrai}\}}$

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \delta_{\{\text{sgn}(\mathbf{w}^T\mathbf{x}_i)=y_i\}} = \frac{1}{n}\sum_{i=1}^{n} \delta_{\{\text{sgn}(y_i\mathbf{w}^T\mathbf{x}_i)<0\}}$$

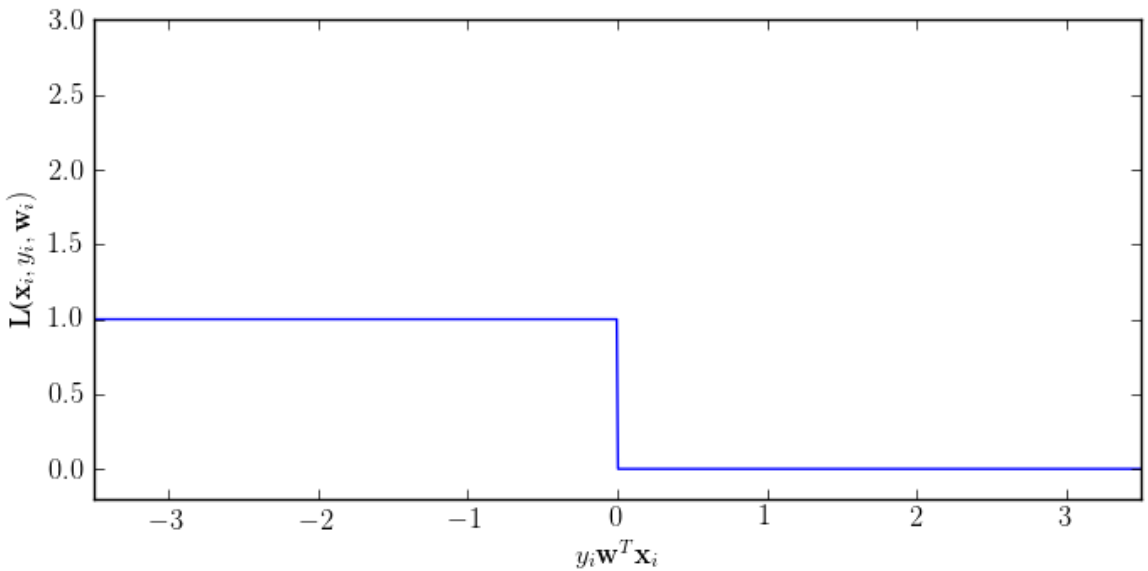


- Problème ?

Classification binaire

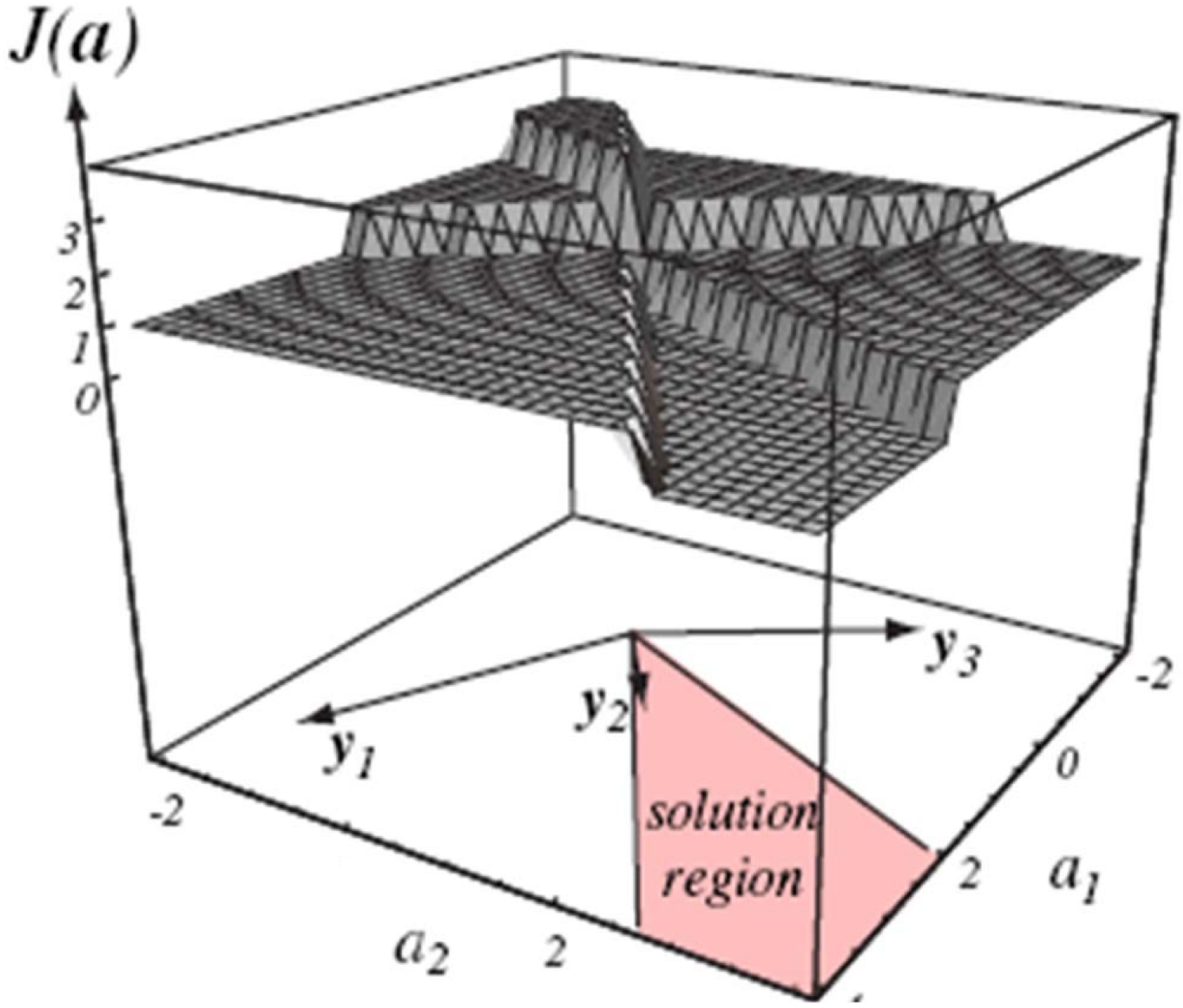
- $\ell(y_{pred}, y_{vrai}) = \delta_{\{y_{pred} - y_{vrai}\}}$

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \delta_{\{\text{sgn}(\mathbf{w}^T\mathbf{x}_i)=y_i\}} = \frac{1}{n}\sum_{i=1}^{n} \delta_{\{\text{sgn}(y_i\mathbf{w}^T\mathbf{x}_i)<0\}}$$



- Problème ?
- f n'est pas convexe !
- Très difficile à minimiser

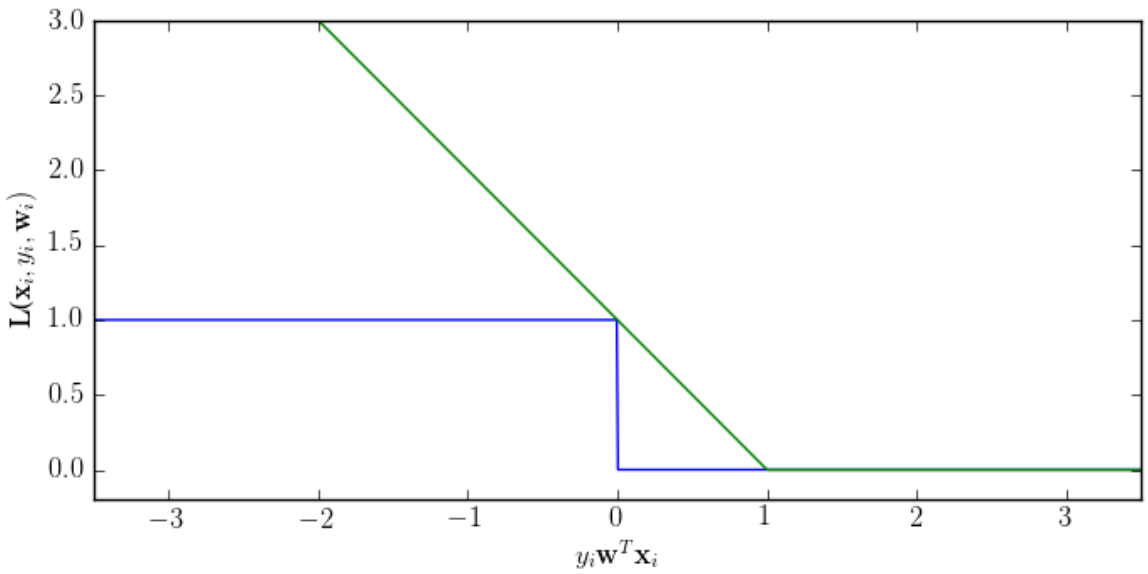
En 3D



Convexification du risque

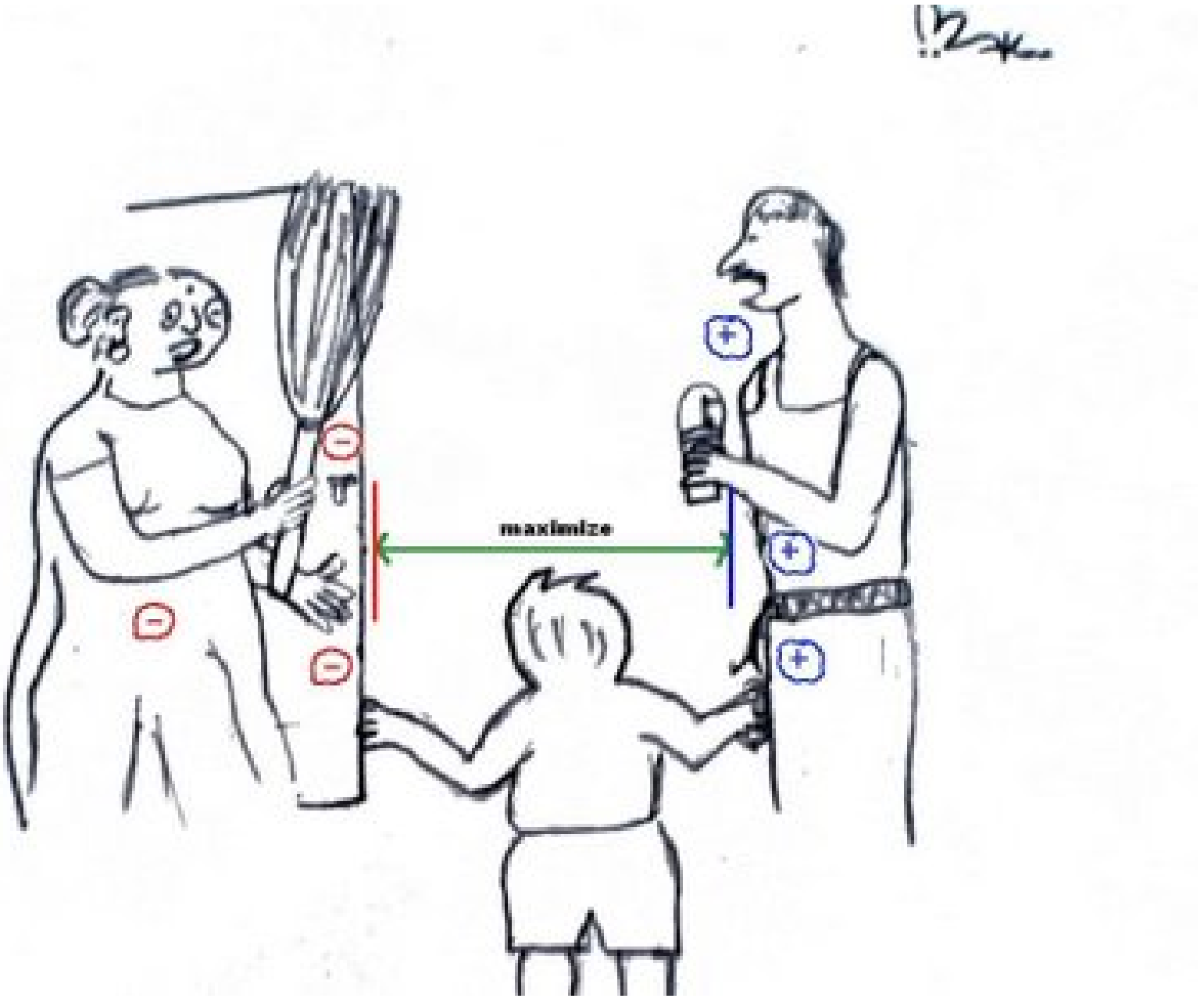
- Comment faire quand un problème n'est pas convexe ?
- L'approcher par un problème convexe !
- Ici par une borne supérieure

Exemple :



Vous avez reconnu ?

Rappel : SVM

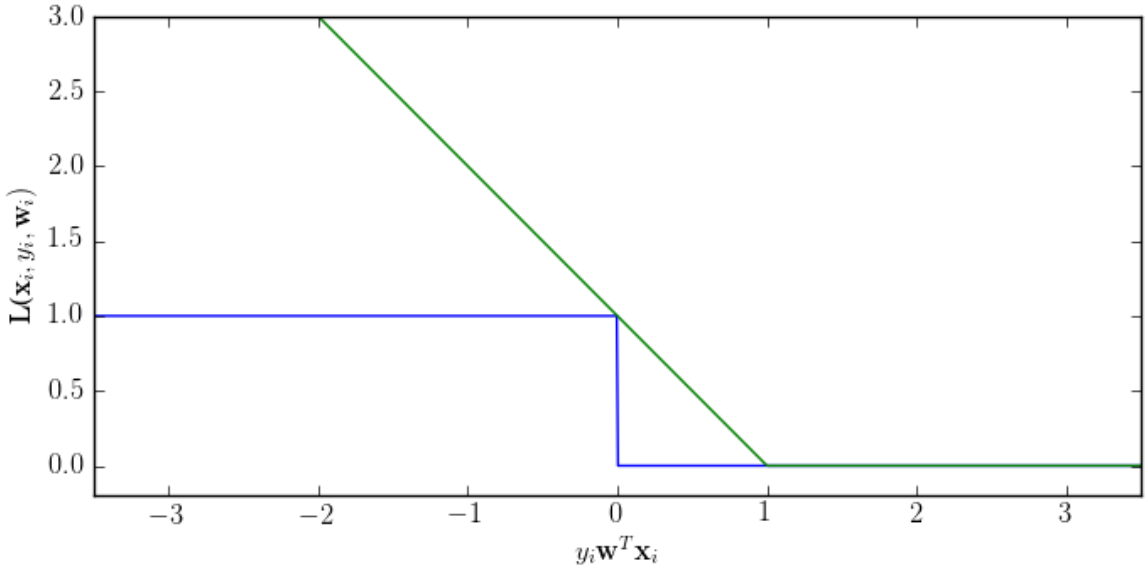


Rappel : SVM

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\phi(y_i\mathbf{w}^T\mathbf{x}_i) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

où $\phi(u) = max(1-u, 0)$

- Exercice : retrouver cette formulation à partir du cours précédent



- On peut montrer que l'on "approche bien" le problème

Cinquième partie V

Large échelle ?

Rappel : optimisation et apprentissage

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{x}_i, y_i; \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

- Problème : n peut être très grand
- E.g. quelques milliards
- E.g. quelques milliers par secondes

- Les exemples sont très redondants (sinon rien à apprendre)
- Doubler le nombre d'exemple apporte un peu d'information
- À quoi ça sert au tout début de la descente ?

Descente de gradient : classique ou stochastique ?

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{x}_i, y_i; \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Descente de gradient classique

Itérer : $\mathbf{w} := \mathbf{w} - \eta\left(\frac{1}{n}\sum_{i=1}^{n} \nabla L(\mathbf{x}_i, y_i; \mathbf{w}) + \lambda \mathbf{w}\right)$

- Tout les exemples sont nécessaires pour chaque itération (le vrai gradient)

Descente de gradient classique

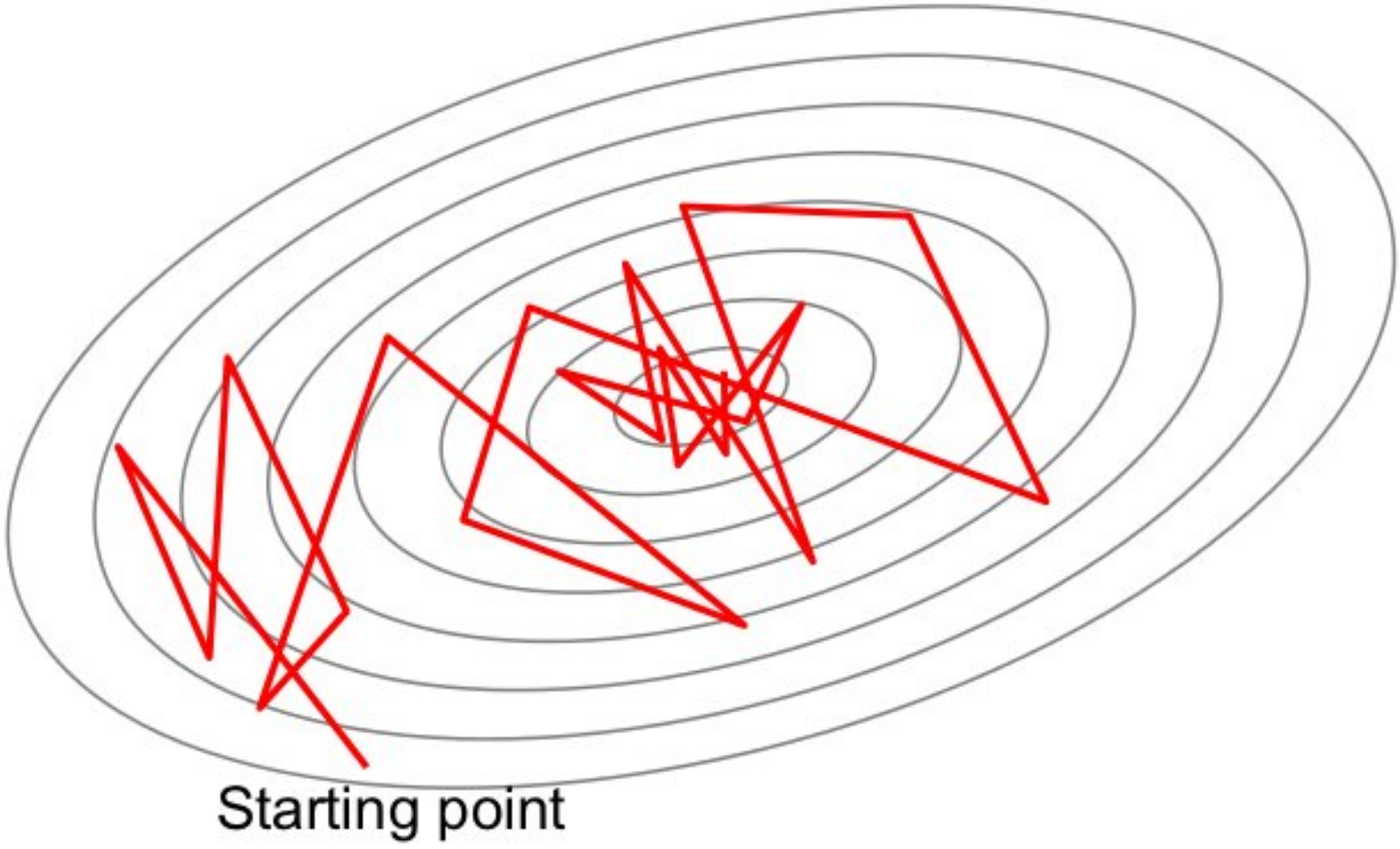
Itérer :

1. Choisir au hasart $(\mathbf{x}_i, y_i)$
2. $\mathbf{w} := \mathbf{w} - \eta\left(\nabla L(\mathbf{x}_i, y_i; \mathbf{w}) + \lambda \mathbf{w}\right)$

- Une itération = un exemple

Descente de gradient stochastique

- À chaque descente le gradient n'est pas bon (très mal estimé)
- Mais il l'est en moyenne



En pratique

En pratique :

- On fait décroitre η à chaque itération
- Beaucoup plus rapide (e.g. 1.4s vs. 6h)
- $\approx$ Indépendant du nombre d'exemples

Rappel : le perceptron !

Exercice : montrer que l'algorithme du perceptron vu en cours est une descente de gradient stochastique. Quelle est la fonction de perte ?

Sixième partie VI

À vous...

Un problème quadratique I

$$f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2) \qquad (\gamma > 0)$$

On prendra $x^{(0)} = (\gamma, 1)$ et $\gamma \in [10^{-3}, 1, 10]$

1. La fonction f est-elle convexe (strictement ?), différentiable ? Quel est son gradient, sa matrice hessienne ? En quel(s) point(s) est elle minimisée ? L'équation d'Euler est-elle vérifiée ?
2. En utilisant `mpl_toolkits.mplot3d.Axes3D`, visualizez cette fonction en 3D. Tracez ses courbes de niveaux à l'aide de `matplotlib.pyplot.contour`

Un problème quadratique II

3. Implémentez la méthode de descente par coordonnées. Pour quel(s) pas a-t-on convergence ? Dessinez les itérations successives sur le graphe des lignes de niveaux. Visualisez l'erreur en fonction du nombre d'itérations. Quelle est l'influence de γ ?
4. De même avec la méthode de gradient à pas constant.
5. De même avec la méthode de descente à pas optimal. Pour ce dernier, exprimer également analytiquement $x^{(k)}$ en fonction de k et de γ. Conclusion ?
6. Que donne la méthode de Newton ici ?