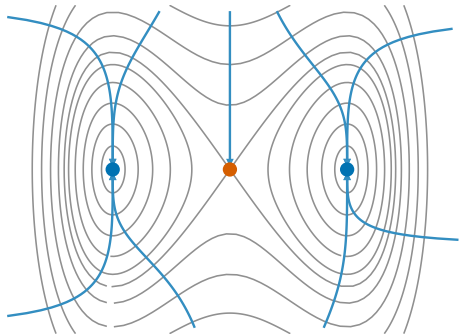
Équations différentielles et optimisation
L3 Mathématiques · Travaux pratiques

Optimisation : les bases

Minima, points critiques, conditionnement

Introduction au TP 6



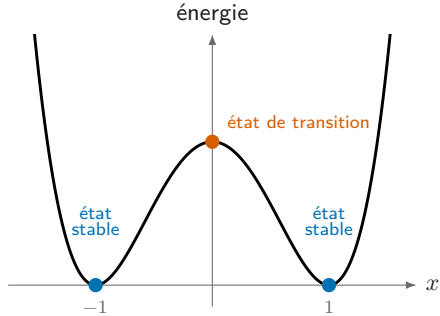
Santiago Toro Oquendo
Université d'Artois · Semestre 1, 2026–2027

Optimiser : un problème partout

- **Chimie, physique** : un système au repos minimise son *énergie potentielle*. Les états stables d'une molécule sont des minima ; pour passer de l'un à l'autre, elle franchit un *col*, l'état de transition.
- **Données** : ajuster une droite $y \approx at + b$ à des mesures (t_i, y_i), c'est minimiser $\mathrm{E}(a,b) = \sum_i (y_i - at_i - b)^2$: c'est le principe de l'apprentissage automatique (TP 9).
- **Ingénierie** : minimiser un coût, une consommation…

Dans ce cours

Minimiser $f : \mathbb{R}^n \to \mathbb{R}$, *régulière* ($\mathcal{C}^1$ ou $\mathcal{C}^2$), *sans contraintes*.



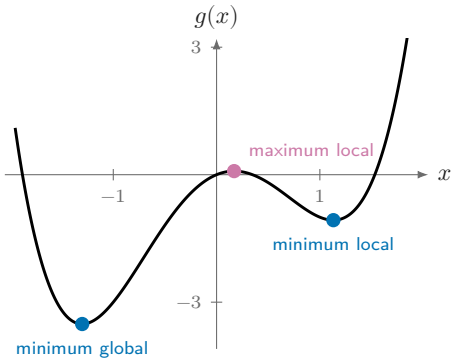
coupe $y = 0$ du « double puits »
$f_3(x,y) = (x^2-1)^2 + y^2$ du TP

Le problème et le vocabulaire

On cherche $\min_{x \in \mathbb{R}^n} f(x)$. Un point x^* est un

- **minimum local** si $f(x^*) \leqslant f(x)$ pour tout x *proche* de x^* ;
- **minimum global** si $f(x^*) \leqslant f(x)$ pour *tout* $x \in \mathbb{R}^n$;
- minimum **strict** si l'inégalité est stricte pour $x \neq x^*$.

On dit aussi que x^* est un **minimiseur**, et que $f(x^*)$ est la valeur minimale. Maximiser f, c'est minimiser $-f$.



$g(x) = x^4 - 3x^2 + x$: minimum global en $x \approx -1{,}3008$, minimum local en $x \approx 1{,}1309$ (on la retrouvera au TP 7)

Existe-t-il un minimum ?

Théorème (Weierstrass)

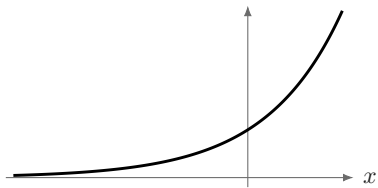
Une fonction continue sur un compact (fermé borné) de $\mathbb{R}^n$ y atteint son minimum.

Corollaire : sur $\mathbb{R}^n$ tout entier

Si f est continue et **coercive**, c'est-à-dire $f(x) \to +\infty$ quand $\|x\| \to \infty$, alors f a un minimum global.
Idée : $\mathrm{K} = \{x : f(x) \leqslant f(x_0)\}$ est fermé et borné ; le minimum sur K est le minimum sur $\mathbb{R}^n$.

Contre-exemples

- $f(x) = \mathrm{e}^x$: minorée par 0, mais 0 n'est jamais atteint ;
- $f(x) = x^3$: pas minorée ;
- $f(x, y) = x^2$: des minimiseurs, mais une infinité (toute la droite $x = 0$).



$\inf \mathrm{e}^x = 0$ n'est jamais atteint

Rappels : gradient, hessienne, formule de Taylor

Soit $f : \mathbb{R}^n \to \mathbb{R}$ de classe $\mathcal{C}^2$ et $x \in \mathbb{R}^n$.

- **Gradient** : $\nabla f(x) = (\partial_1 f(x), \dots, \partial_n f(x)) \in \mathbb{R}^n$.
- **Hessienne** : $\nabla^2 f(x) = \left(\partial_i \partial_j f(x)\right)_{i,j}$, matrice $n \times n$ **symétrique** (Schwarz).
- **Taylor à l'ordre 2** :

$$f(x+h) = f(x) + \nabla f(x)^\top h + \tfrac{1}{2}\, h^\top \nabla^2 f(x)\, h + o(\|h\|^2)$$

Exemple : $f(x, y) = x^2 + 3y^2$

$$\nabla f(x,y) = \begin{pmatrix} 2x \\ 6y \end{pmatrix}, \qquad \nabla^2 f(x,y) = \begin{pmatrix} 2 & 0 \\ 0 & 6 \end{pmatrix}.$$

En Python (TP 6)

Un point :

```
x = np.array([1.0, -2.0])
```

`f(x)` : un nombre ;
`grad_f(x)` : forme `(2,)` ;
`hess_f(x)` : forme `(2, 2)`.

Le gradient est la direction de *plus forte pente* de f ; il est *perpendiculaire* aux lignes de niveau.

Rappel : les matrices symétriques

Théorème spectral

Une matrice symétrique $\mathrm{H} \in \mathcal{M}_n(\mathbb{R})$ a n valeurs propres **réelles** $\lambda_1 \leqslant \cdots \leqslant \lambda_n$, et une base **orthonormée** de vecteurs propres $(v_1, \ldots, v_n)$. Pour tout $h \in \mathbb{R}^n$:

$$\lambda_1 \|h\|^2 \;\leqslant\; h^\top \mathrm{H} h \;\leqslant\; \lambda_n \|h\|^2 \qquad \text{(égalité pour } h = v_1\text{, resp. } h = v_n\text{)}.$$

- H **définie positive** : $\lambda_1 > 0$, c'est-à-dire $h^\top \mathrm{H} h > 0$ pour $h \neq 0$;
- **semi-définie positive** : $\lambda_1 \geqslant 0$;
- **indéfinie** : une valeur propre > 0 et une < 0.

En Python : `np.linalg.eigvalsh(H)` renvoie les valeurs propres d'une matrice *symétrique*, triées.

En dimension 2 seulement : $\det \mathrm{H} > 0$ et $\operatorname{tr} \mathrm{H} > 0$ $\iff$ H définie positive. En dimension n, on regarde *toutes* les valeurs propres.

Condition d'ordre 1 : le gradient s'annule

Théorème

Si f est $\mathcal{C}^1$ et si x^* est un minimum local de f, alors $\nabla f(x^*) = 0$.

Preuve. Pour une direction d, la fonction d'une variable $\varphi(t) = f(x^* + td)$ a un minimum local en $t = 0$, donc

$$0 = \varphi'(0) = \nabla f(x^*)^\top d.$$

C'est vrai pour tout d : prendre $d = \nabla f(x^*)$ donne $\|\nabla f(x^*)\|^2 = 0$. □

Un point où $\nabla f(x) = 0$ est un **point critique** (ou stationnaire).

La réciproque est fausse

- $f(x) = x^3$ en 0 : point critique, ni minimum ni maximum ;
- $f(x, y) = x^2 - y^2$ en $(0, 0)$: minimum dans la direction x, maximum dans la direction y : un **point selle**.

Les points critiques sont les *candidats* : il faut ensuite les trier.

Conditions d'ordre 2 : la hessienne

Condition nécessaire

Si x^* est un minimum local de f ($\mathcal{C}^2$), alors $\nabla^2 f(x^*)$ est **semi-définie positive**.

Idée : $\varphi''(0) = d^\top \nabla^2 f(x^*)\, d \geqslant 0$ pour toute direction d.

Condition suffisante

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

Preuve. Notons $\lambda_1 > 0$ la plus petite valeur propre de $\mathrm{H} = \nabla^2 f(x^*)$. Par Taylor et le théorème spectral,

$$\begin{aligned} f(x^* + h) - f(x^*) &= \tfrac{1}{2}\, h^\top \mathrm{H} h + o(\|h\|^2) \\ &\geqslant \tfrac{\lambda_1}{2}\, \|h\|^2 + o(\|h\|^2) \\ &> 0 \quad \text{si } h \neq 0 \text{ est petit.} \end{aligned}$$

□

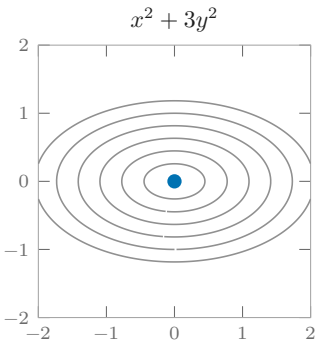
Classer un point critique

En un point critique x^*, on regarde le **signe des valeurs propres** de $\mathrm{H} = \nabla^2 f(x^*)$:

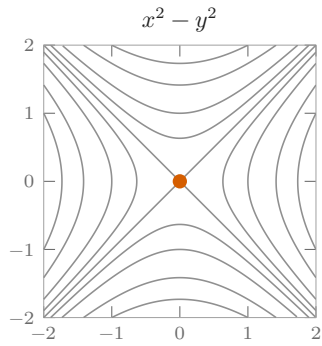
valeurs propres de $\nabla^2 f(x^*)$	nature de x^*	exemple en $(0,0)$
toutes > 0	**minimum local strict**	$x^2 + 3y^2$
toutes < 0	**maximum local strict**	$-x^2 - y^2$
une > 0 et une < 0	**point selle**	$x^2 - y^2$
une nulle, les autres de même signe	**on ne peut pas conclure**	voir plus loin

Point selle : le long d'un vecteur propre v de valeur propre $\lambda < 0$, $f(x^* + tv) = f(x^*) + \frac{\lambda}{2} t^2 \|v\|^2 + o(t^2) < f(x^*)$; le long d'un vecteur propre de valeur propre > 0, f augmente. Sur ordinateur, une valeur propre de 10^{-17} est un zéro arrondi : on compare à une tolérance `tol`.

Lignes de niveau



minimum : des ellipses emboîtées



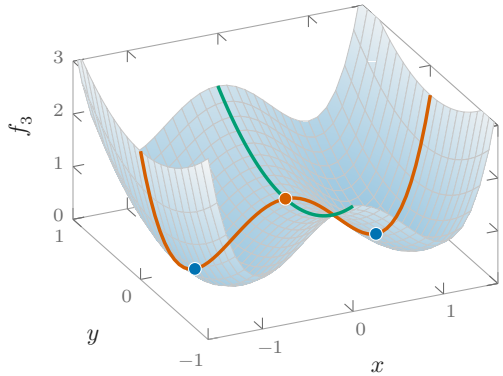
point selle : la ligne $f = 0$ se croise en X

Une **ligne de niveau** est une courbe $\{x : f(x) = c\}$, comme sur une carte topographique.

- près d'un minimum strict, f ressemble à sa partie quadratique de Taylor : les lignes de niveau sont presque des **ellipses** ;
- le gradient est perpendiculaire aux lignes de niveau.

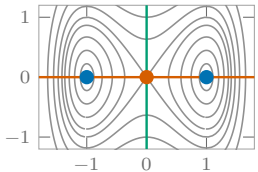
Au TP : `tracer_contours(f, xlim, ylim)`, dans la boîte à outils.

Exercice 4 : le double puits f_3 en relief



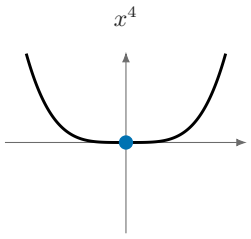
$f_3(x, y) = (x^2 - 1)^2 + y^2$:

- **deux minima** $(\pm 1, 0)$: le fond des cuvettes $(f_3 = 0)$;
- **un point selle** $(0, 0)$: le *col* $(f_3 = 1)$, maximum le long de la courbe orange $(y = 0)$, minimum le long de la verte $(x = 0)$;
- **aucun maximum** : f_3 n'est pas majorée.

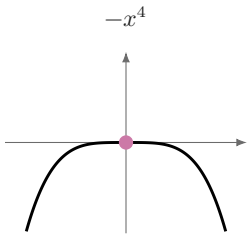


vu de dessus : les lignes de niveau (exercice 4)

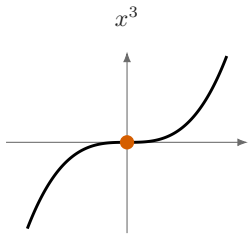
Quand la hessienne ne dit rien



minimum global strict



maximum global strict



ni l'un ni l'autre

Dans les trois cas, $f'(0) = 0$ *et* $f''(0) = 0$: la formule de Taylor à l'ordre 2 ne voit qu'une fonction *plate*. La hessienne (ici f'') ne permet pas de conclure : il faut **étudier f elle-même** (termes d'ordre supérieur, inégalités, contre-exemple sur une droite…).

Au TP : la même situation en dimension 2, et un test numérique… qui n'est pas une preuve.

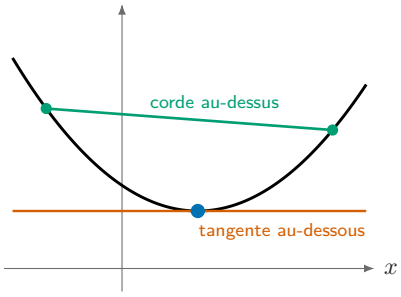
Convexité : quand local = global

f est **convexe** si, pour tous x, y et $\theta \in [0,1]$,

$$f(\theta x + (1-\theta)y) \leqslant \theta f(x) + (1-\theta) f(y)$$

(la corde est au-dessus du graphe).

- *(admis)* pour f de classe $\mathcal{C}^2$: f convexe $\iff$ $\nabla^2 f(x)$ semi-définie positive pour *tout* x ;
- f est **fortement convexe** s'il existe $\mu > 0$ tel que $h^\top \nabla^2 f(x)\, h \geqslant \mu \|h\|^2$ pour tous x, h : toutes les valeurs propres de $\nabla^2 f(x)$ sont $\geqslant \mu$ (noté $\nabla^2 f(x) \succeq \mu \mathrm{I}$). Alors f a un **unique** minimiseur global *(admis)*.



Théorème *(admis)*

Si f est convexe, **tout point critique est un minimum global**. *Idée* : le graphe est au-dessus de ses tangentes : $f(y) \geqslant f(x^*) + \nabla f(x^*)^\top (y - x^*) = f(x^*)$.

Les fonctions quadratiques : gradient et hessienne

Soit A une matrice **symétrique** $n \times n$ ($\mathrm{A}^\top = \mathrm{A}$), $b \in \mathbb{R}^n$, et $f(x) = \frac{1}{2}\, x^\top \mathrm{A} x - b^\top x$. On développe $f(x+h)$:

$$\begin{aligned} f(x+h) &= \tfrac{1}{2}\,(x+h)^\top \mathrm{A}\,(x+h) - b^\top (x+h) \\ &= \underbrace{\tfrac{1}{2}\, x^\top \mathrm{A} x - b^\top x}_{f(x)} + \underbrace{\tfrac{1}{2}\, x^\top \mathrm{A} h + \tfrac{1}{2}\, h^\top \mathrm{A} x - b^\top h}_{\text{termes de degré 1 en } h} + \underbrace{\tfrac{1}{2}\, h^\top \mathrm{A} h}_{\text{degré 2}} \end{aligned}$$

- $h^\top \mathrm{A} x$ est un **nombre**, donc égal à son transposé : $h^\top \mathrm{A} x = (h^\top \mathrm{A} x)^\top = x^\top \mathrm{A}^\top h = x^\top \mathrm{A} h$. Les termes de degré 1 valent donc $x^\top \mathrm{A} h - b^\top h = (\mathrm{A} x - b)^\top h$.
- D'où, **exactement** (sans reste) : $f(x+h) = f(x) + (\mathrm{A} x - b)^\top h + \frac{1}{2}\, h^\top \mathrm{A} h$.
- On reconnaît Taylor à l'ordre 2, $f(x+h) = f(x) + \nabla f(x)^\top h + \frac{1}{2}\, h^\top \nabla^2 f(x)\, h + o(\|h\|^2)$; par identification :

$$\boxed{\nabla f(x) = \mathrm{A} x - b, \qquad \nabla^2 f(x) = \mathrm{A}.}$$

Vérification à la main en dimension 2. Avec $\mathrm{A} = \left(\begin{smallmatrix} a & c \\ c & d \end{smallmatrix}\right)$ et $b = (b_1, b_2)$:
$f(x,y) = \frac{1}{2}\,(ax^2 + 2c\,xy + dy^2) - b_1 x - b_2 y$, donc $\partial_x f = ax + cy - b_1$ et $\partial_y f = cx + dy - b_2$: c'est $\mathrm{A} x - b$. Les dérivées secondes valent a, c, c et d : c'est A.

Quadratique : le cas où A est définie positive

Un point critique x^* vérifie $\nabla f(x^*) = \mathrm{A}x^* - b = 0$, c'est-à-dire $\mathrm{A}x^* = b$. Le développement exact, écrit en x^*, donne pour tout h :

$$f(x^* + h) = f(x^*) + \underbrace{(\mathrm{A}x^* - b)^\top h}_{=0} + \tfrac{1}{2}\, h^\top \mathrm{A} h = f(x^*) + \tfrac{1}{2}\, h^\top \mathrm{A} h.$$

Si A est **définie positive** (plus petite valeur propre $\lambda_1 > 0$), alors $h^\top \mathrm{A} h \geqslant \lambda_1 \|h\|^2$, donc

$$f(x^* + h) \;\geqslant\; f(x^*) + \tfrac{\lambda_1}{2}\,\|h\|^2 \;>\; f(x^*) \qquad \text{pour tout } h \neq 0.$$

Conclusion

Si A est définie positive, elle est inversible : $\mathrm{A}x = b$ a une unique solution x^*, et c'est l'**unique minimiseur global** de f. (Ici f est fortement convexe, avec $\mu = \lambda_1$.)

En Python : `np.linalg.solve(A, b)`. On ne calcule *jamais* A^{-1} : c'est plus cher et moins précis.
Pourquoi elles comptent : près d'un minimum x^*, *toute* fonction $\mathcal{C}^2$ ressemble (Taylor) à $f(x^*) + \frac{1}{2}(x - x^*)^\top \nabla^2 f(x^*)(x - x^*)$: ce qu'on comprend sur les quadratiques vaut près de n'importe quel minimum.

Une quadratique a-t-elle un minimiseur ?

la matrice A	minimiseur de $\frac{1}{2}x^\top \mathrm{A}x - b^\top x$	pourquoi
définie positive	✓ un seul : la solution de $\mathrm{A}x = b$	f fortement convexe
une valeur propre $\lambda < 0$	✗ aucun	$f(tu) \to -\infty$, u vecteur propre
semi-définie positive, singulière	selon b : aucun, ou une infinité	f le long du noyau

Dans le dernier cas, si v engendre le noyau de A, alors $f(x + tv) = f(x) - t\, b^\top v$: tout dépend de $b^\top v$. Au TP, vous classerez cinq problèmes avec `np.linalg.eigvalsh` et `np.linalg.eigh`.

Les erreurs d'arrondi, encore

Les valeurs propres « nulles » sortent de l'ordinateur sous la forme 10^{-16} ou -10^{-17} : on les compare toujours à une tolérance.

Le conditionnement

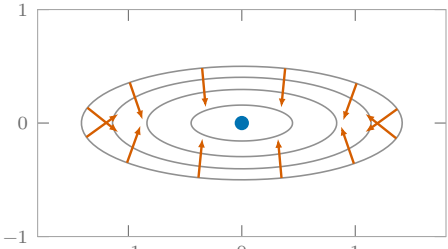
Pour A définie positive, le **conditionnement** est

$$\kappa(\mathrm{A}) = \frac{\lambda_{\max}}{\lambda_{\min}} \geqslant 1.$$

La ligne de niveau $\frac{1}{2}x^\top \mathrm{A}x = c$ est une **ellipse** d'axes portés par les vecteurs propres, de demi-axes $\sqrt{2c/\lambda_i}$: des cercles si $\kappa = 1$, des ellipses très **allongées** si κ est grand.

> **Pourquoi c'est important**
>
> Le gradient est perpendiculaire aux lignes de niveau : quand κ est grand, $-\nabla f$ ne pointe *pas* vers le minimum. Au TP 7, la descente de gradient va zigzaguer.

$\frac{1}{2}(x^2 + 8\,y^2)$, $\kappa = 8$



en orange : la direction $-\nabla f$ sur la ligne de niveau $c = 1$

Vérifier un gradient : les différences finies

Avec e_i le i-ème vecteur de la base canonique :

$$\text{avant :} \quad \partial_i f(x) \approx \frac{f(x + he_i) - f(x)}{h} \quad \mathrm{O}(h)$$

$$\text{centrées :} \quad \partial_i f(x) \approx \frac{f(x + he_i) - f(x - he_i)}{2h} \quad \mathrm{O}(h^2)$$

(Taylor : les termes en h^2 se compensent dans la formule centrée ; le détail est page suivante.)

Mais l'ordinateur arrondit f à $\varepsilon \approx 10^{-16}$ près, et cette erreur est divisée par h :

$$\text{erreur} \approx Ch^2 + \frac{\varepsilon\,|f|}{h} \quad \text{(centrées)}.$$

Le bon pas

Différences centrées : $h \approx \varepsilon^{1/3} \approx 10^{-5}$; avant : $h \approx \sqrt{\varepsilon} \approx 10^{-8}$. Trop petit, c'est pire !

Pour $\sin$ en $x = 1$ et $h = 10^{-6}$, l'erreur centrée est d'environ $3 \cdot 10^{-11}$.

Le réflexe

Avant d'utiliser un gradient écrit à la main, on le compare à `grad_num(f, x)`. Une erreur dans le gradient fait échouer tous les algorithmes.

Différences finies : d'où viennent les erreurs

1. Troncature. Taylor–Lagrange, pour f de classe $\mathcal{C}^3$ d'une variable (pour $\partial_i f$: l'appliquer à $t \mapsto f(x + te_i)$). Avant : $f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\xi)$, donc

$$\frac{f(x+h) - f(x)}{h} - f'(x) = \tfrac{h}{2}\, f''(\xi) = \mathrm{O}(h).$$

Centrées : $f(x \pm h) = f(x) \pm hf'(x) + \frac{h^2}{2} f''(x) \pm \frac{h^3}{6} f'''(\xi_\pm)$; les termes en h^2 s'annulent dans la différence :

$$\frac{f(x+h) - f(x-h)}{2h} - f'(x) = \tfrac{h^2}{12}\big(f'''(\xi_+) + f'''(\xi_-)\big),$$

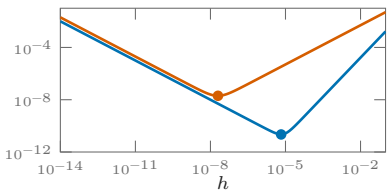
c'est un $\mathrm{O}(h^2)$.

2. Arrondi. L'ordinateur calcule $f(x)\,(1+\delta)$, avec $|\delta| \leqslant \varepsilon \approx 10^{-16}$: l'erreur sur la différence de deux valeurs peut atteindre $2\varepsilon|f|$, et on la divise par h (ou $2h$).

3. Le total, avec $\mathrm{M}_k \geqslant |f^{(k)}|$ près de x :

avant : $\mathrm{E}(h) \approx \frac{\mathrm{M}_2}{2}\, h + \frac{2\varepsilon|f|}{h}$

centrées : $\mathrm{E}(h) \approx \frac{\mathrm{M}_3}{6}\, h^2 + \frac{\varepsilon|f|}{h}$



majorants pour $\mathrm{M}_k = |f| = 1$; pentes -1 (arrondi), 1 et 2 (troncature)

Minimum ($\mathrm{E}'(h) = 0$, les points) : $h^* \approx \sqrt{\varepsilon} \approx 10^{-8}$ (avant), $h^* \approx \varepsilon^{1/3} \approx 10^{-5}$ (centrées) ; erreurs $\approx 10^{-8}$ et $\approx 10^{-11}$.

★ La malédiction de la dimension

Idée naïve : évaluer f sur une grille de N points par axe, et garder la plus petite valeur. En dimension d, la grille a N^d points.

d	points ($\mathrm{N} = 100$)	temps à 10^7 / s
1	10^2	10 µs
2	10^4	1 ms
5	10^{10}	17 min
10	10^{20}	$3 \cdot 10^5$ ans

Conclusion

La recherche exhaustive est impossible dès $d \approx 5$. Il faut des algorithmes qui **utilisent le gradient** : à chaque pas, il indique la direction dans laquelle f diminue.

Au TP (★) : une « boîte noire » de 10 variables, minimisée par `scipy.optimize.minimize` en quelques centaines d'évaluations.

Le lien avec les équations différentielles : le flot de gradient

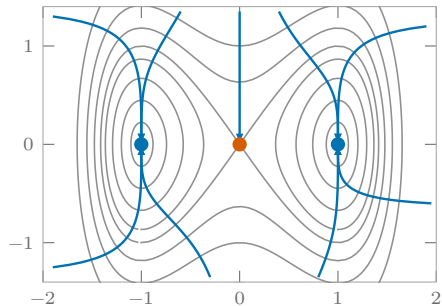
Suivons la plus forte pente : $x'(s) = -\nabla f(x(s))$.

- f **diminue** le long des trajectoires : $\dfrac{d}{ds} f(x(s)) = \nabla f^\top x' = -\|\nabla f\|^2 \leqslant 0$;
- les **équilibres** sont les points critiques de f ;
- la jacobienne en x^* est $-\nabla^2 f(x^*)$: un minimum strict est un équilibre **stable** (comme le critère $f'(a) < 0$ du TP 4).

Au TP 7

La méthode d'Euler explicite (TP 2) appliquée à ce flot donne la **descente de gradient** $x_{k+1} = x_k - t\,\nabla f(x_k)$.

trajectoires pour le double puits f_3



les trajectoires partent vers l'un des deux minima ; celle qui part de l'axe $x = 0$ finit au point selle

Ce que vous allez programmer

```python
def nature(H, tol=1e-10):
    lam = np.linalg.eigvalsh(H)
    if ...:
        return "minimum local strict"
    ...
    return "indéterminé"
```

Puis les points critiques *exacts* avec SymPy (`sp.solve`, sans les solutions complexes), et les lignes de niveau.

```python
def quadratique(A, b):
    ...                # f, grad, hess
    return f, grad, hess

def grad_num(f, x, h=1e-6):
    # différences centrées,
    # une coordonnée à la fois
    ...
```

Et le conditionnement κ, avec les ellipses de lignes de niveau.

En résumé

1. On cherche $\min f$; un minimum existe si f est continue et **coercive**.
2. Un minimum local est un **point critique** : $\nabla f(x^*) = 0$ (réciproque fausse).
3. En un point critique, les **valeurs propres de la hessienne** décident : toutes $> 0 \to$ minimum local strict, de signes opposés $\to$ point selle, une nulle $\to$ on ne peut pas conclure.
4. **Convexe** : tout point critique est un minimum global.
5. **Quadratique** $\frac{1}{2}x^\top \mathrm{A}x - b^\top x$: $\nabla f = \mathrm{A}x - b$, $\nabla^2 f = \mathrm{A}$; le **conditionnement** $\kappa = \lambda_{\max}/\lambda_{\min}$ mesure l'allongement des lignes de niveau.
6. On vérifie un gradient par **différences centrées** ($h \approx 10^{-6}$).

Pour aller plus loin : Boyd et Vandenberghe, *Convex Optimization*, § 9.1 ; Deisenroth, Faisal et Ong, *Mathematics for Machine Learning*, chapitre 5 (deux PDF libres).

Au travail !

Au programme du TP 6

1. gradients et hessiennes du « zoo », nature des points critiques ;
2. points critiques exacts avec SymPy, lignes de niveau ;
3. quand la hessienne ne permet pas de conclure ;
4. fonctions quadratiques, existence d'un minimiseur, conditionnement ;
5. le gradient numérique `grad_num` ;
★ pour aller plus loin : le choix du pas h, la recherche exhaustive.

Nouveau : mise en route automatique

À l'ouverture, attendez le message ✓ **Boîte à outils chargée**. En fin de séance : **File** → **Download**.



storoo.fr/TP_EDO

dossier `TP6`, fichier `TP6_Optimisation.ipynb`