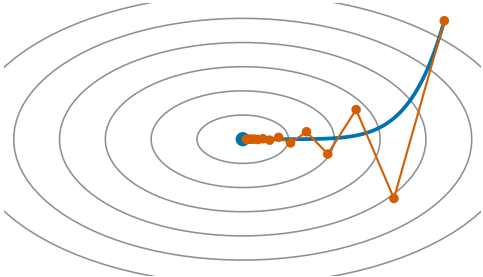
Équations différentielles et optimisation
L3 Mathématiques · Travaux pratiques

# La descente de gradient

## Le lien avec les EDO, et trois applications

Introduction au TP 8



en bleu, le flot de gradient (une EDO) ; en orange, la descente de gradient

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

# Du TP 7 au TP 8

**Au TP 7**, on a descendu une pente en une variable : $a_{k+1} = a_k - e\,f'(a_k)$, avec le rôle du pas et le moment.

**Au TP 8**, le pas est noté $t$ (comme dans le notebook), et :

1. on programme la **descente de gradient** en plusieurs variables, *à partir de rien* ;
2. on la relie au cours d'**équations différentielles** : c'est la méthode d'Euler ;
3. on l'utilise pour **apprendre à partir de données** : régression linéaire et logistique ;
4. et pour un problème de géométrie du XVII$^{\text{e}}$ siècle : le **point de Fermat**.

**Cette fois**

Le notebook donne la théorie et les étapes, mais **pas le code** : les fonctions sont à écrire en entier (un docstring et . . . ). La vérification dit ce qui ne va pas.

**Un peu d'histoire**

Cauchy (1847) pour l'astronomie ; aujourd'hui, c'est la méthode qui entraîne les modèles d'apprentissage automatique (TP 9 : les réseaux de neurones).

# Le gradient : la direction de plus forte pente

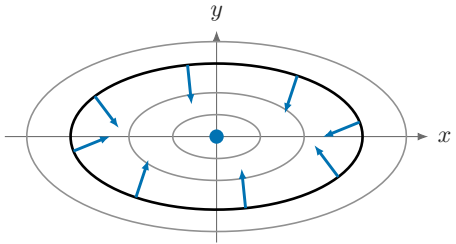
Pour $f : \mathbb{R}^n \to \mathbb{R}$, le gradient $\nabla f(x)$ (TP 6) :

- est **perpendiculaire** à la ligne de niveau qui passe par $x$ ;
- pointe dans la direction où $f$ **monte** le plus vite.

Donc $-\nabla f(x)$ (en bleu) est la direction de **plus forte descente**. D'après Taylor, avec $g = \nabla f(x)$ :

$$f(x - t\,g) \approx f(x) - t\,\|g\|^2 < f(x)$$

pour un pas $t > 0$ assez petit.




$f(x, y) = \frac{1}{2}(x^2 + 4y^2)$ : lignes de niveau, et $-\nabla f$ sur la ligne $f = 2$ (noir)

# L'algorithme

**Descente de gradient à pas fixe**

Données : $\nabla f$, un départ $x_0$, un pas $t > 0$, une tolérance $\mathrm{tol}$, un nombre maximal d'itérations $k_{\max}$.

Pour $k = 0, 1, \dots, k_{\max} - 1$ :

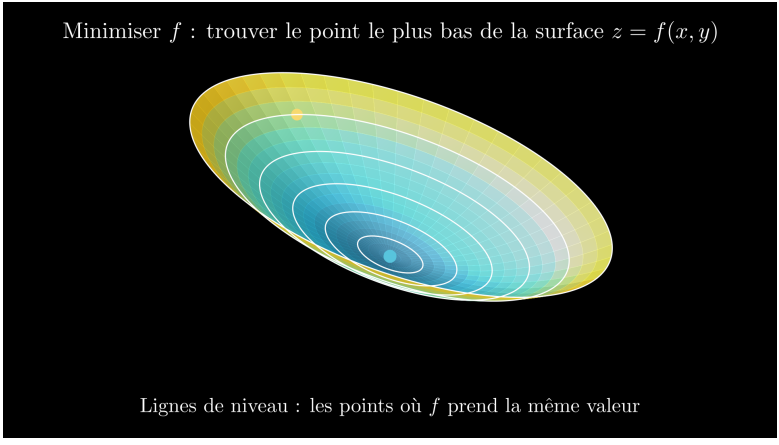
1. $g = \nabla f(x_k)$ ;
2. si $\|g\| < \mathrm{tol}$ : **arrêt** ;
3. sinon, $x_{k+1} = x_k - t\, g$.

**Trois choix** :

- la **direction** : $-\nabla f(x_k)$ ;
- le **pas** $t$ : trop petit, c'est lent ; trop grand, ça diverge ;
- le **test d'arrêt** : au minimum, $\nabla f = 0$ ; on s'arrête quand $\|\nabla f(x_k)\|$ est petit, *avant* de calculer un nouveau point.

On garde l'**historique** $x_0, x_1, \dots, x_k$ pour tracer la trajectoire et compter les itérations.

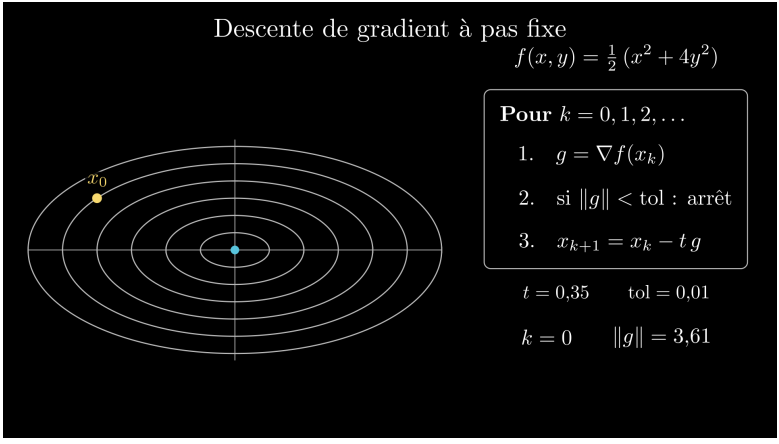
# La descente de gradient, en mouvement




La surface $z = \frac{1}{2}(x^2 + 4y^2)$ : une cuvette. Le départ $x_0$ (jaune), le minimum (bleu) au fond.

image 1 / 8 · ▶ la vidéo (53 s) : clic sur l'image (pdfpc), ou dans le lecteur vidéo

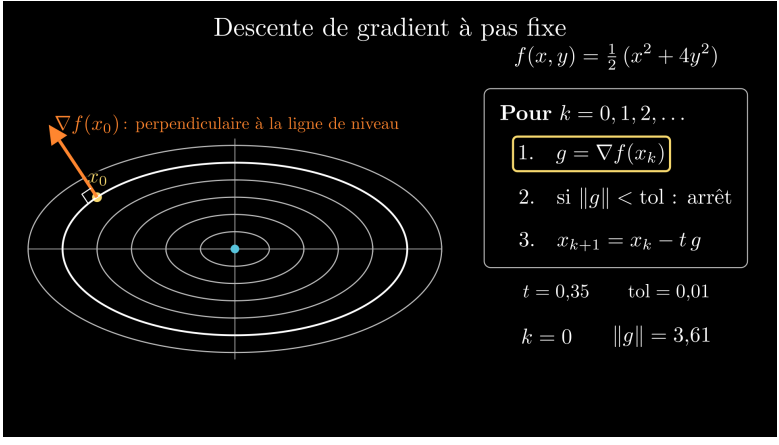
# La descente de gradient, en mouvement




Vue de dessus : la carte des lignes de niveau. À droite, l'algorithme, avec son test d'arrêt.

image 2 / 8 · ▶ la vidéo (53 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

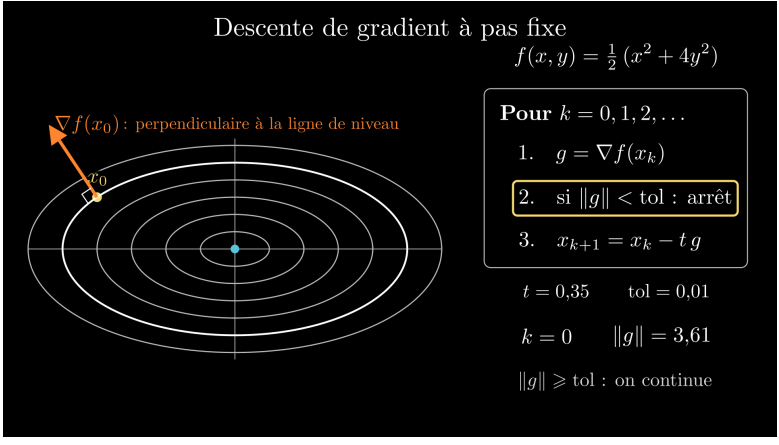
# La descente de gradient, en mouvement




**1.** $g = \nabla f(x_0)$ (orange) : perpendiculaire à la ligne de niveau de $x_0$ (en blanc).

image 3 / 8 · ▶ la vidéo (53 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

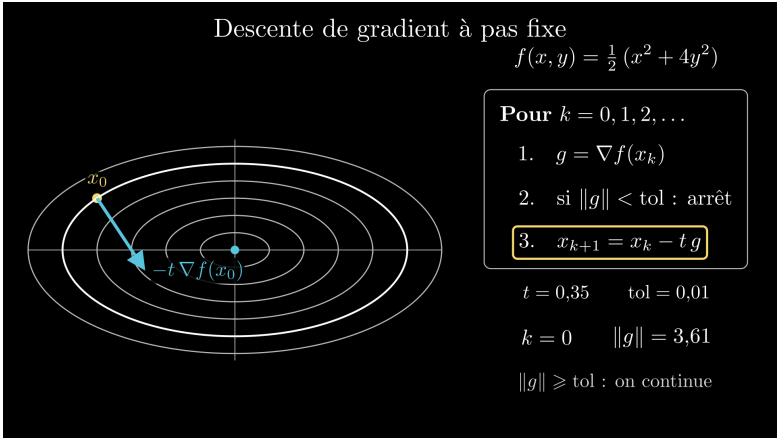
# La descente de gradient, en mouvement




**2.** Test d'arrêt : $\|g\| \geqslant \text{tol}$, on continue.

image 4 / 8 · ▶ la vidéo (53 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

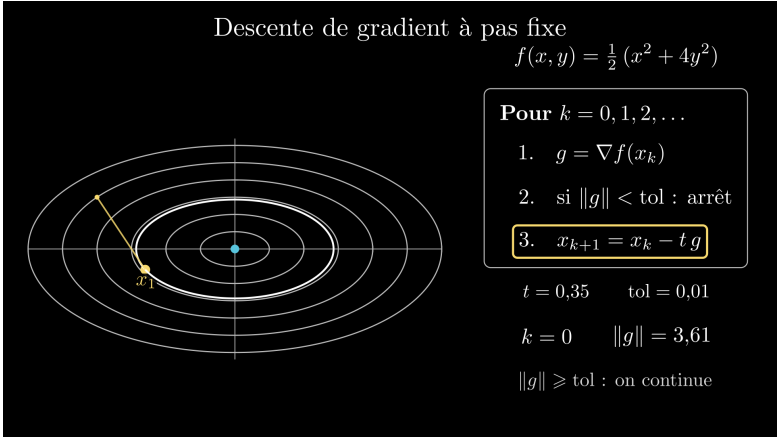
# La descente de gradient, en mouvement




**3.** Le pas $-t\,g$ (bleu) : la direction de plus forte pente.

image 5 / 8 · ▶ la vidéo (53 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

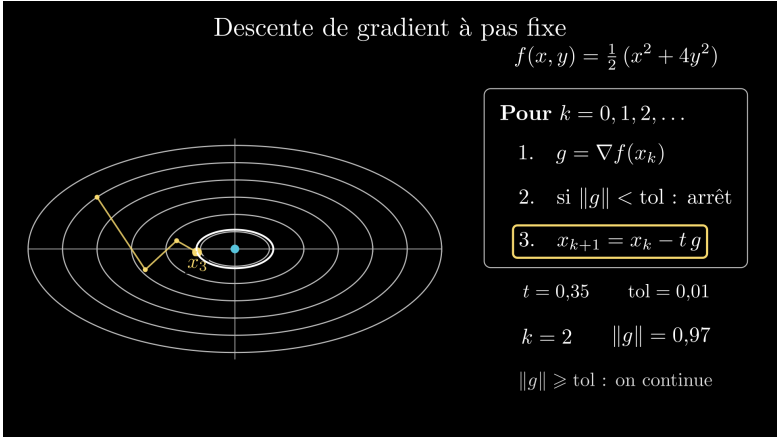
# La descente de gradient, en mouvement




$x_1 = x_0 - t\,\nabla f(x_0)$, sur sa propre ligne de niveau ; on recommence en $x_1$.

image 6 / 8 · ▶ la vidéo (53 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

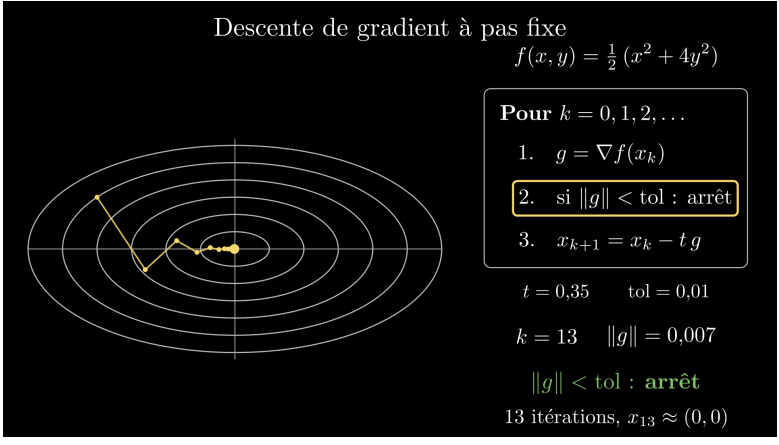
# La descente de gradient, en mouvement




Après trois itérations : les pas raccourcissent, car $\|\nabla f(x_k)\|$ diminue près du minimum.

image 7 / 8 · ▶ la vidéo (53 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# La descente de gradient, en mouvement




$\|g\| < \text{tol}$ : arrêt. Treize itérations pour $t = 0{,}35$ et $\text{tol} = 10^{-2}$.

image 8 / 8 · ▶ la vidéo (53 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# Le choix du pas

**En une variable** (TP 7) : pour $f(x) = \frac{\lambda}{2}x^2$ ($\lambda > 0$), dont le minimum est en $0$,

$$x_{k+1} = x_k - t\,\lambda\,x_k = (1 - t\lambda)\,x_k,$$

qui tend vers ce minimum si et seulement si $|1 - t\lambda| < 1$, soit $0 < t < 2/\lambda$.

**En plusieurs variables**, sur une quadratique $f(x) = \frac{1}{2}x^\top \mathrm{A}x$, chaque direction propre de $\mathrm{A}$ se comporte comme ci-dessus, avec sa valeur propre $\lambda_i$. La plus raide, $\mathrm{L} = \lambda_{\max}$, impose le seuil :

$$\text{convergence} \iff 0 < t < \frac{2}{\mathrm{L}}.$$

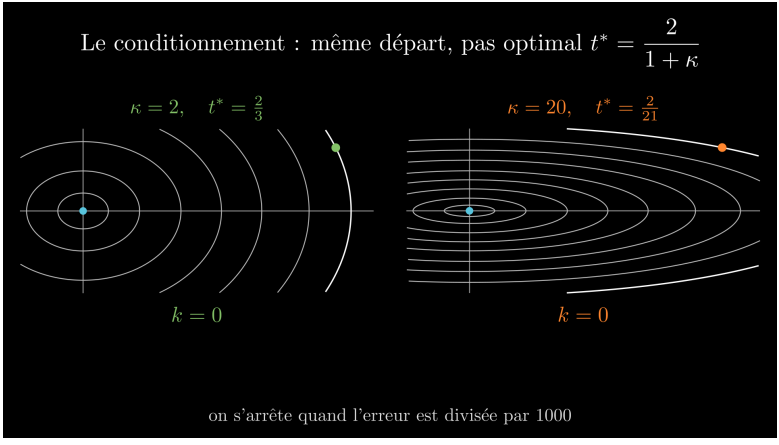
**Le conditionnement**

$\kappa = \mathrm{L}/\mu$, avec $\mu = \lambda_{\min}$ : le rapport des courbures de la direction la plus raide et de la plus plate.

Le meilleur pas fixe est $t^* = \dfrac{2}{\mu + \mathrm{L}}$ ; le nombre d'itérations est à peu près **proportionnel à** $\kappa$.

Près d'un minimum, la hessienne $\nabla^2 f(x^*)$ joue le rôle de $\mathrm{A}$ (Taylor à l'ordre 2, TP 6).

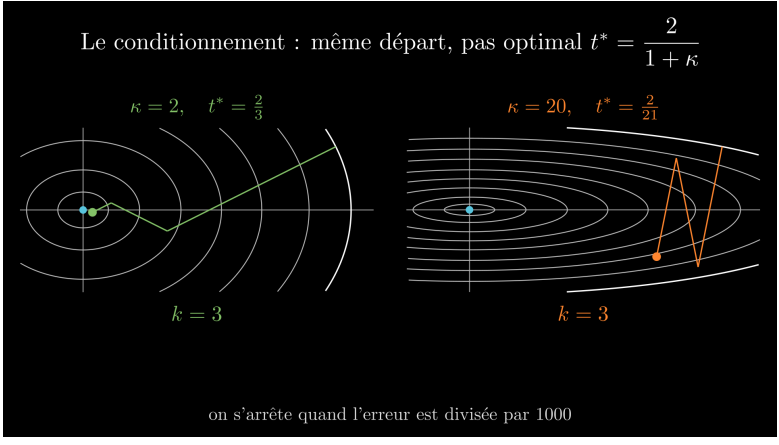
# Une vallée étroite ralentit la méthode




$\frac{1}{2}(x^2 + \kappa y^2)$, même départ (sur la ligne de niveau blanche) et, pour chacune, le meilleur pas.

image 1 / 4 · ▶ la vidéo (22 s) : clic sur l'image (pdfpc), ou dans le lecteur vidéo

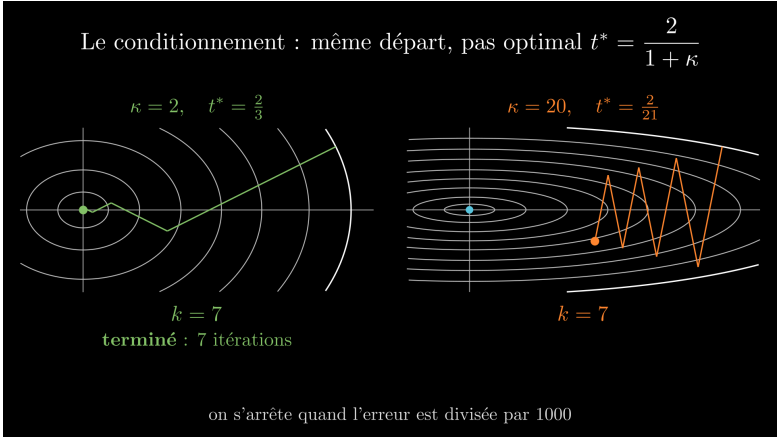
# Une vallée étroite ralentit la méthode




Trois itérations : $\kappa = 2$ est presque arrivé ; $\kappa = 20$ zigzague d'un flanc à l'autre de la vallée.

image 2 / 4 · ▶ la vidéo (22 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

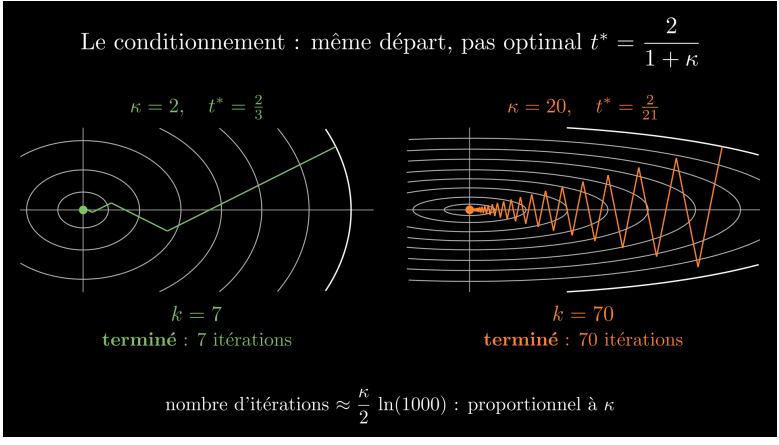
# Une vallée étroite ralentit la méthode




$\kappa = 2$ : terminé en 7 itérations.

image 3 / 4 · ▶ la vidéo (22 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# Une vallée étroite ralentit la méthode




$\kappa = 20$ : 70 itérations, dix fois plus. Au TP, vous verrez $\kappa = 25$.

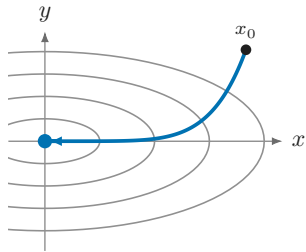
image 4 / 4 · ▶ la vidéo (22 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# Une bille dans du miel : le flot de gradient

Une bille posée sur la surface $z = f(x)$, freinée par un liquide très visqueux, suit la pente sans prendre d'élan : sa vitesse est $-\nabla f$. C'est l'EDO du **flot de gradient**

$$x'(s) = -\nabla f(x(s)), \qquad x(0) = x_0.$$

- $f$ **décroît** le long des solutions :
  $\dfrac{d}{ds} f(x(s)) = -\left\|\nabla f(x(s))\right\|^2 \leqslant 0.$
- Ses **équilibres** ($x' = 0$) sont les **points critiques** de $f$.
- Un minimum strict est un équilibre **stable** : les solutions qui partent près de lui y tendent.




$f = \frac{1}{2}(x^2 + 6y^2)$ : la solution descend la pente, perpendiculairement aux lignes de niveau

**Vocabulaire**

$f$ est une **fonction de Lyapunov** du flot : elle diminue le long de toutes les trajectoires.

# Euler sur le flot = la descente de gradient

La méthode d'**Euler explicite** (TP 2), de pas $h$, pour $x' = \mathrm{F}(x)$ :

$$x_{k+1} = x_k + h\,\mathrm{F}(x_k).$$

Avec $\mathrm{F} = -\nabla f$ :

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

c'est **exactement** la descente de gradient, de pas $t = h$. Les itérés $x_k$ approchent la solution $x(s)$ aux temps $s = k\,h$.

| **EDO** | **Optimisation** |
|---|---|
| flot $x' = -\nabla f(x)$ | minimiser $f$ |
| pas de temps $h$ | pas $t$ |
| méthode d'Euler | descente de gradient |
| équilibre stable | minimum |
| Euler stable | la descente converge |
| EDO raide (TP 5) | ? (exercice 3) |

# La stabilité d'Euler (TP 2) et le seuil du pas

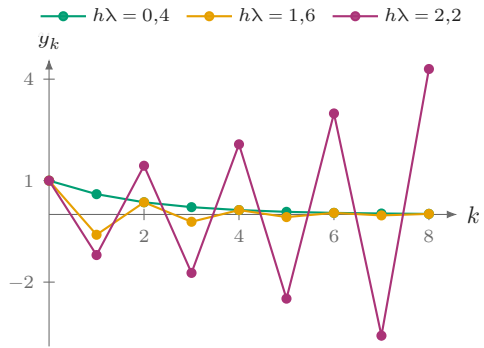
Sur $y' = -\lambda\, y$ ($\lambda > 0$), la solution exacte $y_0\, \mathrm{e}^{-\lambda s}$ tend vers 0. Euler donne

$$y_{k+1} = (1 - h\lambda)\, y_k,$$

qui tend vers 0 si et seulement si $|1 - h\lambda| < 1$ :

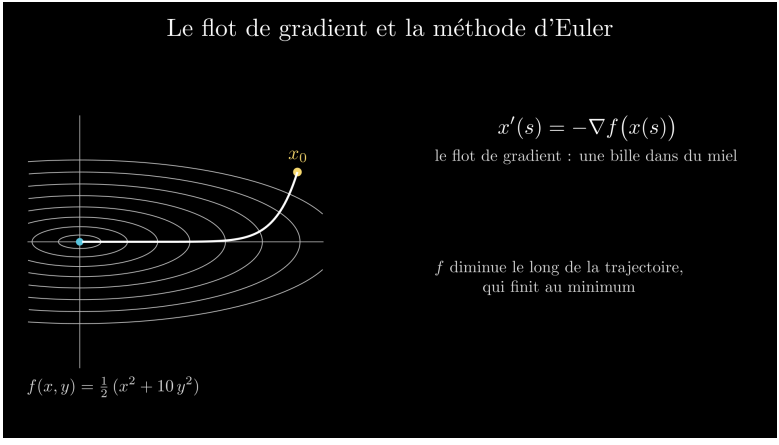
$$\text{Euler est stable} \iff h < \frac{2}{\lambda}.$$

Le flot d'une quadratique $f(x) = \frac{1}{2} x^\top \mathrm{A} x$ est un système de telles équations, une par direction propre de $\mathrm{A}$, avec $\lambda = \lambda_i$. Euler est stable si $h < 2/\lambda_i$ pour tout $i$, soit $h < 2/\mathrm{L}$ avec $\mathrm{L} = \lambda_{\max}$ : c'est **le seuil du pas** de la descente de gradient.




$\lambda = 4$, $y_0 = 1$ : facteur $1 - h\lambda = 0{,}6$ (vert), $-0{,}6$ (oscille et converge), $-1{,}2$ (diverge)

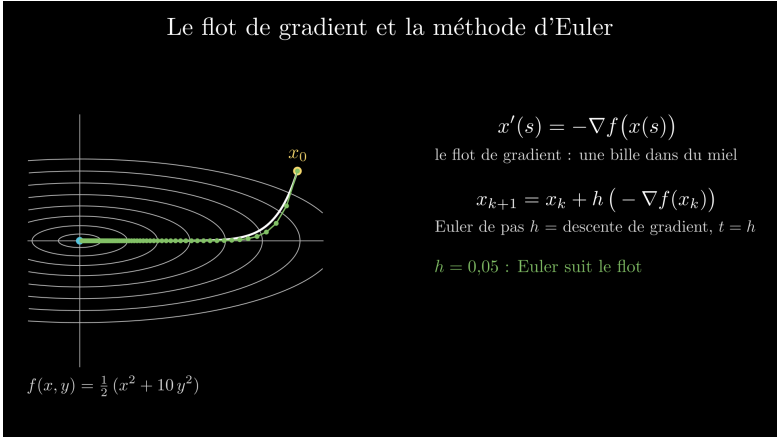
# Le flot et Euler, en mouvement




$f = \frac{1}{2}(x^2 + 10\,y^2)$ (L $= 10$) : la solution exacte du flot (blanc) glisse vers le minimum.

image 1 / 5 · ▶ la vidéo (28 s) : clic sur l'image (pdfpc), ou dans le lecteur vidéo

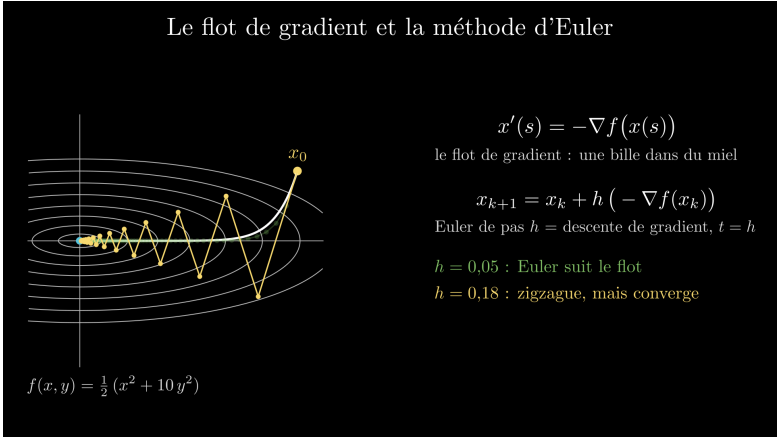
# Le flot et Euler, en mouvement




Euler avec $h = 0{,}05$ : les points suivent la trajectoire du flot.

image 2 / 5 · ▶ la vidéo (28 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

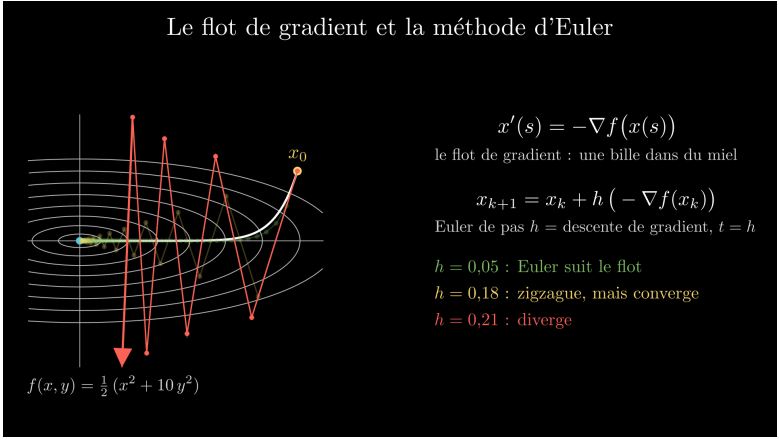
# Le flot et Euler, en mouvement




$h = 0{,}18$ : dans la direction raide, le facteur est $1 - 10h = -0{,}8$ ; les points zigzaguent.

image 3 / 5 · ▶ la vidéo (28 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

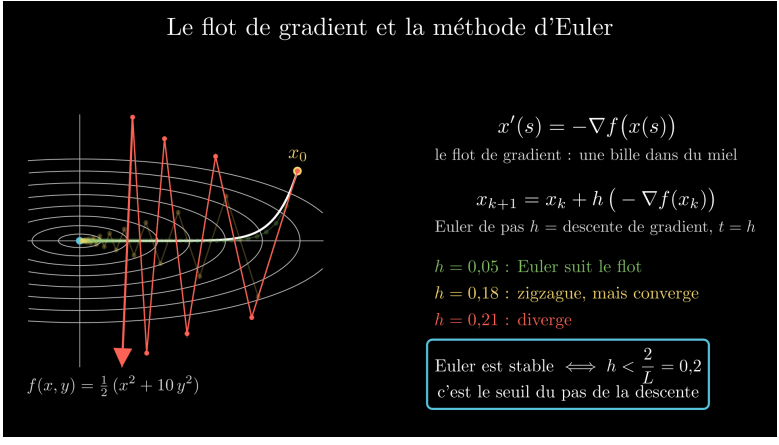
# Le flot et Euler, en mouvement




$h = 0{,}21$ : facteur $-1{,}1$, les points s'éloignent : Euler diverge.

image 4 / 5 · ▶ la vidéo (28 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# Le flot et Euler, en mouvement




Le seuil est $h = 2/\mathrm{L} = 0{,}2$ : le même que pour la descente de gradient.

image 5 / 5 · ▶ la vidéo (28 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# D'autres schémas, d'autres méthodes

### ★ Euler implicite = le point proximal

Le schéma $x_{k+1} = x_k - h\,\nabla f(x_{k+1})$ (TP 2) : chaque pas résout une équation. Pour $f$ convexe, c'est

$$x_{k+1} = \operatorname*{argmin}_x \left( f(x) + \frac{\|x - x_k\|^2}{2h} \right) :$$

on minimise $f$ sans trop s'éloigner de $x_k$. Il est stable **pour tout** $h > 0$ (exercice 9).

### Le moment = une bille pesante

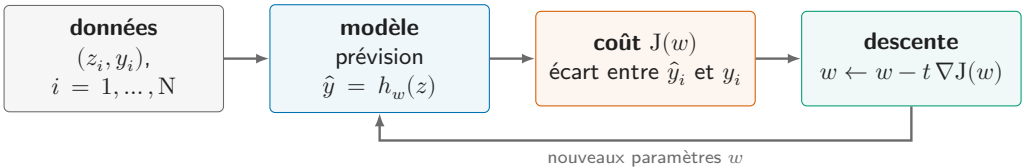
Avec une masse et un frottement, la bille obéit à une EDO du second ordre

$$x'' + \gamma\, x' + \nabla f(x) = 0.$$

Discrétisée, elle donne la méthode du **moment** du TP 7 (Polyak, 1964) : l'élan aide à traverser les zones plates.

Plus généralement : une méthode d'optimisation est souvent un schéma numérique pour une EDO bien choisie.

# Apprendre à partir de données




**Entraîner** un modèle, c'est minimiser $\mathrm{J}(w)$ : chercher les paramètres $w$ qui expliquent le mieux les données.

- **régression** : $y$ est un nombre (une distance, un prix) ;
- **classification** : $y$ vaut 0 ou 1 (échec ou réussite, incident ou non).

# La régression linéaire

**Exemple** : la surface $z_i$ (m²) et le prix $y_i$ (k€) de $49$ appartements à Paris. On cherche la **droite** $\hat{y} = w_0 + w_1 z$ qui prévoit au mieux le prix à partir de la surface.

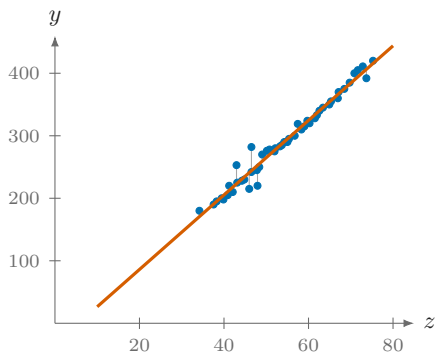
**Coût** : la moyenne des carrés des écarts (en gris),

$$\mathrm{J}(w_0, w_1) = \frac{1}{2\mathrm{N}} \sum_{i=1}^{\mathrm{N}} (w_0 + w_1 z_i - y_i)^2.$$

Minimiser J donne $\hat{y} \approx -33{,}2 + 5{,}96\, z$, soit environ $6\,000$ € le m² ; un 50 m² vaut environ $265$ k€.

**Plusieurs caractéristiques** (surface, pièces…) : $\hat{y} = \mathrm{X}w$, où X a une **colonne de 1**, puis une colonne par caractéristique ;

$$\mathrm{J}(w) = \frac{1}{2\mathrm{N}} \|\mathrm{X}w - y\|^2, \qquad \nabla \mathrm{J}(w) = \frac{1}{\mathrm{N}} \mathrm{X}^\top (\mathrm{X}w - y).$$




surface $z$ (m²), prix $y$ (k€) ; en gris, les écarts $\hat{y}_i - y_i$

# Classer : la régression logistique

Quand $y$ vaut 0 ou 1, on prévoit une **probabilité** :

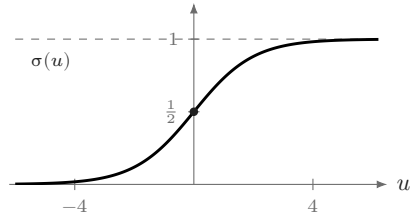
$$p = \sigma(w_0 + w_1 z), \qquad \sigma(u) = \frac{1}{1+\mathrm{e}^{-u}} \in ]0,1[.$$

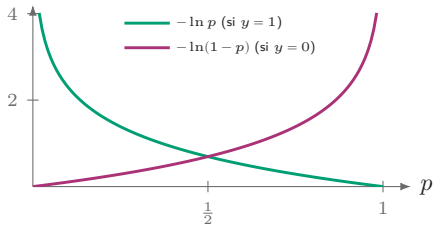
La **sigmoïde** $\sigma$ transforme un nombre quelconque en probabilité ; on prévoit $y = 1$ si $p \geqslant \frac{1}{2}$.

**Coût** : l'**entropie croisée**

$$\mathrm{J}(w) = -\frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \Big[ y_i \ln p_i + (1-y_i) \ln(1-p_i) \Big].$$

Pour une donnée $y_i = 1$, le coût est $-\ln p_i$ : nul si $p_i = 1$, très grand si le modèle est sûr de lui et se trompe ($p_i \to 0$).



# Le gradient de l'entropie croisée

Avec la même matrice $\mathrm{X}$ (colonne de 1, puis les caractéristiques) et $p = \sigma(\mathrm{X}w)$ :

$$\nabla \mathrm{J}(w) = \frac{1}{\mathrm{N}} \mathrm{X}^\top (\sigma(\mathrm{X}w) - y).$$

C'est la **même forme** qu'en régression linéaire, avec $\sigma(\mathrm{X}w)$ à la place de $\mathrm{X}w$.

J est convexe, mais on ne sait pas résoudre $\nabla \mathrm{J}(w) = 0$ à la main : on **doit** utiliser une méthode itérative, comme la descente de gradient.

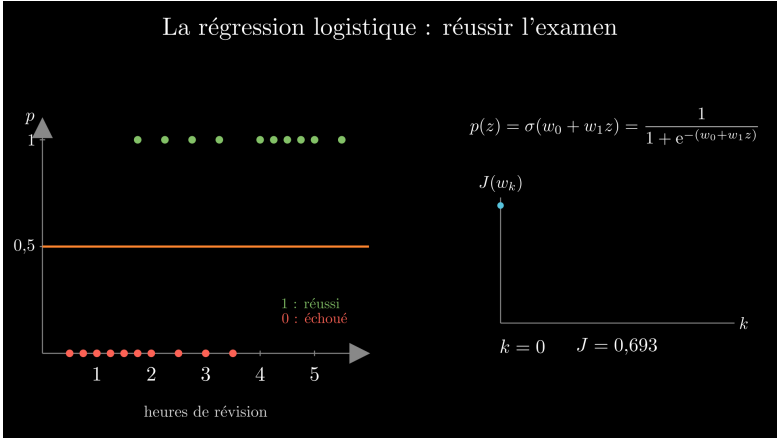
### D'où vient $\mathrm{J}$ ?

C'est le maximum de vraisemblance : chaque $y_i$ suit une loi de Bernoulli de paramètre $p_i$, et $\mathrm{J}(w) = -\frac{1}{\mathrm{N}} \ln(\text{vraisemblance})$.

### Au TP

Les données de la navette **Challenger** : peut-on estimer le risque d'incident en fonction de la température ?

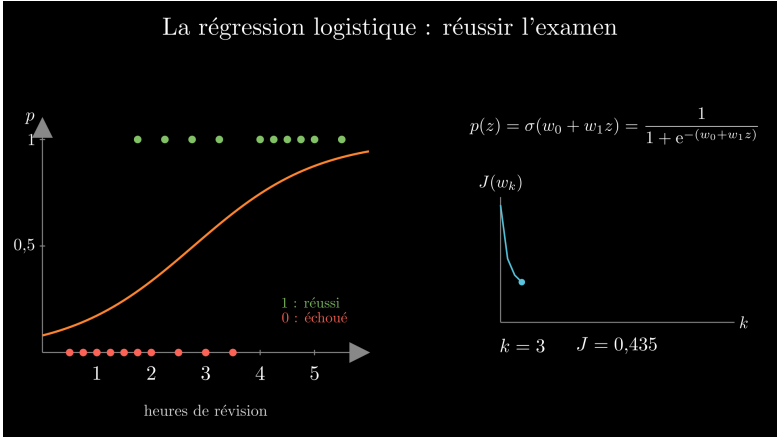
# La régression logistique, en mouvement




20 étudiants : heures de révision, examen réussi (1) ou non (0). Au départ, $w = 0$ : $p = \frac{1}{2}$ partout, $\mathrm{J} = \ln 2$.

image 1 / 4 · ▶ la vidéo (20 s) : clic sur l'image (pdfpc), ou dans le lecteur vidéo

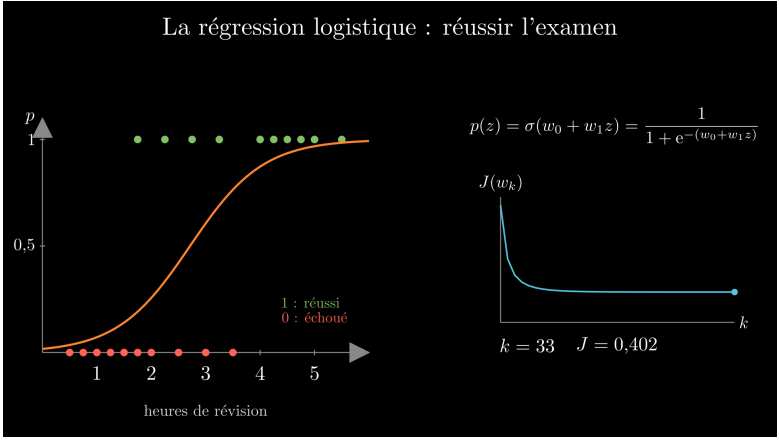
# La régression logistique, en mouvement




Trois itérations : la courbe se redresse, J baisse vite.

image 2 / 4 · ▶ la vidéo (20 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

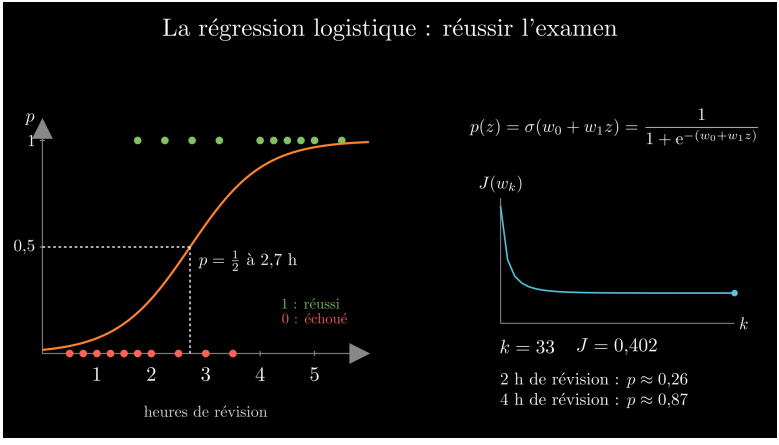
# La régression logistique, en mouvement




33 itérations : $\|\nabla \mathrm{J}\| < 10^{-3}$. Le modèle obtenu : $p = \sigma(-4{,}1 + 1{,}5\, z)$, avec $z$ en heures.

image 3 / 4 · ▶ la vidéo (20 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# La régression logistique, en mouvement




Lecture : $p = \frac{1}{2}$ vers 2,7 h ; $p \approx 0{,}26$ pour 2 h et $0{,}87$ pour 4 h.

image 4 / 4 · ▶ la vidéo (20 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# Une application mathématique : le point de Fermat

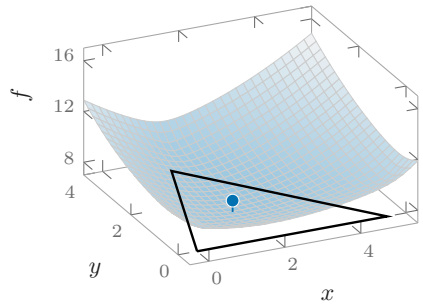
Vers 1640, Fermat pose à Torricelli ce problème : trouver le point $x$ du plan qui minimise la **somme des distances** aux trois sommets $a, b, c$ d'un triangle,

$$f(x) = \|x - a\| + \|x - b\| + \|x - c\|.$$

Où placer un entrepôt pour desservir trois villes avec le moins de route possible ?

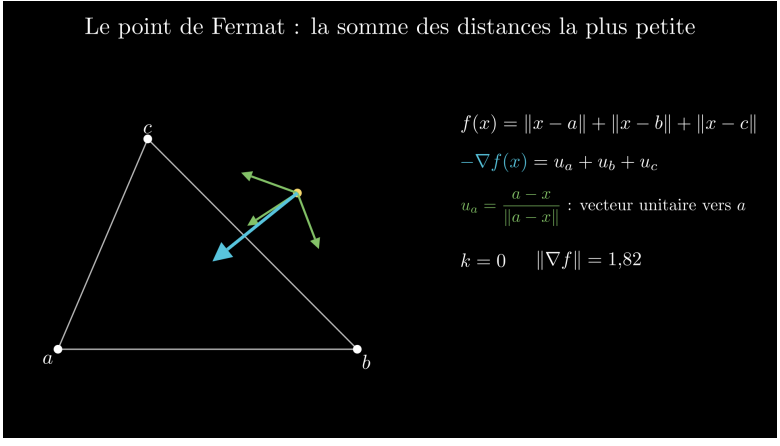
Le gradient de $x \mapsto \|x - a\|$ est le vecteur **unitaire** $\dfrac{x - a}{\|x - a\|}$ (hors de $a$) : $\nabla f$ est la somme de trois vecteurs unitaires.

**Une solution mécanique** (Varignon) : trois ficelles nouées, passées par trois trous en $a$, $b$, $c$, avec des poids égaux. Le nœud s'arrête là où les trois tractions s'équilibrent.




$f$ pour $a = (0, 0)$, $b = (5, 0)$, $c = (1{,}5; 3{,}5)$ : une surface convexe, avec le triangle en dessous ; minimum en bleu

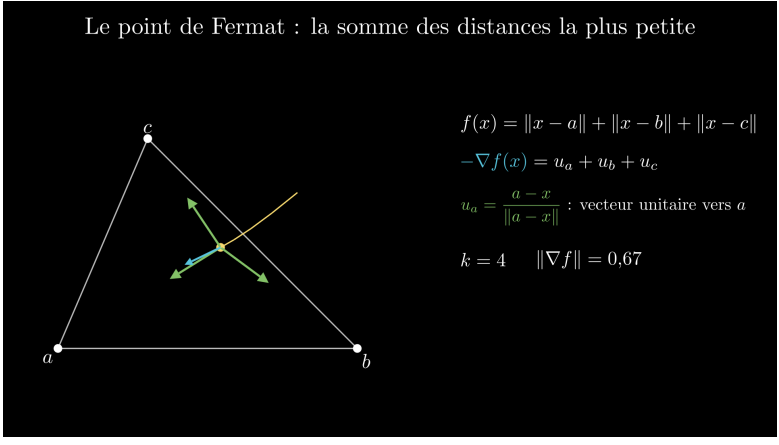
# Le point de Fermat, en mouvement




En $x$ : les vecteurs unitaires vers les sommets (vert) ; leur somme est $-\nabla f(x)$ (bleu), la direction de descente.

image 1 / 3 · ▶ la vidéo (25 s) : clic sur l'image (pdfpc), ou dans le lecteur vidéo

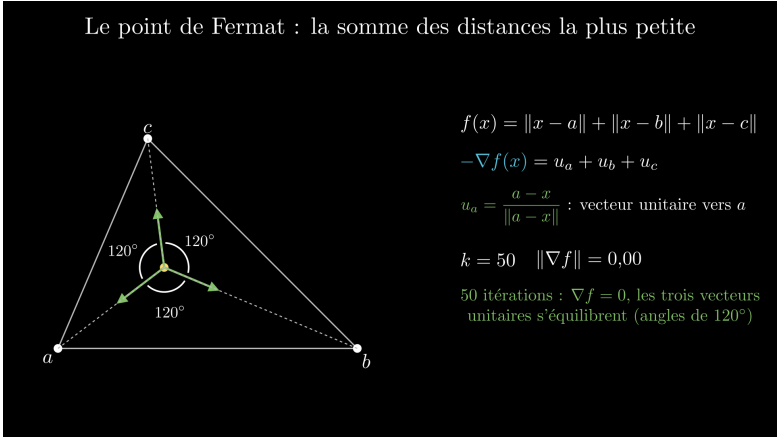
# Le point de Fermat, en mouvement




Pas $t = 0{,}3$ : après quatre itérations, la somme des trois vecteurs a beaucoup diminué.

image 2 / 3 · ▶ la vidéo (25 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

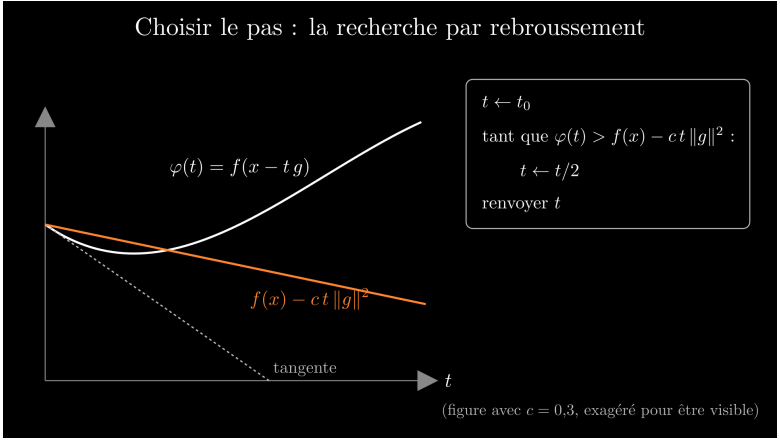
# Le point de Fermat, en mouvement




50 itérations : $x^* \approx (1{,}77; 1{,}35)$, $f(x^*) \approx 7{,}89$. Les trois vecteurs unitaires s'équilibrent : angles de $120°$.

image 3 / 3 · ▶ la vidéo (25 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

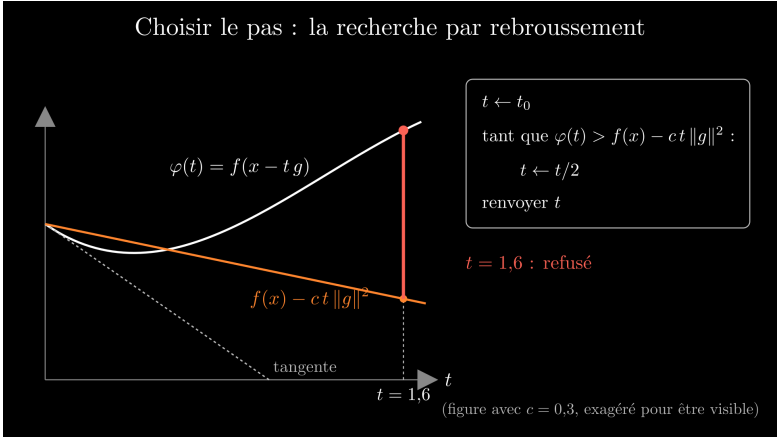
# ★ Choisir le pas tout seul : la règle d'Armijo




Un pas fixe doit convenir partout. Armijo essaie $t_0$, puis $t_0/2$, … jusqu'à une **baisse suffisante** de $f$ (droite orange).

image 1 / 4 · ▶ la vidéo (21 s) : clic sur l'image (pdfpc), ou dans le lecteur vidéo

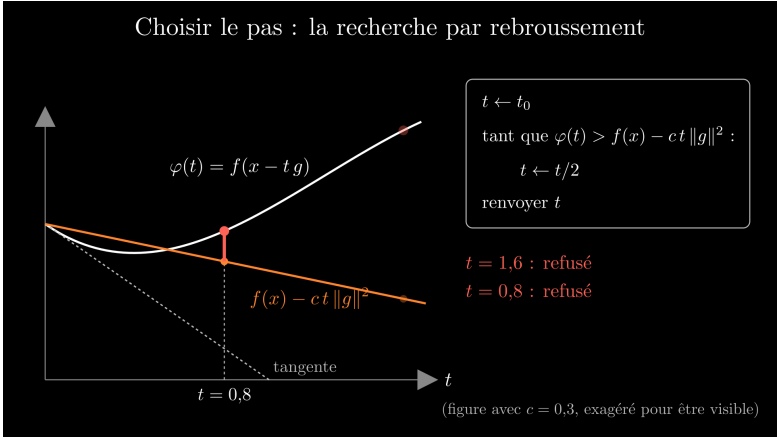
# ★ Choisir le pas tout seul : la règle d'Armijo




$t = 1{,}6$ : $\varphi(t) = f(x - t\,g)$ est au-dessus de la droite, refusé ; on divise $t$ par 2.

image 2 / 4 · ▶ la vidéo (21 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

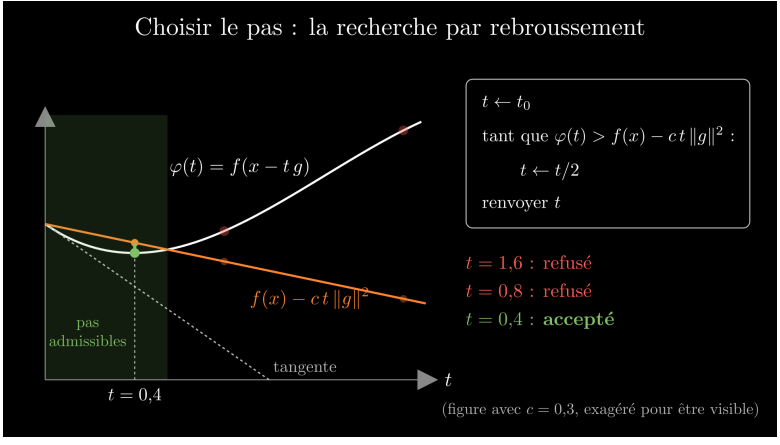
# ★ Choisir le pas tout seul : la règle d'Armijo




$t = 0{,}8$ : encore au-dessus, refusé ; on divise encore par 2.

image 3 / 4 · ▶ la vidéo (21 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# ★ Choisir le pas tout seul : la règle d'Armijo




$t = 0{,}4$ : sous la droite, accepté. En vert, tous les pas admissibles.

image 4 / 4 · ▶ la vidéo (21 s) : clic sur l'image 1 (pdfpc), ou dans le lecteur vidéo

# Ce que vous allez programmer

```
def descente_gradient(grad_f, x0, pas,
                      tol=1e-8, kmax=10_000):
    """... Renvoie (x, historique)."""
    ...      # tout le corps : à vous
```

Elle sert dans tout le TP : écrivez-la avec soin (copie de $x_0$, test d'arrêt avant la mise à jour, `kmax`).

1. `descente_gradient` ;
2. le seuil et le meilleur pas ;
3. Euler = la descente (`flot`) ;

4–5. les maisons : `cout_mc`, `grad_mc`, puis l'entraînement ;

6–7. Challenger : `sigmoide`, `cout_log`, `grad_log` ;

8. `grad_fermat` ;

★ Euler implicite ; Armijo.

# En résumé

1. **Descente de gradient** : $x_{k+1} = x_k - t\,\nabla f(x_k)$, arrêt quand $\|\nabla f(x_k)\|$ est petit.
2. **Le pas** : convergence si $t < 2/\mathrm{L}$ ; le nombre d'itérations croît avec $\kappa = \mathrm{L}/\mu$ (vallées étroites, zigzags).
3. **EDO** : c'est la méthode d'Euler sur le flot de gradient $x' = -\nabla f(x)$, dont les minima sont les équilibres stables ; le seuil du pas est la condition de stabilité d'Euler.
4. **Apprendre** = minimiser un coût : moindres carrés (régression linéaire), entropie croisée (régression logistique), avec le même gradient $\frac{1}{\mathrm{N}}\mathrm{X}^\top(\text{prévisions} - y)$.
5. **Fermat** : le gradient d'une distance est un vecteur unitaire ; au minimum, trois vecteurs unitaires s'équilibrent.

Pour aller plus loin : S. J. Wright et B. Recht, *Optimization for Data Analysis*, Cambridge University Press, 2022 (ch. 1 et 3, et § 4.1 : le lien avec les EDO) ; W. Su, S. Boyd et E. Candès, « A Differential Equation for Modeling Nesterov's Accelerated Gradient Method », *JMLR* 17, 2016.

# Au travail !

**Au programme du TP 8**

1. la descente de gradient, le pas et le conditionnement ;
2. la descente de gradient est la méthode d'Euler ;
3. la régression linéaire : le prix de 100 maisons ;
4. la régression logistique : l'accident de Challenger ;
5. le point de Fermat ;
- ★ Euler implicite, la règle d'Armijo.

**Rappel**

À l'ouverture, attendez le message ✓ **Boîte à outils chargée**. En fin de séance : **File** → **Download**. Au TP 9 : les réseaux de neurones.



storoo.fr/TP_EDO
dossier TP8, fichier TP8_Applications.ipynb