Méthodes numériques pour l'optimisation

Grégory VIAL

École Centrale de Lyon

Filière "Mathématique et Ingénierie du Risque"

MsO "Mathématiques en situation"

Janvier 2013

Objectifs de la formation

✔ Donner quelques exemples représentatifs de problèmes d'optimisation.

✔ Introduire les principales techniques numériques.

✔ Présenter les difficultés et les limitations des méthodes.

✘ Description non exhaustive des problèmes et méthodes.

✘ Seulement les méthodes élémentaires, pas de sophistication.

Plan

I. Quelques problèmes d'optimisation

Optimisation discrète/continue, avec/sans contrainte

II. Optimisation locale sans contrainte

II.1. Méthode de Newton-Raphson

II.2. Gradient, gradient conjugué

III. Prise en compte de contraintes

III.1. Projection sur l'ensemble des contraintes

III.2. Pénalisation des contraintes

III.3. Méthodes lagrangiennes

IV. Optimisation globale ?

IV.1. Méthodes stochastiques : recuit simulé, algorithmes évolutionnaires

IV.2. Notions d'optimisation multi-objectifs

Plan

I. Quelques problèmes d'optimisation

Optimisation discrète/continue, avec/sans contrainte

II. Optimisation locale sans contrainte

II.1. Méthode de Newton-Raphson

II.2. Gradient, gradient conjugué

III. Prise en compte de contraintes

III.1. Projection sur l'ensemble des contraintes

III.2. Pénalisation des contraintes

III.3. Méthodes lagrangiennes

IV. Optimisation globale ?

IV.1. Méthodes stochastiques : recuit simulé, algorithmes évolutionnaires

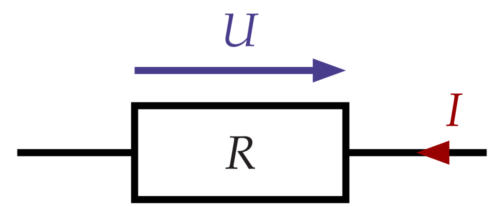
IV.2. Notions d'optimisation multi-objectifs

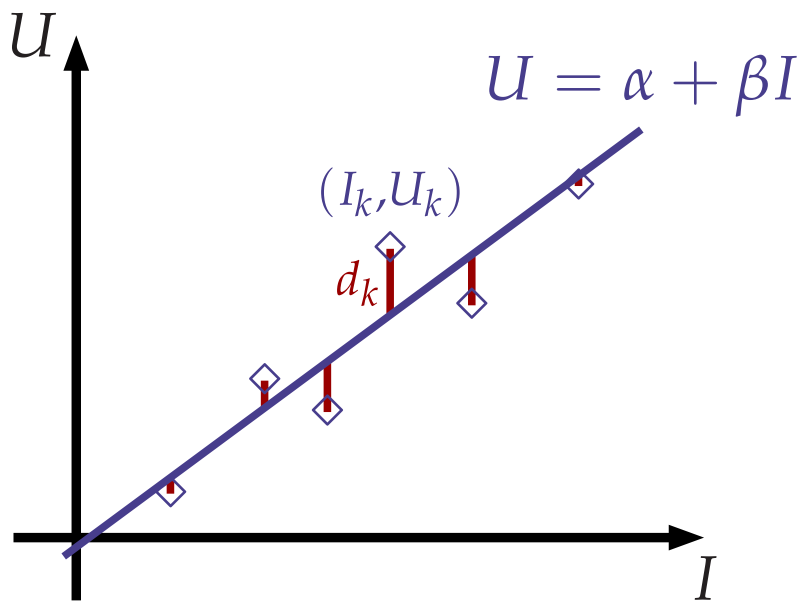
I. Exemples – Optimisation continue sans contrainte

I. Exemples – Optimisation continue sans contrainte

Ajustement linéaire

Mesures de l'intensité et de la tension aux bornes d'un resistor.





On cherche la droite telle que la somme des distances d_k au carré soit minimale.

$$\min_{(\alpha,\beta)\in\mathbb{R}^2} \sum_{k=1}^{N} d_k^2$$

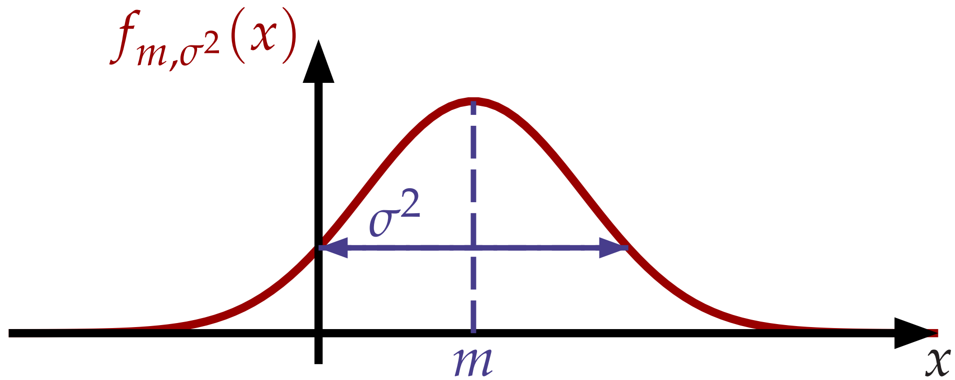
I. Exemples – Optimisation continue sans contrainte

Ajustement linéaire

Estimateur du maximum de vraisemblance

On modélise la taille des bulles d'une mousse par une variable aléatoire gaussienne $\mathcal{N}(m, \sigma^2)$.

→ Comment estimer m et σ^2 ?



On note X_k les tailles mesurées par l'expérience ($k = 1, \ldots, N$). On demande que les paramètres m et σ^2 maximisent la vraisemblance :

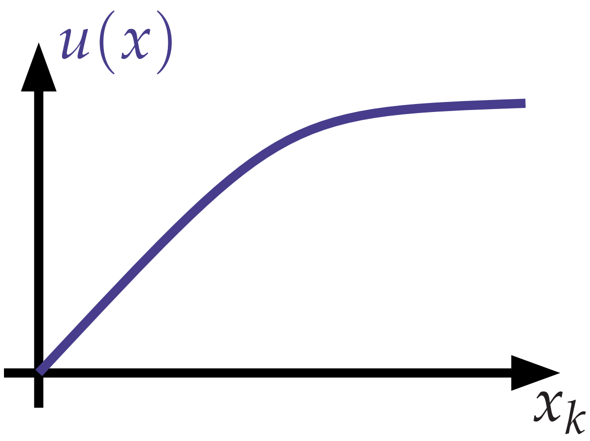
$$\max_{(m,\sigma)\in\mathbb{R}^2} \prod_{k=1}^{N} f_{m,\sigma^2}(X_k)$$

I. Exemples – Optimisation continue avec contrainte

I. Exemples – Optimisation continue avec contrainte

En économie

On s'intéresse à la consommation des ménages.



- Un choix de N produits de prix $p_1, p_2, \ldots, p_N$;
- Quantité consommée x_k du produit n° k ;
- Fonction d'utilité $u(x_1, \ldots, x_N)$.

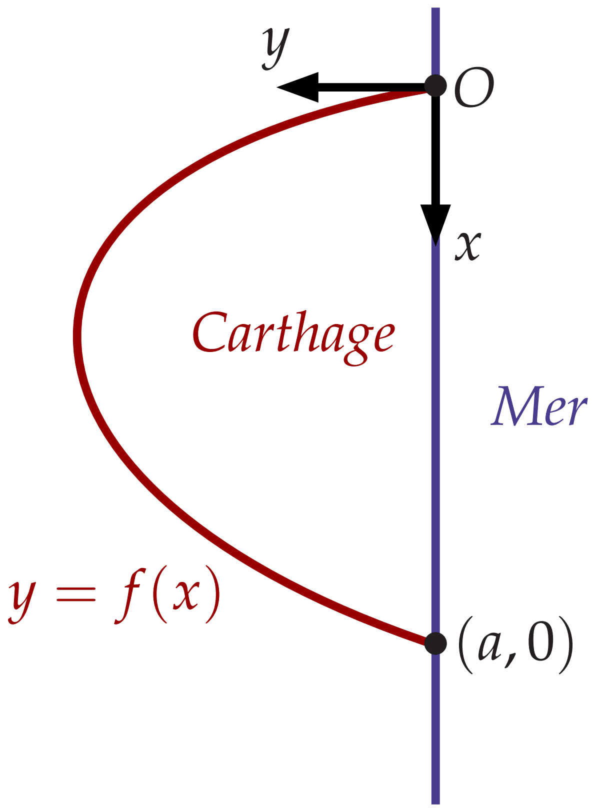
Les ménages cherchent à maximiser leur utilité sous contrainte budgétaire.

$$\max_{\substack{x_1,\ldots,x_N \geq 0 \\ x_1 p_1 + \cdots + x_N p_N \leq b}} u(x_1, \ldots, x_N) \quad \text{ou encore} \quad \max_{x \in \mathbb{R}_+^N,\ x \cdot p \leq b} u(x).$$

I. Exemples – Optimisation continue avec contrainte

En économie

Le problème de Didon



Un problème isopérimétrique

La reine phénicienne reçut comme territoire ce que pouvait contenir une peau de bœuf. La découpant en fines lanières, elle délimita le domaine de la future Carthage en maximisant la surface sous contrainte de périmètre.

$$\max \left\{ \int_0^a f(x)\,\mathrm{d}x \; ; \; \int_0^a \sqrt{1+f'(x)^2}\,\mathrm{d}x = p \right\}$$

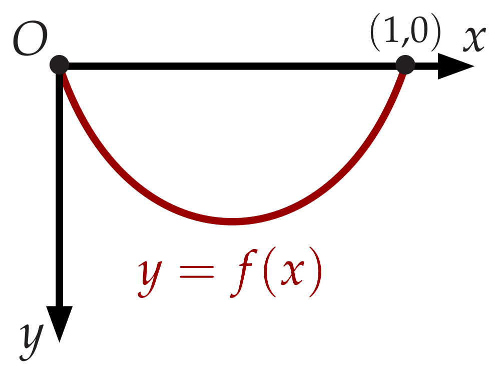
I. Exemples – Optimisation continue avec contrainte

En économie

Le problème de Didon

La chaînette

Quelle est la forme prise par une corde soumise à son seul poids ?



$$\min\left\{-\int_0^1 f(x)\sqrt{1+f'(x)^2}\,\mathrm{d}x\ ;\ \int_0^1 \sqrt{1+f'(x)^2}\,\mathrm{d}x=\ell\right\}$$

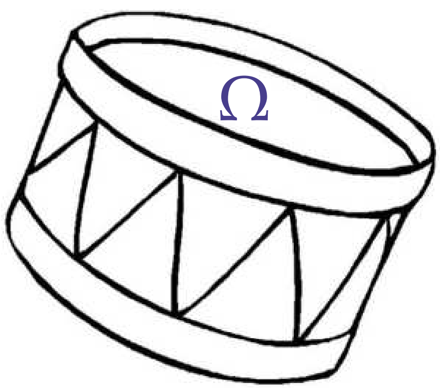
I. Exemples – Optimisation continue avec contrainte

En économie

Le problème de Didon

La chaînette

Le tambour le plus grave



Ω	membrane (2D) ;
Γ	bord attaché de la membrane ;
$\lambda_1(\Omega)$	première valeur propre du Laplacien sur Ω avec condition de Dirichlet sur Γ ;
u_Ω	fonction propre associée.

Quelle forme donner au tambour pour que la vibration fondamentale soit la plus grave ?

$$\min_{\mathrm{Surf}(\Omega)=s} \lambda_1(\Omega)$$

I. Exemples – Optimisation discrète

I. Exemples – Optimisation discrète

Le problème du sac à dos

Comment optimiser la valeur du contenu d'un sac à dos de contenance fixée ?

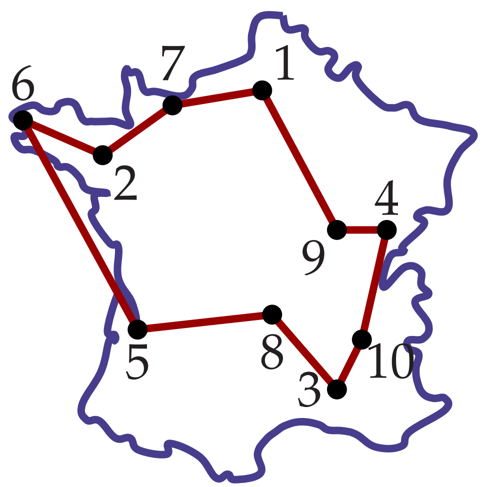
- ◇ objet n° i de taille t_i et de valeur v_i ;
- ◇ si l'objet n° i est sélectionné, $\delta_i = 1$, sinon $\delta_i = 0$.

$$\max\{\text{valeur}\ ;\ \text{taille}\ \leq T\} \quad \text{soit} \quad \max\left\{\sum_i \delta_i v_i\ ;\ \sum_i \delta_i t_i \leq T\right\}$$

I. Exemples – Optimisation discrète

Le problème du sac à dos

Le problème du voyageur de commerce



Dans quelle ordre faire la tournée de ces 10 villes ?

$c[i,j]$	"coût" du trajet $i \to j$;
σ	permutation définissant le trajet, $\sigma(k) = k^{\text{e}}$ ville rencontrée.

$$\min_{\sigma\in\mathfrak{S}_{10}} \sum_{i=1}^{9} c[\sigma(i),\sigma(i+1)]$$

Plan

II. Optimisation locale sans contrainte – théorie

Problème modèle de l'optimisation *continue sans contrainte* :

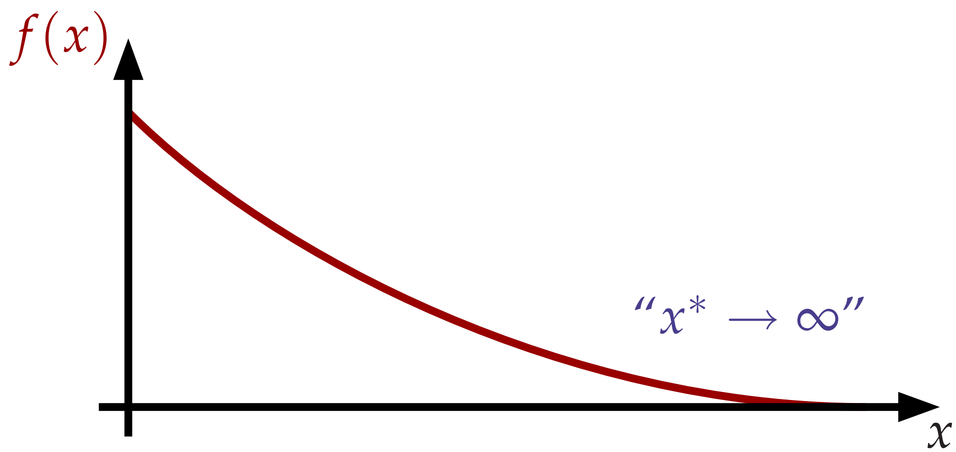
$$\min_{x \in \mathbb{R}^d} f(x).$$

II. Optimisation locale sans contrainte – théorie

Problème modèle de l'optimisation *continue sans contrainte* :

$$\min_{x \in \mathbb{R}^d} f(x).$$

◇ l'existence d'une solution x^* n'est pas assurée ;

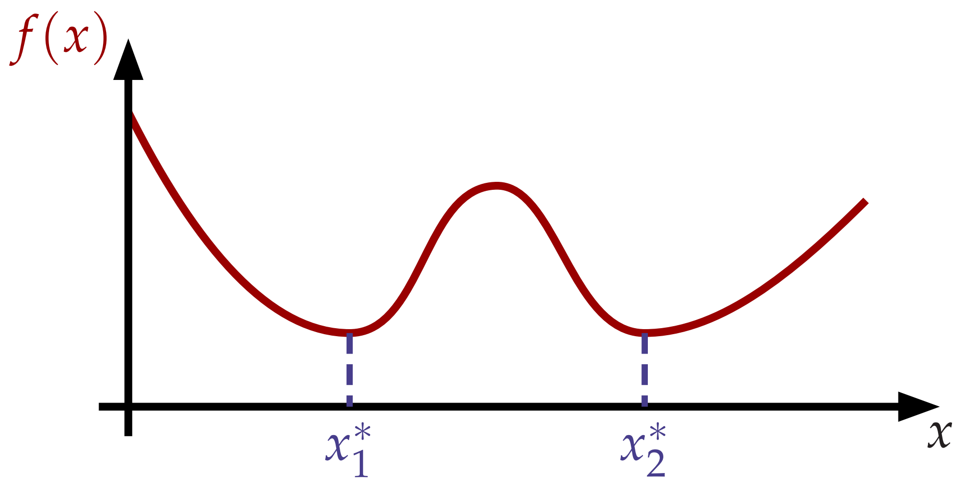


II. Optimisation locale sans contrainte – théorie

Problème modèle de l'optimisation *continue sans contrainte* :

$$\min_{x \in \mathbb{R}^d} f(x).$$

- l'existence d'une solution x^* n'est pas assurée ;
- l'unicité non plus !

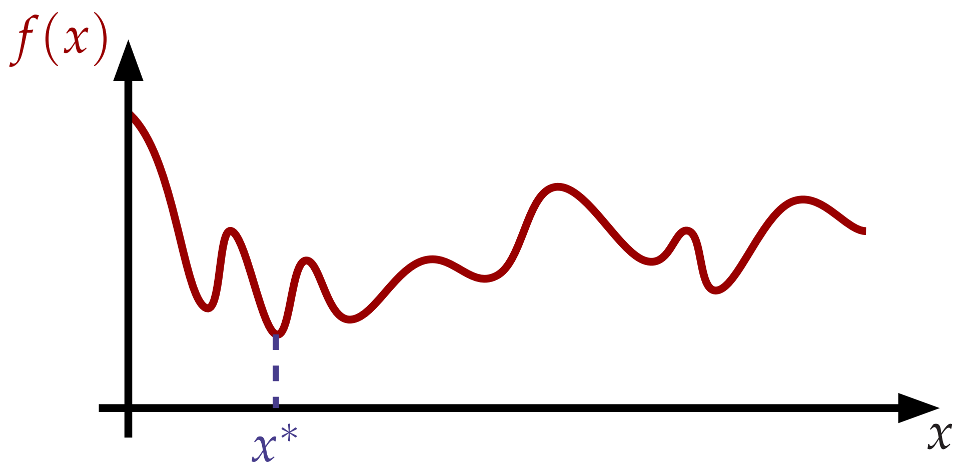


II. Optimisation locale sans contrainte – théorie

Problème modèle de l'optimisation *continue sans contrainte* :

$$\min_{x \in \mathbb{R}^d} f(x).$$

- $\diamond$ l'existence d'une solution x^* n'est pas assurée ;
- $\diamond$ l'unicité non plus !
- $\diamond$ possibilité de nombreux minima locaux non globaux...



II. Optimisation locale sans contrainte – théorie

Problème modèle de l'optimisation *continue sans contrainte* :

$$\min_{x \in \mathbb{R}^d} f(x).$$

- ◇ l'existence d'une solution x^* n'est pas assurée ;
- ◇ l'unicité non plus !
- ◇ possibilité de nombreux minima locaux non globaux...

Ces remarques mathématiques ont une incidence

sur les calculs numériques !

Plan

II. Optimisation locale sans contrainte – Méthode de Newton

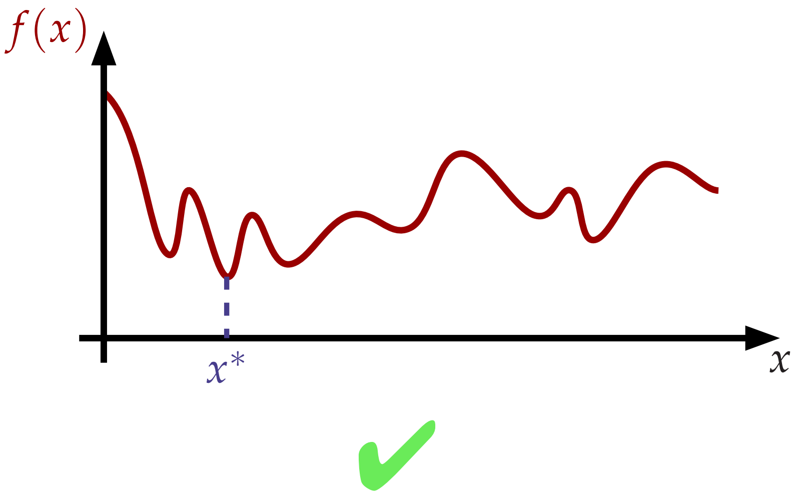
Théorème. Soit x^* une solution au problème $\min_{x \in \mathbb{R}^d} f(x)$.

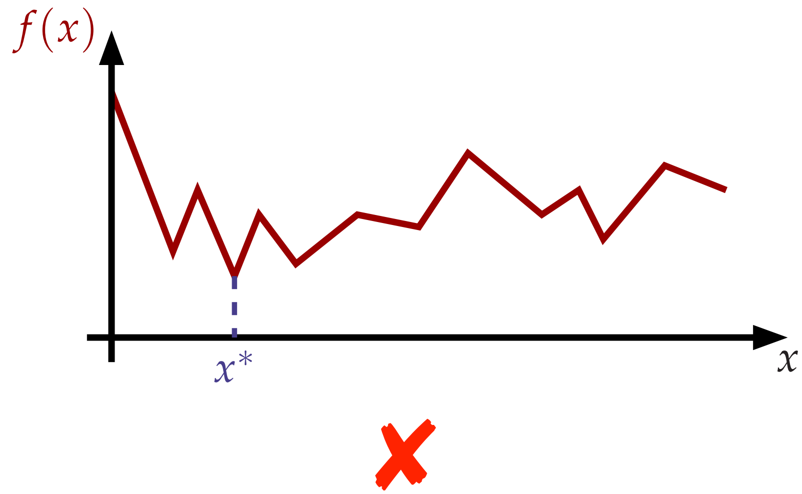
Alors $\nabla f(x^*) = 0$ *(équation d'Euler)*.

II. Optimisation locale sans contrainte – Méthode de Newton

> **Théorème.** Soit x^* une solution au problème $\min_{x \in \mathbb{R}^d} f(x)$.
>
> Alors $\nabla f(x^*) = 0$ *(équation d'Euler)*.

💣 *la fonction f doit être dérivable ;*

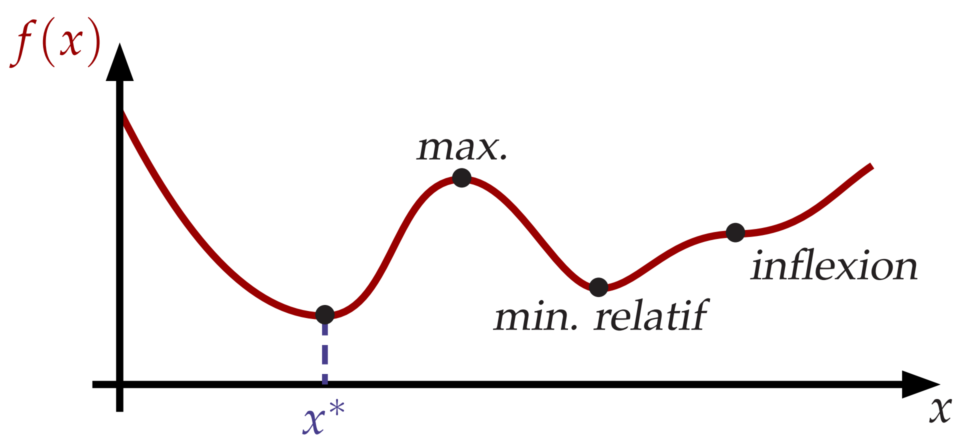




II. Optimisation locale sans contrainte – Méthode de Newton

> **Théorème.** Soit x^* une solution au problème $\min_{x \in \mathbb{R}^d} f(x)$.
>
> Alors $\nabla f(x^*) = 0$ *(équation d'Euler)*.

- *la fonction f doit être dérivable ;*
- *la condition n'est que nécessaire.*



II. Optimisation locale sans contrainte – Méthode de Newton

Théorème. Soit x^* une solution au problème $\min_{x \in \mathbb{R}^d} f(x)$.

Alors $\nabla f(x^*) = 0$ *(équation d'Euler)*.

- 💣 *la fonction f doit être dérivable ;*
- 💣 *la condition n'est que nécessaire.*

Considérer l'*équation* $\nabla f(x^*) = 0$

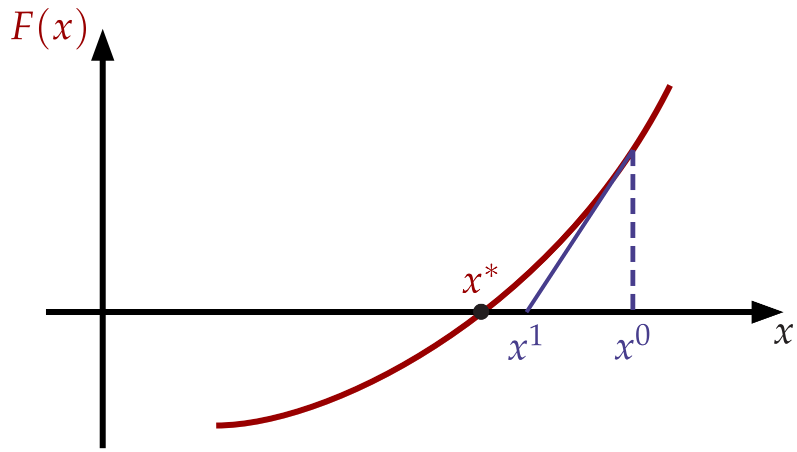
revient à chercher les points critiques.

II. Optimisation locale sans contrainte – Méthode de Newton

On note $F = \nabla f$, et on considère l'équation $F(x) = 0$, avec $F : \mathbb{R}^d \to \mathbb{R}^d$.

En dimension $d = 1$,

$$\left| \begin{array}{l} x^0 \text{ donné} \\ x^{n+1} = x^n - \dfrac{F(x^n)}{F'(x^n)}. \end{array} \right.$$



Résultat. Sous les hypothèses

$$x^0 \text{ est suffisamment proche de } x^* \quad \text{et} \quad F'(x^*) \neq 0$$

la suite (x^n) converge vers x^* quand $n \to \infty$.

Convergence extrêmement rapide : la précision double à chaque itération.

II. Optimisation locale sans contrainte – Méthode de Newton

On note $F = \nabla f$, et on considère l'équation $F(x) = 0$, avec $F : \mathbb{R}^d \to \mathbb{R}^d$.

En dimension $d = 1$,

$$\left| \begin{array}{l} x^0 \text{ donné} \\[2ex] x^{n+1} = x^n - \dfrac{F(x^n)}{F'(x^n)}. \end{array} \right.$$

En dimension d quelconque,

$$\left| \begin{array}{l} x^0 \text{ donné} \\[2ex] x^{n+1} = x^n - [F'(x^n)]^{-1} F(x^n). \end{array} \right.$$

$F'(x^n)$: *matrice des dérivées partielles de F.*

Résultat. Sous les hypothèses

$$x^0 \text{ est suffisamment proche de } x^* \quad \text{et} \quad F'(x^*) \text{ inversible,}$$

la suite (x^n) converge vers x^* quand $n \to \infty$.

Convergence extrêmement rapide : la précision double à chaque itération.

II. Optimisation locale sans contrainte – Méthode de Newton

On note $F = \nabla f$, et on considère l'équation $F(x) = 0$, avec $F : \mathbb{R}^d \to \mathbb{R}^d$.

$$x^{n+1} = x^n - [F'(x^n)]^{-1} F(x^n).$$

- $F'(x^n) = \nabla^2 f(x^n)$ est une matrice de taille $d \times d$:

$$[F'(x^n)]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x^n).$$

- L'inverse $[F'(x^n)]^{-1}$ n'est pas calculée explicitement, on résout

$$F'(x^n)(x^{n+1} - x^n) = -F(x^n).$$

- Variante : on met à jour F' périodiquement.
- La convergence n'est que *locale* : choix crucial de x^0.
- Critères d'arrêt :

$$|F(x^n)| < \varepsilon \; ✗ \qquad \qquad \|x^{n+1} - x^n\| < \varepsilon \; ✓$$

Plan

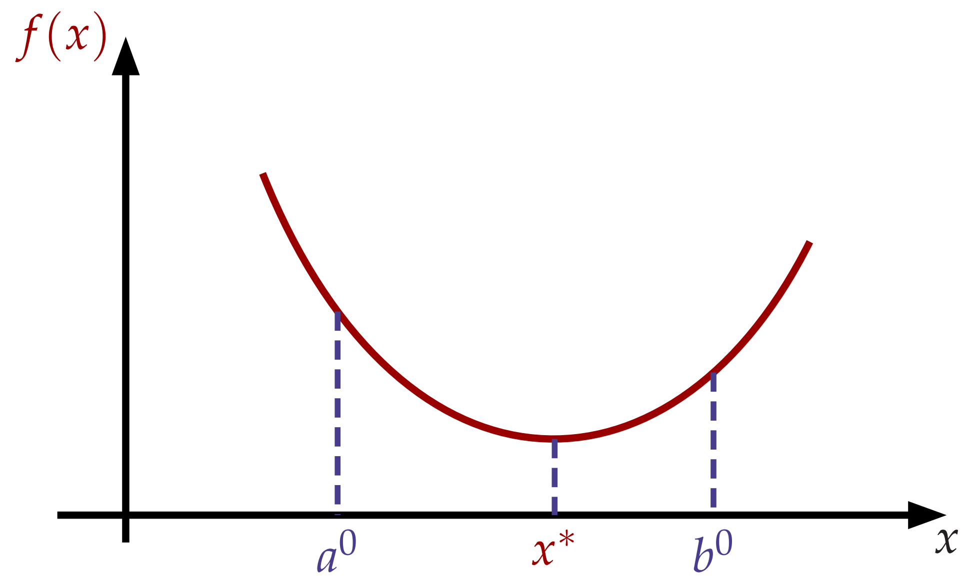
II. Optimisation locale sans contrainte – Dichotomie en 1D

Cas de la dimension 1

◇ Si x^* résout $\min\limits_{x \in \mathbb{R}^d} f(x)$, alors autour de x^*, $\begin{cases} f'(x) < 0 \text{ pour } x < x^*, \\ f'(x) > 0 \text{ pour } x > x^*. \end{cases}$

◇ Méthode de dichotomie : a^0, b^0 étant donnés avec $f'(a^0) < 0 < f'(b^0)$,

$$\left|\begin{array}{l} c^n = \frac{a^n+b^n}{2} \\ \text{Si } f'(c^n) < 0,\ a^{n+1} = c^n \text{ et } b^{n+1} = b^n \\ \text{Si } f'(c^n) < 0,\ a^{n+1} = a^n \text{ et } b^{n+1} = c^n \end{array}\right.$$



II. Optimisation locale sans contrainte – Dichotomie en 1D

Cas de la dimension 1

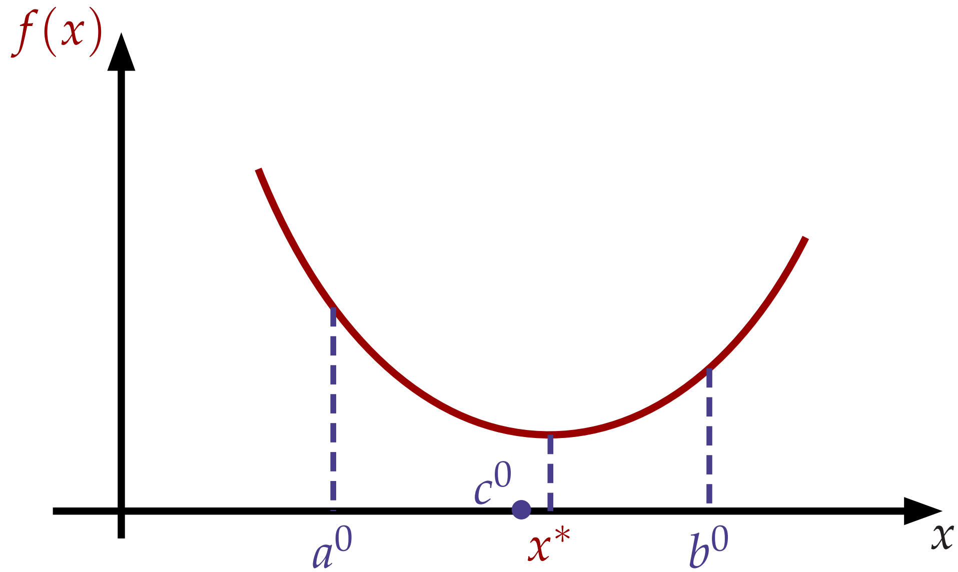
◇ Si x^* résout $\min_{x\in\mathbb{R}^d} f(x)$, alors autour de x^*, $\begin{cases} f'(x) < 0 \text{ pour } x < x^*, \\ f'(x) > 0 \text{ pour } x > x^*. \end{cases}$

◇ Méthode de dichotomie : a^0, b^0 étant donnés avec $f'(a^0) < 0 < f'(b^0)$,

$c^n = \frac{a^n+b^n}{2}$

Si $f'(c^n) < 0$, $a^{n+1} = c^n$ et $b^{n+1} = b^n$

Si $f'(c^n) < 0$, $a^{n+1} = a^n$ et $b^{n+1} = c^n$



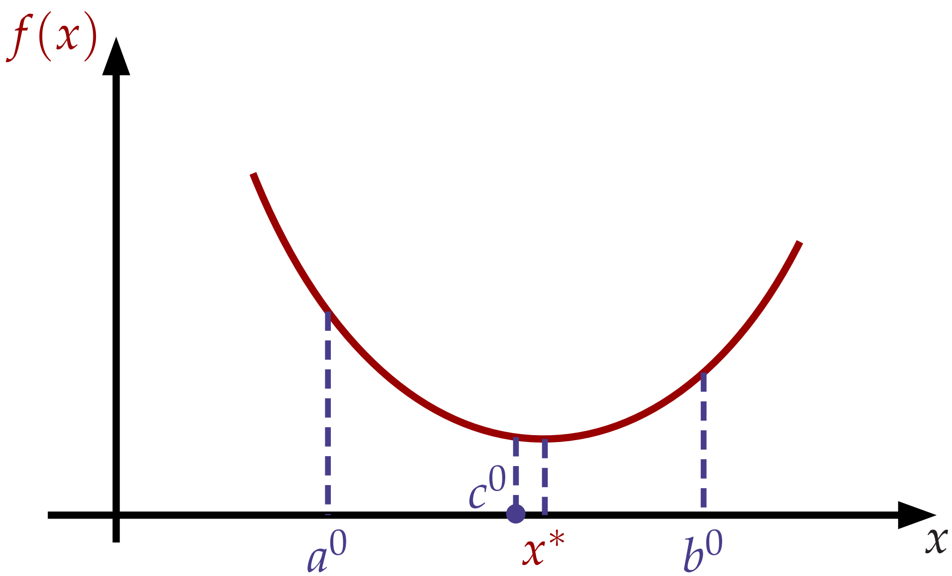
II. Optimisation locale sans contrainte – Dichotomie en 1D

Cas de la dimension 1

- Si x^* résout $\min\limits_{x \in \mathbb{R}^d} f(x)$, alors autour de x^*, $\begin{cases} f'(x) < 0 \text{ pour } x < x^*, \\ f'(x) > 0 \text{ pour } x > x^*. \end{cases}$

- Méthode de dichotomie : a^0, b^0 étant donnés avec $f'(a^0) < 0 < f'(b^0)$,

$$
\left|
\begin{array}{l}
c^n = \frac{a^n + b^n}{2} \\
\text{Si } f'(c^n) < 0,\ a^{n+1} = c^n \text{ et } b^{n+1} = b^n \\
\text{Si } f'(c^n) < 0,\ a^{n+1} = a^n \text{ et } b^{n+1} = c^n
\end{array}
\right.
$$



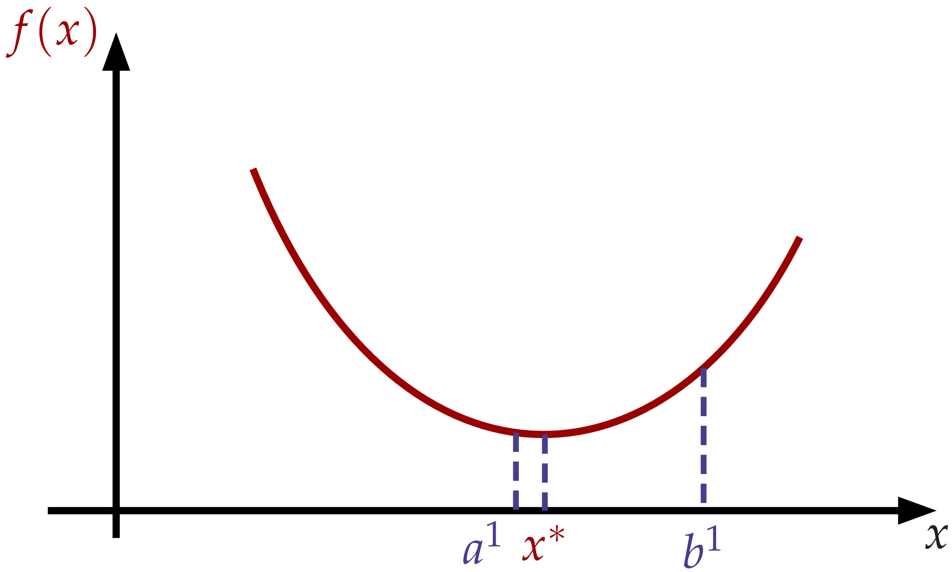
II. Optimisation locale sans contrainte – Dichotomie en 1D

Cas de la dimension 1

- Si x^* résout $\min_{x \in \mathbb{R}^d} f(x)$, alors autour de x^*, $\begin{cases} f'(x) < 0 \text{ pour } x < x^*, \\ f'(x) > 0 \text{ pour } x > x^*. \end{cases}$

- Méthode de dichotomie : a^0, b^0 étant donnés avec $f'(a^0) < 0 < f'(b^0)$,

$$
\left| \begin{array}{l} c^n = \frac{a^n + b^n}{2} \\ \text{Si } f'(c^n) < 0,\ a^{n+1} = c^n \text{ et } b^{n+1} = b^n \\ \text{Si } f'(c^n) < 0,\ a^{n+1} = a^n \text{ et } b^{n+1} = c^n \end{array} \right.
$$



II. Optimisation locale sans contrainte – Dichotomie en 1D

Cas de la dimension 1

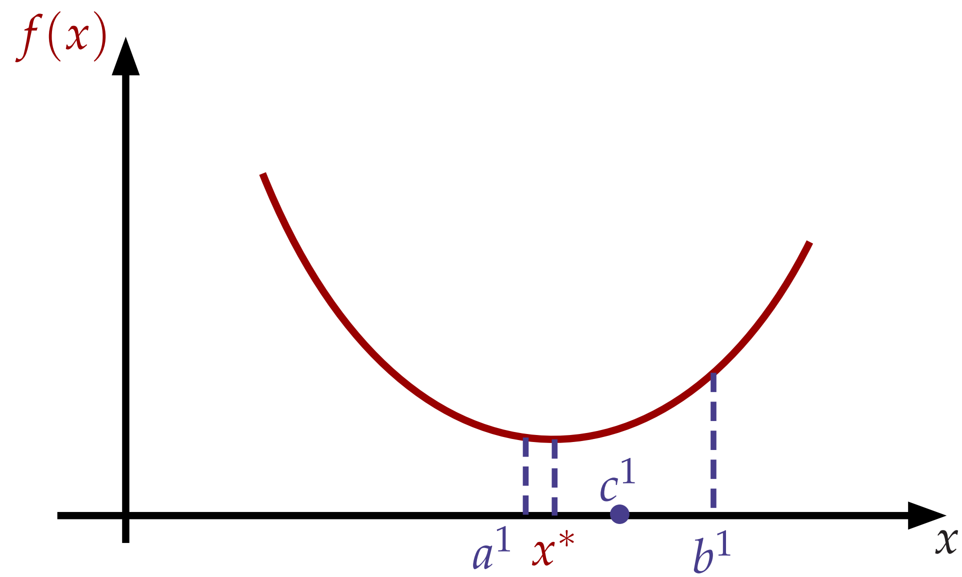
◇ Si x^* résout $\min_{x\in\mathbb{R}^d} f(x)$, alors autour de x^*, $\begin{cases} f'(x) < 0 \text{ pour } x < x^*, \\ f'(x) > 0 \text{ pour } x > x^*. \end{cases}$

◇ Méthode de dichotomie : a^0, b^0 étant donnés avec $f'(a^0) < 0 < f'(b^0)$,

$c^n = \frac{a^n+b^n}{2}$

Si $f'(c^n) < 0,\ a^{n+1} = c^n$ et $b^{n+1} = b^n$

Si $f'(c^n) < 0,\ a^{n+1} = a^n$ et $b^{n+1} = c^n$



II. Optimisation locale sans contrainte – Dichotomie en 1D

Cas de la dimension 1

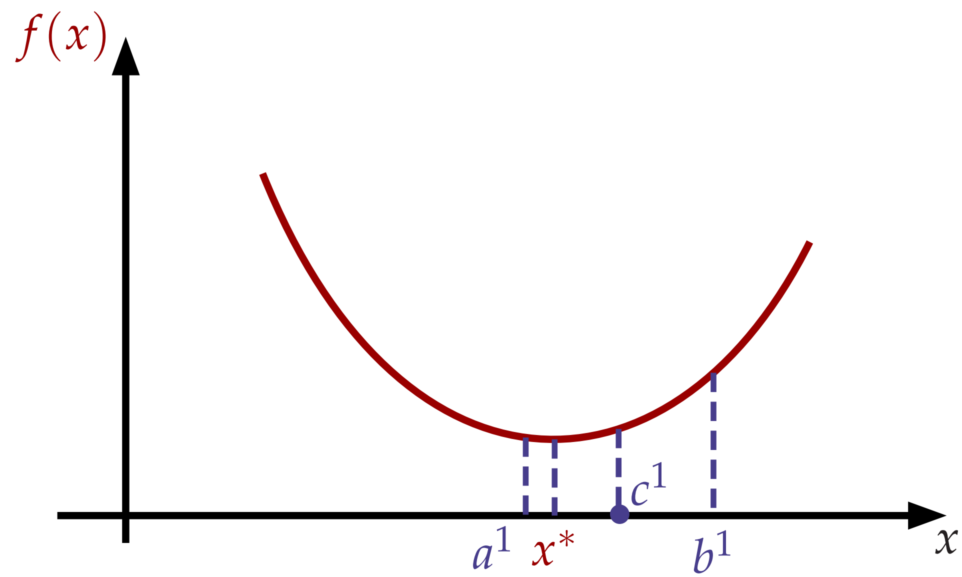
- Si x^* résout $\min_{x \in \mathbb{R}^d} f(x)$, alors autour de x^*, $\begin{cases} f'(x) < 0 \text{ pour } x < x^*, \\ f'(x) > 0 \text{ pour } x > x^*. \end{cases}$

- Méthode de dichotomie : a^0, b^0 étant donnés avec $f'(a^0) < 0 < f'(b^0)$,

$c^n = \frac{a^n+b^n}{2}$

Si $f'(c^n) < 0$, $a^{n+1} = c^n$ et $b^{n+1} = b^n$

Si $f'(c^n) < 0$, $a^{n+1} = a^n$ et $b^{n+1} = c^n$



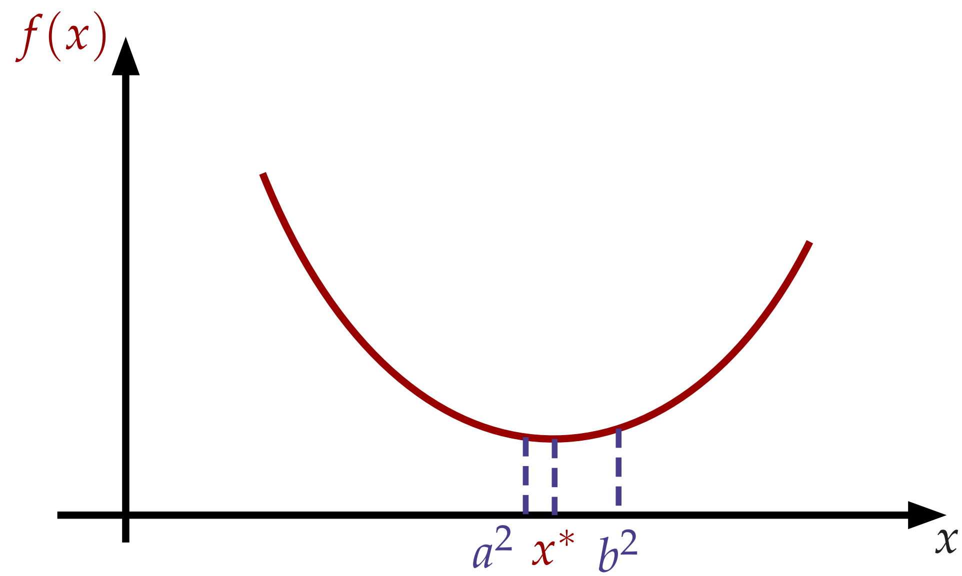
II. Optimisation locale sans contrainte – Dichotomie en 1D

Cas de la dimension 1

◇ Si x^* résout $\min_{x\in\mathbb{R}^d} f(x)$, alors autour de x^*, $\begin{cases} f'(x) < 0 \text{ pour } x < x^*, \\ f'(x) > 0 \text{ pour } x > x^*. \end{cases}$

◇ Méthode de dichotomie : a^0, b^0 étant donnés avec $f'(a^0) < 0 < f'(b^0)$,

$$\left| \begin{array}{l} c^n = \frac{a^n+b^n}{2} \\ \text{Si } f'(c^n) < 0,\ a^{n+1} = c^n \text{ et } b^{n+1} = b^n \\ \text{Si } f'(c^n) < 0,\ a^{n+1} = a^n \text{ et } b^{n+1} = c^n \end{array} \right.$$



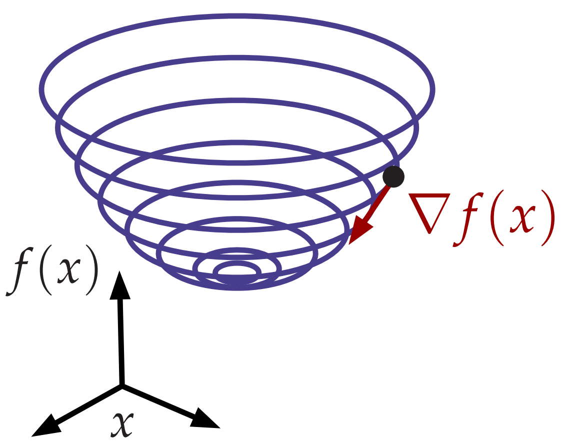
II. Optimisation locale sans contrainte – Méthodes de gradient

- On travaille directement sur la formulation $\min_{x \in \mathbb{R}^d} f(x)$.

- Partant de x^0, on construit la suite (x^n) par

$$x^{n+1} = x^n - \underbrace{\rho_n}_{\textit{pas}} \underbrace{d^n}_{\textit{direction de descente}}$$

- Comment choisir d^n et ρ_n pour assurer $f(x^{n+1}) < f(x^n)$?



Direction de plus grande pente : $d^n = \nabla f(x^n)$.

$$f(x^{n+1}) \simeq f(x^n) - \rho_n \langle d^n, d^n \rangle.$$

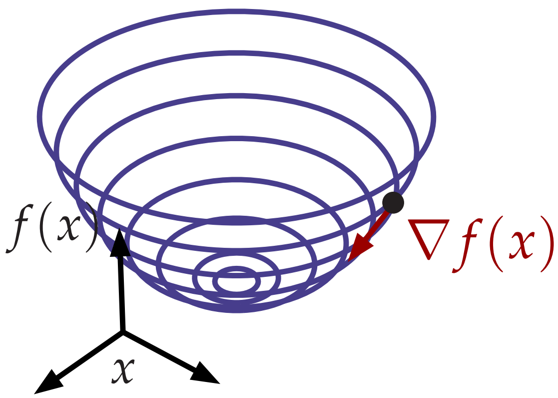
II. Optimisation locale sans contrainte – Méthodes de gradient

- On travaille directement sur la formulation $\min\limits_{x \in \mathbb{R}^d} f(x)$.

- Partant de x^0, on construit la suite (x^n) par

$$x^{n+1} = x^n - \underbrace{\rho_n}_{\textit{pas}} \underbrace{d^n}_{\textit{direction de descente}}$$

- Comment choisir d^n et ρ_n pour assurer $f(x^{n+1}) < f(x^n)$?



Direction de plus grande pente : $d^n = \nabla f(x^n)$.

$$f(x^{n+1}) \simeq f(x^n) - \rho_n \langle d^n, d^n \rangle.$$

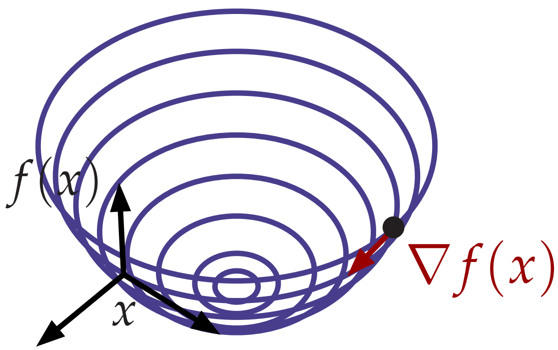
II. Optimisation locale sans contrainte – Méthodes de gradient

- On travaille directement sur la formulation $\min_{x\in\mathbb{R}^d} f(x)$.

- Partant de x^0, on construit la suite (x^n) par

$$x^{n+1} = x^n - \underbrace{\rho_n}_{\textit{pas}}\,\underbrace{d^n}_{\textit{direction de descente}}$$

- Comment choisir d^n et ρ_n pour assurer $f(x^{n+1}) < f(x^n)$?



Direction de plus grande pente : $d^n = \nabla f(x^n)$.

$$f(x^{n+1}) \simeq f(x^n) - \rho_n \langle d^n, d^n \rangle.$$

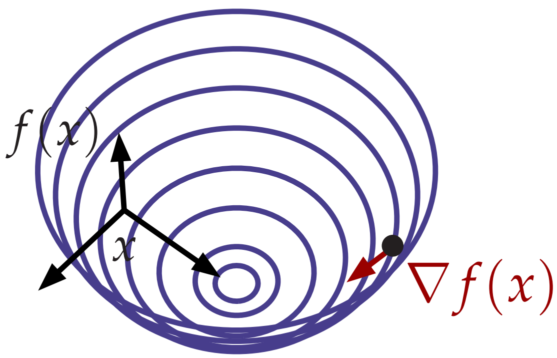
II. Optimisation locale sans contrainte – Méthodes de gradient

◇ On travaille directement sur la formulation $\min_{x\in\mathbb{R}^d} f(x)$.

◇ Partant de x^0, on construit la suite (x^n) par

$$x^{n+1} = x^n - \underbrace{\rho_n}_{\textit{pas}} \underbrace{d^n}_{\textit{direction de descente}}$$

◇ Comment choisir d^n et ρ_n pour assurer $f(x^{n+1}) < f(x^n)$?



Direction de plus grande pente : $d^n = \nabla f(x^n)$.

$$f(x^{n+1}) \simeq f(x^n) - \rho_n \langle d^n, d^n \rangle.$$

II. Optimisation locale sans contrainte – Méthodes de gradient

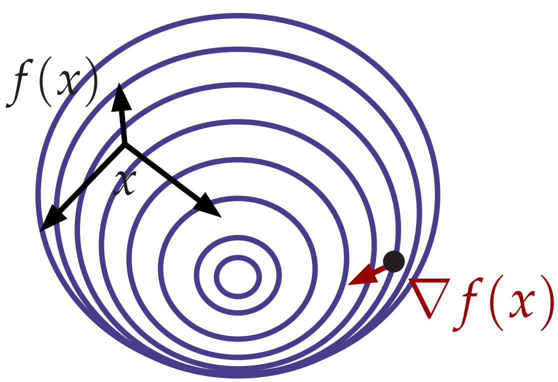
◇ On travaille directement sur la formulation $\min\limits_{x \in \mathbb{R}^d} f(x)$.

◇ Partant de x^0, on construit la suite (x^n) par

$$x^{n+1} = x^n - \rho_n d^n$$

pas *direction de descente*

◇ Comment choisir d^n et ρ_n pour assurer $f(x^{n+1}) < f(x^n)$?



Direction de plus grande pente : $d^n = \nabla f(x^n)$.

$$f(x^{n+1}) \simeq f(x^n) - \rho_n \langle d^n, d^n \rangle.$$

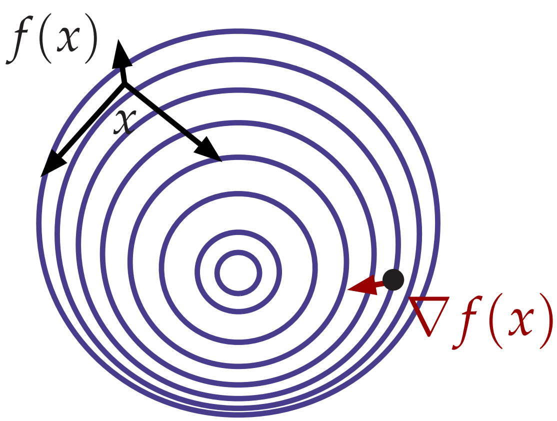
II. Optimisation locale sans contrainte – Méthodes de gradient

◇ On travaille directement sur la formulation $\min_{x \in \mathbb{R}^d} f(x)$.

◇ Partant de x^0, on construit la suite (x^n) par

$$x^{n+1} = x^n - \underbrace{\rho_n}_{\textit{pas}} \underbrace{d^n}_{\textit{direction de descente}}$$

◇ Comment choisir d^n et ρ_n pour assurer $f(x^{n+1}) < f(x^n)$?



Direction de plus grande pente : $d^n = \nabla f(x^n)$.

$$f(x^{n+1}) \simeq f(x^n) - \rho_n \langle d^n, d^n \rangle.$$

II. Optimisation locale sans contrainte – Méthodes de gradient

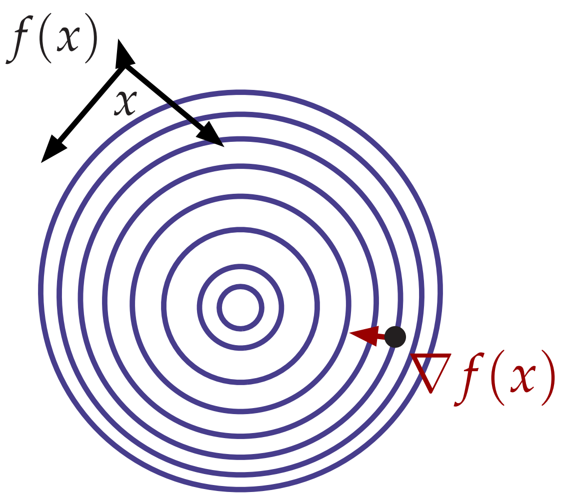
◇ On travaille directement sur la formulation $\min_{x \in \mathbb{R}^d} f(x)$.

◇ Partant de x^0, on construit la suite (x^n) par

$$x^{n+1} = x^n - \rho_n d^n$$

pas *direction de descente*

◇ Comment choisir d^n et ρ_n pour assurer $f(x^{n+1}) < f(x^n)$?



Direction de plus grande pente : $d^n = \nabla f(x^n)$.

$$f(x^{n+1}) \simeq f(x^n) - \rho_n \langle d^n, d^n \rangle.$$

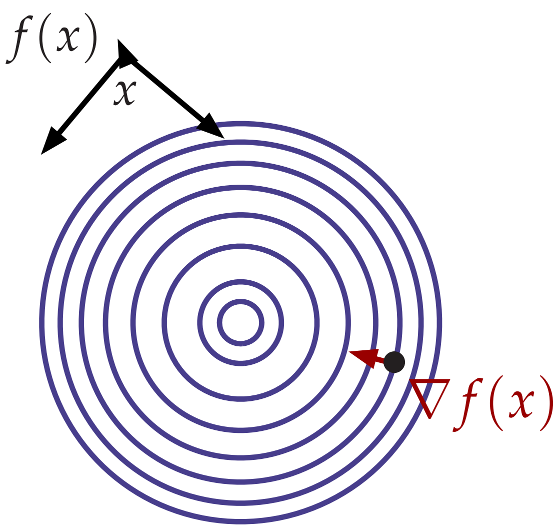
II. Optimisation locale sans contrainte – Méthodes de gradient

- On travaille directement sur la formulation $\min\limits_{x\in\mathbb{R}^d} f(x)$.

- Partant de x^0, on construit la suite (x^n) par

$$x^{n+1} = x^n - \underbrace{\rho_n}_{\textit{pas}} \underbrace{d^n}_{\textit{direction de descente}}$$

- Comment choisir d^n et ρ_n pour assurer $f(x^{n+1}) < f(x^n)$?



Direction de plus grande pente : $d^n = \nabla f(x^n)$.

$$f(x^{n+1}) \simeq f(x^n) - \rho_n \langle d^n, d^n \rangle.$$

II. Optimisation locale sans contrainte – Méthodes de gradient

Méthode du gradient à pas constant

on choisit $\rho_n = \rho$ *:*

$$x^{n+1} = x^n - \rho \nabla f(x^n).$$

> **Résultat.** Sous certaines hypothèses sur la fonction f, et si ρ est assez petit, la suite (x^n) converge vers un point de minimum relatif de f.

◇ Si ρ est trop petit, la convergence est très lente.

◇ Rien n'assure que le minimum soit absolu.

◇ Forte dépendance vis-à-vis de l'initialisation x^0.

II. Optimisation locale sans contrainte – Méthodes de gradient

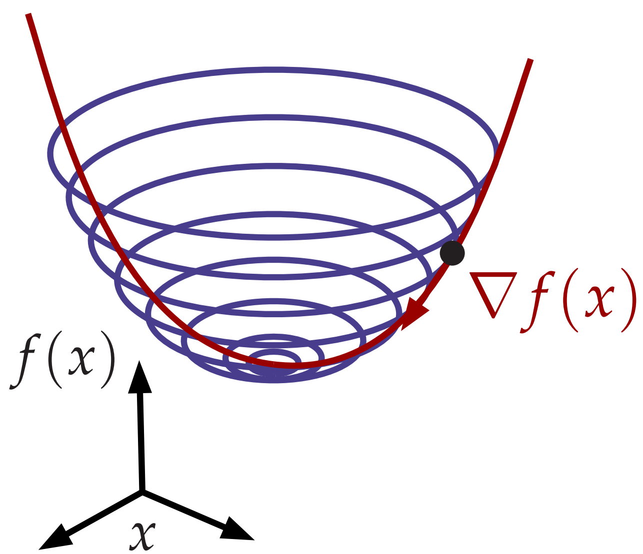
Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\big(x^n - \rho_n \nabla f(x^n)\big) = \min_{\rho \in \mathbb{R}} f\big(x^n - \rho \nabla f(x^n)\big),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$



À chaque étape, on minimise

dans la *direction* de $d^n = \nabla f(x^n)$.

II. Optimisation locale sans contrainte – Méthodes de gradient

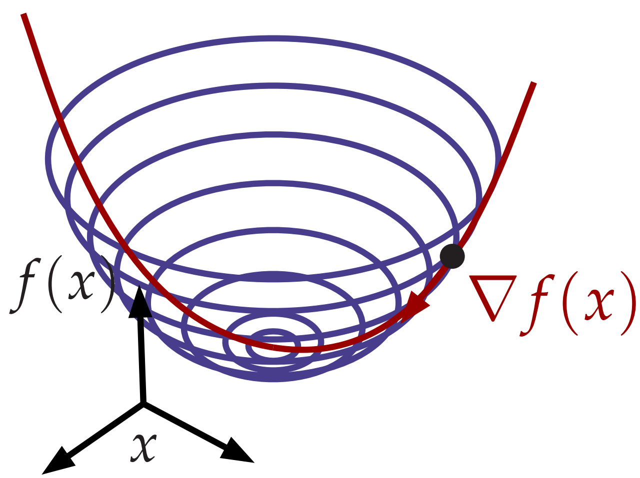
Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\big(x^n - \rho_n \nabla f(x^n)\big) = \min_{\rho \in \mathbb{R}} f\big(x^n - \rho \nabla f(x^n)\big),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$



À chaque étape, on minimise dans la *direction* de $d^n = \nabla f(x^n)$.

II. Optimisation locale sans contrainte – Méthodes de gradient

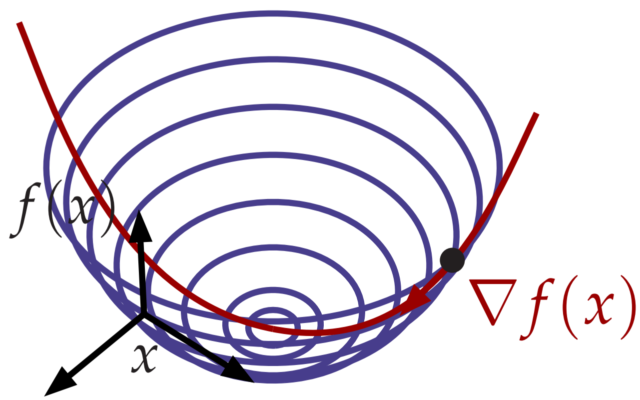
Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\left(x^n - \rho_n \nabla f(x^n)\right) = \min_{\rho \in \mathbb{R}} f\left(x^n - \rho \nabla f(x^n)\right),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$



À chaque étape, on minimise

dans la *direction* de $d^n = \nabla f(x^n)$.

II. Optimisation locale sans contrainte – Méthodes de gradient

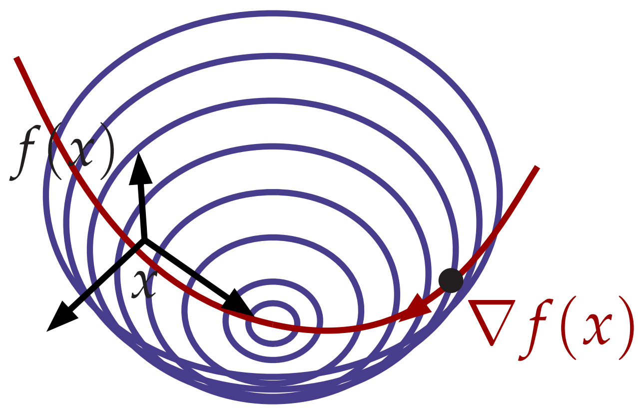
Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\left(x^n - \rho_n \nabla f(x^n)\right) = \min_{\rho \in \mathbb{R}} f\left(x^n - \rho \nabla f(x^n)\right),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$



À chaque étape, on minimise

dans la *direction* de $d^n = \nabla f(x^n)$.

II. Optimisation locale sans contrainte – Méthodes de gradient

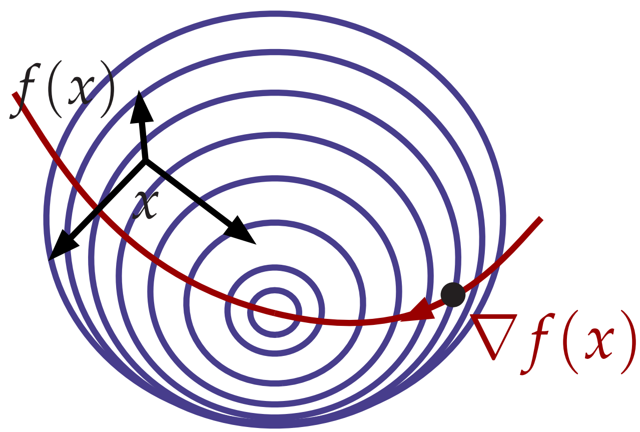
Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\left(x^n - \rho_n \nabla f(x^n)\right) = \min_{\rho \in \mathbb{R}} f\left(x^n - \rho \nabla f(x^n)\right),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$



À chaque étape, on minimise

dans la *direction* de $d^n = \nabla f(x^n)$.

II. Optimisation locale sans contrainte – Méthodes de gradient

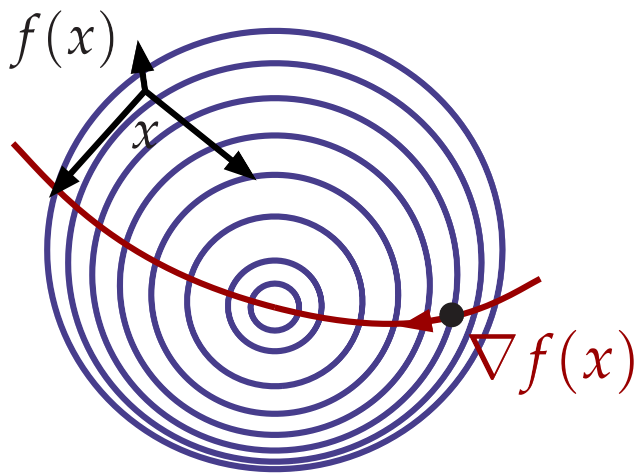
Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\left(x^n - \rho_n \nabla f(x^n)\right) = \min_{\rho \in \mathbb{R}} f\left(x^n - \rho \nabla f(x^n)\right),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$



À chaque étape, on minimise dans la *direction* de $d^n = \nabla f(x^n)$.

II. Optimisation locale sans contrainte – Méthodes de gradient

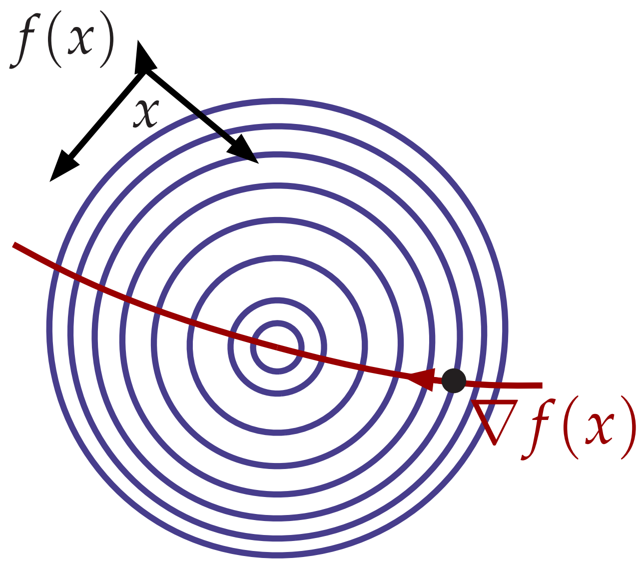
Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\left(x^n - \rho_n \nabla f(x^n)\right) = \min_{\rho \in \mathbb{R}} f\left(x^n - \rho \nabla f(x^n)\right),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$



À chaque étape, on minimise

dans la *direction* de $d^n = \nabla f(x^n)$.

II. Optimisation locale sans contrainte – Méthodes de gradient

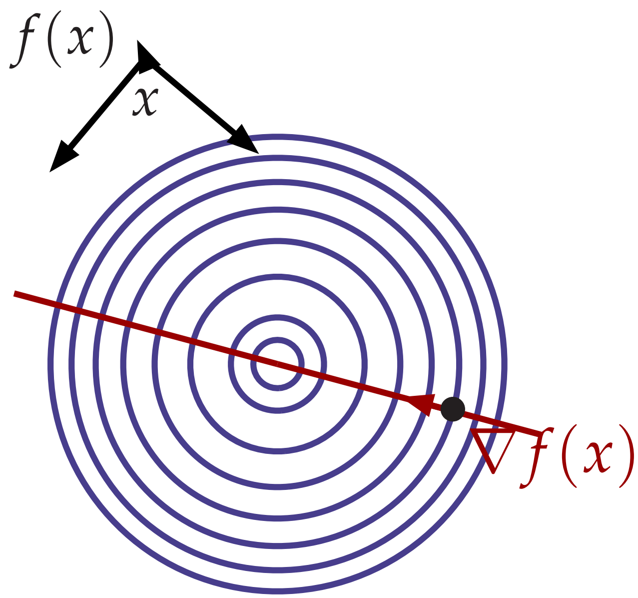
Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\left(x^n - \rho_n \nabla f(x^n)\right) = \min_{\rho \in \mathbb{R}} f\left(x^n - \rho \nabla f(x^n)\right),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$



À chaque étape, on minimise

dans la *direction* de $d^n = \nabla f(x^n)$.

II. Optimisation locale sans contrainte – Méthodes de gradient

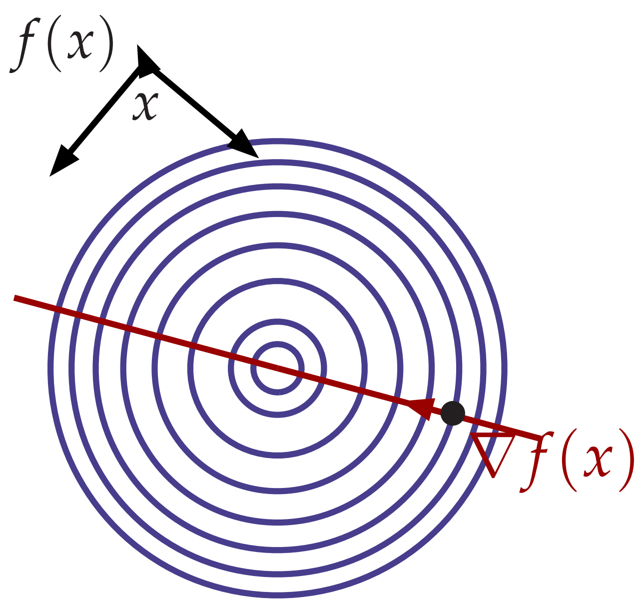
Méthode du gradient à pas optimal

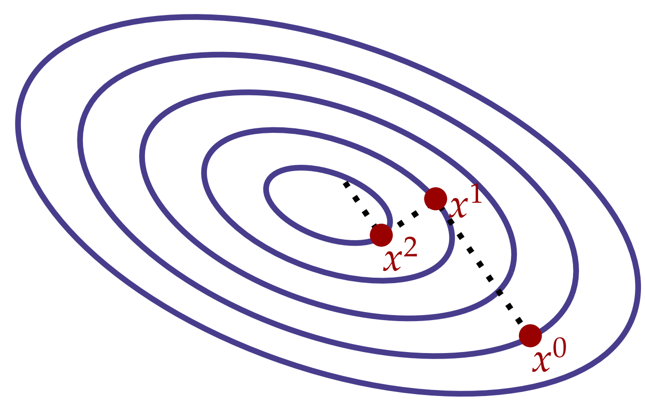
on choisit ρ_n *tel que*

$$f\big(x^n - \rho_n \nabla f(x^n)\big) = \min_{\rho \in \mathbb{R}} f\big(x^n - \rho \nabla f(x^n)\big),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$





Un cas plus général

II. Optimisation locale sans contrainte – Méthodes de gradient

Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\big(x^n - \rho_n \nabla f(x^n)\big) = \min_{\rho \in \mathbb{R}} f\big(x^n - \rho \nabla f(x^n)\big),$$

et x^{n+1} est défini comme

$$x^{n+1} = x^n - \rho_n \nabla f(x^n).$$

> **Résultat** Sous certaines hypothèses sur la fonction f, la suite (x^n) converge vers un point de minimum relatif de f.

- Avantage : le pas est choisi automatiquement.
- Mêmes défauts vis-à-vis des minima locaux.

II. Optimisation locale sans contrainte – Méthodes de gradient

Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\left(x^n - \rho_n \nabla f(x^n)\right) = \min_{\rho \in \mathbb{R}} f\left(x^n - \rho \nabla f(x^n)\right).$$

Résultat Deux directions de descente successives sont orthogonales.

Preuve. La fonction $\varphi(\rho) = f\left(x^n - \rho \nabla f(x^n)\right)$ admet un minimum sur $\mathbb{R}$ en $\rho = \rho_n$, donc

$$\varphi'(\rho_n) = 0.$$

Mais

$$\varphi'(\rho) = \left\langle \nabla f\left(x^n - \rho \nabla f(x^n)\right), \nabla f(x^n) \right\rangle.$$

Pour $\rho = \rho_n$, on obtient

$$\langle \nabla f(x^{n+1}), \nabla f(x^n) \rangle = 0, \text{ i.e. } \langle d^{n+1}, d^n \rangle = 0. \qquad \square$$

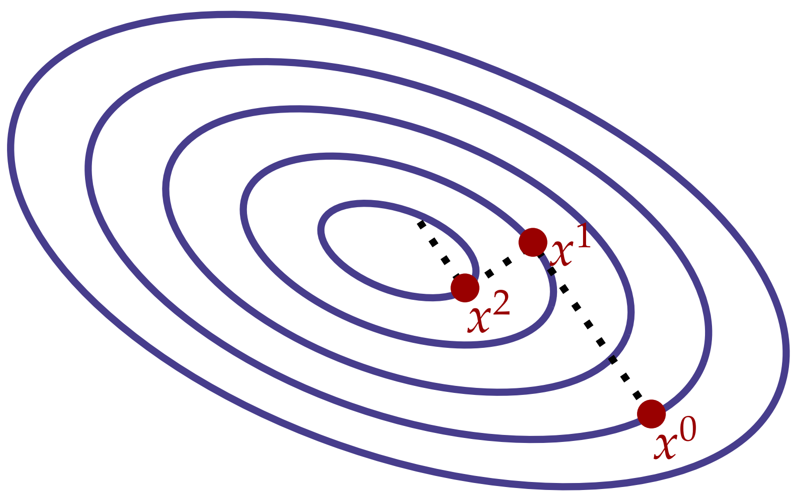
II. Optimisation locale sans contrainte – Méthodes de gradient

Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\left(x^n - \rho_n \nabla f(x^n)\right) = \min_{\rho \in \mathbb{R}} f\left(x^n - \rho \nabla f(x^n)\right).$$

Résultat Deux directions de descente successives sont orthogonales.



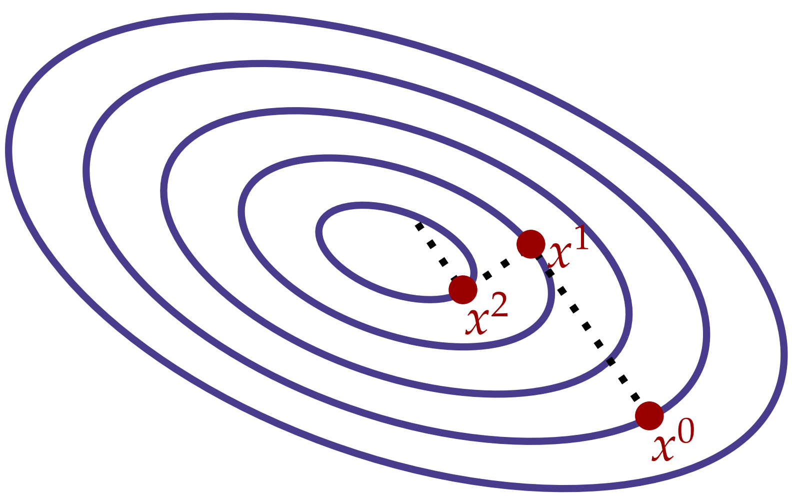
II. Optimisation locale sans contrainte – Méthodes de gradient

Méthode du gradient à pas optimal

on choisit ρ_n *tel que*

$$f\big(x^n - \rho_n \nabla f(x^n)\big) = \min_{\rho \in \mathbb{R}} f\big(x^n - \rho \nabla f(x^n)\big).$$

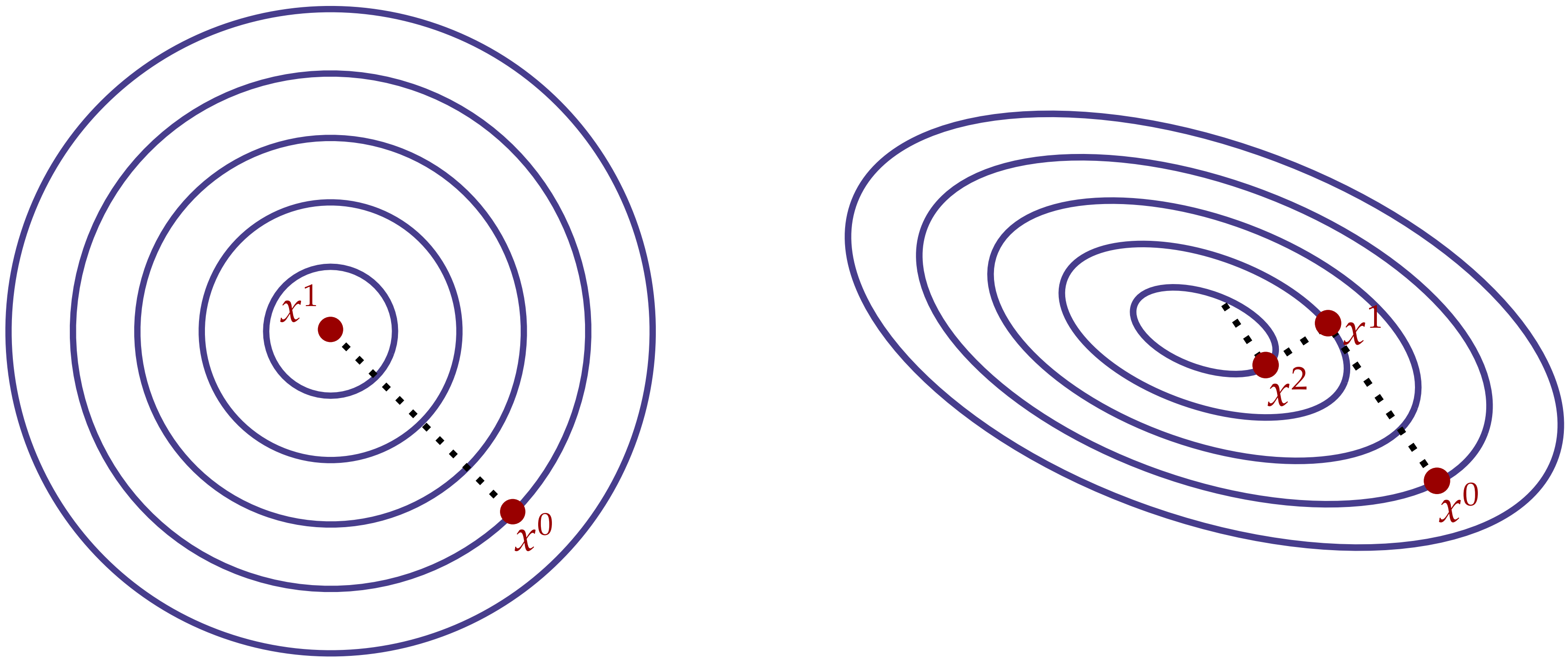
Résultat Deux directions de descente successives sont orthogonales.



La direction de plus grande pente ne “pointe pas” vers le minimum...

II. Optimisation locale sans contrainte – Méthodes de gradient

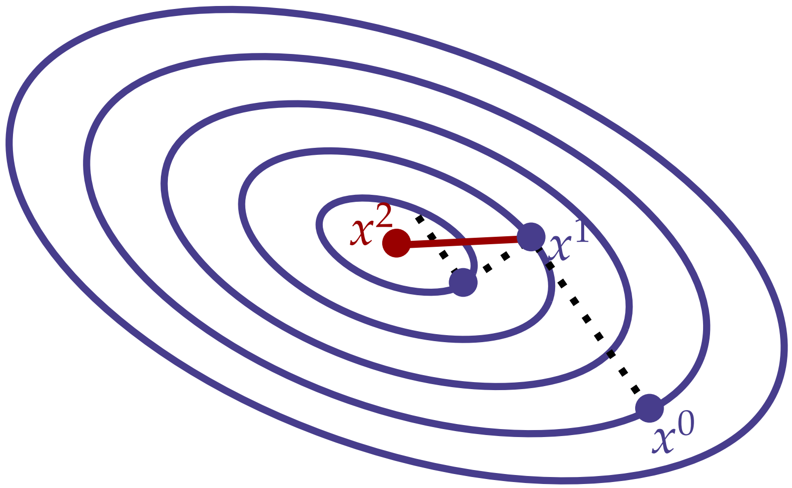
Méthode du gradient conjugué



Pourquoi a-t-on convergence en 1 itération dans un cas et pas dans l'autre ?

II. Optimisation locale sans contrainte – Méthodes de gradient

Méthode du gradient conjugué



Idée : *les directions de descente doivent être "orthogonales" de manière adaptée à la fonction f.*

Algorithme.

x^0 donné, $d^0 = r^0 = -\nabla f(x^0)$,

ρ_n minimise $f(x^n + \rho_n d^n)$,

$$x^{n+1} = x^n + \rho_n d^n,$$

$$r^{n+1} = -\nabla f(x^{n+1}),$$

$$\beta^{n+1} = \|r^{n+1}\|^2 / \|r^n\|^2,$$

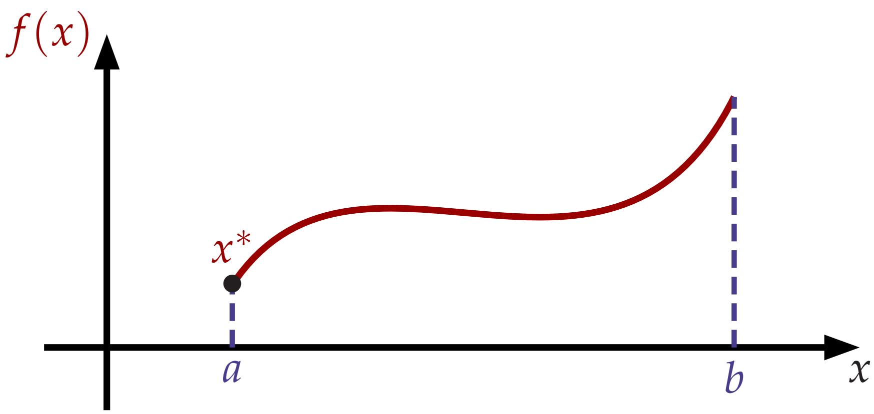
$$d^{n+1} = r^{n+1} + \beta^{n+1} d^n.$$

Plan

III. Prise en compte de contraintes – Généralités

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

- ◇ La condition $\nabla f(x^*) = 0$ n'est plus nécessaire !



$$\min_{x \in [a,b]} f(x) = f(a),$$

donc $x^* = a$, mais

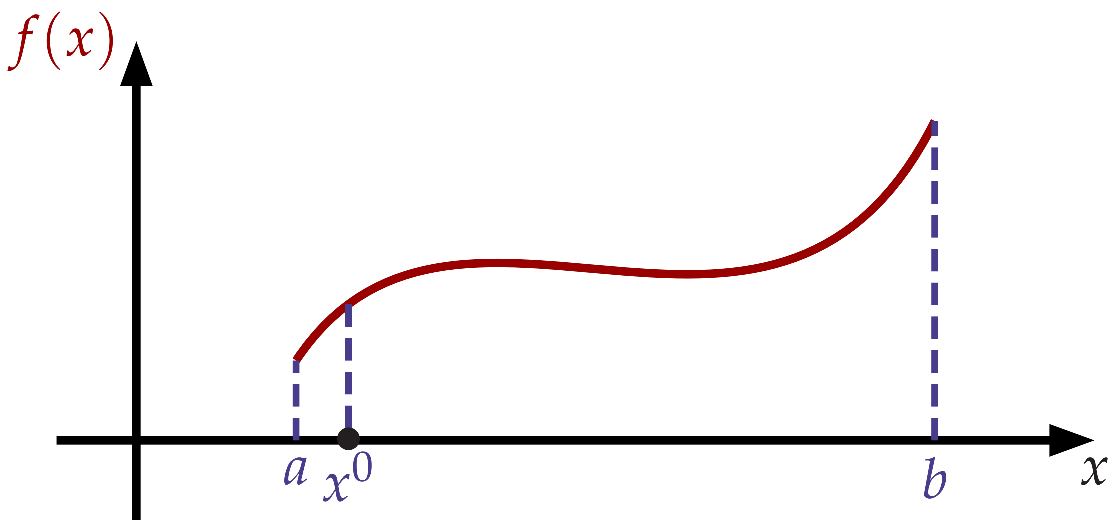
$$f'(x^*) \neq 0 \,!$$

La méthode de Newton ne s'applique plus...

III. Prise en compte de contraintes – Généralités

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

- La condition $\nabla f(x^*) = 0$ n'est plus nécessaire !
- Les itérés des méthodes de descente ne restent pas dans K !

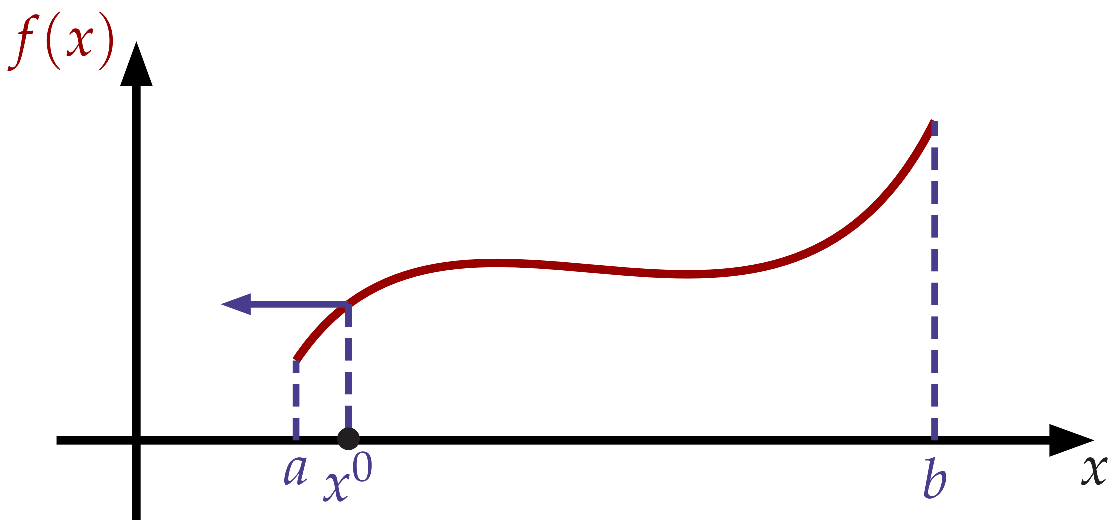


III. Prise en compte de contraintes – Généralités

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

◇ La condition $\nabla f(x^*) = 0$ n'est plus nécessaire !

◇ Les itérés des méthodes de descente ne restent pas dans K !

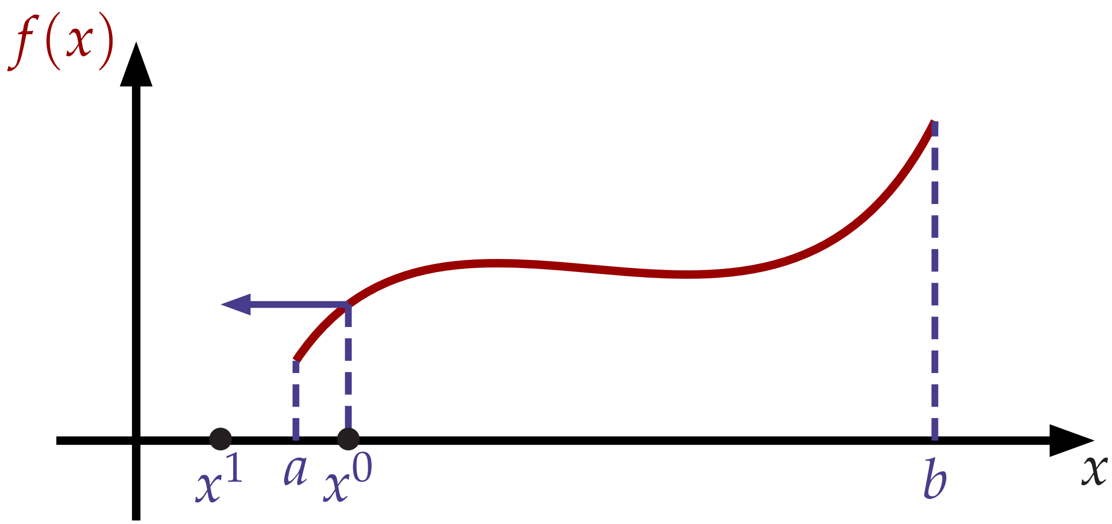


III. Prise en compte de contraintes – Généralités

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

◇ La condition $\nabla f(x^*) = 0$ n'est plus nécessaire !

◇ Les itérés des méthodes de descente ne restent pas dans K !



Plan

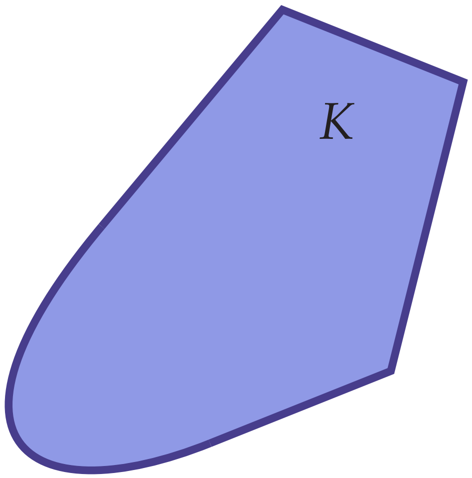
III. Prise en compte de contraintes – Gradient projeté

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

◇ On note p_K la projection sur l'ensemble K (convexe).

◇ Méthode du gradient projeté :

$$\left| \begin{array}{l} x^0 \text{ donné,} \\ x^{n+1} = p_K\big(x^n - \rho \nabla f(x^n)\big). \end{array} \right.$$



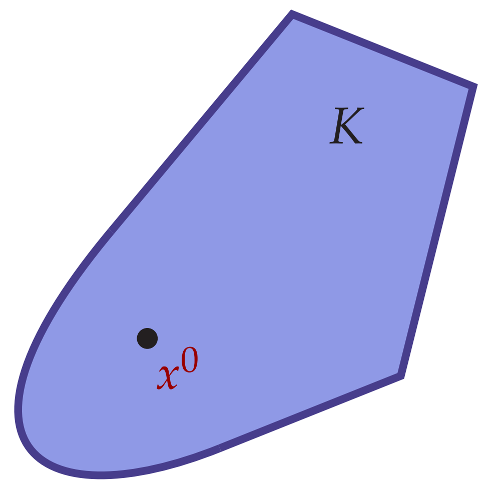
III. Prise en compte de contraintes – Gradient projeté

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

- On note p_K la projection sur l'ensemble K (convexe).

- Méthode du gradient projeté :

$$\left| \begin{array}{l} x^0 \text{ donné}, \\ x^{n+1} = p_K\big(x^n - \rho \nabla f(x^n)\big). \end{array} \right.$$



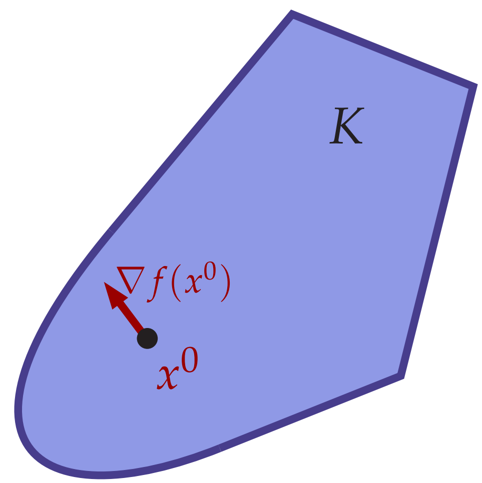
III. Prise en compte de contraintes – Gradient projeté

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

◇ On note p_K la projection sur l'ensemble K (convexe).

◇ Méthode du gradient projeté :

$$\left| \begin{array}{l} x^0 \text{ donné,} \\ x^{n+1} = p_K\big(x^n - \rho \nabla f(x^n)\big). \end{array} \right.$$



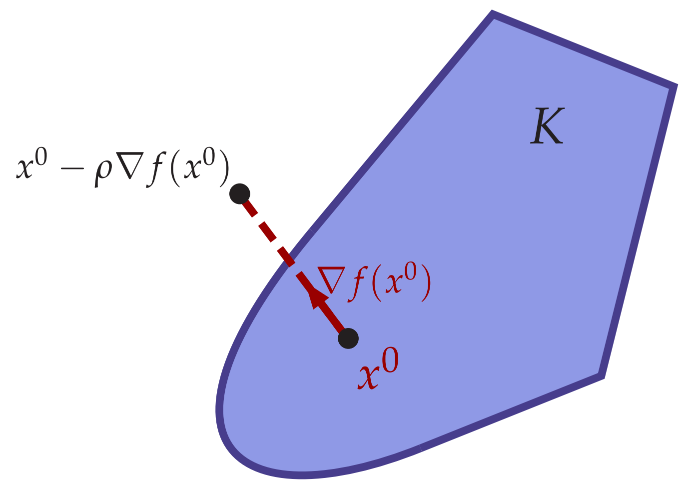
III. Prise en compte de contraintes – Gradient projeté

$$\min_{x\in K} f(x), \quad K \subset \mathbb{R}^d.$$

- On note p_K la projection sur l'ensemble K (convexe).

- Méthode du gradient projeté :

$$\left|\begin{array}{l} x^0 \text{ donné}, \\ x^{n+1} = p_K\big(x^n - \rho\nabla f(x^n)\big). \end{array}\right.$$



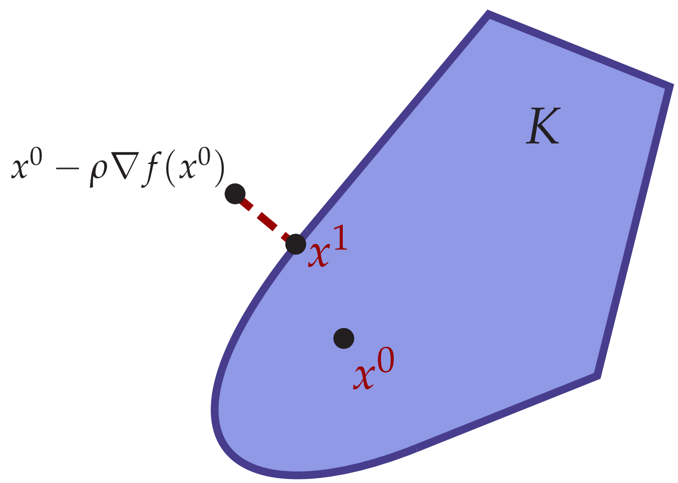
III. Prise en compte de contraintes – Gradient projeté

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

- On note p_K la projection sur l'ensemble K (convexe).

- Méthode du gradient projeté :

$$\left| \begin{array}{l} x^0 \text{ donné,} \\ x^{n+1} = p_K\big(x^n - \rho \nabla f(x^n)\big). \end{array} \right.$$



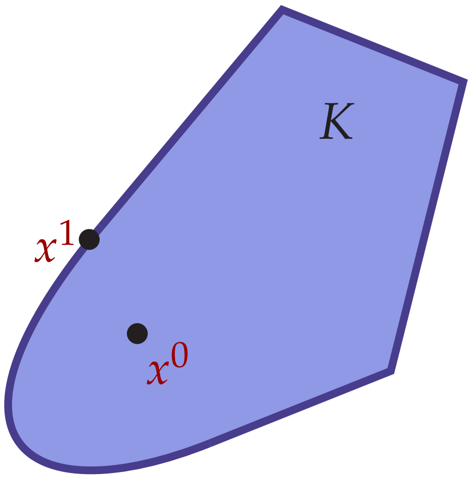
III. Prise en compte de contraintes – Gradient projeté

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

◇ On note p_K la projection sur l'ensemble K (convexe).

◇ Méthode du gradient projeté :

$$\left| \begin{array}{l} x^0 \text{ donné,} \\ \\ x^{n+1} = p_K\big(x^n - \rho \nabla f(x^n)\big). \end{array} \right.$$



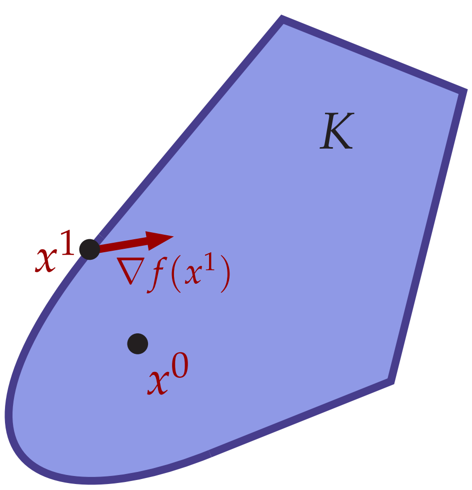
III. Prise en compte de contraintes – Gradient projeté

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

- On note p_K la projection sur l'ensemble K (convexe).

- Méthode du gradient projeté :

$$\left| \begin{array}{l} x^0 \text{ donné,} \\ x^{n+1} = p_K\big(x^n - \rho \nabla f(x^n)\big). \end{array} \right.$$



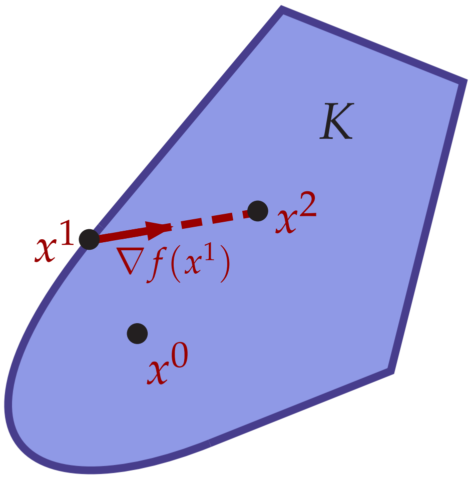
III. Prise en compte de contraintes – Gradient projeté

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

◇ On note p_K la projection sur l'ensemble K (convexe).

◇ Méthode du gradient projeté :

$$\left| \begin{array}{l} x^0 \text{ donné,} \\ x^{n+1} = p_K\big(x^n - \rho \nabla f(x^n)\big). \end{array} \right.$$



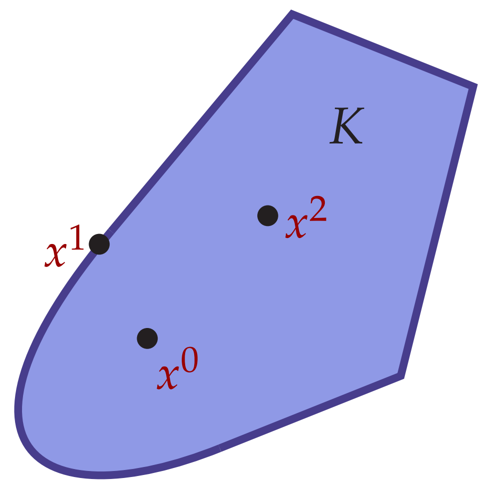
III. Prise en compte de contraintes – Gradient projeté

$$\min_{x \in K} f(x), \quad K \subset \mathbb{R}^d.$$

- On note p_K la projection sur l'ensemble K (convexe).

- Méthode du gradient projeté :

$$\left| \begin{array}{l} x^0 \text{ donné,} \\ x^{n+1} = p_K\big(x^n - \rho \nabla f(x^n)\big). \end{array} \right.$$



III. Prise en compte de contraintes – Gradient projeté

$$\left| \begin{array}{l} x^0 \text{ donné,} \\ \\ x^{n+1} = p_K\big(x^n - \rho \nabla f(x^n)\big). \end{array} \right.$$

◇ Même résultat de convergence que pour la méthode sans contrainte.

◇ Critère d'arrêt : $\|x^{n+1} - x^n\| \leq \varepsilon$.

◇ Difficulté : mise en œuvre de la projection p_K !

✔ *consommation des ménages.*

$$K = \Big\{ x \in \mathbb{R}^N \ ; \ x_i \geq 0 \text{ et } x_1 p_1 + \cdots + x_N p_N \leq b \Big\}.$$

✘ *Problème de Didon, chaînette, tambour.*

$$Ex : \quad K = \left\{ f \in \mathcal{C}^1([0,a]) \ ; \ \int_0^a \sqrt{1 + f'(x)^2}\, \mathrm{d}x = p \right\}.$$

Plan

III. Contraintes d'égalité – Méthode de Pénalisation

◇ **Idée :** $(P) \min_{x \in K \subset \mathbb{R}^d} f(x) \quad \longrightarrow \quad (P_\varepsilon) \min_{x \in \mathbb{R}^d} f_\varepsilon(x).$

◇ Choix de f_ε ? Si $K = \{x \in \mathbb{R}^d \, ; \, g(x) = 0\}$, on peut prendre

$$f_\varepsilon(x) = f(x) + \frac{1}{\varepsilon}[g(x)]^2.$$

Résultat. Sous certaines hypothèses sur f et g, la solution x_ε^* de (P_ε) converge vers x^* solution de (P) lorsque $\varepsilon \to 0$.
On dit que la pénalisation est exacte lorsque $x_\varepsilon^* = x^*$.

(P_ε) est sans contrainte !

on peut appliquer Newton, Gradient, etc.

Attention : choix numérique de ε crucial...

III. Contraintes d'égalité – Multiplicateurs de Lagrange

Théorème des extrema liés. Soit x^* solution de

$$\min_{x \in K} f(x) \quad \text{avec} \quad K = \{x \in \mathbb{R}^d \; ; \; h(x) = 0\}.$$

Alors il existe un *multiplicateur de Lagrange* $\lambda^* \in \mathbb{R}$ tel que

$$\nabla f(x^*) + \lambda^* \nabla h(x^*) = 0 \quad \textit{(équation d'Euler-Lagrange)}.$$

◇ *On peut appliquer la méthode de Newton au système*

$$\begin{cases} \nabla f(x) & + \quad \lambda \nabla h(x) & = & 0 \\ h(x) & & = & 0 \end{cases}$$

◇ *Lien avec la méthode de pénalisation :*

$$\nabla f_\varepsilon(x) = \nabla f(x) + \frac{1}{\varepsilon} \nabla [g(x)]^2.$$

III. Contraintes d'inégalité – Méthode de Pénalisation

◇ **Idée :** $(P)\ \min_{x\in K\subset\mathbb{R}^d} f(x) \quad \longrightarrow \quad (P_\varepsilon)\ \min_{x\in\mathbb{R}^d} f_\varepsilon(x).$

◇ Choix de f_ε ? Si $K = \{x \in \mathbb{R}^d\ ;\ h(x) \leq 0\}$, on peut prendre

$$f_\varepsilon(x) = f(x) + \frac{1}{\varepsilon}[\max(h(x), 0)]^2.$$

Résultat. Sous certaines hypothèses sur f et h, la solution x_ε^* de (P_ε) converge vers x^* solution de (P) lorsque $\varepsilon \to 0$.
On dit que la pénalisation est exacte lorsque $x_\varepsilon^* = x^*$.

(P_ε) est sans contrainte !

on peut appliquer Newton, Gradient, etc.

Attention : choix numérique de ε crucial...

III. Contraintes d'inégalité – Multiplicateurs de Kuhn-Tucker

Théorème de Kuhn-Tucker. Soit x^* solution de

$$\min_{x \in K} f(x) \quad \text{avec} \quad K = \{x \in \mathbb{R}^d \; ; \; h(x) \leq 0\}.$$

Alors il existe un *multiplicateur de Kuhn-Tucker* $\lambda^* \geq 0$ tel que

$$\nabla f(x^*) + \lambda^* \nabla h(x^*) = 0 \quad \textit{et} \quad \lambda^* h(x^*) = 0.$$

- Le multiplicateur est contraint : $\lambda^* \geq 0$;
- l'équation $\lambda^* h(x^*) = 0$ est appelée *relation d'exclusion*.
- comme pour les extrema liés, des conditions sont requises sur f et K (régularité, qualification).

Plan

III. Contraintes d'inégalité – Lagrangien

Définition. On appelle *Lagrangien* associé au problème

$$\min_{x \in K} f(x) \quad \text{avec} \quad K = \{x \in \mathbb{R}^d \; ; \; h(x) \leq 0\}.$$

la fonction

$$\mathcal{L}(x, \lambda) = f(x) + \lambda h(x).$$

◇ Kuhn-Tucker implique $\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$, soit [sous hypothèses sur f et h],

$$\mathcal{L}(x^*, \lambda^*) = \min_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda^*).$$

◇ Kuhn-Tucker implique encore

$$\mathcal{L}(x^*, \lambda^*) = \max_{\lambda \geq 0} \mathcal{L}(x^*, \lambda).$$

En effet, $\mathcal{L}(x^*, \lambda^*) = f(x^*) \geq f(x^*) + \lambda h(x^*) = \mathcal{L}(x^*, \lambda)$.

III. Contraintes d'inégalité – Lagrangien

Définition. On appelle *Lagrangien* associé au problème

$$\min_{x \in K} f(x) \quad \text{avec} \quad K = \{x \in \mathbb{R}^d \; ; \; h(x) \leq 0\}.$$

la fonction

$$\mathcal{L}(x, \lambda) = f(x) + \lambda h(x).$$

$$\mathcal{L}(x^*, \lambda^*) = \min_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda^*) \quad ; \quad \mathcal{L}(x^*, \lambda^*) = \max_{\lambda \geq 0} \mathcal{L}(x^*, \lambda).$$

Définition. On dit que (x^*, λ^*) est un *point selle* du Lagrangien.

III. Contraintes d'inégalité – Méthode d'Uzawa

Algorithme d'Uzawa

$$
\left|
\begin{array}{ll}
\lambda^0 \in \mathbb{R},\ \rho > 0 \text{ donnés,} & \\
x^n \in \mathbb{R}^d \quad \text{résout} \quad \displaystyle\min_{x \in \mathbb{R}^d} f(x) + \lambda^n h(x), & (2) \\
\lambda^{n+1} = \max\left[\lambda^n + \rho h(x^n); 0\right]. & (3)
\end{array}
\right.
$$

◇ *Le problème* (2) *est sans contrainte (Newton, gradient, etc) ;*

◇ *On résout rarement* (2) *entièrement (quelques itérations seulement) ;*

◇ *La relation* (3) *est un gradient projeté sur le problème*

$$\max_{\lambda \in \mathbb{R}^+} f(x^n) + \lambda h(x^n).$$

III. Contraintes d'égalité – Lagrangien augmenté

On considère le problème

$$\min_{x\in K} f(x) \quad \text{avec} \quad K = \{x \in \mathbb{R}^d \; ; \; h(x) = 0\}.$$

Le *Lagrangien augmenté* est défini comme

$$\mathcal{L}_r(x,\lambda) = f(x) + \lambda h(x) + \frac{r}{2}[h(x)]^2.$$

- ◇ r est un paramètre de pénalisation ;
- ◇ d'après le théorème des extrema liés,

$$\nabla_x \mathcal{L}_r(x^*,\lambda^*) = \nabla f(x^*) + \lambda^* \nabla h(x^*) = 0.$$

- ◇ Sous certaines hypothèses sur f et h, la pénalisation est exacte pour r suffisamment grand.

III. Contraintes d'égalité – Méthode des multiplicateurs

On considère le problème

$$\min_{x \in K} f(x) \quad \text{avec} \quad K = \{x \in \mathbb{R}^d \; ; \; h(x) = 0\}.$$

◇ Par définition, $\mathcal{L}_r(x,\lambda) = f(x) + \lambda h(x) + \frac{r}{2}[h(x)]^2$.

donc $\nabla_x \mathcal{L}_r(x,\lambda) = \nabla f(x) + \big[\lambda + rh(x)\big] \nabla h(x)$.

◇ Par extrema liés, $\nabla_x \mathcal{L}_r(x^*,\lambda^*) = \nabla f(x^*) + \lambda^* \nabla h(x^*) = 0$.

$\Longrightarrow$ **Idée** : approcher λ^* par $\lambda^{n+1} = \lambda^n + rh(x^n)$.

$$\left| \begin{array}{l} x^n \text{ résout } \displaystyle\min_{x \in \mathbb{R}^d} \mathcal{L}_r(x,\lambda^n), \\[2ex] \lambda^{n+1} = \lambda^n + rh(x^n). \end{array} \right.$$

Plan

IV. Optimisation globale – Introduction

On considère le problème

$$\min_{x \in K} f(x).$$

- Les méthodes présentées jusqu'ici sont locales :
  $\Longrightarrow$ problème des minima locaux.

- **Idée :** introduire une part d'aléatoire pour explorer mieux l'ensemble K.

- Faire mieux qu'une recherche aléatoire !

Conserver les avantages des méthodes de descente $\rightarrow$ *recuit simulé*,

Adopter une stratégie originale $\rightarrow$ *algorithmes évolutionnaires*.

IV. Optimisation globale – Recherche aléatoire

On considère le problème

$$\min_{x \in K} f(x).$$

Algorithme.

x^0 donné dans K,

y^n choisi au hasard dans K,

Si $f(y^n) < f(x^n)$, on pose $x^{n+1} = y^n$,

sinon, on conserve x^n.

- ◇ Avantage : simple à mettre en œuvre, bonne exploration de K,
- ◇ Inconvénient : lent et peu “intelligent”.

IV. Optimisation globale – Recuit simulé

On considère le problème

$$\min_{x \in K} f(x).$$

Algorithme.

$$\left|\begin{array}{l} x^0 \text{ donné dans } K, \\[1em] y^n \text{ choisi au hasard dans } K, \text{ voisin de } x^n, \\[1em] \text{Si } f(y^n) < f(x^n), \text{ on pose } x^{n+1} = y^n, \\[1em] \text{sinon, on conserve } x^n \text{ avec probabilité } p_n, \\[1em] \qquad\qquad \text{on pose } x^{n+1} = y^n \text{ avec probabilité } 1 - p_n. \end{array}\right.$$

- ◇ Stratégie de choix de voisin,
- ◇ Choix de p_n à préciser.

IV. Optimisation globale – Algorithmes évolutionnaires

Algorithmes "génétiques"

On considère le problème

$$\min_{x \in K} f(x).$$

◇ Algorithmes basés sur la théorie de l'évolution de Darwin.

✔ l'analogie biologique est source d'inspiration,

✔ elle aide à l'explication, la diffusion des méthodes,

✘ elle ne constitue pas une justification, ni une limitation.

IV. Optimisation globale – Algorithmes évolutionnaires

Principe de fonctionnement

On considère le problème

$$\min_{x \in K} f(x).$$

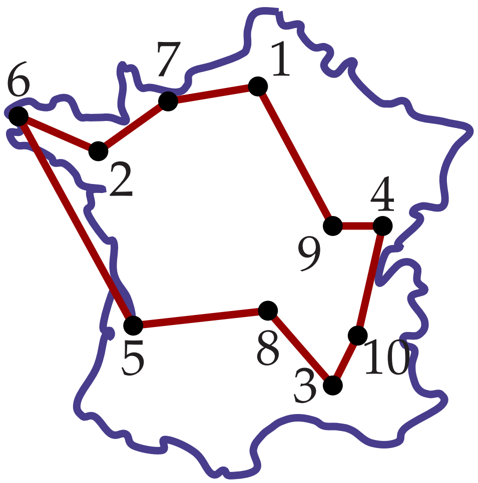
- ◇ On tire une population initiale au hasard :

$$N \text{ individus } x_1, \ldots, x_N \in K,$$

- ◇ On la fait évoluer au fil des générations à l'aide de
  - évaluation de la performance (ou *fitness*) des individus,
  - selection des “meilleurs” individus,
  - croisements entre individus,
  - mutations.

IV. Optimisation globale – Algorithmes évolutionnaires

L'exemple du voyageur de commerce



σ : permutation définissant le trajet,
($\sigma(k) = k^{\mathrm{e}}$ ville rencontrée.)

On recherche $\min\limits_{\sigma \in \mathfrak{S}_{10}} F(\sigma)$,

avec $F(\sigma) = \sum_{i=1}^{9} c[\sigma(i), \sigma(i+1)]$.

- ◇ **Initialisation.** On tire N permutations $\sigma_1, \ldots, \sigma_N$ au hasard.

- ◇ **Évaluation.** On calcule $F(\sigma_i)$ pour chaque i.

IV. Optimisation globale – Algorithmes évolutionnaires

L'exemple du voyageur de commerce (sélection)

◇ **Initialisation.** on tire N permutations $\sigma_1, \ldots, \sigma_N$ au hasard.

◇ **Évaluation.** on calcule le score $F(\sigma_i)$ pour chaque i.

◇ **Plusieurs stratégies de selection** *(avant "croisement/reproduction").*

- Élitisme

  on conserve les $N/2$ individus de meilleur score.

- Tournois

  on tire au hasard un couple d'individus
  et on conserve le gagnant.

- Roue de la fortune

  chaque individu a une probabilité d'être
  conservé proportionnelle à son score.

IV. Optimisation globale – Algorithmes évolutionnaires

L'exemple du voyageur de commerce (codage)

◇ Jusqu'ici, tout est indépendant du mode de représentation des individus.

Nécessité de définir un codage
pour préciser les croisements et mutations !

◇ **Représentation en chaîne de bits :**

- $x \sim 01110001$,
- très simple pour les croisements : un locus (ici 3) choisi au hasard,

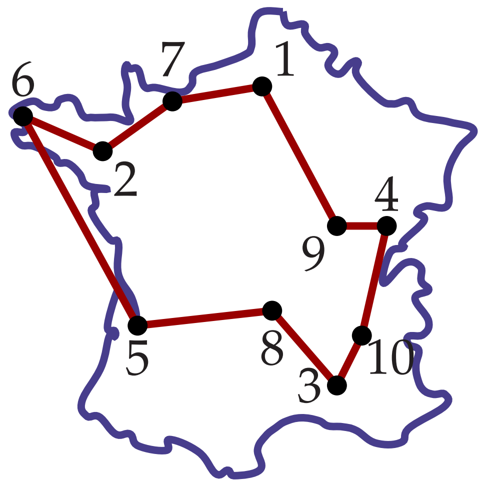
Père : 011|10001 Mère : 101|00111

Fils 1 : 011|00111 Fils 2 : 101|10001

- pas toujours naturel par rapport au problème.

IV. Optimisation globale – Algorithmes évolutionnaires

L'exemple du voyageur de commerce (croisements)



Naturellement, $\sigma \sim [6, 2, 7, 1, 9, 4, 10, 3, 8, 5]$.

Comment définir un croisement avec ce codage ?

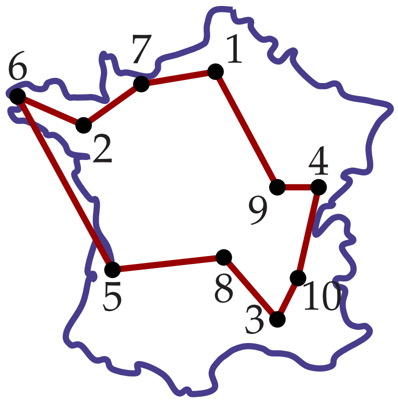
On choisit un locus (par exemple 4),

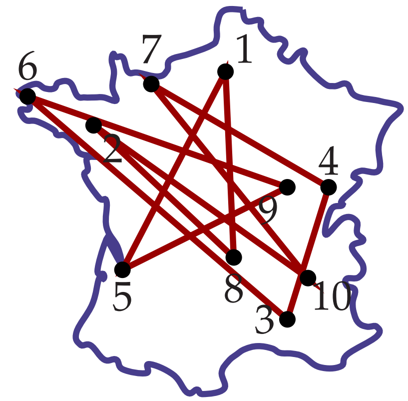
Père :	$[6, 2, 7, 1, 9, 4, 10, 3, 8, 5]$	Mère :	$[8, 2, 10, 7, 4, 3, 6, 9, 5, 1]$
Fils 1 :	$[6, 2, 7, 1, 8, 10, 4, 3, 9, 5]$	Fils 2 :	$[8, 2, 10, 7, 6, 1, 9, 4, 3, 5]$

(on a complété avec les numéros non encore écrits,
dans l'ordre rencontré chez l'autre parent.)

IV. Optimisation globale – Algorithmes évolutionnaires

L'exemple du voyageur de commerce (croisements)



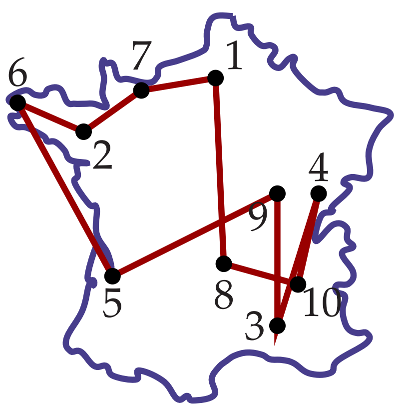


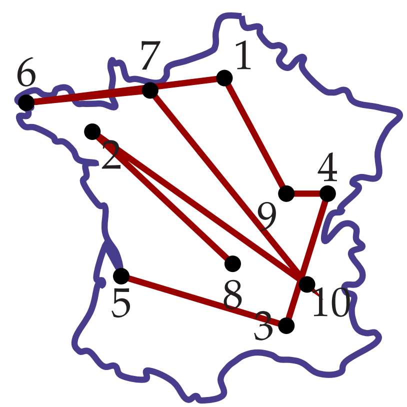
Père : $[6, 2, 7, 1, 9, 4, 10, 3, 8, 5]$

Mère : $[8, 2, 10, 7, 4, 3, 6, 9, 5, 1]$

Fils 1 : $[6, 2, 7, 1, 8, 10, 4, 3, 9, 5]$

Fils 2 : $[8, 2, 10, 7, 6, 1, 9, 4, 3, 5]$



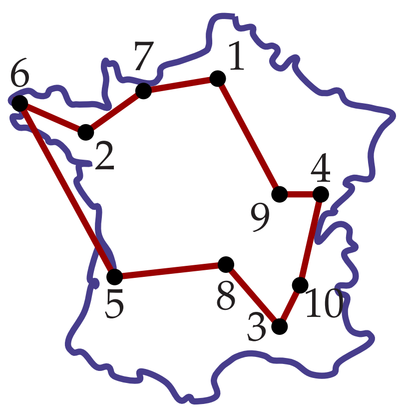


IV. Optimisation globale – Algorithmes évolutionnaires

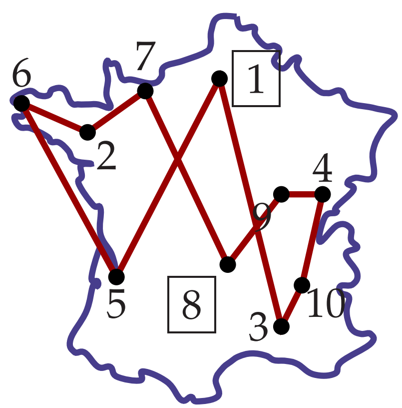
L'exemple du voyageur de commerce (mutations)

◇ Plusieurs stratégies de mutation :

- Remplacement d'un individu quelconque par un individu choisi au hasard (taille de population constante !),
- Inversion de deux villes au hasard dans un individu quelconque :



Avant mutation



Après mutation

IV. Optimisation globale – Algorithmes évolutionnaires

Paramètres de l'algorithme

- ◇ Taille de la population,
- ◇ Nombre de générations,
- ◇ Probabilité de croisement :
  - • on conserve les fils avec probabilité p_c,
  - • on conserve les parents avec probabilité $1 - p_c$,
- ◇ Probabilité de mutation :
  - • chaque individu a une probabilité p_m de subir une mutation.

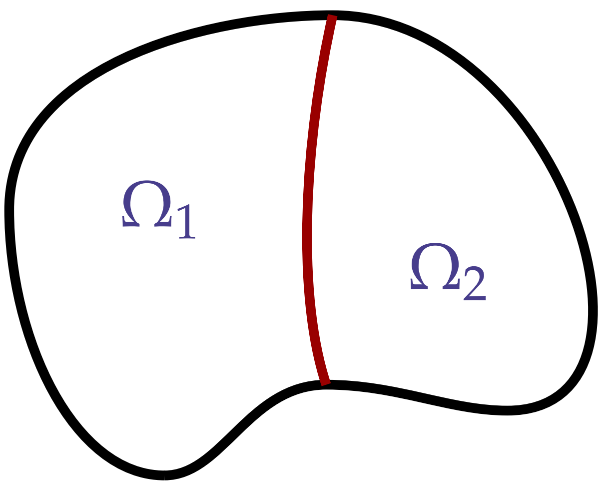
On doit ajuster les paramètres pour assurer convergence et diversité.

IV. Optimisation globale – Algorithmes évolutionnaires

Application à l'optimisation de forme *(cf Thèse Nicolas Landais)*

Étant donné un domaine Ω de $\mathbb{R}^2$, on cherche une partition (Ω_1, Ω_2) minimale au sens suivant.

$$\min \left[\lambda(\Omega_1) + \lambda(\Omega_2)\right],$$



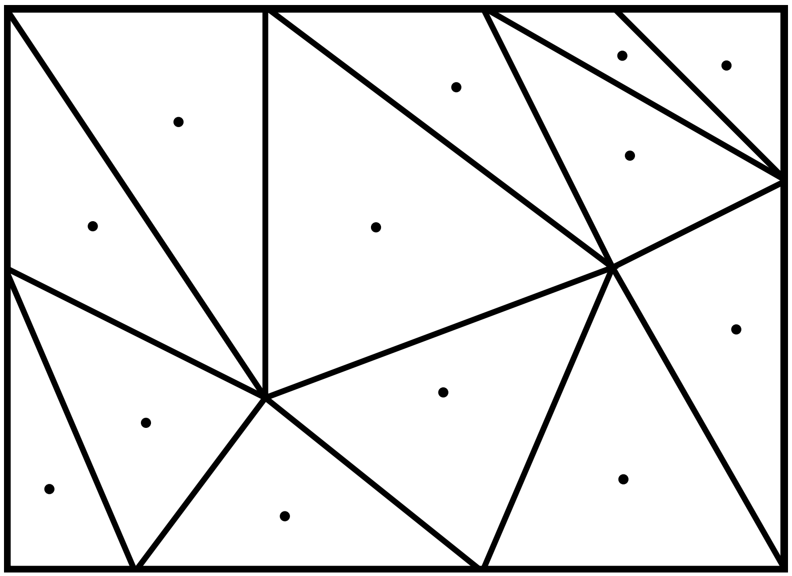
où $\lambda(\omega)$ désigne la plus petite valeur propre telle que

$$\left\{ \begin{array}{rcll} -\Delta u & = & \lambda u & \text{dans } \omega, \\ u & = & 0 & \text{sur } \partial\omega. \end{array} \right.$$

IV. Optimisation globale – Algorithmes évolutionnaires

Application à l'optimisation de forme *(cf Thèse Nicolas Landais)*

◇ Pour le calcul numérique de la partition optimale, on utilise un maillage.



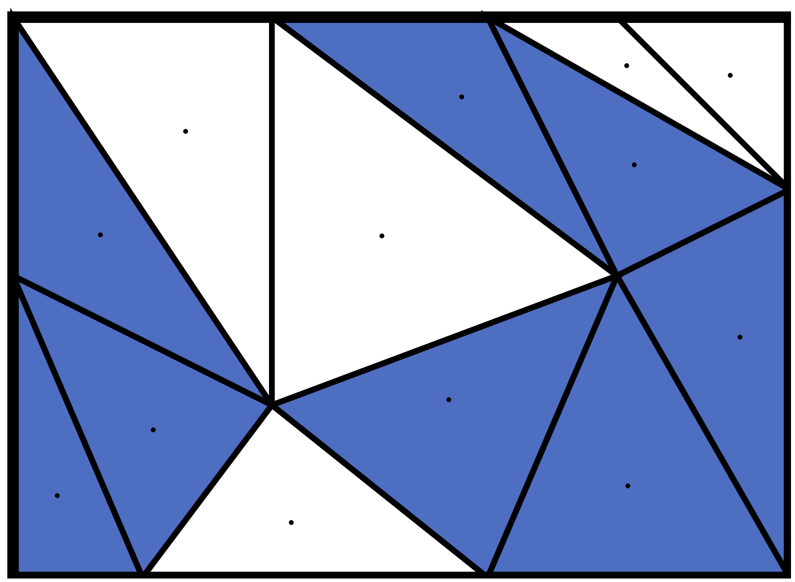
◇ Comment représenter la variable d'optimisation ?

IV. Optimisation globale – Algorithmes évolutionnaires

Application à l'optimisation de forme *(cf Thèse Nicolas Landais)*

◇ Une approche naïve : on attribue à chaque triangle un bit

- 0 correspond à Ω_1,
- 1 correspond à Ω_2.

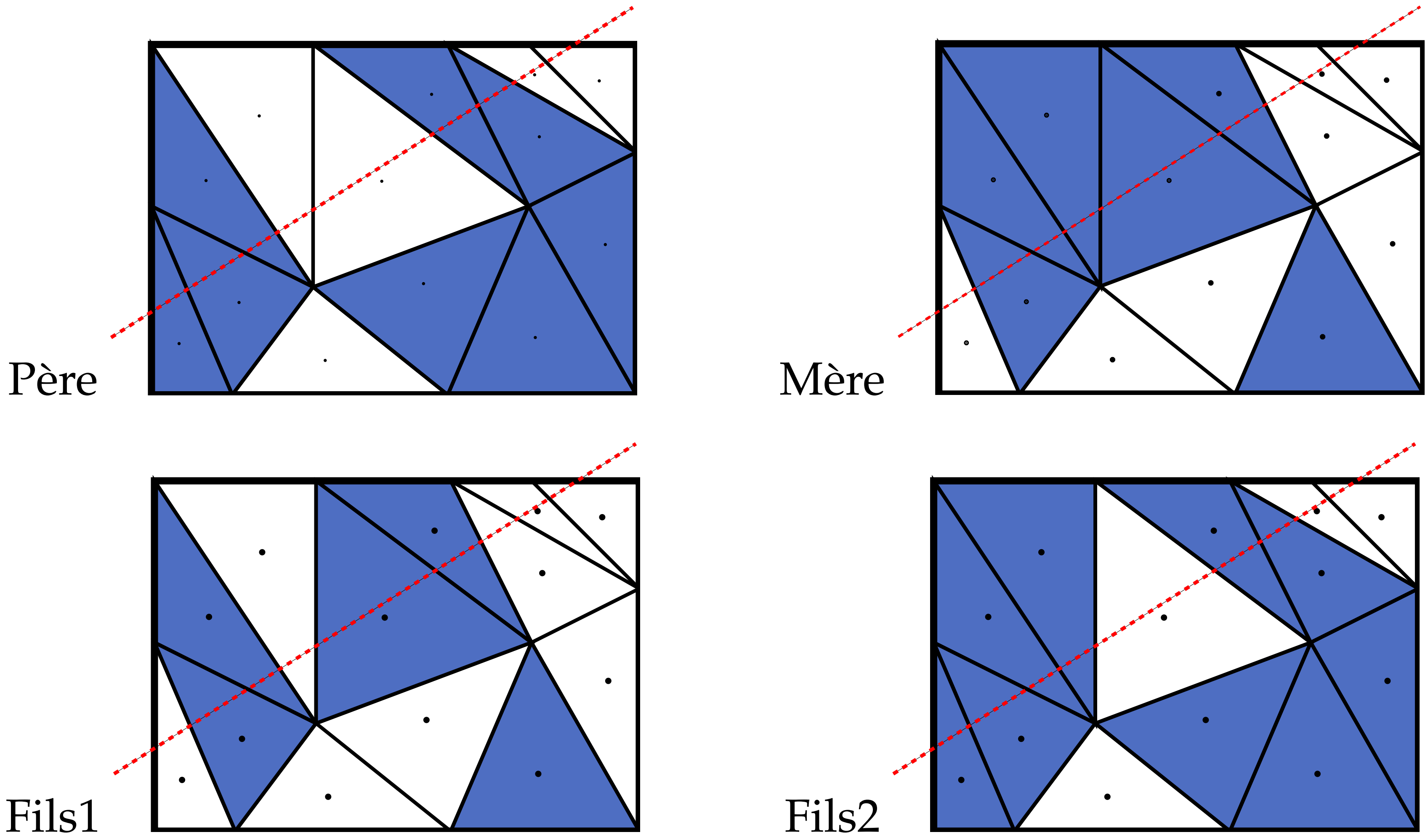


◇ Un individu (= une configuration) est représenté par une chaîne de bits.

IV. Optimisation globale – Algorithmes évolutionnaires

Application à l'optimisation de forme *(cf Thèse Nicolas Landais)*

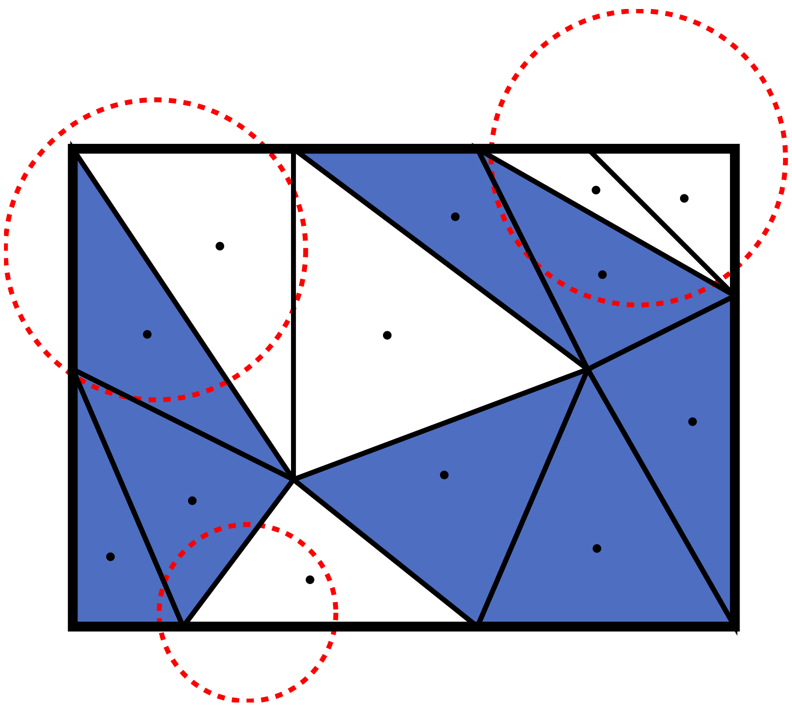
◇ Stratégie de croisement



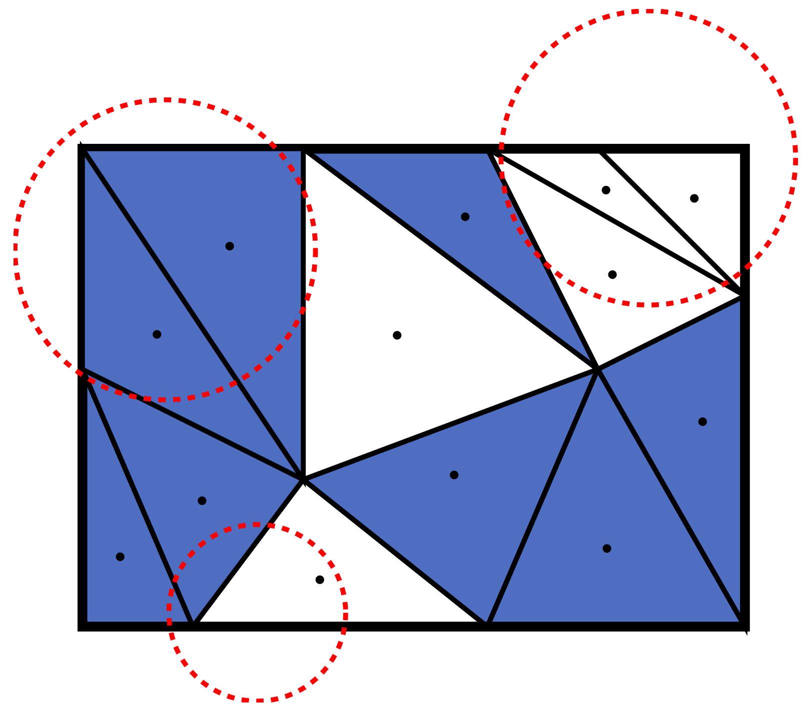
IV. Optimisation globale – Algorithmes évolutionnaires

Application à l'optimisation de forme *(cf Thèse Nicolas Landais)*

◇ Stratégie de mutation



Avant mutation

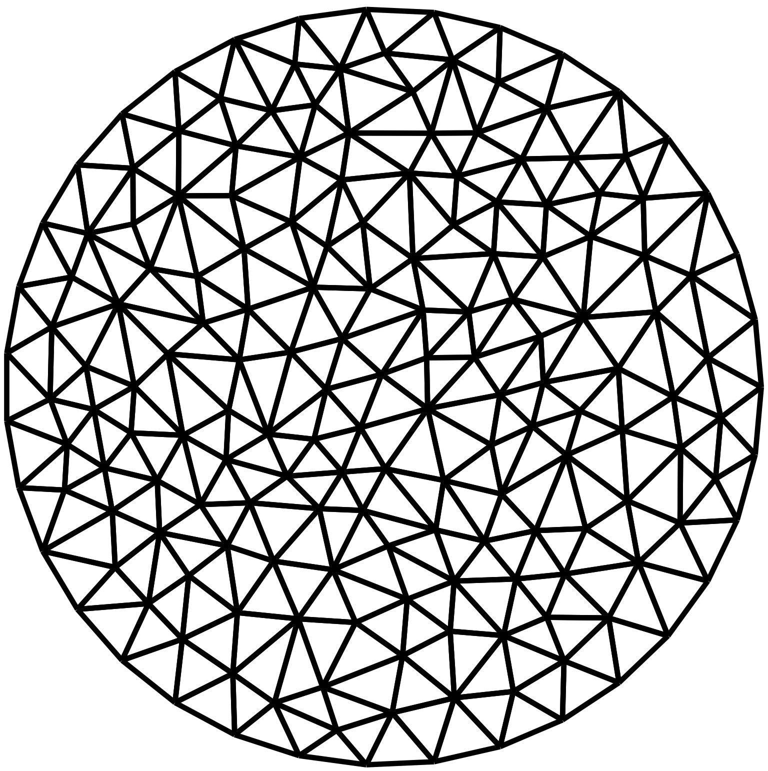


Après mutation

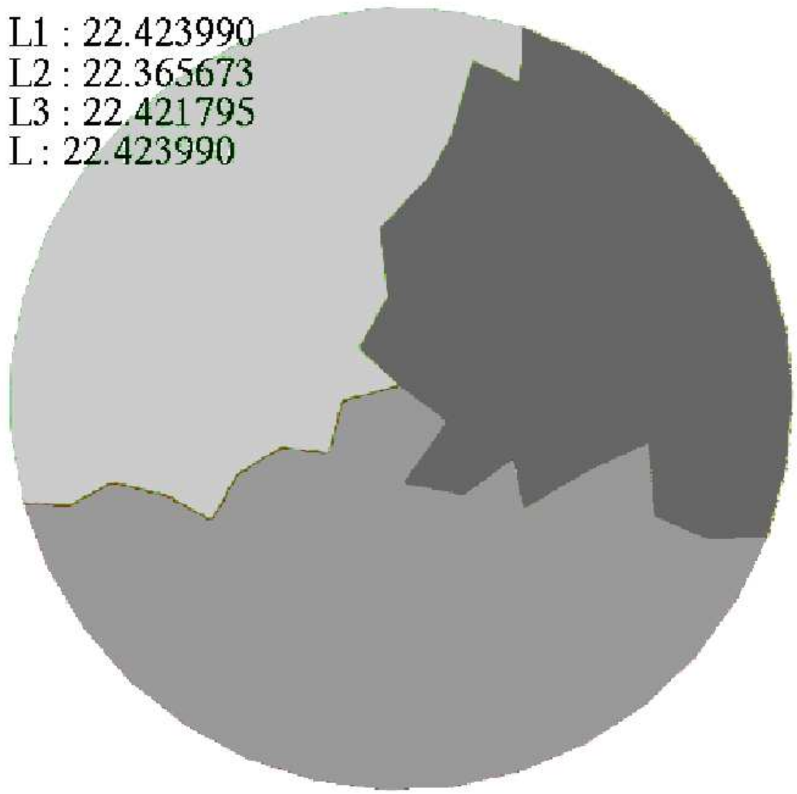
IV. Optimisation globale – Algorithmes évolutionnaires

Application à l'optimisation de forme *(cf Thèse Nicolas Landais)*

◇ Un exemple de résultat avec trois sous-domaines



Maillage



Partition obtenue

Plan

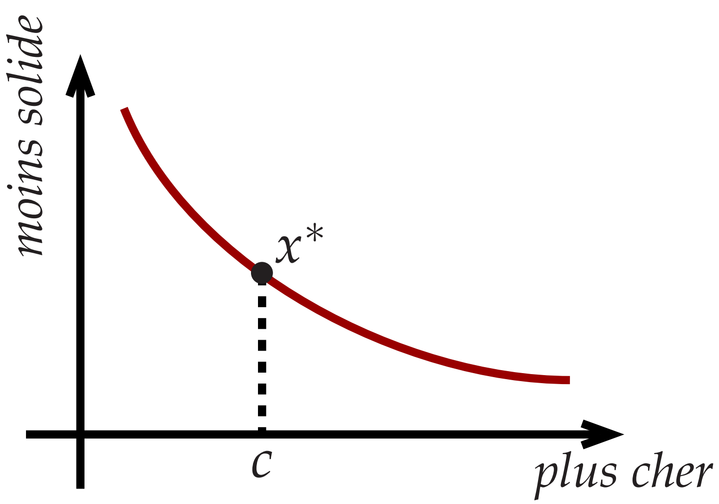
IV. Optimisation globale – optimisation multi-critère

Un exemple

◇ On souhaite construire un pont le plus résistant et le moins cher !

◇ Les deux objectifs/critères sont *concurrents*.

◇ À coût fixé, il existe un pont de solidité maximale :



◇ *Pour un coût égal à c, x^* maximise la solidité,*

◇ *la courbe rouge est appelée front de Paréto.*

◇ Aide à la décision, pas unicité de la solution (individus non dominés).

IV. Optimisation globale – optimisation multi-critère

Utilisation d'un algorithme évolutionnaire

◇ On peut se ramener à un problème mono-critère :

$$F(x) = \alpha \times \text{coût}(x) - \beta \times \text{solidité}(x).$$

En faisant varier α et β, on peut obtenir le front de Paréto.

$\Longrightarrow$ très long si on veut beaucoup de points sur le Front de Paréto !

◇ On peut obtenir le front de Paréto en une seule exécution.

- *Nondominated Sorting Genetic Algorithm* (NSGA),
- Principe : la *fitness* de x est fonction du nombre d'individus de la population qui dominent x.
- Attention à assurer la diversité de la population.