

Chapitre III

Optimisation sans contrainte

Dans cette leçon, nous verrons dans le cas où il n'y a pas de contrainte :

- Les conditions d'optimalité sans contrainte,
- Comment choisir la direction de déplacement à partir d'un point de départ,
- la résolution d'équation (= recherche de 0) avec la méthode de Newton,
- la minimisation d'équation avec la méthode de Newton,
- les méthodes de minimisation par recherche linéaire,
- la méthode directe de minimisation.

Rappel : la quasi-totalité des transparents de ce cours provient du cours d'optimisation de Max Cerf (Université Paris 6 / Ariane Espace).

1 Condition d'optimalité

Problème sans contraintes

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

Conditions nécessaires

$$x^* \text{ minimum local} \Rightarrow \begin{cases} \nabla f(x^*) = 0 \\ \nabla^2 f(x^*) \geq 0 \end{cases} \begin{array}{l} \rightarrow \text{ordre 1 : point critique ou stationnaire} \\ \rightarrow \text{ordre 2 : hessien semi-défini positif} \end{array}$$

Preuve : avec le théorème de Taylor

$$\bullet \text{ Ordre 1 : } f(x^* + d) = f(x^*) + \nabla f(x^*)^T d + o(\|d\|)$$

$$\text{Si } \nabla f(x^*) \neq 0, \text{ on peut trouver } d \text{ petit tel que } \nabla f(x^*)^T d < 0 \Rightarrow f(x^* + d) < f(x^*)$$

$$\begin{aligned} \bullet \text{ Ordre 2 : } f(x^* + d) &= f(x^*) + \nabla f(x^*)^T d + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2) \\ &= f(x^*) + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2) \quad \text{car } \nabla f(x^*) = 0 \end{aligned}$$

Si $\nabla^2 f(x^)$ non semi-définie positive,*

$$\text{on peut trouver } d \text{ petit tel que } d^T \nabla^2 f(x^*) d < 0 \Rightarrow f(x^* + d) < f(x^*)$$

Problème sans contraintes

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

Conditions suffisantes

$$\begin{cases} \nabla f(x^*) = 0 & \rightarrow \text{ordre 1 : point critique ou stationnaire} \\ \nabla^2 f(x^*) > 0 & \rightarrow \text{ordre 2 : hessien défini positif} \end{cases} \Rightarrow x^* \text{ minimum local}$$

Preuve : par l'absurde

Si x^ n'est pas un minimum local, on peut trouver d petit tel que $f(x^*+d) < f(x^*)$*

Théorème de Taylor à l'ordre 2 :

$$\begin{aligned} f(x^*+d) &= f(x^*) + \nabla f(x^*)^T d + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2) \\ &= f(x^*) + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2) \quad \text{car } \nabla f(x^*) = 0 \\ f(x^*+d) < f(x^*) &\Rightarrow d^T \nabla^2 f(x^*) d < 0 \quad \rightarrow \text{contredit l'hypothèse } \nabla^2 f(x^*) \text{ définie positive} \end{aligned}$$

Problème sans contraintes

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

- Condition nécessaire du 1^{er} ordre : $\nabla f(x^*) = 0 \rightarrow$ **point critique ou stationnaire**
- Condition nécessaire du 2^{ème} ordre : $\nabla^2 f(x^*) \geq 0 \rightarrow$ plus difficile à vérifier
suffisante $\nabla^2 f(x^*) > 0$

Méthode pratique

- Recherche des **points stationnaires** en résolvant : $\nabla f(x^*) = 0$
Un point stationnaire peut être un minimum local, un maximum local ou un point selle.
- Vérification de la condition d'ordre 2 : calcul des dérivées secondes
valeurs propres du hessien ≥ 0
 \rightarrow garantit l'obtention d'un minimum local x^*

Minimum global

x^* minimum local

- **f convexe** $\Rightarrow x^*$ minimum global
- f strictement convexe $\Rightarrow x^*$ unique minimum global
- **f quelconque** (cas général) \Rightarrow **On ne peut pas vérifier que x^* est un minimum global.**

Exemple 1**Fonction de Rosenbrock**

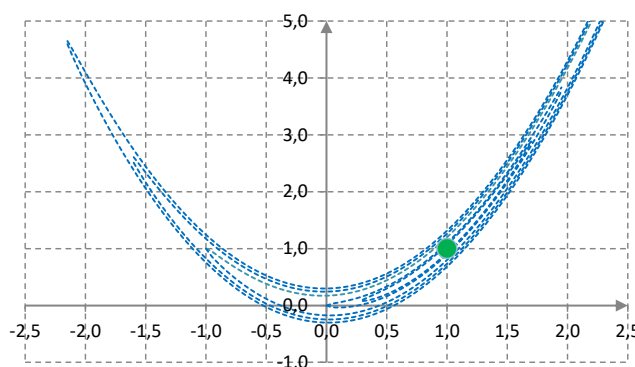
$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Gradient :

$$\nabla f(x_1, x_2) = \begin{pmatrix} 400x_1^3 - 400x_1x_2 + 2x_1 - 2 \\ 200x_2 - 200x_1^2 \end{pmatrix}$$

- Hessien :

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}$$



- **Point stationnaire** : $\nabla f(x_1, x_2) = 0 \Rightarrow \begin{cases} 400x_1^3 - 400x_1x_2 + 2x_1 - 2 = 0 \\ 200x_2 - 200x_1^2 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = 1 \\ x_2 = x_1^2 \end{cases} \Rightarrow x^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
- **Valeurs propres du hessien** : $\nabla^2 f(x^*) = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix} \rightarrow \begin{cases} \sigma_1 = 1001.60 \\ \sigma_2 = 0.39936 \end{cases}$
- Condition d'ordre 2 : $\nabla^2 f(x^*)$ est défini positif
- x^* vérifie les conditions suffisantes de minimum local (strict)
 x^* est un **minimum local** de f

Exemple 2

Fonction : $f(x_1, x_2) = -x_1^4 - x_2^4$

- Gradient : $\nabla f(x_1, x_2) = \begin{pmatrix} -4x_1^3 \\ -4x_2^3 \end{pmatrix}$

- Hessien : $\nabla^2 f(x_1, x_2) = \begin{pmatrix} -12x_1^2 & 0 \\ 0 & -12x_2^2 \end{pmatrix}$

- **Point stationnaire** : $\nabla f(x_1, x_2) = 0 \Rightarrow \begin{cases} x_1 = 0 \\ x_2 = 0 \end{cases} \Rightarrow x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

- **Valeurs propres du hessien** : $\nabla^2 f(x^*) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \rightarrow \begin{cases} \sigma_1 = 0 \\ \sigma_2 = 0 \end{cases}$
 $\nabla^2 f(x^*)$ est semi-défini positif

- x^* vérifie les conditions nécessaires de minimum local
 x^* ne vérifie pas les conditions suffisantes de minimum local

x^* est en fait un **maximum local** de f : $f(x_1, x_2) = -x_1^4 - x_2^4 \leq 0$
 $\Rightarrow \forall (x_1, x_2), f(x_1, x_2) \leq f(0, 0)$

Exemple 3

Fonction : $f(x_1, x_2) = x_1^2 - x_2^3$

• Gradient : $\nabla f(x_1, x_2) = \begin{pmatrix} 2x_1 \\ -3x_2^2 \end{pmatrix}$

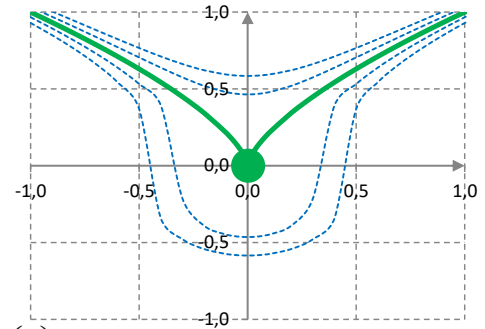
• Hessien : $\nabla^2 f(x_1, x_2) = \begin{pmatrix} 2 & 0 \\ 0 & -6x_2 \end{pmatrix}$

• **Point stationnaire** : $\nabla f(x_1, x_2) = 0 \Rightarrow \begin{cases} x_1 = 0 \\ x_2 = 0 \end{cases} \Rightarrow x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

• **Valeurs propres du hessien** : $\nabla^2 f(x^*) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \rightarrow \begin{cases} \sigma_1 = 2 \\ \sigma_2 = 0 \end{cases}$
 $\nabla^2 f(x^*)$ est semi-défini positif

- x^* vérifie les conditions nécessaires de minimum local
- x^* ne vérifie pas les conditions suffisantes de minimum local

x^* n'est **ni un minimum ni un maximum local** de f : $f(0, x_2) = -x_2^3 \begin{cases} < 0 \text{ si } x_2 > 0 \\ > 0 \text{ si } x_2 < 0 \end{cases}$



2 Principes de minimisation

Problème non linéaire sans contraintes

$$\min_{x \in \mathbb{R}^n} f(x) \rightarrow \text{problème noté (PO)}$$

Méthodes globales

- Capacité à localiser plusieurs minima locaux (éventuellement le minimum global)
- Algorithmes non déterministes (déplacements aléatoires « organisés »)
- Métaheuristiques : algorithmes génétiques, recuit simulé, essaims, colonies de fourmis, recherche tabou,...
- Convergence généralement lente, peu précise

Méthodes locales

- Recherche d'un **minimum local** à partir d'un **point initial** fourni par l'utilisateur
- Méthodes d'ordre 0 : sans dérivées \rightarrow Nelder-Mead, Direct
- d'ordre 1 : avec dérivées premières \rightarrow plus forte pente
- d'ordre 2 : avec dérivées premières et secondes \rightarrow Newton
- Critères d'efficacité : **rapidité** de convergence (nombre d'appels de la fonction)
précision de convergence
robustesse à l'initialisation

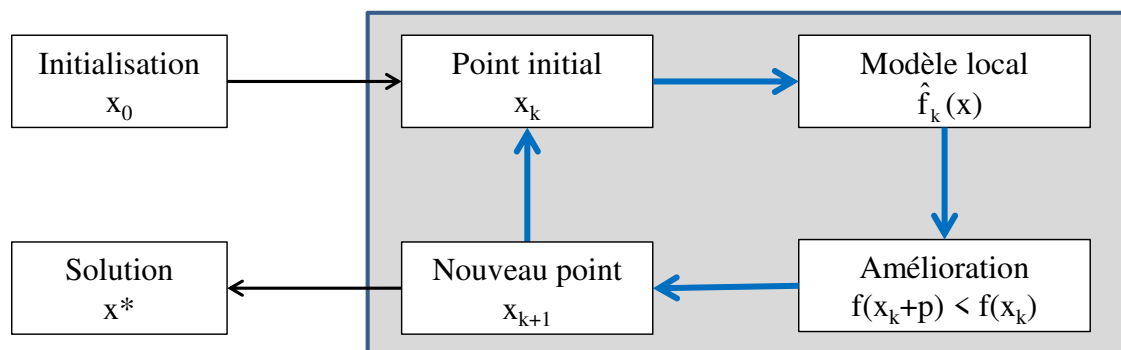
Problème sans contrainte

$$\min_{x \in \mathbb{R}^n} f(x) \quad x^* \text{ minimum local} \Rightarrow \begin{cases} \nabla f(x^*) = 0 \\ \nabla^2 f(x^*) \geq 0 \end{cases}$$

On ne sait pas trouver le minimum global dans le cas général (f quelconque).

Méthode locale

- Initialisation x_0 → recherche d'un minimum local au voisinage de x_0
- Itérations → passage du point x_k au point x_{k+1} meilleur
- Arrêt → solution x^* ou blocage

**Modèle local : prédiction**

- Point courant $x_k, f_k = f(x_k)$
 - Evaluation de $g_k = \nabla f(x_k)$ ou approximation (différences finies)
 $H_k = \nabla^2 f(x_k)$ ou approximation (quasi Newton)
 - Modèle quadratique : $\min_p \hat{f}_k(x_k + p) = f_k + p^t g_k + \frac{1}{2} p^t H_k p \rightarrow \hat{x}_{k+1} = x_k + \hat{p}$ (prédiction)
- **Méthodes de Newton ou quasi-Newton**

Amélioration : correction

- Nouveau point $x_{k+1} = x_k + p$ tel que $f(x_k + p) < f(x_k)$
- Déplacement p à partir de x_k
 par recherche linéaire suivant $d_k = \hat{x}_{k+1} - x_k$
 par région de confiance dans $\|x - x_k\| < r$

→ **Méthodes de globalisation**

- La méthode de Newton appliquée directement ne converge pas systématiquement. La globalisation est nécessaire pour contrôler la convergence.

Les techniques de globalisation ne sont pas abordées dans ce cours. Voir le cours de Max Cerf pour une introduction à ces techniques.

Initialisation

- Les **méthodes locales** recherchent le minimum au voisinage du point de départ.
- Objectifs :
 - rapidité de convergence
 - précision de convergence
 - **méthodes à base de dérivées**
- Le minimum local trouvé est le plus proche du point initial x_0 .
 - initialisation à modifier pour trouver un autre minimum local
- Les **méthodes « globales »** explorent « aléatoirement » les solutions
 - localisation possible de plusieurs minima locaux

Conditions d'arrêt

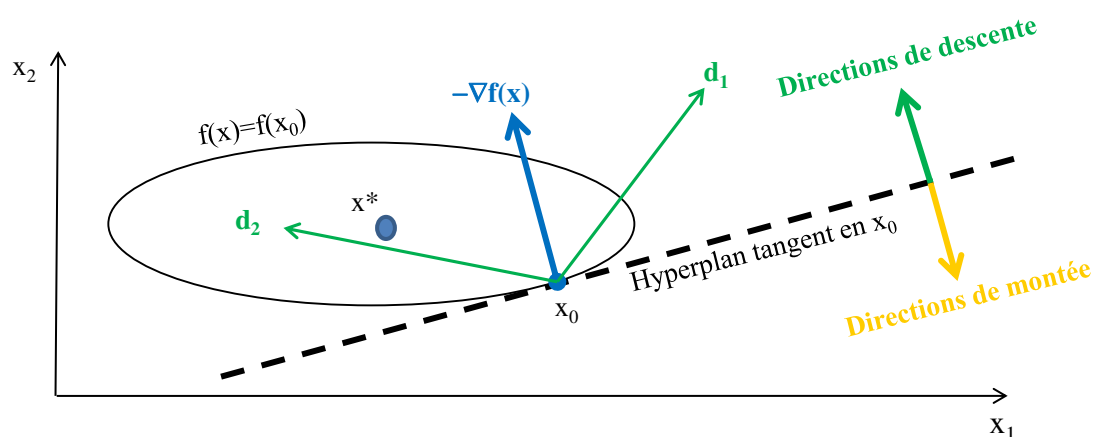
- Déplacement insuffisant : $\|x_{k+1} - x_k\| < \varepsilon_x$
- Amélioration insuffisante : $f_k - f_{k+1} < \varepsilon_f$
- Condition d'ordre 1 vérifiée : $\|g_k\| < \varepsilon_g$
- Nombre maximal d'itérations ou d'appels fonction : $N_{\text{iter}}, N_{\text{fonc}}$

3 Direction de déplacement

Comment choisir la direction de déplacement pour une méthode locale ?

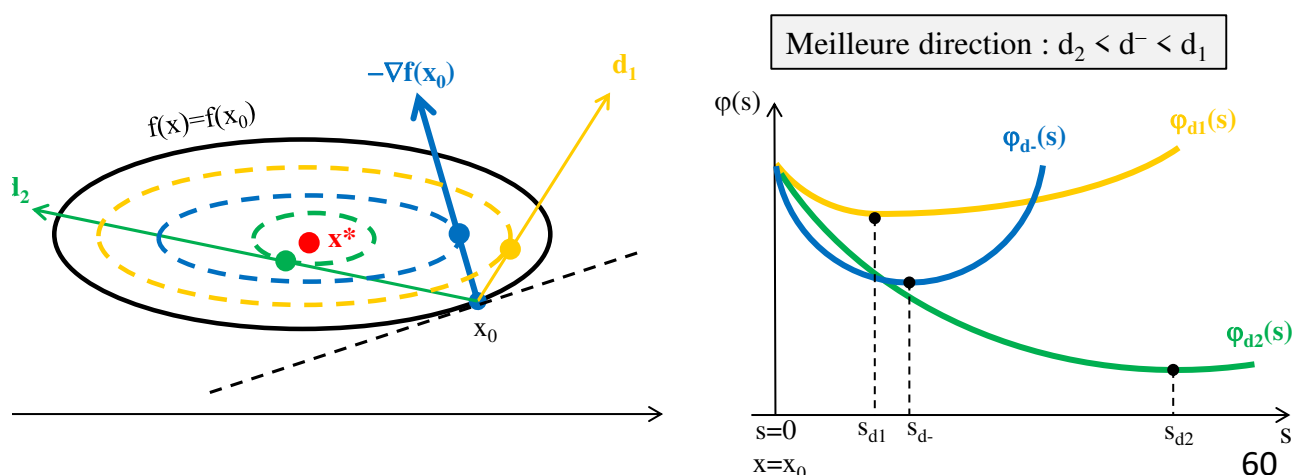
Direction de descente

- Gradient de f en $x \in \mathbb{R}^n$: $g(x) = \nabla f(x)$
- Dérivée directionnelle de f en x suivant $d \in \mathbb{R}^n$: $f_d(x) = g(x)^T d$
 d est une **direction de descente** en x si : $f_d(x) = g(x)^T d < 0$
- La direction de **plus forte pente** d^+ est la direction du gradient : $d^+ = g(x)$
- La direction de **plus forte descente** d^- est opposée au gradient : $d^- = -g(x)$
 $\forall d \in \mathbb{R}^n / \|d\| = \|d^-\|, g(x)^T d \geq g(x)^T d^- = -\|g(x)\|^2$



Variation suivant une direction

- Déplacement $d \in \mathbb{R}^n$ à partir de $x_0 \in \mathbb{R}^n$: $d = sd_0$ avec $d_0 \in \mathbb{R}^n =$ direction de déplacement
 $s \in \mathbb{R} =$ pas de déplacement suivant d_0
- Modèle quadratique au voisinage de x_0 :
 $\hat{f}(x_0 + sd_0) = f(x_0) + sg_0^T d_0 + \frac{1}{2}s^2 d_0^T H_0 d_0 \Rightarrow \hat{\varphi}(s) = \varphi(0) + s\varphi'(0) + \frac{1}{2}s^2 \varphi''(0)$
avec $\begin{cases} g_0 = \nabla f(x_0) \\ H_0 = \nabla^2 f(x_0) \end{cases}$ avec $\begin{cases} \varphi'(0) = g_0^T d_0 \\ \varphi''(0) = d_0^T H_0 d_0 \end{cases}$

**Minimisation locale**

Deux points particuliers sont définis à partir du modèle quadratique de f en x_0 :

- Point de Newton : minimisation de f par rapport à $d \in \mathbb{R}^n \rightarrow x_n = x_0 + d_n$
- Point de Cauchy : minimisation de f suivant $d_0 = -g_0 \rightarrow x_c = x_0 - s_c g_0$

Point de Newton

$$\min_{d \in \mathbb{R}^n} \hat{f}(x_0 + d) = f(x_0) + g_0^T d + \frac{1}{2} d^T H_0 d$$

$$\Rightarrow d_n = -H_0^{-1} g_0 \text{ si } H_0 > 0$$

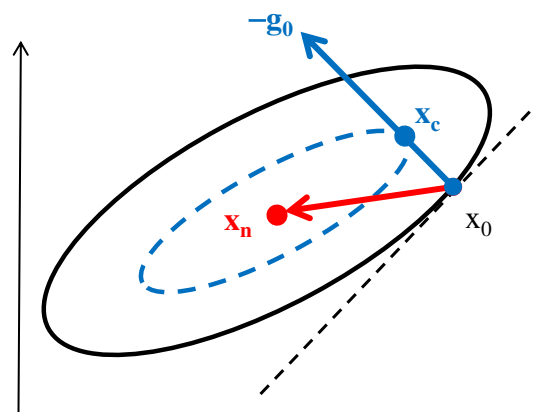
x_n existe si $H_0 = \nabla^2 f(x_0)$ est définie positive.

Point de Cauchy

$$\min_{s \in \mathbb{R}} \hat{\varphi}(x_0 - sg_0) = \varphi(0) - s\varphi'(0) + \frac{1}{2}s^2 \varphi''(0)$$

$$\Rightarrow s_c = \frac{\varphi'(0)^2}{\varphi''(0)} = \frac{(g_0^T g_0)^2}{g_0^T H_0 g_0} \text{ si } \varphi''(0) = g_0^T H_0 g_0 > 0$$

x_c existe si f est convexe suivant g_0 (condition moins forte que H_0 définie positive).



- point de Newton** : résolution de $\nabla^2 f(x_k) d_n = -\nabla f(x_k)$, à savoir du déplacement faisant coïncider le gradient par son estimation à l'ordre 1.
- point de Cauchy** : pas optimal s_c dans la direction de plus forte descente (minimisation du modèle quadratique $\varphi(x_0 - s_k \nabla f(x_0))$).
Noter qu'il est inversement proportionnel à la courbure de la fonction au point.

4 Résolution d'équation

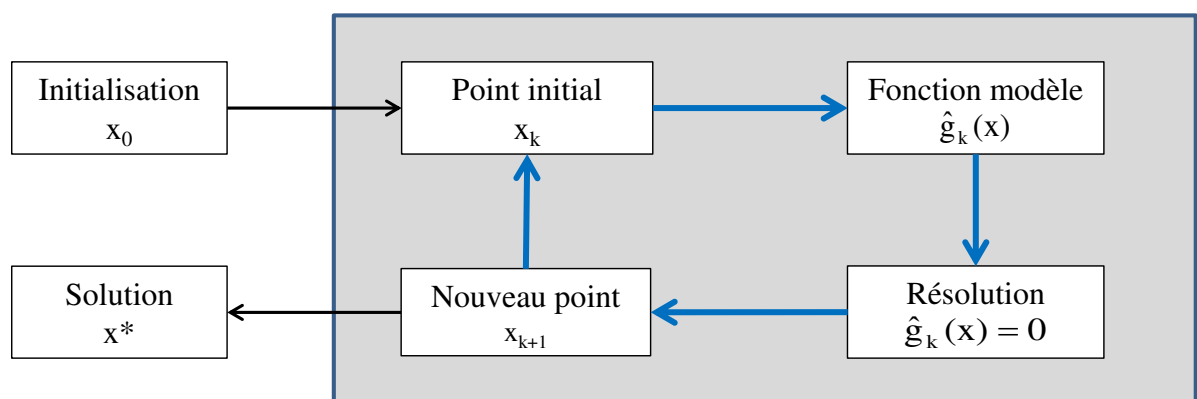
On s'intéresse dans un premier temps à la recherche de 0 pour une fonctionnelle g .

Système d'équations non linéaires

$$\boxed{g(x) = 0 \quad \text{avec } g : \mathbb{R}^n \rightarrow \mathbb{R}^n} \quad \rightarrow \text{ système de } n \text{ équations à } n \text{ inconnues}$$

Principe de la méthode de Newton

- Linéariser g au point initial x_0 \rightarrow fonction modèle linéaire \hat{g} « proche » de g
- Résoudre le système linéaire $\hat{g}(x)=0$ \rightarrow nouveau point x_1
- Itérer jusqu'à vérifier $g(x)=0$ \rightarrow solution x^*



4.1 Méthode de Newton

Voyons maintenant la méthode de Newton.

Fonction modèle

- Développement de Taylor à l'ordre 1 de g en x_k

$$g(x) = g(x_k) + \nabla g(x_k)^T (x - x_k) + o(\|x - x_k\|)$$

- Modèle linéaire de g en x_k :

$$\boxed{\hat{g}_k(x) = g(x_k) + G_k (x - x_k)} \quad \text{avec } G_k \in \mathbb{R}^{n \times n}$$

Choix de la matrice G_k

- Méthode de **Newton** $\rightarrow G_k = \nabla g(x_k)^T =$ matrice jacobienne de g en x_k
- Méthode de **quasi-Newton** $\rightarrow G_k =$ approximation de $\nabla g(x_k)^T$

Résolution

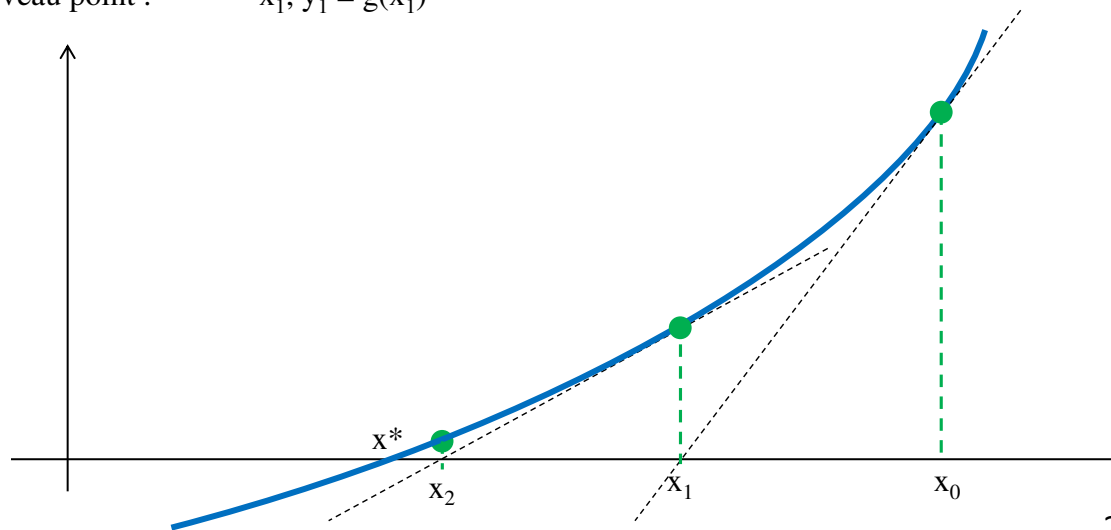
- Système linéaire : $\hat{g}_k(x) = 0 \Rightarrow g(x_k) + G_k(x - x_k) = 0$

- Itération : $\boxed{x_{k+1} = x_k - G_k^{-1} g(x_k)}$ si G_k inversible

- Condition d'arrêt : $\|g(x_k)\| < \varepsilon$

Illustration à une variable

- Equation non linéaire : $g(x) = 0$
- Point initial : $x_0, y_0 = g(x_0) \rightarrow$ **Tangente** en x_0 : $y = y_0 + g'(x_0)(x - x_0)$
- Intersection axe x : $y = 0 \Rightarrow x_1 = x_0 - \frac{y_0}{g'(x_0)}$
- Nouveau point : $x_1, y_1 = g(x_1)$



Géométriquement, le point choisit pour la prochaine itération est la position de la sécante sur l'axe des x .

Modèle linéaire de g en x_k

$$\hat{g}_k(x) = g(x_k) + G_k(x - x_k) \quad \text{avec } G_k = \nabla g(x_k)^T$$

Erreur de linéarisation

M = constante de Lipschitz sur le gradient \approx majorant de la courbure

$$\|g(x) - \hat{g}_k(x)\| \leq \frac{1}{2} M \|x - x_k\|^2 \rightarrow \text{erreur quadratique}$$

Vitesse de convergence

Hypothèses sur la solution x^* : $\nabla g(x^*)$ inversible

$$\|\nabla g(x^*)^{-1}\| \leq \eta$$

Suite (x_k) : $x_{k+1} = x_k - G_k^{-1} g(x_k)$

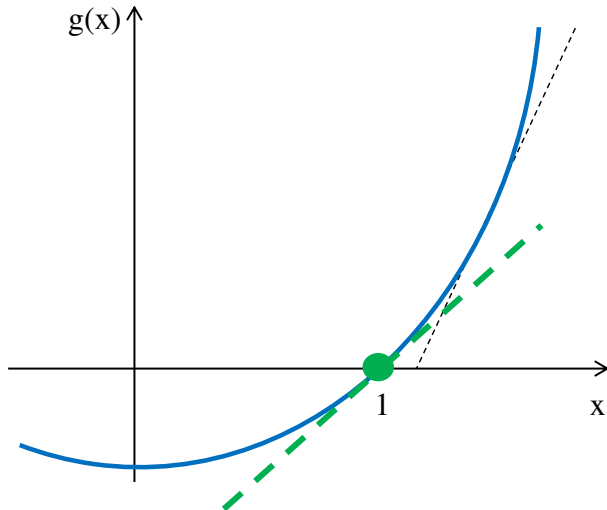
- (x_k) converge vers x^* si x_0 est « assez proche » de x^* : $\exists r > 0 / \|x_0 - x^*\| < r \Rightarrow \lim_{k \rightarrow \infty} x_k = x^*$

• La convergence est **quadratique**

$$\rightarrow \|x_{k+1} - x^*\| \leq M\eta \|x_k - x^*\|^2$$

Exemple 1

- Fonction : $g(x) = x^2 - 1$
- Dérivée : $g'(x) = 2x$
- Solution : $x = 1 \rightarrow g'(1) = 2$

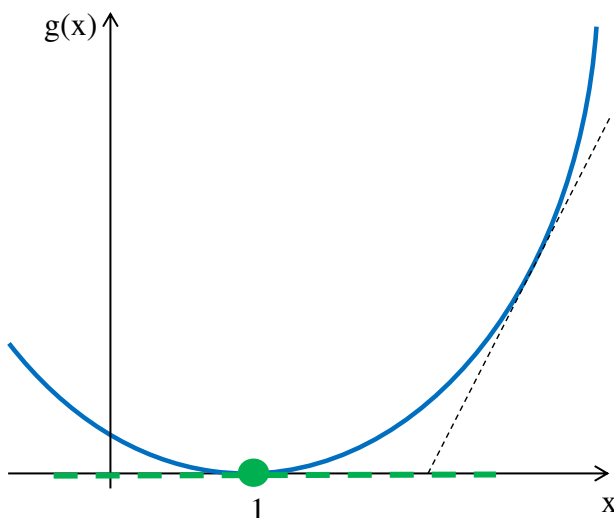


Iteration	x(k)	g(x)=x**2-1	g'(x)=2x	Erreur
0	4,00000000	1,5E+01	8,0000	3,0E+00
1	2,12500000	3,5E+00	4,2500	1,1E+00
2	1,29779412	6,8E-01	2,5956	3,0E-01
3	1,03416618	6,9E-02	2,0683	3,4E-02
4	1,00056438	1,1E-03	2,0011	5,6E-04
5	1,00000016	3,2E-07	2,0000	1,6E-07
6	1,00000000	2,5E-14	2,0000	1,3E-14

Convergence quadratique
 $g'(x^*)$ inversible

Exemple 2

- Fonction : $g(x) = (x - 1)^2$
- Dérivée : $g'(x) = 2(x - 1)$
- Solution : $x = 1 \rightarrow g'(1) = 0$

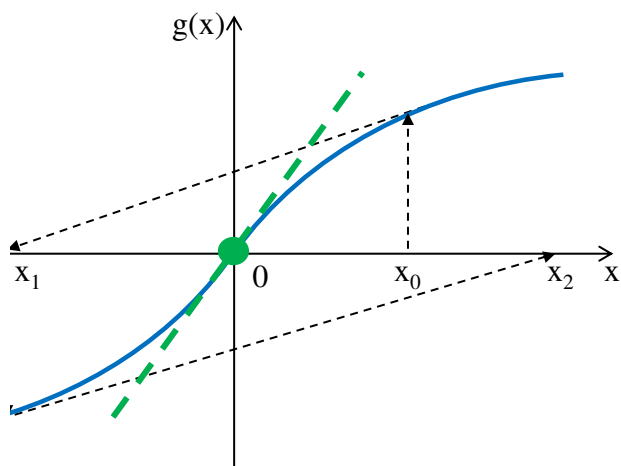


Iteration	x(k)	g(x)=(x-1)**2	g'(x)=2(x-1)	Erreur
0	4,00000000	9,0E+00	6,0000	3,0E+00
1	2,50000000	2,3E+00	3,0000	1,5E+00
2	1,75000000	5,6E-01	1,5000	7,5E-01
3	1,37500000	1,4E-01	0,7500	3,8E-01
4	1,18750000	3,5E-02	0,3750	1,9E-01
5	1,09375000	8,8E-03	0,1875	9,4E-02
6	1,04687500	2,2E-03	0,0938	4,7E-02
7	1,02343750	5,5E-04	0,0469	2,3E-02
8	1,01171875	1,4E-04	0,0234	1,2E-02
9	1,00585938	3,4E-05	0,0117	5,9E-03
10	1,00292969	8,6E-06	0,0059	2,9E-03
15	1,00009155	8,4E-09	0,0002	9,2E-05
20	1,00000286	8,2E-12	0,0000	2,9E-06

Convergence lente
 $g'(x^*)$ non inversible

Exemple 3

- Fonction : $g(x) = \text{Arc tan}(x)$
- Dérivée : $g'(x) = \frac{1}{1+x^2}$
- Solution : $x = 0 \rightarrow g''(1) = 0$



Iteration	x(k)	g(x)=Arctan(x)	g'(x)=1/(1+x**2)	Erreur
0	1,300	0,915	0,372	1,3E+00
1	-1,162	-0,860	0,426	-1,2E+00
2	0,859	0,710	0,575	8,6E-01
3	-0,374	-0,358	0,877	-3,7E-01
4	0,034	0,034	0,999	3,4E-02
5	0,000	0,000	1,000	-2,6E-05
6	0,000	0,000	1,000	1,2E-14

Convergence

Iteration	x(k)	g(x)=Arctan(x)	g'(x)=1/(1+x**2)	Erreur
0	1,500	0,983	0,308	1,5E+00
1	-1,694	-1,038	0,258	-1,7E+00
2	2,321	1,164	0,157	2,3E+00
3	-5,114	-1,378	0,037	-5,1E+00
4	32,296	1,540	0,001	3,2E+01
5	-1575,317	-1,570	0,000	-1,6E+03
6	384976,008	1,571	0,000	3,9E+06

Divergence

246

EXERCICE 15: Recherche d'un zéro par la méthode de Newton (1)

Soit la fonction $f(x) = e^{x-3} + x - 2$. On veut trouver la valeur de x_0 telle que $f(x_0) = 0$.

- calculer la dérivée de $f(x)$.
- en partant de $x = 2$, utiliser le méthode de Newton pour trouver le 0.

EXERCICE 16: Recherche d'un zéro par la méthode de Newton (2)

Soit la fonction $f(x) = \cos x$. On veut trouver la valeur de x_0 telle que $f(x_0) = 0$.

- calculer la dérivée de $f(x)$.
- en partant de $x = 1$, utiliser le méthode de Newton pour trouver le 0.
- que peut-on dire du zéro trouvé?

Intérêt de la méthode de Newton

- Convergence quadratique au voisinage de la solution → **très rapide et précise**
- Méthode à privilégier dans les algorithmes d'optimisation

Difficultés

- Calcul explicite du gradient $\nabla g(x_k)$ à chaque itération → coûteux (n appels fonctions)
- Convergence non garantie → même près de la solution

Adaptations

- Méthodes de **quasi-Newton** → $G_k =$ **approximation** du gradient $\nabla g(x_k)$
construite à partir des itérations précédentes
sans calcul explicite du gradient
- Techniques de **globalisation** → **Contrôle** du point x_{k+1} (meilleur que x_k ?)
 $\|g(x_{k+1})\| < \|g(x_k)\|$

Si le point x_{k+1} n'est pas satisfaisant → Méthodes de recherche linéaire
ou région de confiance
pour minimiser $\|g(x)\|^2$

Les techniques de globalisation ne sont pas abordées dans ce cours. Voir le cours de Max Cerf pour une introduction à ces techniques.

4.2 Méthode de quasi-Newton

On rappelle que $G_k = \nabla g(x_k)^T$, où g est la fonction dont on recherche le zéro.

Méthode de Broyden

On cherche à définir la matrice G_k à partir de la matrice G_{k-1} de l'itération précédente.
Les matrices G_{k-1} et G_k doivent être « proches » au sens de la norme matricielle.

Variation de modèle

- Modèle linéaire de g en x_{k-1} : $\hat{g}_{k-1}(x) = g(x_{k-1}) + G_{k-1}(x - x_{k-1})$
- Modèle linéaire de g en x_k : $\hat{g}_k(x) = g(x_k) + G_k(x - x_k)$
- Différence entre les modèles en x_{k-1} et x_k

$$\begin{aligned}
 \hat{g}_k(x) &= g(x_k) + G_k(x - x_k) && \text{car } g(x_k) = g(x_{k-1}) + G_k(x_k - x_{k-1}) \\
 &= g(x_{k-1}) + G_k(x_k - x_{k-1}) + G_k(x - x_k) && \text{par définition de } G_k \\
 &= g(x_{k-1}) + G_k(x - x_{k-1}) \\
 &= \hat{g}_{k-1}(x) - G_{k-1}(x - x_{k-1}) + G_k(x - x_{k-1}) && \text{car } \hat{g}_{k-1}(x) = g(x_{k-1}) + G_{k-1}(x - x_{k-1}) \\
 &= \hat{g}_{k-1}(x) + (G_k - G_{k-1})(x - x_{k-1}) && \text{par définition de } \hat{g}_{k-1}
 \end{aligned}$$

$$\Rightarrow \hat{g}_k(x) - \hat{g}_{k-1}(x) = (G_k - G_{k-1})(x - x_{k-1})$$

Objectif

Conserver un modèle linéaire de g en x_k

$$\hat{g}_k(x) = g(x_k) + G_k(x - x_k) \quad \text{avec } G_k \approx \nabla g(x_k)^T$$

sans calculer explicitement G_k

Equation de la sécante

On choisit une matrice $G_k \in \mathbb{R}^{n \times n}$ vérifiant :

$$g(x_k) - g(x_{k-1}) = G_k(x_k - x_{k-1}) \Leftrightarrow y_{k-1} = G_k d_{k-1} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

→ **équation de la sécante** entre x_{k-1} et x_k

Choix de G

Il existe une infinité de matrices G vérifiant l'équation de la sécante :

n^2 inconnues (composantes de $G \in \mathbb{R}^{n \times n}$)

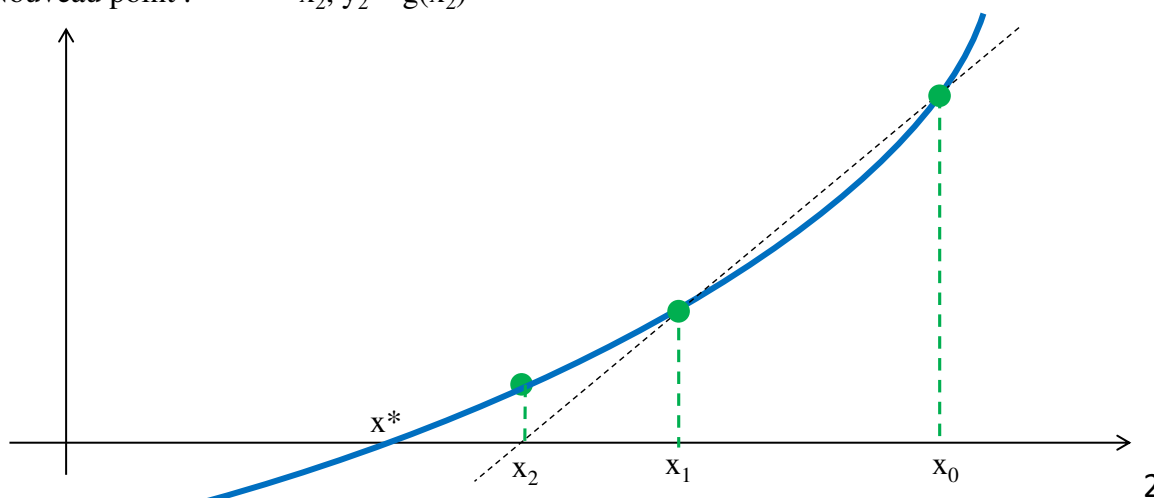
n équations

Chaque ligne de G définit un hyperplan de \mathbb{R}^n passant par x_{k-1} et x_k

→ infinité d'hyperplans possibles

Illustration à une variable

- Equation non linéaire : $g(x) = 0$
- Points initiaux : $x_0, y_0 = g(x_0)$
 $x_1, y_1 = g(x_1)$ → **Sécante** en x_1 : $y = y_0 - \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$
- Intersection axe x : $y = 0 \Rightarrow x_2 = x_0 - \frac{x_1 - x_0}{y_1 - y_0} y_0$
- Nouveau point : $x_2, y_2 = g(x_2)$



Pour la sécante, la dérivée est estimée par la pente de la droite.

Mise à jour de Broyden

L'écart entre les modèles $\hat{g}_{k-1}(x)$ et $\hat{g}_k(x)$ est minimal en choisissant G_k solution de

$$\min_{G \in \mathbb{R}^{n \times n}} \|G - G_{k-1}\| \quad \text{sous } y_{k-1} = G d_{k-1} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

Formule de Broyden :
$$G_k = G_{k-1} + \frac{(y_{k-1} - G_{k-1}d_{k-1})d_{k-1}^T}{d_{k-1}^T d_{k-1}} \rightarrow \text{solution optimale}$$

Convergence

- La matrice G ne converge pas forcément vers $\nabla g \rightarrow$ ne compromet pas la convergence
- Les méthodes de quasi-Newton et de Newton peuvent converger vers des solutions différentes.
- La méthode de quasi-Newton converge généralement moins vite que la méthode de Newton, mais nécessite beaucoup moins d'appels de la fonction g (pas de calcul de gradient).

\rightarrow Peu de résultats théoriques

\rightarrow Méthode efficace en pratique, comportement à vérifier et adapter au cas par cas

Noter que, pour une fonction $g : \mathbb{R} \mapsto \mathbb{R}$, la formule de Broyden devient :

$$G_k = G_{k-1} + \frac{(y_{k-1} - G_{k-1}d_{k-1})d_{k-1}^T}{d_{k-1}^T d_{k-1}} = G_{k-1} + \frac{y_{k-1}}{d_{k-1}} - G_{k-1} = \frac{y_{k-1}}{d_{k-1}} = \frac{g(x_k) - g(x_{k-1})}{x_k - x_{k-1}}$$

et en conséquence, la mise-à-jour $x_{k+1} = x_{k-1} - y_{k-1}/G_k$ revient bien à la méthode de la sécante avec une estimation de la pente de la droite.

Comparaison Newton – Quasi-Newton

- Fonction : $g(x) = x^2 - 1$
- Dérivée : $g'(x) = 2x$
- Solution : $x = 1$

Quasi - Newton

Iteration	x(k)	g(x)=x**2-1	dg/dx	Erreur
	5,00000000	2,4E+01		4,0E+00
0	4,00000000	1,5E+01	9,0000	3,0E+00
1	2,33333333	4,4E+00	6,3333	1,3E+00
2	1,63157895	1,7E+00	3,9649	6,3E-01
3	1,21238938	4,7E-01	2,8440	2,1E-01
4	1,04716672	9,7E-02	2,2596	4,7E-02
5	1,00443349	8,9E-03	2,0516	4,4E-03
6	1,00010193	2,0E-04	2,0045	1,0E-04
7	1,00000023	4,5E-07	2,0001	2,3E-07
8	1,00000000	2,3E-11	2,0000	1,1E-11

Newton

Iteration	x(k)	g(x)=x**2-1	g'(x)=2x	Erreur
0	4,00000000	1,5E+01	8,0000	3,0E+00
1	2,12500000	3,5E+00	4,2500	1,1E+00
2	1,29779412	6,8E-01	2,5956	3,0E-01
3	1,03416618	6,9E-02	2,0683	3,4E-02
4	1,00056438	1,1E-03	2,0011	5,6E-04
5	1,00000016	3,2E-07	2,0000	1,6E-07
6	1,00000000	2,5E-14	2,0000	1,3E-14

EXERCICE 17: Recherche d'un zéro par la méthode de quasi-Newton (1)

Soit la fonction $f(x) = e^{x-3} + x - 2$. On veut trouver le zéro de $f(x)$ par la méthode de quasi-Newton. On initialisera l'algorithme avec $x_0 = 2$ et $x_1 = 1.9$.

EXERCICE 18: Recherche d'un zéro par la méthode de quasi-Newton (2)

Soit la fonction $f(x) = \cos x$. On veut trouver le zéro de $f(x)$ par la méthode de quasi-Newton. On initialisera l'algorithme avec $x_0 = 1$ et $x_1 = 1.1$.

5 Méthodes de type Newton

5.1 Principe

On s'intéresse maintenant à la **minimisation** d'une fonctionnelle.

Problème sans contrainte

$$\min_{x \in \mathbb{R}^n} f(x) \quad \rightarrow \quad \begin{array}{l} \text{gradient : } g(x) = \nabla f(x) \\ \text{hessien : } H(x) = \nabla^2 f(x) \end{array}$$

Condition nécessaire de minimum local

$$x^* \text{ minimum local} \Rightarrow \begin{cases} g(x^*) = 0 \\ H(x^*) \geq 0 \end{cases} \quad (\text{hessien semi-défini positif})$$

Recherche des points stationnaires

Application de la méthode de Newton au système d'équations non linéaires : $\boxed{g(x) = 0}$

$$\text{Modèle linéaire de } g \text{ en } x_k : \hat{g}_k(x) = g(x_k) + G_k(x - x_k) \text{ avec } \begin{cases} g(x_k) = \nabla f(x_k) \\ G_k = \nabla g(x_k)^T = \nabla^2 f(x_k) = H_k \end{cases}$$

- Méthode de **Newton** : $G = H \rightarrow$ calcul explicite du hessien à chaque itération
- Méthode de **quasi-Newton** : $G =$ approximation de H
construite à partir des itérations précédentes
sans calcul explicite du hessien

Donc, on utilise le fait qu'au minimum, le gradient doit être nul (= dont on cherche le zéro avec Newton), et que le Hessien est semi-défini positif (on est dans un trou !).

5.2 Méthode de Newton

Première approche : on utilise un modèle linéaire de ∇f en x

Modèle linéaire de $g=\nabla f$ en x_k

$$\hat{g}_k(x) = g(x_k) + G_k(x - x_k) \quad \text{avec} \quad \begin{cases} g(x_k) = \nabla f(x_k) \\ G_k = \nabla g(x_k)^T = \nabla^2 f(x_k) = H_k \end{cases}$$

- Itération : $x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \rightarrow$ équations de Newton
- Condition d'arrêt : $\|\nabla f(x_k)\| \leq \varepsilon$

Difficultés

- Calcul explicite et inversion du hessien $\nabla^2 f(x_k)$ à chaque itération \rightarrow coûteux
- Convergence non garantie même près de la solution
 \rightarrow mêmes difficultés que pour la résolution d'équations
- 1^{ère} condition nécessaire de minimum : $\nabla f(x^*)=0$
 \rightarrow point stationnaire $x^* =$ minimum local, maximum local ou point selle
 \rightarrow 2^{ème} condition nécessaire de minimum à vérifier : $\nabla^2 f(x^*) \geq 0$

Adaptations

- Méthodes de **quasi-Newton** $\rightarrow G_k =$ **approximation** du hessien $\nabla^2 f(x_k)$
- Techniques de **globalisation** \rightarrow **Contrôle** de la décroissance de f

Deuxième approche : on utilise un modèle quadratique de f en x , puis on utilise ce modèle afin

d'estimer ∇f en x

Modèle quadratique de f en x_k

- Développement de Taylor à l'ordre 2 de f en x_k

$$f(x) = f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2} (x - x_k)^T \nabla^2 f(x_k) (x - x_k) + o(\|x - x_k\|^2)$$

- Modèle quadratique en x_k

$$\hat{f}_k(x) = f_k(x_k) + g_k^T (x - x_k) + \frac{1}{2} (x - x_k)^T H_k (x - x_k)$$

- Lien entre le modèle de f et le modèle de $g = \nabla f$

$$\nabla \hat{f}_k(x) = g_k + H_k (x - x_k) = \hat{g}_k(x)$$

Minimisation du modèle de f en x_k

- Conditions suffisantes de minimum local : $\min_{x \in \mathbb{R}^n} \hat{f}_k(x) \Leftarrow \begin{cases} \nabla \hat{f}_k(x^*) = \hat{g}_k(x^*) = 0 \\ \nabla^2 \hat{f}_k(x^*) = H_k > 0 \end{cases}$

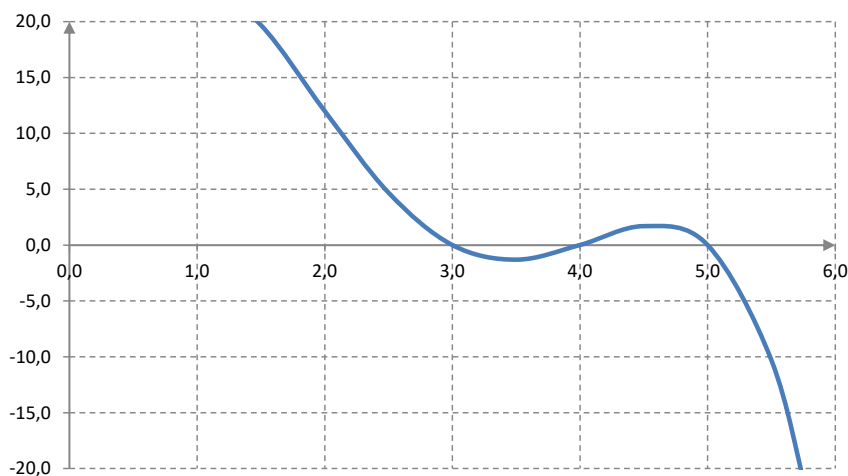
- Si le hessien de f en x_k est défini positif : $\nabla^2 f(x_k) > 0$
Minimisation du modèle quadratique de f en x_k
 \Leftrightarrow Méthode de Newton en x_k pour résoudre $\nabla f(x) = 0$

- Sinon la méthode de Newton n'est pas directement applicable pour une minimisation

Comme précédemment, on utilise Newton avec pour déplacement $x_{k+1} = x_k - H_k^{-1} g_k$ **Exemple :**

Méthode de Newton

- Fonction : $f(x) = -x^4 + 12x^3 - 47x^2 + 60x$
- Dérivée : $f'(x) = -4x^3 + 36x^2 - 94x + 60 \rightarrow 3 \text{ zéros}$
 $\rightarrow 1 \text{ minimum local, } 2 \text{ maxima locaux}$



Regardons maintenant plus précisément ce qu'il se passe lorsqu'on applique la méthode de Newton aux points $x = 3$, $x = 4$ et $x = 5$.

On a :

$$\begin{aligned}f(x) &= -x^4 + 12x^3 - 47x^2 + 60x \\ \nabla f(x) &= -4x^3 + 36x^2 - 94x + 60 \\ \nabla^2 f(x) &= -12x^2 + 72x - 94\end{aligned}$$

Le modèle quadratique de f en x_0 est défini par :

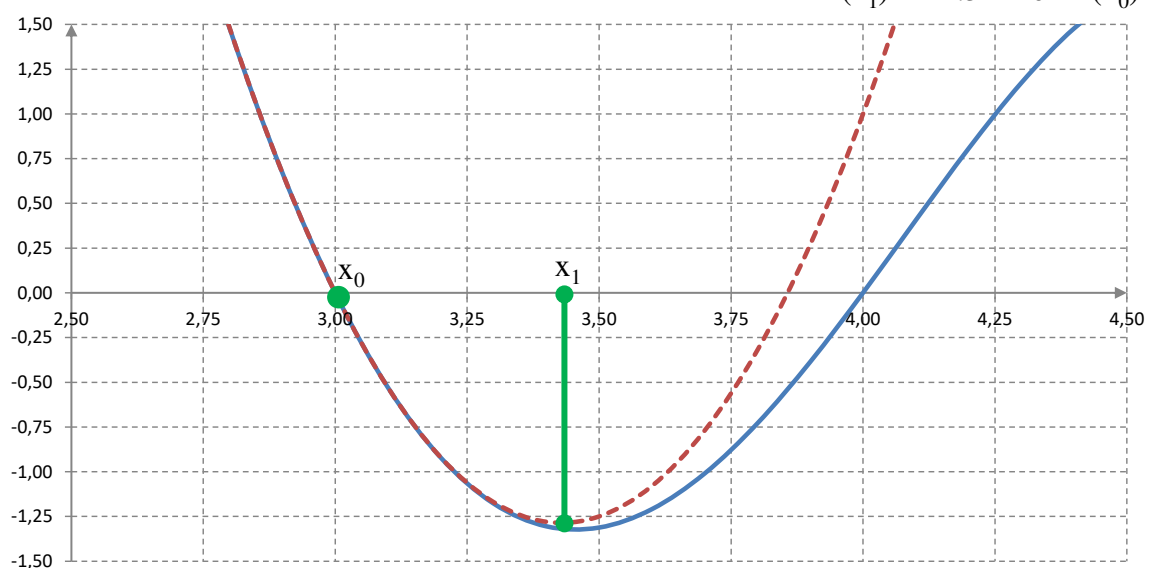
$$\hat{f}(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2$$

On évalue :

$$\begin{array}{lll}\text{en } x_0 = 3, & \hat{f}(3) = 0 - 6(x - 3) + \frac{1}{2}14(x - 3)^2 & = 7x^2 - 48x + 81 \\ \text{en } x_0 = 4, & \hat{f}(4) = 0 + 4(x - 4) + \frac{1}{2}2(x - 4)^2 & = x^2 - 4x \\ \text{en } x_0 = 5, & \hat{f}(5) = 0 - 10(x - 5) - \frac{1}{2}34(x - 5)^2 & = -17x^2 + 160x - 375\end{array}$$

Méthode de Newton

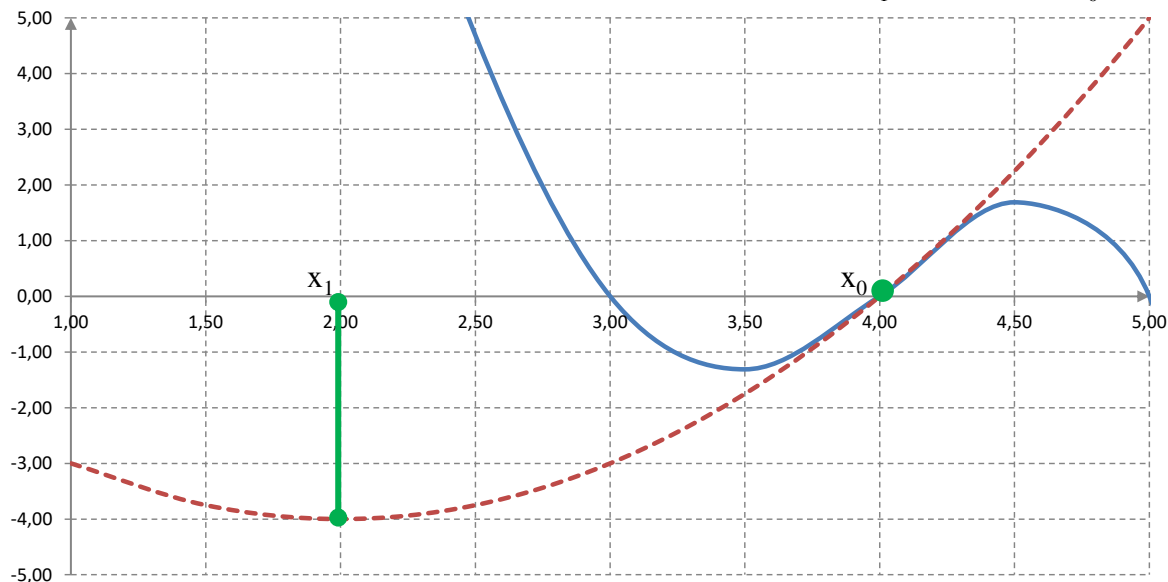
- Modèle quadratique en $x_0 = 3$: $\hat{f}_0(x) = 7x^2 - 48x + 81$
- Itération de Newton en $x_0 = 3$: $\min_x \hat{f}_0(x) \rightarrow x_1 = \frac{24}{7} \rightarrow \text{Meilleur que } x_0$
 $f(x_1) = -1.32 < 0 = f(x_0)$



$$x_{k+1} = x_k - H_k^{-1} g_k \stackrel{x_0=3}{\Rightarrow} x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = 3 - \frac{-6}{14} = \frac{24}{7}$$

Méthode de Newton

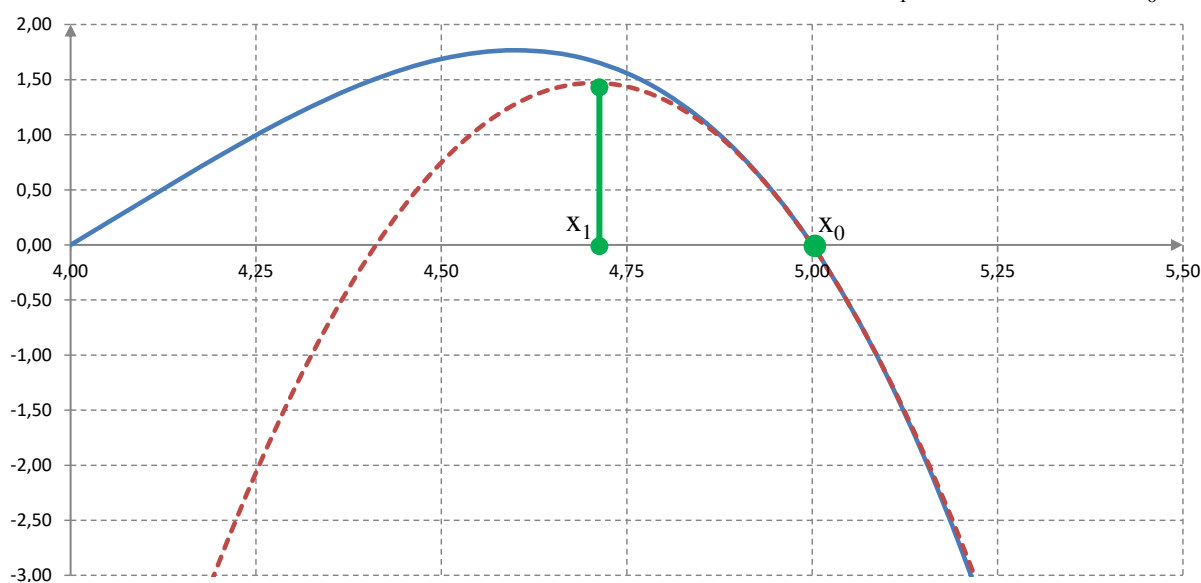
- Modèle quadratique en $x_0 = 4$: $\hat{f}_0(x) = x^2 - 4x$
- Itération de Newton en $x_0 = 4$: $\min_x \hat{f}_0(x) \rightarrow x_1 = 2 \rightarrow$ Moins bon que x_0
 $f(x_1) = 12 > 0 = f(x_0)$



$$x_{k+1} = x_k - H_k^{-1} g_k \stackrel{x_0=4}{\Rightarrow} x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = 4 - \frac{4}{2} = 2$$

Méthode de Newton

- Modèle quadratique en $x_0 = 5$: $\hat{f}_0(x) = -17x^2 + 160x - 375$
- Itération de Newton en $x_0 = 5$: $\min_x \hat{f}_0(x) \rightarrow x_1 = \frac{80}{17} \rightarrow$ Moins bon que x_0 (maximise)
 $f(x_1) = 1.513 > 0 = f(x_0)$



$$x_{k+1} = x_k - H_k^{-1} g_k \stackrel{x_0=5}{\Rightarrow} x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = 5 - \frac{-10}{-34} = \frac{80}{17}$$

EXERCICE 19: Minimisation par la méthode de Newton (1)

Soit la fonction $f(x) = x^3 + 2x^2 - 13x + 10$. On veut minimiser cette fonction.

1. rappeler les conditions nécessaires que doivent respecter un minimum.
2. trouver mathématiquement le ou les minimums de la fonction.
3. Pour chaque minimum trouvé à la question précédente, indiquer s'il est local ou global.
4. appliquer la méthode de Newton pour trouver le minimum de la fonction avec une précision $\epsilon = 10^{-6}$.

EXERCICE 20: Minimisation par la méthode de Newton (2)

Soit la fonction $f(x, y) = e^x + e^{-x} + e^y + e^{-y}$. On veut minimiser cette fonction.

1. Calculer ∇f et $\nabla^2 f$.
2. Appliquer la méthode de Newton pour trouver le minimum de la fonction avec une précision $\epsilon = 10^{-6}$.

5.3 Méthode de quasi-Newton**Objectif**

Conserver un modèle quadratique de f en x_k

$$\hat{f}_k(x) = f_k(x_k) + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k(x - x_k) \quad \text{avec } H_k \approx \nabla^2 f(x_k)$$

sans calculer explicitement H_k

Mise à jour de Broyden

On peut appliquer la méthode de Broyden à la résolution de $g(x) = \nabla f(x) = 0$.

$$H_k = H_{k-1} + \frac{(y_{k-1} - H_{k-1}d_{k-1})d_{k-1}^T}{d_{k-1}^T d_{k-1}} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

Inconvénients

- H_k n'est pas forcément symétrique
- H_k n'est pas forcément positive

→ Modifications de la méthode si l'on souhaite avoir $H_k \approx \nabla^2 f(x_k)$

→ Formules BFGS, DFP ou SR1

Méthode BFGS :**Equation sécante**

Pour la **résolution d'équations**, on cherche la matrice H_k

- vérifiant l'équation sécante $H_k d_{k-1} = y_{k-1}$
- la plus « proche » possible de H_{k-1} → **formule de Broyden**
- sans condition particulière sur la forme de la matrice

Pour une **minimisation**, on impose de plus à la matrice H_k d'être

- symétrique
- définie positive → **formule BFGS**

Méthode de résolution

La matrice H_{k-1} issue de l'itération précédente est symétrique, définie positive.

- On part de la factorisation de Cholesky de H_{k-1} : $H_{k-1} = L_{k-1} L_{k-1}^T$
- On cherche la matrice H_k à partir de H_{k-1} sous la forme : $H_k = A_k A_k^T$

Il faut exprimer la matrice A_k en fonction de L_{k-1} .

→ On décompose l'équation sécante en 2 équations.

$$H_k d_{k-1} = y_{k-1} \Leftrightarrow A_k A_k^T d_{k-1} = y_{k-1} \Leftrightarrow \begin{cases} x = A_k^T d_{k-1} \\ A_k x = y_{k-1} \end{cases}$$

Méthode de résolution de l'équation sécante

$$H_k d_{k-1} = y_{k-1} \Leftrightarrow A_k A_k^T d_{k-1} = y_{k-1} \Leftrightarrow \begin{cases} x = A_k^T d_{k-1} \\ A_k x = y_{k-1} \end{cases}$$

1. Pour x donné, on cherche A_k la plus « proche » possible de L_{k-1} vérifiant : $A_k x = y_{k-1}$
2. On détermine ensuite x en reportant l'expression de A_k dans : $x = A_k^T d_{k-1}$
3. On obtient $H_k = A_k A_k^T$ que l'on exprime en fonction de H_{k-1} .

Résolution

Notations sans indices : $L = L_{k-1}$, $A = A_k$
 $y = y_{k-1}$, $d = d_{k-1}$

1. En appliquant la formule de Broyden à L on obtient A en fonction de x : $A = L + \frac{(y - Lx)x^T}{x^T x}$
2. En reportant : $x = A^T d = L^T d + \frac{x(y - Lx)^T}{x^T x} d = L^T d + \frac{(y - Lx)^T d}{x^T x} x$

Il faut résoudre $x = L^T d + \frac{(y - Lx)^T d}{x^T x} x$ pour trouver x .

Résolution de l'équation sécante

On cherche $x \in \mathbb{R}^n$ vérifiant : $x = L^T d + \frac{(y - Lx)^T d}{x^T x} x$

- Pour qu'une solution existe, le vecteur $L^T d$ doit être colinéaire à x .

$$x = \alpha L^T d \Rightarrow x^T x = \alpha^2 d^T H d$$

- En reportant dans l'équation : $\alpha L^T d = L^T d + \frac{(y - Lx)^T d}{\alpha^2 d^T H d} \alpha L^T d \Rightarrow \alpha^2 L^T d = \frac{y^T d}{d^T H d} L^T d$
- Pour qu'une solution existe, on doit avoir $y^T d > 0 \rightarrow A = L + \frac{1}{y^T d} \left(\alpha y d^T L - \frac{y^T d}{d^T H d} H d d^T L \right)$
On obtient $H_k : H_k = A_k A_k^T$

Formule BFGS

- Mise à jour de H_k :
$$H_k = H_{k-1} + \frac{y_{k-1} y_{k-1}^T}{y_{k-1}^T d_{k-1}} - \frac{H_{k-1} d_{k-1} d_{k-1}^T H_{k-1}}{d_{k-1}^T H_{k-1} d_{k-1}} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$
- Mise à jour **symétrique, de rang 2**
- Mise à jour **définie positive** si $y_{k-1}^T d_{k-1} > 0$
- Initialisation** : $H_0 = I$ ou $H_0 = \tau I$ avec $\tau > 0$

Méthode de quasi Newton BFGS

- Le déplacement correspondant à une itération de la méthode de Newton est solution de

$$H_k d_k = -\nabla f(x_k) \Rightarrow d_k = -H_k^{-1} \nabla f(x_k)$$

→ La matrice utile pour l'itération de Newton est **l'inverse de H_k** .

- On inverse les 2 membres de la formule BFGS pour obtenir H_k^{-1} en fonction de H_{k-1}^{-1} .

$$H_k^{-1} = \left(I - \frac{d_{k-1} y_{k-1}^T}{d_{k-1}^T y_{k-1}} \right) H_{k-1}^{-1} \left(I - \frac{y_{k-1} d_{k-1}^T}{d_{k-1}^T y_{k-1}} \right) + \frac{d_{k-1} d_{k-1}^T}{d_{k-1}^T y_{k-1}} \quad \text{si } y_{k-1}^T d_{k-1} > 0 \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

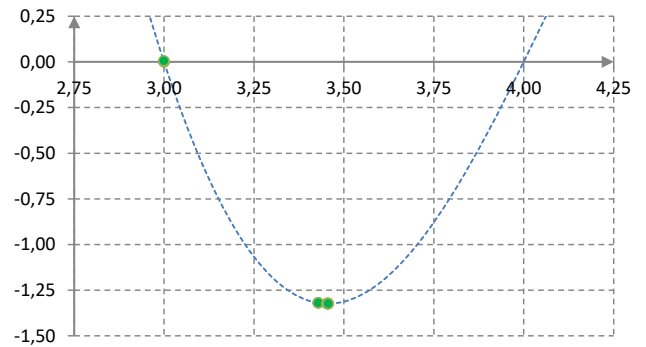
- Méthode élaborée par Broyden, Fletcher, Goldfarb, Shanno à la fin des années 1960
→ reconnue comme l'une des plus efficaces en pratique

Limitations

- Si la condition $y^T d > 0$ n'est pas vérifiée, on ne fait pas de mise à jour.
- Si le hessien n'est pas défini positif, la méthode BFGS ne converge pas vers le hessien.
→ cas d'un hessien indéfini à l'optimum, ou non défini positif loin de l'optimum
→ cas d'optimisation avec contraintes (hessien réduit du lagrangien positif \neq hessien complet)

Comparaison Newton – Quasi-Newton

- Fonction : $f(x) = -x^4 + 12x^3 - 47x^2 + 60x$
- Dérivée : $f'(x) = -4x^3 + 36x^2 - 94x + 60$
- Quasi-Newton : $h_k = \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$
- Point initial : $x_0 = 3 \rightarrow$ convergence
Autres points \rightarrow divergence ou maximum de f

**Quasi - Newton**

Itération	x(k)	f(x)	f'(x)	h(k)	Erreur
0	3,00000000	0,00000000	-6,00E+00	1,000	-4,56E-01
1	2,99900000	0,00600700	-6,01E+00	14,000	-4,57E-01
2	3,42857155	-1,31945027	-3,15E-01	13,267	-2,70E-02
3	3,45230465	-1,32362420	-3,79E-02	11,672	-3,28E-03
4	3,45554876	-1,32368634	-4,68E-04	11,527	-4,06E-05
5	3,45558934	-1,32368635	-7,27E-07	11,509	-6,32E-08
6	3,45558940	-1,32368635	-1,40E-11	11,509	-1,22E-12
7	3,45558940	-1,32368635	-5,68E-14	11,462	-4,44E-15

Newton

Itération	x	f(x)	f'(x)	f''(x)	Erreur
0	3,00000000	0,00000000	-6,00E+00	14,000	-4,56E-01
1	3,42857143	-1,31945023	-3,15E-01	11,796	-2,70E-02
2	3,45526446	-1,32368574	-3,74E-03	11,513	-3,25E-04
3	3,45558935	-1,32368635	-5,77E-07	11,509	-5,01E-08
4	3,45558940	-1,32368635	-5,68E-14	11,509	-4,88E-15

EXERCICE 21: Minimisation par la méthode de quasi-Newton

Soit la fonction $f(x) = x^3 + 2x^2 - 13x + 10$. On veut minimiser cette fonction avec la méthode de quasi-Newton. On partira de $x_0 = 1$ avec un delta $\delta = 0.1$ pour le premier pas. On arrêtera l'algorithme lorsque $\varepsilon < 10^{-3}$.

6 Recherche linéaire

6.1 Principes

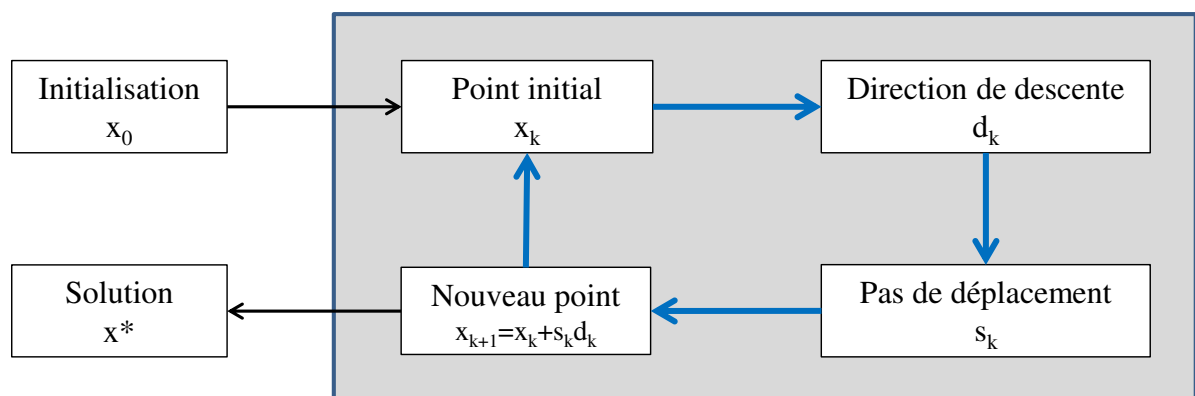
Problème sans contrainte

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

Etapes principales

A chaque itération

- Construction d'une direction de descente d_k à partir du point x_k
- Réglage du pas de déplacement s_k suivant d_k



Etapes principales

A chaque itération

- Construction d'une direction de descente d_k à partir du point x_k
- Réglage du pas de déplacement s_k suivant d_k

Direction de descente

d_k est une direction de descente en x_k si $\nabla f(x_k)^T d_k < 0$

La direction de descente est construite à partir du gradient et du hessien.

- **Plus forte pente** → gradient (méthode d'ordre 1)
- **Préconditionnement** → hessien (méthode d'ordre 2)

Pas de déplacement

Le pas de déplacement s_k suivant d_k doit vérifier $f(x_k + s_k d_k) < f(x_k)$

L'algorithme de recherche linéaire résout un problème de minimisation à une variable s

- Minimisation **exacte** → dichotomie (Fibonacci, nombre d'or, interpolation)
- Minimisation **approchée** → règles de pas acceptable (Armijo, Goldstein, Wolfe)

6.2 Direction de descente

a) Plus forte pente

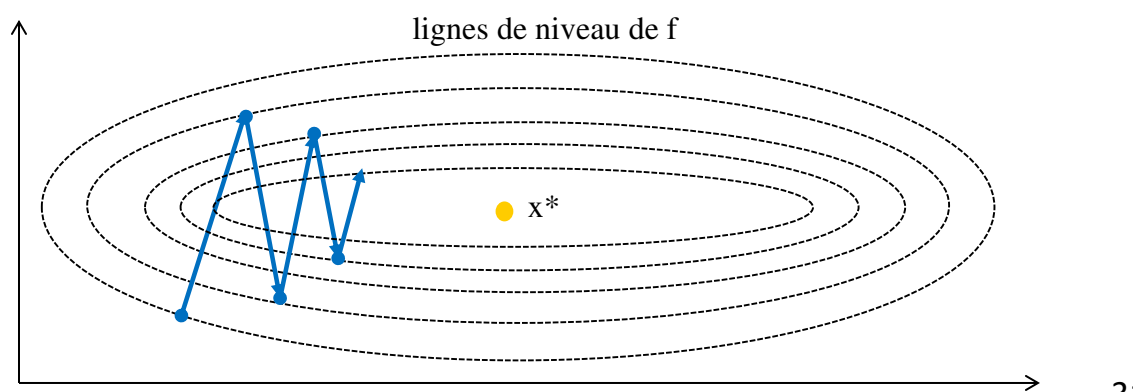
Plus forte pente

La direction de descente « naturelle » est celle du gradient = plus forte dérivée directionnelle

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

- Comportement caractéristique en zigzag
- **Convergence très lente** → méthode inefficace en général
- Améliorations possibles → gradient accéléré, méthode de Nesterov, préconditionnement

Illustration en dimension 2



Plus forte pente

Les directions successives de plus forte pente avec pas optimal sont **orthogonales**.

- Itération k

On cherche le minimum de f à partir de x_k suivant la direction $d_k = -\nabla f(x_k)$

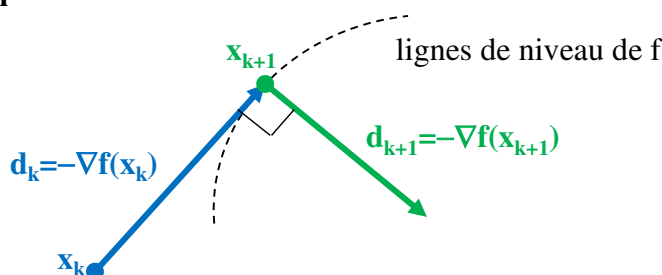
Le nouveau point est $x_{k+1} = x_k + s d_k$ avec le pas $s > 0$ solution de :

$$\min_{s \in \mathbb{R}} f(x_k + s d_k) \Rightarrow \frac{d}{ds} f(x_k + s d_k) = 0 \Rightarrow d_k^T \nabla f(x_k + s d_k) = 0 \Rightarrow d_k^T \nabla f(x_{k+1}) = 0$$

- Itération k+1

La direction de plus forte pente en x_{k+1} est $d_{k+1} = -\nabla f(x_{k+1}) \Rightarrow \boxed{d_k^T d_{k+1} = 0}$

Illustration



En effet, $\frac{d}{ds} f(x_k + s d_k) = \frac{d}{ds} (x_k + s d_k)^T \nabla f(x_k + s d_k) = d_k^T \nabla f(x_{k+1}) = 0$.

Or, $d_k = -\nabla f(x_k)$, et ainsi $d_k^T d_{k+1} = 0$.

Donc les direction consécutives sont **orthogonales**.

Plus forte pente

- Fonction

$$f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$$

- Direction

$$d = -\nabla f(x) = \begin{pmatrix} -x_1 \\ -9x_2 \end{pmatrix}$$

- Pas

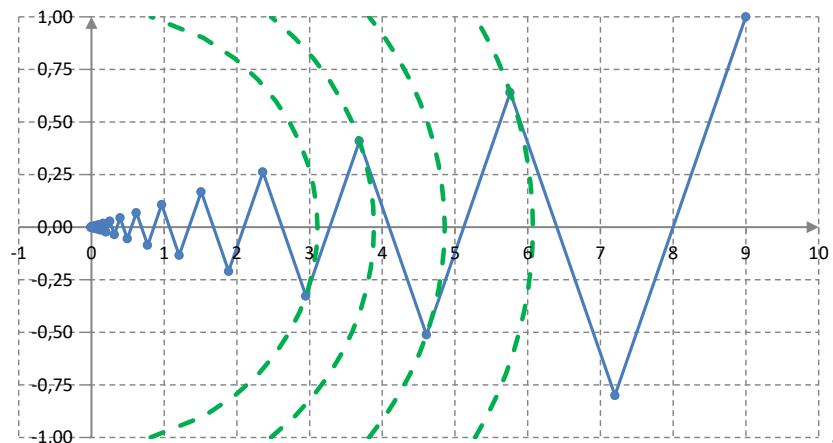
$$\min_s f(x + sd)$$

$$\rightarrow s = \frac{x_1^2 + 81x_2^2}{x_1^2 + 729x_2^2}$$

- Itération

$$x_{k+1} = x_k + s_k d_k$$

Iteration	x1	x2	f(x)	d1	d2	s	Erreur
0	9,000	1,000	45,000	-9,000	-9,000	0,2	9,055
1	7,200	-0,800	28,800	-7,200	7,200	0,2	7,244
2	5,760	0,640	18,432	-5,760	-5,760	0,2	5,795
3	4,608	-0,512	11,796	-4,608	4,608	0,2	4,636
4	3,686	0,410	7,550	-3,686	-3,686	0,2	3,709
5	2,949	-0,328	4,832	-2,949	2,949	0,2	2,967
10	0,966	0,107	0,519	-0,966	-0,966	0,2	0,972
20	0,104	0,012	0,006	-0,104	-0,104	0,2	0,104
30	1,11E-02	1,24E-03	6,90E-05	-1,11E-02	-1,11E-02	0,2	1,12E-02
40	1,20E-03	1,33E-04	7,95E-07	-1,20E-03	-1,20E-03	0,2	1,20E-03
50	1,28E-04	1,43E-05	9,17E-09	-1,28E-04	-1,28E-04	0,2	1,29E-04



Le minimum est atteint lorsque $d_k^T \nabla f(x_k + s d_k) = -\nabla f(x_k)^T \cdot \nabla f(x_k + s \nabla f(x_k)) = 0$.

$$-\begin{bmatrix} -x_1 \\ -9x_2 \end{bmatrix}^T \nabla f \left(\begin{bmatrix} x_1 - s x_1 \\ x_2 - 9s x_2 \end{bmatrix} \right) = \begin{bmatrix} x_1 \\ 9x_2 \end{bmatrix}^T \begin{bmatrix} -(x_1 - s x_1) \\ -9(x_2 - 9s x_2) \end{bmatrix} = s(x_1^2 + 729x_2^2) - (x_1^2 + 81x_2^2) = 0$$

D'où la valeur de s .

EXERCICE 22: Plus forte pente

Soit la fonction $f(x, y) = x^2 - xy + y^2 + 1$. On veut minimiser cette fonction en descendant dans la direction de plus forte pente.

1. Calculer ∇f et $\nabla^2 f$.
2. Quelle est la direction de plus forte pente ?
3. Déterminer le pas optimal de descente.
4. Utiliser la méthode descente par la plus forte pente sur 4 pas.
5. Utiliser la méthode descente de gradient accéléré.
6. Utiliser la méthode de Nesterov.
7. Utiliser le conditionnement avec la méthode de descente par la plus forte pente sur 4 pas.

b) Gradient accéléré

Gradient accéléré

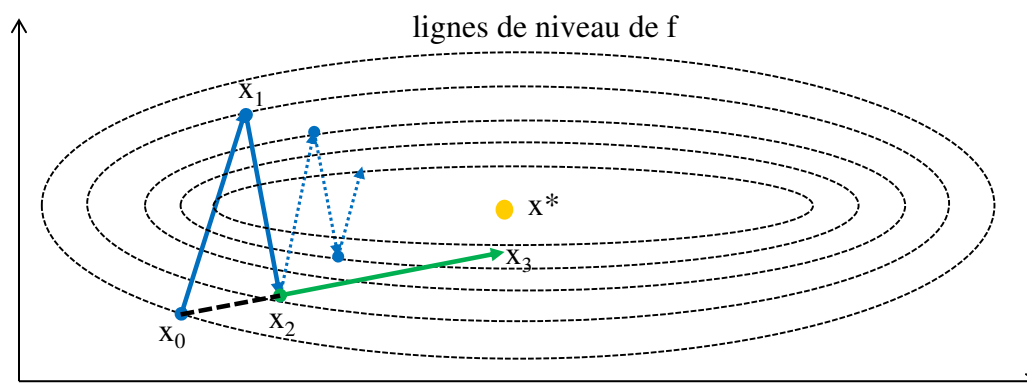
La méthode du gradient accéléré consiste à introduire des déplacements supplémentaires.

- On réalise p déplacements suivant la plus forte pente : $d_k \parallel -g(x_k)$

$$x_0 \xrightarrow{d_0} x_1 \xrightarrow{d_1} x_2 \xrightarrow{d_2} \dots \xrightarrow{d_{p-1}} x_p$$

- On réalise ensuite un déplacement suivant la direction : $d_p = x_p - x_0$.
 $d_p =$ **direction moyenne** des p derniers déplacements (en général $p = n$, dimension de x)

$$x_p \xrightarrow{d_p} x_{p+1}$$

**Justification** (en dimension 2)

- On suppose que la fonction à minimiser est une **fonction quadratique de 2 variables**.

$$f(x) = \frac{1}{2} x^T Q x + c^T x, \quad x \in \mathbb{R}^2 \quad \rightarrow \quad g(x) = \nabla f(x) = Qx + c \quad \rightarrow \quad x^* = -Q^{-1}c$$

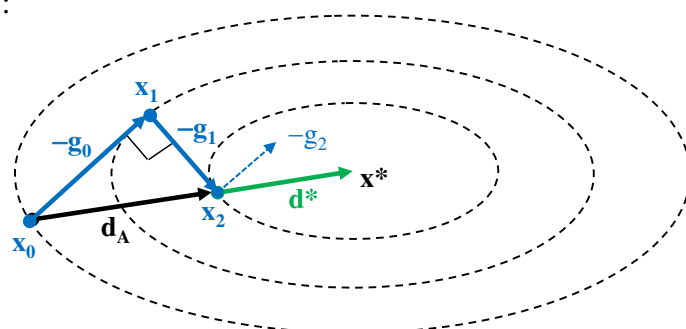
On a les relations :

$$\begin{cases} g_0 = Qx_0 + c \\ g_2 = Qx_2 + c \\ 0 = Qx^* + c \end{cases} \Rightarrow \begin{cases} x_2 - x_0 = \text{noté} \\ x^* - x_2 = \text{noté} \end{cases} \quad \boxed{\begin{matrix} d_A = Q^{-1}(g_2 - g_0) \\ d^* = -Q^{-1}g_2 \end{matrix}}$$

- On considère des itérations de plus forte pente à partir du point x_0 : $d_k \parallel g(x_k) = g_k$
 Les directions successives sont orthogonales :

$$\begin{cases} g_1 \perp g_0 \\ g_2 \perp g_1 \end{cases} \Rightarrow g_2 \parallel g_0 \Rightarrow g_2 - g_0 = \alpha g_2 \Rightarrow \boxed{d_A = \alpha Q^{-1}g_2}$$

Vrai seulement en dimension 2



- La direction d_A (de x_0 vers x_2) est parallèle à la direction d^* (de x_2 vers le minimum x^*).
 Le minimum se trouve suivant la direction d_A et peut être atteint dès la troisième itération.

Gradient accéléré

• Fonction $f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2 \rightarrow g(x) = \nabla f(x) = \begin{pmatrix} x_1 \\ 9x_2 \end{pmatrix} \rightarrow H(x) = \nabla^2 f(x) = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix}$

• Minimisation suivant $d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$: $\min_s f(x + sd) = f(x) + sd^T g(x) + \frac{1}{2} sd^T H(x) d$
 $\rightarrow s = -\frac{d^T g}{d^T H g} = -\frac{d_1 x_1 + 9d_2 x_2}{d_1^2 + 9d_2^2}$

• Point initial : $x^{(0)} = \begin{pmatrix} 9 \\ 1 \end{pmatrix}$

iteration	x1	x2	f(x)	d1	d2	s	Erreur
0	9,0000	1,0000	45,0000	-9,0000	-9,0000	0,2	9,0554
1	7,2000	-0,8000	28,8000	-7,2000	7,2000	0,2	7,2443
2	5,7600	0,6400	18,4320	-3,2400	-0,3600	1,8	5,7954
3	0,0000	0,0000	0,0000				0,0000

- Déplacements 1 et 2

Direction de plus forte pente : $d^{(0)} = -g(x^{(0)}) \rightarrow x^{(1)}$

$d^{(1)} = -g(x^{(1)}) \rightarrow x^{(2)}$

- Déplacement 3 suivant : $d = x^{(2)} - x^{(0)} \rightarrow x^{(3)} = x^*$

\rightarrow **convergence en 3 itérations**

$$x^{(3)} = x^{(2)} + s \cdot d = x^{(2)} - d^T g / d^T H d \cdot (x^{(2)} - x^{(0)})$$

Errata : (1) dans l'expression du min, il manque le carré sur le s pour le terme d'ordre 2. (2) le dénominateur de s est $d^T H d$ et non $d^T H g$.

c) Méthode de Nesterov**Méthode de Nesterov**

La méthode de Nesterov consiste à ajouter un déplacement intermédiaire à chaque itération.

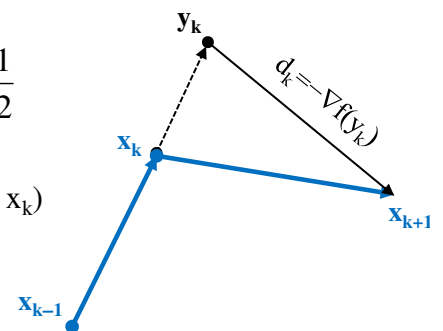
- Les 2 premières itérations sont identiques à la méthode de plus forte pente.
A partir de l'itération 3, chaque itération se compose de 2 déplacements.

- un déplacement à partir de x_k **suivant la direction $x_k - x_{k-1}$**

\rightarrow point y_k : $y_k = x_k + \theta_k (x_k - x_{k-1})$ avec $\theta_k = \frac{k-1}{k+2}$

- un déplacement de plus forte pente **à partir de y_k** (au lieu de x_k)

\rightarrow point x_{k+1} : $x_{k+1} = y_k + s_k d_k$ avec $d_k = -g(y_k)$



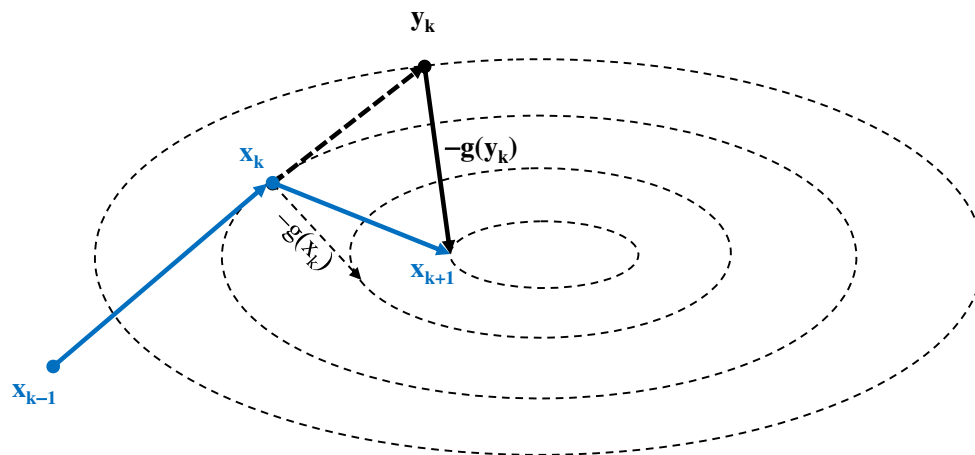
- Le déplacement complémentaire décale le point suivant la dernière direction de déplacement. Le nouveau point y_k est plus favorable pour une itération de plus forte pente, en particulier si la fonction présente une vallée étroite et allongée.

Méthode de Nesterov

- Itération composée de 2 déplacements

$$y_k = x_k + \theta_k (x_k - x_{k-1}) \quad \text{avec} \quad \theta_k = \frac{k-1}{k+2} \quad \text{à partir de } k=2$$

$$x_{k+1} = y_k + s_k d_k \quad \text{avec} \quad d_k = -g(y_k) \quad \text{et minimisation suivant } d_k$$



À l'ordre 2, on a : $\min f(y_k + s_k d_k) = f(y_k) + s_k d_k^T g(y_k) + \frac{1}{2} s^2 d_k^T H(y_k) d_k$. $\frac{df'(y_k)}{ds_k} = 0$ implique $d^T g(y_k) + s d_k^T H(y_k) d_k = 0$, et ainsi $s = -\frac{d_k^T g(y_k)}{d_k^T H(y_k) d_k}$, avec $d_k = -g(y_k)$.

à savoir $s = \frac{g(y_k)^T g(y_k)}{g(y_k)^T H(y_k) g(y_k)}$.

Méthode de Nesterov

- Fonction : $f(x) = \frac{1}{2} x_1^2 + \frac{9}{2} x_2^2$
- Itération :
$$\begin{cases} y_k = x_k + \theta_k (x_k - x_{k-1}) & \text{avec } \theta_k = \frac{k-1}{k+2} \\ x_{k+1} = y_k + s_k d_k & \text{avec } d_k = -\nabla f(y_k) = \begin{pmatrix} -y_1 \\ -9y_2 \end{pmatrix} \end{cases}$$

$$\text{et } \min_s f(y_k + s d_k) \rightarrow s = -\frac{d_1 y_1 + 9 d_2 y_2}{d_1^2 + 9 d_2^2}$$
- Comparaison à la méthode de plus forte pente

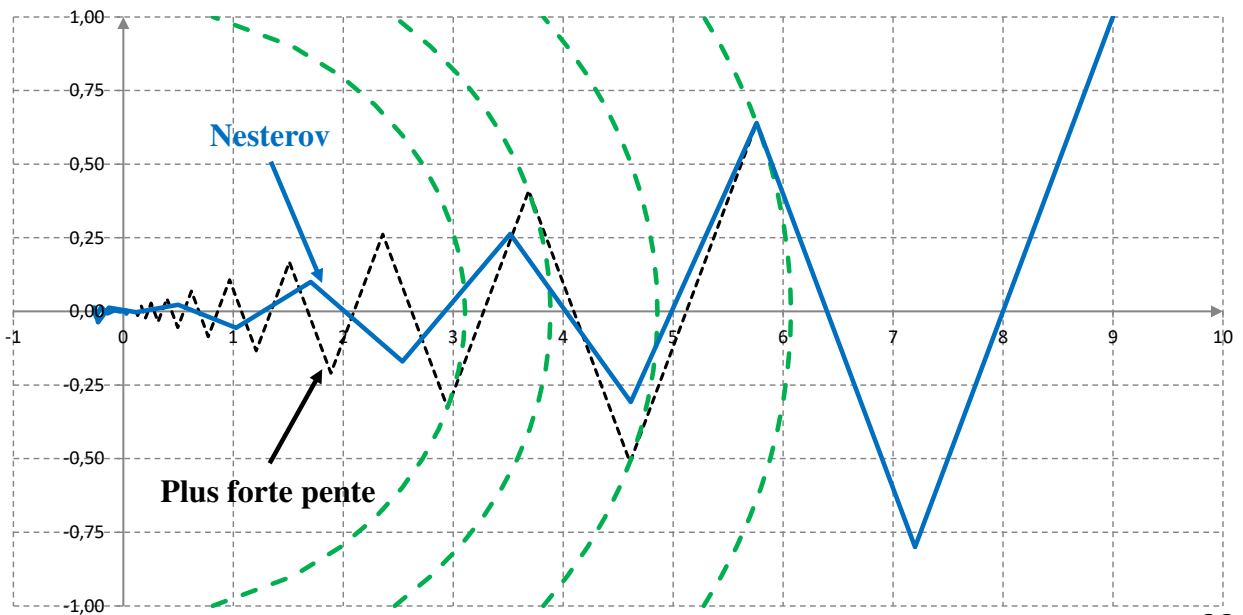
Iteration	Plus forte pente			Nesterov		
	x1	x2	f	x1	x2	f
0	9,000	1,000	45,0000	9,0000	1,0000	45,0000
1	7,200	-0,800	28,8000	7,2000	-0,8000	28,8000
2	5,760	0,640	18,4320	5,7600	0,6400	18,4320
3	4,608	-0,512	11,7965	4,6154	-0,3077	11,0769
4	3,686	0,410	7,5497	3,5187	0,2630	6,5018
5	2,949	-0,328	4,8318	2,5381	-0,1697	3,3507
6	2,359	0,262	3,0924	1,7054	0,0999	1,4991
7	1,887	-0,210	1,9791	1,0259	-0,0559	0,5403
8	1,510	0,168	1,2666	0,5004	0,0221	0,1274
9	1,208	-0,134	0,8106	0,1173	-0,0025	0,0069
10	0,966	0,107	0,5188	-0,1320	0,0129	0,0095

→ 2 itérations initiales identiques

→ après 10 itérations

Méthode de Nesterov

- Fonction : $f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$



Ce sont les différences de courbures suivant les différents axes qui produisent les oscillations, d'où l'idée de préconditionner le problème.

6.3 Préconditionnement

On rappelle que dans le cas où $H(x)$ est définie positive (et donc, par Cholesky, peut s'écrire $H(x) = LL^T$ où L triangulaire inférieure), alors le conditionnement optimal de f en x est $\tilde{x} = L^T x$, ce qui implique $\tilde{H}(\tilde{x}) = I$, à savoir que le courbure est 1 dans toutes les directions.

Préconditionnement

- On se donne une matrice H_k **symétrique définie positive**.
→ factorisation de Cholesky de H_k : $H_k = L_k L_k^T$

- Changement de variable : $\tilde{x}_k = L_k^T x_k$
- Direction de plus forte pente pour : $\tilde{f}(\tilde{x}_k) = f(x_k) = f(L_k^{-T} \tilde{x}_k)$
 $\tilde{d}_k = -\nabla \tilde{f}(\tilde{x}_k) = -\nabla f(L_k^{-T} \tilde{x}_k) = -L_k^{-1} \nabla f(L_k^{-T} \tilde{x}_k) = -L_k^{-1} \nabla f(x_k) = -L_k^{-1} d_k$
- Itération en \tilde{x}_k : $\tilde{x}_{k+1} = \tilde{x}_k - s_k \nabla \tilde{f}(\tilde{x}_k)$
- Itération en x_k : $x_{k+1} = L_k^{-T} \tilde{x}_{k+1} = L_k^{-T} (\tilde{x}_k - s_k \nabla \tilde{f}(\tilde{x}_k)) = L_k^{-T} (L_k^T x_k - s_k L_k^{-1} d_k)$
 $\Rightarrow x_{k+1} = x_k - s_k L_k^{-T} L_k^{-1} d_k = x_k - s_k H_k^{-1} d_k$
- Le **préconditionnement par la matrice H_k** revient à prendre comme direction de descente $d_k = -H_k^{-1} \nabla f(x_k)$

- On vérifie que d_k est une direction de descente
 $d_k^T \nabla f(x_k) = -\nabla f(x_k)^T H_k^{-1} \nabla f(x_k) < 0$ car H_k est définie positive

Choix du préconditionnement

Toute matrice H_k **symétrique définie positive** convient.

Cas d'un hessien défini positif

- Si le hessien $\nabla^2 f(x_k)$ est défini positif, on peut prendre $H_k = \nabla^2 f(x_k)$.

$$d_k = -H_k^{-1} \nabla f(x_k) = -\nabla^2 f(x_k)^{-1} \nabla f(x_k) \Rightarrow x_{k+1} = x_k - s_k \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

On obtient l'itération de Newton si le pas s_k vaut 1.

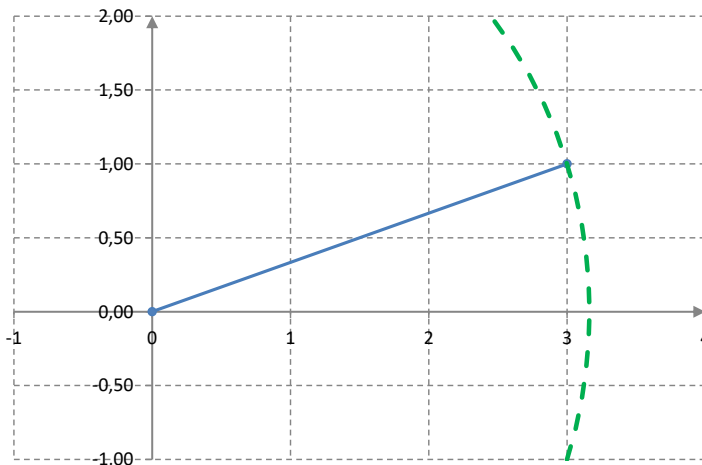
- On peut prendre pour H_k **l'approximation du hessien** donnée par une méthode quasi-Newton.

Cas d'un hessien non défini positif

- On peut effectuer la factorisation de Cholesky modifiée du hessien $\nabla^2 f(x_k)$ (ou de son approximation quasi Newton) pour obtenir une matrice définie positive.
- On peut ajouter un multiple de l'identité : $H_k = \nabla^2 f(x_k) + \tau I$ avec $\tau > 0$ assez grand.
- On peut également prendre H_k diagonale : $(H_k)_{ii} = \max\left(\varepsilon, \frac{\partial^2 f}{\partial x_i^2}(x_k)\right)$, $\varepsilon > 0$ à partir des dérivées secondes diagonales de f .

Préconditionnement

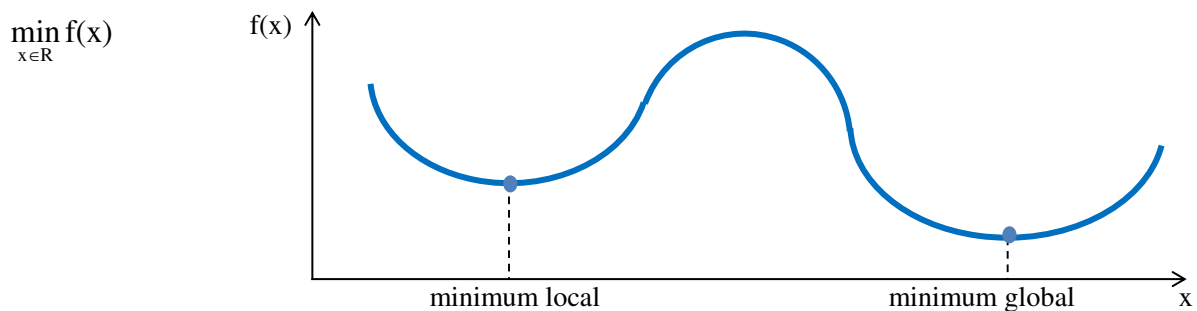
- Fonction : $f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2 \Rightarrow \nabla f(x) = \begin{pmatrix} x_1 \\ 9x_2 \end{pmatrix} \Rightarrow \nabla^2 f(x) = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}^T$
- Préconditionnement : $L = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix} \Rightarrow \begin{cases} \tilde{x}_1 = x_1 \\ \tilde{x}_2 = 3x_2 \end{cases} \Rightarrow \tilde{f}(\tilde{x}) = \frac{1}{2}\tilde{x}_1^2 + \frac{9}{2}\left(\frac{1}{3}\tilde{x}_2\right)^2 = \frac{1}{2}\tilde{x}_1^2 + \frac{1}{2}\tilde{x}_2^2$
- Direction : $\tilde{d} = -\nabla \tilde{f}(\tilde{x}) = \begin{pmatrix} -\tilde{x}_1 \\ -\tilde{x}_2 \end{pmatrix}$
- Pas : $\min_s \tilde{f}(\tilde{x} + s\tilde{d}) \rightarrow s = 1$
- Itération : $\tilde{x}_{k+1} = \tilde{x}_k + s_k \tilde{d}_k = 0$
→ convergence en 1 itération



6.4 Pas de déplacement

On s'intéresse maintenant à la façon de choisir un pas de déplacement lors de la recherche d'un minimum dans un intervalle.

Problème à une variable



Méthodes

- **Minimisation exacte**

On cherche à trouver un minimum local x^* avec une précision donnée

→ réduction itérative de l'intervalle de recherche : dichotomie

→ réduction optimale : Fibonacci, nombre d'or

- **Minimisation approchée**

On cherche une réduction suffisante de la fonction sans déterminer précisément le minimum x^*

→ règles d'acceptation (Armijo, Goldstein, Wolfe)

→ limitation du nombre d'évaluations de la fonction

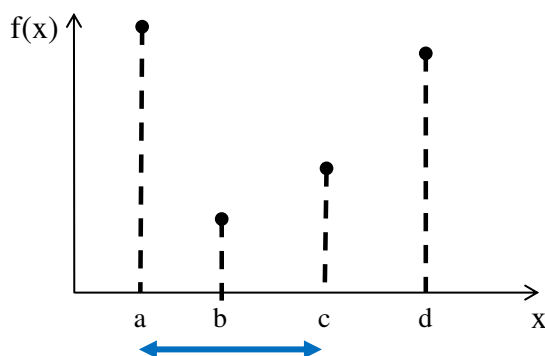
a) Recherche par dichotomie

Recherche par dichotomie

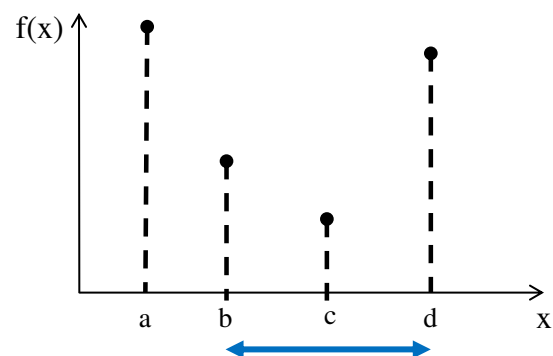
L'allure de la fonction f n'est pas connue.

On suppose qu'il existe un minimum unique dans l'intervalle de recherche $[a, d]$

- On évalue la fonction aux extrémités a et d : → $f(a)$, $f(d)$
- On évalue la fonction en 2 points b et c : $a < b < c < d$ → $f(b)$, $f(c)$
- On conserve soit l'intervalle $[a, c]$, soit l'intervalle $[b, d]$



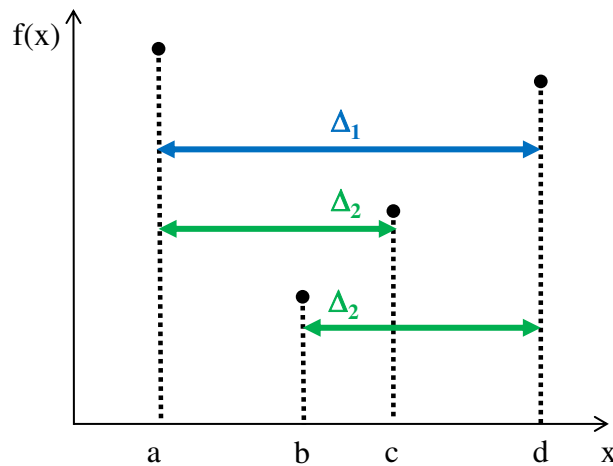
On conserve $[a, c]$



On conserve $[b, d]$

Réduction optimale de l'intervalle

- On cherche b et c pour que l'intervalle restant soit le plus petit possible.
La taille de l'intervalle restant ne doit pas dépendre du choix de [a,c] ou [b,d].
- On note : Δ_1 la longueur de l'intervalle initial $\rightarrow \Delta_1 = d - a$
 Δ_2 la longueur de l'intervalle restant $\rightarrow \Delta_2 = c - a$ si on garde [a,c]
ou $\Delta_2 = d - b$ si on garde [b,d]



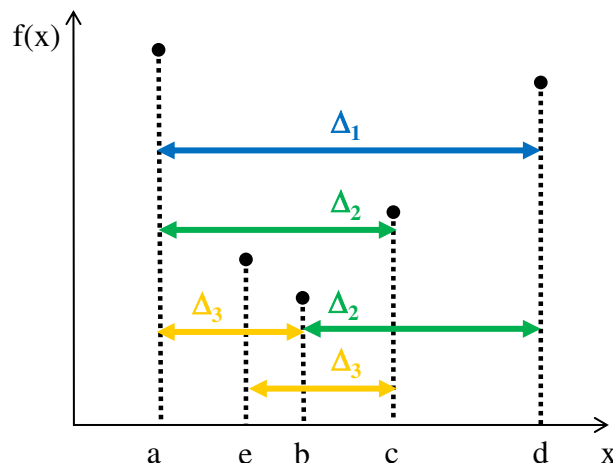
$$\Delta_2 = d - b = c - a$$

$$\Rightarrow a + d = b + c \Rightarrow \frac{a + d}{2} = \frac{b + c}{2}$$

Les points b et c doivent être symétriques par rapport au milieu de l'intervalle [a,d].

Réduction optimale de l'intervalle

- On suppose que l'intervalle restant est [a,c].
- Pour **réutiliser le point b** à l'itération suivante, on choisit le nouveau point e symétrique de b par rapport au milieu de l'intervalle [a,c].
 $\rightarrow \Delta_1 = d - a$
 $\rightarrow \Delta_2 = c - a = d - b$
 $\rightarrow \Delta_3 = b - a = c - e$



$$d - a = (d - b) + (b - a)$$

$$\Rightarrow \Delta_1 = \Delta_2 + \Delta_3$$

Les longueurs Δ_k des intervalles successifs vérifient :

$$\Delta_k = \Delta_{k+1} + \Delta_{k+2}$$

Réduction optimale de l'intervalle

- Les intervalles successifs sont de longueur Δ_k vérifiant : $\Delta_k = \Delta_{k+1} + \Delta_{k+2}$
- Après un nombre N d'itérations, on obtient un intervalle de longueur Δ_{N-1} .
- On définit la suite de nombres F_k par : $\Delta_k = F_{N-k} \Delta_{N-1}$ pour $k = 1, \dots, N-1$
- La suite $(F_n)_{n=1, \dots, N-1}$ vérifie

$$\begin{aligned} \Delta_k = \Delta_{k+1} + \Delta_{k+2} &\Rightarrow \frac{\Delta_k}{\Delta_{N-1}} = \frac{\Delta_{k+1}}{\Delta_{N-1}} + \frac{\Delta_{k+2}}{\Delta_{N-1}} \Rightarrow F_{N-k} = F_{N-k-1} + F_{N-k-2} \\ &\Rightarrow F_n = F_{n-1} + F_{n-2} \text{ pour } n = N-k, n = 3, 4, \dots, N-1 \\ \Delta_{N-1} = F_1 \Delta_{N-1} \text{ pour } k = N-1 &\Rightarrow F_1 = 1 \end{aligned}$$

- La suite est complètement déterminée par la valeur de F_2

$$\begin{cases} F_1 = 1 \\ F_2 \text{ à choisir} \\ F_n = F_{n-1} + F_{n-2} \text{ pour } n = 3, 4, \dots, N-1 \end{cases}$$

b) Méthode de Fibonacci

Méthode de Fibonacci

- Pour un nombre N d'itérations fixé, on choisit F_2 pour que Δ_{N-1} soit minimal.

$$\Delta_{N-1} = \frac{\Delta_1}{F_{N-1}} \text{ minimal} \Rightarrow F_{N-1} \text{ maximal} \Rightarrow F_2 \text{ maximal}$$

- Valeur maximale de F_2 : $\Delta_{N-1} \geq \frac{1}{2} \Delta_{N-2} \left(\text{car } \Delta_k \geq \frac{1}{2} \Delta_{k-1} \right) \Rightarrow F_1 \geq \frac{1}{2} F_2 \Rightarrow F_{2\max} = 2$
- On obtient la **suite de Fibonacci** : 1, 2, 3, 5, 8, 13, 21, ...

Nombre d'itérations

La méthode de Fibonacci est optimale si l'on fixe à l'avance la précision requise sur la solution.

- **Précision requise** : $\Delta_{N-1} = p \Rightarrow \Delta_{N-1} = \frac{\Delta_1}{F_{N-1}} \Rightarrow F_{N-1} = \frac{b-a}{p} \rightarrow \text{valeur de } N$
- Intervalle initial $[a, b]$: $\Delta_1 = b-a$

La méthode de Fibonacci donne le nombre minimal d'itérations N pour obtenir la solution **avec une précision donnée**.

- La disposition des premiers points b et c dépend de N : $\frac{\Delta_1}{\Delta_2} = \frac{F_{N-1}}{F_{N-2}} \rightarrow \Delta_2$

On a un intervalle initial $[a, b]$, $\Delta_1 = b - a$.

On veut atteindre la précision $\varepsilon = 10^{-5}$, il faut atteindre l'ordre n tel que $\varepsilon \geq \frac{b-a}{n}$.

On génère une suite de type Fibonacci :

- $F_1 = 1, F_2 = 1$
- $F_k = F_{k-1} + F_{k-2}$, pour $k = 2 \dots n$.

- $\Delta_n = \Delta_1 / F_n$.
- $\Delta_k = F_{n+1-k} \Delta_n$, pour $k = 1 \dots n - 1$.

La suite de Fibonacci est :

F_k	Δ_k	254	8.13e-03	50549	4.09e-05
1	1.00e+00	411	5.02e-03	81790	2.53e-05
4	6.18e-01	665	3.11e-03	132339	1.56e-05
5	3.82e-01	1076	1.92e-03	214129	9.64e-06
9	2.36e-01	1741	1.19e-03	346468	5.96e-06
14	1.46e-01	2817	7.33e-04	560597	3.68e-06
23	9.02e-02	4558	4.53e-04	907065	2.28e-06
37	5.57e-02	7375	2.80e-04	1467662	1.41e-06
60	3.44e-02	11933	1.73e-04	2374727	8.70e-07
97	2.13e-02	19308	1.07e-04	3842389	5.38e-07
157	1.32e-02	31241	6.61e-05	6217116	3.32e-07

c) Méthode du nombre d'or

Méthode du nombre d'or

La méthode de Fibonacci nécessite de changer la disposition des points à chaque itération.

La disposition des points n'est optimale que **pour une précision donnée**.

La méthode du nombre d'or est plus générale et plus simple à mettre en oeuvre.

- On impose un **rapport de réduction fixe** de l'intervalle à chaque itération.

$$\frac{\Delta_1}{\Delta_2} = \frac{\Delta_2}{\Delta_3} = \dots = \frac{\Delta_k}{\Delta_{k+1}} = \frac{\Delta_{k+1}}{\Delta_{k+2}} = \dots = \gamma \quad \text{avec} \quad \Delta_k = \Delta_{k+1} + \Delta_{k+2}$$

$$\Rightarrow \frac{\Delta_k}{\Delta_{k+1}} = 1 + \frac{\Delta_{k+2}}{\Delta_{k+1}} \Rightarrow \gamma = 1 + \frac{1}{\gamma} \Rightarrow \gamma^2 - \gamma - 1 = 0$$

- On obtient pour le rapport γ le nombre d'or :

→ **Méthode du nombre d'or** (ou de la **section dorée**)

$$\gamma = \frac{1 + \sqrt{5}}{2} \approx 1.618034$$

Optimalité

- La méthode du nombre d'or n'est pas optimale pour un nombre d'itérations donné N.

- Pour un nombre d'itérations grand : $\lim_{N \rightarrow \infty} \frac{F_N}{F_{N-1}} = \gamma$

→ La disposition des points de Fibonacci tend vers celle du nombre d'or.

Comparaison Fibonacci - Nombre d'or

Le rapport de réduction de l'intervalle après n itérations vaut :

- Pour la méthode de Fibonacci : $\frac{\Delta_1}{\Delta_n} = F_n$
- Pour la méthode du nombre d'or : $\frac{\Delta_1}{\Delta_n} = \gamma^{n-1}$

Nombre d'itérations

Une itération de la méthode de Fibonacci ou du nombre d'or correspond à une évaluation de f .

Nombre d'évaluations

Rapport de réduction	Fibonacci	Nombre d'or
10^{-2}	11	13
10^{-3}	15	18
10^{-4}	20	22
10^{-5}	25	27
10^{-6}	30	31

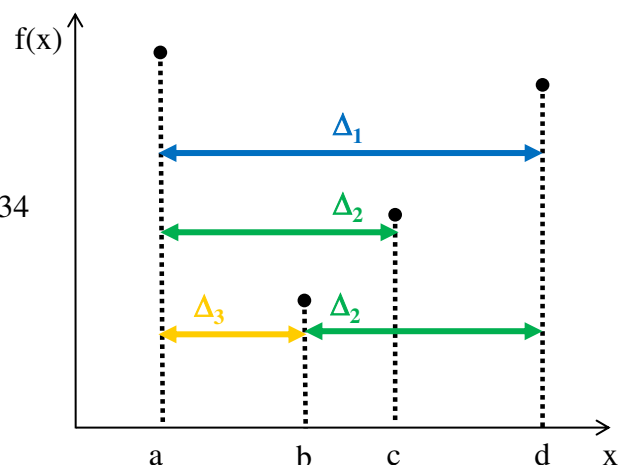
Méthode du nombre d'or

- Positionnement des points

$$\frac{\Delta_1}{\Delta_2} = \frac{\Delta_2}{\Delta_3} = \gamma = \frac{1+\sqrt{5}}{2} \approx 1.618034$$

$$\Rightarrow \begin{cases} \Delta_2 = \frac{1}{\gamma} \Delta_1 = r \Delta_1 \\ \Delta_3 = \frac{1}{\gamma^2} \Delta_1 = r^2 \Delta_1 \end{cases} \text{ avec } r = \frac{1}{\gamma} = \frac{\sqrt{5}-1}{2} \approx 0.618034$$

$$\Rightarrow \begin{cases} b = a + r^2 \Delta_1 \\ c = a + r \Delta_1 \\ d = a + \Delta_1 \end{cases}$$



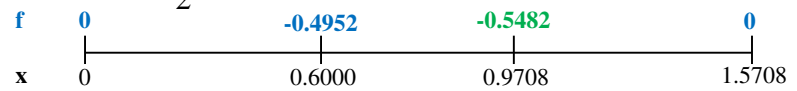
- Itération

$$\text{Si } f(b) < f(c) \rightarrow \begin{cases} d \leftarrow c \\ c \leftarrow b \\ \Delta_1 \leftarrow \Delta_2 \\ b = a + r^2 \Delta_1 \end{cases} \quad \text{Si } f(b) > f(c) \rightarrow \begin{cases} a \leftarrow b \\ b \leftarrow c \\ \Delta_1 \leftarrow \Delta_2 \\ c = a + r \Delta_1 \end{cases}$$

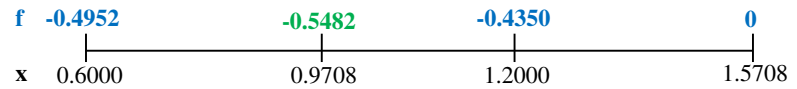
Méthode du nombre d'or

- Minimisation de $f(x) = -x \cos(x)$, $0 \leq x \leq \frac{\pi}{2}$

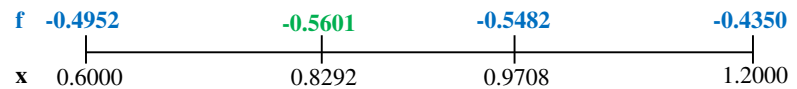
- Itération 1



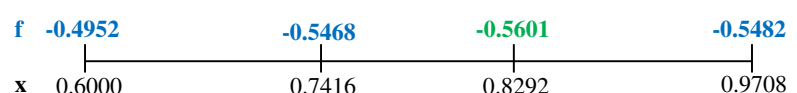
- Itération 2



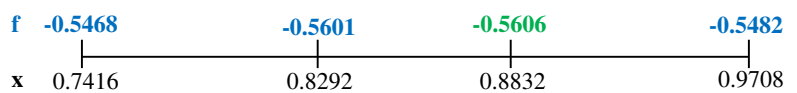
- Itération 3



- Itération 4



- Itération 5



- Solution : $x^* \approx 0.8832 \rightarrow f(x^*) \approx -0.5606$
 au lieu de $x^* \approx 0.8603 \rightarrow f(x^*) \approx -0.5611$

d) Minimisation approchée**Principes**

- Il n'est pas utile de réaliser une minimisation exacte suivant la direction de descente :
 \rightarrow nécessite un grand nombre d'évaluations de la fonction
 \rightarrow n'apporte pas une amélioration significative loin de la solution
- On peut se contenter d'une minimisation approchée
 \rightarrow 2 **règles d'acceptation** d'un pas de déplacement

Notations

- x_k = point courant $\rightarrow f(x_k)$
- d_k = direction de descente $\rightarrow \nabla f(x_k)^T d_k < 0$

- Variation de la fonction f dans la direction d_k : $\varphi(s) = f(x_k + s d_k)$, $s \geq 0$

Règles d'acceptation du pas

- Diminution suffisante** de la valeur de la fonction \rightarrow condition d'Armijo
 1^{ère} condition de Wolfe
- Déplacement suffisant** par rapport au point initial \rightarrow condition de Goldstein
 2^{ème} condition de Wolfe

Diminution suffisante

La fonction f doit décroître suffisamment pour accepter le pas.

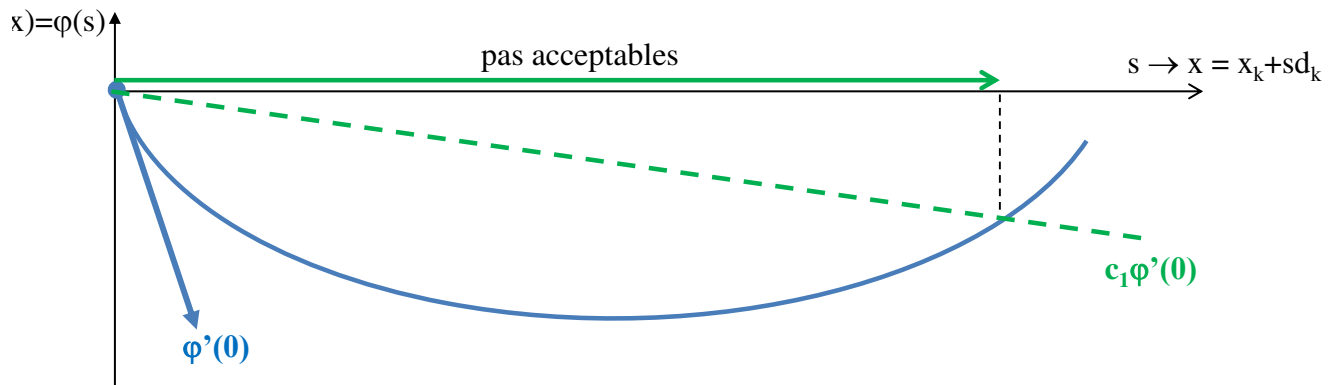
- Dérivée directionnelle de f suivant la direction d_k

$$\varphi(s) = f(x_k + sd_k), s \geq 0 \Rightarrow \varphi'(0) = \nabla f(x_k)^T d_k < 0$$

- On impose une diminution proportionnelle à la dérivée directionnelle, avec $0 < \varepsilon < c_1 < 0.5$

$$\varphi(s) < \varphi(0) + c_1 s \varphi'(0) \Leftrightarrow f(x_k + sd_k) < f(x_k) + c_1 s \nabla f(x_k)^T d_k \rightarrow \text{valeur typique } c_1 = 0.1$$

→ **Condition d'Armijo** ou 1^{ère} condition de Wolfe ou 1^{ère} condition de Goldstein

**Déplacement suffisant**

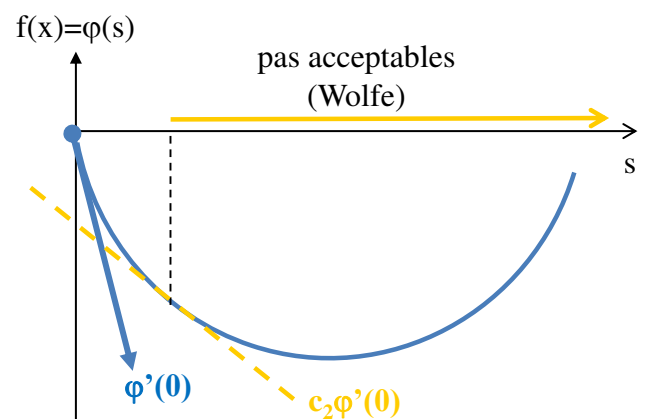
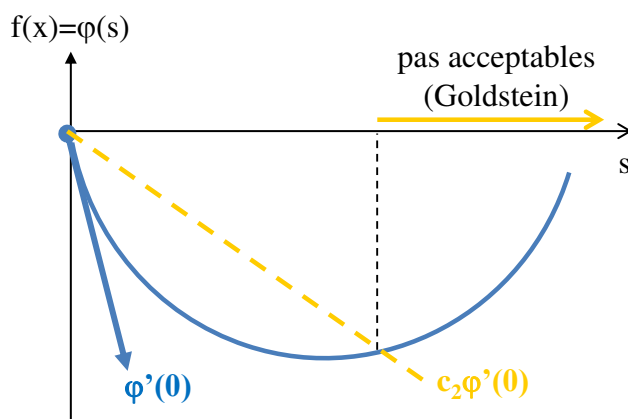
Le déplacement à partir du point initial doit être suffisant pour accepter le pas.

- Condition de Goldstein

$$\varphi(s) > \varphi(0) + c_2 s \varphi'(0) \Leftrightarrow f(x_k + sd_k) > f(x_k) + c_2 s \nabla f(x_k)^T d_k \rightarrow \text{valeur typique } c_2 = 0.9$$

- Condition de Wolfe : réduction de la dérivée

$$\varphi'(s) > c_2 \varphi'(0) \Leftrightarrow \nabla f(x_k + sd_k)^T d_k > c_2 \nabla f(x_k)^T d_k \rightarrow \text{valeur typique } c_2 = 0.9$$

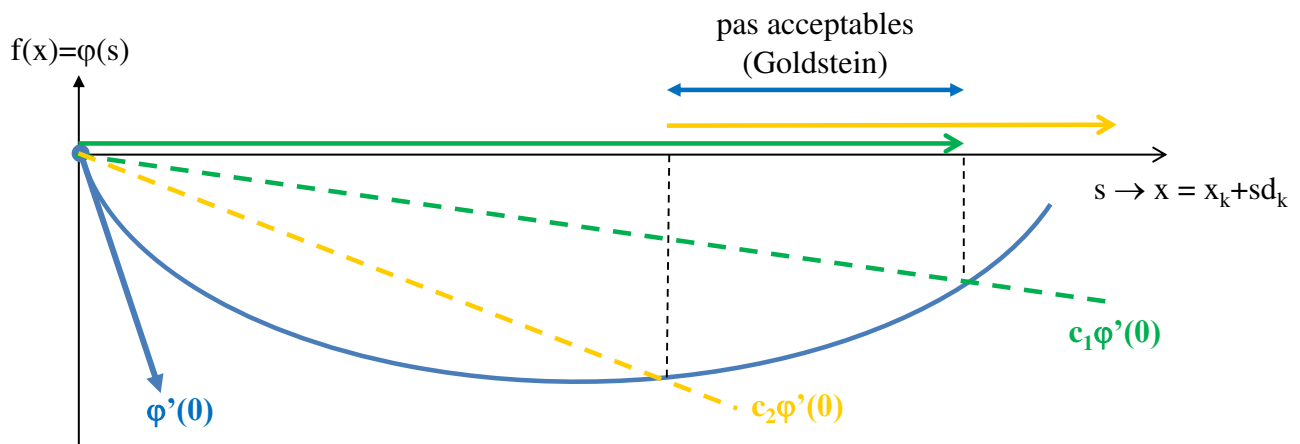


Récapitulatif

- **Conditions de Goldstein** (si la dérivée de f est coûteuse)

$$\begin{cases} \varphi(s) < \varphi(0) + c_1 s \varphi'(0) & \rightarrow c_1 \approx 0.1 \\ \varphi(s) > \varphi(0) + c_2 s \varphi'(0) & \rightarrow c_2 \approx 0.9 \end{cases}$$
- **Conditions de Wolfe** (si la dérivée de f est disponible)

$$\begin{cases} \varphi(s) < \varphi(0) + c_1 s \varphi'(0) & \rightarrow c_1 \approx 0.1 \\ \varphi'(s) > c_2 \varphi'(0) & \rightarrow c_2 \approx 0.9 \end{cases}$$
 ou $|\varphi'(s)| < c_2 |\varphi'(0)| \rightarrow$ assure théoriquement la convergence

**Méthode de dichotomie**

On cherche un pas s vérifiant les conditions de Goldstein : $\begin{cases} \varphi(s) < \varphi(0) + c_1 s \varphi'(0) \\ \varphi(s) > \varphi(0) + c_2 s \varphi'(0) \end{cases}$

Initialisation

- Intervalle initial : $[s_{\min}, s_{\max}]$ avec $s_{\min} = 0$
 s_{\max} assez grand
- Valeur initiale : $s=1$ (pas de Newton)

Itérations

Evaluation de $\varphi(s)$ et comparaison aux droites de Goldstein

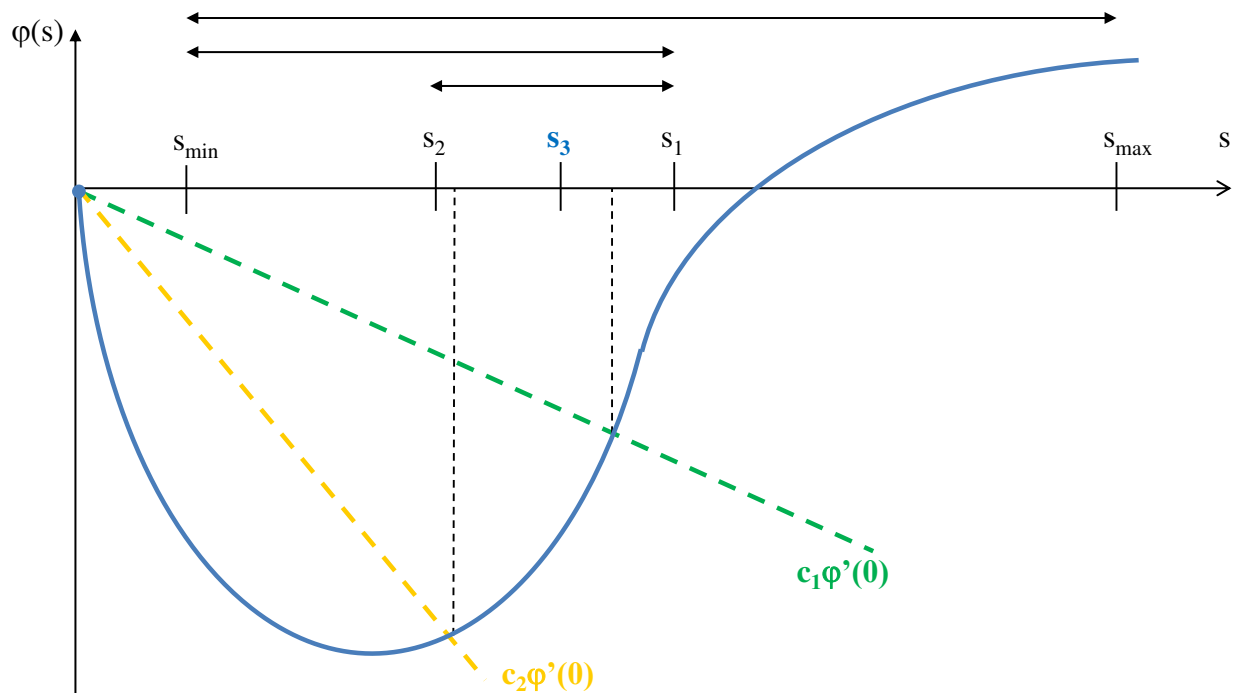
- Si s ne respecte pas la condition 1 \rightarrow amélioration insuffisante, pas trop grand : $s_{\max} = s$
 - Si s ne respecte pas la condition 2 \rightarrow déplacement insuffisant, pas trop petit : $s_{\min} = s$
- \rightarrow Réduction de l'intervalle $[s_{\min}, s_{\max}]$

\rightarrow Essai suivant : $s = \frac{1}{2}(s_{\min} + s_{\max})$

Arrêt

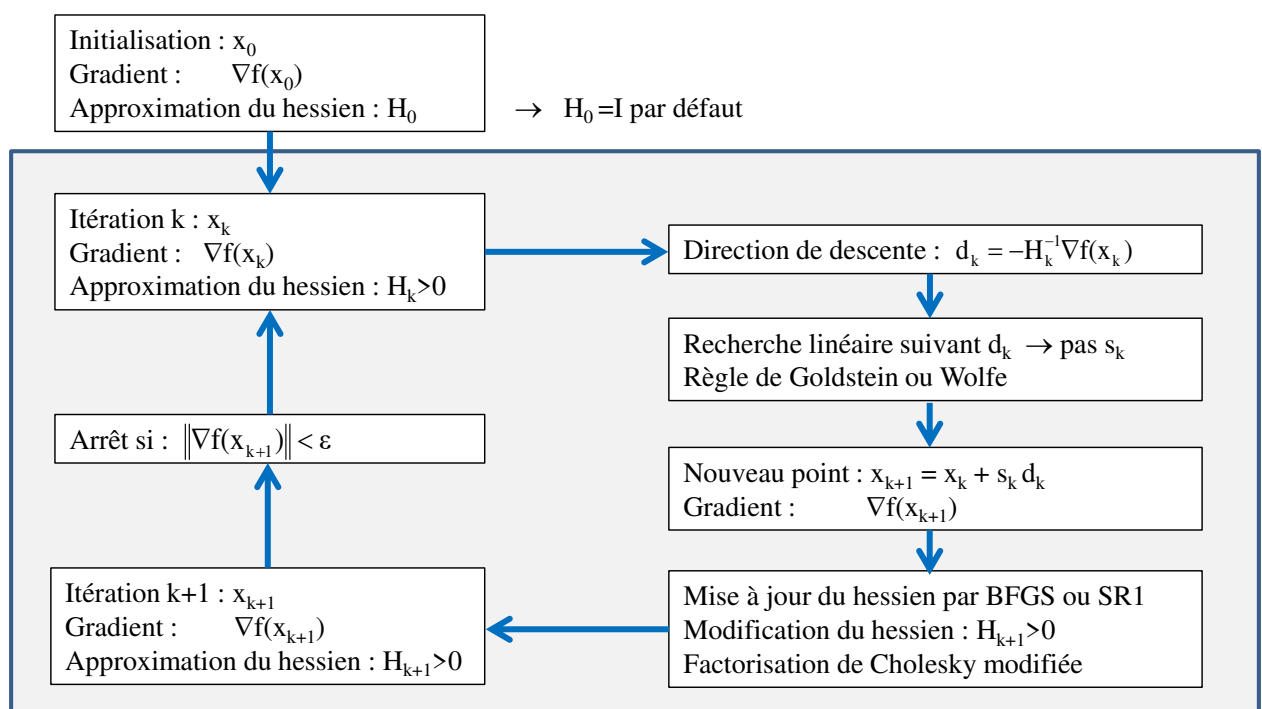
- Si s respecte les 2 conditions \rightarrow pas acceptable
- Si l'intervalle $[s_{\min}, s_{\max}]$ devient inférieur à un seuil donné

Illustration



6.5 Algorithme

Algorithme de recherche linéaire



Principaux résultats de convergence

- Si d est une direction de descente, et f est bornée inférieurement suivant d , alors il existe un pas s suivant d vérifiant les conditions de Wolfe

- Si les directions de descente ne deviennent pas « trop » orthogonales au gradient, l'algorithme de recherche linéaire avec les **conditions de Wolfe** est **globalement convergent**.
 $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0 \rightarrow$ convergence vers un point stationnaire

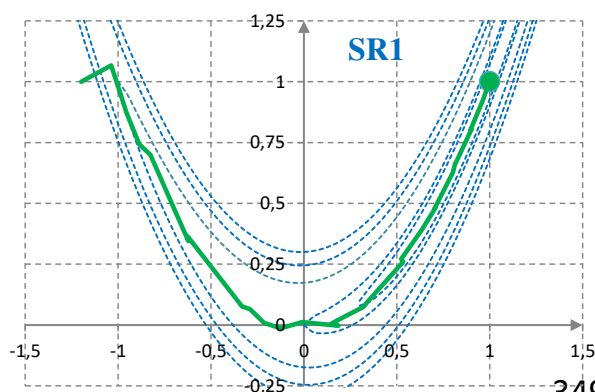
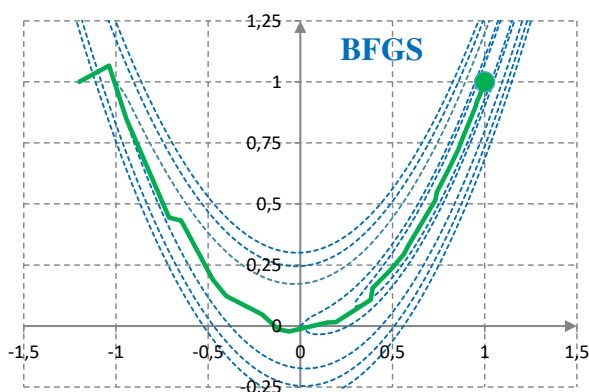
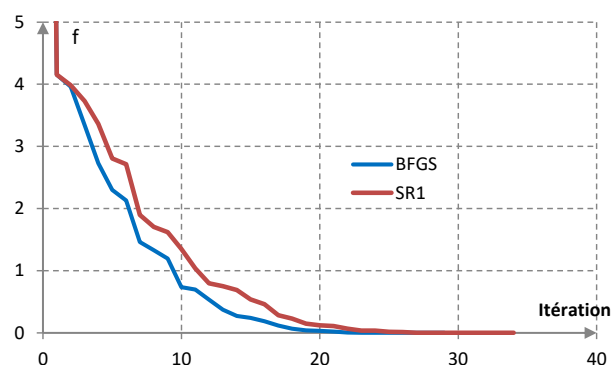
En pratique

- Les directions de descente peuvent être construites avec BFGS ou SR1 (si matrices $H_k > 0$)
- La valeur des coefficients de Goldstein ou Wolfe c_1 et c_2 n'est pas critique pour la convergence.
- Le pas de Newton ($s=1$) est systématiquement testé, car il donne la solution si la fonction est proche de son modèle quadratique.
- La méthode de plus forte pente a la propriété de convergence globale, mais peut être très lente. On peut assurer la convergence globale d'un algorithme de recherche linéaire utilisant d'autres directions **en effectuant périodiquement une itération de plus forte pente**.

Fonction de Rosenbrock

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Point initial : $\begin{pmatrix} -1.2 \\ 1 \end{pmatrix}$
- Recherche linéaire avec BFGS ou SR1



a) Descente non monotone

Condition de décroissance

- La condition d'Armijo impose une décroissance de la fonction **à chaque itération**.

$$f(x_k + sd_k) < \boxed{f(x_k)} + c_1 s \nabla f(x_k)^T d_k$$

Le pas de déplacement s est réduit jusqu'à vérifier la condition de décroissance.

→ **décroissance monotone**

$$f(x_0) > f(x_1) > f(x_2) > \dots > f(x_k) > f(x_{k+1}) > \dots$$

La convergence peut devenir très lente sur une fonction mal conditionnée (« vallée étroite »).

- On peut accélérer la convergence en acceptant une croissance temporaire de la fonction. La condition de décroissance est imposée **entre le point x_{k-m} et le point x_{k+1}** .

$$f(x_k + sd_k) < \boxed{f(x_{k-m})} + c_1 s \nabla f(x_k)^T d_k$$

La fonction peut croître sur m itérations, mais les sous-suites $x_{j+k(m+1)}$ restent décroissantes.

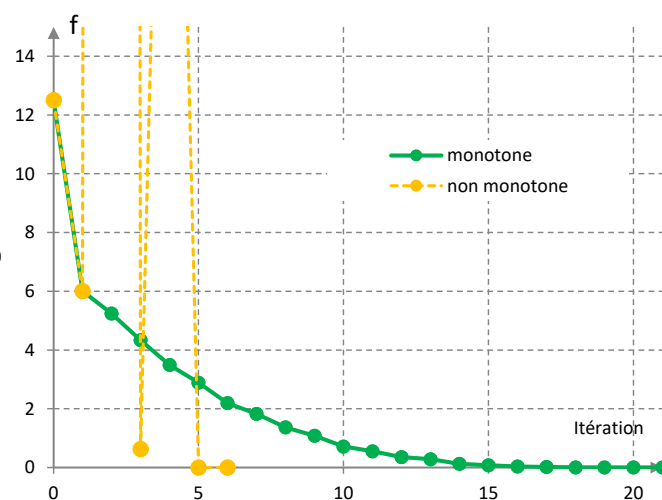
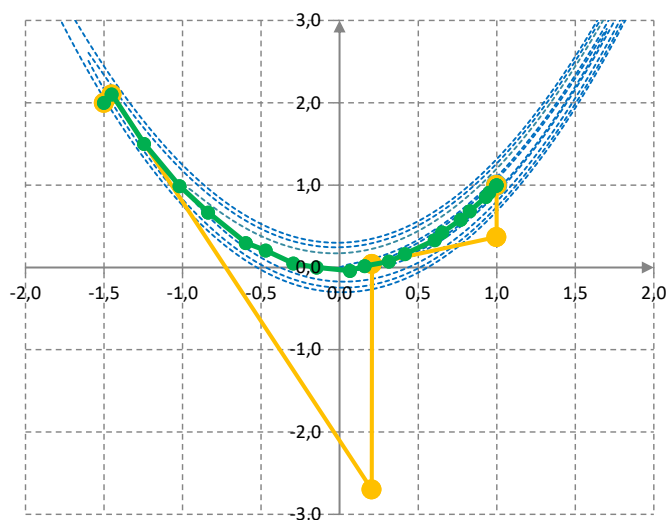
→ **décroissance non monotone** (méthode « watchdog »)

$$f(x_0) > f(x_{m+1}) > f(x_{2(m+1)}) > \dots > f(x_{k(m+1)})$$

$$f(x_1) > f(x_{1+m+1}) > f(x_{1+2(m+1)}) > \dots > f(x_{1+k(m+1)}) \quad \text{sans imposer} \quad f(x_1) < f(x_0)$$

Exemple

Fonction de Rosenbrock à deux variables : $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$



Méthode de Newton avec pas s

- Descente monotone ($s \rightarrow s/2$) : convergence en 21 itérations
- Descente non monotone ($s=1$) : convergence en 6 itérations (2 itérations avec dégradation)

7 Méthode directe

Problème sans contrainte

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

- Les méthodes **sans gradient** (ou **d'ordre 0**) n'utilisent que des évaluations de la fonction.
→ méthodes applicables à toute fonction (non dérivable, bruit numérique)
- Capacité à localiser plusieurs minima locaux (et éventuellement le **minimum global**)
- Convergence généralement lente, peu précise
→ méthodes utilisées pour initialiser un algorithme local plus précis

Principales méthodes

- Méthodes déterministes
 - Nelder-Mead** → méthode locale (polytope convergeant vers un minimum local)
 - Direct** → méthode globale (partition de l'espace de recherche)
- Méthodes stochastiques
 - Métaheuristiques → exploration avec un aspect aléatoire
application aux problèmes combinatoires (variables entières)

7.1 Principe

Principe

$$\min_{x \in \mathbb{R}^n} f(x) \quad \rightarrow \text{problème sans contrainte}$$

- On cherche un **minimum global** de la fonction f sur un domaine D de \mathbb{R}^n .
La fonction est supposée lipschitzienne sur $D \Rightarrow$ bornée inférieurement.
- La méthode Direct consiste à partitionner progressivement le domaine de recherche D .
Direct = « DIviding RECTangles » (1993)
 - Le domaine D est partitionné en sous-domaines D_j .
 - Sur chaque sous-domaine, un minorant de f est estimé sur une condition de type Lipschitz
 - Le sous-domaine de plus bas minorant est « **potentiellement optimal** ».
= sous-domaine ayant le plus de chance de contenir le minimum de f
 - Ce sous-domaine « potentiellement optimal » est sélectionné pour être divisé à son tour.
- La méthode permet de localiser les zones potentielles où se trouve le minimum global.
On peut l'utiliser pour initialiser un algorithme local de descente.
- L'algorithme est développé dans le cas à une variable, puis étendue au cas à n variables.

7.2 Fonction lipschitzienne

Fonction lipschitzienne

$\min_{x \in \mathbb{R}^n} f(x) \rightarrow$ problème sans contrainte

- La fonction f est supposée **lipschitzienne** (ou continue au sens de Lipschitz) sur le domaine D .

$$\exists K > 0, \forall x, y \in D, \|f(y) - f(x)\| \leq K \|y - x\|$$

K est la constante de Lipschitz de la fonction f sur $D \rightarrow$ non connue a priori

- Une fonction f lipschitzienne sur un hyperrectangle D de \mathbb{R}^n est bornée.

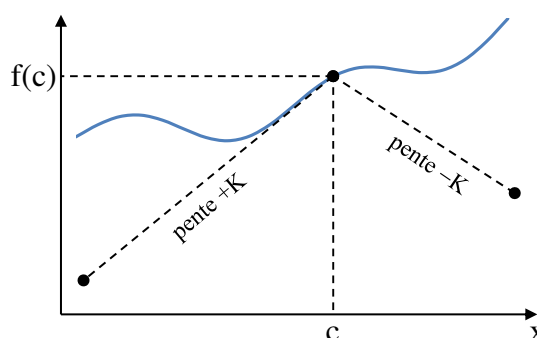
$$\boxed{\forall x \in D, f(c) - Kd \leq f(x) \leq f(c) + Kd}$$

avec c = point quelconque de D

d = majorant de la dimension de D

$$\rightarrow \|x - c\| \leq d$$

- $f(c) - Kd$ est un **minorant** de f sur D .



A savoir, une fonction f est K -Lipschitzienne sur D , si en chaque point x de D , la fonction $f(x)$ a une pente inférieure ou égale à K .

A savoir, la variation de $f(x)$ est bornée.

Exemple

$$f(x) = \sin\left[4\pi\left(x - \frac{1}{2}\right)\right] + 6x^2 + 2 \quad \text{sur } D = [0; 1]$$

$$f'(x) = 4\pi \cos\left[4\pi\left(x - \frac{1}{2}\right)\right] + 12x$$

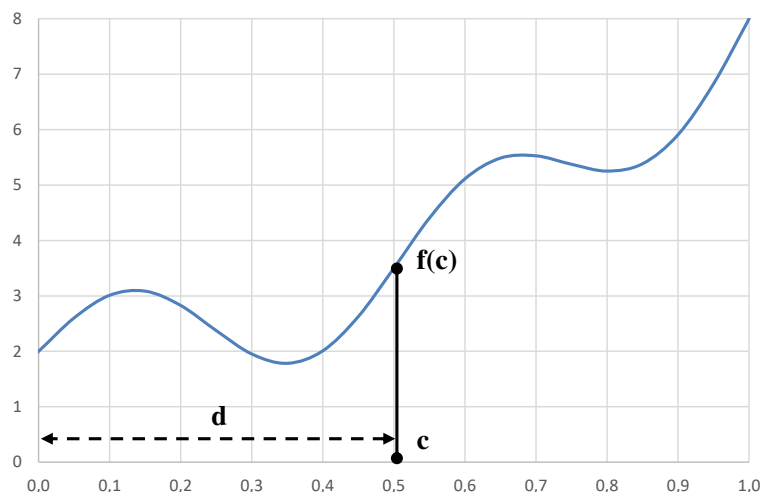
- La dérivée de f est bornée sur $[0; 1]$

$$|f'(x)| \leq 4\pi + 12 \quad \text{sur } D = [0; 1]$$

$\rightarrow f$ est lipschitzienne sur $[0; 1]$

- Si on prend : $K = 25 \geq 4\pi + 12$
 $c = 0,5 \rightarrow f(c) = 3,5$
 $d = 0,5$

$$\rightarrow f(x) \geq 3,5 - 25 \times 0,5 = -9$$



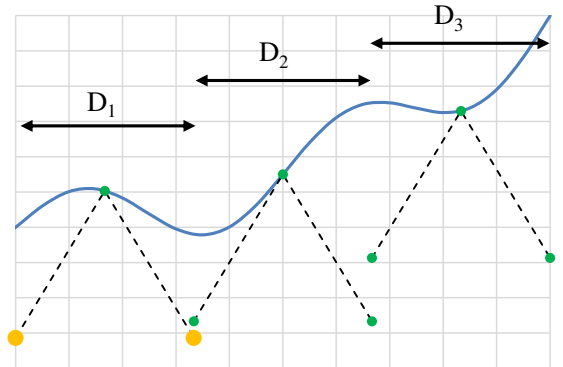
7.3 Division d'intervalle

Division d'intervalles

- On considère une fonction f d'une variable x définie sur un intervalle D .
La fonction est supposée lipschitzienne de constante K (non connue) sur D .
- L'intervalle D est divisé en 3 intervalles égaux D_1, D_2, D_3 .
L'intervalle D_j est de centre c_j
de demi-longueur d_j

- Pour chaque intervalle D_j , on calcule
 - la valeur de f au centre $\rightarrow f_j = f(c_j)$
 - un minorant de f sur $D_j \rightarrow f_{b,j} = f_j - Kd_j$

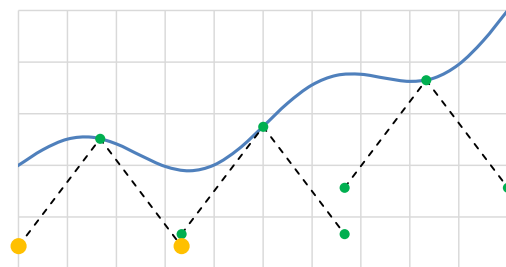
La valeur de la constante K est arbitraire.
(\rightarrow déterminée à partir des points connus)



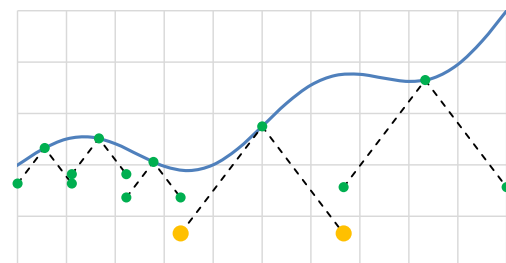
- Le sous-intervalle « **potentiellement optimal** » est celui de plus bas minorant
= intervalle qui a le plus de chance de contenir le minimum de $f \rightarrow D_1$ sur le schéma
Cet intervalle est sélectionné pour être divisé à son tour en 3 intervalles égaux.

Division d'intervalles

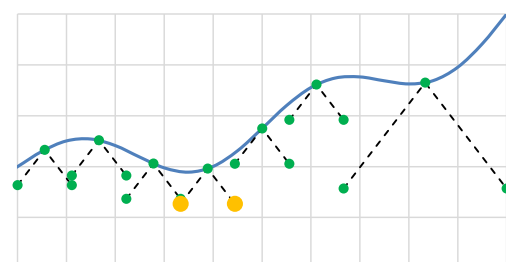
- Première itération : 3 intervalles.
Potentiellement optimal : numéro 1



- Deuxième itération : 5 intervalles.
Potentiellement optimal : numéro 4

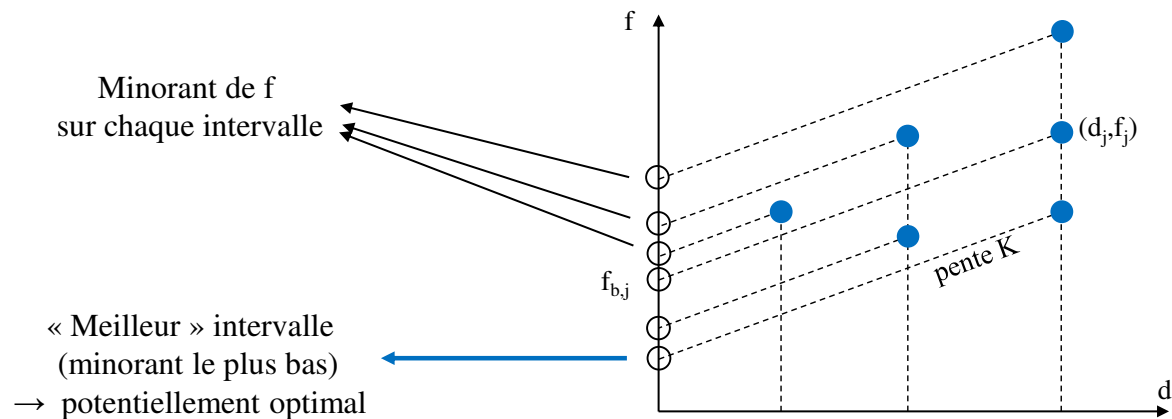


- Troisième itération : 7 intervalles.
Potentiellement optimal : numéro 4



Graphe associé

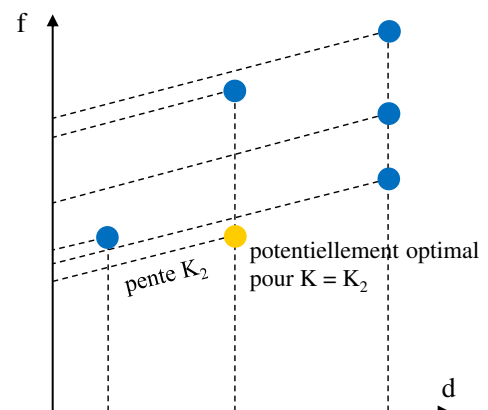
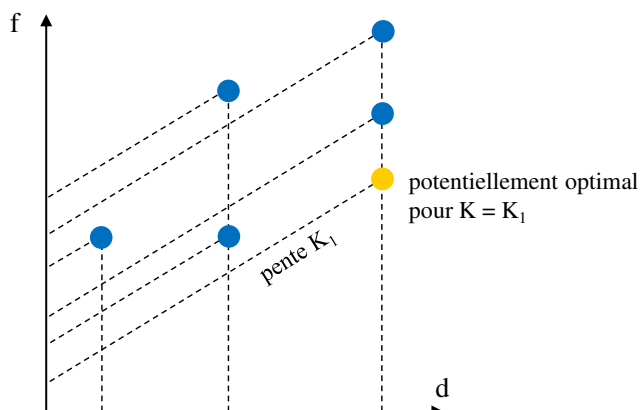
- On représente les intervalles D_j sur un graphe avec : en abscisse la demi-longueur d_j de D_j
en ordonnée la valeur f_j au centre de D_j
L'intervalle D_j est représenté par le point de coordonnées : (d_j, f_j)
La **droite de pente K** passant par ce point a pour équation : $y = K(x - d_j) + f_j$
- Elle coupe l'axe des ordonnées au point de coordonnées : $(0, f_{b,j})$
 $f_{b,j} = f_j - Kd_j$ est un minorant de f sur l'intervalle D_j .

**Intervalle potentiellement optimal**

- Un intervalle est potentiellement optimal s'il existe une constante K positive telle que cet intervalle donne le minorant le plus bas.

$$D_j \text{ potentiellement optimal si } \exists K \geq 0, \forall i, f_{b,j} \leq f_{b,i} \Leftrightarrow f_j - Kd_j \leq f_i - Kd_i$$

- Si l'on fait passer une droite de pente K par chaque point représentatif d'un intervalle, la droite associée à l'intervalle D_j à l'ordonnée à l'origine la plus petite.



Intervalle potentiellement optimal

- L'intervalle $D_j (d_j, f_j)$ est potentiellement optimal si $\exists K \geq 0, \forall i, f_j - Kd_j \leq f_i - Kd_i$
 $\Leftrightarrow \exists K \geq 0, \forall i, f_j - f_i \leq K(d_j - d_i)$

La condition s'exprime différemment selon le signe de $(d_i - d_j)$.

- Pour les intervalles D_i tels que $d_i = d_j$

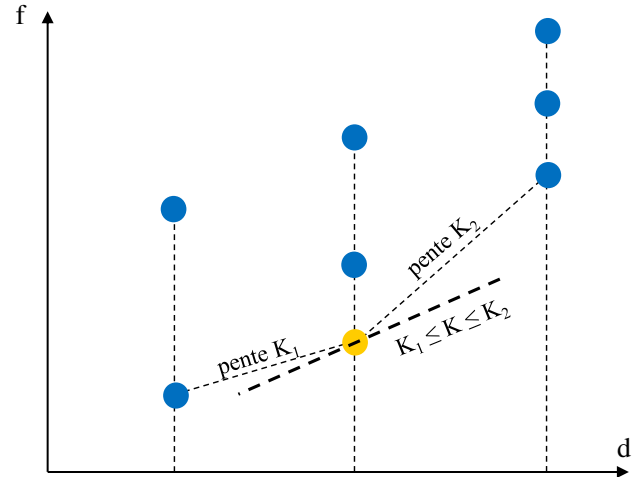
$$f_j \leq f_i$$

- Pour les intervalles D_i tels que $d_i < d_j$

$$K \geq \frac{f_j - f_i}{d_j - d_i} \Rightarrow K \geq K_1 = \max_{i/d_i < d_j} \frac{f_j - f_i}{d_j - d_i}$$

- Pour les intervalles D_i tels que $d_i > d_j$

$$K \leq \frac{f_j - f_i}{d_j - d_i} \Rightarrow K \leq K_2 = \min_{i/d_i > d_j} \frac{f_j - f_i}{d_j - d_i}$$

**Intervalle potentiellement optimal**

- On ajoute une condition de **décroissance suffisante**.

$$D_j \text{ potentiellement optimal si } \exists K \geq 0 \begin{cases} f_j - Kd_j \leq f_i - Kd_i, \forall i \\ f_j - Kd_j \leq f_{\min} - \varepsilon |f_{\min}| \end{cases} \text{ avec } f_{\min} = \min_i f_i$$

f_{\min} est le minimum parmi les évaluations disponibles f_i .

- La première condition exprime que $D_j (d_j, f_j)$ est un point de l'enveloppe convexe inférieure dans le graphe (d, f) . Cette condition est satisfaite si :
 - D_j est le meilleur intervalle ($f_j \leq f_i$) de longueur d_j
 - $\max_{i/d_i < d_j} K_i = K_1 \leq K_2 = \min_{i/d_i > d_j} K_i$ avec $K_i = \frac{f_j - f_i}{d_j - d_i} \rightarrow$ existence de K compris entre K_1 et K_2 .
- La deuxième condition exprime que l'intervalle D_j ne doit pas être trop petit.

$$f_j - Kd_j \leq f_{\min} - \varepsilon |f_{\min}| \Rightarrow d_j \geq \frac{f_{\min} - \varepsilon |f_{\min}| - f_j}{K} \text{ avec } \varepsilon \approx 10^{-3}$$

\rightarrow évite que la recherche devienne trop locale.

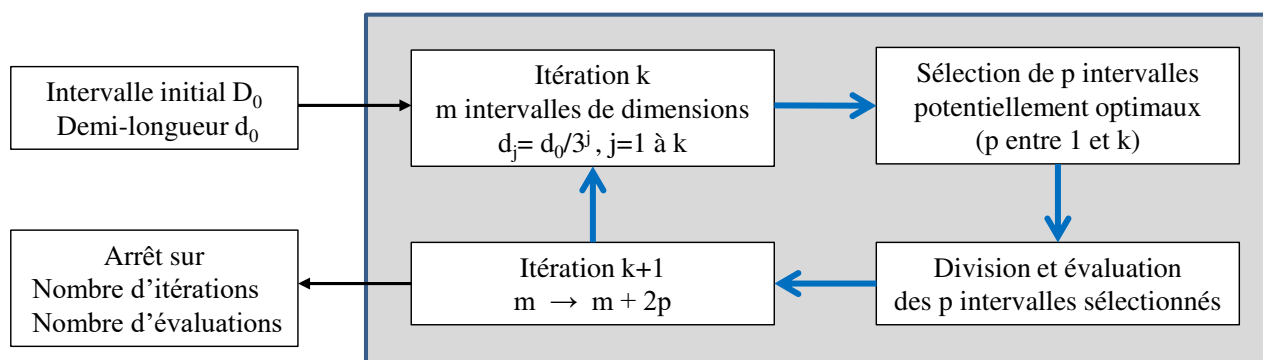
7.4 Algorithme à une variable

Algorithme à une variable

- Une itération consiste à :
 - sélectionner p intervalles parmi ceux potentiellement optimaux
 - les diviser en 3 intervalles égaux.

Le nombre d'intervalles augmente de $2p$ à chaque itération ($p=2$ pour l'exemple).

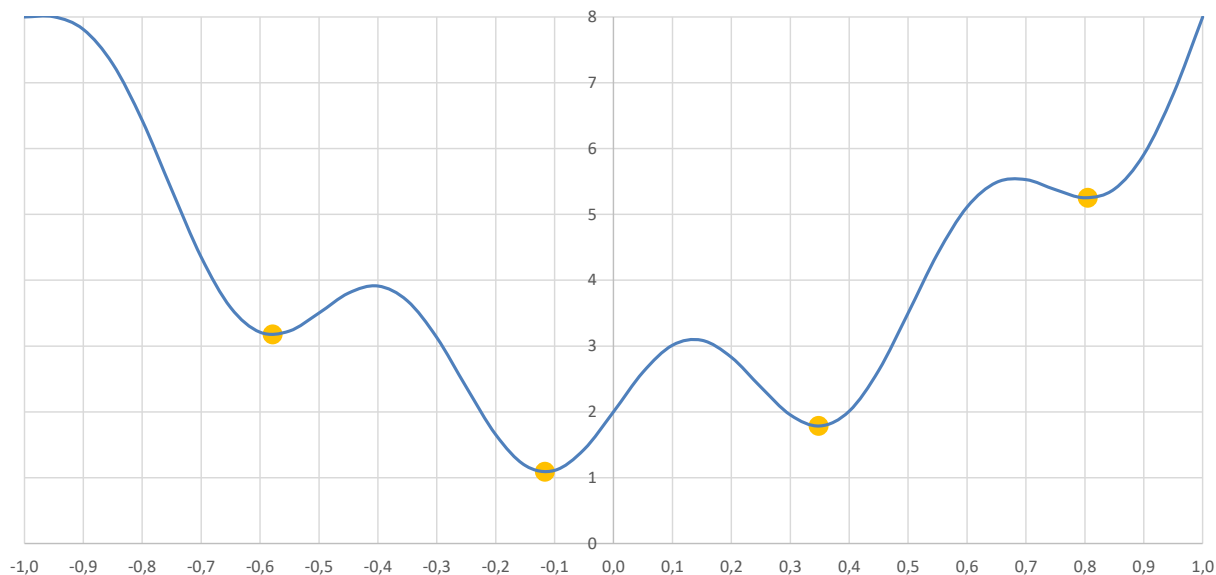
- L'intervalle initial est de demi-longueur d_0 .
A l'itération k , la partition est composée d'intervalles de demi-longueur $d_j = d_0/3^j$, $1 \leq j \leq k$.
Il y a au plus k intervalles potentiellement optimaux (un par dimension d_j).



Exemple à une variable

$$f(x) = \sin\left[4\pi\left(x - \frac{1}{2}\right)\right] + 6x^2 + 2 \quad \text{sur } D = [-1; 1]$$

Minimum	local	global	local	local
x	-0,57843	-0,11615	0,34804	0,80520
f(x)	3,17389	1,08712	1,78363	5,25072



Notations sur les transparents suivants :

- $[x_{\inf}, x_{\sup}]$ est l'intervalle utilisé ; et sa longueur est $x_{\sup} - x_{\inf}$.
- $x_{\text{mid}} = (x_{\inf} + x_{\sup})/2$ est son centre, et $f_{\text{mid}} = f(x_{\text{mid}})$.
- $f_b = f_{\text{mid}} - K \cdot \text{longueur}$ est l'intersection avec l'axe depuis $(x_{\text{mid}}, f_{\text{mid}})$ pour une droite de pente K (à savoir le point le plus bas dans cet intervalle pour ce K).

Itération 1 : 3 intervalles

x inf	x sup	x mid	longueur	fmid	fb
-1,000	-0,333	-0,667	0,667	3,801	3,801
-0,333	0,333	0,000	0,667	2,000	2,000
0,333	1,000	0,667	0,667	5,533	5,533

K
0

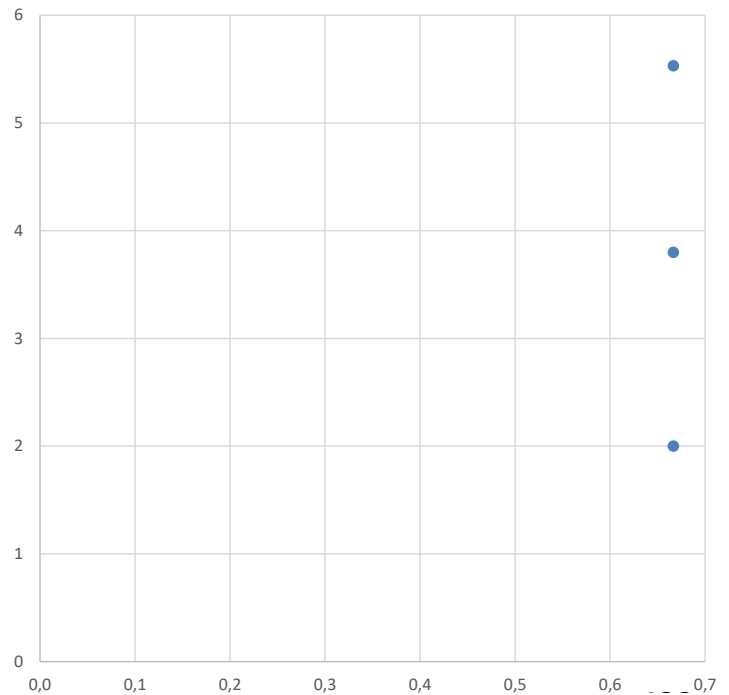
L'intervalle $[-1;1]$ est divisé en 3.

Les 3 intervalles ont même longueur.

Le meilleur intervalle est sélectionné pour être divisé à l'itération suivante.



Passage de 3 à 5 intervalles

**Notes :**

- Au début de l'algorithme, on part avec K initialisé à 0, et $f_b = f_{\text{mid}}$.
- Les lignes sont regroupés en paquets de même longueur ℓ_i .
- Pour chaque paquet, on subdivise l'intervalle pour lequel f_b est minimum.
- Les intervalles subdivisés sont toujours insérés à la fin des blocs.

Itération 2 : 5 intervalles

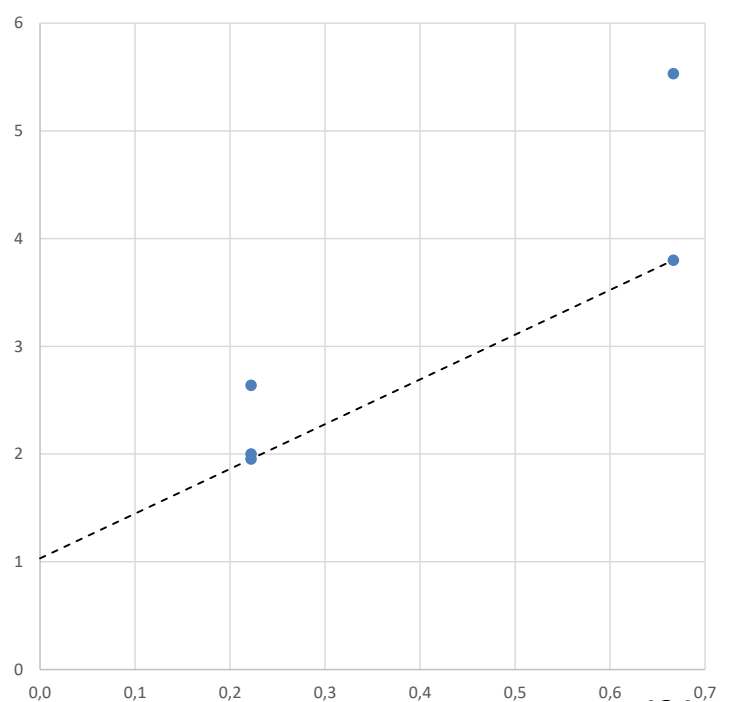
x inf	x sup	x mid	longueur	fmid	fb	K
-1,000	-0,333	-0,667	0,667	3,801	1,031	4,155
0,333	1,000	0,667	0,667	5,533	2,763	
-0,333	-0,111	-0,222	0,222	1,954	1,031	
-0,111	0,111	0,000	0,222	2,000	1,077	
0,111	0,333	0,222	0,222	2,638	1,715	

Les intervalles sont rangés par longueur décroissante.

En gris : nouveaux intervalles issus de l'itération précédente

En jaune : meilleur intervalle parmi ceux de même longueur

En vert : 2 intervalles sélectionnés donnant le minorant le plus bas
→ potentiellement optimaux

**Notes :**

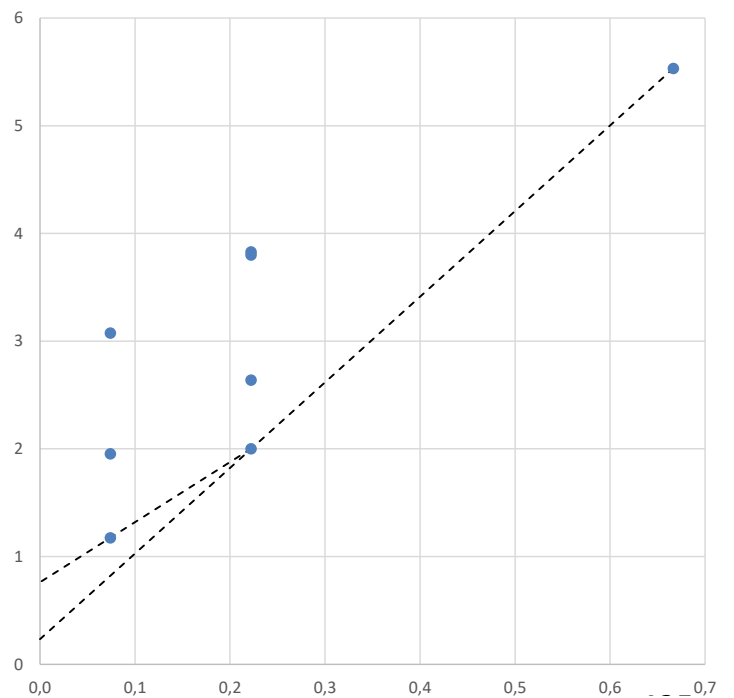
- on note f_{mid}^i le f_{mid} minimal sur le $i^{\text{ème}}$ paquet.
- le K_i du $i^{\text{ème}}$ paquet est calculé comme $K_i = \min_{j>i} (f_{\text{mid}}^i - f_{\text{mid}}^j) / (\ell_i - \ell_j)$.

- pour le dernier paquet, on utilise le K_i du paquet précédent.
- f_b est calculé comme $f_b = f_{\text{mid}} - K \cdot \ell_i$, avec le K approprié.

Dans l'exemple ci-dessus, tous les f_b sont calculés avec le même K (c'est le seul disponible à cette itération).

Itération 3 : 9 intervalles

x inf	x sup	x mid	longueur	fmid	fb	K
0,333	1,000	0,667	0,667	5,533	0,234	7,949
-0,111	0,111	0,000	0,222	2,000	0,234	5,578
0,111	0,333	0,222	0,222	2,638	1,399	
-1,000	-0,778	-0,889	0,222	7,726	6,486	
-0,778	-0,556	-0,667	0,222	3,801	2,561	
-0,556	-0,333	-0,444	0,222	3,828	2,589	
-0,333	-0,259	-0,296	0,074	3,076	2,663	
-0,259	-0,185	-0,222	0,074	1,954	1,541	
-0,185	-0,111	-0,148	0,074	1,174	0,761	



Pour le calcul de K :

- K_i est associé au minimum f_b^i (le minimum de f_b sur le $i^{\text{ème}}$ paquet).
- sur les lignes qui précèdent (resp. suivent) f_b^i , on utilise K_{i-1} (resp. K_i).
- en f_b^i , on utilise K_{i-1} si $K_{i-1} > K_i$, et K_i sinon.

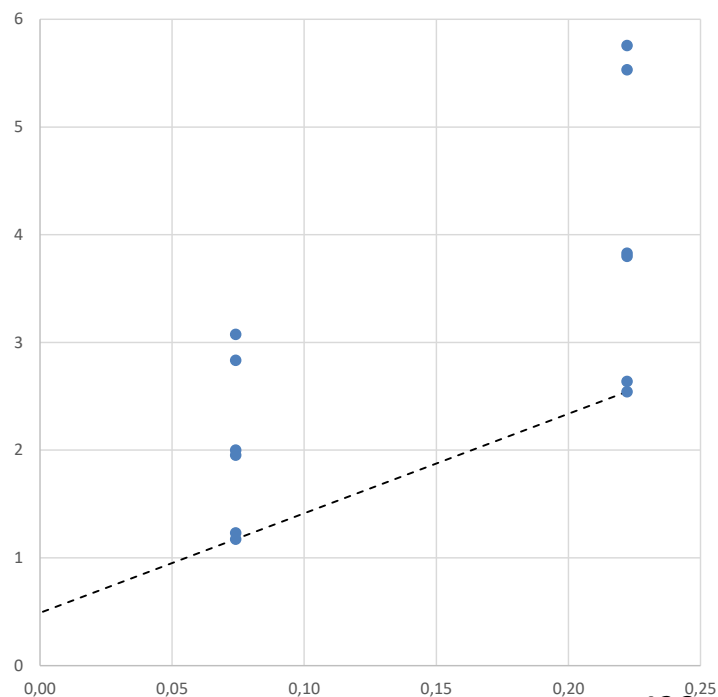
Ici, $K_1 = 7.949$ est utilisée pour les deux premières lignes (puisque $K_1 > K_2$), et K_2 pour le reste des lignes.

Itération 4 : 13 intervalles

x inf	x sup	x mid	longueur	fmid	fb
0,111	0,333	0,222	0,222	2,638	0,585
-1,000	-0,778	-0,889	0,222	7,726	5,672
-0,778	-0,556	-0,667	0,222	3,801	1,748
-0,556	-0,333	-0,444	0,222	3,828	1,775
0,333	0,556	0,444	0,222	2,542	0,489
0,556	0,778	0,667	0,222	5,533	3,480
0,778	1,000	0,889	0,222	5,756	3,703
-0,333	-0,259	-0,296	0,074	3,076	2,392
-0,259	-0,185	-0,222	0,074	1,954	1,270
-0,185	-0,111	-0,148	0,074	1,174	0,489
-0,111	-0,037	-0,074	0,074	1,231	0,546
-0,037	0,037	0,000	0,074	2,000	1,316
0,037	0,111	0,074	0,074	2,835	2,151

K

9,239



Dans l'itération ci-dessus :

- Une seule valeur de K , donc la même est utilisée pour tous les blocs.
- On voit ici que les intervalles subdivisés sont bien insérés en fin de bloc.

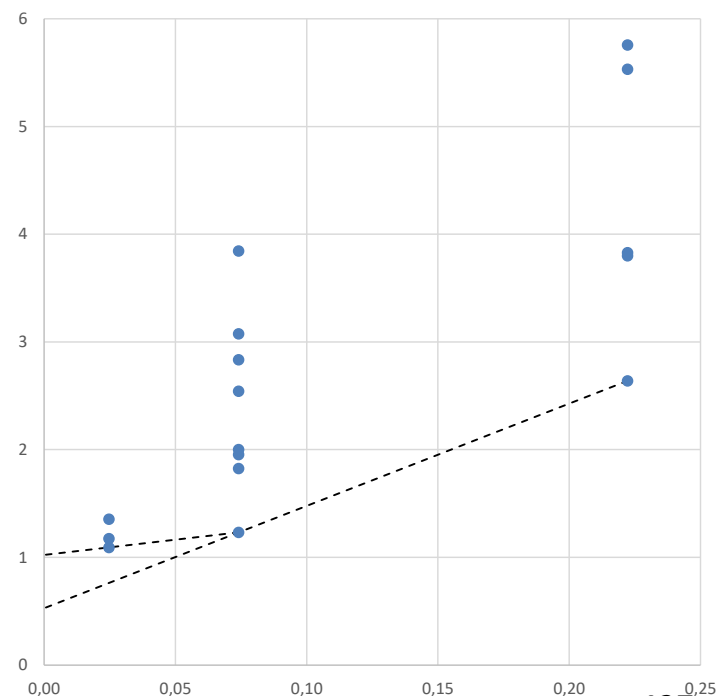
Itération 5 : 17 intervalles

x inf	x sup	x mid	longueur	fmid	fb
0,111	0,333	0,222	0,222	2,638	0,527
-1,000	-0,778	-0,889	0,222	7,726	5,614
-0,778	-0,556	-0,667	0,222	3,801	1,689
-0,556	-0,333	-0,444	0,222	3,828	1,717
0,556	0,778	0,667	0,222	5,533	3,421
0,778	1,000	0,889	0,222	5,756	3,645
-0,333	-0,259	-0,296	0,074	3,076	2,372
-0,259	-0,185	-0,222	0,074	1,954	1,251
-0,111	-0,037	-0,074	0,074	1,231	0,527
-0,037	0,037	0,000	0,074	2,000	1,791
0,037	0,111	0,074	0,074	2,835	2,626
0,333	0,407	0,370	0,074	1,825	1,616
0,407	0,481	0,444	0,074	2,542	2,334
0,481	0,556	0,519	0,074	3,844	3,635
-0,185	-0,160	-0,173	0,025	1,355	1,285
-0,160	-0,136	-0,148	0,025	1,174	1,104
-0,136	-0,111	-0,123	0,025	1,092	1,022

K

9,501

2,818

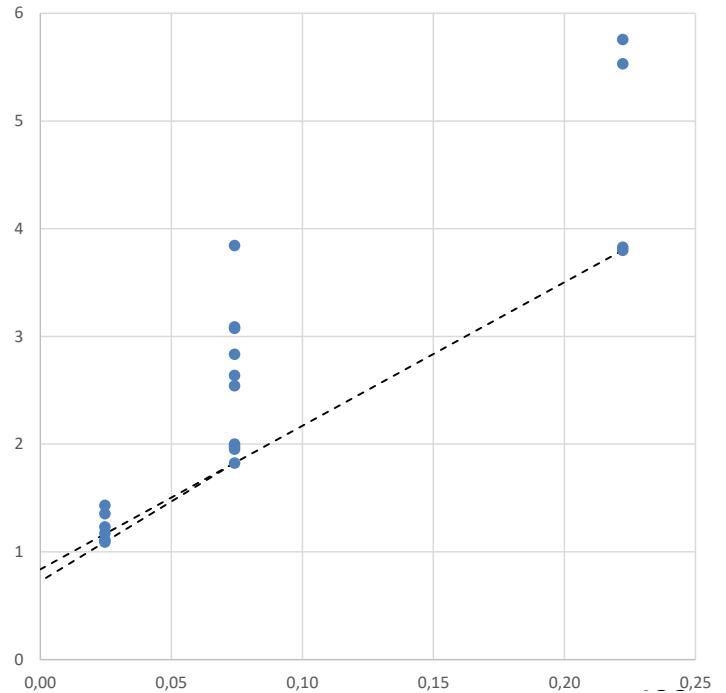


Dans l'itération ci-dessus :

- $K_1 = 9.501$ est utilisé pour le calcul de f_b dans le bloc 1 et pour les trois premières valeurs du bloc 2 (car $K_1 > K_2$)
- $K_2 = 2.818$ est utilisé pour le reste des valeurs du bloc 2, et pour le bloc 3.

Itération 6 : 21 intervalles

x inf	x sup	x mid	longueur	fmid	fb	K
-1,000	-0,778	-0,889	0,222	7,726	4,762	13,338
-0,778	-0,556	-0,667	0,222	3,801	0,837	
-0,556	-0,333	-0,444	0,222	3,828	0,864	
0,556	0,778	0,667	0,222	5,533	2,569	
0,778	1,000	0,889	0,222	5,756	2,792	14,846
-0,333	-0,259	-0,296	0,074	3,076	2,088	
-0,259	-0,185	-0,222	0,074	1,954	0,966	
-0,037	0,037	0,000	0,074	2,000	1,012	
0,037	0,111	0,074	0,074	2,835	1,847	
0,333	0,407	0,370	0,074	1,825	0,725	
0,407	0,481	0,444	0,074	2,542	1,443	
0,481	0,556	0,519	0,074	3,844	2,744	
0,111	0,185	0,148	0,074	3,090	1,990	
0,185	0,259	0,222	0,074	2,638	1,539	
0,259	0,333	0,296	0,074	1,977	0,878	
-0,185	-0,160	-0,173	0,025	1,355	0,988	
-0,160	-0,136	-0,148	0,025	1,174	0,807	
-0,136	-0,111	-0,123	0,025	1,092	0,725	
-0,111	-0,086	-0,099	0,025	1,112	0,746	
-0,086	-0,062	-0,074	0,025	1,231	0,864	
-0,062	-0,037	-0,049	0,025	1,433	1,067	

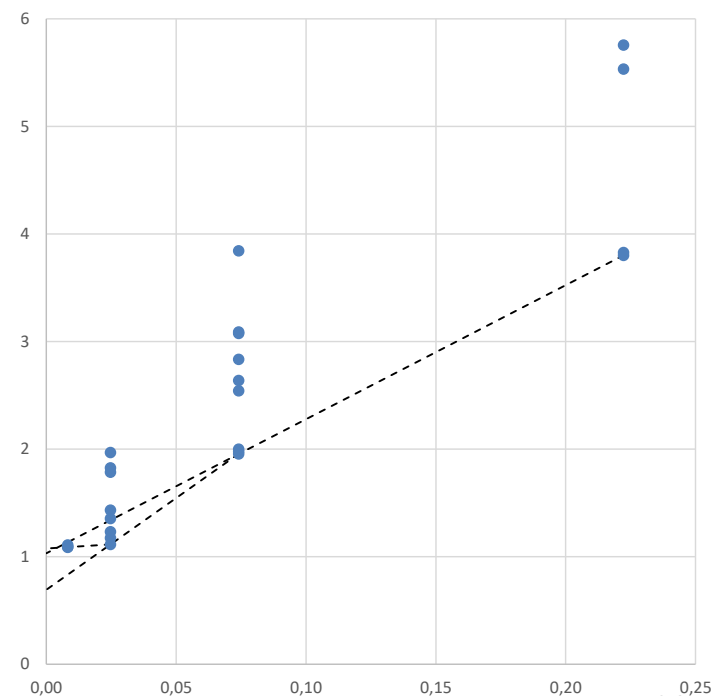


Dans l'itération ci-dessus :

- $K_1 = 13.338$ est utilisé pour le calcul de f_b dans le bloc 1 et pour les quatre premières valeurs du bloc 2 (car $K_1 < K_2$).
- $K_2 = 14.486$ est utilisé pour le reste des valeurs du bloc 2, et pour le bloc 3.

Itération 7 : 25 intervalles

x inf	x sup	x mid	longueur	fmid	fb	K
-1,000	-0,778	-0,889	0,222	7,726	4,956	12,463
-0,778	-0,556	-0,667	0,222	3,801	1,031	
-0,556	-0,333	-0,444	0,222	3,828	1,058	
0,556	0,778	0,667	0,222	5,533	2,763	
0,778	1,000	0,889	0,222	5,756	2,986	17,049
-0,333	-0,259	-0,296	0,074	3,076	2,153	
-0,259	-0,185	-0,222	0,074	1,954	0,691	
-0,037	0,037	0,000	0,074	2,000	0,737	
0,037	0,111	0,074	0,074	2,835	1,572	
0,407	0,481	0,444	0,074	2,542	1,280	
0,481	0,556	0,519	0,074	3,844	2,581	
0,111	0,185	0,148	0,074	3,090	1,827	
0,185	0,259	0,222	0,074	2,638	1,375	
0,259	0,333	0,296	0,074	1,977	0,714	
-0,185	-0,160	-0,173	0,025	1,355	0,934	
-0,160	-0,136	-0,148	0,025	1,174	0,753	
-0,111	-0,086	-0,099	0,025	1,112	0,691	1,530
-0,086	-0,062	-0,074	0,025	1,231	1,193	
-0,062	-0,037	-0,049	0,025	1,433	1,395	
0,333	0,358	0,346	0,025	1,784	1,746	
0,358	0,383	0,370	0,025	1,825	1,787	1,530
0,383	0,407	0,395	0,025	1,968	1,930	
-0,136	-0,128	-0,132	0,008	1,108	1,095	
-0,128	-0,119	-0,123	0,008	1,092	1,079	1,530
-0,119	-0,111	-0,115	0,008	1,087	1,075	



Dans l'itération ci-dessus :

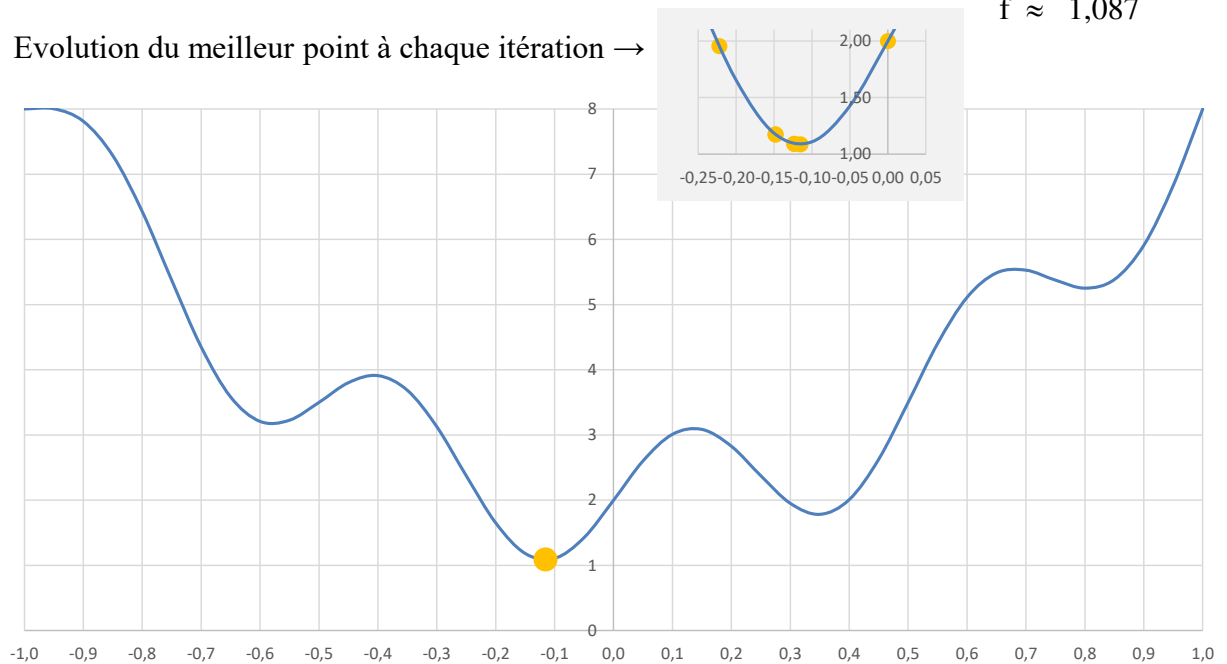
- $K_1 = 12.463$ est utilisé pour le calcul de f_b dans le bloc 1 et pour la première valeur du bloc 2 (car $K_1 < K_2$).

- $K_2 = 17.049$ est utilisé pour le reste des valeurs du bloc 2, et pour les trois premières valeurs du bloc 3 (car $K_2 > K_3$).
- $K_3 = 1.530$ est utilisé pour le reste des valeurs du bloc 2, et pour le bloc 3.

Exemple à une variable

On obtient après **25 évaluations** une approximation du minimum global : $x \approx -0,115$
 $f \approx 1,087$

Evolution du meilleur point à chaque itération →



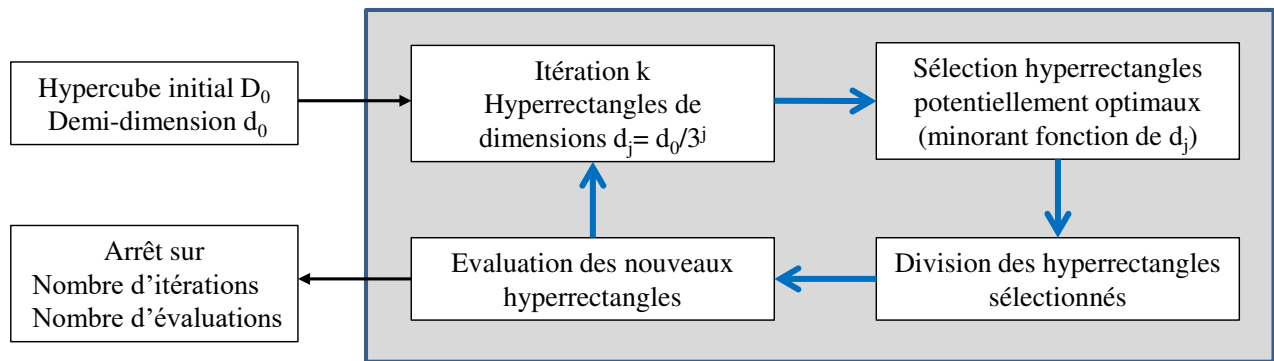
EXERCICE 23: Minimisation par la méthode directe

Soit la fonction $f(x) = 4\sin(6x) + 6x^2$ que l'on veut minimiser sur l'intervalle $[-2; +2]$.
 Effectuer les trois premières itérations de l'algorithme de minimisation directe.

7.5 Algorithme à n variables

Algorithme à n variables

- On peut appliquer la méthode de partitionnement de façon similaire dans le cas à n variables. Les intervalles sont remplacés par des hyperrectangles dans \mathbb{R}^n .
- La méthode de division est définie dans le cas d'un hypercube, puis adaptée au cas d'un hyperrectangle.
- La sélection d'un hyperrectangle potentiellement optimal est basée sur l'évaluation d'un minorant dépendant de la dimension de l'hyperrectangle.



Division d'un hypercube

- L'hypercube a pour centre c_0 et pour côté δ .

- On évalue 2 points à distance $\pm\delta/3$ de c_0 suivant chaque direction e_i pour $i=1$ à n .

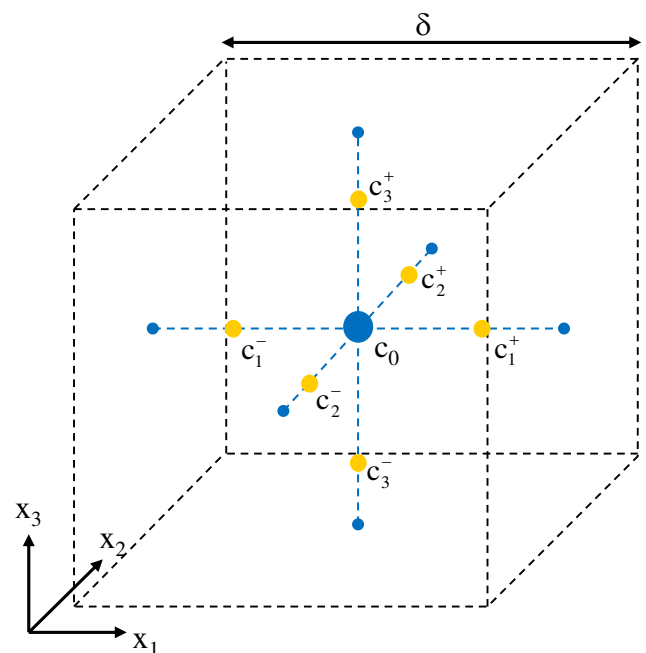
$$\begin{cases} c_i^- = c_0 - \frac{1}{3}\delta e_i \rightarrow \varphi_i^- = f(c_i^-) \\ c_i^+ = c_0 + \frac{1}{3}\delta e_i \rightarrow \varphi_i^+ = f(c_i^+) \end{cases} \rightarrow 2n \text{ points}$$

- On retient la valeur minimale φ_i suivant chaque direction e_i .

$$\varphi_i = \min(\varphi_i^-, \varphi_i^+), \quad i = 1 \text{ à } n$$

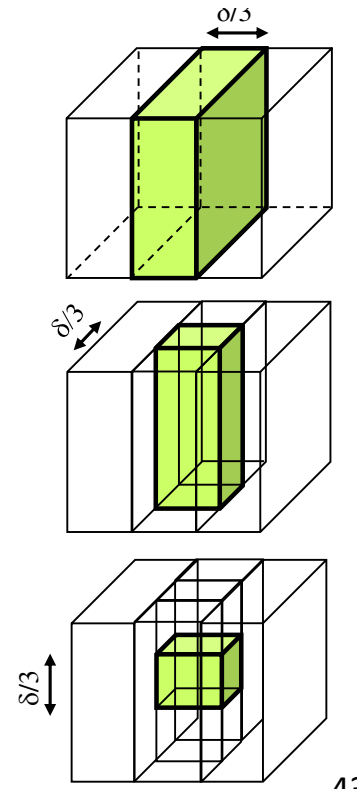
- Les φ_i sont rangés dans l'ordre croissant.
 $\varphi_{i1} \leq \varphi_{i2} \leq \dots \leq \varphi_{in}$

→ détermine les meilleures directions
et **l'ordre de division** des côtés de l'hypercube

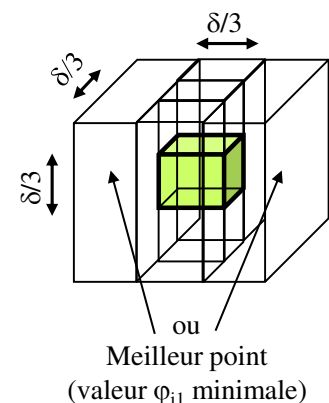


Division d'un hypercube

- L'hypercube initial a pour centre c_0 et pour côté δ .
- On divise l'hypercube en 3 suivant la **meilleure direction** (i_1).
→ 3 hyperrectangles : centres c_{i1}^- , c_0 et c_{i1}^+
avec 1 côté de longueur $\delta/3$
et $n-1$ côtés de longueur δ
- On sélectionne l'**hyperrectangle central** (contenant c_0)
et on le divise en 3 suivant la 2^{ème} meilleure direction (i_2).
→ 3 hyperrectangles : centres c_{i2}^- , c_0 et c_{i2}^+
avec 2 côtés de longueur $\delta/3$
et $n-2$ côtés de longueur δ
- On poursuit en divisant à chaque étape l'hyperrectangle central
suivant la direction suivante : i_3, \dots, i_n .
- La dernière division donne un hypercube de centre c_0
et de côté $\delta/3$.

**Division d'un hypercube**

- Chaque division génère 2 hyperrectangles supplémentaires.
Les hyperrectangles ont des côtés de **longueur $\delta/3$ ou δ** .
- On obtient après les divisions suivant les n directions :
 - $2n$ hyperrectangles de centres c_i^- , c_i^+ pour $i=1$ à n .
 - 1 hypercube central de centre c_0 et de côté $\delta/3$
- Le meilleur point (c_{i1}^- ou c_{i1}^+) est au centre de l'un
des 2 hyperrectangles les **plus grands** (première division).

**Division d'un hyperrectangle**

- Les côtés d'un hyperrectangle sont soit de longueur $\delta/3$ → p côtés
soit de longueur δ → $n-p$ côtés
- On applique la même procédure de division que pour un hypercube
mais **en se restreignant aux p plus grands côtés** de longueur δ .
La division génère $2p$ hyperrectangles supplémentaires avec des côtés de **longueur $\delta/3$ ou δ** .

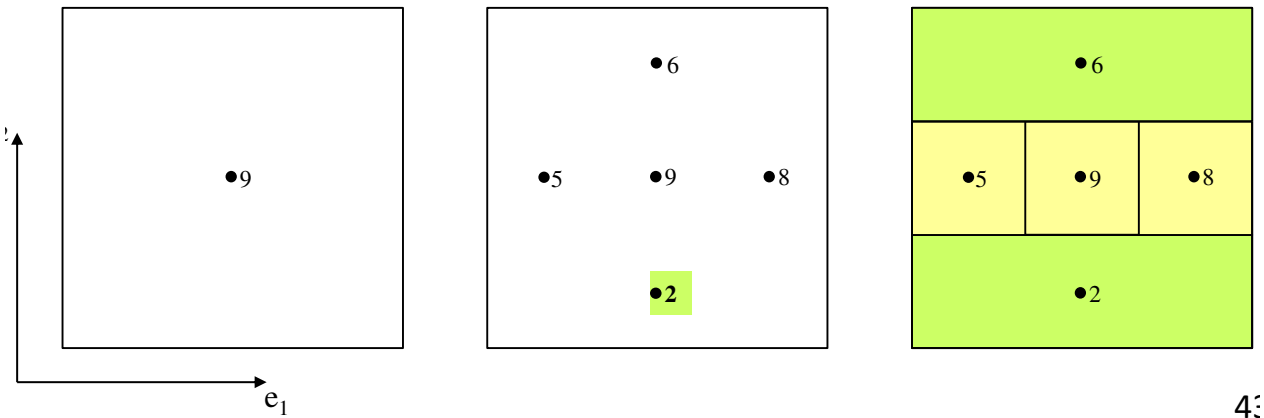
Division des hyperrectangles

- Première itération

Hypercube initial
Evaluation au centre = 9

→ Evaluations suivant $\pm e_1, e_2$
Meilleure évaluation = 2
→ suivant e_2

Division suivant e_2
puis suivant e_1
→ 5 hyperrectangles



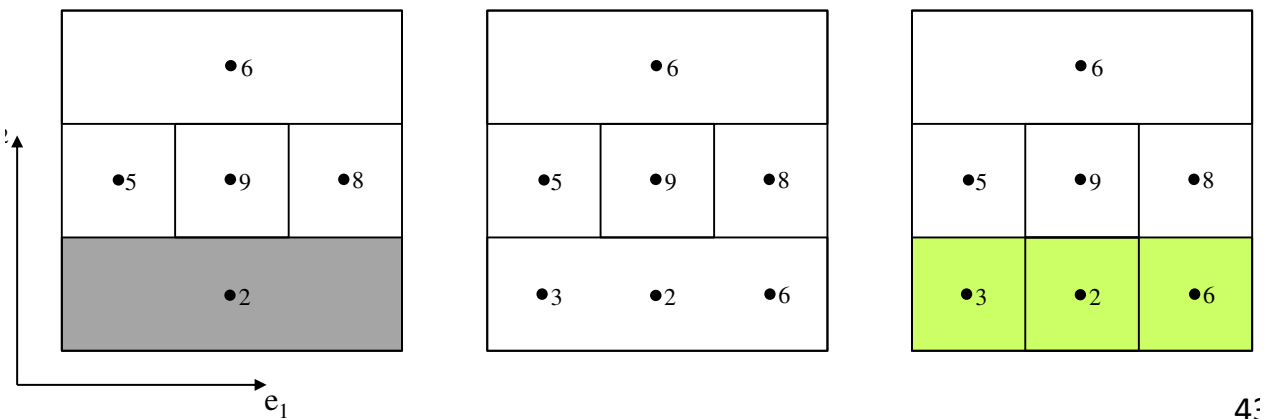
Division des hyperrectangles

- Deuxième itération

1 hyperrectangle sélectionné
(potentiellement optimal)

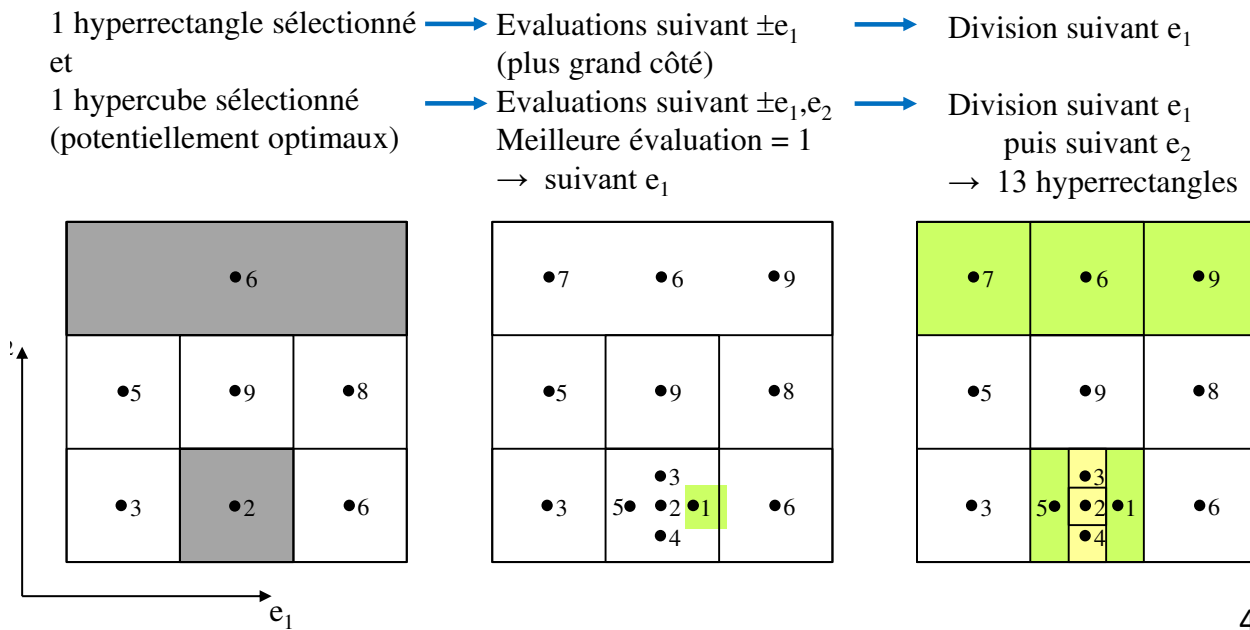
→ Evaluations suivant $\pm e_1$
(plus grand côté)

Division suivant e_1
→ 7 hyperrectangles



Division des hyperrectangles

- Troisième itération



Dimension des hyperrectangles

- On suppose que le domaine de recherche initial D_0 est un **hypercube** de côté δ_0 .
On peut toujours se ramener à cette situation par une mise à l'échelle sur les variables.

$$a_i \leq x_i \leq b_i \Leftrightarrow 0 \leq x_i' \leq \delta_0 \quad \text{avec} \quad x_i' = \delta_0 \frac{x_i - a_i}{b_i - a_i}$$

- Les divisions génèrent des hyperrectangles D_j dont les côtés ont **2 longueurs différentes**.
 - p côtés sont de longueur $\delta_i = \delta_0 / 3^{k+1}$, $i = 1$ à p
 - $n-p$ côtés sont de longueur $\delta_i = \delta_0 / 3^k$, $i = p+1$ à n

- La **dimension** d'un hyperrectangle D_j est mesurée par sa norme 2 ou sa norme ∞ .

$$\begin{aligned} \text{- norme 2 : } \|D_j\|_2 &= \sqrt{\sum_{i=1}^n (\delta_i)^2} = \sqrt{\sum_{i=1}^p (\delta_0 / 3^{k+1})^2 + \sum_{i=p+1}^n (\delta_0 / 3^k)^2} = \sqrt{p(\delta_0 / 3^{k+1})^2 + (n-p)(\delta_0 / 3^k)^2} \\ &\Rightarrow \|D_j\|_2 = \delta_0 / 3^k \sqrt{\frac{p}{9} + (n-p)} = \delta_0 / 3^k \sqrt{n - \frac{8}{9}p} \end{aligned}$$

$$\text{- norme } \infty : \|D_j\|_\infty = \max_{i=1 \text{ à } n} \delta_i = \delta_0 / 3^k \rightarrow \text{longueur du plus grand côté}$$

Hyperrectangle potentiellement optimal

- A l'itération k , le domaine de recherche D est partitionné en hyperrectangles D_j .
La procédure pour choisir l'hyperrectangle à diviser est identique au cas à une variable.
- Pour chaque hyperrectangle D_j , on calcule un **minorant** $f_{b,j}$ de f sur D_j .
L'hyperrectangle « **potentiellement optimal** » est celui de plus bas minorant
= hyperrectangle qui a le plus de chance de contenir le minimum de f
- Le minorant pour l'hyperrectangle D_j est calculé par : $f_{b,j} = f_j - Kd_j$
 - f_j est la valeur au centre de l'hyperrectangle $\rightarrow f_j = f(c_j)$ valeur déjà calculée
 - d_j est la demi-dimension de l'hyperrectangle $\rightarrow d_j = \frac{1}{2} \|D_j\|$ norme 2 ou ∞
- D_j est potentiellement optimal si $\exists K \geq 0 \begin{cases} f_j - Kd_j \leq f_i - Kd_i, \forall i \\ f_j - Kd_j \leq f_{\min} - \varepsilon |f_{\min}| \end{cases}$ avec $f_{\min} = \min_i f_i$

On peut représenter l'ensemble des hyperrectangles sur le graphe associé (d_j, f_j) .
La constante K est calculée de la même façon que dans le cas à une variable.

Exemple à deux variables

$$f(x, y) = (x - 0.2)^2 + 2(y - 0.1)^2 - 0.3 \cos[3\pi(x - 0.2)] - 0.4 \cos[4\pi(y - 0.1)] + 0.7$$

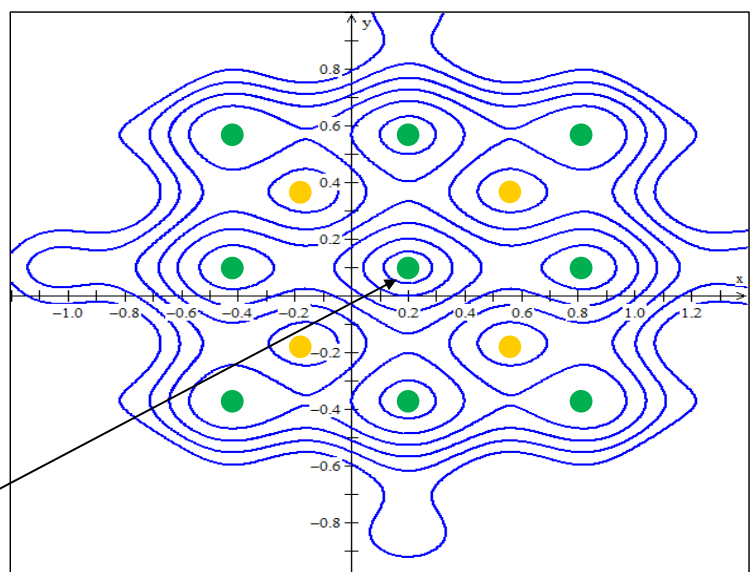
- Hypercube initial : $-1 \leq x \leq 1$
 $-1 \leq y \leq 1$

- Extrema locaux

0,570	0,883	1,190	0,470	1,190	0,883
0,367	1,346	1,654	0,933	1,654	1,346
0,100	0,413	0,720	0,000	0,720	0,413
-0,167	1,346	1,654	0,933	1,654	1,346
-0,370	0,883	1,190	0,470	1,190	0,883
y / x	-0,419	-0,161	0,200	0,561	0,819

En vert : minimum
En orange : maximum
En blanc : point-selle

- Minimum global : $x = 0,2$
 $y = 0,1$
 $f = 0$



Itération 1 : Division de l'hypercube initial

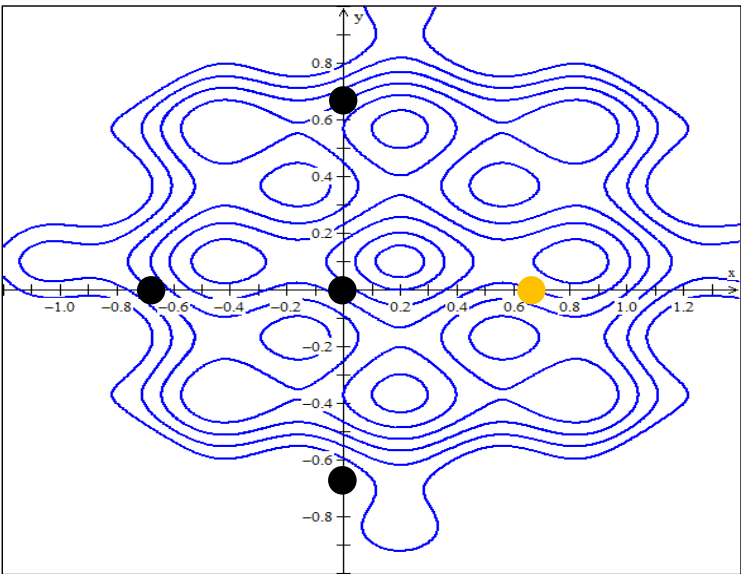
x	y	f
-0,667	0,000	1,440
0,667	0,000	0,907
0,000	-0,667	2,400
0,000	0,667	1,207
0,000	0,000	0,729

La fonction est évaluée :

- suivant x en : $x = 0 \pm \frac{2}{3}$
- suivant y en : $y = 0 \pm \frac{2}{3}$

La meilleure valeur est suivant x.
→ en orange

L'hypercube est divisé suivant x,
puis suivant y.
→ 5 hyperrectangles.



Itération 1 : 5 hyperrectangles

x mid	y mid	dx	dy	d max	fmid	fb	K
-0,667	0,000	0,333	1,000	1,000	1,440	1,174	0,267
0,667	0,000	0,333	1,000	1,000	0,907	0,640	
0,000	-0,667	0,333	0,333	0,333	2,400	2,311	
0,000	0,667	0,333	0,333	0,333	1,207	1,118	
0,000	0,000	0,333	0,333	0,333	0,729	0,640	

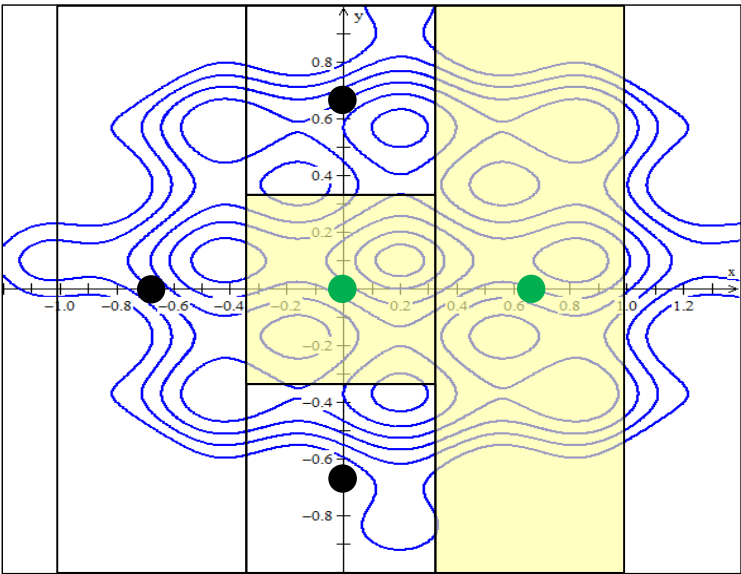
Les hyperrectangles sont rangés
par dimension décroissante.

Gris : nouveaux hyperrectangles
issus de l'itération précédente

Orange : meilleure direction
→ divisée en premier

Jaune : meilleur hyperrectangle
parmi ceux de même dimension

Vert : 2 hyperrectangles sélectionnés
donnant le minorant le plus bas
→ potentiellement optimaux

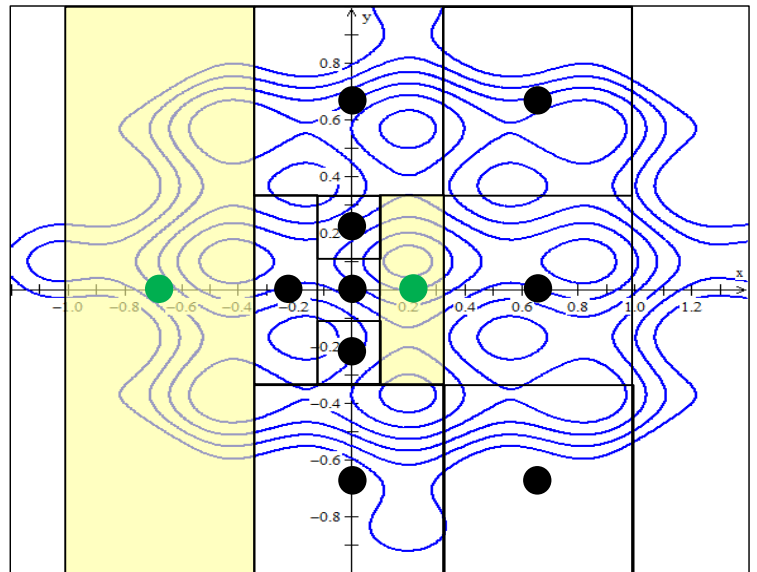


Itération 2 : 11 hyperrectangles

x mid	y mid	dx	dy	d max	fmid	fb	K
-0,667	0,000	0,333	1,000	1,000	1,440	-0,265	1,705
0,000	-0,667	0,333	0,333	0,333	2,400	1,831	
0,000	0,667	0,333	0,333	0,333	1,207	0,639	
0,667	-0,667	0,333	0,333	0,333	2,577	2,009	
0,667	0,667	0,333	0,333	0,333	1,385	0,817	
0,667	0,000	0,333	0,333	0,333	0,907	0,338	
-0,222	0,000	0,111	0,333	0,333	0,975	0,407	
0,222	0,000	0,111	0,333	0,333	0,303	-0,265	-1,915
0,000	-0,222	0,111	0,111	0,111	1,287	1,499	
0,000	0,222	0,111	0,111	0,111	0,849	1,061	
0,000	0,000	0,111	0,111	0,111	0,729	0,942	

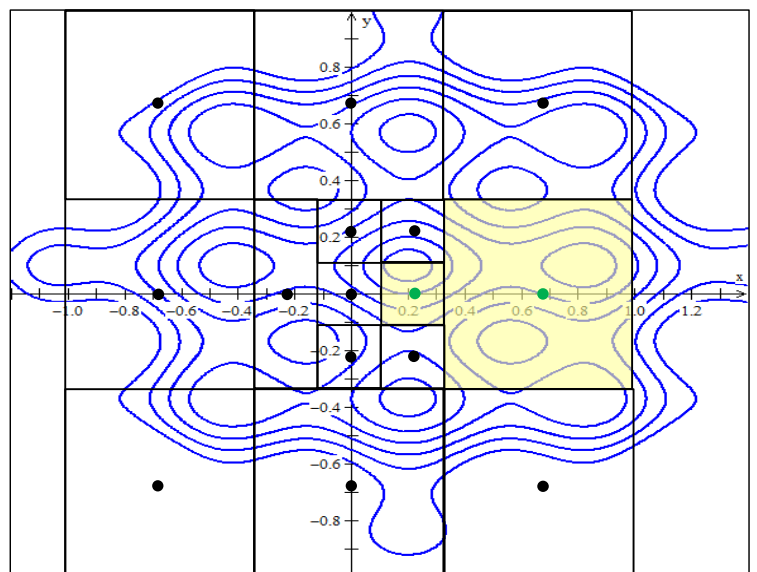
L'hyperrectangle est divisé
suivant sa plus grande dimension.
→ 2 hyperrectangles suivant y

L'hypercube est divisé
suivant ses deux dimensions
en commençant par la meilleure direction
→ 2 hyperrectangles suivant x
puis 2 hyperrectangles suivant y



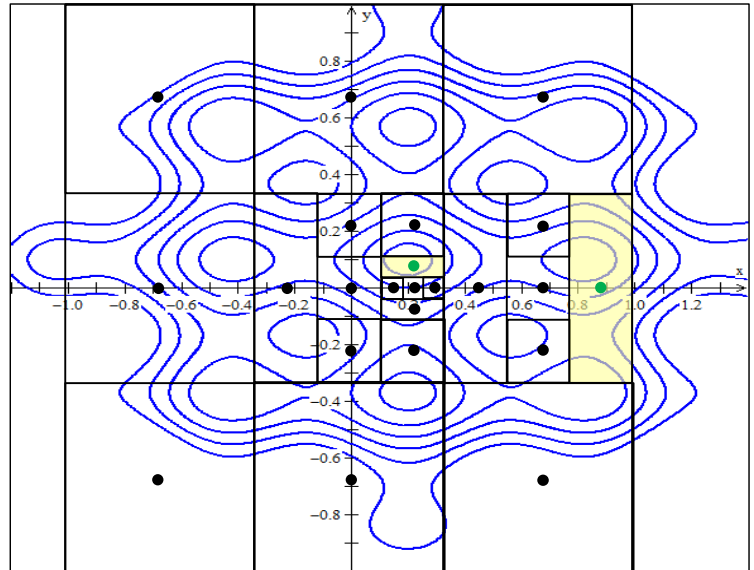
Itération 3 : 15 hyperrectangles

x mid	y mid	dx	dy	d max	fmid	fb	K
0,000	-0,667	0,333	0,333	0,333	2,400	1,494	
0,000	0,667	0,333	0,333	0,333	1,207	0,302	
0,667	-0,667	0,333	0,333	0,333	2,577	1,672	
0,667	0,667	0,333	0,333	0,333	1,385	0,480	
0,667	0,000	0,333	0,333	0,333	0,907	0,002	2,715
-0,222	0,000	0,111	0,333	0,333	0,975	0,070	
-0,667	-0,667	0,333	0,333	0,333	3,111	2,205	
-0,667	0,667	0,333	0,333	0,333	1,918	1,013	
-0,667	0,000	0,333	0,333	0,333	1,440	0,535	
0,000	-0,222	0,111	0,111	0,111	1,287	0,985	
0,000	0,222	0,111	0,111	0,111	0,849	0,547	
0,222	-0,222	0,111	0,111	0,111	0,861	0,559	
0,222	0,222	0,111	0,111	0,111	0,423	0,121	
0,222	0,000	0,111	0,111	0,111	0,303	0,002	
0,000	0,000	0,111	0,111	0,111	0,729	0,942	



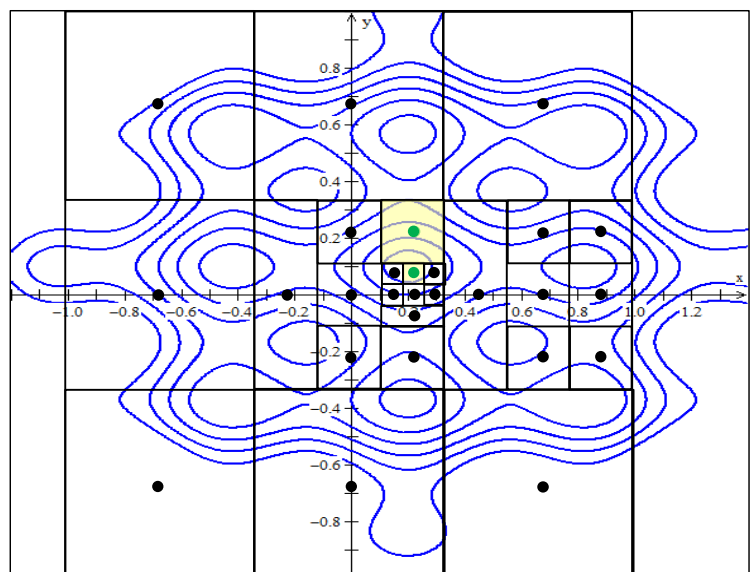
Itération 4 : 23 hyperrectangles

x mid	y mid	dx	dy	d max	fmid	fb	K
0,000	-0,667	0,333	0,333	0,333	2,400	1,277	
0,000	0,667	0,333	0,333	0,333	1,207	0,085	
0,667	-0,667	0,333	0,333	0,333	2,577	1,455	
0,667	0,667	0,333	0,333	0,333	1,385	0,263	
-0,222	0,000	0,111	0,333	0,333	0,975	-0,147	
-0,667	-0,667	0,333	0,333	0,333	3,111	1,989	
-0,667	0,667	0,333	0,333	0,333	1,918	0,796	
-0,667	0,000	0,333	0,333	0,333	1,440	0,318	
0,444	0,000	0,111	0,333	0,333	0,857	-0,265	
0,889	0,000	0,111	0,333	0,333	0,778	-0,345	
0,667	-0,222	0,111	0,111	0,111	1,464	1,090	3,366
0,667	0,222	0,111	0,111	0,111	1,026	0,652	
0,667	0,000	0,111	0,111	0,111	0,907	0,533	
0,000	-0,222	0,111	0,111	0,111	1,287	0,913	
0,000	0,222	0,111	0,111	0,111	0,849	0,475	
0,222	-0,222	0,111	0,111	0,111	0,861	0,487	
0,222	0,222	0,111	0,111	0,111	0,423	0,049	
0,000	0,000	0,111	0,111	0,111	0,729	0,355	
0,222	-0,074	0,111	0,037	0,111	0,699	0,325	
0,222	0,074	0,111	0,037	0,111	0,029	-0,345	
0,148	0,000	0,037	0,037	0,037	0,334	0,471	-3,699
0,296	0,000	0,037	0,037	0,037	0,421	0,558	
0,222	0,000	0,037	0,037	0,037	0,303	0,440	



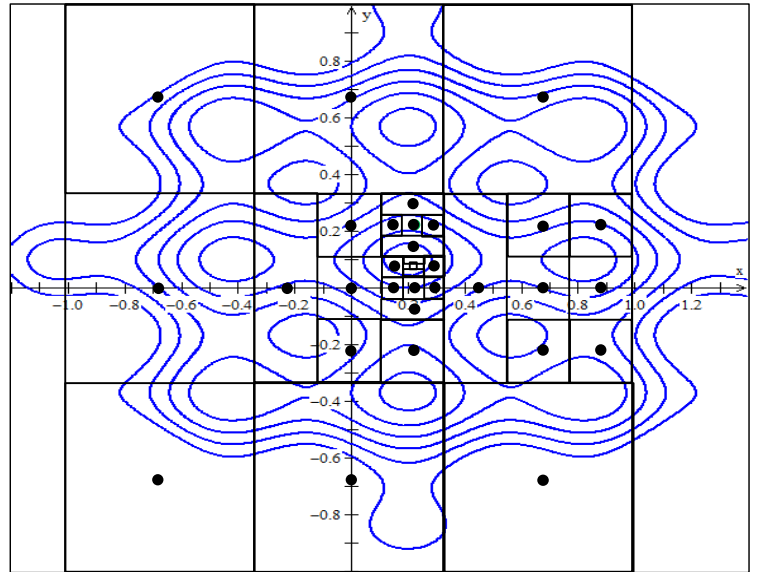
Itération 5 : 27 hyperrectangles

x mid	y mid	dx	dy	d max	fmid	fb	K
0,000	-0,667	0,333	0,333	0,333	2,400	1,749	
0,000	0,667	0,333	0,333	0,333	1,207	0,556	
0,667	-0,667	0,333	0,333	0,333	2,577	1,926	
0,667	0,667	0,333	0,333	0,333	1,385	0,734	
-0,222	0,000	0,111	0,333	0,333	0,975	0,325	
-0,667	-0,667	0,333	0,333	0,333	3,111	2,460	
-0,667	0,667	0,333	0,333	0,333	1,918	1,268	
-0,667	0,000	0,333	0,333	0,333	1,440	0,789	
0,444	0,000	0,111	0,333	0,333	0,857	0,206	
0,667	-0,222	0,111	0,111	0,111	1,464	1,247	
0,667	0,222	0,111	0,111	0,111	1,026	0,809	1,953
0,667	0,000	0,111	0,111	0,111	0,907	0,690	
0,000	-0,222	0,111	0,111	0,111	1,287	1,070	
0,000	0,222	0,111	0,111	0,111	0,849	0,632	
0,222	-0,222	0,111	0,111	0,111	0,861	0,644	
0,222	0,222	0,111	0,111	0,111	0,423	-0,167	
0,000	0,000	0,111	0,111	0,111	0,729	0,512	
0,222	-0,074	0,111	0,037	0,111	0,699	0,109	
0,889	-0,222	0,111	0,111	0,111	1,335	0,745	
0,889	0,222	0,111	0,111	0,111	0,897	0,307	
0,889	0,000	0,111	0,111	0,111	0,778	0,187	5,313
0,148	0,000	0,037	0,037	0,037	0,334	0,471	
0,296	0,000	0,037	0,037	0,037	0,421	0,558	
0,148	0,074	0,037	0,037	0,037	0,060	-0,137	
0,296	0,074	0,037	0,037	0,037	0,147	-0,050	
0,222	0,074	0,037	0,037	0,037	0,029	-0,167	
0,222	0,000	0,037	0,037	0,037	0,303	0,107	



Itération 6 : 35 hyperrectangles

x mid	y mid	dx	dy	d max	fmid	fb
0,000	-0,667	0,333	0,333	0,333	2,400	1,238
0,000	0,667	0,333	0,333	0,333	1,207	0,046
0,667	-0,667	0,333	0,333	0,333	2,577	1,416
0,667	0,667	0,333	0,333	0,333	1,385	0,224
-0,222	0,000	0,111	0,333	0,333	0,975	-0,186
-0,667	-0,667	0,333	0,333	0,333	3,111	1,949
-0,667	0,667	0,333	0,333	0,333	1,918	0,757
-0,667	0,000	0,333	0,333	0,333	1,440	0,279
0,444	0,000	0,111	0,333	0,333	0,857	-0,304
0,667	-0,222	0,111	0,111	0,111	1,464	1,077
0,667	0,222	0,111	0,111	0,111	1,026	0,639
0,667	0,000	0,111	0,111	0,111	0,907	0,520
0,000	-0,222	0,111	0,111	0,111	1,287	0,900
0,000	0,222	0,111	0,111	0,111	0,849	0,462
0,222	-0,222	0,111	0,111	0,111	0,861	0,474
0,000	0,000	0,111	0,111	0,111	0,729	0,342
0,222	-0,074	0,111	0,037	0,111	0,699	0,312
0,889	-0,222	0,111	0,111	0,111	1,335	0,948
0,889	0,222	0,111	0,111	0,111	0,897	0,510
0,889	0,000	0,111	0,111	0,111	0,778	0,390
0,222	0,148	0,111	0,037	0,111	0,083	-0,304
0,222	0,296	0,111	0,037	0,111	0,796	0,409
0,148	0,222	0,037	0,037	0,037	0,454	0,416
0,296	0,222	0,037	0,037	0,037	0,540	0,503
0,222	0,222	0,037	0,037	0,037	0,423	0,385
0,148	0,000	0,037	0,037	0,037	0,334	0,137
0,296	0,000	0,037	0,037	0,037	0,421	0,224
0,148	0,074	0,037	0,037	0,037	0,060	0,197
0,296	0,074	0,037	0,037	0,037	0,147	0,284
0,222	0,000	0,037	0,037	0,037	0,303	0,107
0,222	0,049	0,037	0,012	0,037	0,090	0,053
0,222	0,099	0,037	0,012	0,037	0,007	-0,031
0,198	0,074	0,012	0,012	0,012	0,022	0,030
0,247	0,074	0,012	0,012	0,012	0,053	0,061
0,222	0,074	0,012	0,012	0,012	0,029	0,037



447

Exemple à deux variables

$$f(x, y) = (x - 0.2)^2 + 2(y - 0.1)^2 - 0.3 \cos[3\pi(x - 0.2)] - 0.4 \cos[4\pi(y - 0.1)] + 0.7$$

- On obtient après **35 évaluations** une approximation du minimum global.

$$\begin{cases} x \approx 0,222 & (\text{au lieu de } 0,2) \\ y = 0,099 & (\text{au lieu de } 0,1) \\ f \approx 0,007 & (\text{au lieu de } 0,0) \end{cases}$$

- Evolution du meilleur point au cours des itérations

Itération	x	y	f
1	0,000	0,000	0,729
2	0,222	0,000	0,303
3	0,222	0,000	0,303
4	0,222	0,074	0,029
5	0,222	0,074	0,029
6	0,222	0,099	0,007

