Algorithmes de minimisation

Sébastien Charnoz & Adrian Daerr

Université Paris 7 Denis Diderot
CEA Saclay

De nombreux problèmes nécessitent de minimiser une fonction :

-Minimiser la distance (XHI2) entre des points de mesures et une courbe

- Trouver l' état d'équilibre d'un système mécanique (Minimiser E_{pot})

- Trouver l'état d'équilibre d'un gaz, d'un mélange (Maximiser Entropie)

- Tous problèmes d'optimisation (minimiser le coût , certaine fonctions etc…)

Etc….

Tous ces problèmes entrent dans une grande catégorie : L'optimisation

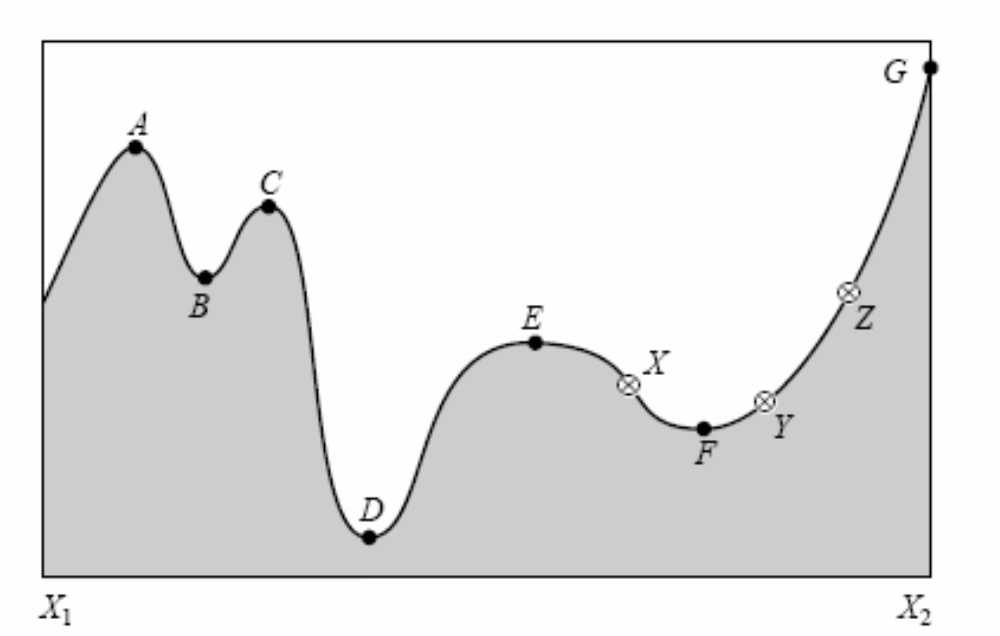
Remarque : Maximiser F(x) = Minimiser –F(x)
Donc minimiser et maximiser sont en fait le même problème !

Une fois de plus, c’est un problème mal posé :

Il n’y a pas de méthode idéale.

Il est facile de trouver un minimum local.

Il est difficile trouver un minimum absolu



B,D,F sont des minimums locaux

D est le minimum absolu

Cette recherche est d’autant plus longue est difficile si la fonction dépend de plusieurs variables => minimisation en N dimensions

Certains problèmes sont dit : « contraints » quand on doit minimiser une fonction et en plus, respecter certaines conditions sur les variables:

Ex 1 : Réaction chimique à l'équilibre :
Pour trouver la composition à l'équilibre on doit minimiser le potentiel chimique du système MAIS on doit respecter le fait que la masse totale du système est conservée…

Ex 2 : Calculer la forme d'un récipient de surface minimale qui contient un volume fixé… (ex : boîtes de conserve)

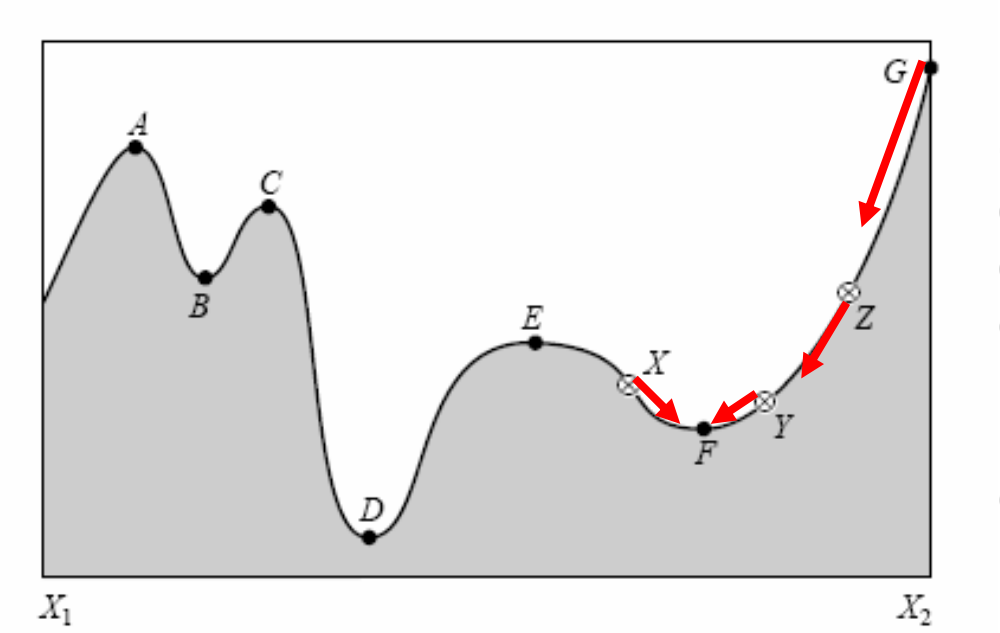
Etc….

Il existe aussi des méthodes spécifiques pour ces cas.

La méthode classique de minisation est la descente de gradient :

On part d'un point X0 donné par l'utilisateur

On descend le long de la plus grande pente locale



C'est pourquoi la plus-part
des algorithmes trouvent
des *minimums locaux*
et non le *minimum absolu*

Une grand partie du calcule numérique
consiste à trouver la dérivée F'(x)

IMPORTANCE DE F’(X)

Soit F(x) à minimiser (x peut être un vecteur)

En général, si vous connaissez F’(X) (le gradient) cela pourra vous faciliter la tache.

Si vous utiliser un algorithme de minimisation cela signifie que vous ne pouvez calculer analytiquement les zéros de la fonction F’(X)

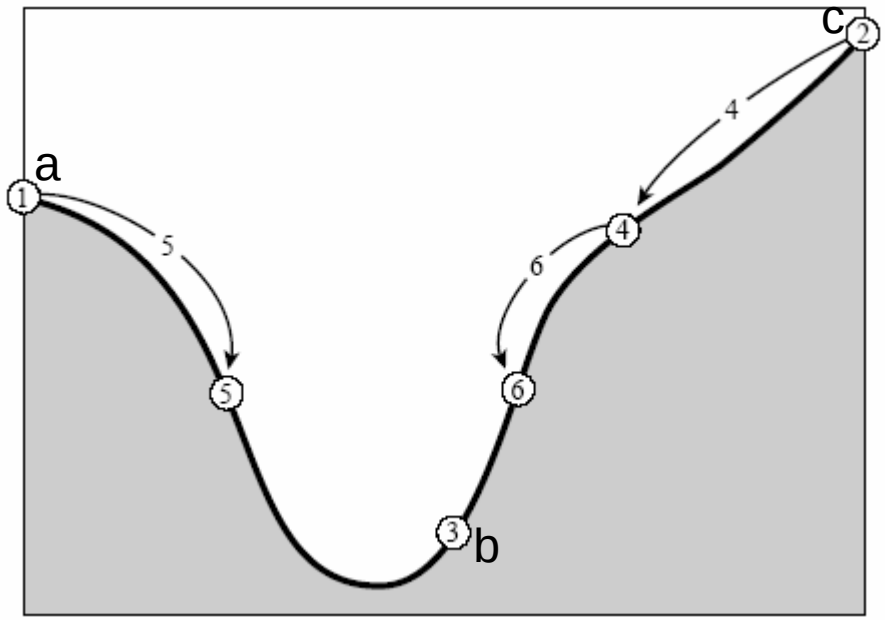
-Soit par ce que vous ne connaissez pas F’(x)

-Soit par ce que résoudre F’(X)=0 est trop compliqué.

D’une manière générale, si vous pouvez calculer F’(X) faites le, cela facilitera grandement la tâche de l’algorithme de minimisation.

MINIMISATION A 1 DIMENSION SANS GRADIENT

C'est la méthode la plus simple, très semblable à la méthode de bissection pour trouver un zéro



Méthode de base

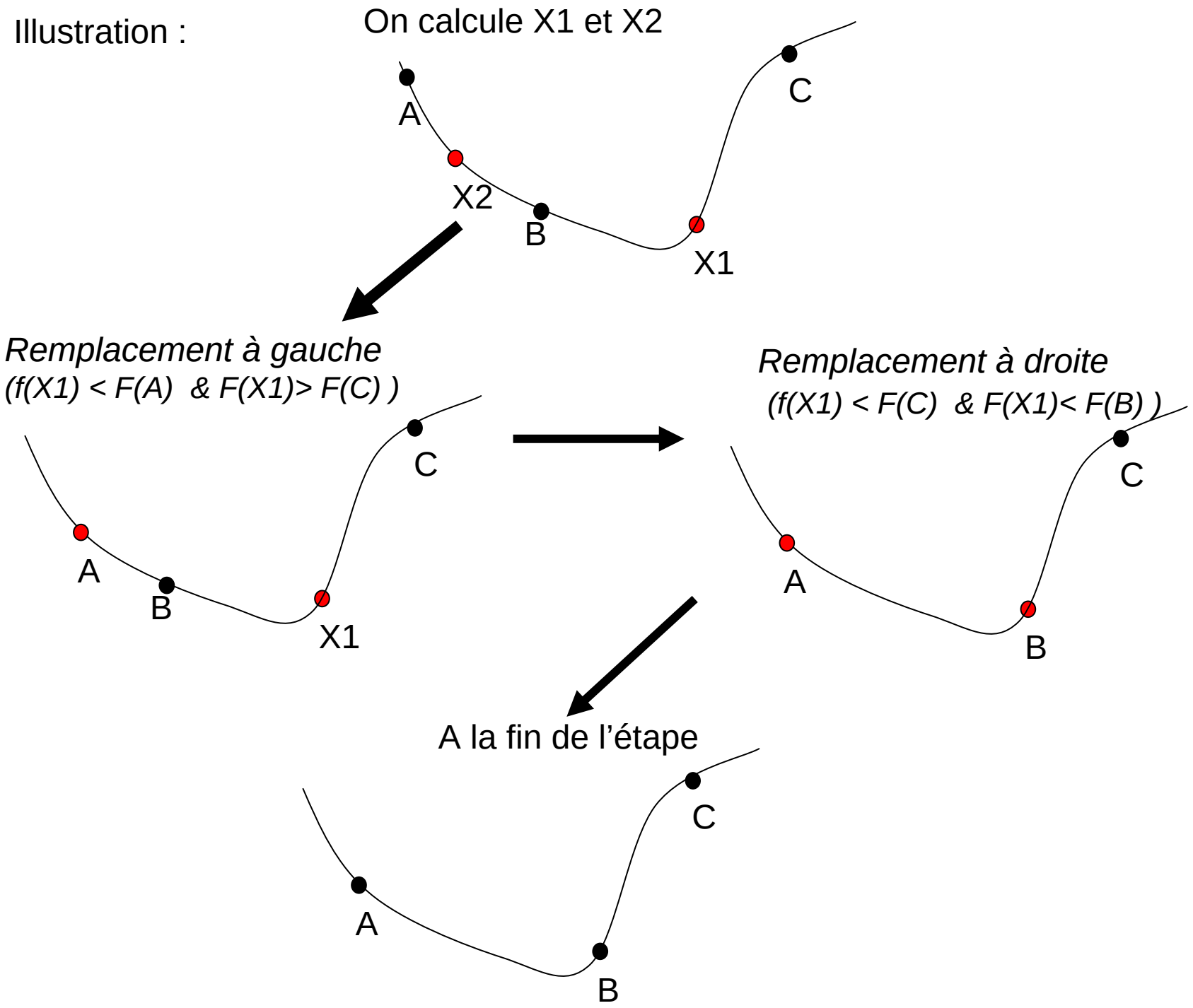
Méthode générale

Encadrer le minimum entre 3 points :
a<b<c

tel que f(b)< f(a)
f(b) < f(c)

1. On part de a et b et c
2. calcule f(a),f(b),f(c)
3. On prend x1 entre a et b
4. Si f(x1)< f(a) alors a=x1 *Test x à gauche*
   Si f(x1)< f(b) alors b=x1
5. On prend x2 entre b et c *Test x à droite*
   Si f(x2)< f(c) alors c=x2
   Si f(x2)< f(b) alors b=x2
6. Retourne en 2

Illustration :



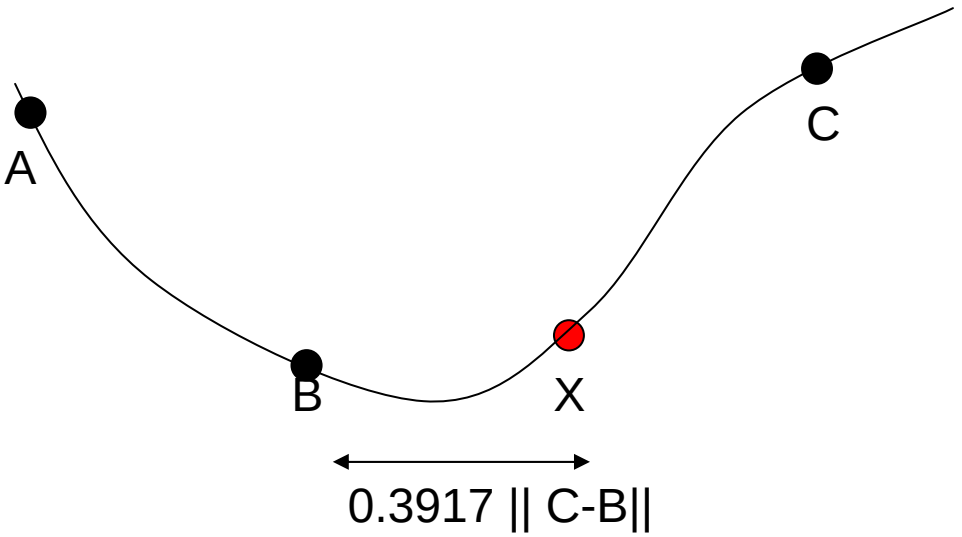
Méthode de la « régle d'Or » pour trouver X

En fait on peut montrer qu'il y a une manière optimale de choisir le nouveau point X en une seule étape.

On détermine le plus grand des deux segments droite gauche:
|| A-B|| et ||C-B|| . Soit D la longueur du segment

Et on prend le nouveau point X dans ce segment à la distance :
0.39197 x D (nombre d'or) du point B

Si F(x) < F(B) alors B=X ou si F(x)< F(c) alors C=X



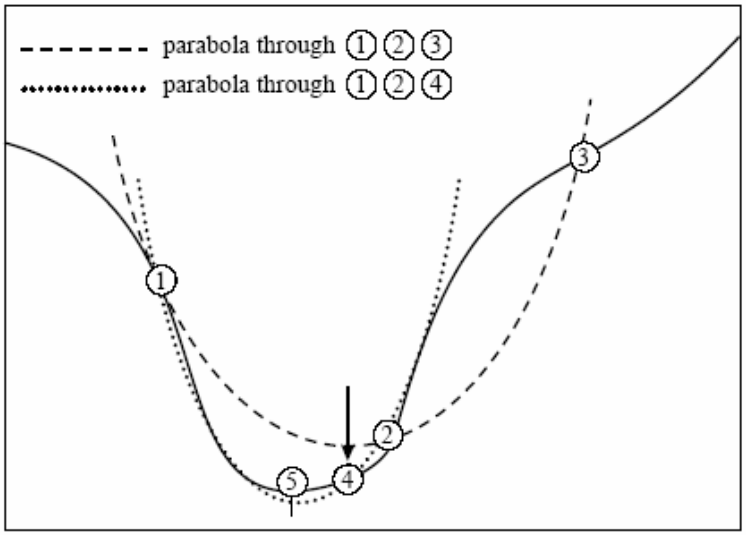
Méthode parabolique pour trouver X

On peut faire mieux que la régle d'Or si F est gentiment parabolique prés du minimum.
Si F est *effectivement* parabolique sur l'intervalle [a,c], alors on peut analytiquement calculer le minimum

$$x = b - \frac{1}{2}\frac{(b-a)^2[f(b)-f(c)] - (b-c)^2[f(b)-f(a)]}{(b-a)[f(b)-f(c)] - (b-c)[f(b)-f(a)]}$$

Rem : ne marche pas si F est linéaire

En pratique, F est rarement *exactement* parabolique. On peut cependant utiliser Ce calcule de X pour trouver le nouvel encadrement du minimum



Pour l'étape suivante

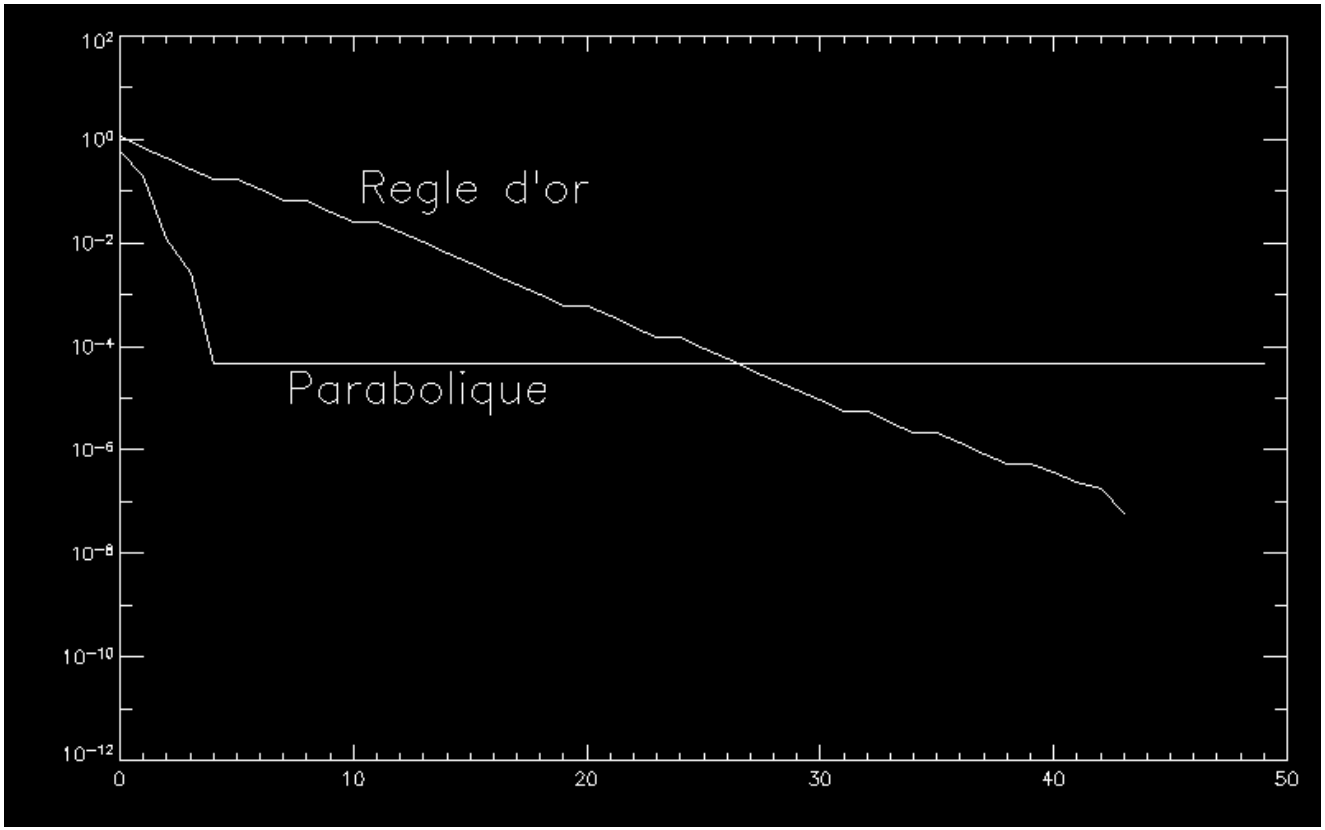
Si x<b alors ABC=[a,x,b] (gauche)

Si x> b alors ABC=[b,x,c] (droite)

Comparaison des deux méthode : minimisation de Sin(x)^2

Erreur:

||A-C||/B



N itérations

On voit que la méthode parabolique converge beaucoup plus rapidement que la méthode de la règle d'Or.

Cependant sa convergence est limitée (> 10^{-5}) …. Pourquoi ?

La méthode parabolique ne peut marcher pour des fonctions linéaires…

Donc si on est trop prés du minimum, la fonction devient localement linéaire et la méthode arrête de converger !!

En pratique ce qui se passe : [A,B,C] deviennent constant.

Si cela arrive il suffit de considérer que la solution est B.

Minimisation à 1D en utilisant le gradient (dérivée)

En 1 dimension, en fait le gradient n'apporte qu'une information assez limité…

Pourquoi ?
A 1D le domaine de recherche est assez étroit, donc on gagne peu.

3 possibilités

1
On utilisera la méthode d'encadrement à trois points [a,b,c],
et la dérivée au point c ,f '(x) nous permettra de choisir plus rapidement
la direction du prochain point (dans [a,b] ou dans [b,c])

2 *
On descend le long du gradient à pas fixe

3 *
Si on connaît la dérivée seconde , on peut descendre le gradient avec un
pas optimal

* ***Les méthodes 2 et 3 sont intéressantes pour les fonctions à plusieurs dimensions (prochain chapitre)***

MINIMISATION AVEC DESCENTE DE GRADIENT, On connaît la dérivée

On connaît F et F'(X)

On part d'un point de départ X_0

On calcule la suite

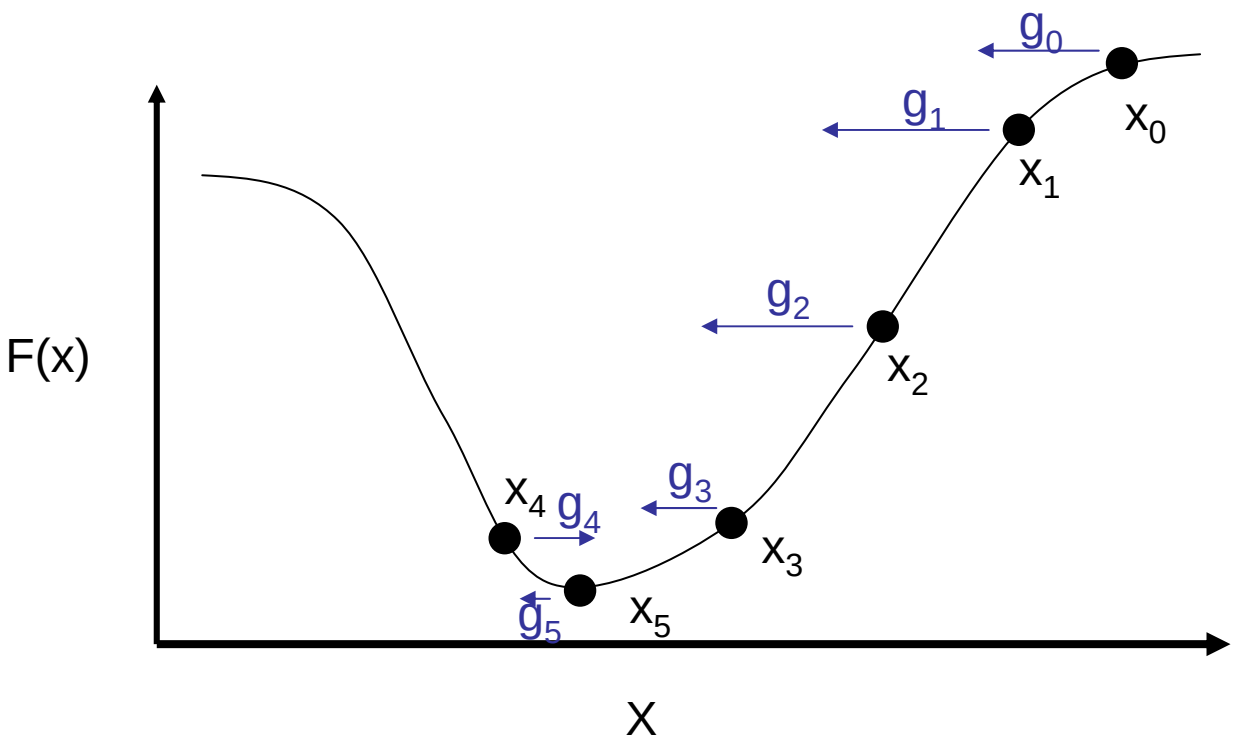
$$X_{k+1} = X_k + d_k g_k$$

Où g_k est la direction de descente et d_k est le pas de descente

Le choix de g_k est d_k se fera de sorte que $F(X_{k+1}) < F(X_k)$

$$g_k = -\left.\frac{df}{dx}\right|_{X_k}$$

Dans la méthode la plus simple de descente de gradient, g est simplement – le gradient de F



Exemple à 1D

g_i est moins le gradient du point i

$$X_{k+1} = X_k + d_k g_k$$

2 questions :
- Comment choisir le pas d_k ?
- Quel est le critère d'arrêt du calcul ?

La descente à pas fixe

Si on ne connaît pas la dérivée seconde on descend le gradient à pas fixe.
On impose $\mathbf{d_k = cst}$ au début du calcul.

Quelle valeur prendre pour **d ?** Difficile à dire à priori. En général on prend **d< la taille caractéristique du domaine de recherche**.
Par exemple la largeur de la vallée.

La descente à pas fixe ne marche que si la fonction est douce, et que si le point de départ est proche de la solution.

Le critère d'arrête sera quand

$\| \mathbf{F'(x_k)} \| < \mathbf{epsilon}$

ou $\mathbf{\|X_k - X_{k+1}\| / \| X_k \| < epsilon}$

Attention: « à pas fixe » est une dénomination abusive.
Le VRAI pas d'avancement est : d x g_k où g_k est –F'(X)

C'est le facteur multiplicatif d qui est constant.

La descente à pas optimal

Permet de répondre à la question « quelle est la distance à parcourir ».

MAIS nécessite de connaître la dérivée seconde !!

On s'inspire du DL de F(x) au point X_k

$$f(X_k + dx_k) = f(X_k) + \left.\frac{df}{dX}\right|_{x_k} dx_k + \frac{1}{2}\left.\frac{d^2 f}{dX^2}\right|_{x_k} dx_k^2 + o(d_k^2)$$

$$\Rightarrow$$

$$f'(X_k + dx_k) = f'(X_k) + \left.\frac{d^2 f}{dX^2}\right|_{x_k} dx_k$$ DL de la dérivée.

Minimiser signifie « annuler la dérivée » donc, on choisit d_k de sorte que $F'(X_k+dx_k)=0$

$$dx_k = \frac{-f'(X_k)}{\left(\frac{\partial^2 f}{\partial x^2}\right)_{X_k}}$$

Or avec nos notation :

$$X_{k+1} = X_k + d_k g_k = X_k + dx_k$$

Et g_k=-F(X_k) d'où le « pas » d'avancement d_k = dx_k/ -F(X_k)

$$d_k = \frac{1}{\left(\frac{d^2 f}{dx^2} \right)_{X_k}}$$

Cette méthode est équivalente à la méthode quadratique : l'idée est la même : trouver le minimum de la parabole locale.

L'intérêt de cette méthode est qu'elle se généralise bien à N dimensions

Minimisation multi-dimensionnelles

Minimiser F(X) où $X=(x_1 \ldots , x_n)$

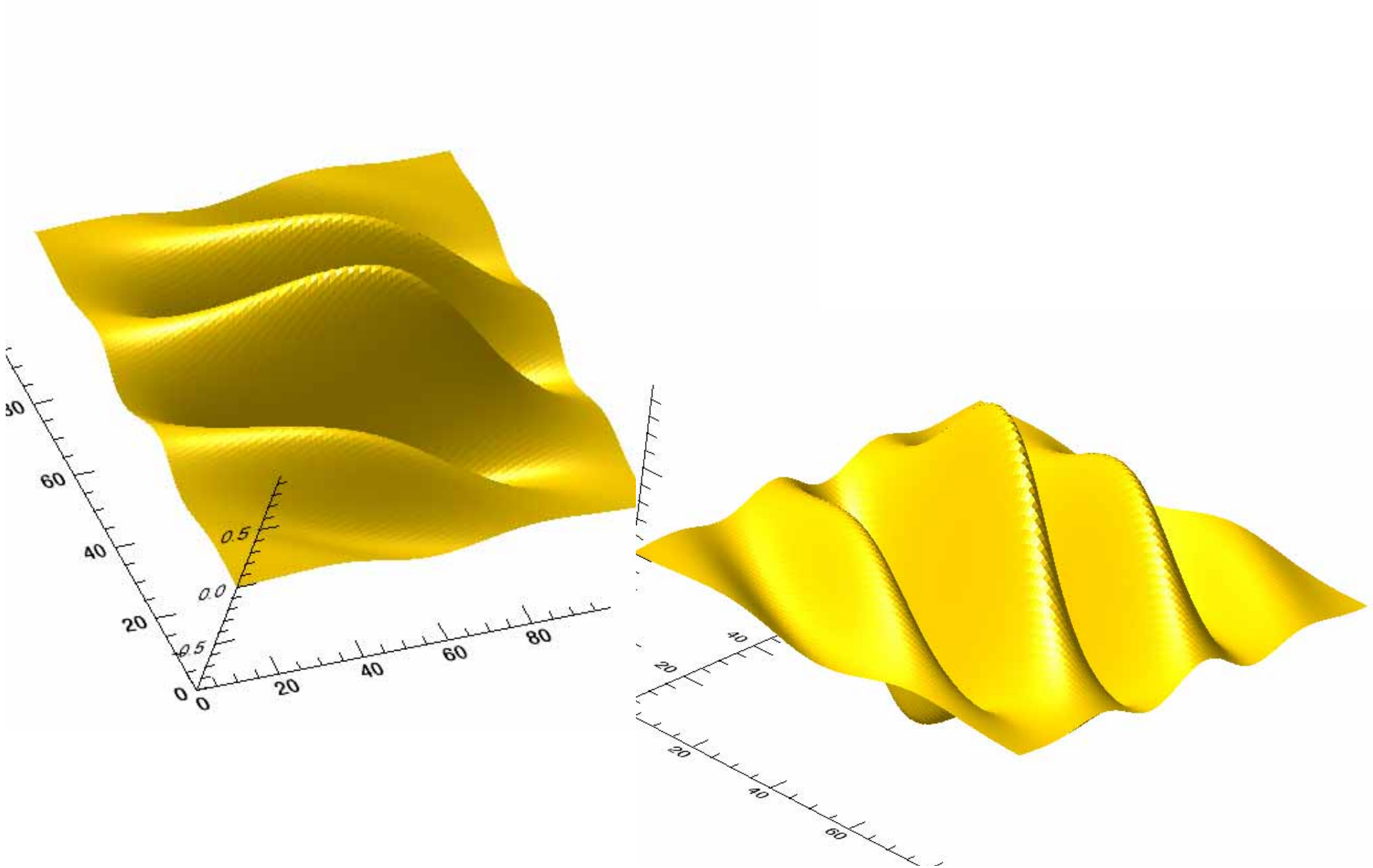
Le problème est beaucoup plus retord ...et beaucoup plus long à résoudre

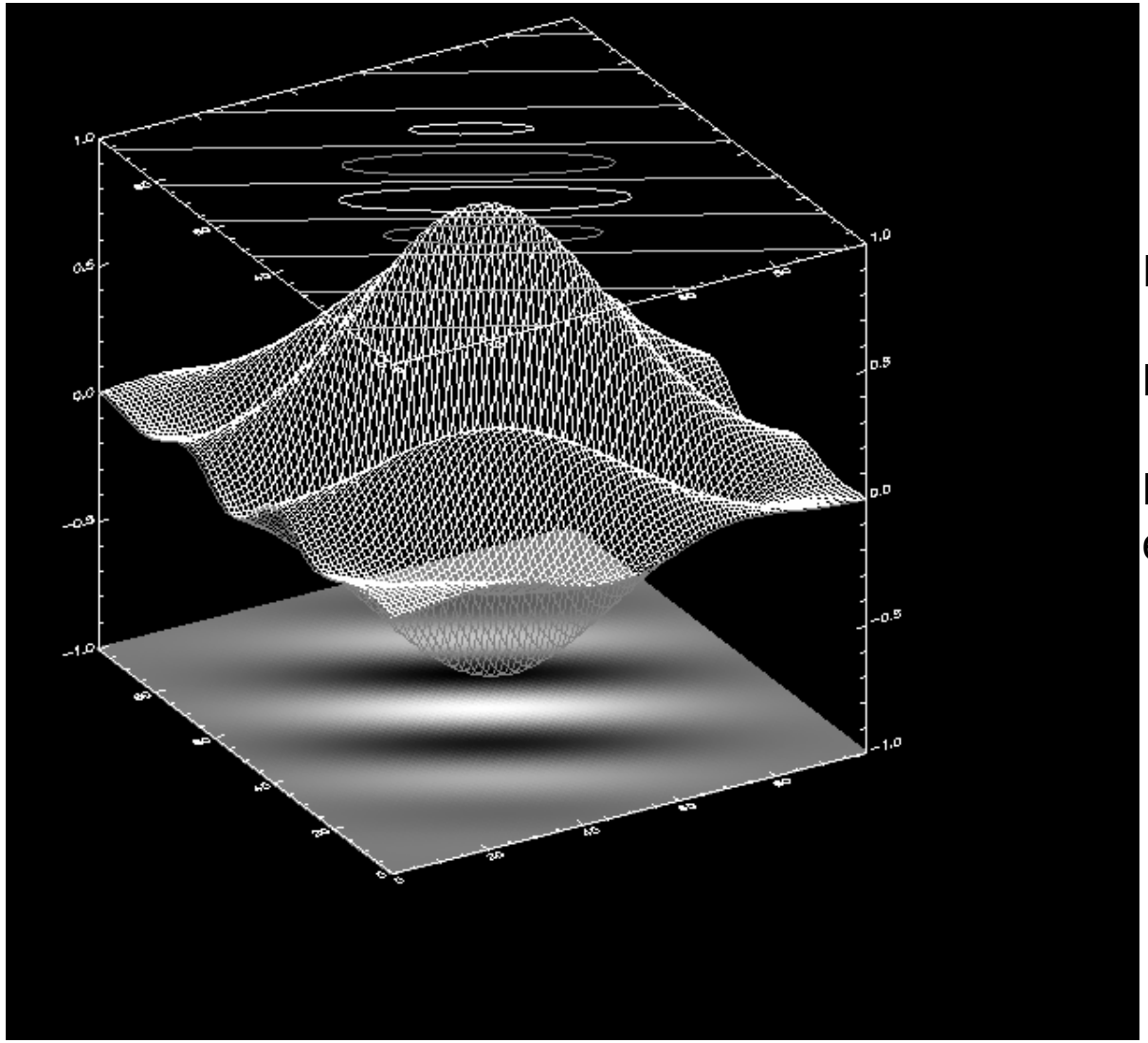
Le grand problème est souvent le CHOIX de la direction de descente

Une fois que l'on sait dans quelle direction on descend, l'autre problème est de savoir la *longueur* de la descente.

En N dimensions, ces choix sont cruciaux.

Exemple de fonction à minimiser, en 2D seulement F(x,y)





Ici : plusieurs minimums,

Existence de « vallées » étroites,

Maximums et minimums locaux etc…

Plusieurs techniques pour attaquer un tel problème. Aucune n'est universellement efficace.

TOUTES nécessitent un point de départ choisi par l'utilisateur X_0
=> Nécessité de partir d'un point X_0 pas très éloigné du minimum

La pluspart utilisent le DL de la fonction à plusieurs variables (N variables)
$X=P+[x_1, \ldots, x_n]$ où P est un vecteur

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$\approx \; c \; - \; \mathbf{b} \cdot \mathbf{x} \; + \; \frac{1}{2} \, \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}$$

$$c \equiv f(\mathbf{P}) \qquad \mathbf{b} \equiv -\nabla f|_{\mathbf{P}} \qquad [\mathbf{A}]_{ij} \equiv \left. \frac{\partial^2 f}{\partial x_i \partial x_j} \right|_{\mathbf{P}}$$

b est le gradient
A est appelé le « Hessien » de f(x). C'est simplement la matrice des dérivées secondes au point P.

Exemple de Hessien : Ici la fonction est f(x,y,z)

Hessien : A= $$\left[\begin{matrix} \frac{\partial^2 f}{\partial x \partial x} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y \partial y} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z \partial z} \end{matrix}\right]_P$$ Calculée au point P

Alors le terme en : $$\sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j = \vec{X} \cdot \left(A\vec{X} \right)$$

Est « Produit scalaire de **X** avec **AX** «

Les méthodes CLASSIQUES descendent les pentes.
Elles se divisent en deux grandes familles :

Méthodes à plusieurs points : pas de gradient ni de Hessien

* Méthode Simplex (aussi appelée « Amibe », « Amoeba » en anglais)

Méthodes mixes :

* Relaxation

Méthodes à 1 point : utilisent le gradient , ou le Hessien (si on les connaît)

- Descente de gradient à pas fixe (Gradient)
- Descente de gradient à pas optimal (Gradient + Hessien) ou
  Gradients conjugués (Gradient + Hessien)

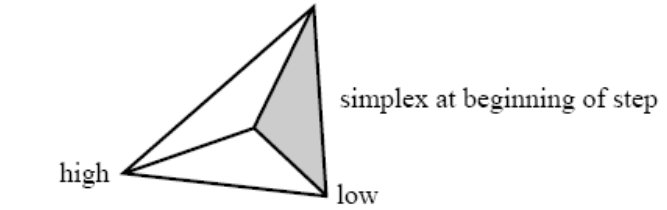
Toutes ce méthodes convergent vers un minimum local

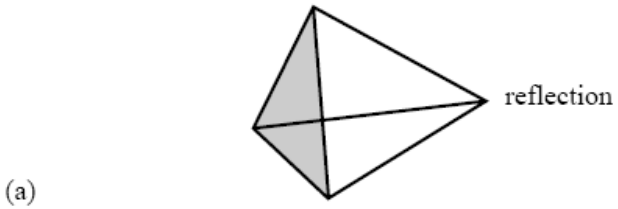
Pour trouver un minimum global, il faut faire appelle à des méthodes plus complexes qui s'inspirent souvent de la thermodynamique :

Méthode de Relaxation lente du type
« Simulated Annealing »

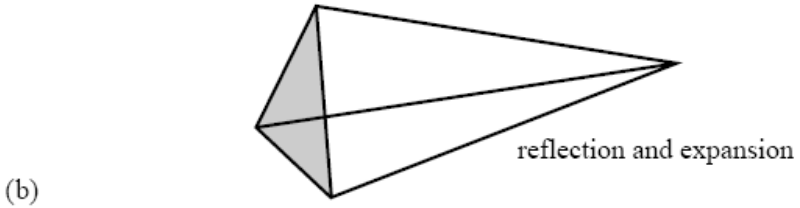
…….

On localise le point le plus haut et le point le plus bas.
Ensuite il faut resserrer cet encadrement pas à pas avec quelques règles d'évolution

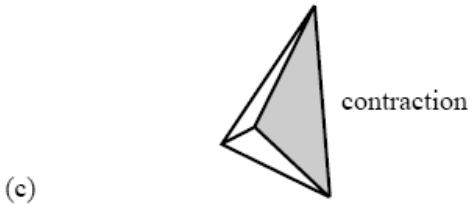




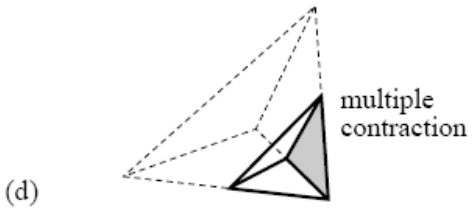
Réflexion : on refléchit le point le plus haut dans la direction opposée



Réflexion + expansion



Contraction



Contraction multiple

IL faut donc choisir de manière adéquate le cycle de transformation.

Nous ne le ferrons pas cette année.

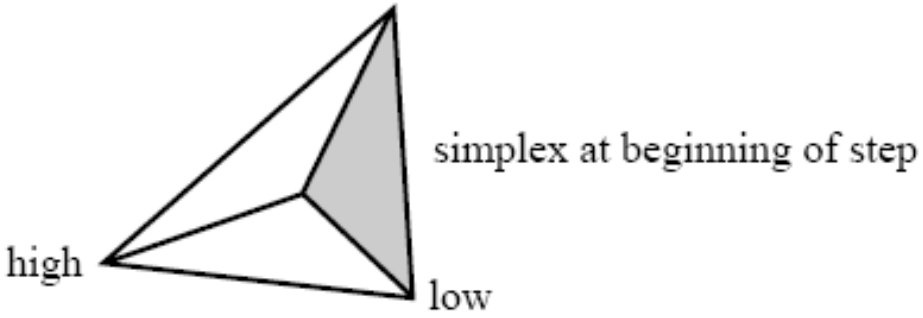
SIMPLEX - AMOEBA

Assez subtile à programmer. Nous la présentons ici pour mémoire.
Elle est assez efficace quand on ne connaît ni le gradient ni le Hessien de f

Idée : Entourer le minimum dans un **simplex**

Un Simplex est un ensemble de N+1 points qui entoure le minimum (N=nb de dimensions)

A 2D , un simplex est un triangle
A 3D , un simplex est une pyramide
Etc ….



Le minimum est dans le volume de la pyramide

Méthodes mixtes : La relaxation

L'espace dans lequel nous évoluons est orthonormée **$\{e_1, \ldots, e_n\}$**

L'idée est de parcourir successivement tous les axes de l'espace . Sur chaque axe on fait une minimisation à 1D.

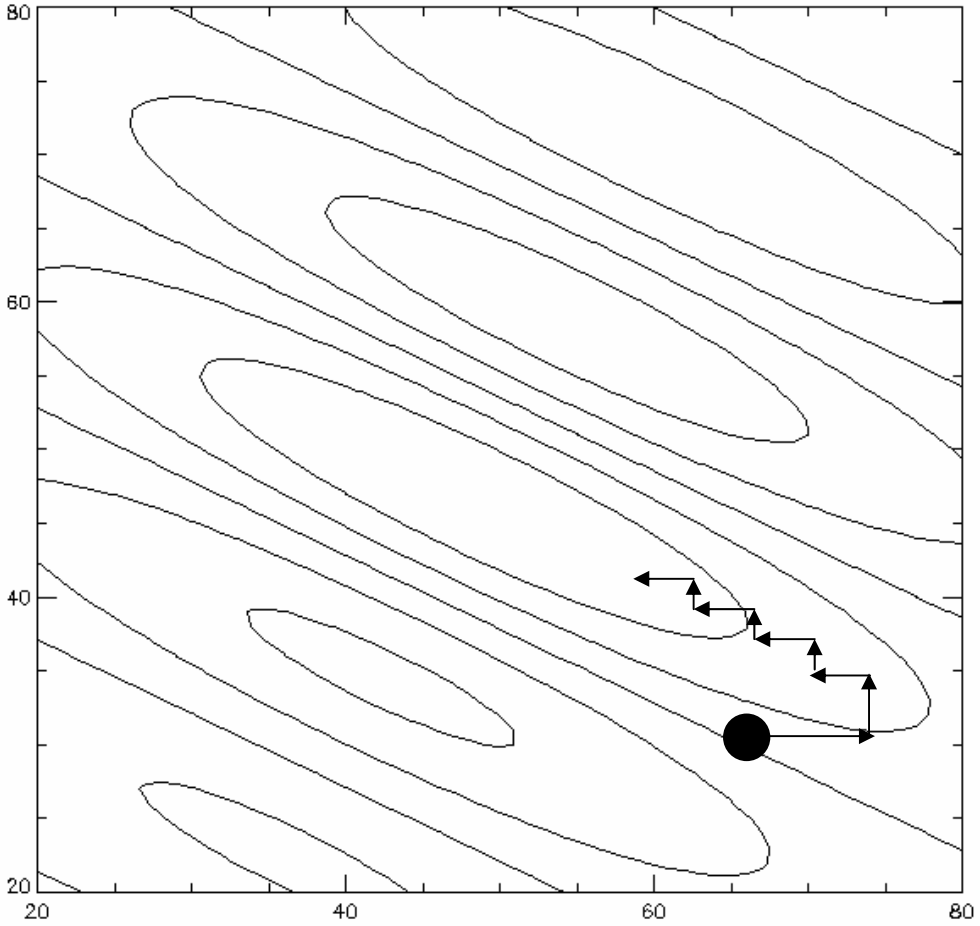
L'idée est de décomposer un problème de minimisation à N dimensions en N problèmes à 1 dimension

1- Choisir un axe dans **$\{e_1, \ldots, e_n\}$,** en commançant par **e_1**

2- Minimiser la fonction de long de cet axe (méthode au choix) $F(x_i)$

3- Si la méthode n'a pas convergé, passer à l'axe suivant

4- Si tous les axes ont été testés et que la méthode n'a pas convergé, recommencer à e1



A 2D :
1- Se mettre sur l'axe **X**
2- minimiser le long de **X**
3- Se mettre sur l'axe **Y**
4 minimiser le long de **Y**
5- Recommencer en 1
si on a pas atteint le minimum

Remarque : « minimiser le long de l'axe $\mathbf{e_i}$ » signifie

Quand nous sommes au point **P** : **$P=(p_1,\ldots,p_n)$**

On minimise la fonction **$G(x)= F[(p1, \ldots, p_i+x, \ldots p_n)]$**

Seule la $i^{ème}$ dimension évolue. Les $p_1, \ldots, p_n$ sont fixés.
Donc G(x) est une fonction à 1 seule variable.

Pour mener cette minimisation de G(x) on peut utiliser n'importe quelle méthode déjà vue à 1D.

Soit la méthode à trois points (alors se pose la question du choix des points)

Soit la méthode de descente de gradient à pas fixe

Soit la méthode de descente de gradient à pas optimal.

DESCENTE DE GRADIENT A PAS FIXE

On connaît le gradient. Même méthode qu'en 1D

On connaît F et F'(X)

X est multidimensionnel $X=(x_1 , \dots , x_n)$

On part d'un point de départ X_0

On calcule la suite

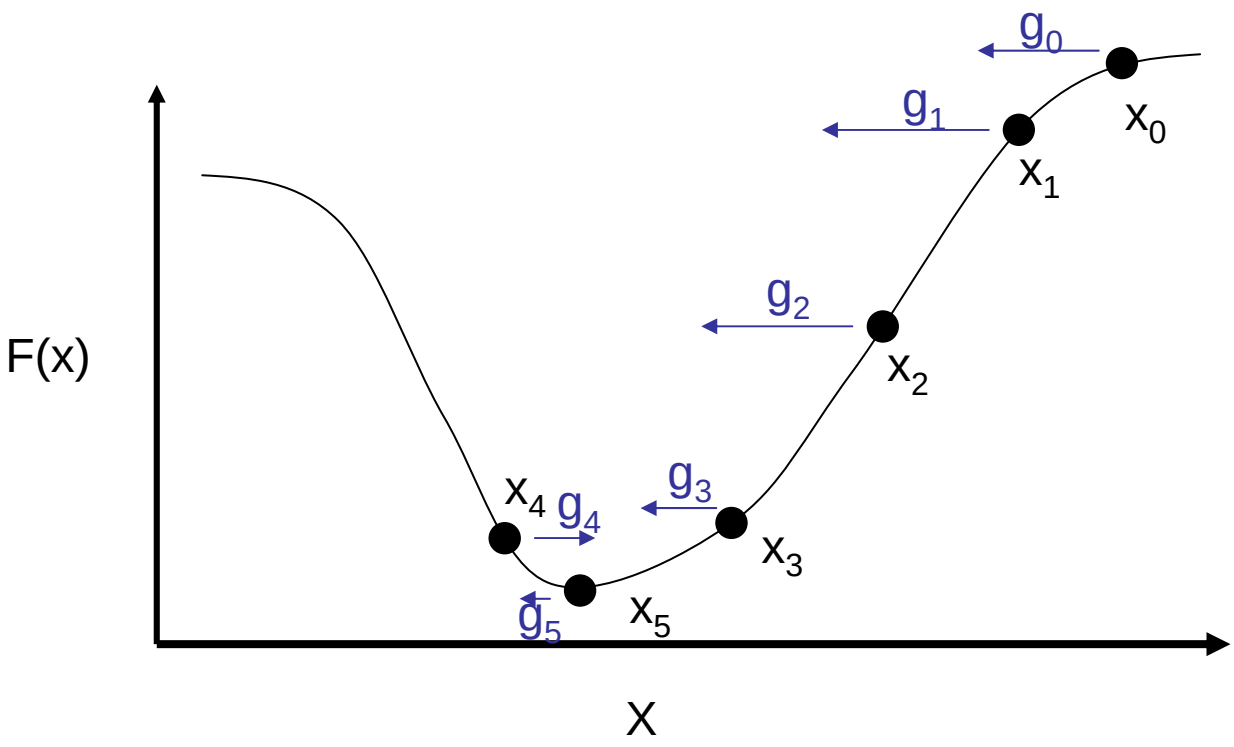
$$\vec{X}_{k+1} = \vec{X}_k + \vec{dx}_k = \vec{X}_k + d_k \vec{g}_k$$

Où g_k est la direction de descente et d_k est le pas de descente
$g_k = (g_{k1} , \dots, g_{kn})$ vecteur gradient
d_k un nombre réel

Le choix de g_k est d_k se fera de sorte que $F(X_{k+1}) < F(X_k)$

$$\vec{g}_k = -\vec{\nabla} f(X_k)$$

Dans la méthode la plus simple de descente de gradient, g est simplement – le gradient de F



Exemple à 1D

g_i est moins le gradient du point i

$$X_{k+1} = X_k + d_k g_k$$

Soit on choisit d_k comme constant (!! d_k est un vecteur)

Descente de gradient à pas optimal , en utilisant le Hessien

$$\vec{X}_{k+1} = \vec{X}_k + \vec{d}_k \cdot \vec{g}_k = \vec{X}_k + d\vec{X}_k$$

Où on pose $\mathbf{dX_k} = g_k * dk$ et où

$$\vec{g}_k = -\vec{\nabla} f(X_k)$$

$$f(X_k + dX_k) = f(X_k) + \vec{\nabla} f(X_k) \cdot \vec{d}X_k + \frac{1}{2} \sum_{i,j} \frac{d^2 f}{dx_i dx_j} dX_k^2 + o(dX_k^2)$$

$$\Rightarrow$$

$$\vec{\nabla} f(X_k + dX_k) = \vec{\nabla} f(X_k) + \sum_{i,j} \frac{d^2 f}{dx_i dx_j} \vec{d}X_k + o(dX_k)$$

Les termes en dérivée 1er et 2nd
peuvent s'écrire plus simplement.

$$\vec{\nabla} f(X_k + dX_k) = \vec{\nabla} f(X_k) + \sum_{i,j} \frac{d^2 f}{dx_i dx_j} \vec{d}X_k + o(dX_k)$$

$$\begin{pmatrix} \frac{\partial f}{dx} \\ \frac{\partial f}{dy} \\ \frac{\partial f}{dz} \end{pmatrix}_{X_k} = \vec{g}$$

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x \partial x} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y \partial y} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z \partial z} \end{bmatrix}_{X_k} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = A\vec{d}x$$

D'où $$\vec{\nabla} f(X_k + dX_k) = \vec{G} + A\vec{d}x$$

$$\vec{\nabla} f(X_k + dX_k) = \vec{g} + A\vec{dx}$$

Or nous voulons trouver le minimum, donc annuler le gradient donc :

$$0 = \vec{g} + Ad\vec{x} \Rightarrow$$

$$or \quad d\vec{x} = -d\vec{g} \Rightarrow$$

$$0 = \vec{g} - (A\vec{g})d \Leftrightarrow$$

$$0 = \vec{g}^2 - \vec{g} \cdot (A\vec{g})d \Rightarrow$$

$$d = \frac{g^2}{\vec{g} \cdot (A\vec{g})}$$

d est la distance optimale de descente le long du gradient..

Mise en pratique :

A chaque étape du calcul :

- Calculer la direction de descente : $\vec{g}_k = -\vec{\nabla} f(X_k)$

-Calculer la longueur de descente : $d_k = \frac{g_k^{\ 2}}{\vec{g}_k \cdot (A\vec{g}_k)}$

- En déduire le nouveau point X_k $X_{k+1} = X_k + d_k \times \vec{g}_k$

Mais cette méthode a ses limites :

Parfois descendre le long du gradient (i.e la plus grande pente) n’est pas toujours optimal.

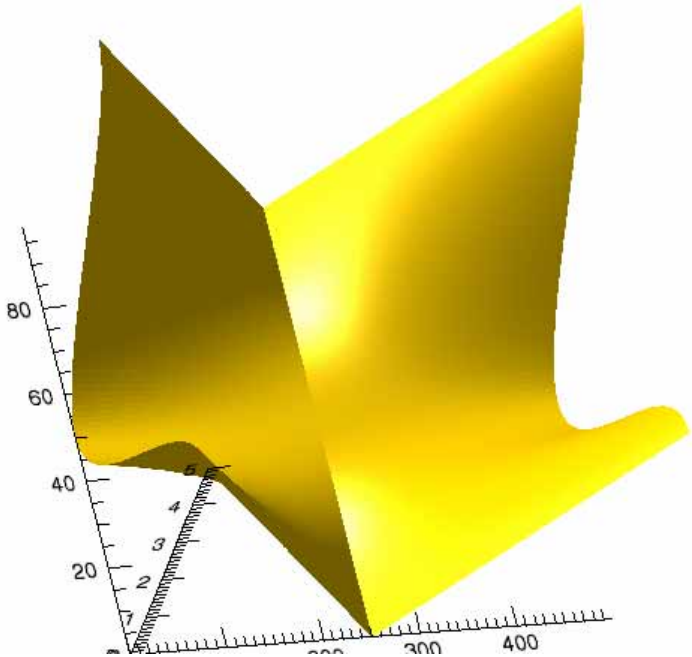
Parfois il ne FAUT pas descendre le long de la ligne de plus grande pente !!!

La méthode de descente de gradient à pas constant « tombera » facilement dans ce piège !!!

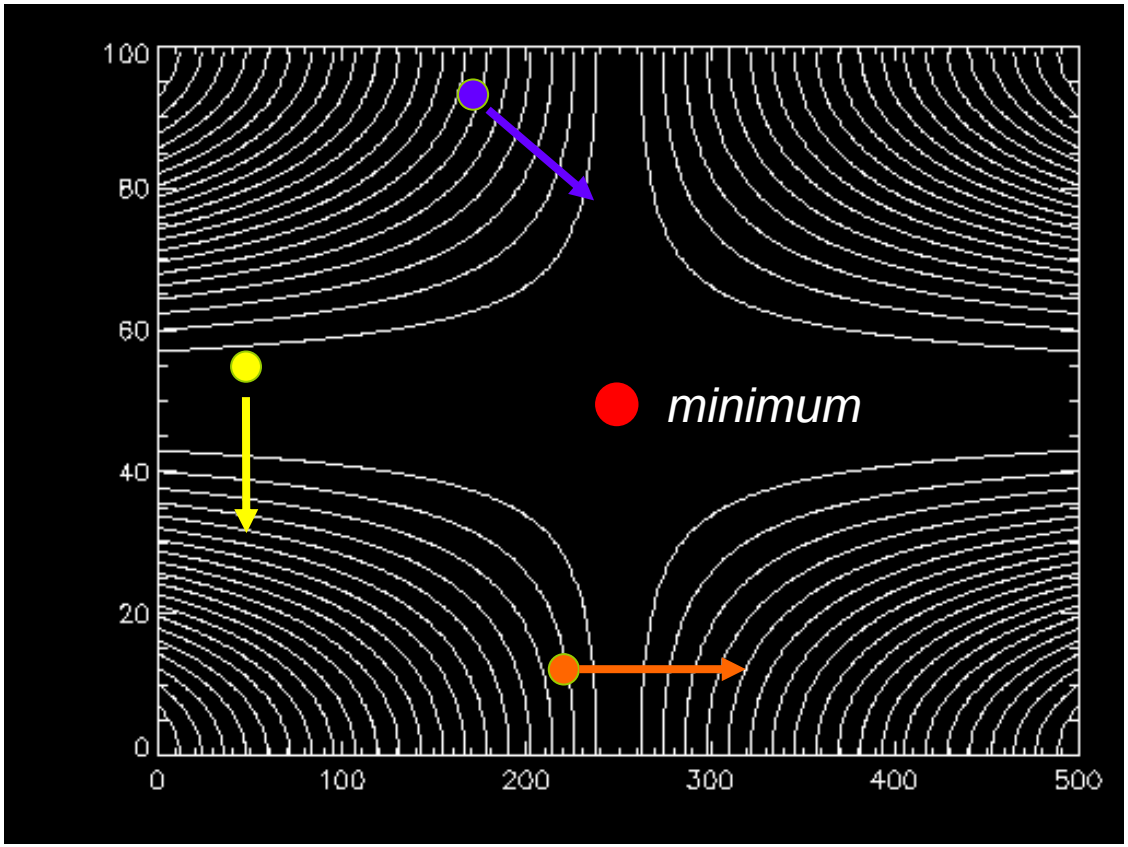
EXEMPLE :

Soit à minimiser F(x,y) (dessin ci-dessous)

Imaginons que nous sommes dans une « vallée » étroite.



La flèche est la direction de - gradient.
Pour les points jaune et orange le gradient n'est pas la meilleure direction !



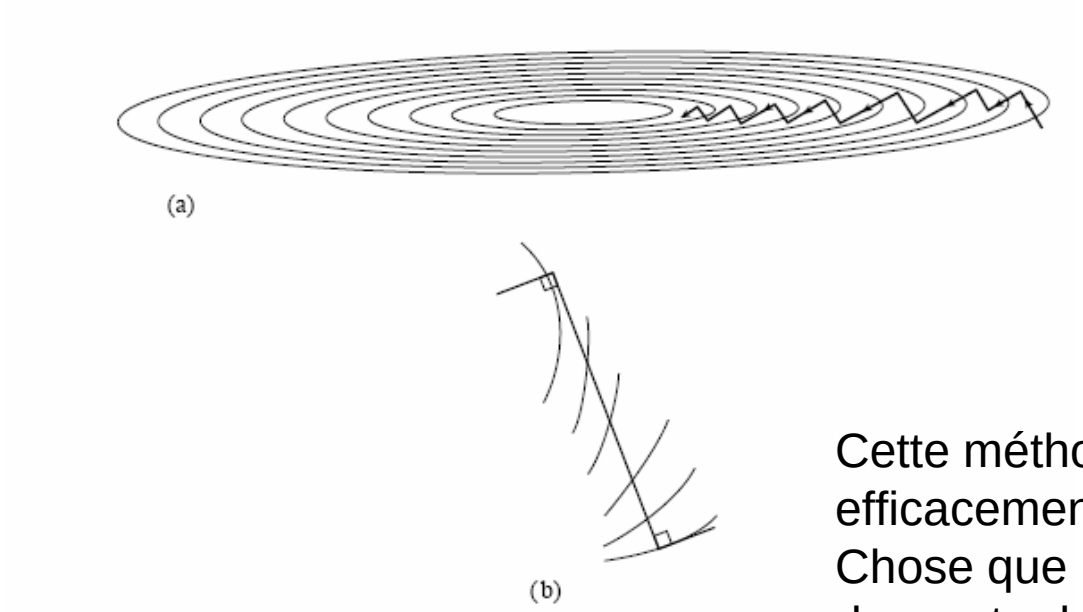
=> gradient conjugué

La méthode du Gradient Conjugué

Idée forte:

On minimise une fonction selon une première direction $\mathbf{u_k}$ .On arrive à un point X_k qui est le minimum sur cette direction.
Quelle est la meilleure direction suivante $\mathbf{u_{k+1}}$ pour continuer à minimiser **f** ?

C'est une direction perpendiculaire à $\mathbf{u_k}$



Comme cela on est sure de ne jamais «détruire » le minimum Que nous venons de trouver dans une dimension.

Cette méthode permet de parcourir efficacement une « vallée » étroite… Chose que ne peut faire une simple descente de gradient

L’algorithme sera donc comme cela :

$$\vec{X}_{k+1} = \vec{X}_k + d_k \vec{u}_k$$

Où d_k est la distance de descente et $\mathbf{u_k}$ est la direction de descente.
MAIS dans la méthode du gradient conjugué, $\mathbf{u_k}$ n’est pas le gradient (à priori),
Ce sera quelquechose de plus compliqué..

En fait la direction de descente $\mathbf{u_k}$ variera à chaque étape.
En particulier on doit avoir

$$\vec{u}_k \perp \vec{u}_{k+1}$$

Donc , 2 questions :
1) Comment calculer la direction de descente $\mathbf{u_k}$ **?**
2) Connaissant $\mathbf{u_k}$ quelle est la meilleure valeur de la longueur de descente d_k ?

Comment calculer U_k ?

En fait la condition : $\vec{u}_k \perp \vec{u}_{k+1}$

n'est pas suffisante pour déterminer $\mathbf{u_{k+1}}$ en fonction de $\mathbf{u_k}$ en N dimensions :

A 3 dimensions ou plus il y a une *infinité* de vecteurs perpendiculaires…
Donc il nous faut une condition supplémentaire pour déterminer $\mathbf{u_{k+1}}$.

IDEE :
Reprenons le DL de f

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$\approx c - \mathbf{b} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}$$

$$c \equiv f(\mathbf{P}) \qquad \mathbf{b} \equiv -\nabla f|_{\mathbf{P}} \qquad [\mathbf{A}]_{ij} \equiv \left. \frac{\partial^2 f}{\partial x_i \partial x_j} \right|_{\mathbf{P}}$$

Posons : $\vec{X}_{k+1} = \vec{X}_k + d_k \vec{u}_k$

Donc : $F(\vec{X}_{k+1}) = F(\vec{X}_{k+1} + d_k \vec{u}_k) =>$

$$\vec{\nabla} F(\vec{X}_{k+1}) \sim \vec{\nabla} F(\vec{X}_k) + (A\vec{X}_k) \cdot d_k \vec{u}_k$$

$$F(\vec{X}_{k+1}) = F(\vec{X}_{k+1} + d_k \vec{u}_k) =>$$
$$\vec{\nabla} F(\vec{X}_{k+1}) \sim \vec{\nabla} F(\vec{X}_k) + (A\vec{X}_k) \cdot d_k \vec{u}_k$$

Or si $\vec{\nabla} F(\vec{X}_k) \approx 0$

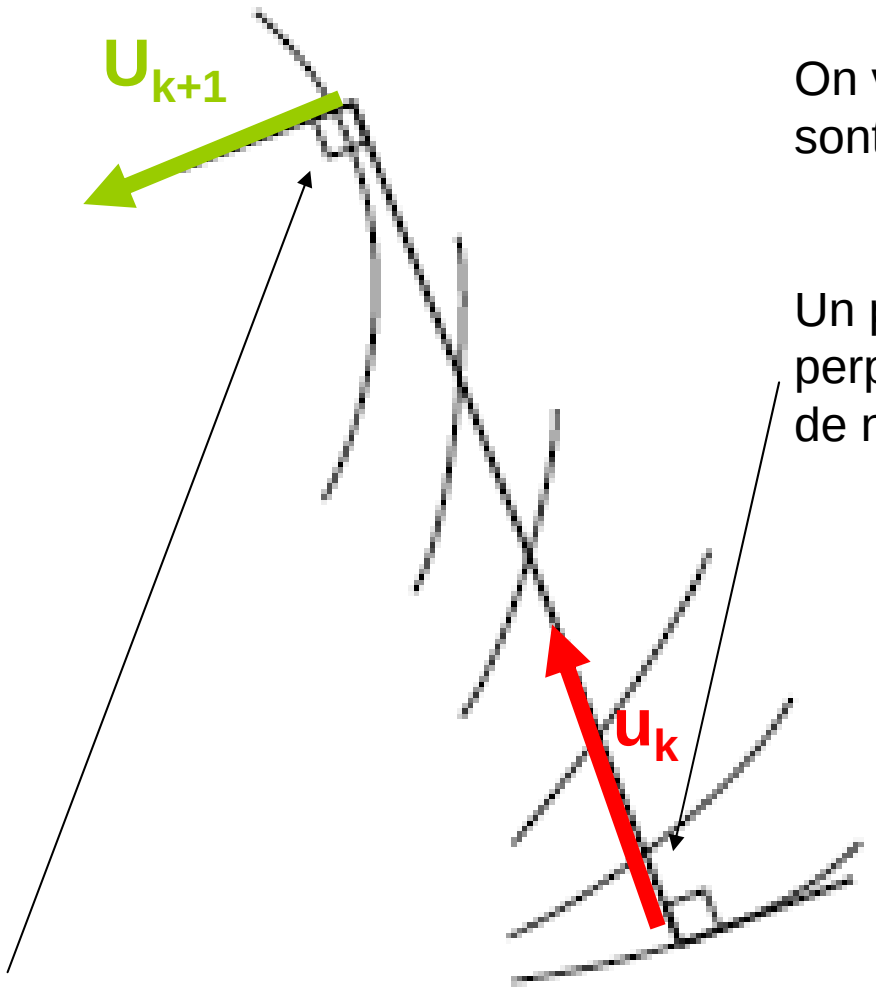
(étape précedente) on veut aussi

$$(A\vec{X}_k) \cdot d_k \vec{u}_k = 0 \Leftrightarrow$$
$$(A\vec{X}_k) \cdot \vec{u}_k = 0$$

« Le produit scalaire de A$\mathbf{X_k}$ avec $\mathbf{u_k}$ est nul»

C'est cela que l'on appelle la forme « conjuguée » :
Les u_k et les X_k sont dit conjugués par pour tout k
$AX_k . u_k = 0$

Voilà donc notre deuxième conditions sur les $\mathbf{u_k}$



On voit que les U_k succéssifs sont perpendiculaires entre eux.

Un pas démarre perpendiculairement aux lignes de niveaux

Le pas finit toujours parallèlement aux ligne de niveau

De manière pratique, comment construire les $\mathbf{U_k}$ **?**

Si U_0 est les gradient de $F(X_0)$ alors les U_k se construisent successivement de la sorte :

$$\vec{u}_k = \vec{\nabla} f(X_k) + \frac{\left\| \vec{\nabla} f(X_k) \right\|}{\left\| \vec{\nabla} f(X_{k-1}) \right\|} \vec{u}_{k-1}$$

Remarque 1 : nous n'avons pas besoin de connaître le HESSIEN pour connaître **u**

Remarque 2 : il faut stocker en mémoire $\mathbf{u_k}$ et $\mathbf{u_{k-1}}$

Comment construire les d_k ?

Ce sont les distances de descente.

La version « classique » de l'algorithme de gradient conjugué utilise le « pas optimal » pour trouver la distance d descente.

$$d_k = \frac{-\vec{\nabla} F(X_k) \cdot \vec{u}_k}{(A\vec{u}_k) \cdot \vec{u}_k}$$

Qui s'obtient facilement en minimisant $F(X_k+d_k u_k)$ en fonction de d_k

Mauvaise nouvelle : Nous avons besoin du Hessien A pour calculer la distance de descente….

Nous verrons plus loin comment faire sans.

Résumé de l'algorithme du gradient conjugué.

0. Choisir un point de départ X_0

1. Construire le nouveau vecteur $\mathbf{u_k}$

$$\vec{u}_k = \vec{\nabla} f(X_k) + \frac{\left\| \vec{\nabla} f(X_k) \right\|}{\left\| \vec{\nabla} f(X_{k-1}) \right\|} \vec{u}_{k-1}$$

2. Calculer la longueur de descente d_k

$$d_k = \frac{-\vec{\nabla} F(X_k) \cdot \vec{u}_k}{(A\vec{u}_k) \cdot \vec{u}_k}$$

3. Calculer le nouveau vecteur X_{k+1}

$$\vec{X}_{k+1} = \vec{X}_k + d_k \vec{u}_k$$

On s'arrête soit quand le gradient devient très petit, soit quand les X_k n'évoluent plus.

Convergence de l'algorithme de gradient conjugué :

On peut montrer que Si f est EXACTEMENT quadratique

$$f(\vec{X}) = c - \vec{B} \cdot \vec{X} + \frac{1}{2} \vec{X} \cdot (A\vec{X})$$

Alors il suffit de N itérations en N dimensions.

En pratique, on étudie rarement des fonctions exactement quadratiques (sinon on connaît analytiquement leur minimum).

Alors que faire si l'algorithme ne converge pas au bout de N itérations ?

Il suffit de le relancer à partir du dernier point obtenu.

Intérêt de la méthode des gradients conjugués :

Elle converge TRES vite vers le minimum :
On peut montrer qu'en N dimensions il lui suffit de N etapes de calculs
Si la fonction est exactement quadratique.

En pratique :
Très souvent utilisée dans les calculs « sérieux » .

Inconvénient :
Nécessite de connaître le Hessien de la fonction F pour déterminer le pas de descente.

L’algorithme de Fletcher-Reeves

Cet algorithme se base sur le Gradient Conjugué

Nous avons vu que l’algorithme de gradient conjugué choisit de manière optimale les directions de descente par le biais des $\mathbf{u_k}$.

Le calcul des $\mathbf{u_k}$ ne fait pas *explicitement* appel aux Hessien

Ensuite on minimise la fonction le long de l’axe $\mathbf{u_k}$ => minimisation à 1D.
et pour le gradient conjugué , la longueur de descente, d_k, se calcule avec le Hessien

Principe de Fletcher Reeves
On reprend les mêmes directions de descente que le gradient conjugué
On minimise le long de l’axe $\mathbf{u_k}$ avec une méthode classique de minimisation à 1D (en utilisant 3 points par exemple.)
Avantage : pas besoin de connaître le Hessien

En pratique : L'algorithme de Fletcher Reeves

0. Commencer d'un point de départ X0

1. Calculer $\mathbf{u_k}$

$$\vec{u}_k = \vec{\nabla} f(X_k) + \frac{\left\| \vec{\nabla} f(X_k) \right\|}{\left\| \vec{\nabla} f(X_{k-1}) \right\|} \vec{u}_{k-1}$$

2. Minimiser la fonction

$$g(d) = F(\vec{X}_k - d\vec{u}_k)$$

d est un réel positif. C'est donc une minimisation à 1D. Utilisez votre méthode préférée (Trisection, quadrature etc…). Le minimum est d_k

3. X_{k+1} est alors $\vec{X}_{k+1} = \vec{X}_k - d_k \vec{u}_k$

4. Si on a pas convergé, recommencer en 1

Convergence de Fletcher-Reeves

Il dépend de votre choix d'algorithme de minimisation à 1D.

Si vous prenez la méthode de quadrature, alors, le comportement de Fletcher-Reeves sera très similaire à celui du gradient conjugué.

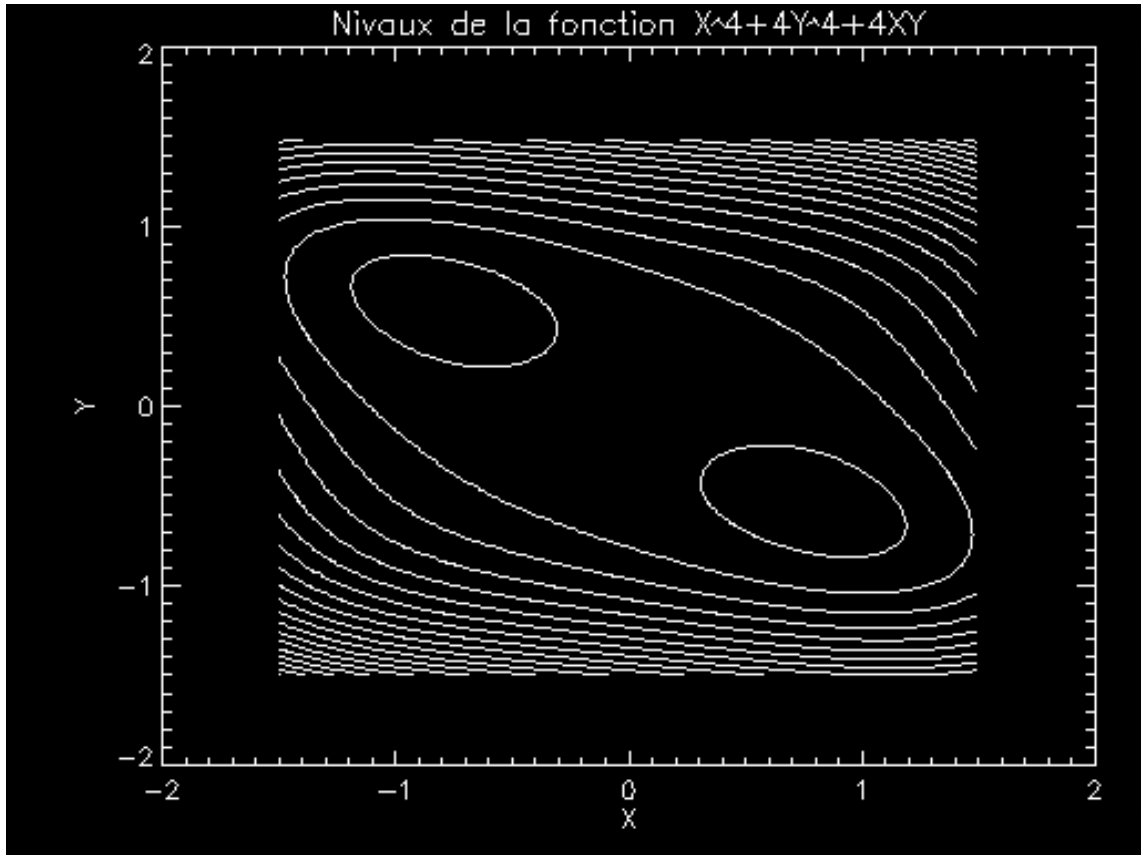
Règle à toujours respecter (pour tous les algorithmes):
Quand on obtient un minimum , il faut *relancer*, l'algorithme depuis ce point (nouveau point de départ) pour être *sur* de la convergence.

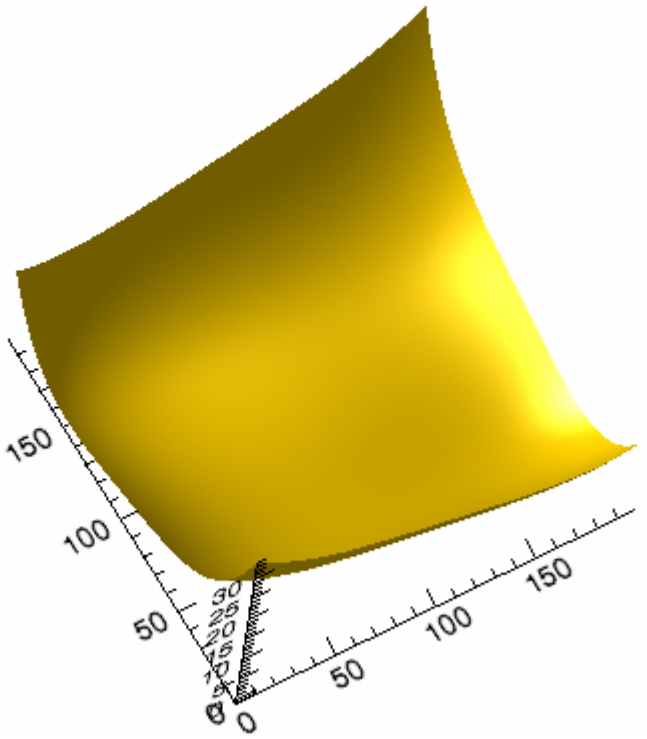
Certains auteurs préconisent même de réinitialiser l'algorithme périodiquement
(⇔ imposer $\mathbf{u_{k-1}=0}$)

Gradient Conjugué Vs. Fletcher Reeves

On minimise la fonction:

$F(x,y)=x^4+4\,y^4+4x\,y$





Qui a deux minimums réels

$X^*1=[\,-8^{1/4}/2,\ 2^{1/4}/2\,]$ $\quad$ $X^*2=[8^{1/4}/2,\ -2^{1/4}/2\,]$

$X^*1=[-0.84,\ 0.59]$ $\quad$ $X^*2=[0.84,\ -0.59]$

gradient conjugué

Fletcher-Reeves, avec
comme méthode de minimisation 1D
la méthode de la « règle d'or ».

N itérations

Initialisation			$(\star)$		$(\star\star)$	
$(0.1, 0.1)$	14	$x_2^\star$	divergence		6	$x_2^\star$
$(0.5, 0.5)$	8	$x_2^\star$	divergence		6	$x_2^\star$
$(1, 1)$	4	$x_2^\star$	6	$x_2^\star$	5	$x_2^\star$
$(1, -1)$	9	$x_2^\star$	7	$x_2^\star$	5	$x_2^\star$
$(10, 10)$	10	$x_1^\star$	9	$x_1^\star$	7	$x_1^\star$
$(10, -10)$	12	$x_1^\star$	9	$x_1^\star$	8	$x_1^\star$
$(100, 100)$	12	$x_1^\star$	10	$x_2^\star$	9	$x_2^\star$
$(100, -100)$	18	$x_1^\star$	10	$x_2^\star$	10	$x_2^\star$
$(1000, 1000)$	16	$x_1^\star$	13	$x_1^\star$	11	$x_1^\star$
$(1000, -1000)$	16	$x_1^\star$	11	$x_1^\star$	12	$x_1^\star$

Remarque 1:
Le nb d'itérations
ne dépend pas
beaucoup de la
distance à la solution

Remarque 2:
Le meilleur est
Fletcher-Reeves
avec relance périodique

*: Pas de relance
** : relance toutes les 2 itérations

Conclusion sur les méthodes déterministes :

Elles sont très efficaces **tant que la fonction étudiée est relativement simple** :
Par exemple : fonction quadratique

Elles convergent vers un UNIQUE minimum :
Elles ne peuvent vous renseigner de manière automatique sur tous les minima de f. Pour trouver plusieurs minima, il faut relancer l'algorithme avec des points de départ différents.

MAIS elles peuvent se révéler inefficaces si :

Si la fonction à étudier est complexe : Nombre de dimensions très important, présence de nombreux minima locaux « parasites »

Ou si vous avez besoin de connaître le minimum GLOBAL de la fonction :

Il faut alors utiliser d'autres méthodes !!!

Méthode pour trouver un minimum global :

« Simulated Annealing » en anglais

« Recuit simulé » en français ….
Appelé aussi « relaxation lente »

Que faire pour trouver un minimum global ?

La méthode la plus utilisée aujourd'hui est celle du « Recuit Simulé »
Cette méthode, très populaire ces dernières années a permis de résoudre des problèmes très complexes du type « voyageur de commerce » où les méthodes déterministes sont rapidement piégées dans des minima locaux.

Nous la décrirons brièvement ici.

C'est une méthode de type « Monte Carlo » donc nécessite des nombres aléatoires.

Elle s'inspire de la thermodynamique…
Méthode élégante.

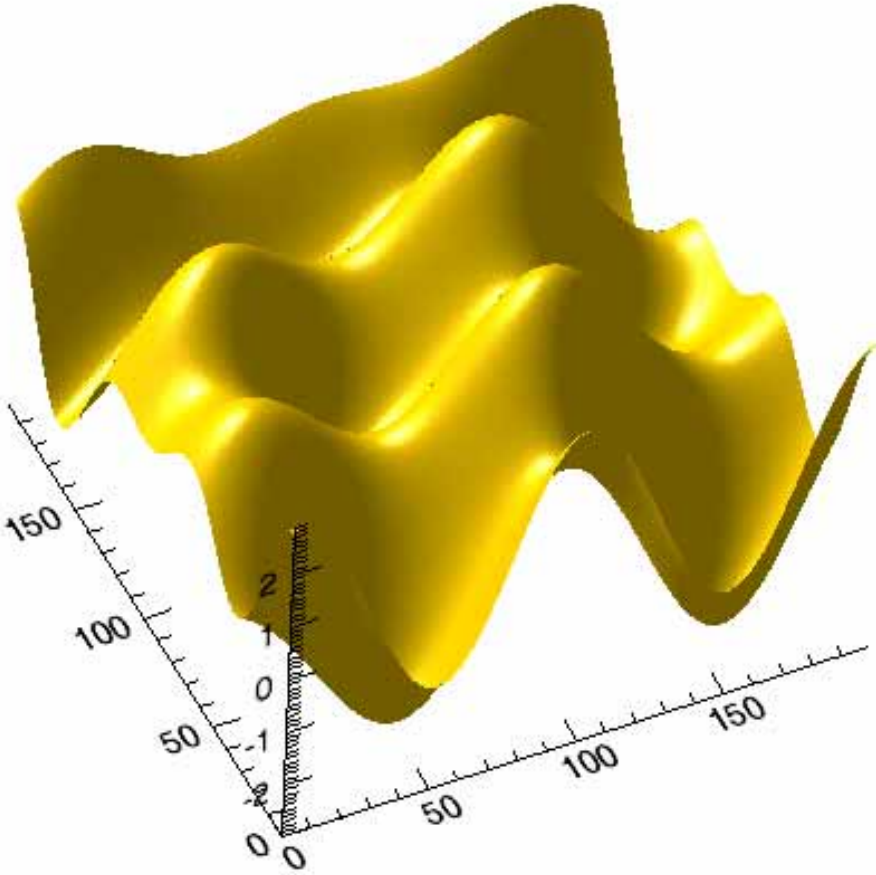
Principe :
Le systèmes naturels évoluent spontanément vers un minimum d' énergie ou maximum d'entropie global :

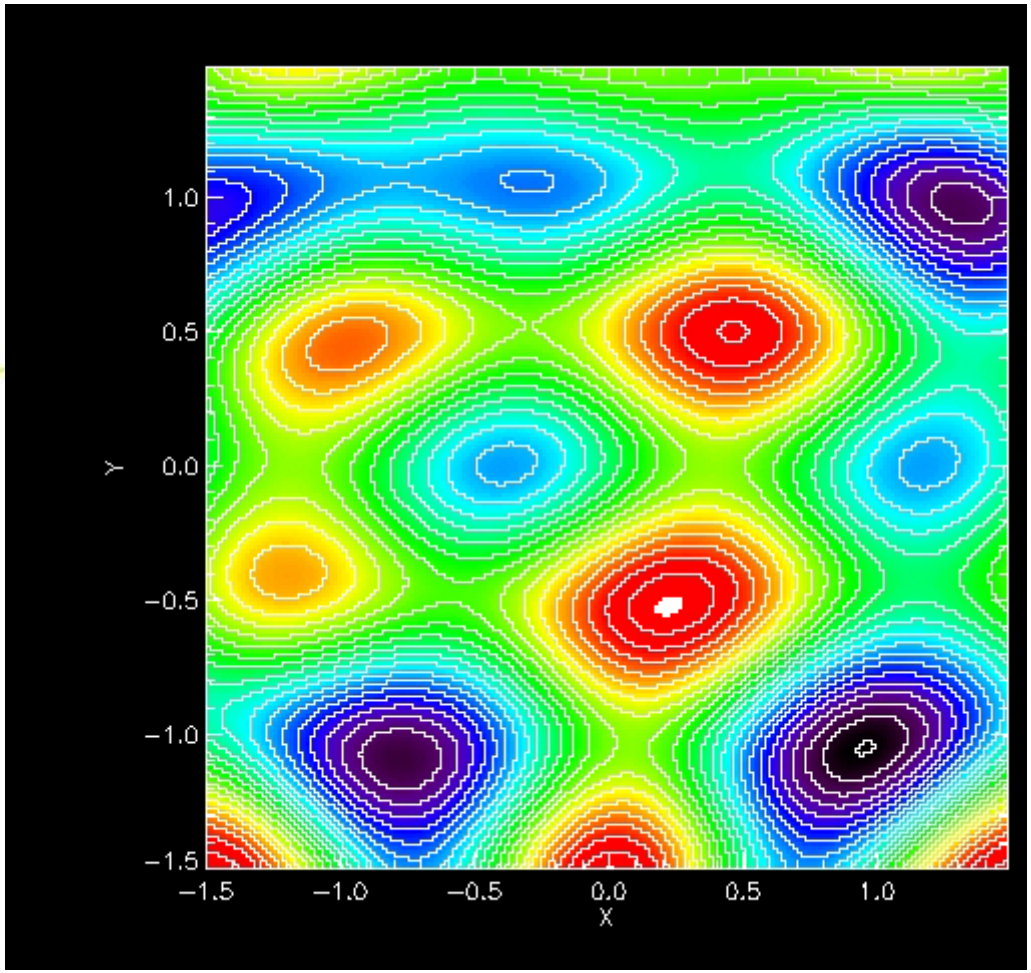
Exemple : gaz (maximum d'entropie)
conformation spatiale d'une molécule (minimum d'énergie)
cristallisation (maximum d'entropie)
etc….

Le système explore toutes les possibilités pendant un temps (relaxation lente) puis se stabilise dans un minimum. C'est un processus de refroidissement.
Plus ce refroidissement est lent, plus le système sera proche du minimum absolu.

La fonction F(X) à minimiser s'appellera la fonction de « coût ».
Il faut donc minimiser le « côut »

Exemple : Voici une fonction F(X,Y) dont il est très difficile de déterminer le minimum absolu. Une méthode déterministe vous donnera juste le minimum local le plus proche





La situation peut-être BIEN pire si votre fonction est en N>2 dimensions !!

Maintenant inspirons nous de la thermodynamique :

En thermodynamique la probabilité d'un état d'énergie E est

$$P(E) \propto e^{-E/kT}$$

Exprimant l'idée que si un système est en équilibre thermique à la température T, alors ses états d'énergie possibles sont distribués de manière probabiliste avec une loi de probabilité qui décroît exponentiellement avec l'énergie.

Donc même si l'état de moindre énergie est le plus probable, il est toujours poss de faire des « sauts » en énergie (avec un faible probabilité) permettant de passe aux travers de minima locaux d'énergie, pour converger finalement vers le minimum global d'énergie.

Donc le système remonte parfois vers des énergies plus grandes pour redescendre vers des énergies plus faibles.
C'est la FORCE de cette approche. Une méthode directe ne peut faire cela

$$P(E) \propto e^{-E/kT}$$

La méthode du recuit-simulé reprend cette probabilité .

•E sera la fonction de coût F : on veut minimiser F

•T sera un paramètre de contrôle fictif : On diminuera lentement la température T du système.

•La constante de Boltzmann K sera remplacée par une constante arbitraire pour que P(E) reste dans des bornes numériques accessibles par la machine.

La probabilité de passer d'un état E_k à l'état E_{k+1} sera :

$$p(E_k \rightarrow E_{k+1}) = e^{-\frac{(E_{k+1}-E_k)}{kT}}$$

Si $E_{k+1} < E_k$ alors P >=1 => la transition sera toujours acceptée. (on minimise !!!)

Si $E_{k+1} > E_k$ alors 0<P<1 => on tire un chiffre X entre 0 et 1. Si X< P alors la transition est acceptée

En pratique, la méthode sera la suivante :

Opération répétée N fois à T constant

1- On part d'un état numéro K, d'énergie Ek, et d'une température T

2- On génère aléatoirement un nouvel état, K+1, proche de l'état K, d'énergie Ek+1 ($\Delta E \sim KT$)

3- On regarde maintenant si on accepte l'état K+1
Pour cela on calcule la probabilité P de passe de K à K+1 :
On tire un chiffre X entre 0 et 1 (distrib. Uniforme)
Si X> P alors on accepte le pas , sinon on le rejette

4- Quand E ne décroît plus, on diminue T et on recommence en 1

Remarque : KT est de l'ordre du ΔE acceptable. Donc plus on diminue KT plus les variations possibles de E seront faibles.

Dans l'étape 2, le générateur aléatoire doit être capable de générer de nouvelles configurations dont le ΔE typique est de l'ordre de KT…
C'est dans l'étape 2 que se cache l'efficacité de la méthode.

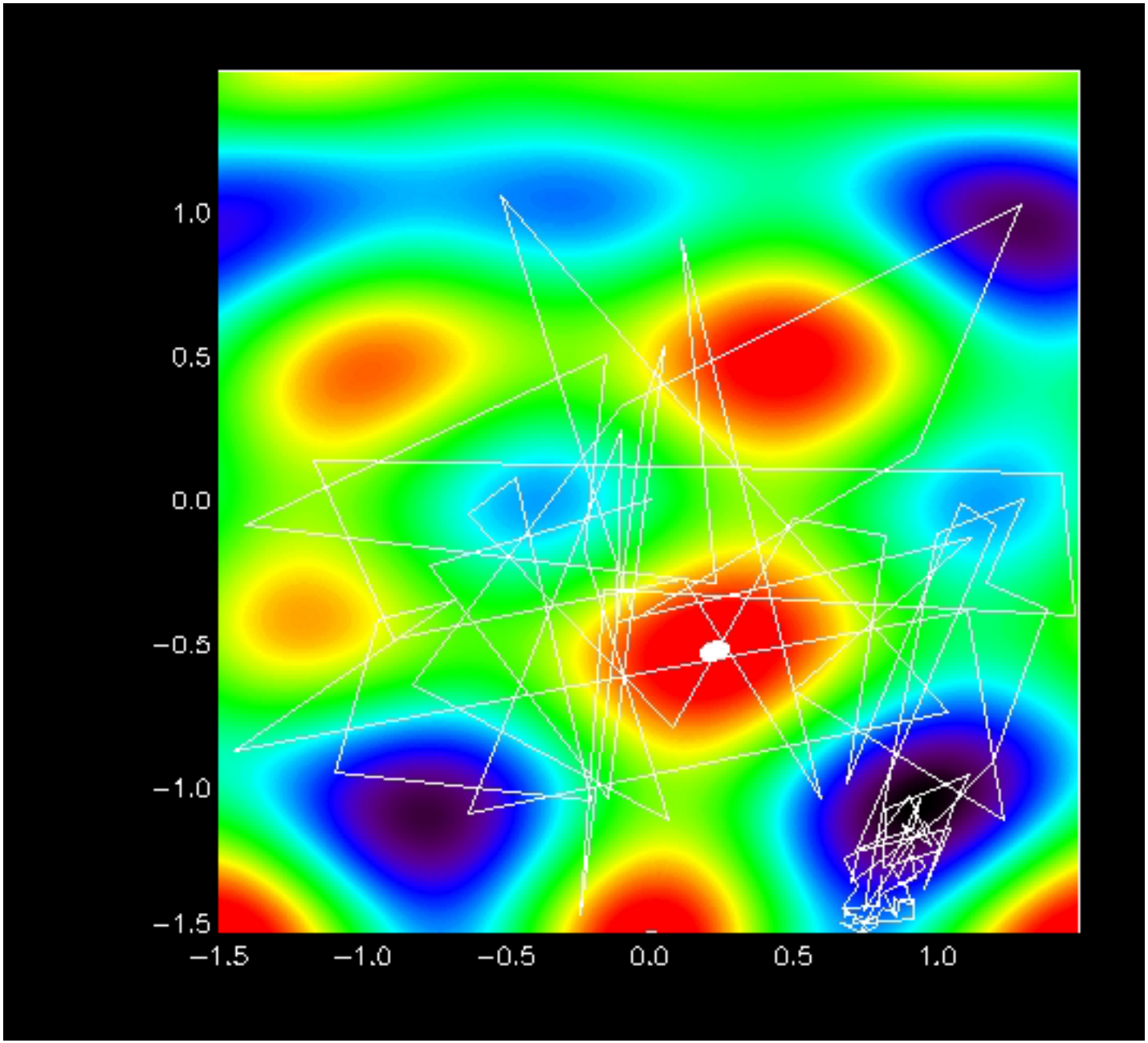
Exemple :

Point de départ:
[0,0]

Point d'arrivée :
[0.68, -1.42]

10^5 itérations

310 changements acceptés



Conclusion sur le recuit simulé :

Méthode qui peut s'avérer efficace quand on cherche un minimum absolu.

Le résultat dépend FORTEMENT de la méthode utilisée pour générer un nouvel état aléatoire.

Très coûteuse en temps calcul => A n'utiliser que si les autres méthodes sont inefficaces.

Le méthode du « recuit simulé » est à la base d'autres méthodes de simulations physiques de type « Monté-Carlo », appelées méthodes de « Métropolis » pour simuler un gaz ou un crystal.

Alternative au recuit simulé : Algorithmes génétiques.

Conclusion sur la minimisation sans contrainte

Le problème est de minimiser une fonction F(X) à N dimensions sans contrainte sur les X.

-Toutes les méthodes nécessitent un point de départ X0.
- Les méthodes déterministes converge vers le *minimum local* le plus proche.
- Plus vous en saurez sur la fonction (Gradient, Hessien) plus la minimisation sera efficace. (gradient conjugué, Fletcher-Reeves)
- Les méthodes à N dim. essayent TOUTES de ramener le pb à 1 dimension : on minimise successivement sur des directions données

Si la fonction est «parasitée » de nombreux minima locaux… alors il faut utilisé une méthode de type « relaxation lente » telle que le « Simulated Anealing ». C'est une méthode Monte Carlo.
Mais le prix à payer est … (1) un temps de calcul très grand
(2) Il faut trouver un générateur de nouveaux états efficace et là il n'y a de « recette pratique »

Minimisation avec Contraintes

Beaucoup de problèmes en physique , en ingénierie et en économie nécessitent de minimiser une fonction mais soums à plusieurs contraintes.

Exemples :

Aéronautique :

Le design optimal d'une aile qui minimise les frottements, mais en sorte que les Contraintes sur l'aile ne soient pas trop fortes.

Economie :

Prendre une décision économique qui optimise une situation , mais à budget constant … par exemple ..

Chimie :

Composition chimique à l'équilibre, qui minimise l'énergie de Gibbs mais qui conserve les masse totale

Etc…

Plusieurs techniques existent.

Nous n’en verrons qu’une seule ici , que l’on rencontre souvent en thermodynamique :

La méthode des multiplicateurs de Lagrange.

Méthode qui peut être aussi utilisée analytiquement.

La méthode des multiplicateurs de Lagrange permet de résoudre le problème suivant :

Soient **X**=[x1,x2, …,xn] un ensemble de N variables

Nous voulons miniser la fonction f(**X**) sachant que **X** est soumis à la contrainte suivante : g(X)=0

Exemple (tiré de l'économie) :

Nous voulons maximiser la fonction :

$$f(x,y) = x^{2/3} y^{1/3}$$

X et Y et y représentent l'investissement en travail (x) et en capital (y)

Sachant que le coup total de l'opération est

$$g(x,y) = p_1 x + p_2 y = c$$

S’il n’y avait pas de contrainte , il suffirait d’augmenter x et y indéfiniment.

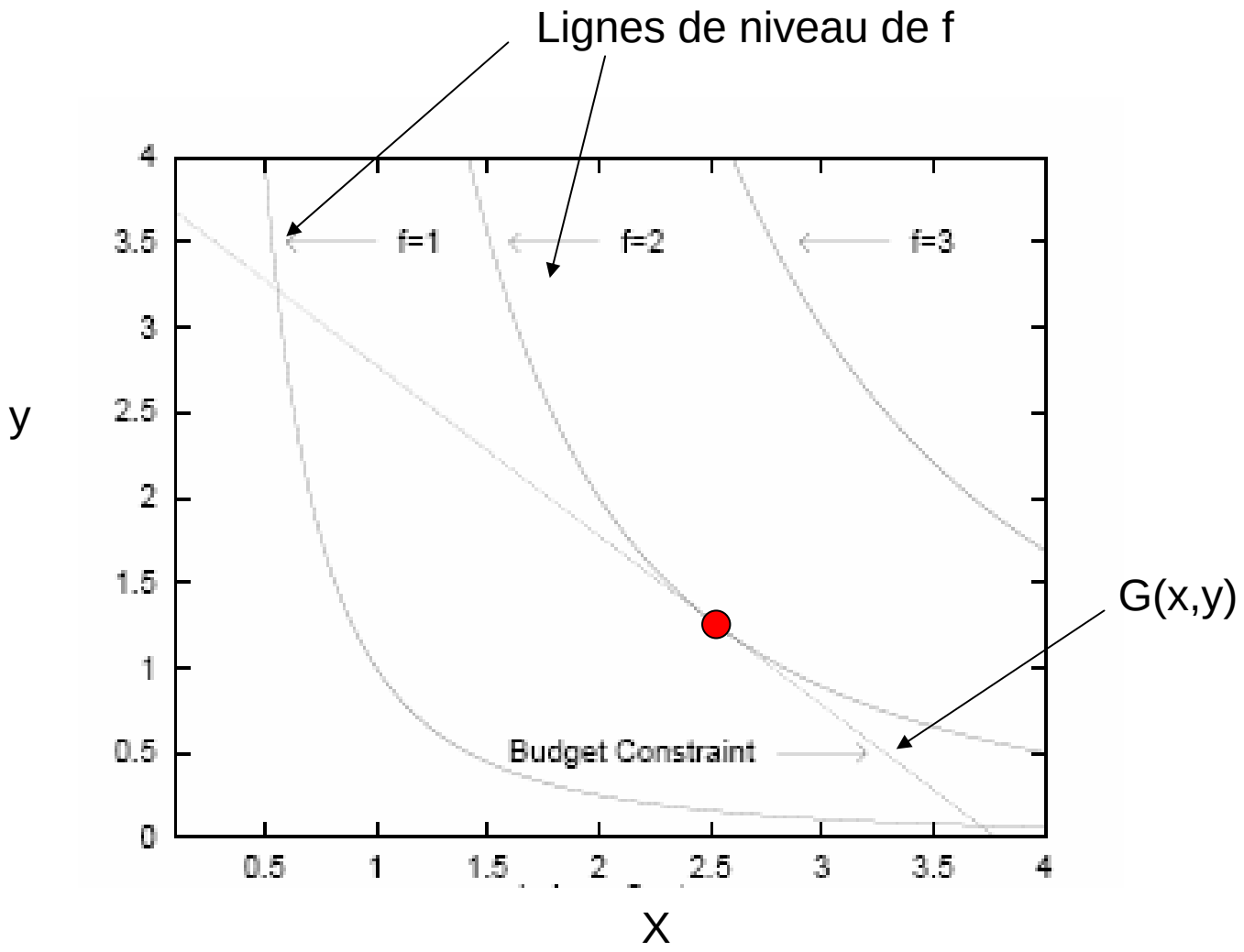
Mais la contrainte G(x,y)=c rend cela impossible. Il font donc maximiser f(x,y) avec la contrainte G(x,y)=c

On est typiquement dans un problème d’optimisation contraint.

Comment résoudre ce problème ?

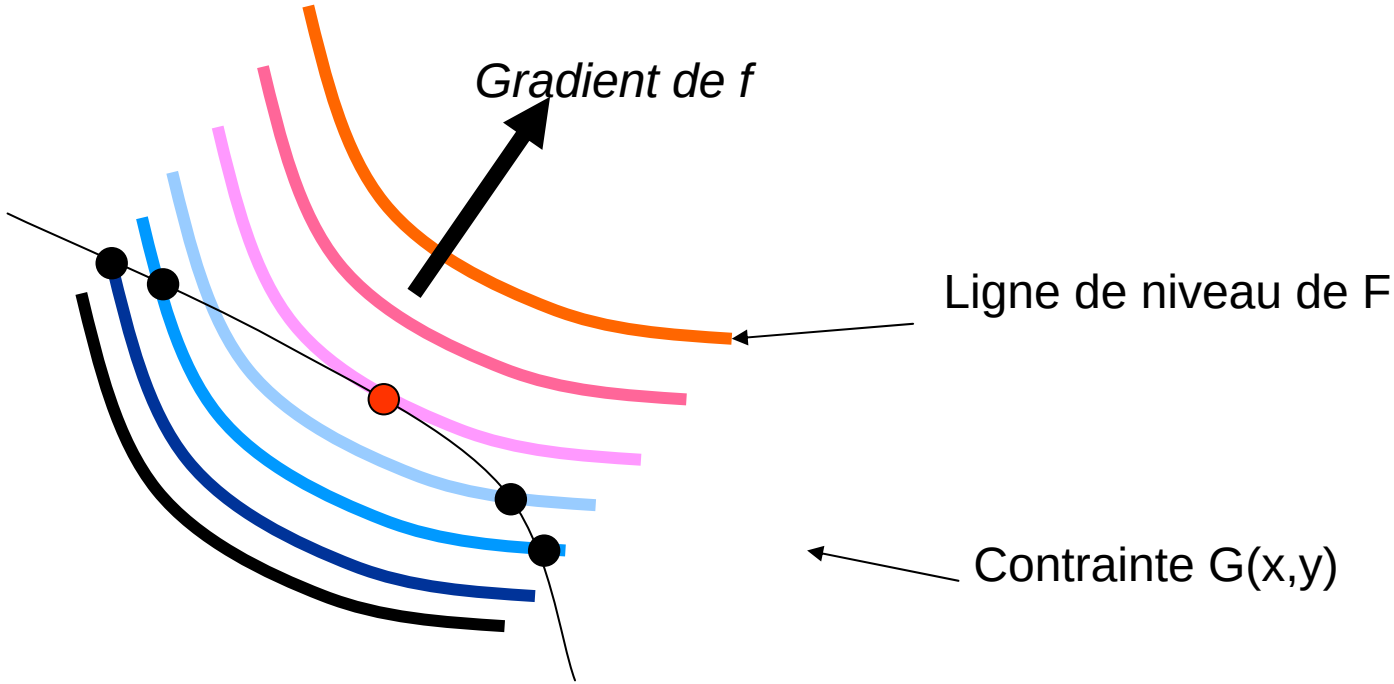
Regardons les lignes de niveaux de F et de G

On prendra pour G(x,y)=c $\Leftrightarrow$ x+y=3.78 …



On remarque qu'au niveau de la solution possible (rouge) la ligne de niveau de f
Est parallèle à la ligne de niveau de g.
A ce point là $\vec{\nabla} f = k\vec{\nabla} G$

Démonstration graphique



- : On peut augmenter f encore, en se deplaçant à droite ou à gauche
- : On ne peut augmenter f : les directions droite et gauche minimisent f

On voit donc que la condition de maximum s'écrit :

« le gradient de f est // au gradient de G »

$$\vec{\nabla} f = \lambda \vec{\nabla} G \, , \lambda \in \Re$$

$$\Leftrightarrow$$

$$\begin{cases} \dfrac{\partial f}{\partial x} - \lambda \dfrac{\partial g}{\partial x} = 0 \\ \dfrac{\partial f}{\partial y} - \lambda \dfrac{\partial g}{\partial y} = 0 \\ \dfrac{\partial f}{\partial z} - \lambda \dfrac{\partial g}{\partial z} = 0 \end{cases}$$

λ est appelé
« Multiplicateur de Lagrange »

N équations (Tous les gradients)

à N inconnues (x,y,z etc…)

+ 1 équation : G(x,y)=0 +1 inconnue :λ

Donc N+1 équations a N+1 inconnues

Ce système se résoud ensuite numériquement par les méthodes que l'on a vu en cours…

Résumé : On maximise F(X) avec la contrainte G(X)=0. X est un vecteur à N composantes

1- Poser H(X,λ)= F(x)-λG(X)

2- Ecrire $\forall i \quad \frac{\partial f}{\partial x_i} - \lambda \frac{\partial g}{\partial x_i} = 0$

3. Trouver les x_i et λ solutions des N équations du gradient + de l'équation G(X)=0

Exemple :
On veut produire des boîtes de conserve cylindrique de volume V0 fixé mais
De surface minimale S. Les deux paramètres sont r et h

Fonction à minimiser

$$S(r,h) = 2\pi rh + 2\pi r^2$$

Contraintes

$$V(r,h) = V_0 = \pi r^2 h$$

$$H = S(r,h) - \lambda V(r,h)$$

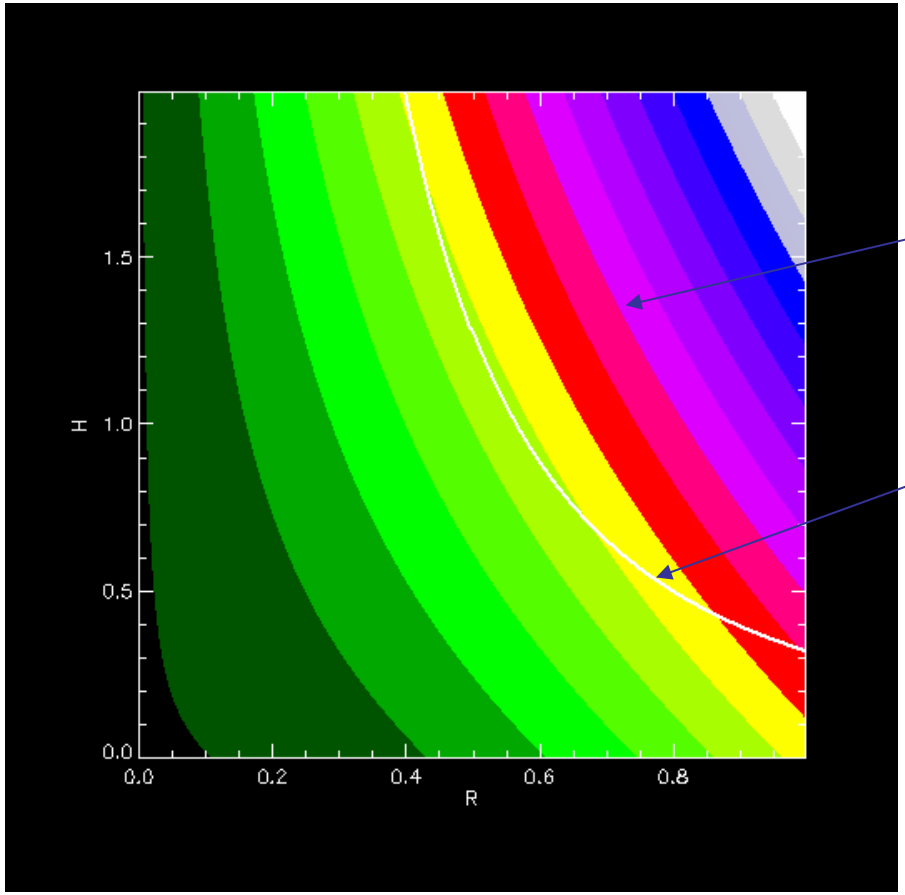
$$\Rightarrow$$

$$\begin{cases} \dfrac{\partial S}{\partial r} - \lambda \dfrac{\partial V}{\partial r} = 0 \Leftrightarrow 2\pi h + 4\pi r - \lambda 2\pi rh = 0 \\ \dfrac{\partial S}{\partial h} - \lambda \dfrac{\partial V}{\partial h} = 0 \Leftrightarrow 2\pi r - \lambda \pi r^2 = 0 \\ \pi r^2 h = V_0 \end{cases}$$

3 équations
3 inconnues : r,h,λ

$$\begin{cases} 2\pi h + 4\pi r - \lambda 2\pi r h = 0 \\ 2\pi r - \lambda \pi r^2 = 0 \\ \pi r^2 h = V_0 \end{cases}$$

Système d'équations non linéaire
qui se résout par substitution

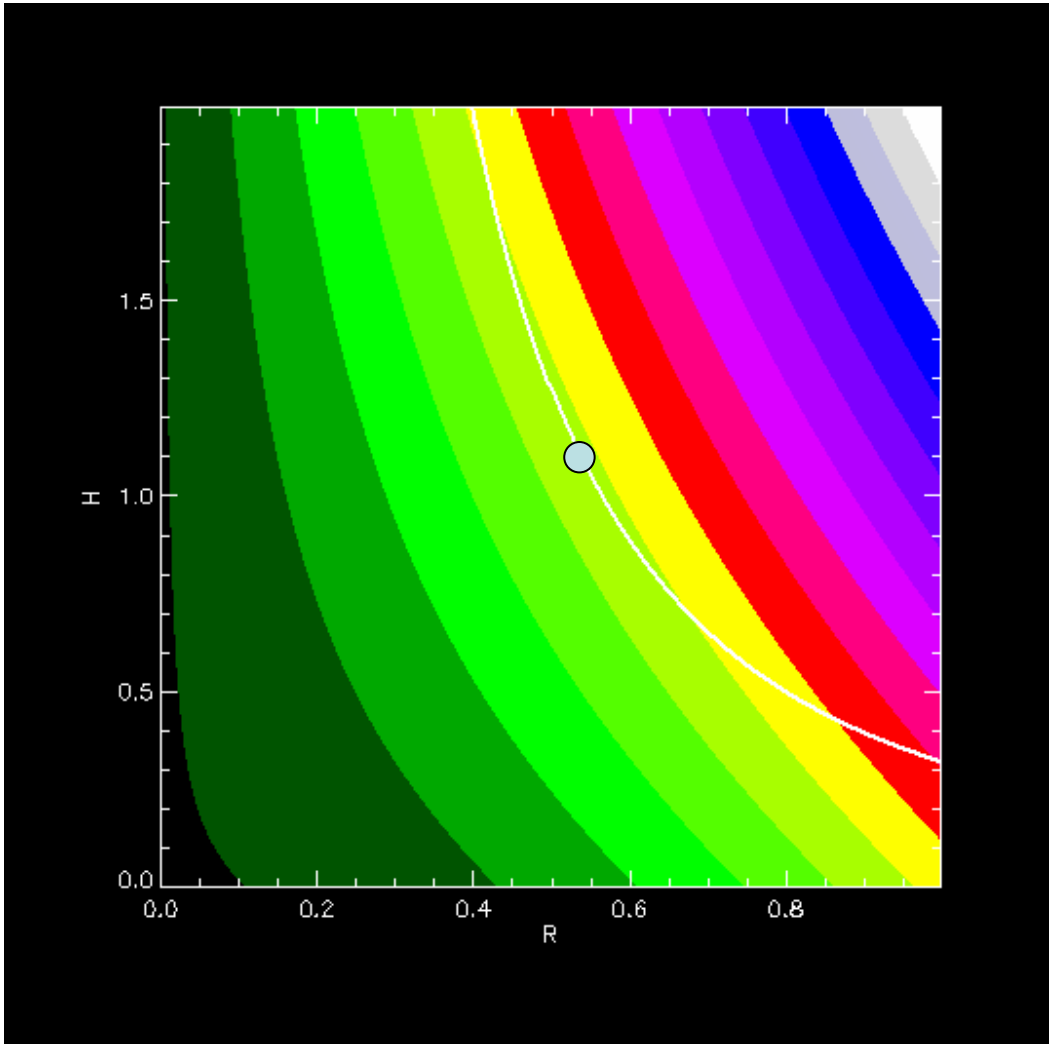


Couleurs : Lignes de niveaux
de S à maximiser

Ligne : Contrainte Pi r^2 $h=V_0=1$

$$\begin{cases} 2\pi h + 4\pi r - \lambda 2\pi r h = 0 \\ 2\pi r - \lambda \pi r^2 = 0 \\ \pi r^2 h = V_0 \end{cases} \Rightarrow \begin{cases} 2\pi h + 4\pi r - \lambda 2\pi r h = 0 \\ \lambda = 2/r \\ \pi r^2 h = V_0 \end{cases}$$

$$\begin{cases} h = 2r \\ \lambda = 2/r \\ \pi r^2 h = V_0 \end{cases} \Rightarrow \begin{cases} h = 2r \\ \lambda = 2/r \\ r = \left(\dfrac{V_0}{2\pi} \right)^{1/3} \end{cases}$$



$$\begin{cases} h = 1.08 \\ \lambda = 3.69 \\ r = 0.54 \end{cases}$$

Pour V=1

Que faire quand nous avons plus d'une contrainte ?

On introduit autant de multiplicateurs de Lagrange que de contraintes :

Exemple : **N** variables et **M** contraintes

Soient x1, …, Xn N variables.

On veut maximiser F(x1,…,xN)

Avec M contraintes du type : $G_i(x1,\ldots,Xn)=0$, pour i=1…M

Methode

On introduit M multiplicateurs de Lagrange : $\lambda_1, \ldots, \lambda_M$

Et n résout les M équations $$\frac{\partial f}{\partial x_i} = \sum_{j=1}^{M} \lambda_j \frac{\partial G_i}{\partial x_i}$$

+ Les M contraintes. Cela permet de trouver les λ_i et tous les x_i

La méthode des multiplicateurs de Lagrange est la méthode la plus simple pour optimiser un problème avec contraintes.

Le paramètre λ n'a pas de réelle signification ici.

D'autres méthodes existent que nous ne détaillerons pas ici.

Elle est très utilisée en thermodynamique où , par exemple, elle permet de calculer la composition chimique à l'équilibre d'une réaction chimique

En résumé : Comment s'attaquer à un problème de minimisation

Soit à Minimiser la fonction F(x1, …, Xn) à N variables.

1. **Le problème est-il contraint ou non ?**
   Si oui : Multiplicateurs de Lagrange ou autre

2. Le problèmes est-il simple ? (idée du minimum, absence de nombreux minima parasites)

Si oui alors on peut utiliser toutes les méthodes déterministes

3. Problème simple : Connaissez vous le gradient de F ?

Si oui, vous êtes dans le meilleur cas : utilisez un gradient conjugué par exemple (Hessien), ou Fletcher Reeves

Si non et N=1 : Utilisez une méthode de trisection ou de quadrature
Si non et N>1 : Seule la méthode « Amibe » (« Amoeba ») non étudiée ici peut vous aider…

4. Si le problème est complexe (Pas d'idée du minimum, nombreux minima parasites etc…)

Vous êtes dans le cas le plus difficile. Tentez une méthode Monte-Carlo telle que le recuit-simulé (« Simulated Annealing »)