M1 BBS - EM8BBSEM

Simulation de Systèmes Biologiques

(#3)

Georges Czaplicki, UPS / IPBS-CNRS
Tél. : 05.61.17.54.04, email : cgeorge@ipbs.fr

Système positionnel de numération

$$\boxed{123 = ?}$$

$$123 = \boxed{} \cdot 10^2 + \boxed{} \cdot 10^1 + \boxed{} \cdot 10^0$$

$$Valeur = \sum_{i=1}^{n} chiffre \cdot \left(Base\right)^{n-i}$$

$$chiffre = \left\{ 0, 1, 2, \ldots, Base - 1 \right\}$$

$$123_5 = 1 \cdot 5^2 + 2 \cdot 5^1 + 3 \cdot 5^0 = 25 + 10 + 3 = 38$$

$$123_8 = 1 \cdot 8^2 + 2 \cdot 8^1 + 3 \cdot 8^0 = 64 + 16 + 3 = 83$$

Systèmes utiles : binaire, octal, hexadécimal

B=2 : { 0, 1 } | B=8 : {0,1,2,3,4,5,6,7} | B=16 : {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

1 bit :

0		0
1		1

$2^1=2$

2 bits :

0	0		0
0	1		1
1	0		2
1	1		3

$2^2 = 4$

3 bits :

0	0	0		0
0	0	1		1
0	1	0		2
0	1	1		3
1	0	0		4
1	0	1		5
1	1	0		6
1	1	1		7

$2^3 = 8$

4 bits :

0	0	0	0		0
0	0	0	1		1
0	0	1	0		2
...	...	...	...		
1	0	0	0		4
1	0	0	1		5
...	...	...	...		
1	1	1	1		15

$2^4 = 16$

8 bits = 1 octet

0	0	0	0	0	0	0	0		0
0	0	0	0	0	0	0	1		1
...	...	...	...	...	...	...	...		
0	1	0	0	0	0	0	0		64
...	...	...	...	...	...	...	...		
1	0	0	0	0	0	0	0		128
...	...	...	...	...	...	...	...		
1	1	1	1	1	1	1	1		255

$2^8 = 256$

$$\boxed{123_{10} = 1111011_2 = 173_8 = 7B_{16}}$$

Précision, erreurs d’arrondi, stabilité…

this bit could be “phantom”

	sign bit	8-bit exponent	23-bit mantissa
$\frac{1}{2}$ =	0	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 =	0	1 0 0 0 0 0 1 0	1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$\frac{1}{4}$ =	0	0 1 1 1 1 1 1 1	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10^{-7} =	0	0 1 1 0 1 0 0 1	1 1 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 0 1 0
=	0	1 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$3 + 10^{-7}$ =	0	1 0 0 0 0 0 1 0	1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

$$s \times M \times B^{e-E}$$

1. représentation interne de nombres (entiers, virgule flottante)
2. précision de la machine (liée avec la mantisse)
3. le moins grand nombre représentable (liée avec l’exposant)
4. arrondissage (si cumulable, calcul instable)

IEEE 754

Exemple :

$$r = \sqrt{x^2 + y^2}$$ **pour éviter le dépassement des registres :** $$x^2 \le M$$ $$x \le \sqrt{M} = M^{\frac{1}{2}} \xrightarrow{32bits} 2^{\frac{32}{2}} = 2^{16}$$	$$x > y \quad \Rightarrow \quad r = \sqrt{x^2\left(1 + \left(\frac{y}{x}\right)^2\right)} = x\sqrt{1 + \left(\frac{y}{x}\right)^2}$$ **le gain est donné par :** $$M = 2^n \quad \Rightarrow \quad n = \frac{\ln M}{\ln 2}$$ $$M \to \frac{M}{\sqrt{2}} \quad \Rightarrow \quad n = 31.5$$

How Java's Floating-Point Hurts Everyone Everywhere

by
Prof. W. Kahan and Joseph D. Darcy
Elect. Eng. & Computer Science
Univ. of Calif. @ Berkeley

Originally presented 1 March 1998
at the invitation of the
ACM 1998 Workshop on Java for
High–Performance Network Computing
held at Stanford University

http://www.cs.ucsb.edu/conferences/java98

This document: http://www.cs.berkeley.edu/~wkahan/JAVAhurt.pdf or
http://www.cs.berkeley.edu/~darcy/JAVAhurt.pdf

L’explosion de la fusée Ariane 5 (le 4 juin 1996)

- Après 37 secondes de vol, les fortes accélérations de la fusée provoquent un dépassement de capacité dans le calculateur du système de guidage inertiel principal, qui se met aussitôt hors service.

- Le système de guidage de secours (identique à l'autre) subit la même avarie, et s'arrête à la même seconde.

- Le pilote automatique se met en route. Suite à une mauvaise interprétation du signal de panne des deux guidages inertiels hors service, le pilote automatique provoque une violente correction de trajectoire. La fusée dérape de sa trajectoire, et les boosters sont arrachés par le courant d'air décentré. Ce qui déclenche le mécanisme d'autodestruction préventive de la fusée.

L’échec du missile *Patriot* (le 25/02/1991, guerre du Golfe)

Un missile américain *Patriot* à Dharan (Arabie Saoudite) a échoué de détruire un missile iraquien *Scud*. Ce dernier a frappé les casernes de l’armée américaine, tuant 28 soldats et blessant ~100 personnes.

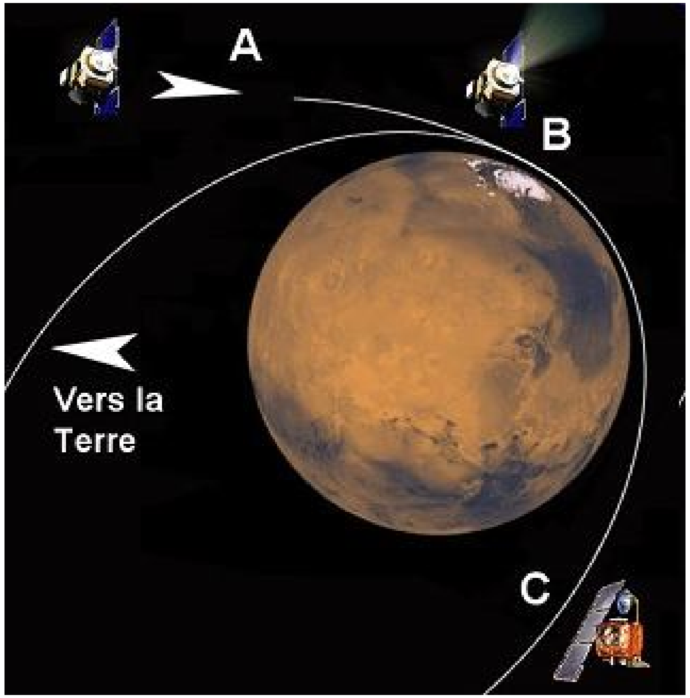
Les résultats d’enquête ont été publiés dans le rapport *Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia*. Il s’avère que le problème venait du calcul de temps écoulé depuis le démarrage du système, qui montrait les erreurs d’arrondi.

Concrètement, l’horloge interne mesurait le temps en dixièmes de secondes, ce qui était multiplié par 10 pour arriver aux secondes. Le calcul se faisait sur des registres de 24 bits. Puisque la valeur 1/10 en système binaire va au-delà de 24 bits, ce qui dépassait était coupé. La petite erreur d’arrondi, multipliée par des grandes chiffres représentant le temps en 1/10 s, conduisait de façon cumulative à une grande erreur finale. L’erreur d’arrondi sur chaque mesure était d’ordre de 0.1μs, mais sur 100h après le démarrage, ceci donne
$0.095 \cdot 10^{-6} \cdot 100 \cdot 60 \cdot 60 \cdot 10 = 0.342$ s.

Un missile *Scud* a la vitesse de 1676 m/s, ce qui veut dire qu’il parcourt presque 600 m pendant ce temps (0.342 s). Par conséquent, il était au-dehors du *tracking range* (champ de poursuite) du missile *Patriot*.

Perte de la sonde spatiale *Mars Climate Orbiter* (le 23/09/1999)

Le 23/09/1999 débutait la mise en orbite de la sonde *Mars Climate Orbiter*. Pour placer la sonde en orbite martienne, il est nécessaire de la ralentir suffisamment pour qu'elle puisse être happée par le champ de gravité de Mars. Si le freinage n'est pas assez fort, la sonde survole Mars puis dépasse la planète en continuant sur sa lancée. Le moteur devait fonctionner pendant ~16 minutes pour assurer un freinage suffisant. A 11:01, alors que la sonde est en train de frôler le pôle nord martien, le moteur de 640 Newtons de poussée est mis à feu. La sonde devait réapparaître de l'autre côté de Mars à 11:26:25, mais l'antenne n'a capté aucun signal. A 11:41, la NASA annonce officiellement que le contact avec la sonde est perdu.



Il semble que la perte de *Mars Climate Orbiter* doit être mise sur le compte d'un problème d'unité dans l'expression d'une force de poussée. Les ingénieurs de *Lockheed Martin Astronautics* (Denver, Colorado), la firme qui a conçu et fabriqué la sonde martienne, avaient apparemment gardé l'habitude de travailler avec les unités du système anglo-saxons (poussée du moteur en livres). De leur côté, les ingénieurs du *Jet Propulsion Laboratory* (Pasadena, Californie) travaillaient depuis des années dans le système métrique (poussée calculée en Newtons ; une livre équivaut à 4,48 Newtons). lors du transfert des données entre le centre de Lockheed et celui du JPL, personne ne se soit rendu compte qu'il fallait convertir les données, chacun étant persuadé que l'un utilisait les mêmes unités que l'autre !

Calcul numérique (dérivées, intégrales, équations différentielles,…)

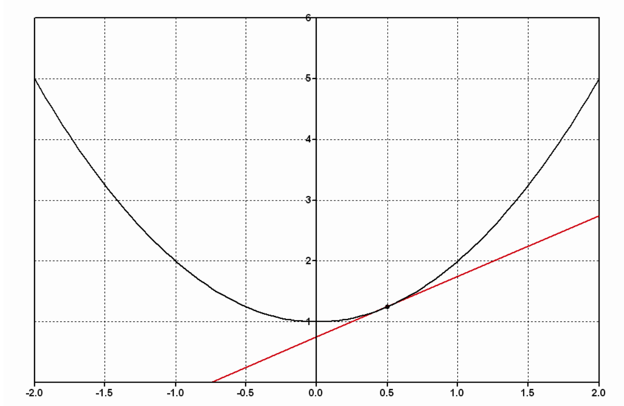
Développements de Taylor :

$$f(x+h) = f(x) + \frac{df}{dx}h + \frac{1}{2}\frac{d^2 f}{dx^2}h^2 + O(h^3)$$

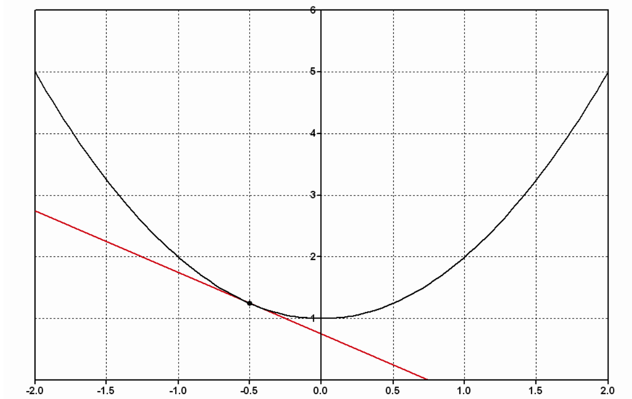
$$f(x-h) = f(x) - \frac{df}{dx}h + \frac{1}{2}\frac{d^2 f}{dx^2}h^2 + O(h^3)$$

Schémas numériques (avant, arrière, centré) :

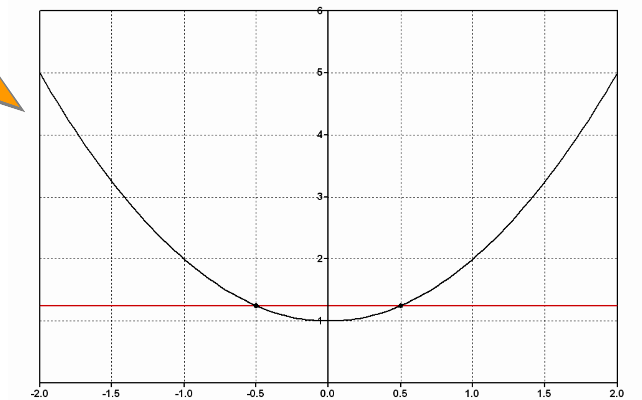
$$f'(x) \approx \frac{f(x+h) - f(x)}{h} + O(h)$$



$$f'(x) \approx \frac{f(x) - f(x-h)}{h} + O(h)$$



$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

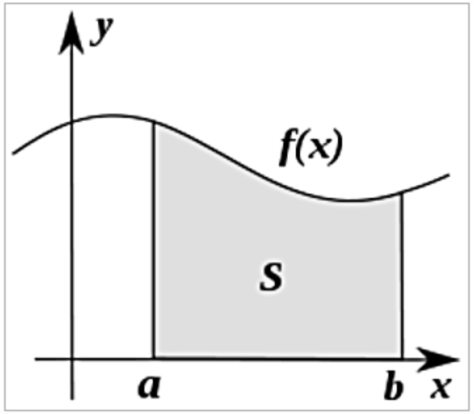


h grand => calcul peu précis
h petit => erreurs d'arrondi

Calcul direct de dérivées :

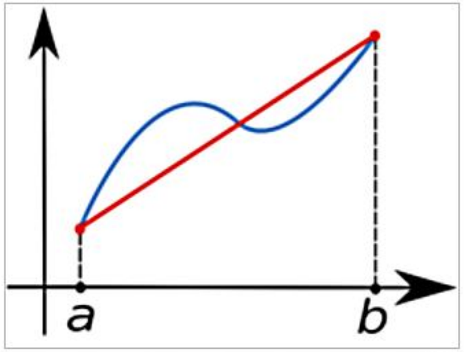
$$f'(x_k) \approx \frac{f(x_{k+1}) - f(x_{k-1})}{x_{k+1} - x_{k-1}}$$

Intégration numérique : comment trouver $\int_a^b f(x)dx$?



Numerical integration consists of finding numerical approximations for the value S

Méthode de trapèzes



The function $f(x)$ (in blue) is approximated by a linear function (in red).

Intervalle unique :

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2}$$

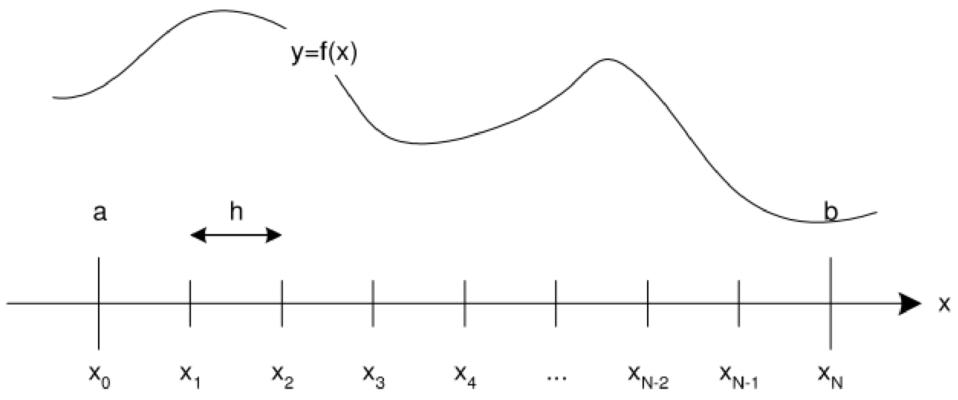


Figure 5.1: Partition of $[a,b]$ into N equal pieces of length h

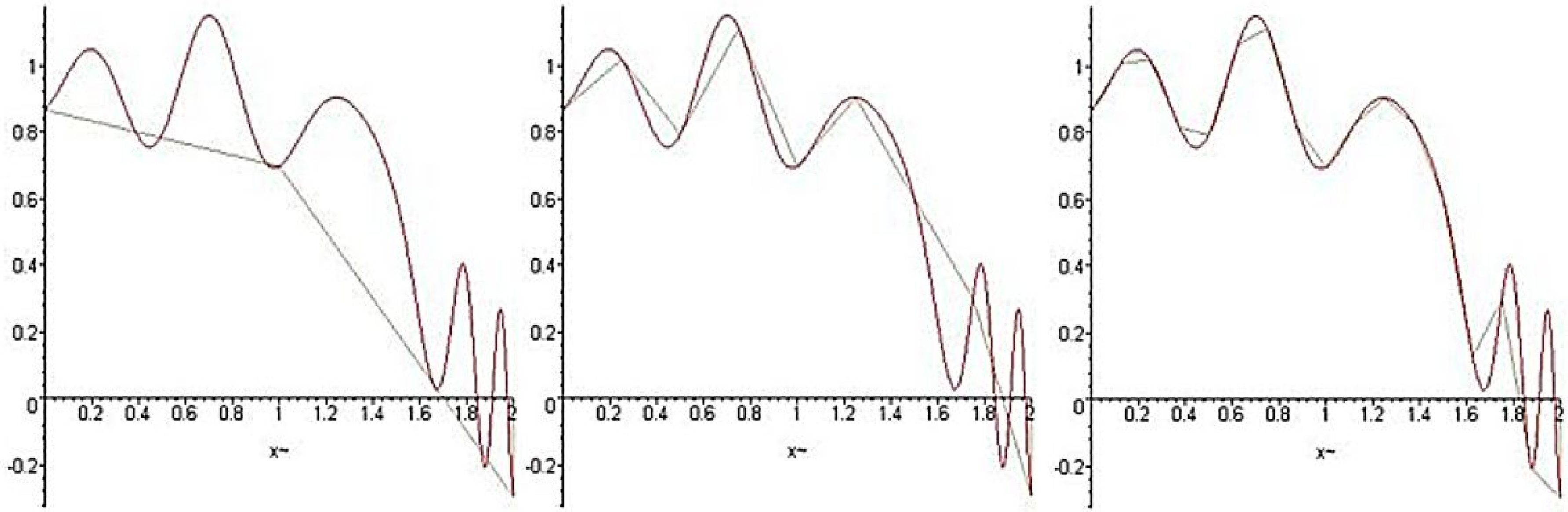
Intervalles multiples :

$$\int_a^b f(x)dx \approx \frac{b-a}{n}\left[\frac{f(a)+f(b)}{2}+\sum_{k=1}^{n-1} f\left(a+k\frac{b-a}{n}\right)\right]$$

Exemple d’intégration numérique

$$= 1.1 + \ln\left(e - \frac{x}{100} + \frac{3}{5}\tanh(\ln(x + 10^{-7}) + 1)\right)\cos\, x + \frac{2}{5}\left(x - \frac{\cos(3x)}{5}\right)^2$$
$$\cdots + \frac{11}{100}\sqrt{2 + 2x}\,\sin\left(\frac{44}{25}(4 + 3\sqrt{x})x - \frac{19}{20}x^5\right) - e^{\frac{x}{3}}$$

Découpage pour différentes valeurs de *n* (2,8 et 16).



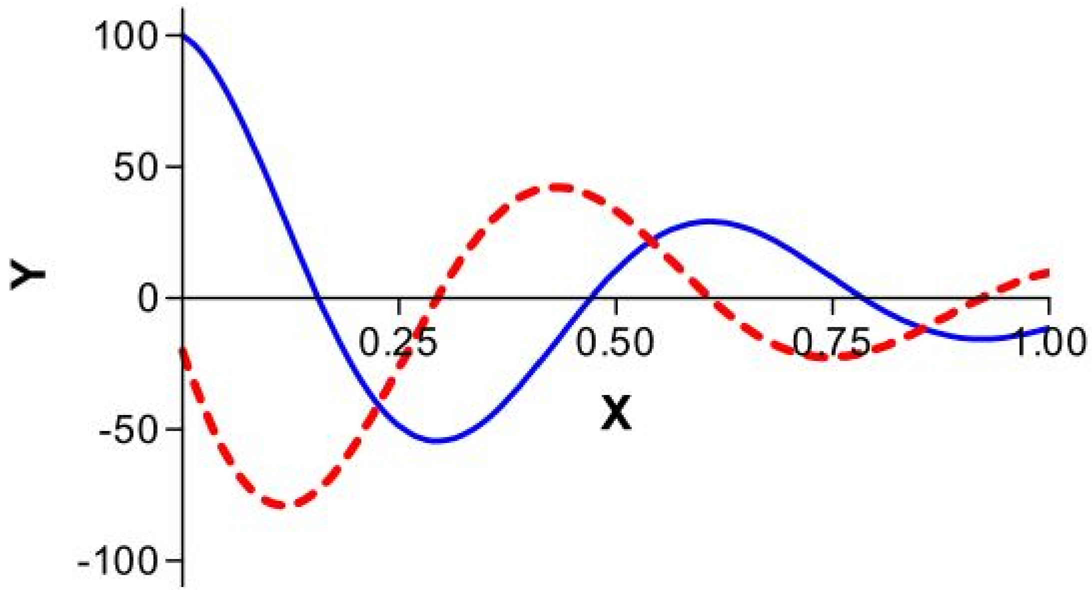
Résolution des équations numériques

La dérivée possède les propriétés suivantes :

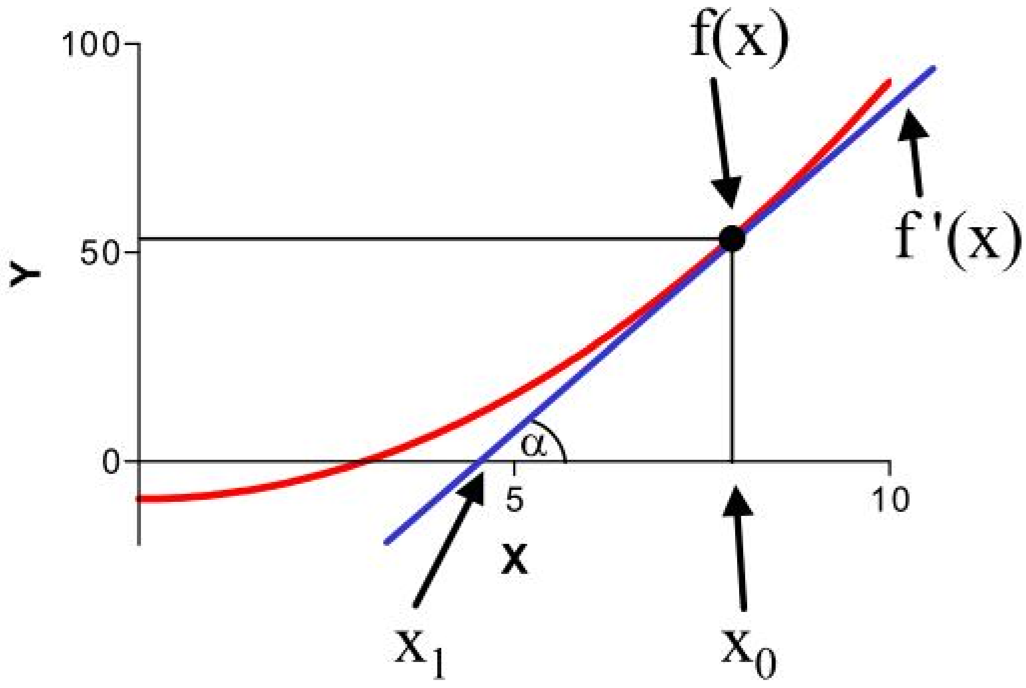
$$f'(x) = \frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

fonction	dérivée
croissante	positive
décroissante	négative
maximum ou minimum	zéro

fonction et sa dérivée



La recherche du passage par 0 : méthode de Newton

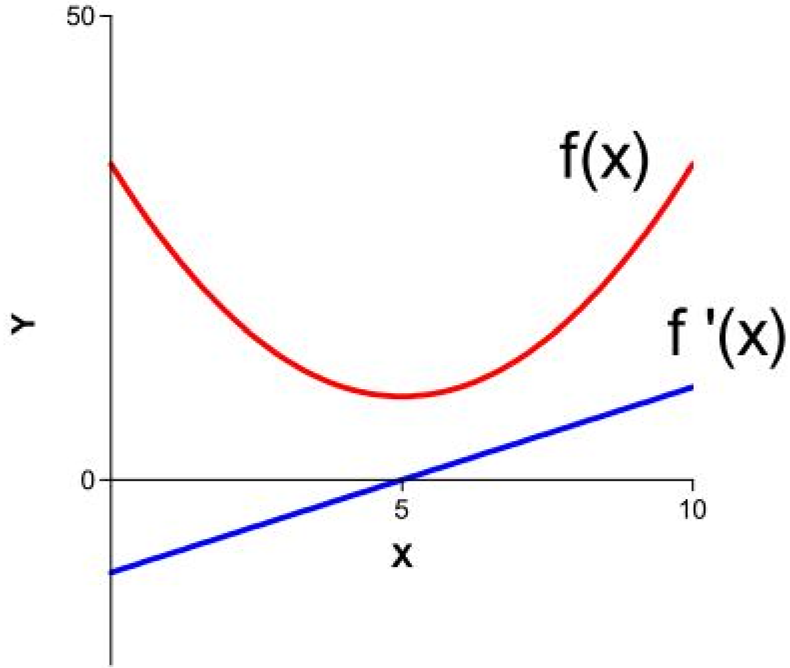


$$f\,'(x_0) = \tan\alpha = \frac{f(x_0) - 0}{x_0 - x_1}$$

$$x_1 = x_0 - \frac{f(x_0)}{f\,'(x_0)}$$

La méthode de Newton peut être utilisée pour trouver le minimum d’une fonction :

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$



$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$$

Application de la méthode de Newton aux algorithmes numériques

Exemple : comment calculer la racine carrée de A ?
Solution : Créer une fonction dont le passage par zéro donne la solution du problème…

$$x = \sqrt{A}$$
$$x - \sqrt{A} = 0$$

…sous la condition que la fonction soit correctement créée !

$$x = \sqrt{A}$$
$$x^2 = A$$
$$x^2 - A = 0$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{x_i^2 - A}{2x_i} = \frac{x_i}{2} + \frac{A}{2x_i}$$

$$\boxed{x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{x_i^2 - A}{2x_i} = \frac{x_i}{2} + \frac{A}{2x_i}}$$

Résultat du calcul pour A=9 et A=16 :

4.50000000000000	8.00000000000000
3.25000000000000	5.00000000000000
3.009615384615385	4.10000000000000
3.000015360039322	4.001219512195122
3.000000000039321	4.00000185844589
3.00000000000000	4.00000000000004
3.00000000000000	4.00000000000000

Intégration numérique des équations différentielles

Méthode d'Euler : $\left\{ \begin{aligned} & y' = f(x, y(x)) \\ & y(x_0) = y_0 \end{aligned} \right\}$ $\boxed{y(x_{k+1}) = y(x_k) + h \cdot f(x_k, y(x_k))}$

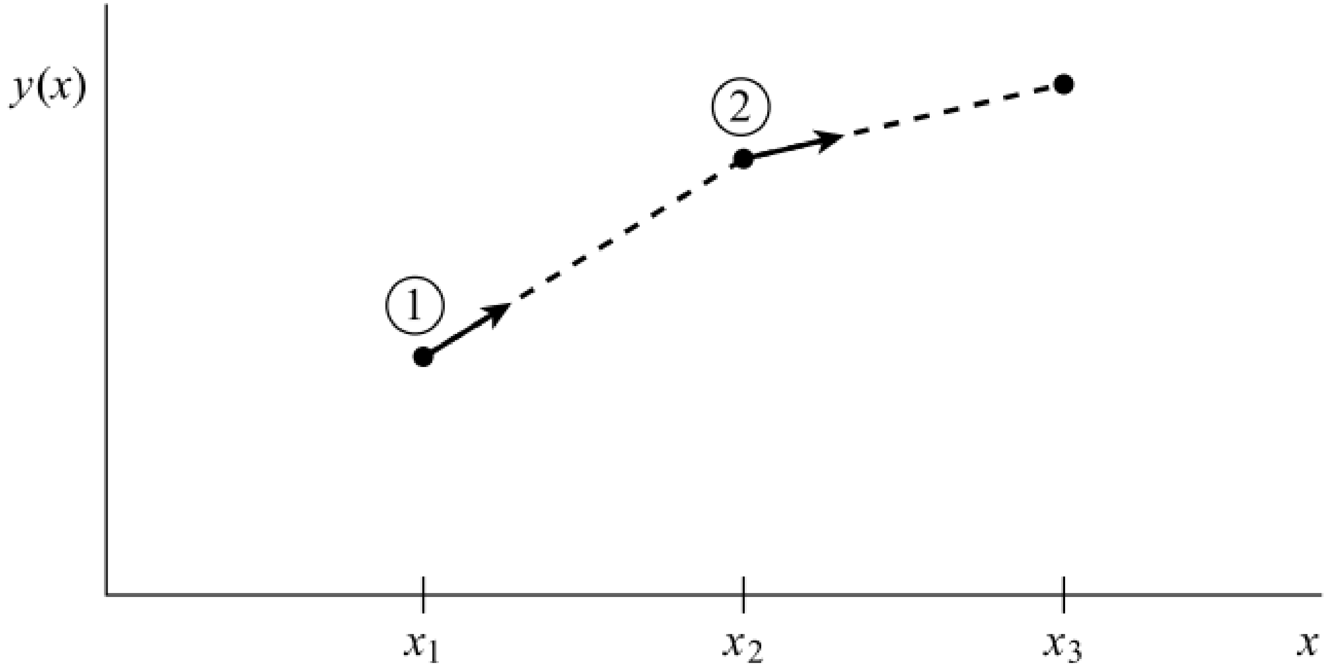


Figure 16.1.1. Euler's method. In this simplest (and least accurate) method for integrating an ODE, the derivative at the starting point of each interval is extrapolated to find the next function value. The method has first-order accuracy.

Méthode du point médian :

$$\begin{cases} k_1 = hf(x_n, y_n) \\ k_2 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \\ y_{n+1} = y_n + k_2 + O(h^3) \end{cases}$$

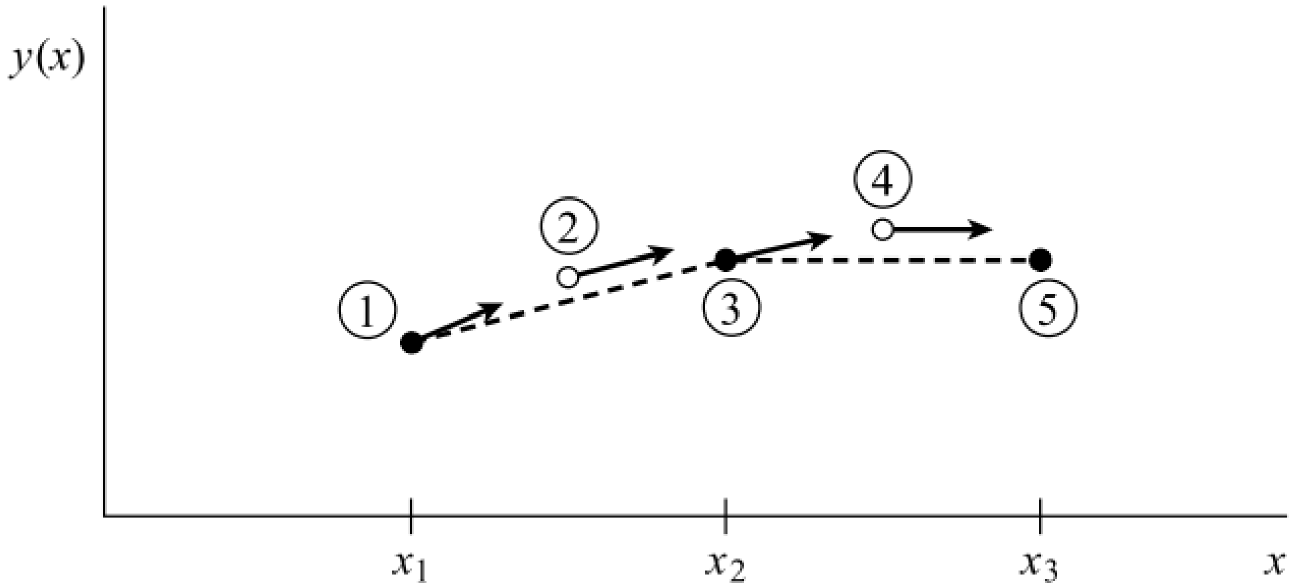


Figure 16.1.2. Midpoint method. Second-order accuracy is obtained by using the initial derivative at each step to find a point halfway across the interval, then using the midpoint derivative across the full width of the interval. In the figure, filled dots represent final function values, while open dots represent function values that are discarded once their derivatives have been calculated and used.

Méthode Runge-Kutta de l'ordre 4 :

$$\begin{cases} k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 = hf(x_n + h, y_n + k_3) \\ y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5) \end{cases}$$

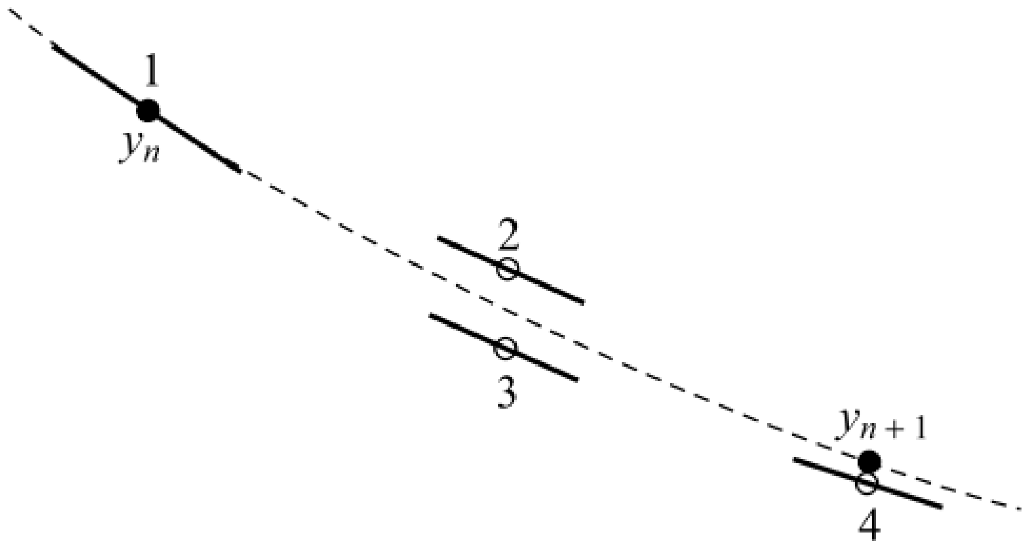
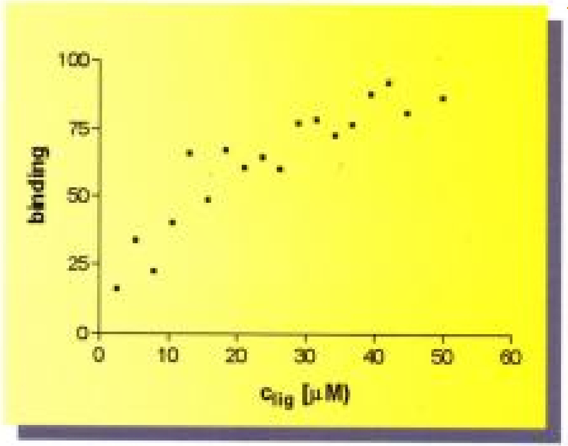


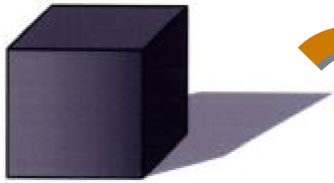
Figure 16.1.3. Fourth-order Runge-Kutta method. In each step the derivative is evaluated four times: once at the initial point, twice at trial midpoints, and once at a trial endpoint. From these derivatives the final function value (shown as a filled dot) is calculated. (See text for details.)

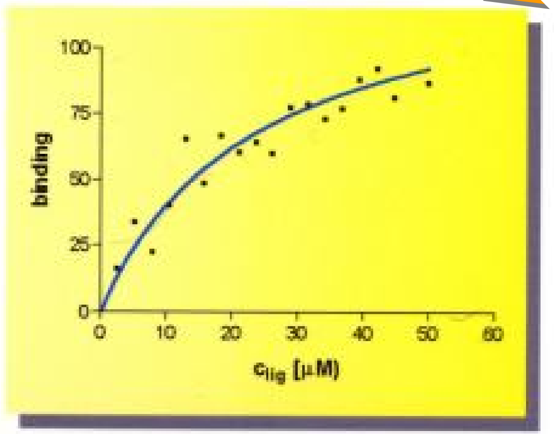
Techniques d'optimisation

Données expérimentales



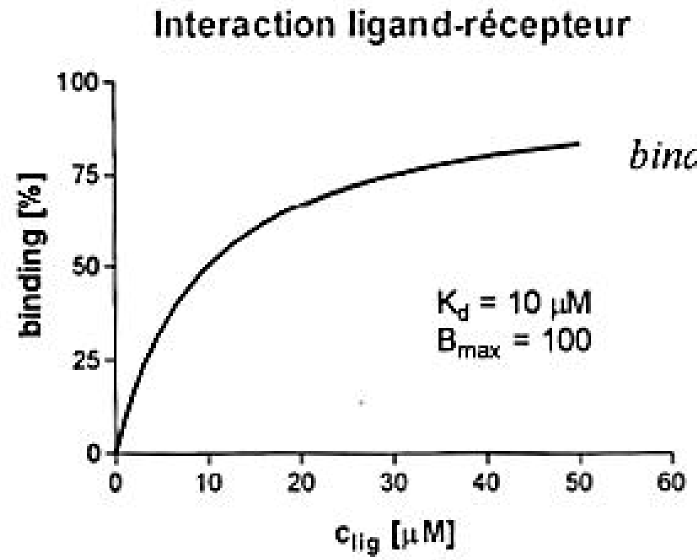
Une boîte noire





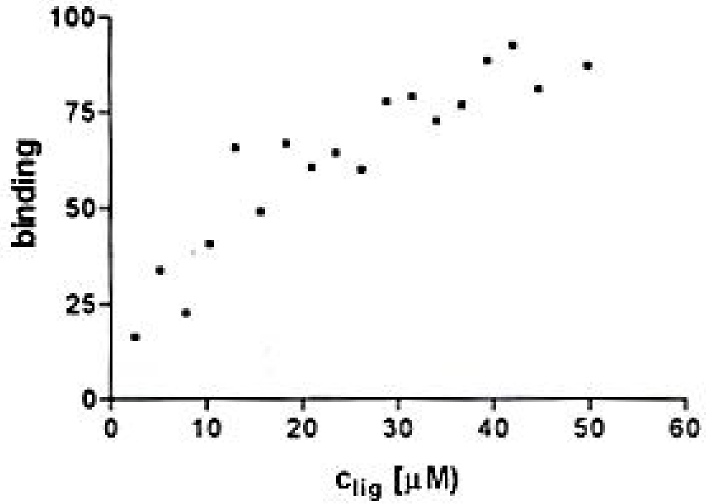
Les résultats du traitement ⇒ paramètres du modèle

A) Modèle d'un phénomène biochimique ou biologique

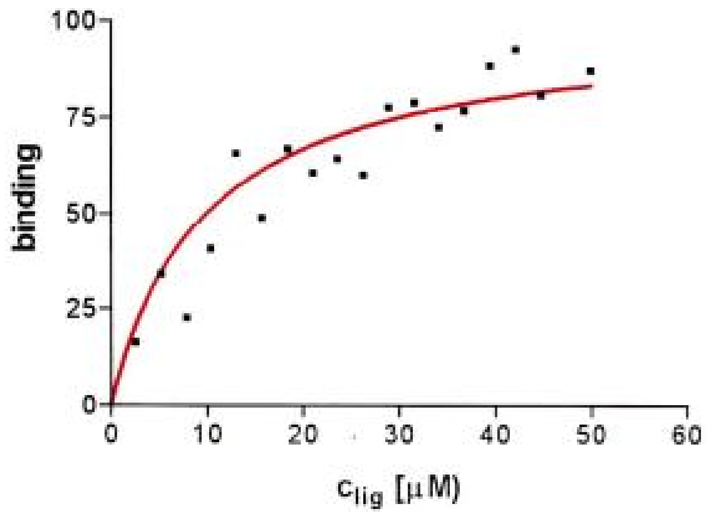


$$binding = \frac{B_{max} \cdot c_{lig}}{K_d + c_{lig}}$$

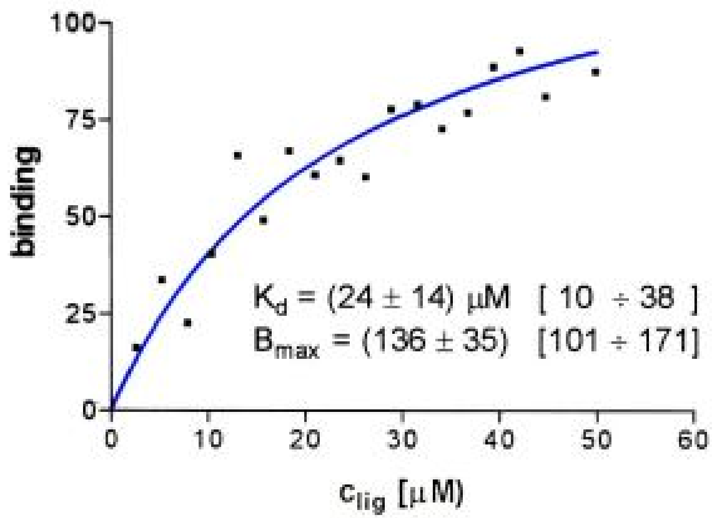
B) Résultats d'une expérience



C) Superposition de la courbe théorique et des points expérimentaux

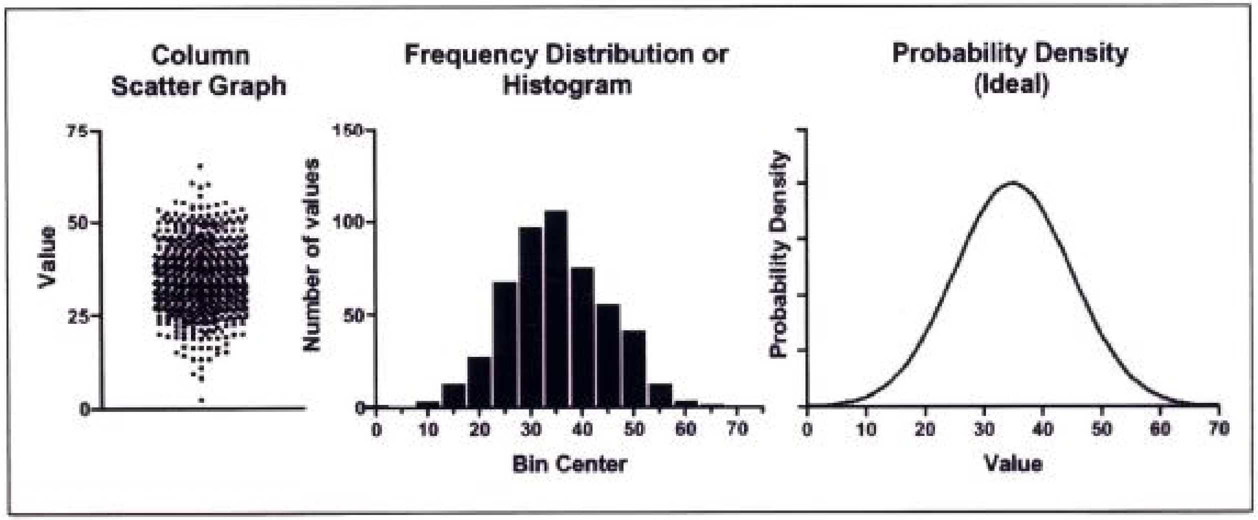


D) Résultat du *fitting*



Distribution gaussienne

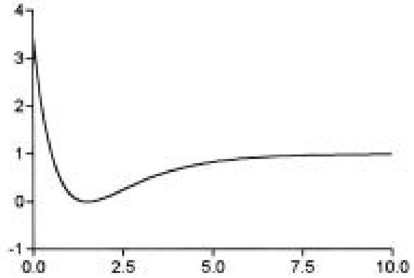
Quand un grand nombre de facteurs aléatoires et indépendants s'accumulent, les données apparaissent sous la forme de la distribution dite normale, ou gaussienne :



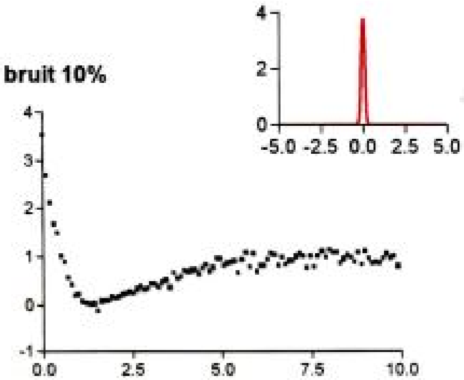
Distribution d'une grande quantité de données	**Distribution de la fréquence d'apparition de points (histogramme)**	**Distribution gaussienne idéale**

Influence du bruit sur les résultats de mesures

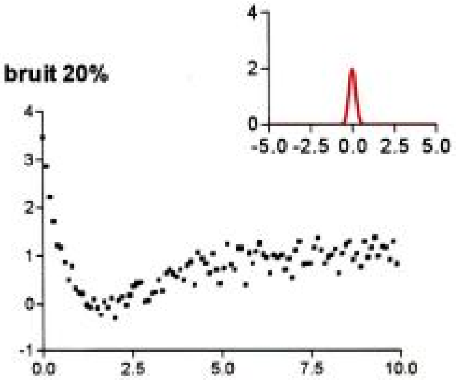
bruit 0%



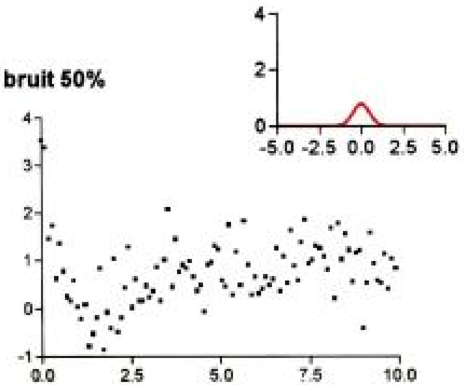
bruit 10%



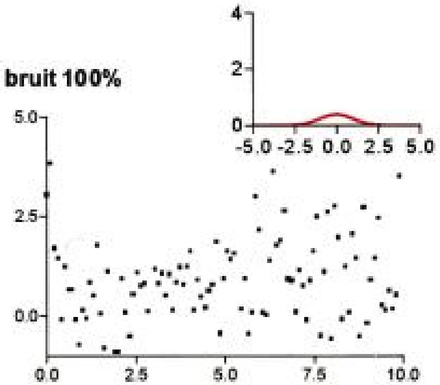
bruit 20%



bruit 50%

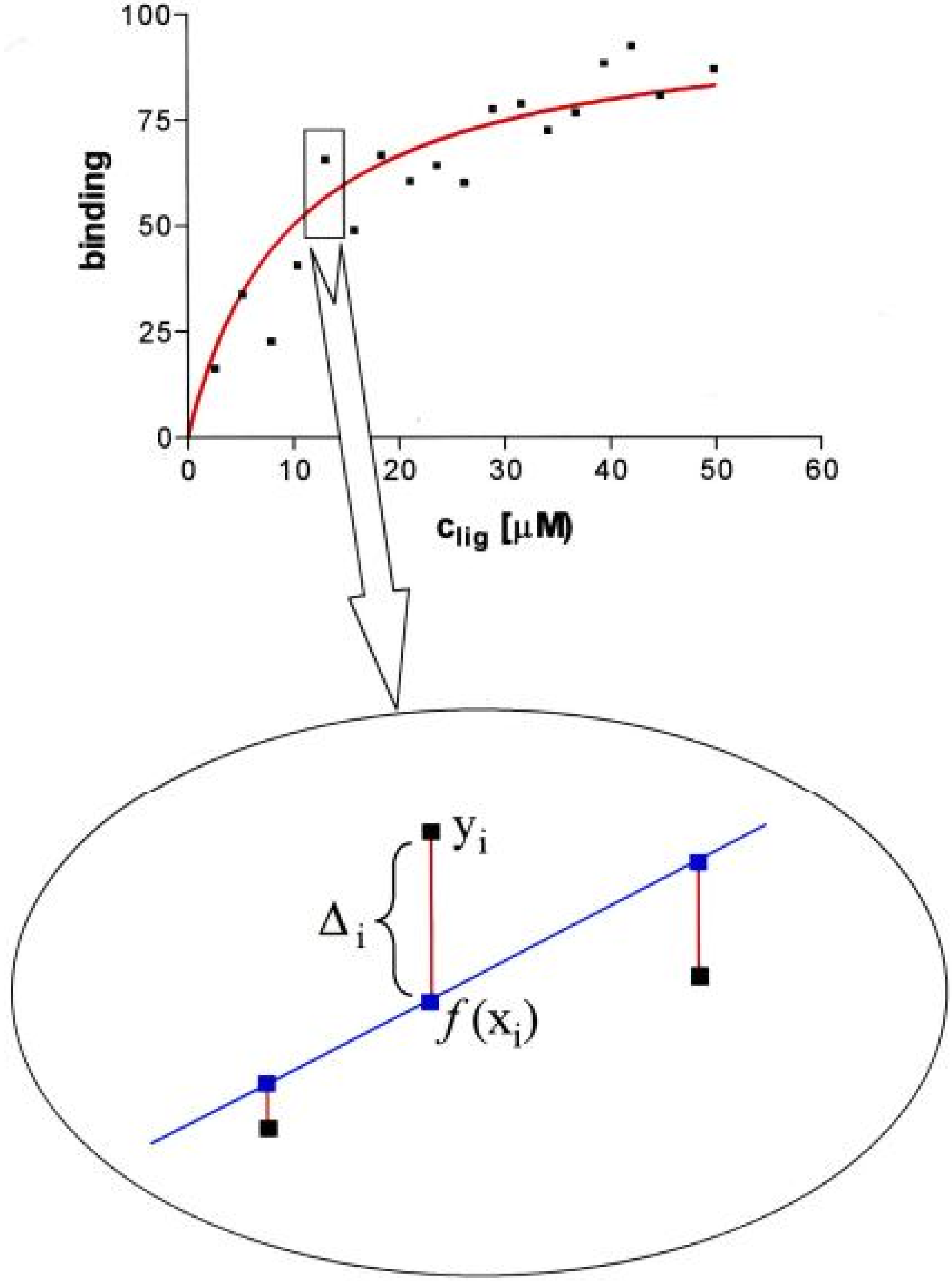


bruit 100%

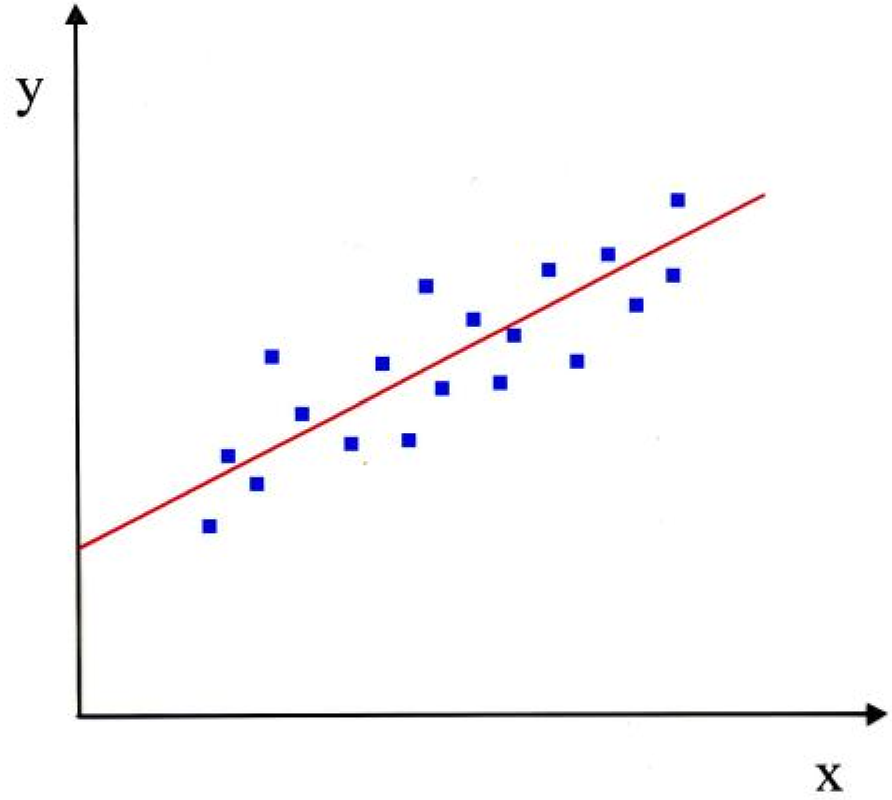


L'enjeu :

Trouver le minimum de la somme des déviations entre les points mesurés et ceux calculés d'après un modèle.



Exemple : ligne droite



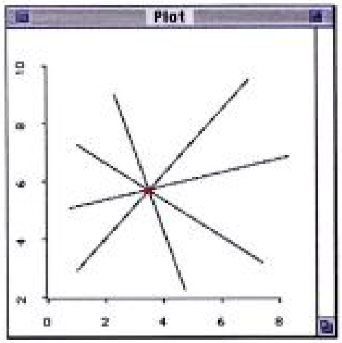
$$y_i = a \cdot x_i + b + \sigma_i$$

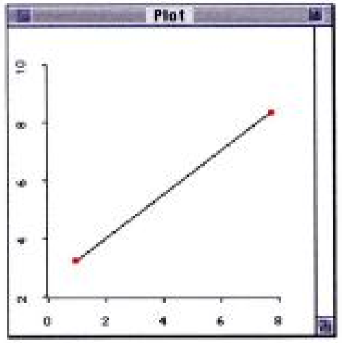
$$y_i = \alpha \cdot x_i + \beta$$

a, b : les paramètres du modèle
α, β : les estimateurs des paramètres **a, b**

Nombre de degrés de liberté
(degrees of freedom)

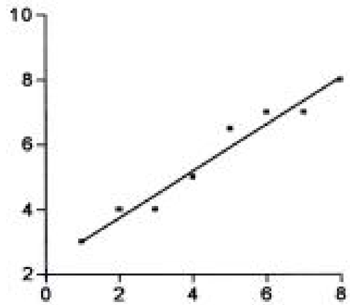
Le nombre de degrés de liberté (DL) représente "**le nombre d'observations moins le nombre de relations parmi ces observations**" (Walker, 1940).





Il est impossible de dessiner une droite de régression si on dispose seulement d'un point de mesure (c.à.d. une observation). En effet, beaucoup de droites peuvent passer par ce point. Il y a une ambiguïté.
(**DL = 1 – 2 = -1**).

Pour deux observations, il y a une droite unique qui passe par les deux points et la corrélation est parfaite (r = 1). Mais le bruit contenu dans les mesures détermine le résultat ! La valeur **DL = 2 – 2 = 0**, donc on n'a **aucune liberté** dans l'estimation de paramètres de la droite.



Le nombre de degrés de liberté spécifie le nombre de données utiles pour estimation. **La confiance en résultat s'augmente avec la valeur de DL.**

Régression linéaire

$$d = \frac{1}{\sigma^2} \sum_{i=1}^{N} \left[y_i - f(x_i) \right]^2 = \min$$

$$f(x) = \sum_{j=1}^{n} a_j \Phi_j(x)$$

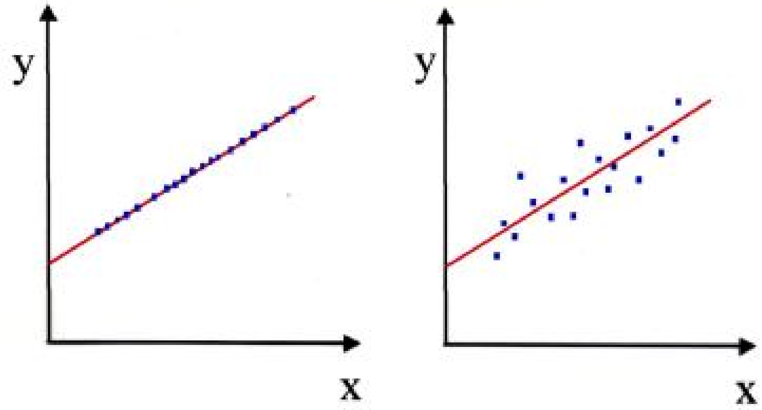
$$f(x) = a_1 \cdot \Phi_1(x) + a_2 \cdot \Phi_2(x) + \ldots$$

$$a_1 = a \qquad \Phi_1(x) = x$$
$$a_2 = b \qquad \Phi_2(x) = 1$$
$$f(x) = ax + b$$

$$a_1 = a \qquad \Phi_1(x) = \sin(x)$$
$$a_2 = b \qquad \Phi_2(x) = e^{-x}$$
$$a_3 = c \qquad \Phi_3(x) = x^4$$
$$f(x) = a \cdot \sin(x) + b \cdot e^{-x} + c \cdot x^4$$

$$a = \frac{\langle xy \rangle - \langle x \rangle \cdot \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2}$$

$$b = \langle y \rangle - a \cdot \langle x \rangle$$



r=0.999

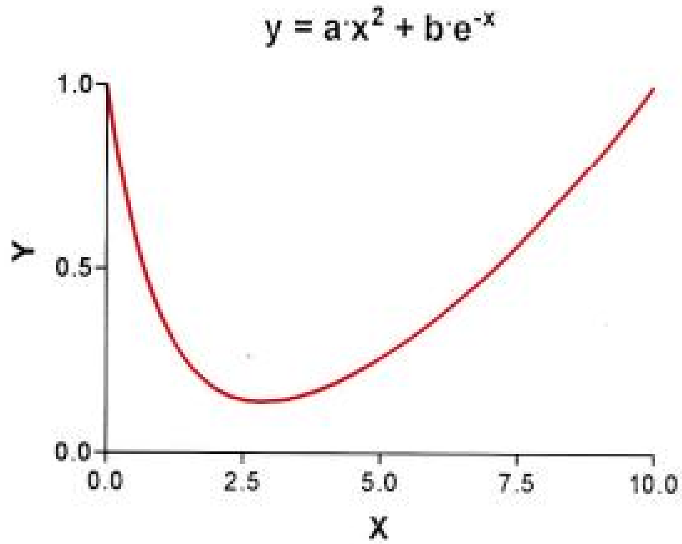
r=0.750

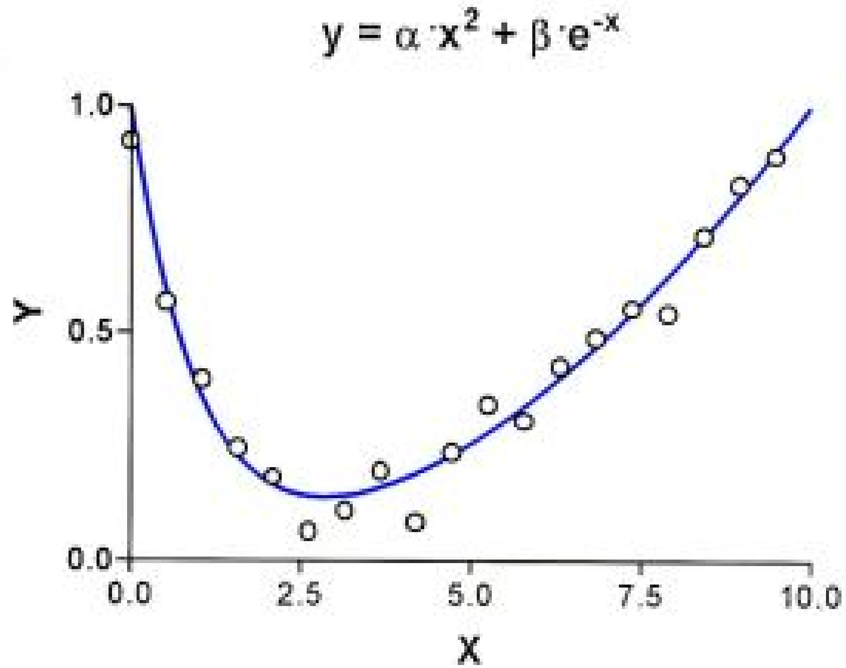
$$r = \frac{\sum_{i=1}^{N} (x_i - \langle x \rangle)(y_i - \langle y \rangle)}{\sqrt{\left(\sum_{i=1}^{N} (x_i - \langle x \rangle)^2\right)\left(\sum_{i=1}^{N} (y_i - \langle y \rangle)^2\right)}}$$

$$r = \frac{\langle xy \rangle - \langle x \rangle \cdot \langle y \rangle}{\sqrt{\left(\langle x^2 \rangle - \langle x \rangle^2\right)\left(\langle y^2 \rangle - \langle y \rangle^2\right)}} = a \cdot \sqrt{\frac{\langle x^2 \rangle - \langle x \rangle^2}{\langle y^2 \rangle - \langle y \rangle^2}}$$

Régression linéaire

(approche linéaire par rapport aux paramètres cherchés,
courbe non-linéaire par rapport à la variable *x*)





Limitations de la régression linéaire

- $f(x) = a \cdot \sin(x) + b \cdot e^{-x} + c \cdot \sqrt{x}$
- $g(x) = \sin(a \cdot x) + e^{-b \cdot x} + \sqrt{x - c}$

- $y = e^{-b \cdot x}$

  $\ln(y) = -b \cdot x$
- $Y = -b \cdot x$

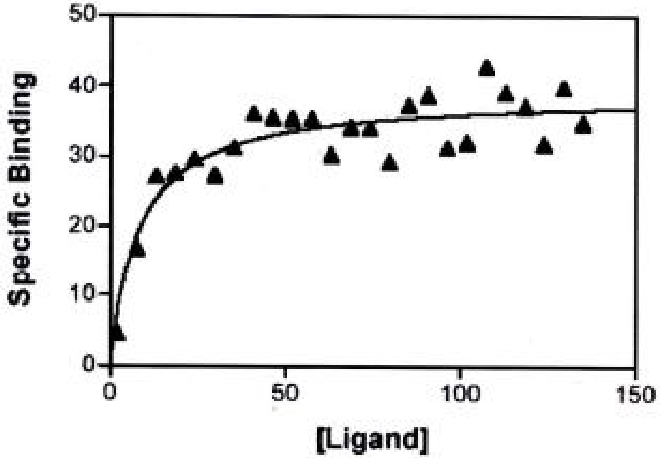
$$y = e^{-a \cdot x} + e^{-b \cdot x}$$

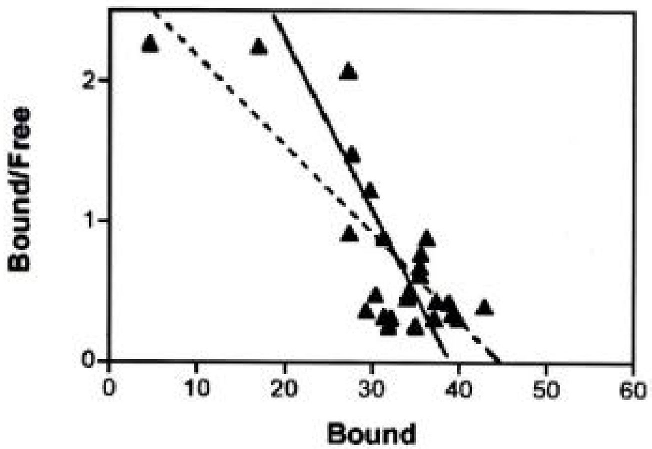
???

Cas linéaire : calcul direct

Cas nonlinéaire : recherche itérative

Exemple d'une transformation nonlinéaire





Cas nonlinéaire (la recherche itérative)

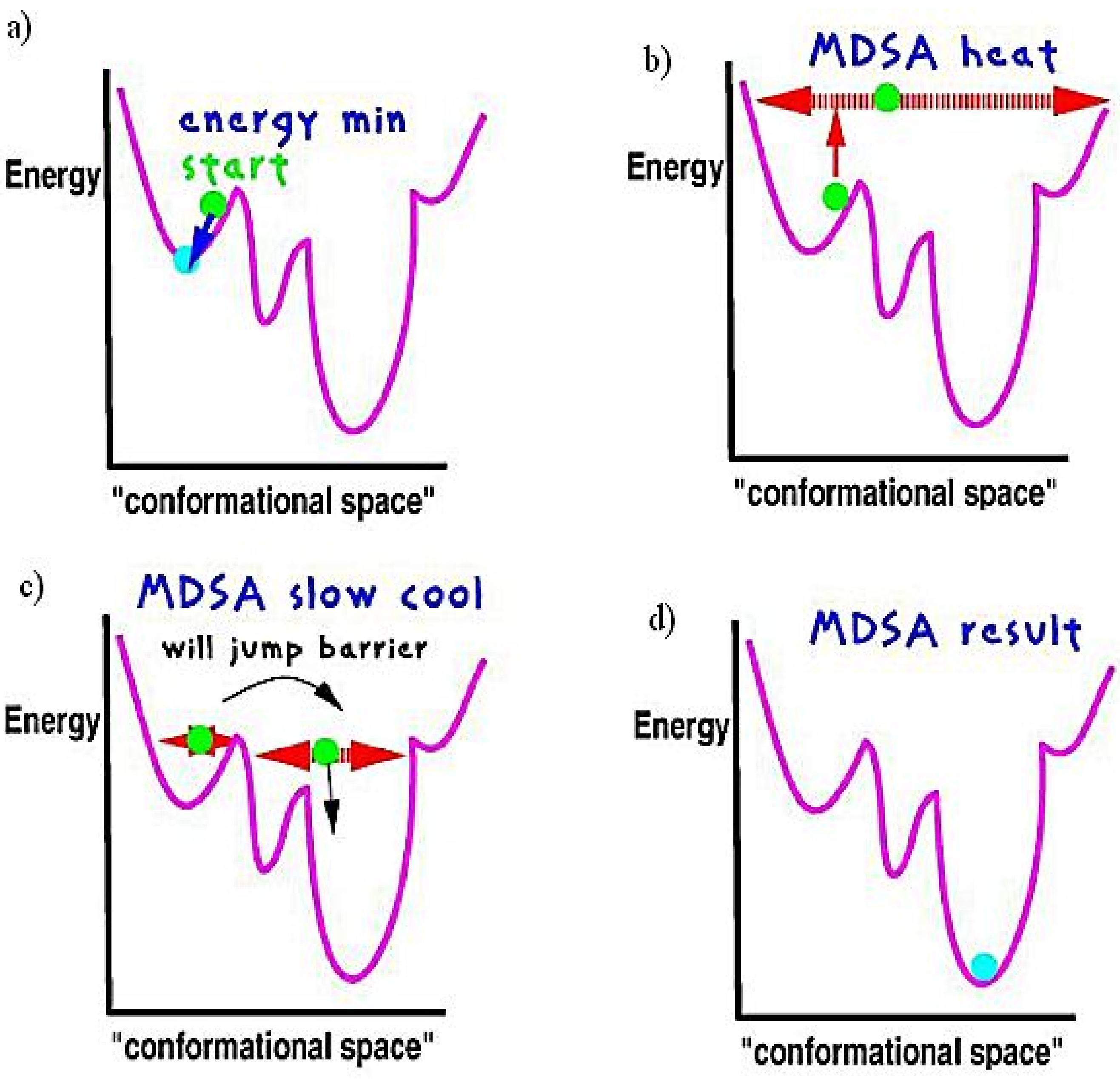
Problème :

Le minimum trouvé peut être local plutôt que global

Solution :

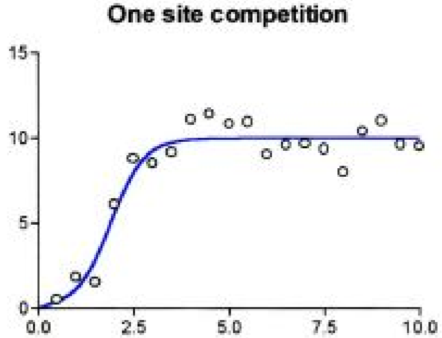
répéter la recherche plusieurs fois avec des points de départ différentes et choisir la solution avec les moindres carrés

Recuit simulé

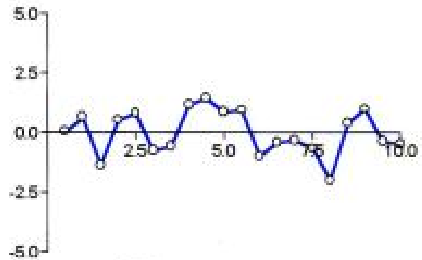


Le principe :
La recherche du minimum global *via* refroidissement lent (adiabatique) du système étudié.

Valeurs résiduelles

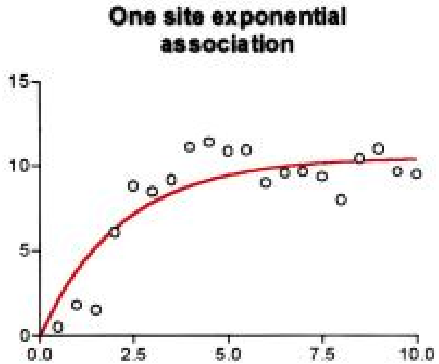


Residuals:
Points above curve: 10
Points below curve: 10
Number of runs: 8
P value (runs test): **0.1276**
Deviation from Model: **Not Significant**

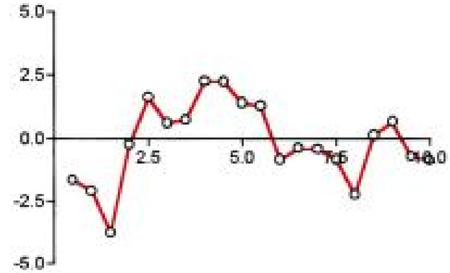


$$y_i^{\exp} = f(x_i) + \sigma_i$$

$$y_i^{\exp} - f(x_i) = \sigma_i$$



Residuals:
Points above curve: 9
Points below curve: 11
Number of runs: 5
P value (runs test): **0.004882**
Deviation from Model: **Significant**



$$y_i^{\exp} = g(x_i) + \sigma_i$$

$$f(x_i) - g(x_i) \overset{?}{=} \sigma_i$$

Analyse de valeurs résiduelles