Description statistique d'une série temporelle

Jacques Le Bourlot

Janvier 2007

Contents

1 Introduction

Un problème fréquent, à la frontière des statistiques, du traitement du signal et de l'analyse numérique, consiste à tenter de retrouver dans une suite de nombre une "loi" sous-jacente. Nous ne chercherons pas ici à définir avec rigueur ce que peut être une telle loi ou quelle est son origine[1], mais nous illustrerons sur un exemple quelques techniques de base.

L'exemple choisi est le résultat d'une marche au hasard simulée sur ordinateur. Le but est de caractériser "au mieux" les données brutes. Définir ce que veut dire "au mieux" fait partie de l'exercice.

Les données de base se présentent sous la forme de deux fichiers de n points comportant pour le premier les positions successives (x_i, y_i) du marcheur, et le deuxième les caractéristiques (θ_i, l_i) de chaque pas. Si M_i a pour coordonnées (x_i, y_i), on a:

$$\theta_i = \left(\overrightarrow{Ox}, \overrightarrow{\overrightarrow{M_iM_{i+1}}}\right)$$

$$l_i = \left\| \overrightarrow{M_iM_{i+1}} \right\|$$

Ces deux informations sont bien sûr redondantes, mais il est pratique de pouvoir travailler directement à partir de l'une ou de l'autre.

Dans un premier temps, nous allons travailler indépendamment sur chacune des quatre séries de nombres. Dans un deuxième, nous chercherons des liens éventuels entre elles.

Beaucoup d'algorithmes et de remarques sont inspirés de "Numerical Recipes in C", de Press, Teukolsky, Vetterling & Flannery, 1992, Cambridge University Press. Ce livre est très controversé: sur chacun des sujets qu'il traite, on peut trouver un ouvrage spécialisé qui soit meilleur ou plus complet. Il comporte également un certain nombre d'erreurs ou de partis pris discutables. En même temps, il comporte une richesse phénoménale de méthodes et d'algorithmes, présentés de façon très claire et accompagnés de programmes directement fonctionnels. C'est donc une excellente base de départ, à condition de savoir le compléter.

[1] Pour un peu (beaucoup) de rigueur, on se reportera au poly de cours de Didier Pelat "Bruits et Signaux", Master d'Astrophysique d'Ile-de-France.

2 Données brutes (ou presque)

2.1 Une marche de 200000 pas

Les données peuvent contenir un très grand nombre de points. Ici, typiquement nous travaillerons avec $n = 200000$. Les figures 1 et 2 montrent la successions des positions.

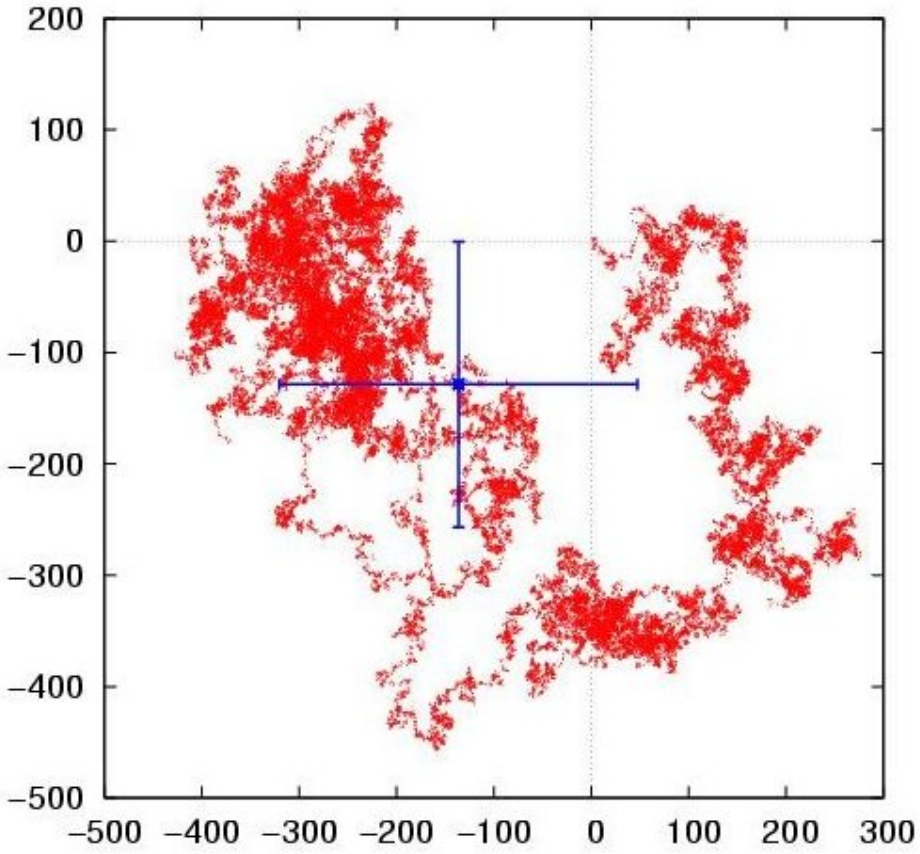


Figure 1: Positions. La moyenne et l'écart type (en bleu) représentent mal les données.

On voit que les positions ne sont pas du tout réparties de façon uniforme (figures 1 et 2). En revanche, si l'on examine la répartition des angles (figure 3) ou des longueurs de pas (figure 4), les résultats sont beaucoup plus réguliers. Les angles semblent régulièrement répartis entre $-\pi$ et π. Les distances, en revanche, se regroupent autour de 1 mais ne sont pas réparties de façon homogène. Les deux indicateurs ne semblent pas évoluer au court du "temps" (le nombre de pas).

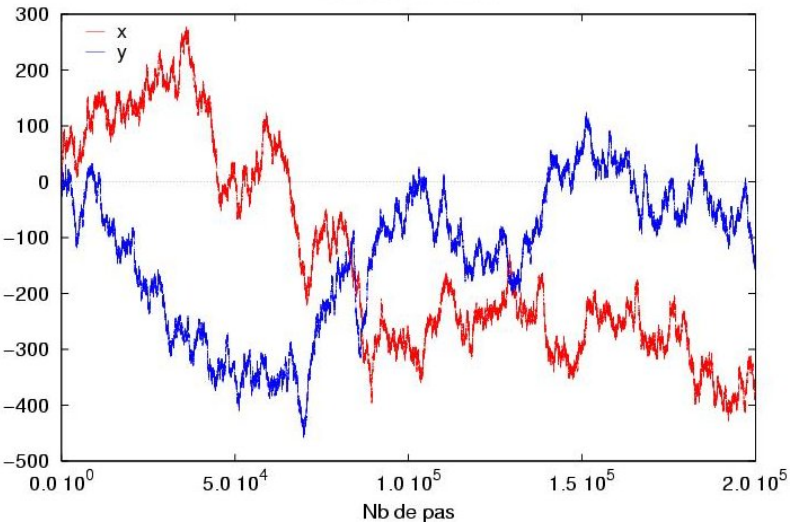


Figure 2: Suite. Les points successifs sont très corrélés.

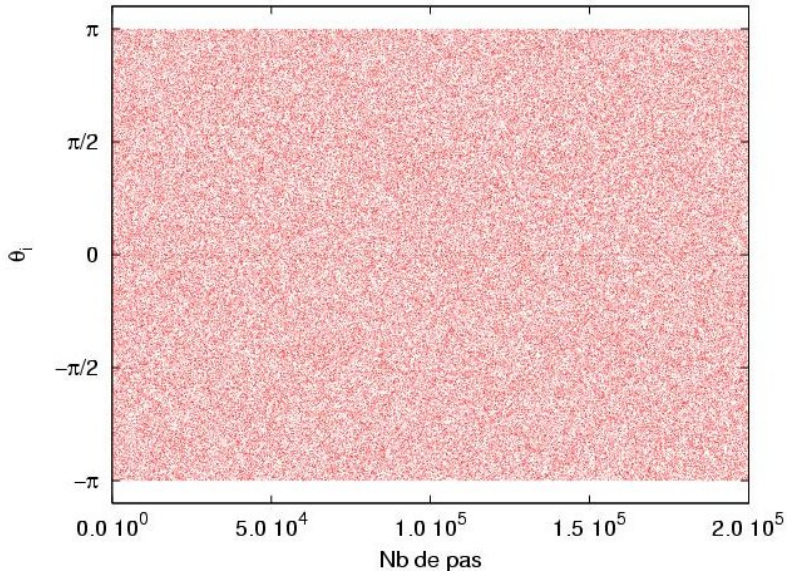


Figure 3: Angles

2.2 Description statistique

A partir de chaque série temporelle, on peut calculer les deux premiers moments:

Moyenne empirique

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

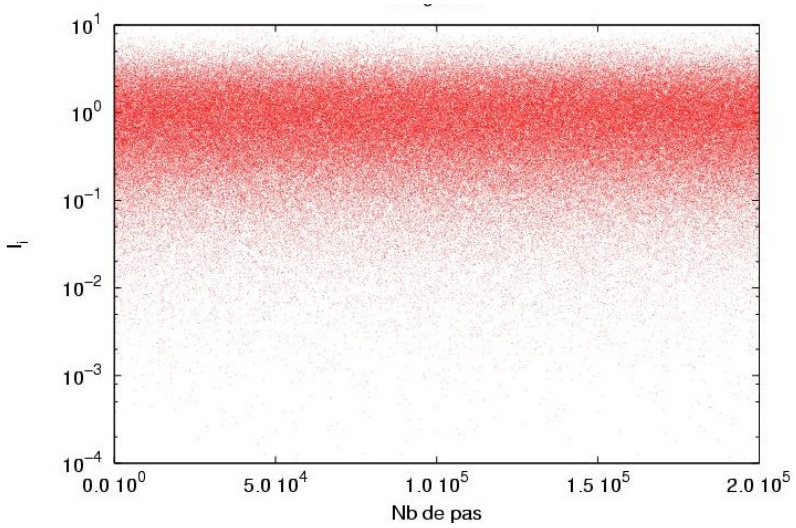


Figure 4: Longueurs des pas

Écart type empirique

$$\sigma = \sqrt{v}$$

Avec

$$v = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

Note: la variance empirique[2] v est la moyenne quadratique des écarts à la moyenne, l'écart type empirique est la racine de la variance.

calcul pratique Il est immédiat, et sans piège. On calcule d'abord la moyenne, puis la variance, par deux boucles indépendantes. Si on souhaite une très grande précision, et que les x_i peuvent prendre des valeurs couvrant plusieurs ordres de grandeur, il est conseillé de trier[3] d'abord les données, puis de sommer en partant des plus petits nombres en valeur absolue. En général, ce n'est pas une grosse difficulté.

2.3 Représentation des distributions

On représente souvent les "fonctions de répartition" à partir de l'histogramme du nombre d'occurences des différentes valeurs dans des intervalles prédéfinis. On a ici quatre séries, dont les histogrammes sont donnés figures 5 à 8. Sur chaque figure, la moyenne et l'écart type sont indiqués par des barres bleu. La médiane par une barre mauve.

[2]La moyenne (variance, ...) empirique est celle de l'échantillon, et non celle de la loi.
[3]Nous reviendrons plus loin sur la question du tri des données.

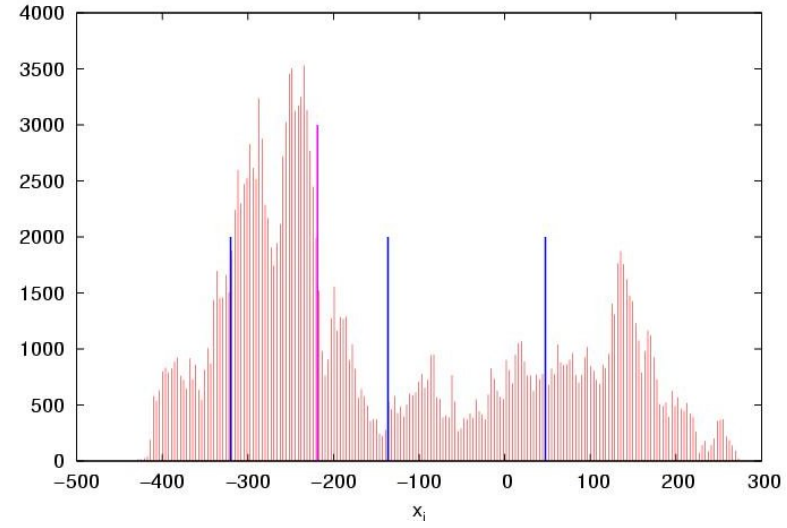


Figure 5: Histogramme des x_i

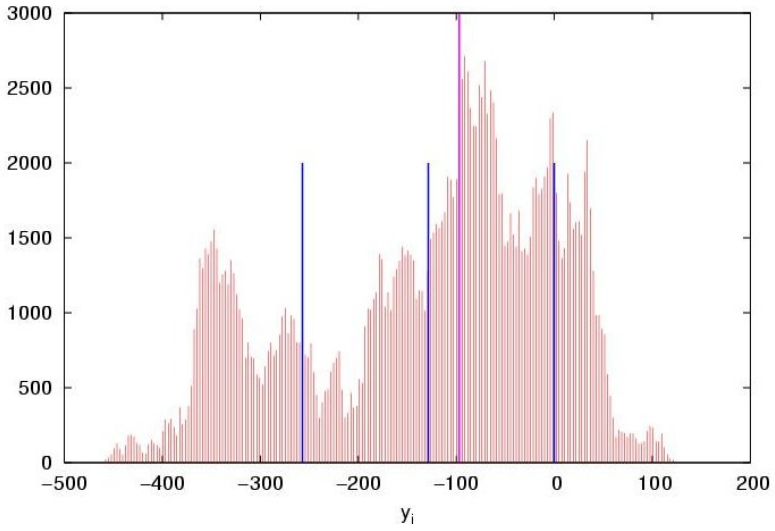


Figure 6: Histogramme des y_i

On a choisi ici de diviser l'intervalle entre les valeurs minimales et maximales en 200 "bins" égaux. Avec un tirage de 200000 points, on devrait donc avoir, "en moyenne", 1000 points par bin. On constate que c'est effectivement "à peu près"

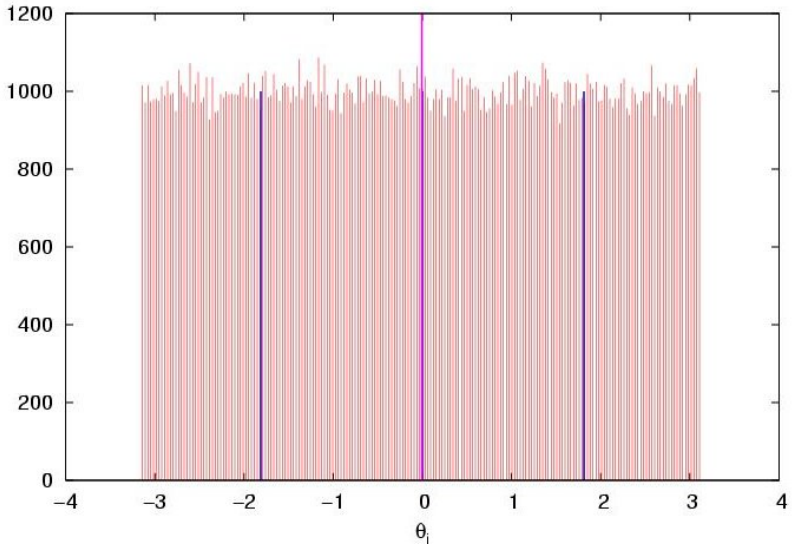


Figure 7: Histogramme des θ_i

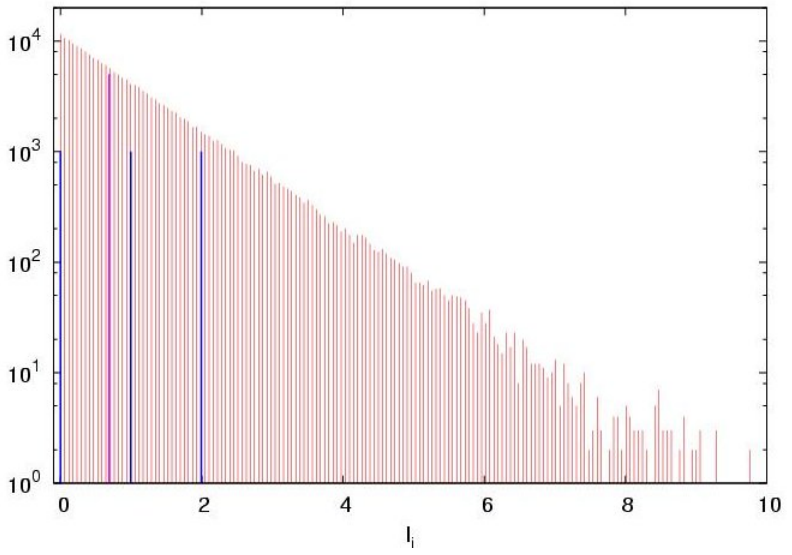


Figure 8: Histogramme des l_i. Noter l'échelle exponentielle des y.

le cas pour la distribution des angles, qui semblent répartis uniformément. Il y a cependant des fluctuations sur lesquelles nous reviendrons.

Les positions (x_i et y_i) en revanche n'obéissent pas à une distribution régulière. On voit ici aussi que moyenne et écart type sont de très mauvais indicateurs de la "forme" de la distribution.

La distribution des l_i, elle, est également régulière, mais absolument pas uniforme. une représentation "lin-log" semble suivre une droite, au moins dans la partie où le nombre de point par bin est grand. On cherchera donc si la distribution inconnue peut être exponentielle.

Calcul pratique Étant donnée la liste des valeurs prises par la série temporelle, il n'est pas si facile de trouver "une" bonne représentation de la "PDF" (Probability Distribution Function). Trouver les valeurs extrêmes est immédiat (une simple boucle suffit), mais il faut ensuite décider comment découper cet intervalle en "bins" significatifs. Il y a deux écueils à éviter: si le nombre de bins est trop grand, chacun contient trop peu de points et la distribution est très bruitée. La présence possible de bins vides peut également entraîner des difficultés purement numériques (si on veut calculer des logarithmes par exemple) qui compliquent la programmation. Si le nombre de bins est trop petit, la distribution est trop peu échantillonnée. En général, un nombre de bins raisonnable est de l'ordre du centième du nombre de points, mais cela doit toujours être vérifié *a posteriori*. En outre, il n'est pas évident que des bins de même taille soient un choix optimal. Ici, il aurait probablement été préférable pour la distribution des l_i de prendre des bins dont la taille augmente progressivement. Encore faut-il avoir une idée de la répartition, ...et l'on tourne en rond.

2.4 Fonction de répartition cumulée

La raison principale pour laquelle il est difficile de représenter la PDF, est que celle-ci n'est **pas** la "bonne" quantité statistique à prendre en compte. En pratique, il faut travailler à partir de sa primitive. En effet, si $f(x)$ est telle que:

$$p(x_0 < x < x_0 + dx) = f(x_0)\,dx$$

($f(x_0)\,dx$ est la probabilité de trouver x entre x_0 et $x_0 + dx$), alors on a:

$$p(x < x_0) = \int_{-\infty}^{x_0} f(x)\,dx = F(x_0)$$

$F(x_0)$ est la probabilité qu'une valeur x quelconque soit inférieure à x_0. On voit (par construction) que $F(-\infty) = 0$ et $F(+\infty) = 1$. Or, F est très simple à obtenir. Il suffit en effet de trier la série temporelle observée par valeurs croissantes, de prendre comme abscisse la valeur de x et comme ordonnée le rang du point trié divisé par le nombre total de points.

Ici, on obtient les figures 9 à 12.

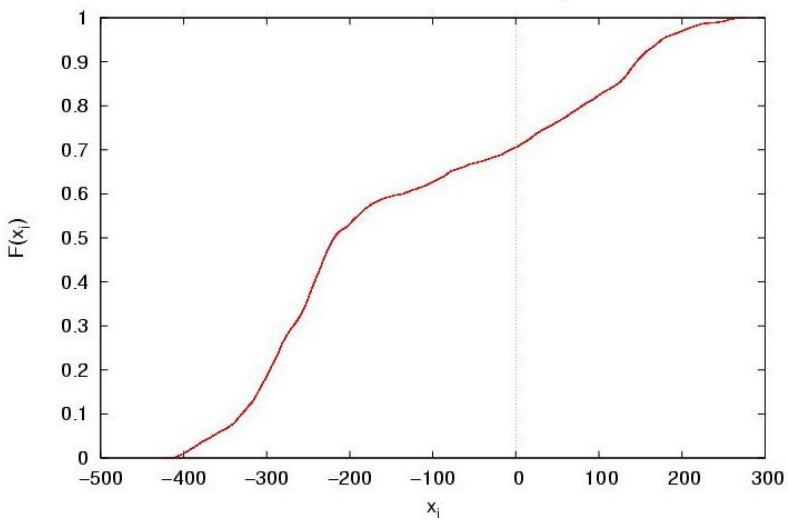


Figure 9: Fonction de répartition des x_i.

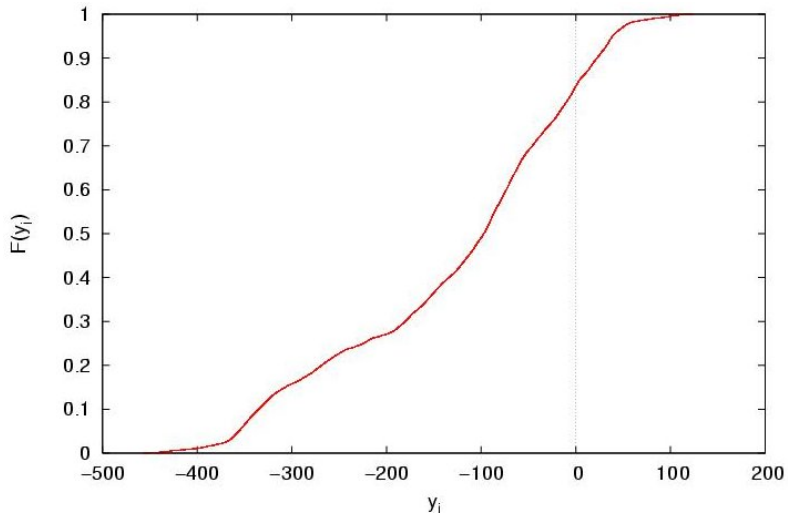


Figure 10: Fonction de répartition des y_i.

Les deux premières courbes (x_i et y_i) montrent l'irrégularité de la distribution. Les θ_i en revanche semblent bien répartis de façon homogène. Enfin, les l_i suivent une courbe précise. Un changement de variable (figure 13) permet de vérifier qu'il s'agit bien d'une exponentielle.

Analytiquement, on soupçonne donc ici deux lois:

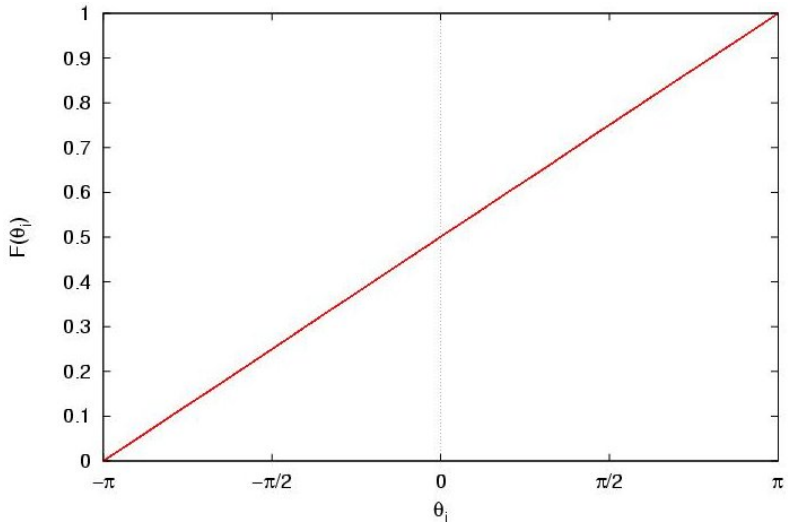


Figure 11: Fonction de répartition des θ_i.

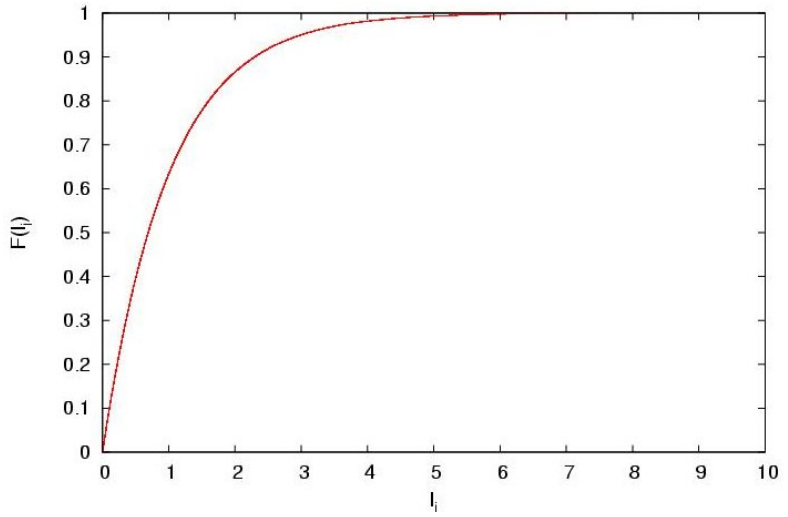


Figure 12: Fonction de répartition des l_i.

Loi des θ_i

Si les θ_i sont distribués uniformément dans l'intervalle $[-\pi : \pi]$, $f_\theta(x) = \alpha$, où α est une constante. En utilisant la condition de normalisation $\int_{-\pi}^{\pi} \alpha \, dx = 1$, on trouve $\alpha = 1/2\pi$. On obtient donc une fonction de répartition cumulée F_θ telle que:

$$F_\theta(x) = \int_{-\pi}^{x} \frac{dt}{2\pi} = \frac{x + \pi}{2\,\pi}$$

On tentera un peu plus loin de confirmer quantitativement cette spéculation.

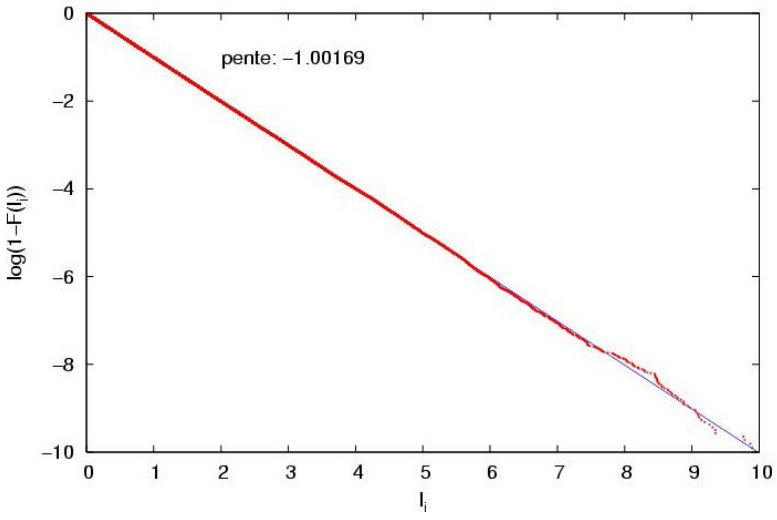


Figure 13: Fonction de répartition des l_i, après changement d'axes.

Loi de l_i

Si les l_i sont distribués exponentiellement dans l'intervalle $[0 : +\infty[$, $f_l(x) = \alpha \exp(-\beta x)$, ou α et β sont des constantes. La normalisation nous dit que: $\alpha \int_0^\infty \exp(-\beta t)\, dt = 1$ soit $\alpha = \beta$. La fonction de répartition cumulée F_l est donc telle que:

$$F_l(x) = \alpha \int_0^x \exp(-\alpha t)\, dt = 1 - \exp(-\alpha x)$$

Les résultats ci-dessus suggèrent fortement $\alpha = 1$, ...ce qui reste évidemment à vérifier!

2.5 Le problème du tri:

Trier des données peut sembler simple: on prend deux valeurs numériques, on les compare, et on range en premier la plus petite des deux. En pratique, tout algorithme dérivé de ce principe se révèle vite inutilisable. En effet, cela revient à chercher d'abord la plus petite valeur ($n-1$ tests), puis la deuxième plus petite ($n-2$ tests), etc... Soit un total de $(n-1)(n-2)/2$ tests, ou bien encore un algorithme en "n^2". En général, ce n'est pas très encourageant. Ici, il est facile de faire mieux: prenons un premier nombre au hasard (par exemple le premier). On compare successivement toutes les autres valeurs, et on les range dans un vecteur en partant de la $n$ième case s'ils sont plus grand et de la première s'il sont plus petits. Après $n-1$ tests, on a bien trouvé la place de ce premier nombre, mais surtout on a divisé l'ensemble en deux parties d'environ $n/2$ nombres chacune (si le premier nombre choisi est en gros "vers" le milieu, ce qui est "statistiquement" probable

pour un ensemble désordonné). On peut maintenant recommencer l'opération sur chacun des deux sous-ensembles pour parvenir après environ n tests à quatre paquets d'environ $n/4$ éléments. A chaque étape, il faut bien n tests (ou à peu près), mais le nombre d'étape est beaucoup plus faible que dans le cas précédent. Le nombre de fois où l'on doit diviser l'ensemble en deux est de l'ordre de $\log_2 n$. On a donc un algorithme en "$n \log n$". Le gain est donc (à une constante inconnue près) d'un facteur $n/\log n$, soit un facteur 16000 pour nos 200000 points. Ce qui ferait passer une opération de tri de 2s à ...9h!

En pratique, il est déconseillé d'écrire soi même un tel algorithme. "Quicksort", ou la variante "Heapsort" (moins rapide dans les meilleurs cas, mais plus robuste dans les mauvais) sont disponibles dans la plupart des bonnes librairies et prennent soin des subtilités cachées.

3 Analyse

Les résultats présentés jusque là sont très descriptifs, bien que l'on soit tenté d'identifier certaines courbes, "à l'oeil". Pour caractériser les distributions, on peut maintenant se poser deux types de questions:

1. Si l'on veut "faire passer" une fonction a priori par un nuage de points, quels sont les meilleurs paramètres?

2. Avec quel degré de confiance peut-on affirmer que les points observés proviennent bien d'une loi donnée?

Contrairement aux apparences, ces deux questions sont très différentes l'une de l'autre. La première relève des techniques d'approximation. Il est (presque) toujours possible de lui donner une réponse précise et satisfaisante (ce qui ne veut pas dire que l'approximation, elle, est satisfaisante). On utilise pour cela le plus souvent[4] les techniques dites de "moindre carré", sous une forme plus ou moins élaborée. La deuxième relève des statistiques. La réponse est tout aussi précise, mais pas toujours aussi simple à interpréter. Nous allons les étudier successivement sur les distributions θ_i et l_i. Les suites de x_i et y_i, elles, montrent des distributions irrégulières, mais de fortes corrélations d'un point au suivant qui nécessitent une analyse différente. Pour l'instant, la section correspondante dans ce poly n'est pas rédigée.

[4]Attention: lorsque l'on a de bonnes raisons de penser que la distribution des erreurs et incertitudes n'est pas "normale" (i.e. gaussienne), il faut utiliser des estimateurs plus fiables (ou "robustes") que les moindres carrés. Voir la discussion préliminaire sur ce sujet dans "Numerical Recipes", chapitre 15.7.

Dans les deux cas, ce qui fait la qualité de la réponse, c'est son objectivité. Il est indispensable que deux personnes indépendantes, confrontées au même problème parviennent à la même solution. Cela exclut, par exemple, de faire passer "à l'oeil" une droite à travers un nuage de points.

3.1 Moindres carrés

On cherche ici à ajuster à une série de points expérimentaux donnée (obtenue par une méthode qui ne nous concerne pas, ...pour l'instant), une fonction, donnée *a priori* et dépendant d'un certain nombre p de paramètres. Pour que l'opération ait un sens, le nombre de paramètres doit rester petit par rapport aux nombre de points de la série ($p << n$). Afin de déterminer "objectivement" si un jeu de paramètres est "meilleur" qu'un autre, il faut choisir un critère de qualité. C'est dans ce choix que se trouve concentré l'art du modélisateur. Il y a de nombreuses variantes, mais qui se ramènent toutes plus ou moins à minimiser une mesure de l'écart entre les données et la fonction. Le cas particulier présenté ici dérive de l'analyse du "maximum de vraisemblance".

3.1.1 Définition du χ^2

Considérons ici une fonction $f_{a,b}(x)$ dépendant de deux paramètres a et b. Pour l'instant, cette fonction est quelconque, et en particulier, il n'y a aucune raison qu'elle soit linéaire, ni par rapport à x, ni par rapport à a et b. Si l'on note ici (x_i, y_i) les points expérimentaux[5], on va chercher à minimiser les écarts entre les y_i et $f_{a,b}(x_i)$. On peut mesurer ces écarts par les données de $y_i - f_{a,b}(x_i)$, mais ce n'est pas très commode parce que ces écarts changent de signe. Prendre des valeurs absolues n'est pas pratique non plus[6], à cause de la non-dérivabilité. Il est donc d'usage de considérer les carrés des écarts. On construit donc une fonction "mérite", que l'on nomme habituellement χ^2 par:

$$\chi^2(a,b) = \sum_{i=1}^{n} \left(f_{a,b}(x_i) - y_i\right)^2$$

Notons qu'il s'agit bien d'une fonction des deux paramètres a et b. Une fois les valeurs des x_i et de chaque y_i correspondant fixées, $\chi^2(a,b)$ possède les caractères de

[5] Attention au changement de signification de la notation x_i et y_i par rapport à la section précédente.

[6] C'est pourtant un choix fréquent quand on cherche une méthode "robuste", c'est-à-dire peu sensible à la présence de points aberrants.

continuité et de dérivabilité de $f_{a,b}$, et c'est une fonction positive (par construction) qui ne peut s'annuler que si f passe par **tous** les points expérimentaux.

Cette première expression est satisfaisante si tous les points (x_i, y_i) sont connus avec la même précision. Souvent, et en particulier lorsqu'il s'agit de points de mesure, ce n'est pas le cas, et chaque valeur de y_i est entachée d'une incertitude σ_i[7]. Dans ce cas, il est raisonnable de pondérer les écarts entre $f_{a,b}$ et les y_i par ces incertitudes, et on pose:

$$\chi^2(a,b) = \sum_{i=1}^{n} \left(\frac{f_{a,b}(x_i) - y_i}{\sigma_i} \right)^2$$

Il arrive parfois que l'on divise le χ^2 par le nombre de points. La raison en est simple: pour chaque point (x_i, y_i) on n'a aucune raison d'obliger $f_{a,b}(x_i)$ à s'approcher de y_i à mieux que σ_i: on n'a pas d'information particulière à l'intérieur de cette "boite d'incertitude". On peut donc s'attendre à ce que, en moyenne, chaque facteur $\left(\frac{f_{a,b}(x_i)-y_i}{\sigma_i}\right)^2$ soit de l'ordre de 1. En divisant la somme par n, le χ^2 réduit final lui-même sera de l'ordre de 1. S'il est nettement plus grand, la solution n'est probablement pas très bonne. S'il est nettement plus petit, il y a probablement quelque chose de louche (au minimum, on a sur-évalué les incertitudes)[8].

Ces affirmations peuvent être quantifiées de façon précise, mais nous en parlerons après avoir vu comment calculer les "meilleures" valeurs des paramètres.

3.1.2 Optimisation

Nous supposerons maintenant que la fonction $\chi^2(a,b)$ est "suffisamment" dérivable pour ce qui suit. Pour trouver le minimum du χ^2, il "suffit" de travailler sur sa dérivée. Les valeurs de a et b recherchées sont telles que:

$$\frac{\partial \chi^2}{\partial a} = \frac{\partial \chi^2}{\partial b} = 0$$

Attention: si $f_{a,b}(x)$ n'est pas linéaire en a et b il peut y avoir plusieurs minima locaux! En revanche, f peut être non-linéaire en x sans inconvénient majeur. On obtient:

$$\frac{\partial \chi^2(a,b)}{\partial a} = \sum_{i=1}^{n} \frac{2}{\sigma_i} \frac{\partial f_{a,b}(x_i)}{\partial a} \left(\frac{f_{a,b}(x_i) - y_i}{\sigma_i} \right)$$

...et de même en b.

[7] Nous ne considérerons pas ici le cas où x_i lui-même a une incertitude.

[8] Pour le χ^2 non réduit, $< \chi^2 >= n$, $\mathrm{Var}(\chi^2) = 2n$.

Considérons d'abord le cas particulier où f est une droite. Dans ce cas:

$$f_{a,b}(x) = ax + b$$

$$\frac{\partial f}{\partial a} = x$$

$$\frac{\partial f}{\partial b} = 1$$

Le système d'équations à résoudre peut alors s'écrire:

$$\frac{\partial \chi^2(a,b)}{\partial a} = \sum_{i=1}^{n} \frac{2}{\sigma_i}\, x_i \left(\frac{ax_i + b - y_i}{\sigma_i}\right) = 0$$

$$\frac{\partial \chi^2(a,b)}{\partial a} = \sum_{i=1}^{n} \frac{2}{\sigma_i} \left(\frac{ax_i + b - y_i}{\sigma_i}\right) = 0$$

Utilisons les notations de "Numerical Recipes", chapitre 15.2:

$$S = \sum_{i=1}^{n} \frac{1}{\sigma_i^2}; \qquad S_x = \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2}; \qquad S_y = \sum_{i=1}^{n} \frac{y_i}{\sigma_i^2}$$

$$S_{xx} = \sum_{i=1}^{n} \frac{x_i^2}{\sigma_i^2}; \qquad S_{xy} = \sum_{i=1}^{n} \frac{x_i y_i}{\sigma_i^2}$$

Et on obtient, en regroupant les termes:

$$a\, S_{xx} + b\, S_x = S_{xy}$$

$$a\, S_x + b\, S = S_y$$

On a donc un système de deux équations à deux inconnues dont les coefficients peuvent être calculés directement à partir des points expérimentaux. Notons que, si les σ_i sont tous identiques, ils se simplifient. Dans la plupart des cas, il y a peu de chance qu'un tel système ait un déterminant nul ou presque. Il faut quand même être prudent. On peut cependant affirmer sans trop de risque que le problème est résolu.

Si les erreurs sur les y_i sont distribuées suivant une loi "normale" (c'est à dire une gaussienne), **alors** on peut également calculer simplement des incertitudes sur les valeurs de a et b. Si ce n'est pas le cas, il est fréquent d'utiliser quand même ces incertitudes, en particulier lorsque la loi réelle n'est pas connue. Il faut alors être prudent. Tout programme calculant un "moindre carré" **doit** obligatoirement fournir une incertitude sur les paramètres calculés. Ici, on peut montrer que ces

incertitudes sont:

$$\sigma_a^2 = \frac{S}{SS_{xx} - S_x^2}; \qquad \sigma_b^2 = \frac{S_{xx}}{SS_{xx} - S_x^2}$$

3.1.3 Evaluation du fit

Il reste à quantifier la "qualité du fit". Ici aussi, il existe une réponse précise, à condition que les hypothèses habituelles de normalité soient respectées. Le calcul fait appel à plusieurs "fonctions spéciales", bien connues en statistique ou en mathématiques appliquées. On peut démontrer que la probabilité Q que le χ^2 soit "aussi grand" qu'il l'est alors que le modèle est en réalité correct est:

$$Q(\chi^2|\nu) = Q(m, c)$$

où $m = \frac{n-2}{2}$ et $c = \frac{\chi^2}{2}$. $Q(m, c)$ est appelée "fonction Gamma incomplète". Elle est évaluée soit par un développement en série, soit par un développement en fraction continue (voir "Numerical Recipes" chapitre 6). Les principales fonctions utiles sont présentées en annexe (A).

Lorsque le nombre de degrés de liberté ν est grand, il devient très difficile d'évaluer $Q(m, c)$ numériquement (il y a du $n!$ caché dedans...). Il faut alors utiliser une relation approchée:

$$Q(\chi^2|\nu) \simeq Q(x_1), \qquad x_1 = \sqrt{2\chi^2} - \sqrt{2\nu - 1}$$

Attention. Si χ est petit, x_1 est négatif. On a alors:

$$Q(\chi^2|\nu) \simeq \frac{1}{2}\, erfc\left(\frac{x_1}{\sqrt{2}}\right) = 1 - \frac{1}{2}\, erfc\left(-\frac{x_1}{\sqrt{2}}\right) = 1 - \frac{1}{2}\, Q\left(\frac{1}{2}, \frac{x_1^2}{2}\right)$$

Si χ^2 est plus grand que ν, alors:

$$Q(\chi^2|\nu) \simeq \frac{1}{2}\, erfc\left(\frac{x_1}{\sqrt{2}}\right) = \frac{1}{2}\, Q\left(\frac{1}{2}, \frac{x_1^2}{2}\right)$$

Avec:

$$\frac{x_1^2}{2} = \chi^2 + \nu - \frac{1}{2} - \sqrt{\chi^2(\nu - \frac{1}{2})}$$

Pour une fonction $f_{a,b}$ qui n'est pas une droite, l'analyse ci-dessus se généralise aisément, mais les expressions des différents coefficients sont plus compliquées. Il devient souhaitable d'utiliser une bibliothèque de programmes bien que l'on puisse encore s'en sortir seul (à titre d'exercice pédagogique par exemple...)

3.1.4 Système non-linéaire

Pour une fonction $f_{a,b}$ quelconque, on peut rarement écrire explicitement le calcul de façon aussi simple. En général, on n'a d'ailleurs même pas de méthode simple pour calculer les dérivées partielles de f, et celles-ci doivent être évaluées numériquement (ce qui est un problème difficile dont nous parlerons plus loin dans ce cours). La méthode générale "pratique" tient en deux étapes:

1. Reformuler le problème pour le rendre aussi proche que possible d'un problème linéaire. Cela passe en général par un changement de variable. Par exemple, il est idiot de chercher à ajuster $f_{a,b}(x) = a\,\exp(-bx)$ à une série de points (x_i, y_i). Il est bien préférable de chercher à ajuster $g_{a,b}(x) = a - b\,x$ aux couples $(x_i, \log y_i)$. Malheureusement, cela n'est pas toujours possible (par exemple, dans le cas de la somme de plusieurs gaussiennes, qui est un des pires cas possibles, mais que beaucoup d'astronomes s'obstinent à vouloir traiter).

2. Utiliser **la** méthode standard, prise dans une bonne librairie: la méthode de Levenberg-Marquardt. On en trouvera un exposé succinct dans "Numerical Recipes", et il ne faut surtout pas chercher à la programmer soit même. En revanche, pour les cas les plus simples, on peut utiliser la procédure de "fit" fournie dans le programme libre "gnuplot", qui est excellente.

3.1.5 Applications

Nous avons vu que la fonction de répartition cumulée des θ_i semblait affine. Peut-on faire passer une droite à travers ces points? On peut directement utiliser gnuplot pour répondre à cette question. La syntaxe est la suivante:

```
plot "t.out" using 4:1 notitle with line
f(x) = a * x + b
a = 0.5 / pi ; b = 0.5
fit f(x) "t.out" using 4:1 via a, b ; replot
```

En quelques secondes (et malgré les 200000 points), le programme fourni les réponses suivantes:

```
After 2 iterations the fit converged.
final sum of squares of residuals : 0.0404036
```

rel. change during last iteration : -1.27226e-11

degrees of freedom (ndf) : 199998

rms of residuals (stdfit) = sqrt(WSSR/ndf) : 0.000449466

variance of residuals (reduced chisquare) = WSSR/ndf : 2.0202e-07

Final set of parameters Asymptotic Standard Error:

a = 0.159227 +/- 5.544e-07 (0.0003482%)

b = 0.500144 +/- 1.005e-06 (0.0002009%)

Ici, bien entendu, la fonction f a été initialisée avec les "vraies" valeurs de a et b (connues parce qu'utilisées pour générer les points). On s'aperçoit que l'algorithme converge très bien (ce n'est pas une surprise), et retrouve les valeurs théoriques ($a = 1/2\pi$, $b = 0.5$) avec une excellente précision. L'écart, bien que faible, est cependant significativement différent de 0. Est-il acceptable?

La probabilité Q d'avoir "par hasard" un χ^2 "aussi grand", étant donnés les points expérimentaux peut être calculée facilement, à condition de disposer d'une fonction Gamma incomplète. Encore une fois, il faut évidemment utiliser une bibliothèque de programmes. Le calcul des fonctions "spéciales" est une branche de l'analyse numérique extrêmement sophistiquée sur laquelle des spécialistes travaillent depuis des dizaines d'années et il ne faut surtout pas réinventer la roue. Ici la réponse était prévisible: la formule approchée donne:

$$Q = 1.0$$

Le χ^2 est beaucoup plus petit que le nombre de degrés de liberté; on est très largement à l'intérieur de l'intervalle défini par l'écart type.

On peut (et ici on doit) se poser une autre question, subtilement différente. Il se trouve qu'ici la fonction que l'on cherche à ajuster n'est pas un nuage de points quelconques, mais représente une fonction de répartition cumulative **et** que l'on connaît la forme exacte de la courbe que l'on **aurait du** trouver est:

$$f_{a,b}(x) = \frac{1}{2}\left(1 + \frac{x}{\pi}\right)$$

Est-ce que la distribution obtenue est susceptible de provenir de cette loi? La réponse est apportée par un test statistique différent, car il s'agit ici non pas d'examiner la cohérence interne des données, mais de les comparer à une information externe. Le test le plus courant pour cela est celui de Kolmogorov-Smirnov. Il examine l'écart maximum D_n entre la fonction de répartition cumulée "observée"

(avec n points) et la relation théorique et donne la probabilité que l'écart soit "aussi grand que ça" alors que les données viennent bien de la loi en question. Précisément, la fonction de répartition cumulée de Kolmogorov est (voir le cours de "Bruits et Signaux" de D. Pelat):

$$K(z) = \sum_{k=-\infty}^{+\infty} (-1)^k \exp\left(-2k^2z^2\right) \tag{1}$$

et

$$\Pr\{\sqrt{n}D \leq z\} = K(z)\left(1 - \frac{2k^2z}{3\sqrt{n}} + o(\frac{1}{n})\right)$$

Comme d'habitude en statistique, ce test permet de rejeter avec un certain degré de confiance une hypothèse, ici l'hypothèse que les données viennent de la loi théorique. Ici, on trouve:

$$D_{\mathrm{Obs}} = 0.0011412$$

$$Prob(D > D_{\mathrm{Obs}}) = 0.96$$

La probabilité de trouver "par hasard" un tel écart est donc extrêmement grande. On ne peut donc pas rejeter l'hypothèse ci-dessus. **Attention!** Cela ne prouve **pas** l'hypothèse inverse (c'est à dire que les données viennent bien de la loi théorique). En effet, on n'aurait par exemple aucune chance de faire la différence entre deux lois dont la différence reste en permanence inférieure à 10^{-10}... On peut seulement dire que les données expérimentales et la loi théorique sont compatibles.

Passons maintenant à la distribution des θ_i. Malgré ce qui est écrit plus haut, on peut commencer par essayer d'ajuster la fonction:

$$f_1(x) = 1 - e^{-ax}$$

Gnuplot s'en sort très bien et nous donne:

After 3 iterations the fit converged.

final sum of squares of residuals : 0.029316

rel. change during last iteration : 0

degrees of freedom (ndf) : 199999

rms of residuals (stdfit) = sqrt(WSSR/ndf) : 0.000382859

variance of residuals (reduced chisquare) = WSSR/ndf : 1.46581e-07

Final set of parameters Asymptotic Standard Error

a = 1.00557 +/- 3.166e-06 (0.0003149%)

Le Q reste toujours égal à 0.5.

La valeur théorique utilisée pour fabriquer les données était bien évidemment $a = 1$. Qu'en pense le test de Kolmogorov-Smirnov? Eh bien, il nous dit que:

$$D_{obs} = 0.002727$$

$$Prob(D > D_{obs}) = 0.10$$

Et là, on est en droit de se poser des questions... Pris au pied de la lettre, ce résultat nous annonce qu'il n'y a qu'une chance sur 10 pour que les points observés viennent bien de la distribution utilisée pour les fabriquer. L'écart est illustré sur la figure 14.

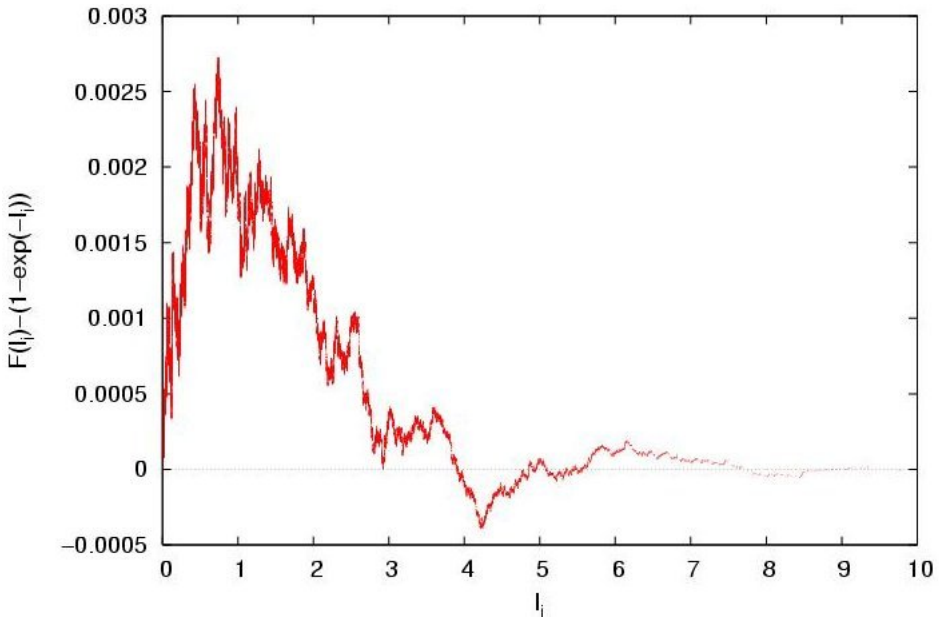


Figure 14: Test de Kolmogorov-Smirnov appliqué à l_i.

Une partie de la difficulté vient de ce qu'on ne dispose que d'un seul tirage de n points. Ici, il est facile de recommencer l'expérience en changeant l'initialisation du générateur de nombres aléatoires, et donc en ayant une série différente, mais tirée de la même loi. La figure 15 montre l'ensemble des valeurs de $p(D > D_{max})$ obtenues en fonction des D_{max} de chacun des 1000 tirages effectués.

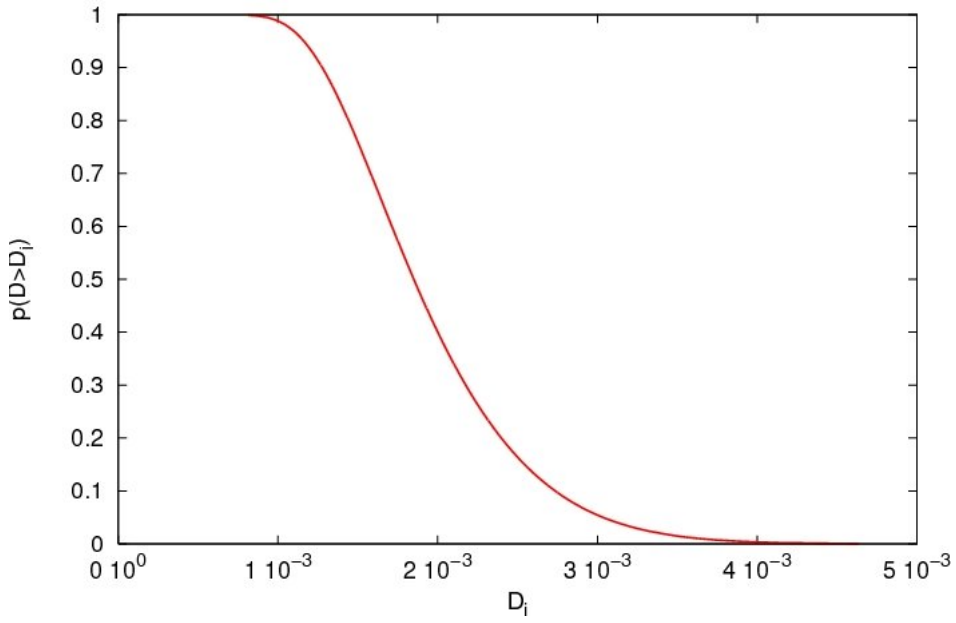


Figure 15: Test de Kolmogorov-Smirnov pour 1000 tirages différents des l_i suivant la même loi.

On peut constater trois choses:

1. La courbe obtenue reproduit bien la forme de la distribution de Kolmogorov-Smirnov.

2. Si on examine la liste ordonnée des D_{max} obtenus, on constate que la 500ème valeur correspond à une probabilité de 0.497, et le tirage produisant un p le plus proche de 1/2 est classé en position 503.

3. Le tirage initial que nous avions réalisé ($p = 0.1$) apparaît en position 108.

On constate donc que le nombre $N_>$ de tirages présentant un écart supérieur à une valeur D_0 est bien de l'ordre de $n\,p(D > D_0)$, où p est donné par la formule (1).

Changement de variable avant le fit Voyons ce qui se passe en utilisant des variables logarithmiques. Nous allons chercher à ajuster la droite $f_2 = -b\,x$ aux point $(x_i, \log(1 - y_i))$. Gnuplot nous dit:

After 2 iterations the fit converged.

final sum of squares of residuals : 5.53702

rel. change during last iteration : -1.53999e-11

degrees of freedom (ndf) : 199999

rms of residuals (stdfit) = sqrt(WSSR/ndf) : 0.00526167

variance of residuals (reduced chisquare) = WSSR/ndf : 2.76852e-05

Final set of parameters Asymptotic Standard Error

b = 1.00293 +/- 8.345e-06 (0.0008321%)

L'écart entre b et la valeur théorique (1.0) est à peu près moitié de celui trouvé à partir de l'exponentielle. On constate sur la figure 13 que le coefficient était encore plus proche de 1.0, mais le fit n'avait été fait que jusqu'à $x = 10.0$.

3.2 Exercice

Nous allons maintenant refaire la même étude, mais en utilisant une loi différente pour les longueurs des pas l_i. Les programmes proposés permettent de générer une marche au hasard de longueur donnée en utilisant différentes lois (pour l'instant 2...). La première est la loi exponentielle que nous avons examinée ci-dessus. On cherche à vérifier que la deuxième est une gaussienne (affirmation péremptoire de l'autorité pédagogique).

Si cela est vrai, alors $f_l(x) = \alpha \, \exp\left(-\frac{x^2}{\beta^2}\right)$. La normalisation nous donne (avec le changement de variable $t = x/\beta$):

$$\alpha \int_{-\infty}^{+\infty} \exp\left(-\frac{x^2}{\beta^2}\right) \, dx = \alpha \, \beta \int_{-\infty}^{+\infty} \exp\left(-t^2\right) \, dt = 1$$

Cette intégrale est standard, mais un tout petit peu plus subtile à calculer. On l'écrit sous la forme d'une intégrale double qui permet un changement de variables des coordonnées cartésiennes en polaires: $dx\,dy = 2\pi\, r\, dr$. Soit:

$$\int_{-\infty}^{+\infty} \exp\left(-x^2\right) \, dx \int_{-\infty}^{+\infty} \exp\left(-x^2\right) \, dy$$

$$= 2\pi \int_0^{\infty} \exp\left(-r^2\right) \, r \, dr = \pi \left[-\exp\left(-r^2\right)\right]_0^{\infty} = \pi$$

d'où l'on déduit que l'intégrale recherchée vaut $\alpha \, \beta \, \sqrt{\pi}$, et donc que:

$$\alpha = \frac{1}{\sqrt{\pi \beta^2}}$$

La fonction cumulée $F_l(x)$ est donc (avec $u = t/\beta$):

$$F_l(x) = \frac{1}{\sqrt{\pi\beta^2}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{\beta^2}\right) dt = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{x/\beta} \exp\left(-u^2\right) du$$

On reconnaît là la fonction erreur, définie par:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} \exp\left(-u^2\right) du$$

et telle que $\mathrm{erf}(-x) = -\mathrm{erf}(x)$ et $\mathrm{erf}(\infty) = 1$. On en déduit que:

$$F_l(x) = \frac{1}{2} \left(\frac{2}{\sqrt{\pi}} \int_{-\infty}^{0} \exp\left(-u^2\right) du + \frac{2}{\sqrt{\pi}} \int_{0}^{x/\beta} \exp\left(-u^2\right) du \right)$$

$$F_l(x) = \frac{1}{2} \left(1 + \mathrm{erf}(x/\beta)\right)$$

Nous sommes maintenant en mesure de tester la distribution proposée.

A Fonctions spéciales

A.1 Loi normale ou de Gauss

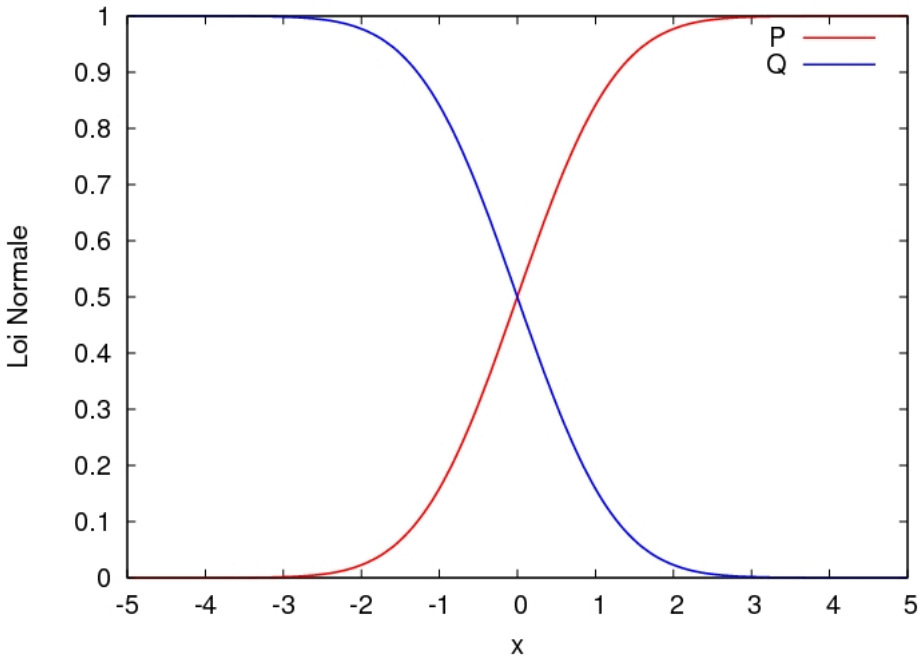
Les notations sont celle de Abramowit & Stegun (1972).

$$P(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt$$

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-\frac{t^2}{2}} dt$$

On a:

$$P(x) + Q(x) = 1$$



A.2 Fonction erreur

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$$

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2}\, dt$$

On a:

$$erf(x) + erfc(x) = 1$$

$$erf(-x) = -erf(x)$$

$$erfc(-x) = 2 - erfc(x)$$

En faisant le changement de variable $t \to \frac{u}{\sqrt{2}}$, on voit également que:

$$P(x) = \frac{1}{2}\left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right)$$

$$Q(x) = \frac{1}{2}\, erfc\left(\frac{x}{\sqrt{2}}\right)$$

A.3 Fonction Γ et Γ incomplète

$$\Gamma(z) = \int_0^\infty e^{-t}\, t^{z-1}\, dt$$

Pour n entier:

$$n! = \Gamma(n+1)$$

Et:

$$\Gamma(\frac{1}{2}) = \int_0^\infty \frac{e^{-t}}{\sqrt{t}}\,dt = 2\int_0^\infty e^{-u^2}\,du = \sqrt{\pi}$$

Fonction "incomplète":

$$P(a,x) = \frac{1}{\Gamma(a)}\int_0^x e^{-t}\,t^{a-1}\,dt$$

$$Q(a,x) = \frac{1}{\Gamma(a)}\int_x^\infty e^{-t}\,t^{a-1}\,dt$$

Cette dernière notation n'est pas universelle.
En faisant le changement de variable $t \to u^2$, on trouve que:

$$P\left(\frac{1}{2}, x^2\right) = erf(x)$$

$$Q\left(\frac{1}{2}, x^2\right) = erfc(x)$$

A.4 Loi du χ^2

$$P(\chi^2|\nu) = \left[2^{\nu/2}\,\Gamma\left(\frac{\nu}{2}\right)\right]^{-1}\int_0^{\chi^2}(t)^{\frac{\nu}{2}-1}\,e^{-\frac{t}{2}}\,dt$$

$$Q(\chi^2|\nu) = \left[2^{\nu/2}\,\Gamma\left(\frac{\nu}{2}\right)\right]^{-1}\int_{\chi^2}^{\infty}(t)^{\frac{\nu}{2}-1}\,e^{-\frac{t}{2}}\,dt$$

En faisant le changement de variable $t \to 2u$, on en déduit:

$$P(\chi^2|\nu) = P\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right)$$

$$Q(\chi^2|\nu) = Q\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right)$$

Pour les grandes valeurs de ν:

$$Q(\chi^2|\nu) \simeq Q(x_1), \qquad x_1 = \sqrt{2\chi^2} - \sqrt{2\nu - 1}$$

B Programmes fournis

Pour réaliser ce TP, on utilise trois programmes écrits en C:

debut.c Initialise le générateur de nombres aléatoires, et crée quelques fichiers auxiliaires.

unpas.c Réalise la marche au hasard.

analys.c Lit les fichiers de sortie, calcule les histogrammes et fonctions de répartition cumulées et réalise les tests statistiques

Le Makefile utilise un header "**param.h**", et un fichiers de fonctions utilitaires "**util.c**". La plupart de celles-ci sont des versions modifiées de routines tirées de Numerical Recipes. Pour utiliser l'ensemble, il faut:

1. Compiler: **make**
2. Lancer: **debut i**
3. Lancer: **unpas n1 i**

La commande "**debut**" peut prendre un argument **i**. Dans ce cas l'entier **i** est utilisé pour initialiser le générateur de nombres aléatoires. S'il n'est pas fourni, **i** = -1.

La commande "**unpas**" peut prendre deux arguments. Le premier (n1) est le nombre de pas de la marche, le deuxième sélectionne la distribution de longueur des pas. Actuellement, 0: exponentielle, 1: gaussienne. S'ils ne sont pas fournis, par défaut on fait 1 pas suivant la statistique exponentielle[9].

La commande "**analys**" prend un argument "**n1**", inférieur ou égal à celui utilisé pour "**unpas**". On l'utilise deux fois. Au premier passage, on obtient deux fichiers de sorties:

hist.out contient les histogrammes des quatre distributions, sous la forme de huit colonnes.
1 et 2: x_i et $N(x_i)$
3 et 4: y_i et $N(y_i)$
5 et 6: θ_i et $N(\theta_i)$
7 et 8: l_i et $N(l_i)$

t.out Contient les fonctions de répartition cumulées, sous la forme de cinq colonnes.
1: Ordonnées, de 0.0 à 1.0
2 à 5: x_i, y_i, θ_i, l_i

[9] Actuellement, il faut recopier "à la main" le fichier "pas.out" créé par "**unpas**" dans un fichier "pos.out" pour pouvoir utiliser directement "**analys**". Cela est lié à la procédure graphique d'animation sous **gnuplot**.

Ces fichiers permettent de tracer des histogrammes et les fonctions de répartition cumulées.

On peut ensuite (éventuellement) ajuster une fonction à l'une ou l'autre de ces courbes (à l'aide de gnuplot), puis utiliser les indications statistiques données (en plus des valeurs des paramètres) pour appliquer l'un où l'autre des tests décrits dans les paragraphes précédents. Pour cela, il est nécessaire d'éditer **analys.c** pour adapter les quelques dernières lignes (et fonctions associées), puis refaire tourner **analys n1**.

L'archive contenant les sources contient également quelques procédures shell et gnuplot permettant d'illustrer les différentes possibilités . En particulier:

gnuplot pas.gp permet de visualiser une animation de la marche au hasard. les fichiers de sortie sont manipulés en interne de façon à suivre les derniers points seulement, tout en gardant une trace de l'ensemble de la marche.

le_ks permet de faire tourner un grand nombre de fois la même marche au hasard en initialisant différemment le générateur de nombres aléatoires à chaque itération. C'est ainsi que la statistique sur le test de Kolmogorov a été obtenue.

gnuplot pdf4b.gp permet de tracer une distribution superposée à une fonction analytique qui lui est ajustée. Le paramètre ajusté est extrait du résultat (fit.log) par une commande shell et ajouté directement sur le graphique.

La combinaison d'exécutables, de commandes shell et de scripts gnuplot se révèle extrêmement souple et puissante, et permet de réaliser des "micro-applications" en quelques minutes.

References

Abramowitz, M., Stegun, I.A., 1972, "Handbook of Mathematical Functions", Dover Publications,