

Université Paul Sabatier
Toulouse III

Petit manuel
à l'usage des agrégatifs préparant
l'oral de modélisation stochastique

Djèlil CHAFAÏ

Département de Mathématiques

Laboratoire de Statistique et Probabilités

<http://www.lsp.ups-tlse.fr/Chafai/agregation.html>

<mailto:chafai@math.ups-tlse.fr>

Version primitive et pleine d'erreurs (à trouver !)
Identification du document : DJC-AGREG-1.0-2002-09-30

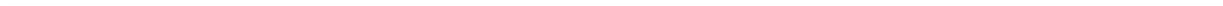
Copyright © 2002, Djélil Chafai



Permission vous est donnée de copier, distribuer et/ou modifier ce document selon les termes de la Licence « GNU Free Documentation License », version 1.1 ou ultérieure publiée par la Free Software Foundation ; sans sections inaltérables, sans texte de première page de couverture et sans texte de dernière page de couverture. Une copie de cette Licence est incluse page 115. À titre *indicatif*, une traduction française est également fournie page 118. Les noms de marques et de marques déposées cités dans ce document appartiennent à leurs propriétaires respectifs.

Historique des versions

2001-09-30	Version embryonnaire
2002-09-30	Version DJC-AGREG-1.0





Charles BABBAGE, l'un des pères fondateurs de l'informatique.
<http://www-groups.dcs.st-andrews.ac.uk/history/Mathematicians/Babbage.html>

Avant propos

Ce document a été écrit à l'origine pour les étudiants de l'Université Paul Sabatier de Toulouse préparant l'agrégation de mathématiques. Il aborde certains thèmes de probabilités et statistique pour l'épreuve de modélisation stochastique. Le texte consiste en quelques rappels de cours, le plus souvent sans démonstration, enrichis d'exemples et exercices de programmation, la plupart en Matlab. L'objectif principal est de faire découvrir aux étudiants les concepts et les problématiques qui interviennent dans le monde de la modélisation stochastique en excitant leur curiosité et leur intuition.

L'informatique est intimement liée aux mathématiques, qu'elle sert utilement, et réciproquement. Tout comme les mathématiques, elle est devenue indispensable à toutes les autres sciences. Aussi, il me semble important que les futurs enseignants en mathématiques possèdent une culture informatique et algorithmique minimale. J'espère que ce petit manuel aidera à combler quelques lacunes et donnera le goût de la modélisation algorithmique stochastique. Il n'a absolument pas la prétention d'être exhaustif, et l'on pourra se reporter aux nombreux ouvrages disponibles sur les sujets abordés.



Certains exercices de la version actuelle sont tirés des feuilles de TP d'agrégation de Jean-Marc BARDET (intervalles de confiance, ...). Je tiens à remercier chaleureusement les quelques personnes qui ont relu certaines parties du document à différentes étapes de sa mise au point. Un grand merci également à tous les agrégatifs qui ont servit de cobayes dans la joie et la bonne humeur !

Djèlil CHAFAÏ, Toulouse, juin 2002.



Ada BYRON, Lady LOVELACE. Le premier programmeur de l'histoire était une programmeuse.
<http://www-groups.dcs.st-andrews.ac.uk/history/Mathematicians/Lovelace.html>

Table des matières

Avant propos	i
Table des matières	iii
0 Initiation à Matlab	1
0.1 M-a-Ma... Matlab	1
0.2 Calculs matriciels élémentaires	3
0.3 Calculs matriciels plus élaborés	5
0.4 Générateurs aléatoires, graphiques	6
0.5 Chaînes de caractères	7
0.6 Entrées et sorties	8
0.7 Opérations logiques, boucles et exécutions conditionnelles	9
0.8 Fonctions et fichiers .m	10
0.8.1 Nombre variable d'arguments et de valeurs de retour	11
0.8.2 Laïus sur les fichiers .m et Matlab	12
0.8.3 Récursivité	13
0.8.4 Fonctions en ligne	13
0.9 Débogage & optimisation	13
0.10 Derniers conseils	14
1 Qu'est-ce que la simulation ?	16
1.1 Complexité de Kolmogorov et entropie de Shannon	16
1.1.1 Complexité de Kolmogorov	17
1.1.2 Entropie de Shannon	17
1.2 Loi uniforme et théorème fondamental de la simulation	18
1.3 Simulation de lois par leur fonction de répartition	19
1.4 Simulation de lois par la méthode du rejet	19
1.5 Simulation de lois discrètes	20
1.5.1 Lois fractales	21
1.5.2 Simulation des chaînes de Markov à valeurs dénombrables	23
1.5.3 Simulation du processus de Poisson simple	24
1.6 Simulation des lois gaussiennes	26
1.6.1 Méthode polaire pure	26
1.6.2 Méthode polaire - rejet	27
1.6.3 Méthode de Marsaglia (algorithme du Ziggurat)	28
1.6.4 Simulation d'un vecteur gaussien	28
1.6.5 Simulation d'un mouvement brownien	29
1.7 Simulation de la loi uniforme sur les p-sphères	32
1.8 Un exemple : processus de Poisson composé gaussien	34
1.9 Évaluation de la qualité d'une simulation	35
1.10 Un exemple de simulation d'une loi compliquée	35
1.10.1 Rudiments de fiabilité	36

2	Théorèmes limites classiques	39
2.1	Lois des grands nombres & théorème de Glivenko-Cantelli	39
2.1.1	Mesure empirique	40
2.2	Théorème Central Limite & principe d'invariance	41
2.3	Vitesse de convergence dans le TCL	44
2.4	Vitesse de convergence dans le théorème de Glivenko-Cantelli	45
2.5	Principes de Grandes Déviations	46
2.6	Loi du logarithme itéré	49
2.7	Convergence des histogrammes	51
2.7.1	Estimées uniformes sur les densités	52
2.7.2	Un exemple avec une illustration du théorème central limite	52
2.8	Le théorème de Wigner	53
3	Intervalles de confiance & tests statistiques	55
3.1	Rudiments de statistique paramétrique	55
3.1.1	L'exemple des sondages binaires : une approche naïve	56
3.2	Divers procédés d'obtention d'intervalles de confiances	57
3.2.1	Intervalles sur la moyenne quand la variance est connue	57
3.2.2	Intervalles sur la moyenne quand la variance est inconnue	58
3.2.3	Intervalles sur la variance	59
3.2.4	Intervalles sur la différence de deux moyennes	59
3.2.5	Méthode de Monte-Carlo	60
3.2.6	Exercices sur les intervalles de confiance	61
3.3	Tests statistiques	62
3.3.1	Tests : niveau, puissance et notions asymptotiques	62
3.3.2	Test du khi deux	62
3.3.3	Test de Kolmogorov-Smirnov	65
4	Vecteurs aléatoires gaussiens	67
4.1	Lois conditionnelles	67
4.2	Entropie de Shannon d'un vecteur aléatoire	68
4.3	Le filtre de Kalman	70
4.4	Le modèle linéaire gaussien	70
5	Chaînes de Markov & martingales	71
5.1	Chaînes de Markov à espace d'état fini	71
5.1.1	Réurrence et irréductibilité	71
5.1.2	Apériodicité	72
5.1.3	Théorème ergodique	72
5.1.4	L'exemple du modèle d'Ehrenfest	72
5.2	Martingales et temps d'arrêt	73
5.2.1	Temps d'arrêt	73
5.2.2	Convergence des martingales	76
6	Files d'attente	78
6.1	Notion de générateur infinitésimal	78
6.2	File d'attente M/M/1	79
6.3	Exercices	81
7	Évolution de la taille d'une population	82
8	Quelques lois classiques	86
8.1	Fonctions génératrices – la transformée du pauvre	87
8.2	Lois discrètes à support sur entiers naturels	88
8.2.1	Loi uniforme discrète	88
8.2.2	Loi de Bernoulli	88
8.2.3	Loi binomiale	88
8.2.4	Loi géométrique	88

8.2.5	Loi négative-binomiale	89
8.2.6	Loi de Poisson	89
8.2.7	Loi multinomiale	89
8.2.8	Loi hypergéométrique	90
8.3	Transformées de Fourier & de Laplace	90
8.4	Lois continues	91
8.4.1	Loi uniforme	91
8.4.2	Loi triangulaire	91
8.4.3	Loi gaussienne	92
8.4.4	Loi exponentielle et loi gamma	92
8.4.5	Loi double exponentielle	93
8.4.6	Loi de Dirichlet	93
8.4.7	Lois du chi-deux	93
8.4.8	Loi exponentielle et loi de Weibull	93
8.4.9	Loi de Cauchy	93
8.4.10	Loi Beta	94
8.4.11	Loi de Pareto	94
8.4.12	Loi log-normale	94
8.4.13	Loi de Fisher-Snedecor	94
8.4.14	Loi de Student	94
8.4.15	Loi de Kolmogorov-Smirnov	95
8.4.16	Statistique d'ordre, quantiles, médiane	95
8.5	Convolutions & convergences	96
A	Informations utiles (et volatiles)	100
A.1	Liste des leçons d'oral	100
A.2	Thèmes applicatifs 2001	102
A.3	Textes 2001	102
A.4	Information en provenance du jury	103
A.4.1	Utilisation des ordinateurs lors de l'épreuve Modélisation	103
A.4.2	Matériels et Logiciels disponibles	103
A.5	Boite à outils Stixbox pour Matlab	103
A.6	Boite à outils statistique de Matlab	105
B	Petite galerie de portraits	111
C	GNU Free Documentation License	115
C.1	Preamble	115
C.2	Applicability and definitions	115
C.3	Verbatim copying	115
C.4	Copying in quantity	115
C.5	Modifications	116
C.6	Combining documents	116
C.7	Collections of documents	116
C.8	Aggregation with independent works	116
C.9	Translation	117
C.10	Termination	117
C.11	Future revisions of this licence	117
D	Licence de documentation libre GNU	118
D.1	Préambule	118
D.2	Applicabilité et définitions	118
D.3	Copies conformes	119
D.4	Copies en nombre	119
D.5	Modifications	119
D.6	Fusion de documents	119
D.7	Regroupement de documents	120
D.8	Agrégation avec des travaux indépendants	120

D.9 Traduction	120
D.10 Révocation	120
D.11 Révisions futures de cette licence	120
Table des figures	121
Liste des tableaux	123
Liste des programmes	125
Bibliographie	127

Fragment 0

Initiation à Matlab

Ce qui est dit dans ce chapitre concerne Matlab 6¹, mais reste en grande partie valable pour Matlab 5 et les logiciels Octave 2.1 et Scilab 2.6. De façon générale, chaque bout de code « Matlab » figurant dans ce document fonctionne avec au moins l'un de ces logiciels, la plupart du temps avec Matlab, et parfois avec tous.

Matlab est un logiciel commercial, et je vous encourage à jeter un œil aux logiciels libres Octave et Scilab. Pour savoir ce qu'est exactement un logiciel libre, lisez la page w3 suivante :

<http://www.fsf.org/philosophy/free-sw.fr.html>

Il serait temps que les responsables de l'organisation de l'épreuve orale fassent en sorte que Octave (<http://www.octave.org/>) soit disponible le jour de l'épreuve. C'est un véritable logiciel libre, et donc en parfaite adéquation avec l'idée que l'on peut se faire de l'école publique et de l'éducation nationale. En tant que logiciel libre, il est gratuit, ce qui permet une certaine égalité d'accès à cet outil.

Matlab est un logiciel commercial de calcul matriciel développé par la société MathWorks (<http://www.mathworks.com/>). Son nom est la contraction de « Matrix Laboratory ». Il consiste essentiellement en un *interpréteur de commandes*, écrites dans un langage de programmation spécifique appelé langage Matlab. Les commandes Matlab sont saisies et interprétées ligne à ligne dans une fenêtre (console). Comme nous le verrons plus loin, elles peuvent également être regroupées dans un fichier dont le nom se termine par `.m`. Les versions récentes de Matlab comportent un éditeur de tels fichiers, qui permet d'exécuter les commandes pas à pas (débugage).

Les variables Matlab sont toutes des *tableaux*, « arrays » en anglais, définies au moment de leur affectation. Il n'y a donc pas besoin de les déclarer. Les matrices et tenseurs sont des tableaux particuliers. Un nombre complexe (ou réel ou entier) est un tableau de taille 1×1 . Le langage Matlab a été conçu pour faciliter les opérations sur les matrices.

Le langage Matlab permet de manipuler des données de différents types dont certains sont imbriqués : entiers, nombres réels, nombres complexes, caractères, booléens. Ces données peuvent être assemblées en tableaux de différents types imbriqués : vecteurs, matrices, tenseurs (matrices à plus de 2 dimensions), tableaux cellulaires, tableaux structurés. Les chaînes de caractères sont des vecteurs de caractères.

Les tableaux cellulaires et structurés ne sont pas abordés dans cette introduction. Le lecteur curieux pourra faire `help lists`, `help cell` et `help struct` pour obtenir de l'aide sur ces aspects.

Ce chapitre a été écrit dans le souci d'être à la fois intuitif et peu rébarbatif. Il peut donc manquer de rigueur pour un lecteur familier avec les présentations rationalisées des langages de programmation.

0.1 M-a-Ma... Matlab

En Matlab, les nombres réels sont représentés en virgule flottante avec 52 chiffres significatifs en base 2, soit un peu moins de 16 en base 10 (cf. `eps`). Les opérations élémentaires sur les nombres réels sont :

+ - * / \ ^

¹Testé avec Matlab 6.1.0.450 Release 12.1 sous Irix sur la machine `ondine.cict.fr` du Centre Interuniversitaire de Calcul de Toulouse.

qui représentent respectivement l'addition, la soustraction, la multiplication, la division à gauche et la division à droite, et enfin l'élévation à une puissance. Ces opérations s'appliquent également aux matrices, et dans ce cas, `\` et `/` sont différentes.

Comme nous allons le voir, Matlab différencie minuscules et majuscules dans les noms de variables. L'affectation est notée `=`. La table 0.1 donne quelques variables spéciales de Matlab, qui sont toujours définies.

<code>[]</code>	matrice vide
<code>ans</code>	dernière évaluation effectuée par Matlab
<code>computer</code>	type de machine (cf. aussi <code>ispc</code> et <code>isunix</code>)
<code>cputime</code>	temps CPU depuis le lancement de Matlab
<code>eps</code>	précision des flottants
<code>i</code> ou <code>j</code>	$\exp(i\pi/2) = \sqrt{-1}$
<code>Inf</code>	infini positif
<code>NaN</code>	pas-un-nombre (Not a Number)
<code>pi</code>	valeur de π
<code>realmax</code>	plus grand nombre réel flottant positif
<code>realmin</code>	plus petit nombre réel flottant positif

TAB. 1 – Quelques variables spéciales de Matlab.

Matlab effectue ses calculs dans l'ensemble des tableaux à plusieurs dimensions, et donc en particulier les matrices carrées ou non, et les vecteurs ligne et colonne. Les opérations sur ces tableaux ne sont faites que lorsqu'elles ont un sens bien entendu. En général, Matlab donne un sens assez intuitif aux opérations entre matrices. Cela dit, son langage contient de nombreuses spécificités que nous allons apprendre à utiliser.

Voici un exemple de code Matlab, à saisir ligne par ligne dans la fenêtre de commandes. Vous pouvez vous dispenser de saisir les lignes précédées du caractère `%` car ce sont des commentaires, et sont donc ignorées par l'interpréteur de Matlab.

```

1 % Cette ligne est un commentaire car elle est préfixée par le caractère %
2 % On crée une variable réelle nommée a, initialisée à 0
3 % Matlab affiche sa valeur une fois que l'on a appuyé sur la touche entrée
4 a=0
5 % En suffixant par un point-virgule, on évite l'affichage de la valeur de a
6 a=0;
7 % Pour connaître le contenu d'une variable, il suffit d'invoquer son nom
8 a
9 % Matlab différencie majuscules et minuscules.
10 % Ainsi, on peut créer la variables A, différente de a
11 A=1
12 % Vous pouvez rappeler les commandes précédemment exécutées au moyen
13 % des touches fléchées de votre clavier (haut et bas).
14 %
15 % Voici un calcul compliqué à base des variables a et A précédentes.
16 A*a+cos(a)/(1+sqrt(1+A^2)) % sqrt est la racine carrée (« square root »)
17 % La variable spéciale ans contient la dernière réponse de Matlab
18 % qui n'a pas été affectée à une variable par le symbole =
19 ans
20 % On peut lister les variables actuellement définies avec la commande whos
21 % On voit que pour Matlab, les variables a et A sont des matrices 1 x 1
22 whos
23 % Pour une liste plus succincte, on peut utiliser la commande who
24 who
25 % On peut détruire une variable au moyen de la commande clear
26 clear a % destruction de la variable a
27 % Vérifions que la variable a n'existe plus
28 who
29 % On peut aussi détruire toutes les variables comme suit
30 clear

```

```

31 % Vérification
32 who
33 % Pour obtenir de l'aide (en anglais !) sur une commande, on peut
34 % utiliser la commande help
35 help who
36 % La commande helpwin permet d'obtenir une liste
37 % des commandes Matlab classée par thème.
38 helpwin
39 % Valeurs particulières que l'on est amené à rencontrer
40 1/0 % Infini positif Inf
41 -1/0 % Infini négatif -Inf
42 0/0 % Not a Number (NaN)
43 0*Inf % Not a Number (NaN)
44 1/Inf % Donne bien zéro
45 pi % Donne bien le nombre pi
46 help pi % Donne sa définition pour Matlab
47 i % Racine carrée (complexe) de -1.
48 help i % Sa définition pour Matlab.
49 % On peut quand même utiliser i comme variable...

```

Remarque 0.1.1 (À l'aide!). La commande `help` permet d'obtenir de l'aide sur les commandes Matlab, `help help` donne de l'aide sur l'aide! Enfin, la commande `lookfor` permet de lister les commandes Matlab par mots clés.

Remarque 0.1.2 (Affichages). Si `X` est un tableau :

- la commande `disp(X)` affiche son contenu sans afficher son nom. S'il est vide, rien n'est affiché. La variable `X` vide est symbolisée par `[]`, on peut tester si `X` est vide grâce à la commande `isempty(X)`.
- la commande `display(X)` affichera le nom de la variable (`X` ici) ainsi que son contenu, même s'il est vide. Cette commande est utilisée automatiquement par Matlab lorsque qu'une expression ne se termine pas par un point-virgule.

```

1 % Une matrice à deux lignes et trois colonnes en spécifiant ses entrées
2 % Les lignes sont séparées par des points virgules
3 % Les entrées d'une ligne sont séparées par des virgules
4 A=[2,4,8;3,9,27]
5 % On peut accéder à une entrée particulière de la matrice A comme suit
6 A(2,3)
7 % b est un vecteur ligne contenant la deuxième ligne de A
8 b=A(2,:)
9 % c est un vecteur colonne contenant la troisième colonne de A
10 c=A(:,3)
11 % B est la sous-matrice 2x2 principale de A
12 B=A(1:2,1:2)
13 % Pour afficher à nouveau le contenu de la matrice A
14 A
15 % U est un vecteur ligne contenant les entiers de 1 à 20
16 U=[1:20]
17 % V est un vecteur ligne contenant les entiers de 1 à 20 par incrément de 2
18 V=[1:2:20]
19 % Devrait donner un vecteur ligne nul
20 V-U(1:2:20)
21 % W est un vecteur contenant les réels de Pi/2 à Pi par incrément de 0.1
22 W=[pi/2:.1:pi]

```

Les commandes comme `sqrt` qui prennent un ou plusieurs arguments (ou paramètres) entre parenthèses constituent des *fonctions*. Nous verrons plus loin comment en créer de nouvelles.

Exercice 0.1.3. À quoi peuvent servir les fonctions `fix`, `round`, `floor`, `ceil`, `mod` et `rem`? Idem avec les commandes `imag`, `conj`, `angle` et `abs`?

0.2 Calculs matriciels élémentaires

```

1 A=[3,4;1,0] % Matrice 2x2
2 B=(A>0) % B = matrice tq Bij=1 si Aij>0 et 0 sinon
3 V=[5;5] % Vecteur colonne de dimension 2, s'écrit aussi [5,5]'
4 length(V) % Renvoie la longueur du vecteur V
5 size(A) % La fonction size renvoie la taille
6 %
7 %% Opérations matricielles classiques
8 %
9 B=A' % B = transposée de A
10 A*B % Multiplication matricielle de A et de B
11 A+B % Addition matricielle de A et de B
12 A^99 % Puissance matricielle
13 expm(A) % Exponentielle matricielle
14 inv(A) % Matrice inverse
15 sqrtm(A) % Racine carrée matricielle cf. help sqrtm
16 logm(A) % Logarithme matriciel. cf. help logm
17 A*V % Image du vecteur colonne V par la matrice carrée A
18 V*A % Devinez...
19 A\V % Solution du système linéaire AX=V (par pivot, et pas par inv(A))
20 help slash % Aide explicative sur la division matricielle précédente
21 %
22 %% Opérations entrée par entrée
23 %
24 C=A.*B % C(i,j)=A(i,j)*B(i,j)
25 C=A./B % C(i,j)=A(i,j)/B(i,j)
26 C=A.^3 % C(i,j)=A(i,j)^3
27 C=3+A % C(i,j)=3+A(i,j)
28 C=3*A % C(i,j)=3*A(i,j)
29 C=A./3 % C(i,j)=A(i,j)/3
30 C=cos(A) % C(i,j)=cos(A(i,j))
31 C=log(M) % C(i,j)=log(A(i,j)). Ne pas confondre avec logm(A) !
32 C=sqrt(M) % C(i,j)=sqrt(A(i,j)). Ne pas confondre avec sqrtm(A) !
33 C=exp(A) % C(i,j)=exp(A(i,j)). Ne pas confondre avec expm(A) !
34 C=abs(A) % C(i,j)=|A(i,j)|
35 % Notez qu'à chaque fois, la variable C est redéfinie,
36 % sans que cela pose problème.
37 % Listons les variables actuellement définies
38 whos
39 % Consultons l'aide sur les opérations élémentaires
40 help ops
41 % et sur les opérations élémentaires matricielles
42 help elmat
43 %
44 % Pour finir...
45 %
46 n=4;
47 A=ones(n,n) % Matrice de taille 4x4 dont tous les éléments valent 1
48 B=zeros(n,n) % Matrice nulle de taille 4x4
49 C=eye(n,n) % Matrice identité de taille 4x4. Que donne eye(3,7) ?
50 J=[A,B,C] % J = juxtaposition horizontale des matrices A, B et C
51 K=[A;B;C] % K = juxtaposition verticale des matrices A, B et C
52 N=0*J % Astuce pour obtenir une matrice N nulle de même taille que J
53 % sans connaître la taille de J.
54 % Création d'une matrice M 4x3 « creuse » à partir d'un vecteur de données
55 % [0.5,0.1,0.2], et de vecteurs de position des données [1,2,4] et [2,1,3]
56 % Donc M(i,j)=0 sauf M(1,2)=0.5, M(2,1)=0.1 et M(4,3)=0.2
57 M = sparse ([1,2,4], [2,1,3], [0.5,0.1,0.2], 4, 3)

```

On a donc les opérations supplémentaire suivantes sur les matrices :

.* ./ .\ .^

qui opèrent entrée par entrée. Vous pouvez faire `help arith` pour de l'aide sur les opérations arithmétiques et `help @` pour de l'aide sur les opérateurs en général et les caractères spéciaux. Comme on vient

de le voir, les sous matrices s'obtiennent en spécifiant des intervalles d'indices. Si i, j, k sont des entiers relatifs, alors :

1. $i:j$ est identique à $i, i+1, \dots, j$
2. $i:j:k$ est identique à $i, i+k, i+2k, \dots, j$

L'intervalle vide est représenté par la matrice vide []. Une ligne ou une colonne entière peut être obtenue en utilisant le caractère : seul. Enfin, une matrice d'entiers E peut servir à spécifier les indices d'une matrice A, en écrivant A(E). Il ne faut pas confondre les expressions de la forme [...], qui permettent de fabriquer des matrices, avec celles de la forme M(...), qui permettent de considérer une sous-matrice de la matrice M.

Exercice 0.2.1. À quoi peuvent servir les fonctions `fliplr`, `flipud` et `sort`? Dites-vous bien que `lr` est l'abréviation de « left,right » et que `ud` est l'abréviation de « up,down ». Vous connaissez à présent `ones`, `zeros` et `eye`, à quoi correspondent `vander`, `toeplitz`, `pascal`, `magic`, `kron`, `hadamard` et `hilbert`?

0.3 Calculs matriciels plus élaborés

```

1 M=ones(3,3)+eye(3,3)
2 V=[1:10]
3 det(M) % Déterminant de M
4 rank(M) % Rang de la matrice
5 trace(M) % Trace de M
6 D=eig(M) % Renvoie le vecteur colonne des valeurs propres de M
7 [P,D]=eig(M) % Diagonalise M = P*D*P^-1 avec D diagonale
8 null(M) % Renvoie une BON du noyau (« null space » en anglais)
9 tril(M) % Partie triangulaire inférieure (lower) de M
10 tril(M,-1) % Idem sans la diagonale, cf. help tril
11 triu(M) % Partie triangulaire supérieure (upper) de M
12 triu(M,+1) % Idem sans la diagonale, cf. help triu
13 diag(M) % Diagonale de M, dans un vecteur colonne
14 diag(V) % Matrice diagonale dont la diagonale est le vecteur V
15 blkdiag(eye(3,3),M) % Matrice block diagonale de blocks eye(3,3) et M
16 %
17 sum(V) % Somme des éléments du vecteur V
18 sum(M) % Somme des colonnes de la matrice M (renvoie un vecteur ligne)
19 sum(M,2) % Somme les éléments de la matrice M selon la dimension 2
20 sum(sum(M)) % Somme totale des éléments de la matrice M
21 cumsum(V) % Sommes cumulatives des entrées du vecteur V
22 cumsum(M) % Matrice des sommes cumulatives des colonnes de M
23 cumprod(V) % Produits cumulatifs des entrées du vecteur V
24 cumprod(M) % Matrice des produits cumulatifs des colonnes de M
25 max(M) % Renvoie un vect. ligne = au max sur chaque col. de M
26 max(max(M)) % Maximum des entrée de la matrice M. Que donne min ?
27 range(V) % Renvoie l'étendue de V, c'est-à-dire max(V)-min(V)
28 %% Réplication de matrice par blocs
29 repmat(M,3,4) % Matrice par block 3x4 où chaque block = M
30 % M n'a pas besoin d'être carrée.
31 %% Redimensionnement de matrice, colonne par colonne
32 reshape([1,2,3,4],2,2) % Donne la matrice 2x2 [1,3;2,4]
33 reshape([1,2,3;4,5,6],3,2) % Donne la matrice 3x2 [1,5;4,3;2,6]
34 %% Aide sur les opérations matricielles élémentaires
35 %% et sur les fonctions matricielles
36 helpwin elmat
37 helpwin matfun

```

Exercice 0.3.1. À quoi peuvent servir les fonctions `svd`, `norm` et `chol`?

Remarque 0.3.2. La fonction Matlab `funm` permet d'évaluer une fonction sur une matrice. Vous pouvez l'ignorer en première lecture (et même en seconde). Elle nécessite la connaissance de la notion de descripteur de fonction (*function handle* en anglais). Cf. `help functions`, `help function_handle`, `help str2func` et `help func2str`.

Exercice 0.3.3. À quoi peuvent servir les fonctions `expm1`, `expm2`, `expm3`?

0.4 Générateurs aléatoires, graphiques

```

1 rand % Renvoie un nombre pseudo-aléatoire en 0 et 1 selon la loi uniforme
2 randn % Renvoie un nombre pseudo-aléatoire tiré selon la loi normale
3 % Une matrice pseudo-aléatoire 4x4 dont les entrées sont
4 % des réalisations pseudo-indépendantes de loi uniforme sur [0,1]
5 rand(4,4)
6 % Idem mais avec la loi normale
7 randn(4,4)
8 % 1000 réalisations pseudo-uniformes et pseudo-indépendantes
9 Y=randn(1000,1); % Y est un vecteur colonne ici.
10 mean(Y) % Moyenne arithmétique = ( Y(1)+...+Y(1000) ) / 1000
11 median(Y) % Médiane
12 std(Y) % Écart type normalisé en N-1
13 plot(Y) % Tracé du vecteur Y. Abscisses ? Ordonnées ?
14 plot(Y,'r-') % Autre tracé avec couleur et type de ligne
15 X=[1:2:2*length(Y)];
16 plot(X,Y,'r-') % Vous comprenez...
17 hist(Y) % Histogramme (10 classes par défaut, cf. help hist)
18 [ Effectifs ,Classes]=hist(Y,50) % Histogramme à 50 classes de Matlab
19 % Classes = centres des 50 classes
20 % L'histogramme n'est pas tracé.
21 [ Effectifs ,Classes]=histo(Y,50) % Histogramme à 50 classes de Stixbox
22 % Classes = 51 extrémités des classes
23 % L'histogramme est tracé.
24
25 %
26 % Tracé des moyennes empiriques successives, converge vers 0 d'après la LGN
27 title('Loi_des_grands_nombres') % Titre de la figure
28 xlabel('Nombre_de_réalisations_iid_normales') % Titre des abscisses
29 ylabel('Moyenne_empirique') % Titre des ordonnées
30 legend('Legende_1') % Légende
31 plot(cumsum(Y)'./[1:length(Y)],'b-') % Tracé des moyennes empiriques
32 %
33 %
34 A=eye(100,100)+rand(100,100);
35 figure % Création d'une nouvelle fenêtre graphique
36 title('Ma_matrice') % Titre de la figure
37 xlabel('Dimension_1') % Titre des abscisses
38 ylabel('Dimension_2') % Titre des ordonnées
39 imagesc(A); % Trace la matrice, couleurs = valeurs relatives,
40 hold off % On ne va pas utiliser la même fenêtre graphique
41 contour(A); % Trace les lignes de niveau

```

Exercice 0.4.1. À quoi peuvent servir les fonctions `mesh`, `bar` et `stairs` ?

Exercice 0.4.2 (Sous-graphiques). Apprenez à vous servir de la commande `subplot` qui permet de tracer des sous-graphiques.

Exercice 0.4.3. Étudiez tous les arguments des fonctions `histo` et `hist`.

Exercice 0.4.4 (Tracés de graphiques faciles ?). À quoi peuvent servir les fonctions `ezplot`, `ezcontour`, `ezcontourf`, `ezmesh`, `ezmshrc`, `ezplot3`, `ezpolar`, `ezsurf`, et `ezsurf` ?

Exercice 0.4.5. Les fonctions `sprand` et `sprandn` permettent de générer des matrices creuses aléatoires selon la loi uniforme ou normale. Elles sont en quelque sorte obtenues par croisement de `sparse`, `rand` et `randn`. À quoi peuvent servir les fonctions `speye`, `spones` ?

La bibliothèque Stixbox fournit quelques fonctions supplémentaires utiles pour faire de la simulation. Nous les utiliserons au fur et à mesure de nos avancées, tout au long de l'année. Vous en trouverez la liste page 103. Nous n'utiliserons pas la boîte à outils de statistique de Matlab, qui est pourtant bien plus complète, car elle n'est pas disponible le jour de l'épreuve. Vous en trouverez un descriptif page 105.

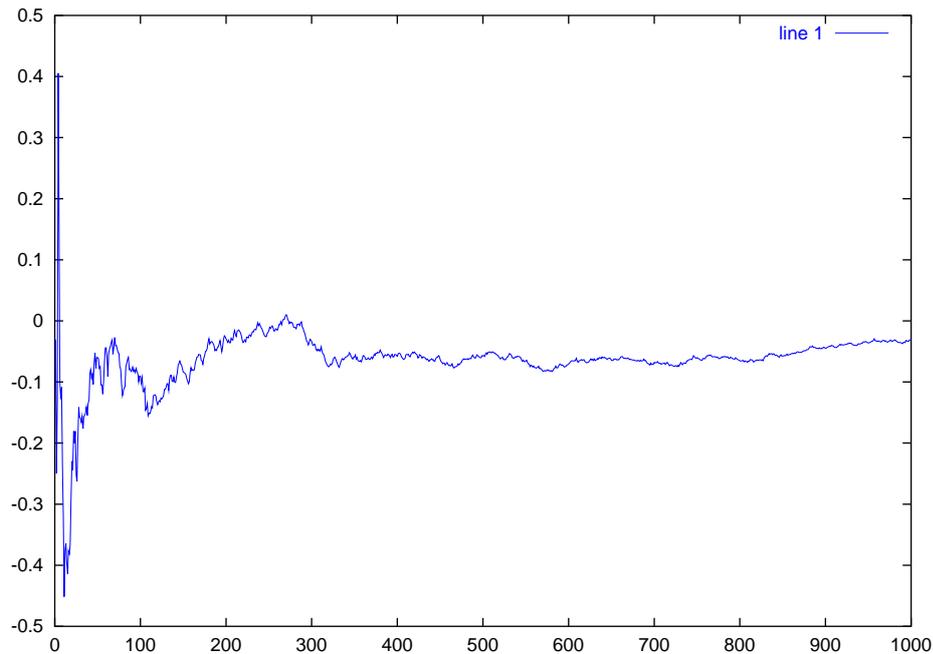


FIG. 1 – Ce que donne `plot(cumsum(randn(1,1000))./[1:1000], 'b-')`.

0.5 Chaînes de caractères

En Matlab, les caractères constituent un type de données, au même titre que les nombres réels. Ainsi, '1' représente le caractère « 1 », qui n'est pas de même nature que le nombre 1. Une chaîne de caractères est un vecteur dont les entrées sont des caractères. Une présentation de la gestion des chaînes de caractère sous Matlab est donnée en faisant `help strings`. Le code Matlab :

```
1 S='Ceci_est_une_chaine'
```

définit une variable `S` de type chaîne de caractères, qui n'est rien d'autre qu'un vecteur de caractères. En tant que vecteur, `S(1)` vaut 'C' et `S(4)` vaut 'i', tandis que `length(S)` vaut 19. La commande `ischar` permet de tester si une expression est du type chaîne de caractères. Ainsi, `ischar('1.32')` renvoie 1, alors que `ischar(1.32)` renvoie 0. Les chaînes de caractères étant des vecteurs, elles sont concaténables, ainsi, le code Matlab

```
2 ['Ceci_est', '_une_chaine']
```

donnera

```
3 'Ceci_est_une_chaine'
```

On peut également définir sans difficulté des matrices de caractères.

Il est possible de convertir un réel en une chaîne de caractères qui représente son écriture en base 10. Ainsi `num2str(1.32)` renvoie la chaîne de caractères '1.32'. L'opération inverse peut être obtenue par la commande `str2num`. La commande `dec2base` permet d'obtenir la chaîne de caractères qui représente l'écriture dans une base de numération d'un entier. Ainsi `dec2base(11,3)` renvoie la chaîne '102', qui est l'écriture en base 3 du nombre 11. Les commandes `dec2bin` et `dec2hex` ont des raccourcis pour les base 2 et 16 respectivement. La commande `base2dec` permet d'effectuer l'opération inverse : `dec2base('102',3)` renvoie 11, qui n'est pas une chaîne mais un entier donc un nombre.

La commande `char` permet d'obtenir une chaîne de caractères à partir d'un vecteur de codes ASCII. Ainsi, `char([65,66,67])` renvoie 'ABC'. Les codes supérieurs ou égaux à 127 donneront des caractères qui dépendent de la page de code du système.

Les commandes `lower` et `upper` permettent de convertir la chaîne de caractères passée en argument en minuscules et en majuscules respectivement.

La commande `sprintf` permet de construire des chaînes de caractères à partir de variables de différents types, cf. `help sprintf`. Réciproquement, la commande `sscanf` permet de décomposer une chaîne de

caractères en plusieurs variables de différents type, cf. `help sscanf`. Ces deux commandes généralisent les commandes `num2str` et `str2num` respectivement. Elles sont très pratiques. Voici un exemple explicatif :

```

1 %% Illustration de l'utilisation de sprintf.
2 clear;
3 r=1.23;
4 n=5;
5 s='blabla';
6 chaine=sprintf('r_vaut_%f,n_vaut_%d,et_s_vaut_%s',r,n,s);
7 disp(chaine)
8 %% Illustration de l'utilisation de sscanf.
9 clear;
10 chaine='1.2 5.5 3.8 51';
11 X=sscanf(chaine,'%f;%f;%f;%d;%s');
```

0.6 Entrées et sorties

La commande `input` permet de demander à l'utilisateur de saisir des valeurs de variables. La commande `pause` permet de stopper l'exécution de Matlab pendant un temps déterminé, cf. `pause`. La forme spéciale `pause off` désactive les pauses tandis que `pause on` les réactive.

La commande `save` permet de sauvegarder le contenu des variables en cours ainsi que leur nom dans un fichier, dont le nom est par défaut `matlab.mat`. Ce fichier peut être lu par la commande `load`, qui restaure donc toutes les variables.

```

1 % La commande input permet d'afficher une question et de récupérer la
2 % réponse dans une variable
3 n=input('Donnez la valeur de n: ')
4 %
5 % La commande sprintf permet de faire un affichage formaté de chaînes
6 % de caractères comprenant des variables. Faire help sprintf.
7 %
8 entier=1;
9 reel=pi;
10 sprintf('Voici un entier %i et un réel %12.8f\n',entier,reel)
11 %
12 %% Sauvegarde et lecture des variables dans des fichiers
13 %
14 % La commande save permet de sauvegarder une variable dans un fichier
15 A=rand(1000,10);
16 save 'mata' A; % La matrice A est sauvegardée dans le fichier mata.dat
17 % La commande load permet de lire un fichier de données
18 clear A % On détruit A
19 load 'mata' % On lit le fichier mata.dat
20 whos % A apparaît de nouveau
21 % Pour sauvegarder A sous forme de fichier texte
22 % (le nom de la variable est perdu)
23 save -ASCII 'mata.txt' A
24 % Pour lire la matrice dans un fichier texte et l'affecter à une variable
25 B=load('mata.txt')
26 % Pour sauvegarder toutes les variables en cours
27 save % Sauvegarde toutes les variables dans le fichier matlab.mat
28 load % Pour restaurer les variables sauvegardées
```

Les commandes `fprint` et `fscanf` permettent de faire des lectures et écritures formatées vers des fichiers de données :

```

1 %% Exemple d'entrées sorties sur un fichier
2 clear;
3
4 % Une matrice
5 position=rand(1,10);
6 vitesse=rand(1,10);
```

```

7
8 % Ouverture du fichier en mode réécriture (w+)
9 % Le contenu précédent est détruit
10 id=fopen('donnees.txt','w+');
11 % Ecriture des variables dedans
12 fprintf(id,'Position_=%f,vitesse_=%f\n',position,vitesse);
13 % Fermeture du fichier
14 fclose(id);
15
16 % Jetez un coup d'oeil au fichier donnees.txt !
17
18 % Ouverture du fichier en mode lecture
19 id=fopen('donnees.txt','r');
20 % Lecture de 10x10 éléments
21 PV=fscanf(id,'Position_=%f,vitesse_=%f\n',[10 10]);
22 % Fermeture
23 fclose(id);
24
25 % Comparer PV et [position',vitesse']

```

0.7 Opérations logiques, boucles et exécutions conditionnelles

Pour Matlab, tout nombre peut être considéré comme une valeur logique, en identifiant les nombres non nuls à *vrai* (noté 1) et le nombre zéro à *faux* (noté 0). La commande `boolean` permet d'obtenir le booléen associé à un nombre vu comme une valeur logique. Les principales opérations sur les valeurs logiques sont :

- la négation logique `~`
- le « ou » logique `|`
- le « et » logique `&`

On peut également utiliser les respectivement les fonctions `or`, `not` et `and`. Il y en a d'autres, faites donc un `help ops`. On peut créer des valeurs logiques à partir de nombres avec les opérations de comparaisons, qui activent automatiquement la nature logique de leurs arguments :

- « a égal à b » s'écrit : `a == b`
- « a différent de b » s'écrit : `a ~= b`
- « a supérieur strictement à b » s'écrit : `a > b`
- « a supérieur ou égal à b » s'écrit : `a >= b`
- « a inférieur strictement à b » s'écrit : `a < b`
- « a inférieur ou égal à b » s'écrit : `a <= b`

On peut également utiliser les respectivement les fonctions `eq`, `ne`, `gt`, `ge`, `lt` et `leq`. Le code Matlab `R=randn(5,5)` ; `M=(R>2.1)` crée une matrice M de même taille que R qui contient des 0 là où R est inférieure à 2.1 et des 1 là où R est supérieure ou égale à 2.1. C'est donc une matrice « booléenne ». La commande `any(V)` renvoie 1 si au moins l'un des éléments du vecteur V est non nul, et 0 sinon. La fonction `all(V)` renvoie 1 si tous les éléments du vecteur V sont non nuls, et 0 sinon. Enfin la fonction `find(V)` renvoie les indices correspondants aux éléments non nul du vecteur V. Ainsi, `V(find(V>2))` renvoie les valeurs supérieures à 2 du vecteur V.

Les commandes Matlab `isreal`, `ischar`, `issparse`, `isnan`, `isinf`, `isfinite` permettent de tester la nature des variables. Ainsi, `isreal(a)` renverra 1 si la variable a est un réel et 0 sinon.

```

1 %% Exemples de calculs logiques
2 V=[1,0,-3,3.4]; % Un vecteur de nombres
3 boolean(V) % Doit donner [1,0,1,1]
4 ~V % Doit donner [0,1,0,0]
5 ischar(V) % Doit donner 0
6 isreal(V) % Doit donner 1
7 isfinite(V) % Doit donner [1,1,1,1]
8 isinf(1./V) % Doit donner [0,1,0,0]
9 isnan(cos(1./V)) % Doit donner [0,1,0,0]
10 V>0.5 % Doit donner [1,0,0,1]
11 (V>-10 & V<=3) % Doit donner [1,1,1,0]
12 (V>-10 | V<=3) % Doit donner [1,1,1,1]

```

```

13 find(V>0.5)      % Doit donner [1,4]
14 V(find(V>0.5)).^2 % Doit donner [1.0,11.56]
15 any(V)          % Doit donner 1
16 all(V)          % Doit donner 0
17 % Aide sur les opérations élémentaires
18 help ops

```

Exercice 0.7.1. Que donne `find(M)` pour une matrice `M`? Que donne alors `[I,J]=find(M)`? `[I,J,K]=find(M)`?

Le langage Matlab comprend les boucles du type `for` et `while` ainsi que les structures d'exécutions conditionnelles du type `if`. L'argument d'un `if` et d'un `while` est de type logique, ce qui n'est pas le cas de celui d'un `for`, qui est de type matriciel. La commande `break` permet de sortir de la boucle `while` où `for` la plus interne.

```

1 % Boucles FOR
2 N=10;
3 for n=1:N,
4   for m=1:N,
5     A(n,m)=1/(n+m+1);
6   end
7 end
8
9 % Boucle WHILE (TANT QUE en français)
10 n=10;
11 r=-Inf;
12 while (A(n,n)>0),
13   r=A(n,n);
14   n=n+1;
15 end
16
17 % Exécution conditionnelle IF
18 if n==m
19   A(n,m)=1;
20 elseif abs(n-m)==1
21   A(n,m)=-1;
22 else
23   A(n,m)=0;
24 end
25
26 % Exécution ESSAI/ERREUR/INTERCEPTION
27 % Tente de déterminer l'inverse de M
28 % Affiche NaN si ça n'est pas possible
29 % Aucune erreur n'est provoquée.
30 % À tester avec M=eye(2) et M=ones(1,2)
31 try, inv(M), catch, disp(NaN); end

```

0.8 Fonctions et fichiers .m

Les *commandes* Matlab peuvent toutes être considérées comme des *fonctions*, c'est-à-dire des *entités nommées*, qui prennent des paramètres éventuels (arguments) et qui renvoient des résultats (valeurs de retour). Ce sont en quelque sorte des sous-programmes, bien que la notion de programme n'ait pas beaucoup de sens pour un interpréteur de langage comme Matlab.

Beaucoup de fonctions Matlab, comme `mean` par exemple, sont en réalité écrites en Matlab, et le code Matlab correspondant est stocké dans un fichier dont le nom se termine par `.m`. Pour `mean`, il s'agit de `mean.m`. Pour ajouter de nouvelles fonctions à Matlab, il nous suffit d'écrire de nouveaux fichiers de ce type. Vous aurez compris qu'un fichier `.m` ne contient qu'une seule fonction, qui a le même nom que le fichier, au suffixe `.m` près.

Voici un code qui définit une fonction `stat`, qui prend comme paramètre un vecteur `x` et qui renvoie sa moyenne et son écart type. Ce code devra être stocké dans le fichier nommé `stat.m` pour que Matlab fasse le lien avec la fonction `stat`. Le commentaire de la ligne n°2 constitue l'aide qui est affichée lorsque l'utilisateur tape `help stat`.

```

1 function [moyenne,ecartype] = stat(x)
2 %STAT Revoie la moyenne et l'écart type du vecteur passé en argument.
3 n = length(x);
4 moyenne = sum(x) / n;
5 ecartype = sqrt(sum((x - moyenne).^2)/n);
6 return % En fait, inutile en fin de fonction...

```

Le fichier stat.m devra être placé dans un répertoire que Matlab scrutera. En général, Matlab cherche automatiquement dans le répertoire en cours. Pour afficher le répertoire en cours, utilisez la fonction `pwd`, pour lister son contenu, utilisez la fonction `ls` et pour changer de répertoire courant, utilisez la fonction `cd`.

Remarque 0.8.1 (Sous-fonctions). On peut définir des fonctions dans le corps d'une fonction. Elles ne seront visibles que pour la fonction principale, et l'on parle alors de sous-fonctions, cf. `help function`. Vous pouvez vous en passer en première lecture.

0.8.1 Nombre variable d'arguments et de valeurs de retour

Il est parfois commode d'écrire des fonctions qui prennent un nombre variable d'arguments. Ceci peut être fait en Matlab grâce à la commande `nargin` qui renvoie le nombre d'arguments passés à la fonction lors de son appel. Voici un exemple :

```

1 function fd = firstdigit(x,b,convert)
2 %fd = firstdigit(x,b)
3 % Renvoie le premier chiffre non nul dans l'écriture en base b
4 % de x, en partant de la gauche.
5 % * x doit être un vecteur de réels >0 pas trop grands.
6 % * b doit être un entier entre 2 et 36
7 %   36 = 10 + 26, histoire de pouvoir écrire avec l'alphabet...
8 %   ce paramètre est optionnel et vaut par défaut 10.
9 % * convert est un paramètre optionnel valant par défaut 0.
10 % si convert n'est pas nul, la sortie fd est un entier correspondant
11 % à la valeur en base 10 du premier chiffre, alors que si convert est
12 % nul (par défaut), la sortie fd est un caractère alphanumérique.
13
14 %% base 10 par défaut
15 if (nargin==1), b=10; end
16 %% si dépassement, on sort avec une erreur
17 if ~(ceil(b)==b & b>1 & b<37)
18   error('b_n''est_pas_un_entier_entre_2_et_36');
19 end
20 %% si x n'est pas positif, on prend sa valeur absolue
21 if ~(x>0)
22   warning('x_n''est_pas_positif,_il_est_remplacé_par_-x');
23   x=-x;
24 end
25 x=reshape(x,[1,length(x)]);
26 %% décalage de x en base b tel que y>=1 en base b
27 y=x.*b.^ceil(-log(x)/log(b));
28 %% si dépassement, on sort avec une erreur
29 if isinf(y) | isnan(y) | ~isreal(y),
30   error('x_trop_petit_ou_trop_grand_par_rapport_à_b');
31 end
32 c=floor(y);           % c = premier chiffre en base 10
33 s=dec2base(c,b);     % conversion en base b
34 if nargin==3,
35   if convert,
36     fd=base2dec(s(:,1),b); % valeur en base 10 du premier caractère
37   end
38 else
39   fd=s(:,1);         % Premier caractère en base b.
40 end
41 return;

```

De la même manière, on peut définir grâce à `nargout` des fonctions qui renvoient un nombre variable de valeurs de retour, un peu comme le fait la fonction `eig` que nous avons déjà vue. Voici un exemple de fonction Matlab, qui renvoie une approximation constante par morceaux de la fonction de répartition empirique d'un échantillon aléatoire, en utilisant la fonction Matlab `hist` :

```

1 function [ecdf,mi,ma,dt,rg] = empcdf(e,n)
2 %EMPCDF renvoie une approximation constante par morceaux de la
3 %fonction de répartition empirique d'un échantillon.
4 %[ecdf,min,max] = empicdf(e,n)
5 % * e est un échantillon (vecteur de réalisations i.i.d.)
6 % * n est un entier optionnel valant par défaut 10.
7 % * L'intervalle [min(e),max(e)] est subdivisé en n sous-intervalles
8 % de même longueur. ecdf est un vecteur de longueur n.
9 % ecdf(i) représente la valeur de la fonction de répartition
10 % empirique à l'extrémité droite du i-ème sous-intervalle (classe).
11 % * Les valeurs de retour mi,ma,rg et dt sont optionnelles
12 % et correspondent respectivement à l'étalement de l'échantillon et au
13 % pas de discrétisation associé à n.
14 % * Les sorties peuvent être exploitées pour un tracé d'une approximation
15 % affine par morceaux de la fonction de répartition empirique :
16 % plot([mi+dt:dt:ma],ecdf)
17 %EMPCDF fait appel à HIST.
18 if (nargout==1), n=10; end;
19 [frequences, classes]=hist(e,n,1); %histogramme normalisé en n classes.
20 ecdf=cumsum(frequences);
21 if (nargout >= 2), mi=min(e); end
22 if (nargout >= 3), ma=max(e); end
23 if (nargout >= 4), dt=(ma-mi)/n; end
24 if (nargout >= 5), rg=ma-mi; end %i.e. rg=range(e)
25 return;

```

0.8.2 Laïus sur les fichiers .m et Matlab

La commande Matlab `type` permet de lister le contenu du fichier `.m` d'une fonction. Ainsi, `type rbinom` va vous montrer le code source Matlab de la fonction Stixbox `rbinom`. Un certain nombre de fonctions Matlab ne correspondent à aucun fichier `.m`, elles sont « internes » à Matlab pour une plus grande efficacité et l'on parle de fonctions « built-in » en anglais. C'est par exemple le cas de la fonction `type` elle-même ! Cette séparation entre fonctions internes et externes se retrouve dans la plupart des interpréteurs de langages. La commande `exist` permet de connaître le type d'une commande où d'une variable, cf. `help exist`.

Sur la plupart des systèmes informatiques, les fichiers sont organisés en une structure arborescente de répertoires qui contiennent des fichiers et des sous-répertoires. La commande Matlab `which` donne le chemin du fichier `.m` associé à une fonction Matlab, lorsqu'il existe. Par exemple, `which mean -ALL` affichera tous les fichiers `mean.m` avec leurs chemins. Seul le premier sera utilisé par Matlab ! La commande `what` liste les fichiers `.m` du répertoire en cours. Faites `help what` et `help which` pour plus d'information.

Matlab a besoin de savoir où chercher les fichiers `.m` dans l'arborescence des fichiers du système. Il recherchera automatiquement dans une liste de répertoires appelée « PATH ». La commande `path` vous liste les répertoires du « PATH ». Les commandes `addpath` et `rmpath` permettent d'ajouter et d'enlever des répertoires de la liste « PATH ». Ces modifications ne sont pas permanentes et sont perdues lorsque vous quittez Matlab.

Par défaut, la liste « PATH » contient les répertoires de Matlab, qui contiennent eux-même les fichiers `.m` qui constituent Matlab. Si Stixbox a été correctement installée, le répertoire qui contient les fichiers `.m` qui la constituent a été ajouté au « PATH » par l'administrateur système. Afin d'accélérer la lecture des fichiers `.m`, Matlab maintient en permanence une base de données des fichiers `.m` avec le répertoire associé qui les contient. La commande `rehash` permet de mettre à jour cette base de données en relisant la liste « PATH ».

Il faut enfin savoir que Matlab cherche d'abord les fichiers `.m` des fonctions dans le répertoire courant (« working directory »). Répétons-le, la commande `pwd` affiche le répertoire courant (« print working

directory »), la commande `cd` change de répertoire courant (« change directory »), et les commande `ls` ou `dir` listent le contenu du répertoire courant.

En général, vous n'aurez pas à modifier la liste « PATH ». Sa modification permanente dépend du système utilisé (Unix, MS-Windows, ...).

0.8.3 Récursivité

Le langage Matlab autorise la récursivité, comme le montre l'exemple archi-classique suivant :

```

1 function fn=fact(n);
2 %FACT calcule de façon récursive et inefficace la factorielle de n
3 %fn=fact(n);
4 if (n<=0), fn=1; else fn=n*fact(n-1); end;
5 return;

```

0.8.4 Fonctions en ligne

Pour des fonctions très légères, on peut procéder autrement que par la création d'un fichier `.m` : la directive `inline` permet de créer des fonction nouvelles à la volée. Voici un exemple :

```

1 % On définit une nouvelle fonction trisin qui prend 3 paramètres a,b et c
2 trisin=inline('sin(a)*sin(b)*sin(c)');
3 % On teste notre nouvelle fonction
4 trisin(pi/2,pi/4,pi/2)

```

Matlab détermine automatiquement le nombre de paramètres de la fonction. Bien entendu, une fois que vous quitterez Matlab, cette fonction sera perdue car elle n'est pas sauvegardée dans un fichier.

Exercice 0.8.2. Apprenez à utiliser la fonction `fplot` pour tracer les graphes de fonctions définies avec `inline`.

0.9 Débogage & optimisation

Diverses commandes permettent de déboguer les programmes Matlab. Citons par exemple `dbstack`, `dbstep`, `dbstop`, `dbcont`, `dbclear`, `dbtype`, `dbup`, `dbdown`, `dbstatus` et `dbquit`. Leur aide vous renseignera sur leur usage.

La commande `keyboard` permet à l'utilisateur de reprendre le contrôle en interrompant momentanément l'interpréteur Matlab, afin par exemple d'examiner le contenu des variables, cf. `help keyboard`.

Dans une fonction, la commande `error` permet d'afficher un message d'erreur et provoque la sortie immédiate de la fonction, tandis que la commande `warning` permet d'afficher un message d'avertissement, sans pour autant interrompre l'exécution de la fonction. Les messages d'avertissement peuvent être désactivés, cf. `help warning`. Les commandes `lasterr` et `lastwarn` permettent d'afficher respectivement le dernier message d'erreur et le dernier message d'avertissement. La commande `mfilename` permet de connaître le fichier `.m` qui contient la fonction en cours d'exécution.

Les commandes `tic` et `toc` permettent de déterminer le temps d'exécution d'une commande, ce qui est très utile pour l'optimisation et la comparaison lors du test de différentes méthodes. Voici un exemple dont la sortie graphique est donnée par la figure 2 :

```

1 % Comparaison de la rapidité de plusieurs méthodes
2 % de calcul de la factorielle grace a tic & toc.
3
4 clear; nmin=2; nmax=150; N=[nmin:nmax];
5 for n=N,
6     m=n-nmin+1;
7     tic; fact(n); T(m,1)=toc;
8     tic; f=nmin; for i=nmin+1:n; f=f*i; end; T(m,2)=toc;
9     tic; cumprod(nmin:n); T(m,3)=toc;
10    tic; gamma(n-1); T(m,4)=toc;
11 end
12 clf; hold on; title('Calcul_de_la_factorielle');
13 plot(N,T(:,1),N,T(:,2),N,T(:,3),N,T(:,4));

```

```

14 xlabel('n'); ylabel('Temps');
15 legend('Recuratif', 'Boucle', 'cumprod', 'gamma_built-in', 2);

```

La commande `cputime` permet d'obtenir le *temps machine* utilisé par Matlab depuis son lancement, cf.

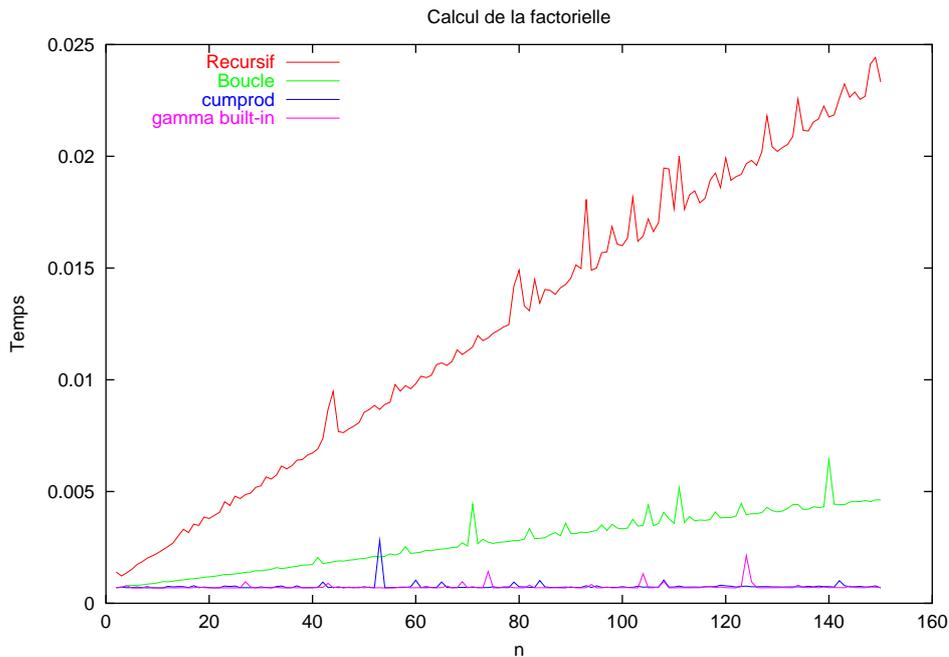


FIG. 2 – Comparaison des temps de calcul de la fonction factorielle (Octave 2.1).

`help cputime.`

Exercice 0.9.1. Pourquoi les tracés sont linéaires dans l'exemple précédent ?

Exercice 0.9.2. Utiliser `cmtime` en lieu et place de `tic` et `toc` dans l'exemple précédent.

0.10 Derniers conseils

Nous verrons, tout au long des travaux pratiques, plusieurs autres fonctions Matlab dont il n'a pas été question dans ce chapitre. Voici quelques petits conseils avant de vous lancer seul :

1. Expérimentez vous-même, en vous servant de l'aide au moyen des fonctions `help` et `lookfor`.
2. La combinaison de touches `Ctrl-c` (appuyer sur la touche `Ctrl` et sur la touche `c` en même temps) permet d'interrompre l'exécution de la commande en cours, ce qui peut être très utile pour sortir d'une longue boucle... La touche `tab` permet de compléter la commande que l'on a commencé à saisir.
3. N'oubliez pas de désactiver l'affichage d'une commande en la suffixant par un point-virgule, surtout dans une boucle ou pour une grosse matrice.
4. Sauvegardez régulièrement votre code Matlab dans un fichier pour éviter de le perdre en cas de problème.
5. Dites-vous bien que vous vous trompez bien plus souvent que Matlab !
6. Le langage Matlab étant par défaut interprété, il est souvent plus efficace d'adopter une formulation matricielle en lieu et place de boucles pour certains calculs. Cependant, les boucles sont parfois plus lisibles, à vous de voir.
7. Si vous disposez d'un micro-ordinateur PC chez vous, je vous conseille d'installer Scilab ou Octave, qui sont plus ou moins libres et ressemblent à Matlab. La licence de ce dernier est chère et je ne peux pas vous encourager à le pirater...
8. Exercez-vous tout au long de l'année pour vous familiariser et réfléchir à des développements, n'attendez pas le lendemain des écrits pour vous y mettre !

9. Dites-vous bien que le jury n'attend pas des « petits génies » de l'informatique, mais plutôt des candidats qui l'épatent avec des mathématiques appliquées...
10. Ne prenez pas cette liste pour les dix commandements.

Un dernier petit exercice avant de clore ce chapitre introductif :

Exercice 0.10.1. Comprenez-vous le code suivant ?

```
1 clear;
2 n=100; a=1; b=2;
3 A= repmat([-n:n].^2, 2*n+1, 1);
4 B=(sqrt(a*A+b*A')<n-1)+(sqrt(a*A+b*A')>n+1);
5 figure(1); clf; imagesc(B);
```

Fragment 1

Qu'est-ce que la simulation ?

Considérons une suite de chiffres $x_0, x_1, x_2, \dots, x_n$ de l'intervalle $[0, 1]$ ou d'entiers entre 0 et 9. Quand peut-on dire qu'elle est constituée de réalisations indépendantes d'une loi uniforme sur $[0, 1]$ (resp. sur $\{0, \dots, 9\}$) ? Et bien dans tout les cas bien sûr. Cependant, l'intuition nous indique que ce sont les propriétés « statistiques » des suites qui sont importantes, et pas celles d'une suite donnée... Pour certains, une suite qui peut être générée par un algorithme n'est pas aléatoire. De ce point de vue, une suite finie n'est jamais aléatoire ! Cf. [Bou86, chap. IV, page 73] et [Knu81]. Les notions de complexité de Kolmogorov et d'entropie de Shannon, brièvement présentées dans les sections 1.1.1 et 1.1.2 qui suivent, permettent de mieux définir et quantifier le caractère « désordonné » d'une suite.

Pour nous, simuler une loi de probabilité \mathcal{L} consistera à écrire un programme informatique pouvant générer des suites finies dont on considèrera que ce sont des réalisations indépendantes de loi \mathcal{L} . Bien entendu, comme nous ferons appel le plus souvent à des algorithmes déterministes, les suites générées ne sont pas du tout aléatoires et encore moins indépendantes. On parlera de suites *pseudo-aléatoires*. Cependant, elles se montreront suffisantes dans la pratique et l'on vérifiera expérimentalement plus loin qu'elles ont de « bonnes » propriétés en utilisant des tests statistiques.

La loi uniforme est la première loi que l'on simule. Diverses méthodes permettent ensuite de simuler à partir de la loi uniforme, un grand nombre de lois, plus ou moins quelconques. Dans la pratique, on met en œuvre des algorithmes rapides de génération de suites pseudo-aléatoires.

Matlab permet de simuler une loi uniforme (resp. normale) via la fonction `rand` (resp. `randn`). La bibliothèque Stixbox fournit de nombreuses autres simulations de lois discrètes et continues qui en sont dérivées : faites `help stixbox` ou reportez vous à la table 1.1, page 38, qui donne la liste des générateurs aléatoires de la bibliothèque Stixbox.

1.1 Complexité de Kolmogorov et entropie de Shannon

Les sections 1.1.1 et 1.1.2 peuvent être ignorées en première lecture. Les notions évoquées sortent du cadre du programme de l'oral. Elles donnent cependant des idées et enrichissent l'intuition probabiliste. Commençons par un exemple tout simple. La suite de dix chiffres suivante

1, 1, 1, 1, 1, 1, 1, 1, 1, 1

nous semble beaucoup moins complexe que la suite

1, 2, 3, 1, 0, 9, 0, 8, 0, 5.

Pour coder brièvement la première, il suffit d'écrire « dix fois un », alors que la seconde nécessite plus de « détails ». On peut également observer que ce phénomène se constate immédiatement sur les fréquences des chiffres dans ces deux suites. En revanche, contrairement à un algorithme de génération, les fréquences ne contiennent aucune information sur la structure de la deuxième suite, et l'information retenue est donc pauvre, à moins de faire intervenir les fréquences d'apparition des couples et plus généralement des n -uplets.

Ce sont ces deux idées qui sont à la base des notions de complexité de Kolmogorov et d'entropie de Shannon : la complexité pour le versant algorithmique de description minimale et l'entropie pour le versant probabiliste lié aux fréquences d'apparition. Nous allons voir de plus que contrairement à ce que l'on pourrait croire, ces deux notions sont intimement reliées.

Pour la petite histoire, Shannon a introduit l'entropie qui porte son nom à la fin des années 1940 – en s'inspirant de l'entropie thermodynamique de Boltzmann en théorie cinétique des gaz – pour les besoins de ce qui sera appelé plus tard théorie de l'information ou de la communication. Kolmogorov, père fondateur des probabilités modernes (i.e. basées sur la théorie de la mesure), a repris et développé cette idée d'entropie-information pour l'étude des systèmes dynamiques en particulier. Ses élèves ont beaucoup contribué à donner un cadre mathématique rigoureux à la théorie de l'information. La notion de complexité de Kolmogorov constitue une tentative de quantification de l'aléatoire par une approche algorithmique.

1.1.1 Complexité de Kolmogorov

Considérons une machine informatique M pouvant exécuter des programmes. On dit que cette machine est universelle lorsqu'elle peut émuler n'importe quelle autre machine informatique. La machine universelle de Turing en est un exemple [TG95].

On note \mathcal{P}_M l'ensemble des programmes écrits pour la machine M . Pour un programme $p \in \mathcal{P}_M$, on note $l(p)$ sa longueur en nombre d'instructions pour la machine M et $s(p)$ sa sortie. La complexité de Kolmogorov $\mathbf{K}_M(x)$, ou complexité algorithmique, d'une suite $x := (x_i)_i$ pour une machine M est définie par :

$$\mathbf{K}_M(x) := \min_{p \in \mathcal{P}_M, s(p)=x} l(p).$$

C'est donc la longueur du plus petit programme écrit pour la machine M qui génère la suite x . Une suite constante a une complexité faible car les programmes qui la génèrent peuvent être très courts.

Reste à savoir dans quelle mesure la fonction \mathbf{K}_M dépend de la machine M , car on peut tout à fait imaginer une machine possédant des instructions simples pour générer certaines suites complexes. La réponse est la suivante : soit U une machine universelle et M une machine, alors il existe une constante c_M telle que pour toute suite x , on ait :

$$\mathbf{K}_U(x) \leq \mathbf{K}_M(x) + c_M.$$

On parle alors d'universalité de la complexité de Kolmogorov, en ce sens qu'elle ne dépend pas, à une constante additive près, de la machine considérée. Ainsi, on peut se ramener à une machine universelle dont la définition est élémentaire (celle de Turing). Une suite peut alors être considérée comme d'autant plus « aléatoire » que sa complexité est grande par rapport à sa taille. De ce point de vue, les décimales des nombres π , e ou $\sqrt{2}$ ne sont pas vraiment aléatoires puisqu'il existe des algorithmes très simples pour les générer. Cela dit, on montre qu'il n'existe pas d'algorithme pour calculer la complexité de Kolmogorov en toute généralité ! On peut cependant obtenir des encadrements et des estimations. Mais arrêtons là cette digression culturelle, elle nous mènerait trop loin ! Pour en savoir plus, consultez par exemple [CT91, chap. 7].

1.1.2 Entropie de Shannon

Une autre approche au « désordre » d'une suite est celle de Shannon. L'entropie de Shannon d'une loi de probabilité discrète

$$P := p_1\delta_1 + \dots + p_n\delta_n$$

est définie par :

$$\mathbf{H}(P) := \sum_{i=1}^n p_i \log \frac{1}{p_i}.$$

Elle est maximale pour la loi uniforme pour laquelle elle vaut $\log n$ et minimale pour les masses ponctuelles pour lesquelles elle est nulle (car la fonction $x \geq 0 \mapsto x \log x$ est nulle en 0 et 1). Elle mesure donc le désordre en quelque sorte. On pourra consulter [App96, chap. 6] pour plus d'explications.

Soit $m \in \mathbb{N}^*$ un entier non nul, U une machine universelle et \mathcal{P}_m l'ensemble des programmes écrits pour cette machine dont la longueur de la sortie est de m . On note $\mathbf{K}(x|m)$ la complexité de Kolmogorov de taille m d'une suite $y = (y_1, \dots, y_m)$ de longueur m , définie par :

$$\mathbf{K}(y|m) := \min_{p \in \mathcal{P}_m, s(p)=y} l(p).$$

Soit à présent une suite de variables aléatoires i.i.d. $(X_i, i \in \mathbb{N})$ de loi discrète $P := p_1\delta_{x_1} + \dots + p_n\delta_{x_n}$ où les x_i sont tous différents. On note $\mathcal{X} := \{x_1, \dots, x_n\}$, qui représente un alphabet dans lequel les suites seront écrites. Alors, on montre qu'il existe une constante $c > 0$ telle que pour tout $m \in \mathbb{N}^*$:

$$\mathbf{H}(P) \leq \frac{1}{m} \sum_{y \in \mathcal{X}^m} p_{y_1} \cdots p_{y_m} \mathbf{K}(y|m) \leq \mathbf{H}(P) + \frac{n \log m}{m} + \frac{c}{m}.$$

Cf. [CT91, théorème 7.3.1 page 154] pour une preuve. Ainsi, on a :

$$\lim_{m \rightarrow +\infty} \mathbb{E} \left(\frac{1}{m} \mathbf{K}((X_1, \dots, X_m)|m) \right) = \mathbf{H}(P).$$

La complexité de Kolmogorov et l'entropie de Shannon sont donc très liées.

1.2 Loi uniforme et théorème fondamental de la simulation

Les générateurs de nombres aléatoires sont très importants dans les applications. Ils permettent d'une part de simuler des phénomènes dont la modélisation fait appel à des lois de probabilités et d'autre part de sécuriser les échanges de données, dans les réseaux téléinformatiques par exemple. Dans la pratique, les générateurs aléatoires sont basés sur des algorithmes déterministes, comme nous allons le voir, ou encore sur une source d'« entropie » extérieure difficilement prédictible liée au fonctionnement de la machinerie informatique (par exemple le temps séparant deux interruptions du processeur), ces deux approches pouvant être combinées dans un même processus¹.

Simuler la loi uniforme consiste à produire une suite de nombres qui peuvent être considérés comme autant de réalisations indépendantes de variables aléatoires uniformes sur $[0, 1]$. Bien entendu, un algorithme est toujours déterministe lorsqu'on le connaît, et il vaut donc mieux parler de générateur pseudo-aléatoire de la loi uniforme. Mathématiquement, les n sorties successives d'un tel générateur seront considérées comme la donnée de $U_1(\omega), \dots, U_n(\omega)$ pour un $\omega \in \Omega$ où les U_i sont des v.a.r. i.i.d. $U_i : (\Omega, \mathcal{A}, \mathbb{P}) \rightarrow [0, 1]$ de loi uniforme.

La mémoire des ordinateurs étant finie, il est impossible de stocker des suites infinies. En conséquence, les nombres réels sont toujours approximés, avec une certaine précision. Soit donc $p \in \mathbb{N}^*$ qui va jouer le rôle de précision. Un réel x_n dans $[0, 1]$ s'écrira y_n/p où y_n est un entier dans $\{0, \dots, p\}$. Un algorithme très usité pour générer les y_n consiste à utiliser la relation de récurrence $x_n = y_n/p$ où $y_{n+1} \equiv ay_n[m]$ où a et m sont des entiers et m est assez grand.

En général, la période de $(y_n)_n$ est plus petite que m . Plus précisément, si a et m sont premiers entre eux, et si $0 < y_0 < m$, alors y_n n'est jamais nul et la période de $(y_n)_n$ n'est rien d'autre que l'ordre de a dans \mathbb{Z}_m . On rappelle que d'après le (petit) théorème de Fermat-Euler, l'ordre de a divise l'indicateur d'Euler de a (qui vaut $m - 1$ quand m est premier).

Matlab 4 utilise un algorithme de ce type pour implémenter la fonction `rand` mais les versions plus récentes de Matlab utilisent un algorithme différent et plus performant. Dans certaines bibliothèques de calcul scientifique, on utilise $(a, m) = (13^{13}, 2^{59})$ ou encore $(a, m) = (7^5, 2^{31} - 1)$, cf. [Bou86, page 74]. L'algorithme utilisé par Octave 2.1 pour `rand` est le « Mersenne Twister », cf. <http://www.math.keio.ac.jp/~matumoto/ent.html>. On pourra également consulter [Knu81] à propos de la génération aléatoire en général. Le théorème suivant nous assure la possibilité, théorique, de simuler n'importe quelle loi de probabilité sur \mathbb{R}^d à partir de la loi uniforme :

Théorème 1.2.1 (Théorème fondamental de la simulation). *Soit $m > 1$ un entier, et U_1, \dots, U_m des variables aléatoires i.i.d. de loi uniforme sur $[0, 1]$. Pour toute loi de probabilité μ sur \mathbb{R}^d , il existe une fonction borélienne $f_{\mu, m} : \mathbb{R}^m \rightarrow \mathbb{R}^d$ dont l'ensemble des points de discontinuité est Lebesgue-négligeable telle que la variable aléatoire $Y := f_{\mu, m}(U_1, \dots, U_m)$ suive la loi μ .*

Ce théorème donne un résultat théorique intéressant, bien que l'on ne s'en serve pas dans la pratique. On en trouvera une preuve dans [Bou86, chap. X page 267]. En lisant la preuve, on s'aperçoit que la fonction f peut être explicitée. On peut donc théoriquement simuler toute loi à partir d'une simulation de la loi uniforme. En effet, supposons que nous désirions obtenir n réalisations indépendantes de loi μ . Prenons alors $m = 1$ dans le théorème précédent. Un générateur pseudo-aléatoire uniforme, du type

¹Sur le système Linux, on dispose d'une source algorithmique `/dev/urandom` et d'une source « entropique » `/dev/random` utile pour la sécurisation car difficilement prédictible.

rand, nous fournit une suite $U_1(\omega), \dots, U_n(\omega)$, qui sont considérées comme les réalisations pour $\omega \in \Omega$ de n v.a.r. i.i.d. suivant une loi uniforme sur $[0, 1]$. Le théorème précédent affirme alors l'existence d'une fonction f_μ telle que $f_\mu(U_1(\omega)), \dots, f_\mu(U_n(\omega))$ soient les réalisations de n v.a. i.i.d. de loi μ .

Remarque 1.2.2. Remarquons que d'après la loi des grands nombres, si

$$(X_n : (\Omega, \mathcal{A}, \mathbb{P}) \rightarrow \mathbb{R}^d, n \in \mathbb{N}^*)$$

est une suite de v.a. i.i.d. de loi μ , en d'autres termes un « échantillon de taille infinie », alors pour μ -presque tout $\omega \in \Omega$, la mesure empirique

$$\mathcal{E}_n(\omega) := \frac{1}{n} \sum_{i=1}^n \delta_{X_i(\omega)}$$

converge étroitement vers la loi μ . Il en découle que pour μ -presque tout $\omega \in \Omega$, la donnée de la *suite infinie* $(X_n(\omega), n \in \mathbb{N}^*)$ caractérise *théoriquement* la loi μ . Un générateur de nombres pseudo-aléatoires ne fournit que des *suites finies* $(X_1(\omega), \dots, X_n(\omega))$, qui ne suffisent pas à caractériser la loi μ . De même, les échantillons provenant de données réelles utilisés en statistique sont toujours *finis*, et l'un des objectifs principaux de la statistique est de se servir de ces suites finies de ce type pour approximer ou estimer la loi μ de certains phénomènes, à laquelle on n'a pas directement accès.

1.3 Simulation de lois par leur fonction de répartition

Soit μ une loi sur \mathbb{R} de fonction de répartition F . Si G désigne l'inverse continue à gauche de F , définie pour tout y dans $[0, 1]$ par

$$G(p) := \inf\{x : F(x) \geq p\},$$

alors, pour toute v.a. uniforme U sur $[0, 1]$, la v.a. $G(U)$ est de loi μ . Par exemple, pour la loi de Cauchy, de densité $(\pi(1+x^2))^{-1}$ sur \mathbb{R} , on a $F(x) = \arctan(x)/\pi + 1/2$ et $G(p) = \tan(\pi(p-1/2))$. La bibliothèque Stixbox pour Matlab fournit les fonctions de répartition inverses pour quelques lois usuelles, cf. tableau 1.1 page 38.

Théorème 1.3.1 (Simulation de la loi exponentielle). *Soit U une v.a.r. de loi uniforme sur $[0, 1]$ et $\lambda > 0$ un réel, alors la v.a.r. $-\lambda^{-1} \log U$ suit la loi exponentielle de paramètre λ .*

Cette méthode s'applique bien entendu aux lois de Weibull (cf. section 8.4.8 page 93), dont les lois exponentielles ne constituent qu'un cas très particulier. Voici un exemple d'illustration en Matlab dont la sortie graphique est donnée par la figure 1.1 :

```

1 lambda=0.5; clf; hold on;
2 title('Simulation_d''une_loi_exponentielle')
3 ylabel('Densite'); xlabel('Valeurs');
4 [E,C]=histo(-log(rand(500,1))/lambda,100,0,1);
5 plot(C,lambda*exp(-lambda*C),'k-*')
6 legend('Empirique','Theorique')

```

La bibliothèque Stixbox fournit la fonction `rexpweib` qui permet d'obtenir des réalisations de loi de Weibull et donc en particulier de loi exponentielle. Faites `help rexpweib`.

1.4 Simulation de lois par la méthode du rejet

On attribue parfois cette méthode à J. von Neumann. Soit une loi μ de densité f sur \mathbb{R} , que l'on désire simuler.

- Supposons que f est continue et à support compact $[a, b]$. Le graphe de f est donc contenu dans un rectangle $[a, b] \times [0, M]$. On considère alors des vecteurs aléatoires indépendants $((X_n, Y_n))_{n \geq 0}$ suivant une loi uniforme sur $[a, b] \times [0, M]$. On définit ensuite T comme le plus petit n tel que $f(X_n) \leq Y_n$. Alors on montre que T est fini presque sûrement et que X_T a pour loi μ .
- Supposons qu'il existe une densité g facile à simuler telle que $f \leq cg$ pour une constante $c > 0$. Soit alors $(W_n)_n$ et $(U_n)_n$ deux suites de v.a.r. i.i.d. indépendantes avec les W_n de loi de densité g et les U_n de loi uniforme sur $[0, 1]$. On pose $Y_n = cU_n g(W_n)$ et T le plus petit n tel que $Y_n < f(W_n)$. Alors la loi de U_T a pour densité f .

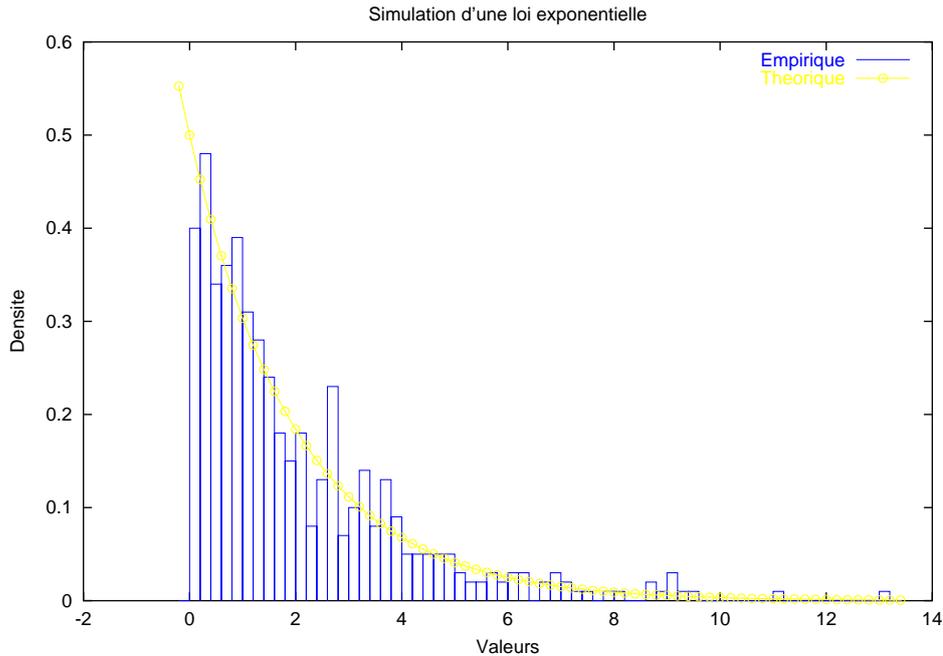


FIG. 1.1 – Simulation de la loi exponentielle de paramètre $1/2$ au moyen de sa fonction de répartition.

Ces deux méthodes s'étendent sans difficultés à des lois à densité sur \mathbb{R}^d . Les méthodes de rejet ont été développées et améliorées, et sont largement utilisées dans la pratique. Elles ont l'avantage de nécessiter peu de calculs lorsqu'elles sont couplées à des tables. La bibliothèque Stixbox fournit trois fonctions `rjbinom`, `rjgamma` et `rjpoiss` qui permettent d'obtenir des réalisations de lois binomiales, gamma et de Poisson avec la méthode du rejet. On pourra consulter [Bou86, IV.4.3, p. 79] ou encore [BL98, VI.5.15, p. 180] pour des preuves.

1.5 Simulation de lois discrètes

Soit p_1, \dots, p_n des nombres dans $[0, 1]$ tels que $p_1 + \dots + p_n = 1$. En partitionnant l'intervalle $[0, 1]$ en morceaux adjacents de longueurs p_1, \dots, p_n , il vient :

Théorème 1.5.1 (Simulation d'une loi discrète à support fini). *Soit U une v.a.r. de loi uniforme sur $[0, 1]$ et $P := p_1\delta_{x_1} + \dots + p_n\delta_{x_n}$ une loi discrète où les x_i sont tous différents. Alors la v.a.r.*

$$x_1\mathbf{I}_{\{U < p_1\}} + x_2\mathbf{I}_{\{p_1 \leq U \leq p_1 + p_2\}} + \dots + x_n\mathbf{I}_{\{p_1 + \dots + p_{n-1} \leq U \leq 1\}}$$

suit la loi P .

Le code Matlab suivant fournit la fonction `rdist` qui implémente le résultat du théorème 1.5.1 :

```

1 function realis = rdist(x,p)
2 %RDIST réalisation aléatoire d'une loi discrète à support fini (atomique).
3 % realis = rdist(x,p)
4 % p est un vecteur de réels positifs tel que sum(p)=1.
5 % x est un vecteur de réels de même taille que p, donnant le support.
6 % realis = réalisation de la loi discrète tq P(x_1)=p_1,...,P(x_n)=p_n.
7 % Les réalisations successives sont considérées comme indépendantes car
8 % elles correspondent à des appels successifs de rand.
9
10 n=length(p);
11 r=rand;
12 a=0;
13 b=p(1);

```

```

14 for i=1:n-1,
15     if ((r>=a)&(r<=b))
16         realis =x(i);
17         return;
18     end
19     a=b;
20     b=b+p(i+1);
21 end
22 realis =x(n);
23 return;

```

Exercice 1.5.2. Montrez que cette méthode s'étend sans difficultés aux lois discrètes à support dénombrable. Donnez un petit programme d'illustration.

La bibliothèque Stixbox fournit la fonction `quantile` qui permet d'obtenir les quantiles d'une loi à partir d'un échantillon empirique. Faites `help quantile`.

Le nombre moyen d'instructions `if` exécutées dans le code de la fonction `rdist` est voisin de $\log n$. On pourra consulter par exemple [Knu81] pour des méthodes plus rapides faisant intervenir des tables précalculées à partir des probabilités p_i . La loi uniforme discrète peut être simulée très rapidement, sans aucune instruction conditionnelle, comme le montre le code suivant :

```

1 function realis = randiscr(x,n,m)
2 %RANDISCR
3 % Similaire à rand. Renvoie des réalisations i.i.d. de loi uniforme
4 % sur x(1),...,x(length(x)).
5 % n et m sont des entiers optionnels, valant 1 par défaut.
6
7 if (nargin==0),
8     error('Pas_assez_de_paramètres');
9 elseif (nargin==1),
10    n=1; m=1;
11 elseif (nargin==2)
12    m=1;
13 elseif (nargin>3)
14    error('Trop_de_paramètres');
15 end
16 realis =reshape(x(ceil(length(x)*rand(n,m))),n,m);
17 return;

```

Théorème 1.5.3 (Simulation de la loi binomiale). Soient U_1, \dots, U_n des v.a.r. i.i.d. de loi uniforme sur $[0, 1]$, alors la v.a.r.

$$\sum_{i=1}^n \mathbf{I}_{\{U_i < p\}}$$

suit une loi binomiale de taille n et de paramètre p .

Théorème 1.5.4 (Simulation de la loi de Poisson). Soient $(E_i)_{i \geq 1}$ des v.a.r. i.i.d. de loi exponentielle de paramètre $\lambda > 0$, alors la v.a.r.

$$\mathbf{I}_{\{E_1 \leq 1 < E_1 + E_2\}} + 2\mathbf{I}_{\{E_1 + E_2 \leq 1 < E_1 + E_2 + E_3\}} + 3\mathbf{I}_{\{E_1 + E_2 + E_3 \leq 1 < E_1 + E_2 + E_3 + E_4\}} + \dots$$

suit une loi de Poisson de paramètre λ . Il en est de même pour :

$$\mathbf{I}_{\{E_1 \leq 1\}} + \mathbf{I}_{\{E_1 + E_2 \leq 1\}} + \mathbf{I}_{\{E_1 + E_2 + E_3 \leq 1\}} + \dots$$

Comme nous le verrons plus loin, cette méthode correspond à la simulation de la loi de N_1 où $(N_t)_{t \geq 0}$ est un processus de Poisson simple d'intensité λ . L'événement $\{E_1 + \dots + E_i \leq 1 < E_1 + \dots + E_{i+1}\}$ correspond au cas où le processus a sauté exactement i fois avant le temps 1.

1.5.1 Lois fractales

Soit $n \in \mathbb{N}^*$ une base de numération. Chaque résultat d'un schéma de Bernoulli consistant en une suite d'expériences i.i.d. $(X_i, i \in \mathbb{N}^*)$ à n issues possibles de loi uniforme sur $\{0, \dots, n-1\}$ peut se représenter

par l'écriture en base n d'un nombre réel dans $[0, 1]$. En d'autres termes, on a la surjection \mathcal{I}_n suivante :

$$\mathcal{I}_n : (x_1, \dots, x_k, \dots) \in \{0, \dots, n-1\}^{\mathbb{N}^*} \mapsto \sum_{i=1}^{+\infty} n^{-i} x_i = \underbrace{0, x_1 \cdots x_k \cdots}_{\text{en base } n} \in [0, 1].$$

On pourra penser à la base $n = 2$ et au schéma de Bernoulli classique de pile ou face avec une pièce équilibrée. La variable aléatoire U définie par :

$$U := \mathcal{I}_n(X_1, \dots, X_k, \dots) := \sum_{i=1}^{+\infty} n^{-i} X_i,$$

suit une loi uniforme sur $[0, 1]$. En effet, si $a := a_1 n^{-1} + \dots + a_m n^{-m}$ est un nombre n -adique, on a :

$$\mathbb{P}(a < U < a + n^{-m}) = \mathbb{P}(X_1 = a_1, \dots, X_m = a_m) = \mathbb{P}(X_1 = a_1) \cdots \mathbb{P}(X_m = a_m) = n^{-m}.$$

Réciproquement, ce calcul montre que les coefficients $(X_m)_{m \geq 1}$ de l'écriture en base n d'une variable aléatoire uniforme sur $[0, 1]$ sont i.i.d. de loi uniforme sur $\{0, \dots, n-1\}$. Remarquons que l'ensemble des nombres n -adiques est de mesure nulle pour la loi uniforme. Ainsi, les nombres de $[0, 1]$ dont l'écriture en base n est constante à partir d'un certain rang « ne comptent pas » en quelque sorte, et \mathcal{I}_n est presque-sûrement une injection, et donc une bijection p.s.

Exercice 1.5.5. En déduire une méthode pour générer d'un seul coup k réalisations indépendantes de loi uniforme sur $\{1, \dots, n\}$ à partir d'une réalisation de précision k en base n d'une loi uniforme sur $[0, 1]$. Implémenter ce résultat en écrivant une fonction Matlab.

Que se passe-t-il pour U si l'on suppose que les X_i ne suivent plus une loi uniforme sur $\{0, n-1\}$ mais plutôt une loi $\{p_1, \dots, p_n\}$ quelconque fixée d'avance? En d'autres termes, quelles sont les lois images possibles pour la fonction \mathcal{I}_n sur $[0, 1]$? Comme $(n^r U) \bmod n^r$ a la même loi que U pour tout $r \in \mathbb{N}$, le graphe de la fonction de répartition de U possède une propriété d'invariance d'échelle : c'est un fractal. La fonction de répartition de U est continue, mais la loi de U est étrangère à la mesure de Lebesgue et les lois de U sont toutes étrangères entre elles quand la loi des X_i balaye l'ensemble des lois discrètes sur $\{0, \dots, n-1\}$. L'exemple de code pour Octave qui suit avec sa sortie graphique 1.2 page 23 illustre ce phénomène. On pourra consulter [DCD82b, ex. 3.3.15 page 81], [BL98, expl. IV.3.6.iii pages 100-103 et V.5.3 page 144] et enfin [BL98, ex. V.6.15 page 155] à propos de ces « lois fractales »

```

1  % Loïs Fractales dans un schéma de Bernoulli à n issues non uniformes
2  clear;
3  more off;
4  nr=1000;
5  prec=100;
6  %
7  base1=2;
8  p1=1/4;
9  sprintf('Cas_des_lois_de_Bernoulli_de_parametre_%f.\n',p1)
10 sprintf('%d_réal indép de loi de Bernoulli sur {0,1}...\n',\
11   prec*nr)
12 coefs= repmat(cumprod(ones(prec,1)/base1),1,nr);
13 realis = sum(rbinom([prec,nr],base1-1,p1).*coefs);
14 [ effectifs , classes ] = hist(realis , prec*10);
15 clf;
16 title(' Différents cas pour la fonction de répartition de U');
17 hold on;
18 plot( classes , classes );
19 plot( classes , cumsum(effectifs)/nr);
20 %
21 base2=7;
22 p2=5/9;
23 sprintf('Cas_des_lois_binomiales_de_taille_%d_et_de_parametre_%f.\n',\
24   base2,p2)
25 sprintf('%d_réal indép de loi binomiales sur {0,1}...\n',\
26   prec*nr)
27 coefs= repmat(cumprod(ones(prec,1)/base2),1,nr);

```

```

28 realis =sum(rbinom([prec,nr],base2-1,p2).*coefs);
29 [ effectifs , classes]=hist( realis ,prec*10);
30 plot( classes , cumsum(effectifs)/nr);
31 %
32 base3=2;
33 p3=1/2;
34 sprintf('Cas_des_lois_de_Bernoulli_symétriques_%.1f.\n',base3,p3)
35 sprintf('%d_réal._ indép._ de_loi_de_Bern._sym._sur_{0,1}...\n',\
36   prec*nr)
37 coefs= repmat(cumprod(ones(prec,1)/base3),1,nr);
38 realis =sum(rbinom([prec,nr],base3-1,p3).*coefs);
39 [ effectifs , classes]=hist( realis ,prec*10);
40 plot( classes , cumsum(effectifs)/nr);
41 %
42 xlabel('Valeur');
43 ylabel('Fonction_de_repartition');
44 legend('Loi_uniforme',\
45   sprintf('Avec_des_Bernoulli_%.12f',p1),\
46   sprintf('Avec_des_binomiales_(%d,%.12f)',base2,p2),\
47   sprintf('Avec_des_Bernoulli_symétriques'),\
48   4);

```

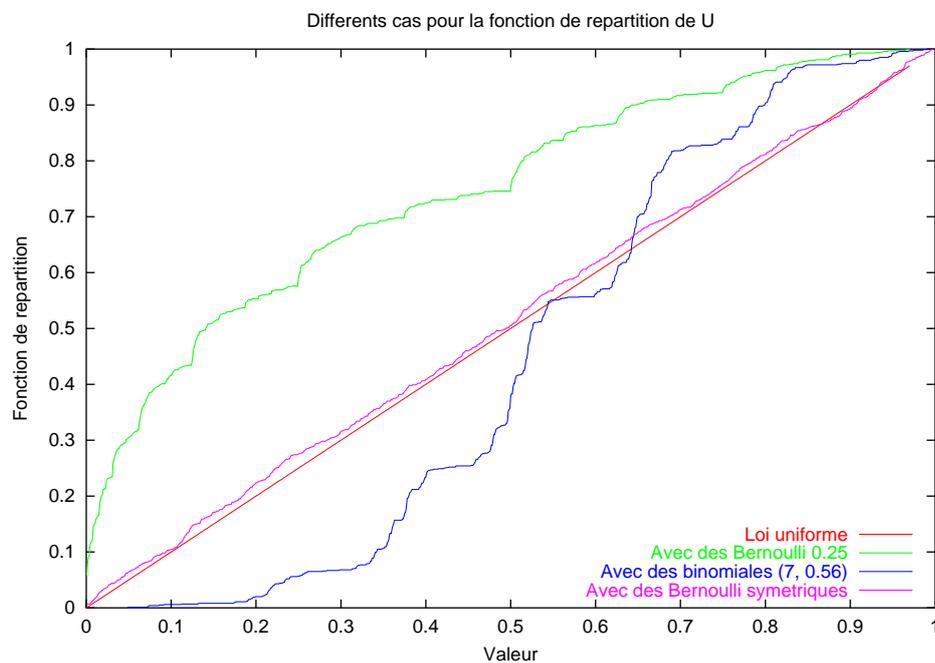


FIG. 1.2 – Loïs fractales associées à un schéma de Bernoulli.

1.5.2 Simulation des chaînes de Markov à valeurs dénombrables

Soit $(X_n, n \geq 0)$ une chaîne de Markov à espace d'états \mathbb{N} (pour simplifier). On note

$$K : \mathbb{N} \times \mathbb{N} \rightarrow [0, 1]$$

son noyau de transition. On a donc par définition $K(x, y) := \mathbb{P}(X_{n+1} = y | X_n = x)$. Le noyau de transition itéré est $K_n(x, y) := \mathbb{P}(X_{n+1} = y | X_1 = x)$. Comme

$$\mathcal{L}(X_n | X_0 = x) = \sum_{y \in \mathbb{N}} K_n(y, x) \delta_y,$$

on peut utiliser le théorème 1.5.1 pour simuler $\mathcal{L}(X_n | X_0 = x)$ lorsque l'on connaît K_n . Pour simuler $\mathcal{L}(X_n)$ lorsque $\mathcal{L}(X_0)$ est connue, on simule d'abord X_0 puis $\mathcal{L}(X_n | X_0)$, toujours avec le théorème 1.5.1.

Lorsque la chaîne de Markov est à espace d'états fini, K peut être identifié à une matrice de transition P et on a $K_n = P^n$. Il suffit donc de calculer la puissance de la matrice de transition P sous Matlab pour pouvoir appliquer ensuite le théorème 1.5.1. Voici un code Matlab illustrant la méthode décrite pour une matrice de transition 3×3 . Il fait appel à la fonction Matlab `rdist` donnée page 21.

```

1  %% Simulation d'une chaîne de Markov
2
3  clf; clear;
4  % Une matrice de transition 3x3 récurrente irréductible aperiodique
5  P=[ 1/3,1/3,1/3 ; 1/4,2/4,1/4 ; 2/5,2/5,1/5 ];
6
7  % Tracé d'un histogramme pour un temps grand fixé (loi de X_n).
8  % Cela revient juste à tester rdist puisque la loi cherchée est P^n(:,X_0).
9  n=500; % temps à considérer
10 m=10000; % Nombre de réalisations.
11 xzero=1; % Point initial.
12 Q=P^n;
13 for i=1:m,
14     realis(i)=rdist([1,2,3], Q(:,xzero));
15 end
16 figure(1)
17 hist(realis,3)
18
19 % Mesure invariante mu et tracé de l'erreur relative de son approximation.
20 % mu est bien approximée par P^n(:,X_0) quand n est grand (X_0 quelconque).
21 R=P^100;
22 Q=R;
23 for i=100:5000:10000,
24     Q=Q*R;
25     mu=Q(:,1);
26     erreur(i)=max((mu'*P-mu)'/mu);
27 end
28 figure(2)
29 plot(erreur,'b-') % Tracé de l'erreur relative par pas de 100.
30
31 % Tracé d'une trajectoire particulière (une seule réal. de X(1),...,X(m)).
32 m=50; % temps maximum à considérer
33 X(1)=1; % Point initial.
34 for i=1:m-1,
35     X(i+1)=rdist([1,2,3], P(:,X(i)));
36 end
37 figure(3)
38 plot(X,'b*')
```

1.5.3 Simulation du processus de Poisson simple

Un processus de Poisson simple d'intensité $\lambda > 0$ est une famille $(N_t, t \in \mathbb{R}_+)$ de v.a. indexée par \mathbb{R}_+ et à valeurs dans \mathbb{N} telle que :

1. Pour tout $t \in \mathbb{R}_+$, la v.a. N_t suit une loi de Poisson de paramètre λt
2. Pour tout $(s, t) \in \mathbb{R}_+ \times \mathbb{R}_+$, les v.a. $N_{t+s} - N_s$ et N_s sont indépendantes.

On définit les instants de saut de ce processus par récurrence en posant $T_0 := 0$ et :

$$T_n := \inf \{t \geq 0, N_t > N_{T_{n-1}}\}.$$

On montre facilement que les v.a.r. $(S_n := T_n - T_{n-1}, n \in \mathbb{N}^*)$ sont i.i.d. de loi exponentielle de paramètre λ , et l'on a la décomposition suivante :

$$N_t = \sum_{n=1}^{+\infty} \mathbb{I}_{\{T_n \leq t\}}, \quad (1.1)$$

Réciproquement, si l'on considère des v.a.r. $(S_n, n \in \mathbb{N}^*)$ i.i.d. de loi exponentielle de paramètre $\lambda > 0$, et si l'on pose $T_0 := 0$ et $T_n = S_1 + \dots + S_n$, alors l'équation (1.1) définit un processus de Poisson $(N_t, t \in \mathbb{R}_+)$ d'intensité λ . Les v.a.r. $(S_n, n \in \mathbb{N}^*)$ représentent les temps inter-sauts du processus. On montre enfin à partir de la propriété des accroissements i.i.d. du processus que l'on a une « loi des grands nombres » :

$$\frac{N_t}{t} \underset{t \rightarrow +\infty}{\text{p.s.}} \lambda. \quad (1.2)$$

Le processus $(N_t, t \in \mathbb{R}_+)$ apparaît comme le *processus de comptage* de la suite à accroissements indépendants $(T_n)_n$. Il permet de modéliser des *files d'attentes* pour lesquelles les v.a.r. T_n représentent les temps d'arrivées dans la file et N_t le nombre d'éléments dans la file à l'instant t . Pensez par exemple à un guichet d'une administration, à une caisse de supermarché ou encore à un aéroport. Dans la pratique, une fois que l'on a décidé de simuler un phénomène avec un tel processus, il nous faut ajuster l'intensité λ en se servant des données (échantillons) dont on dispose sur le phénomène en question. On parle alors d'estimation, et la statistique pointe le bout de son nez. Ici, les choses sont très simples puisque $\mathbb{E}(N_t) = (t\lambda)^{-1}$, moyenne que l'on peut estimer à partir des données réelles.

Pour simuler une trajectoire du processus $(N_t, t \in \mathbb{R}_+)$, il suffit de simuler des lois exponentielles (par exemple avec le théorème 1.3.1) et de « sauter » en conséquence. Voici un exemple de code Matlab faisant appel au générateur exponentiel `rexpweib` de la bibliothèque Stixbox, dont la sortie graphique est donnée par la figure 1.3

```

1 %% Code concis pour simuler un processus de Poisson Simple.
2 %% Fait appel à rexpweib de Stixbox
3 clf; xlabel('t'); ylabel('Nt')
4 legend('Trajectoire_d'un_Processus_de_Poisson_Simple',4);
5 stairs(cumsum(rexpweib(30,1/2,1)),cumsum([1:30]));

```

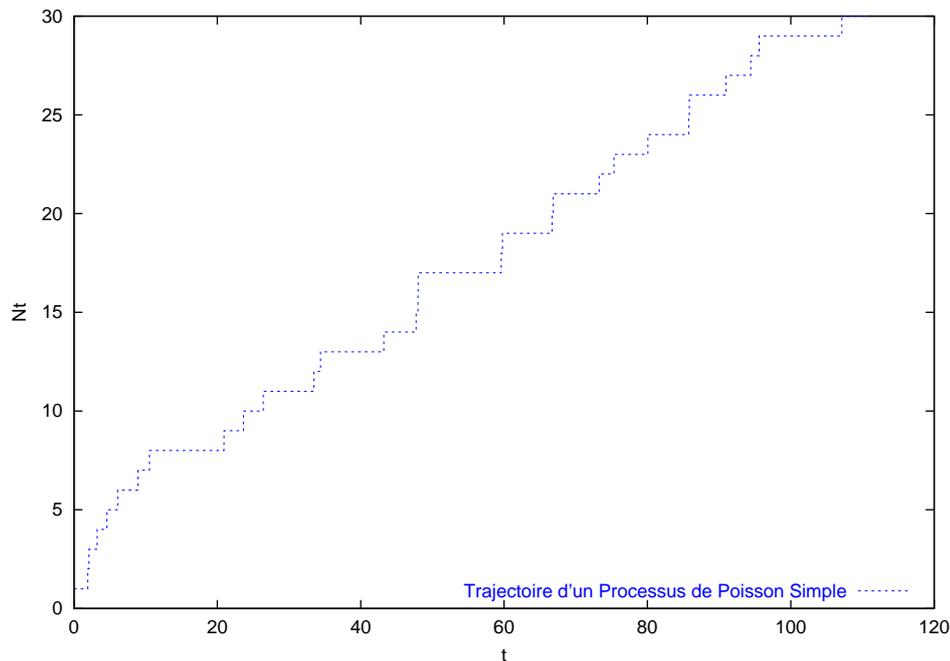


FIG. 1.3 – Simulation d'une trajectoire d'un processus de Poisson simple.

Exercice 1.5.6. Assurez-vous de bien avoir compris le code précédent. En simulant plusieurs fois ce genre de trajectoire, retrouver empiriquement que la v.a. N_t suit bien une loi de Poisson de paramètre λt où λ est le paramètre de la loi exponentielle des sauts.

Remarque 1.5.7 (Assurances et réservoir percé). La valeur X_t du compte en banque à l'instant t d'un assureur communiste peut être modélisée par un processus de Poisson avec dérive :

$$X_t := \alpha t - N_t,$$

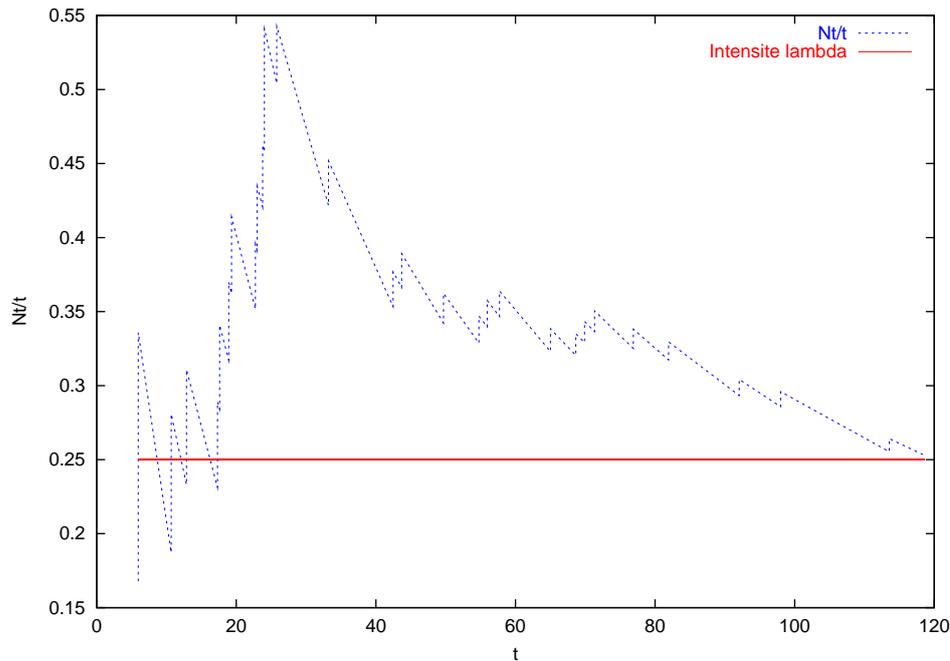


FIG. 1.4 – Simulation d'une trajectoire renormalisée d'un processus de Poisson simple.

où la partie αt représente les apports, supposés continus et positifs des assurés (cotisations) et N_t les dépenses correspondant aux remboursements liés aux sinistres des assurés. L'assureur étant communiste, il ne touche aucun intérêt lié à son capital, et ne place pas son argent en bourse par conviction. Modéliser le compte en banque d'un assureur capitalise est plus compliqué, et nécessite l'intervention de processus du type « exponentielle de brownien » pour modéliser les placements destinés à faire des bénéfices. Une mutuelle constitue un cas intermédiaire, puisqu'elle ne doit pas faire de bénéfices mais fait quand même fructifier son capital, y compris en bourse, pour mieux servir ses adhérents. On peut aussi voir $-X_t$ comme la quantité d'eau dans un réservoir à ciel ouvert percé d'un petit trou : N_t représente l'eau apportée par les averses, supposées brèves, et $-\alpha t$ l'eau qui s'écoule continûment par le petit trou.

1.6 Simulation des lois gaussiennes

Dans la pratique, les méthodes du type « Marsaglia » semblent être de loin les plus performantes. Cependant, nous commençons par la méthode « polaire », qui est à la fois très classique et... très lente.

1.6.1 Méthode polaire pure

Théorème 1.6.1 (Simulation d'une loi normale). Soit $(X, Y) = (r \cos(\theta), r \sin(\theta))$ un vecteur aléatoire de \mathbb{R}^2 . Alors (X, Y) suit une loi $\mathcal{N}(0, I_2)$ si et seulement si $r^2 = X^2 + Y^2$ et $\theta = \arg(X + iY)$ sont indépendantes et suivent respectivement une loi exponentielle de paramètre $1/2$ et une loi uniforme sur $[0, 2\pi]$.

Il en découle que si U_1 et U_2 sont indépendantes et uniformes sur $[0, 1]$, alors les v.a.r.

$$\sqrt{-2 \log U_1} \cos(2\pi U_2) \quad \text{et} \quad \sqrt{-2 \log U_1} \sin(2\pi U_2)$$

sont i.i.d. de loi normale centrée réduite. Bien entendu, pour tout $(m, \sigma) \in \mathbb{R} \times \mathbb{R}_+^*$, une variable aléatoire réelle Z suit la loi $\mathcal{N}(0, 1)$ si et seulement si la v.a.r. $m + \sigma Z$ suit la loi $\mathcal{N}(m, \sigma^2)$. Nous avons donc là une méthode de simulation de lois gaussiennes unidimensionnelles quelconques. Voici un exemple d'illustration dont la sortie graphique est donnée par la figure 1.5 :

```

1 clf; hold on;
2 title('Simulation_d''une_loi_normale');
```

```

3 ylabel('Effectifs'); xlabel('Valeurs');
4 [E,C]=histo(sqrt(-2*log(rand(5000,1))).*cos(2*pi*rand(5000,1)),100,0,1);
5 plot(C,(2*pi)^(-1/2)*exp(-C.^2/2),'k-*');
6 legend('Empirique','Theorique');

```

Il est préférable d'utiliser la fonction Matlab `randn` (ou bien la fonction Stixbox `rnorm`) qui est déjà

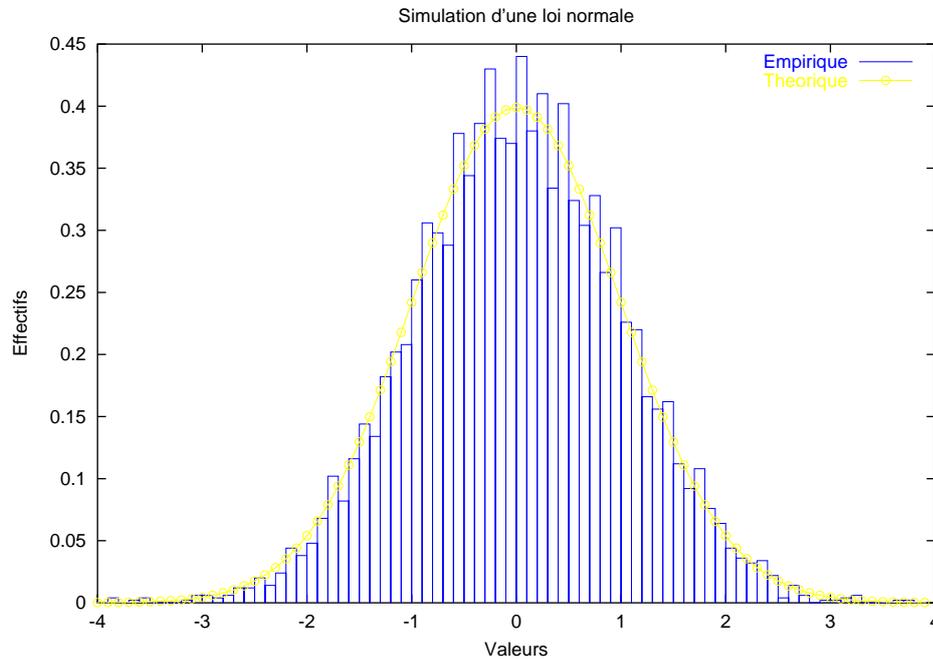


FIG. 1.5 – Simulation de la loi normale par méthode « polaire »

toute programmée et qui fait appel à une méthode plus efficace. Cette méthode de simulation polaire est coûteuse en temps de calcul car elle nécessite l'évaluation des fonctions \log , $\sqrt{\quad}$ et \cos . On peut se passer du \cos en faisant appel à la méthode du rejet, comme décrit ci-après.

1.6.2 Méthode polaire - rejet

Si $(X, Y) = (\rho \cos(\theta), \rho \sin(\theta))$ suit la loi uniforme sur le disque unité du plan, alors le vecteur aléatoire

$$\frac{\sqrt{-4 \log(\rho)}}{\rho} (X, Y)$$

suit une loi normale centrée réduite bidimensionnelle. La loi de (X, Y) est facile à simuler par la méthode du rejet à partir d'une loi uniforme sur le carré $[-1, +1]^2$ (rejet dans 21% des cas puisque $\pi/4 \simeq 0.79$).

On remarquera qu'ici, θ et ρ sont indépendantes, que ρ suit une loi de densité $u \mapsto 2uI_{[0,1]}(u)$ et que θ suit une loi uniforme sur $[0, 2\pi]$. D'autre part, on a

$$\frac{\sqrt{-4 \log(\rho)}}{\rho} (X, Y) = \sqrt{-4 \log(\rho)} (\cos(\theta), \sin(\theta)).$$

On vérifie alors facilement que $-4 \log(\rho)$ suit une loi exponentielle de paramètre $1/2$:

$$\int_0^1 f(-4 \log(\rho)) 2\rho d\rho = \int_0^{+\infty} f(u) \frac{1}{2} e^{-u/2} du.$$

Cette méthode, que l'on peut qualifier de « polaire - rejet », est celle utilisée dans Matlab 4 pour implémenter la fonction `randn`. Elle demeure cependant assez lente, en raison à la fois des 21% de rejet et de l'évaluation des fonctions \log et $\sqrt{\quad}$.

1.6.3 Méthode de Marsaglia (algorithme du Ziggurat)

Les versions 5 et 6 de Matlab utilisent plutôt une méthode due à Marsaglia, appelée « algorithme du Ziggurat² ». Elle consiste à utiliser la méthode du rejet conjointement avec des tables donnant une approximation de la densité gaussienne par des rectangles. Ces tables étant précalculées, la génération de nombres aléatoires gaussiens est très rapide, quasiment aussi rapide que celle des nombres aléatoires uniformes. On pourra en trouver une version moins optimisée dans le tome II du livre de Knuth : « The Art of Computer Programming » [Knu81].

On observe tout d'abord qu'il suffit de savoir simuler une v.a.r. de densité

$$f(u) := \sqrt{\frac{\pi}{2}} e^{-u^2/2} \mathbf{I}_{[0,+\infty[}(u),$$

puisque le signe peut être obtenu via une loi de Bernoulli symétrique. Considérons à présent n points $x_1 < \dots < x_n$. En prenant $x_1 = 0$ et $x_{n+1} = +\infty$, les points du plan de coordonnées $(x_i, f(x_i))$ constituent les coins inférieurs droits de n rectangles plats adjacents (horizontaux), qui recouvrent par excès l'aire sous la courbe de f . La hauteur du $k^{\text{ième}}$ rectangle vaut $f(x_k) - f(x_{k+1})$.

On choisit les x_i de telle sorte que la portion d'aire de f située dans chaque rectangle soit constante (et égale à $1/n$). Ensuite, l'algorithme consiste à choisir l'un des rectangles en simulant une loi uniforme sur les entiers $\{1, \dots, n\}$. Une fois le rectangle choisi, on utilise :

- la méthode du rejet exacte pour le premier et le dernier rectangle, via l'expression de f ;
- la méthode du rejet approximative pour les rectangles intermédiaires, qui consiste à utiliser la partie du rectangle sous la courbe de f , via le rapport $\sigma_k := x_{k-1}/x_k$, sans faire appel à l'expression de f .

Les x_i et les σ_i sont précalculés une fois pour toutes et stockés dans une table, et n est choisi suffisamment grand³ pour que les taux de rejet soient faibles et les approximations bonnes. Les calculs sont très réduits une fois la phase d'initialisation passée.

Il n'est pas nécessaire de considérer la partie positive de la densité gaussienne, puisque le découpage en rectangles est symétrisable.

1.6.4 Simulation d'un vecteur gaussien

Soit $m \in \mathbb{R}^n$ et $\Gamma \in \mathcal{S}_n^+(\mathbb{R})$ une matrice réelle de taille n , symétrique positive, et soit Y est un vecteur aléatoire gaussien standard de moyenne nulle et de matrice de covariance \mathbf{Id}_n , alors le vecteur aléatoire $X := \Gamma^{\frac{1}{2}}Y + m$ est un vecteur gaussien de moyenne m et de matrice de covariance Γ . Réciproquement, si X est un vecteur gaussien de loi $\mathcal{N}(m, \Gamma)$, alors $\Gamma^{-1/2}(X - m)$ est un vecteur aléatoire gaussien de loi $\mathcal{N}(0, \mathbf{Id}_n)$.

On peut également utiliser la décomposition de Cholesky de $\Gamma = AA^\top$ où A est une matrice triangulaire inférieure, en lieu et place de la décomposition en racine carrée $\Gamma = \Gamma^{1/2}\Gamma^{1/2}$, ce qui moins coûteux en terme de calculs (on évite de diagonaliser Γ et A est creuse).

Si Y est un vecteur gaussien de loi $\mathcal{N}(0, \mathbf{Id}_n)$, ses coordonnées sont i.i.d. de loi $\mathcal{N}(0, 1)$. Ainsi, on peut simuler une réalisation de Y en simulant n réalisations indépendantes $Y_1(\omega), \dots, Y_n(\omega)$ de loi $\mathcal{N}(0, 1)$. Il suffit alors de calculer $\Gamma^{1/2}Y(\omega) + m$ pour obtenir une réalisation de loi $\mathcal{N}(m, \Gamma)$.

Voici un exemple de simulation de vecteur gaussien écrit en Matlab :

```

1 % Simulation d'une loi gaussienne multivariée
2 clear
3 % G = une matrice symétrique positive quelconque
4 X = rand(4,4);
5 G = X'*X/4
6 % m = un vecteur moyenne quelconque
7 m = rand(4,1)
8 % A = Racine carrée de G
9 A=sqrtm(G);
10 %
11 n = 10000;
12 moyemp = zeros(4,1);
13 realis = zeros(4,n);

```

²Il s'agit de temples de Mésopotamie, dont la forme consiste en un empilement de parallépipèdes rectangles de plus en plus petits.

³Dans Matlab, $n = 128$.

```

14 for i=1:n,
15   realis(:,i) = m + A*randn(4,1);
16   moyemp = moyemp + realis(:,i);
17 end
18 moyemp = moyemp/n;
19 % Variation relative par rapport à la moyenne théorique
20 rap_moy_relat=max(abs(moyemp-m))/norm(m)
21 % Calcul de la matrice de covariance empirique
22 covemp=zeros(4,4);
23 for i=1:4,
24   for j=1:4,
25     covemp(i,j)=dot(realis(i,:),realis(j,:))/n-moyemp(i)*moyemp(j);
26   end
27 end
28 covemp
29 % Degrés de liberté par rapport à la matrice théorique
30 X=covemp/G % Faire 'help slash' pour en savoir plus sur cette division
31 dfs=trace(eye(4,4)-X)
32 % Réduction d'entropie
33 ent=-.5*log2(det(X))

```

1.6.5 Simulation d'un mouvement brownien

Vers 1827, le jeune botaniste écossais Robert Brown observe au microscope le mouvement incessant de grains de pollen dans une goutte d'eau, et a alors l'intuition que cette agitation est d'origine physique, et non pas biologique. Ce faisant, il vient de mettre en évidence le phénomène qui porte aujourd'hui son nom : le mouvement brownien. Vers 1900, le mathématicien Louis Bachelier s'intéressa dans sa thèse intitulée « La théorie de la spéculation », méprisée à l'époque et tombée longtemps dans l'oubli, à l'évolution des cours de la bourse, dont le comportement ressemble à un mouvement brownien. Quelques années plus tard (1905), et de façon indépendante, le physicien Albert Einstein donne une explication théorique du mouvement brownien à base de physique classique : chaque particule subit des chocs incessants du milieu qui provoquent son agitation « brownienne ». Un peu plus tard, le physicien Paul Langevin améliorera cette approche en tenant compte de l'effet du fluide ou d'une force extérieure sur la diffusion des particules. Tous ces développements vont conduire le mathématicien Norbert Wiener à introduire vers 1923 la notion mathématique de mouvement brownien, telle que nous la connaissons aujourd'hui. Ce travail fondamental constitue le socle d'un pan important de la théorie des probabilités modernes construites à partir de la théorie de la mesure, au milieu du vingtième siècle, par de nombreux mathématiciens dont les plus illustres sont sans doute Kolmogorov, Doob et Lévy⁴. Les objets et assertions mathématiques construits à partir du mouvement brownien forment ce que l'on appelle de nos jours la « théorie des processus stochastiques » [Bas95, KS91]. Très développée par l'école française de probabilités, elle possède un vaste champ d'application : analyse mathématique, géométrie, physique, biologie, finance...

Un mouvement brownien unidimensionnel standard sur l'intervalle de temps $[0, 1]$, issu de $x \in \mathbb{R}$, est une famille $(B_t, t \in [0, 1])$ de variables aléatoires

$$\begin{aligned}
 B_t : (\Omega, \mathcal{A}, \mathbb{P}) &\rightarrow \mathbb{R} \\
 \omega &\mapsto B_t(\omega),
 \end{aligned}$$

indexée par $t \in [0, 1]$ et vérifiant les propriétés suivantes :

1. $B_0 = x$
2. Pour presque tout ω , la fonction $t \in [0, 1] \mapsto B_t(\omega)$ est continue.
3. Pour tout $t \in]0, 1]$, la v.a. B_t est gaussienne centrée de variance t .
4. Pour tous $(t, s) \in]0, 1]^2$, les variables B_t et $B_{t+s} - B_t$ sont indépendantes.

Pour résumer, il s'agit d'un *processus gaussien centré à trajectoires continues et à accroissement indépendants*. On montre de différentes manières qu'il est possible de construire un tel processus⁵, cf. par

⁴Citons également Wolfgang Doeblin, qui appartient au club très ouvert des oubliés de l'histoire [BDY00].

⁵Un théorème de Kolmogorov assure l'unicité de la loi du processus lorsque l'on fixe les lois marginales finidimensionnelles.

exemple [KS91]. Le processus $B := (B_t, t \in [0, 1])$ peut également être vu comme *une* variable aléatoire à valeurs dans l'espace des fonctions continues nulles en zéro $\mathcal{C}_0([0, 1], \mathbb{R})$, muni de la topologie adéquate⁶ :

$$\begin{aligned} B : (\Omega, \mathcal{A}, \mathbb{P}) &\rightarrow \mathcal{C}_0([0, 1], \mathbb{R}) \\ \omega &\mapsto (t \in [0, 1] \mapsto B_t(\omega)). \end{aligned}$$

On parle de loi du processus pour désigner la loi de cette variable aléatoire à valeurs dans un espace de fonctions. La loi \mathcal{W} du mouvement brownien standard est appelée *mesure de Wiener*. C'est en quelque sorte une mesure gaussienne de dimension infinie sur $\mathcal{C}_0([0, 1], \mathbb{R})$ puisque pour tout t , la variable aléatoire à valeurs réelles

$$\begin{aligned} b_t : (\mathcal{C}_0([0, 1], \mathbb{R}), \mathcal{W}) &\mapsto \mathbb{R} \\ f &\mapsto \omega_t(f) := f(t) \end{aligned}$$

est une gaussienne dans \mathbb{R} de même loi que B_t . Ainsi, les v.a.r. B_t apparaissent comme les lois marginales de la loi gaussienne de dimension infinie \mathcal{W} . Nous verrons plus loin que la coraviance de B_t et B_s est exactement $\min(t, s)$.

On généralise sans trop de difficultés la définition précédente à tout \mathbb{R}_+ en temps. Un mouvement brownien quelconque à valeurs dans \mathbb{R}^d s'obtient en considérant l'image de $(B_t^{(1)}, \dots, B_t^{(d)})$ par une matrice constante $M \in \mathcal{M}_d(\mathbb{R})$, où les $(B_t^{(1)}, t \in \mathbb{R}_+), \dots, (B_t^{(d)}, t \in \mathbb{R}_+)$ sont des mouvements brownien unidimensionnels standards et *indépendants* sur \mathbb{R}_+ .

Le mouvement brownien joue un rôle central dans la théorie des probabilités. Heuristiquement, il constitue l'analogie continu d'une marche aléatoire simple, dont il peut d'ailleurs se déduire par le théorème central limite (cf. théorème 2.2.3 page 43). À chaque instant, une marche aléatoire simple choisit de façon indépendante de son passé l'accroissement de sa position actuelle dans son voisinage, un peu comme un ivrogne. Le mouvement brownien est l'analogie continu de ce processus dans la mesure où les instants de choix de nouvelle position ne sont plus discrets mais « infiniment rapprochés ». Si l'on revient à l'image physique d'une particule brownienne, B_t représente la position (ou la vitesse selon que l'on considère les modèles d'Einstein ou de Langevin) de la particule à l'instant t .

Pour simuler une trajectoire d'un mouvement brownien, il suffit de remarquer que pour tout $0 < t_1 < \dots < t_n$, le vecteur aléatoire $(B_{t_1}, \dots, B_{t_n})$ suit une loi gaussienne centrée de matrice de covariance Γ connue. En effet, pour $i < j$, on a :

$$\Gamma_{i,j} = \mathbf{E}(B_{t_i} B_{t_j}) = \mathbf{E}(B_{t_i} (B_{t_j} - B_{t_i})) + \mathbf{E}(B_{t_i}^2) = t_i.$$

On a donc $\Gamma_{i,j} = \min(t_i, t_j)$, et de façon plus générale :

$$\mathbf{Cov}(B_t, B_s) = \min(t, s) = \frac{1}{2}(t + s - |t - s|) \quad \text{et} \quad \mathbf{E}(|B_t - B_s|^2) = |t - s|^2.$$

Il ne reste donc plus qu'à utiliser la méthode présentée dans la section 1.6.4 pour simuler une réalisation du vecteur gaussien $(B_{t_1}, \dots, B_{t_n})$. Cette méthode peut être utilisée pour simuler la loi d'un processus gaussien quelconque $(X_t, t \in \mathbb{R}_+)$ à partir de sa fonction de covariance $(t, s) \in \mathbb{R}_+^2 \mapsto \mathbf{Cov}(X_t, X_s)$. Le code Matlab suivant en donne un exemple, dont la sortie graphique est donnée par la figure 1.6 :

```

1 %% Simulation d'une trajectoire d'un mouvement brownien
2 clf; clear
3 t1=.1; dt=.01; n=300;
4 T=[t1:dt:t1+(n-1)*dt];
5 M= repmat(T,n,1);
6 G=min(M,M');
7 plot([0,T],[0;sqrtm(G)*randn(n,1)]);
8 hold on;
9 legend('Simulation_d''une_trajectoire_d''un_mouvement_brownien');
10 xlabel('t'); ylabel('Bt');
```

Le mouvement brownien étant à accroissements indépendants, il satisfait en particulier à la propriété de Markov. En d'autres termes, le futur du processus est indépendant du passé lorsque l'on connaît le présent. Il a donc une mémoire courte en quelque sorte. Ainsi, pour prolonger une trajectoire simulée, on peut utiliser la même méthode à partir du dernier point de cette trajectoire, ce qui n'est pas possible

⁶Topologie de la convergence simple sur $\mathcal{C}([0, 1], \mathbb{R})$, induite sur $\mathcal{C}([0, 1], \mathbb{R}) \subset \mathbb{R}^{[0,1]}$ par celle générée sur $\mathbb{R}^{[0,1]}$ par les cylindres $A_1 \times \dots \times A_m \times \mathbb{R} \times \dots \subset \mathbb{R}^{[0,1]}$ où $m \geq 0$ et les A_1, \dots, A_m sont des ouverts de \mathbb{R} . C'est-à-dire la plus petite topologie qui rend les applications $b_t : \omega \in \mathcal{C}([0, 1], \mathbb{R}) \mapsto \omega(t)$ continues pour tout $t \in [0, 1]$.

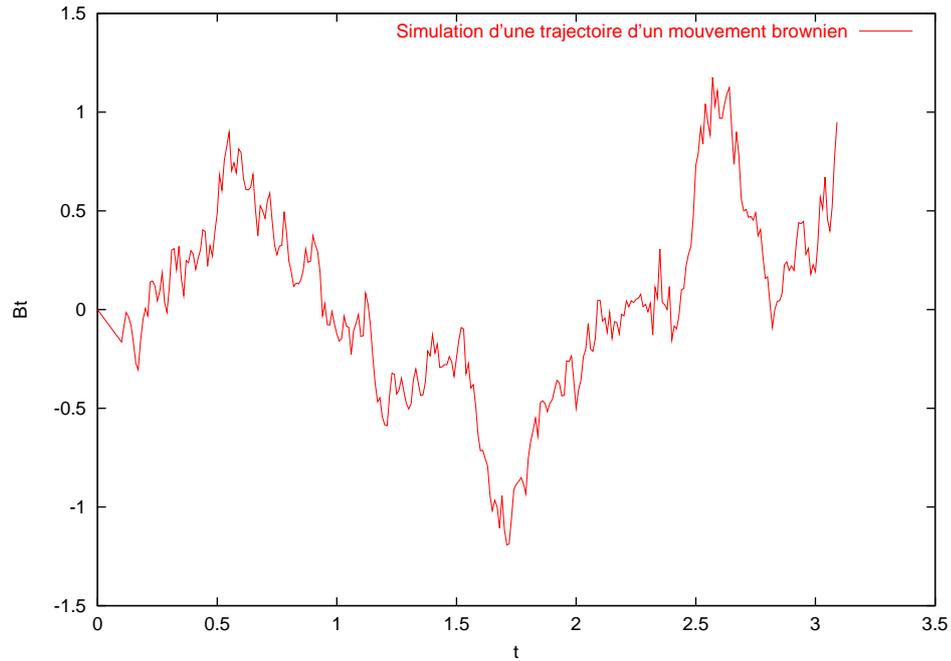


FIG. 1.6 – Simulation d’une trajectoire d’un processus gaussien à partir de sa fonction de covariance (mouvement brownien ici).

avec des processus à mémoire longue. Ainsi, on a pour un mouvement brownien standard $(B_t, t \in \mathbb{R}_+)$ avec $B_0 := 0$

$$(B_{t_1}, \dots, B_{t_n}) := (Z_1, Z_1 + Z_2, \dots, Z_1 + \dots + Z_n), \text{ où } Z_k := B_{t_k} - B_{t_{k-1}}.$$

Les Z_k sont des v.a.r. gaussiennes indépendantes de variances respectives $t_{k+1} - t_k$. Ceci est bien conforme à l’intuition que l’on peut avoir du mouvement brownien : une succession de petits sauts gaussiens indépendants. Voici un code Matlab faisant appel à cette méthode pour simuler une trajectoire brownienne, et dont la sortie graphique est donnée par la figure 1.6 :

```

1 %% Simulation d'une trajectoire brownienne
2 clear; a=0; b=1; n=5000; dt=(b-a)/n;
3 figure; clf; hold on;
4 plot([a:dt:b], cumsum(rnorm(n+1,0,sqrt(dt))));
5 title('Simulation_d'une_trajectoire_brownienne');
6 xlabel('Temps_t'); ylabel('Valeur_de_B(t)');
7 legend('Trajectoire_brownienne_unidimensionnelle_t->_B(t)_(omega_fixe)');

```

Cette méthode est beaucoup plus efficace que la précédente, mais repose de façon cruciale sur l’indépendance des accroissements du mouvement brownien, propriété que n’a pas forcément un processus gaussien quelconque. Comme nous l’avons dit, les trajectoires browniennes sont presque-sûrement continues. En revanche, elles ne sont pas dérivables, mais sont höldériennes d’indice $1/2 - \varepsilon$ pour tout $\varepsilon > 0$, c’est-à-dire que pour presque tout ω , il existe une constante $c > 0$ telle que pour tous t et s :

$$|B_t(\omega) - B_s(\omega)| \leq c |t - s|^{1/2 - \varepsilon}.$$

Cela a motivé l’introduction et l’étude de processus gaussiens de régularité höldérienne arbitraire, appelés « browniens fractionnaires », de fonction de covariance $(t^h + s^h + |t - s|^{2h})/2$ avec $0 < h < 1$ fixé (paramètre de Hurst). Le brownien fractionnaire d’indice h a des trajectoires höldériennes d’indice $h - \varepsilon$, et il correspond au mouvement brownien pour $h = 1/2$. Ces processus ne sont pas à accroissements indépendants, et ne sont pas de Markov, pour $h \neq 1/2$. Il permettent donc de modéliser des phénomènes à mémoire longue. À l’heure actuelle, de nouveaux processus à régularité non constante en temps ou en espace, qualifiés de « multifractionnaires », font l’objet de recherches dans le monde des probabilités.

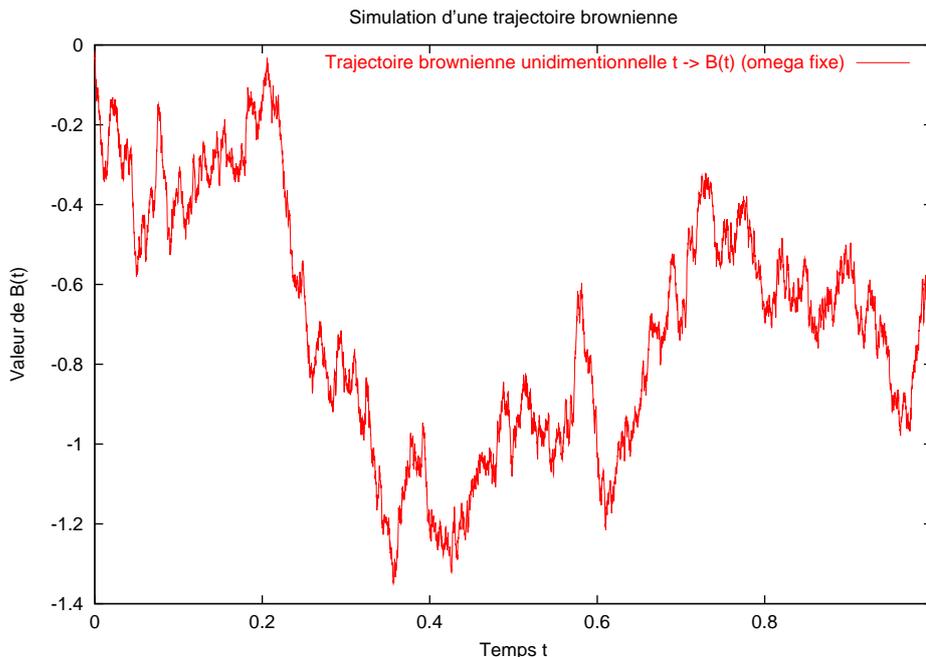


FIG. 1.7 – Simulation d'une trajectoire d'un mouvement brownien.

Remarque 1.6.2. Le mouvement brownien pouvant être vu comme une somme infinie de petits accroissements gaussiens indépendants, il est naturel de s'attendre à une sorte de « loi des grands nombres ». Cette intuition est juste, et l'on montre que l'on a :

$$\frac{B_t}{t} \xrightarrow[t \rightarrow +\infty]{p.s.} 0. \quad (1.3)$$

Remarque 1.6.3. Insistons sur le fait que chaque « trajectoire simulée » obtenue par ces procédés correspond plutôt à une réalisation $(X_{t_1}(\omega), \dots, X_{t_n}(\omega))$, pour un ω donné, de la loi fini-dimensionnelle du processus $(X_t, t \in \mathbb{R}_+)$ aux instants t_1, \dots, t_n . Simuler la véritable loi du processus n'a pas beaucoup de sens car il s'agirait alors de d'obtenir une courbe

$$t \in \mathbb{R}_+ \mapsto X_t(\omega),$$

or \mathbb{R} est infini donc inaccessible dans sa totalité sur un ordinateur ! On simule donc des lois marginales fini-dimensionnelles de la variable aléatoire à valeurs fonctionnelles ($\omega \in \Omega \mapsto (t \in \mathbb{R}_+ \mapsto X_t(\omega))$). Elle sont à percevoir comme des discrétisations de trajectoires probables du processus.

1.7 Simulation de la loi uniforme sur les p-sphères

Nous savons que si (X, Y) est un vecteur gaussien de loi normale centrée réduite, et si (r, θ) désigne son écriture en coordonnées polaires : $(X, Y) = (r \cos(\theta), r \sin(\theta))$, alors r^2 et θ sont des v.a.r. indépendantes qui suivent respectivement une loi exponentielle de paramètre 1/2 et une loi uniforme sur $[0, 2\pi]$. Le vecteur aléatoire $(X, Y)/r$ suit la loi uniforme sur le cercle unité. De façon générale, si (X_1, \dots, X_n) est un vecteur gaussien de loi $\mathcal{N}(0, I_n)$ et

$$\|X\|_2 := \sqrt{X_1^2 + \dots + X_n^2},$$

alors $(X_1/\|X\|_2, \dots, X_n/\|X\|_2)$ et $\|X\|_2$ sont indépendants, $\|X\|_2^2$ suit une loi du χ^2 à n degrés de liberté, et $(X_1/\|X\|_2, \dots, X_n/\|X\|_2)$ suit la loi uniforme sur la sphère euclidienne unité de \mathbb{R}^n . Attention, les v.a.r. $X_i/(\sqrt{n}\|X\|_2)$ ne suivent pas des lois de Student, car $\|X\|_2$ et X_i ne sont pas indépendantes !

Ce qui précède peut s'étendre à des normes non euclidiennes : pour tout $n \in \{2, 3, \dots\}$, tout $p \in \mathbb{R}_+^*$ et tout $x \in \mathbb{R}^n$, on pose

$$\|x\|_p := (|x_1|^p + \dots + |x_n|^p)^{1/p}.$$

On étend cette définition au cas $p = \infty$ en posant

$$\|x\|_\infty := \max(|x_1|, \dots, |x_n|).$$

La p -sphère positive de \mathbb{R}^n , notée $\mathbb{S}(n, p)$, est définie par :

$$\mathbb{S}_+(n, p) := \left\{ x \in \mathbb{R}_+^n, \|x\|_p = 1 \right\}.$$

Pour $p = 1$, on obtient le « simplexe » standard, pour $p = 2$, la portion à coordonnées positives de la sphère euclidienne de rayon 1, et pour $p = \infty$, la surface du cube unité $[0, 1]^n$. Des dessins peuvent aider à mieux se repérer. $\mathbb{S}_+(n, p)$ est donc une surface compacte de \mathbb{R}^n , de dimension $n - 1$. Il serait possible de définir une mesure uniforme sur $\mathbb{S}_+(n, p)$, en considérant la trace de la mesure de Lebesgue sur $\mathbb{S}_+(n, p)$. Par soucis de simplicité, nous ne le feront pas ici, et nous nous restreignons volontairement aux cas $p \in \{1, 2, \infty\}$.

En tant que surface du cube $[0, 1]^n$, $\mathbb{S}_+(n, \infty)$ peut être vu comme la réunion disjointe, mais connexe, de $2n$ faces, chacune étant identifiable à $[0, 1]^{n-1}$. La mesure uniforme sur $\mathbb{S}_+(n, \infty)$ peut donc être définie en considérant la mesure uniforme sur chaque face. Cette loi uniforme sur $\mathbb{S}_+(n, \infty)$ est très simple à simuler puisqu'il suffit de choisir la face uniformément sur $\{1, \dots, 2n\}$, puis de générer une réalisation d'un vecteur aléatoire (X_1, \dots, X_{n-1}) dont les entrées sont i.i.d. et de loi uniforme sur $[0, 1]$, obtenues par **rand** par exemple. Si l'on désigne par $(A, I) \in \{0, 1\} \times \{1, \dots, n\}$ la face aléatoirement choisie, le vecteur aléatoire (Y_1, \dots, Y_n) obtenu en insérant A en position I de la suite X_1, \dots, X_{n-1} suit la loi uniforme sur $\mathbb{S}_+(n, \infty)$.

Pour $p = 2$, la loi uniforme sur $\mathbb{S}_+(n, 2)$ est la restriction normalisée de la mesure uniforme sur la sphère. Cette dernière se simule comme expliqué à la fin de la section 1.6, à partir d'une loi gaussienne. En fait, si (X_1, \dots, X_n) est un vecteur aléatoire à composantes i.i.d. positives de loi de densité $\sqrt{2/\pi} \exp(-x^2/2) \mathbb{I}_{[0, +\infty[} dx$, alors le vecteur aléatoire $X/\sqrt{X_1^2 + \dots + X_n^2}$ suit la loi uniforme sur $\mathbb{S}_+(n, 2)$.

Pour $p = 1$, $\mathbb{S}_+(n, 1)$ est un simplexe, qui est constitué de la partie à coordonnées positives de l'hyperplan \mathbb{H}_n d'équation $x_1 + \dots + x_n = 1$. Pour $n = 2$, on obtient un segment de longueur $\sqrt{2}$, pour $n = 3$, un triangle équilatéral de côté $\sqrt{2}$, pour $n = 4$, un tétraèdre régulier de côté $\sqrt{2}$, etc. Mentalement, nous effectuons naturellement une identification entre \mathbb{H}_n et \mathbb{R}^{n-1} lorsque n vaut 2 et 3, en identifiant par rotation de centre $(1, 0, \dots, 0)$ l'hyperplan \mathbb{H}_n à $\mathbb{R}^{n-1} \times \{0\}$. En considérant la mesure Lebesgue λ_{n-1} sur \mathbb{H}_n découlant de celle de \mathbb{R}^{n-1} par cette identification, l'ensemble $\mathbb{S}_+(3, 2)$ par exemple a pour mesure $\sqrt{3}/2$, qui est la surface du triangle équilatéral de côté $\sqrt{2}$ (souvenez-vous, « base fois hauteur sur 2 »). De façon plus générale, on montre que $\lambda_{n-1}(\mathbb{S}_+(n, 1)) = \sqrt{n}/(n-1)!$. Il est naturel d'appeler loi uniforme sur $\mathbb{S}_+(n, 1)$ la restriction de λ_{n-1} à $\mathbb{S}_+(n, 1)$, normalisée en la divisant par $\lambda_{n-1}(\mathbb{S}_+(n, 1))$. Comment simuler cette loi? En d'autres termes, comment faire pour obtenir des réalisations i.i.d. d'un vecteur aléatoire (X_1, \dots, X_n) dont la loi serait la loi uniforme sur $\mathbb{S}_+(n, 1)$ que nous venons de décrire?

On pourrait croire que si $X := (X_1, \dots, X_n)$ est un vecteur aléatoire de \mathbb{R}^n à coordonnées i.i.d. et uniformes sur $[0, 1]$, alors le vecteur aléatoire $X/(X_1 + \dots + X_n)$, qui est à valeurs dans $\mathbb{S}_+(n, 1)$, suit la loi uniforme sur $\mathbb{S}_+(n, 1)$. C'est faux malheureusement, comme on peut s'en convaincre déjà lorsque $n = 2$.

La solution consiste à imiter la méthode gaussienne utilisée pour $p = 2$. Si $X := (X_1, \dots, X_n)$ est un vecteur aléatoire à coordonnées i.i.d. de loi exponentielle de paramètre 1, alors le vecteur aléatoire $X/(X_1 + \dots + X_n)$ suit la loi uniforme sur $\mathbb{S}_+(n, 1)$. La preuve n'est pas difficile, mais nous l'omettons ici.

De manière plus générale, pour tout $p \in \mathbb{R}_+$, on peut montrer que si $X := (X_1, \dots, X_n)$ est un vecteur aléatoire à composantes i.i.d. suivant une loi de densité

$$\frac{p}{\Gamma(p)} \exp(-t^p) \mathbb{I}_{\mathbb{R}_+}(t)$$

alors le vecteur aléatoire $X/\|X\|_p$ suit la loi uniforme sur $\mathbb{S}_+(n, p)$, et qu'il est de plus indépendant de la v.a.r. $\|X\|_p$.

Le cas $p = 1$, pour lequel $\mathbb{S}_+(n, 1)$ est un simplexe, est très important car $\mathbb{S}_+(n, 1)$ représente l'ensemble des lois de probabilités discrètes sur $\{1, \dots, n\}$. Simuler la loi uniforme sur $\mathbb{S}_+(n, 1)$ revient à choisir uniformément une loi de probabilité discrète sur $\{1, \dots, n\}$. La loi uniforme sur $\mathbb{S}_+(n, 1)$ est un cas particulier de la loi de Dirichlet, cf. 8.4.6. Voici un exemple de fonction Matlab qui renvoie une réalisation de la loi de Dirichlet :

```

1 function rdir=rdirichlet(n,alpha);
2 %RDIRICHLET renvoie une réalisation de la loi de Dirichlet de taille n
3 %et de paramètre alpha. C'est la loi du vecteur aléatoire
4 % (x_1/(x1+...+xn), ... , x_n/(x1+...+xn)) lorsque les xi sont i.i.d.
5 %et suivent des lois gamma de paramètre respectifs (alpha(i),1).
6 %Ainsi, lorsque alpha=(1,...,1), on obtient la loi uniforme sur le
7 %simplexe des lois de probabilités discrètes sur {1,...,n}.
8 %Le paramètre alpha doit être un vecteur de taille n à coordonnées
9 %strictement positives. Il est optionnel est vaut par
10 %défaut (1,...,1). La valeur de retour rdir est un vecteur ligne.
11 %Les appels successifs donnent des réalisations pseudo-i.i.d. puisque
12 %rdirichlet fait appel à rgamma pour simuler la loi gamma et à
13 %rexpweib(.,1,1) si alpha est omis.
14 %rdir=rdirichlet(n);
15 if (nargin==1),
16     % Loi expo, on peut aussi utiliser gamma(n,1)
17     x=rexpweib(n,1,1);
18     rdir=reshape(x/sum(x),[1,n]);
19 else
20     if (length(alpha)~=~n), error('alpha_doit_etre_de_dim_n'); end;
21     if ~ (all(alpha>0)), error('alpha_doit_etre_a_coords_>0'); end;
22     for i=1:n,
23         x(i)=rgamma(1,alpha(i));
24     end
25     rdir=reshape(x/sum(x),[1,n]);
26 end
27 return;

```

Cette méthode permet également de choisir de manière « uniforme » la matrice de transition d'une chaîne de Markov, en utilisant des réalisations indépendantes de la loi uniforme sur le simplexe. Voici un exemple de code Matlab :

```

1 function P = rmatmar(n)
2 %RMATMAR renvoie une matrice de Markov aléatoire de taille n x n
3 %dont les lignes sont des probabilités discrètes choisies de façon
4 %uniforme et indépendante en utilisant rdirichlet.
5 %P = rmatmar(n);
6 for i=1:n, P(i,:)= rdirichlet (n); end;
7 return;

```

Bien entendu, tout repose sur le sens que l'on donne à la « loi uniforme » sur les classes d'objets considérés. Il y a une part d'arbitraire que l'on ne peut pas évacuer.

1.8 Un exemple : processus de Poisson composé gaussien

Soit $(X_n, n \in \mathbb{N}^*)$ une suite de v.a.r. i.i.d. de loi normale et $(U_n, n \in \mathbb{N}^*)$ une suite de v.a.r. i.i.d. de loi exponentielle de paramètre $\lambda > 0$, indépendante de la suite précédente. On pose $T_0 := 0$ et $T_n := U_1 + \dots + U_n$ pour tout $n \in \mathbb{N}$. On définit à présent le processus $(Y_t, t \in \mathbb{R}_+)$ par :

$$Y_t := \sum_{n=0}^{+\infty} X_n I_{[T_n, T_{n+1}[}(t).$$

Ce type de processus peut par exemple modéliser la quantité d'argent dans une caisse commune. Les temps de sauts correspondent à des apports (sauts positifs) des membres ou à des dépenses (sauts négatifs). Bien entendu, on tolère alors des dettes arbitrairement grandes ici... On dit que $(Y_t, t \in \mathbb{R}_+)$ suit un *processus de Poisson composé*. Cette terminologie n'est pas très heureuse dans la mesure où la loi du processus à un instant donné n'est pas forcément une loi de Poisson. Ici, on peut vérifier sans difficulté (exercice!) que le processus est bien défini et que sa loi conditionnelle sachant les instants de sauts $(T_n, n \in \mathbb{N}^*)$ est gaussienne. La loi de Y_t pour $t > 0$ fixé n'est pas une loi de Poisson. Voici un code Matlab simulant une trajectoire d'un tel processus, dont la sortie graphique est donnée par la figure 1.8 :

```

1 %% Code concis pour simuler un processus de Poisson Composé
2 %% à sauts gaussiens et temps de sauts expo.
3 %% Fait appel à rexpweib de Stixbox
4 clf; xlabel('t'); ylabel('Yt')
5 legend('Trajectoire d''un processus a sauts gaussiens et expo.',4)
6 stairs(cumsum(rexpweib(30,1/2,1)),cumsum(randn(1,30)));

```

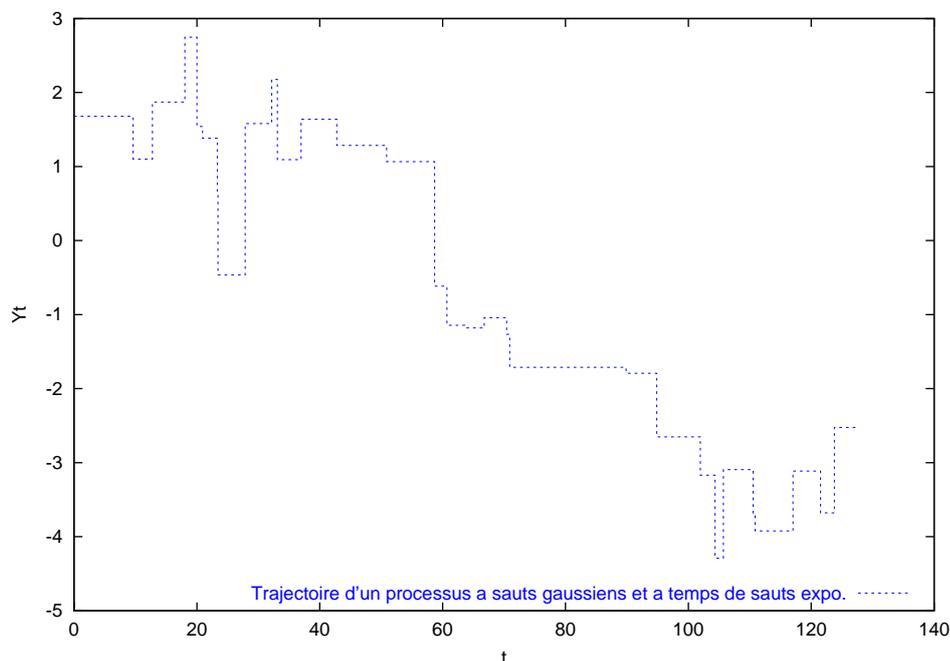


FIG. 1.8 – Simulation d’une trajectoire d’un processus à temps de sauts exponentiels et à amplitudes de sauts gaussiennes.

Exercice 1.8.1. À votre avis, que permet de simuler le code suivant ?

```
stairs(cumsum(rgamma(30,3,1)),cumsum(rpoiss(30,1)+randn(30,1)));
```

1.9 Évaluation de la qualité d’une simulation

De nombreuses méthodes existent. Les tests statistiques, comme le test du chi-deux ou le test de Kolmogorov-Smirnov, permettent de comparer la loi empirique avec la loi véritable, cf. chapitre 3 page 55.

1.10 Un exemple de simulation d’une loi compliquée

On se propose de transposer en Matlab un petit exercice qui figure dans [Bou86, page 82]. La figure 1.9 représente un réseau électrique constitué de 7 éléments entre deux points A et B . Les durées de vie des éléments sont représentées par des variables aléatoires T_1, \dots, T_7 . La durée de vie du système est alors donnée par

$$T := T_7 \vee ((T_1 \vee T_2 \vee T_3) \wedge (T_6 \vee (T_4 \wedge T_5)))$$

où \vee désigne le supremum et \wedge l’infimum. On peut donc simuler T à partir d’une simulation de la loi jointe des T_i . On considère le cas où (T_1, T_2, T_3, T_4) sont indépendants de loi exponentielles de paramètres respectifs $(1/2, 1/2, 1/2, 1/3)$ et que le couple (T_5, T_6) sont indépendantes des précédentes, de loi jointe de densité

$$\frac{1}{6} \mathbf{1}_{\{2 \leq x+y \leq 8\}} \mathbf{1}_{\{-1 \leq x-y \leq 1\}} dx dy,$$

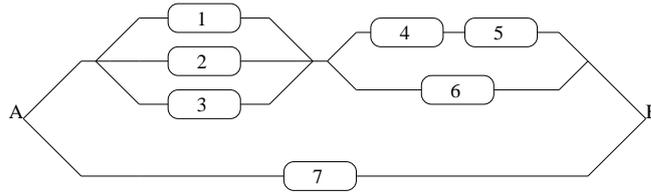


FIG. 1.9 – Schéma d'un réseau

et enfin, $T_7 = T_4$. Notons que le couple $((T_5 + T_6 - 2)/6, (T_5 - T_6 + 1)/2)$ suit une loi uniforme sur $[0, 1]^2$, produit tensoriel de deux lois uniformes sur $[0, 1]$ (donc indépendantes).

```

1 n=input('Taille_des_échantillons_?');
2 R=rand(n,6); % n réel. indép. de loi unif sur [0,1]
3 T1=-2*log(R(:,1)); % n réel. indép. de T4
4 T2=-2*log(R(:,2)); % n réel. indép. de T4
5 T3=-2*log(R(:,3)); % n réel. indép. de T4
6 T4=-3*log(R(:,4)); % n réel. indép. de T4
7 T7=T4; % n réel. indép. de T7
8 T5=3*R(:,5)+R(:,6)+1/2; % n réel. indép. de T5
9 T6=3*R(:,5)-R(:,6)+3/2; % n réel. indép. de T6
10 % n réel. indép. de T
11 T=max(T7,min(max(T1,max(T2,T3)),max(T6,min(T4,T5))));
12 % moyenne et écart type empiriques de T
13 M=[mean(T),std(T)];
14 % histogramme de l'échantillon
15 hist(T,fix(sqrt(n)));

```

Intéressons nous à la durée de vie moyenne du système, qui n'est rien d'autre que $\mathbb{E}(T)$. Si X_1, \dots, X_n sont n v.a.r. i.i.d. de même loi que T , nous savons simuler des réalisations $X_1(\omega), \dots, X_n(\omega)$. L'inégalité de Chebychev donne :

$$\mathbb{P}\left(\left|\frac{X_1 + \dots + X_n}{n} - \mathbb{E}(T)\right| > \varepsilon\right) \leq \frac{\mathbf{Var}(T)}{n\varepsilon^2}.$$

La quantité $(X_1 + \dots + X_n)/n$ représente la moyenne empirique que nous pouvons obtenir par simulation. Reste à majorer la variance $\mathbf{Var}(T)$. L'encadrement $T_7 \leq T \leq T_7 + T_1 + T_2 + T_3$ permet d'écrire :

$$\begin{aligned} \mathbf{Var}(T) &= \mathbf{E}(T^2) - \mathbf{E}(T)^2 \\ &\leq \mathbf{E}((T_7 + T_1 + T_2 + T_3)^2) - \mathbf{E}(T_7)^2 \\ &= \mathbf{Var}(T_7 + T_1 + T_2 + T_3) + (\mathbf{E}(T_7) + \mathbf{E}(T_1) + \mathbf{E}(T_2) + \mathbf{E}(T_3))^2 - \mathbf{E}(T_7)^2 \\ &= \frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2} + \frac{1}{\lambda_3^2} + \frac{1}{\lambda_4^2} + \left(\frac{1}{\lambda_1} + \frac{1}{\lambda_2} + \frac{1}{\lambda_3} + \frac{1}{\lambda_4}\right)^2 - \frac{1}{\lambda_4^2} \\ &= 93. \end{aligned}$$

On constate empiriquement que cette majoration de la variance est très mauvaise puisque la variance empirique est plutôt de l'ordre de quelques unités. Cependant, elle a le mérite d'exister. On peut donc affirmer que l'on a

$$\mathbb{P}\left(\left|\frac{X_1 + \dots + X_n}{n} - \mathbb{E}(T)\right| > \frac{1}{10}\right) \leq \frac{1}{10},$$

pour $n \geq n_0 := 93000$. Bien évidemment, ce seuil n_0 sur n sera d'autant meilleur (i.e. petit) que notre majoration de la variance $\mathbf{Var}(T)$ sera bonne.

Exercice 1.10.1. Supposons que T_7 suive une loi exponentielle indépendante des autres de paramètre θ . Écrire un programme Matlab qui détermine une valeur de θ telle que $\mathbf{E}(T_7)$ soit minimale et $\mathbf{E}(T) \geq 10$.

1.10.1 Rudiments de fiabilité

Soit T la date de première défaillance d'un système mis en marche à la date $t = 0$, alors :

- La *durée de vie du système* est la variable T .
- La *fiabilité* à la date t est : $R(t) := \mathbb{P}(T \geq t)$.
- La *durée de survie* du système à l'instant t est la variable T_t de distribution :

$$\mathbb{P}(T_t \geq s) = \mathbb{P}(T - t \geq s \mid T \geq t) = \frac{R(t+s)}{R(t)} \text{ pour } s \geq 0.$$

- Le *taux de défaillance* ou *taux de panne* du système est la fonction :

$$h(t) := -\frac{R'(t)}{R(t)}.$$

- Le *temps moyen de panne* du système (Mean Time To Failure, MTTF) est : $m := \mathbb{E}(T)$.

Soit un système à n composantes, de durée de vie respectives T_1, \dots, T_n . On note T la durée de vie du système. Alors:

- Si les composants sont en série, $T = \min(T_1, \dots, T_n)$ et $R(t) = R_1(t)R_2(t) \dots R_n(t)$.
- Si les composants sont en parallèle, $T = \max(T_1, \dots, T_n)$ et $R(t) = 1 - (1 - R_1(t))(1 - R_2(t)) \dots (1 - R_n(t))$.

La loi exponentielle (de paramètre λ) est souvent utilisée pour modéliser une durée de vie T , sa fiabilité est $R(t) = e^{-\lambda t} \mathbf{I}_{\mathbb{R}_+}(t)$. Sa durée de survie à l'instant t est indépendante de t , et l'on parle alors de *système sans mémoire* :

$$\mathbb{P}(T \geq t + s \mid T \geq t) = \mathbb{P}(T \geq s).$$

On peut montrer que la loi exponentielle est la seule loi non triviale à support dans \mathbb{R}^+ qui a cette propriété. La somme de n v.a. i.i.d. de loi exponentielle de paramètre λ est une loi Gamma de paramètres (λ, n) , dite aussi loi de Erlang. Une autre loi utilisée pour modéliser les durées de vie est la de Weibull $W(a, b)$, définie à partir de sa fiabilité, qui vaut par définition $R(t) = 1 - F(t) = e^{-(bt)^a} \mathbf{I}_{\{t \geq 0\}}$. Elle modélise bien le comportement de a composants de loi exponentielle de paramètre b en parallèle pour t pas trop grand. La loi exponentielle de paramètre $\lambda > 0$ correspond à $W(\lambda, 1)$.

Remarque 1.10.2 (Terminologie anglaise). En anglais, la densité est souvent appelée « probability density function » (pdf), et la fonction de répartition « distribution function » ou encore « cumulative distribution function » (cdf). La connaissance de cette terminologie peut vous aider à la compréhension de l'aide des fonctions de Stixbox. Les noms des générateurs aléatoires de Stixbox commencent par la lettre **r** comme « random », les densités ou lois par **d** comme « density », les fonctions de répartitions inverses par **q** comme « quantile », les fonctions de répartitions par **p** comme « probability » et enfin les générateurs aléatoires via la méthode de rejet par **rj** comme « reject ».

Loi	Densité	F. de Rép.	F. R. Inverse	Générateur	Remarques
Beta	dbeta	pbeta	qbeta	rbeta	
Binômiale	dbinom	pbinom	qbinom	rbinom	Voir aussi rjbinom
Chi-2	dchisq	pchisq	qchisq	rchisq	
Fisher	df	pf	qf	rf	
Gamma	dgamma	pgamma	qgamma	rgamma	Voir aussi rjgamma
Géométrique				rgeom	
Hyper géom.	dhyppg	phyppg	qhyppg	rhyppg	
Normale	dnorm	pnorm	qnorm	rnorm	Voir aussi randn de Matlab
Poisson				rpoiss	Voir aussi rjpoiss
Student	dt	pt	qt	rt	
Weibull & exp.				rexpweib	
Kolmog.-Smir.		pks			

TAB. 1.1 – Générateurs aléatoires de la bibliothèque Stixbox.

Fragment 2

Théorèmes limites classiques

Dans ce chapitre, on s'intéresse aux « théorèmes limites » classiques comme la loi des grands nombres (LGN) et le théorème central limite (TCL), ainsi qu'à certains de leurs raffinements et extensions.

2.1 Lois des grands nombres & théorème de Glivenko-Cantelli

Pour des variables aléatoires centrées et de carré intégrable, l'indépendance deux à deux entraîne l'orthogonalité. La somme de n vecteurs orthogonaux de même norme étant de l'ordre de \sqrt{n} , on s'attend donc à ce que la moyenne arithmétique des v.a. soit de l'ordre de $1/\sqrt{n}$, et donc à ce qu'elle converge vers 0. C'est ce que confirme la loi des grands nombres (LGN) :

Théorème 2.1.1 (Loi des Grands Nombres). *Soit $(X_n, n \in \mathbb{N}^*)$ une suite de variables aléatoires i.i.d. de loi μ sur \mathbb{R} , alors leur moyenne arithmétique*

$$S_n := \frac{X_1 + \cdots + X_n}{n}$$

converge μ -presque-surement si et seulement si les X_n sont intégrables. De plus, dans ce cas, la limite vaut $m_\mu := \mathbf{E}(X_1)$ et la convergence a également lieu dans \mathbf{L}^1 .

Ce théorème est très facile à démontrer si l'on renforce les hypothèses. Supposons par exemple que les v.a. $(X_n, n \in \mathbb{N}^*)$ sont i.i.d., dans \mathbf{L}^4 et indépendantes 4 à 4. On peut supposer sans perdre de généralité qu'elles sont centrées. On a alors :

$$\mathbf{E}\left((S_n)^4\right) = \frac{1}{n^3} \mathbf{E}(X_1^4) + \frac{4n(n-1)}{n^4} \mathbf{E}(X_1^2),$$

d'où $\sum_{n=1}^{+\infty} \mathbf{E}\left((S_n)^4\right) < +\infty$, or d'après le théorème de convergence monotone :

$$\sum_{n=1}^{+\infty} \mathbf{E}\left((S_n)^4\right) = \mathbf{E}\left(\sum_{n=1}^{+\infty} (S_n)^4\right).$$

Cette espérance étant finie, la v.a. $\sum_{n=1}^{+\infty} (S_n)^4$ est finie p.s. et donc $(S_n)^4$ converge p.s. vers 0 en tant que terme général d'une série convergente, d'où la convergence p.s. vers 0 de S_n , CQFD.

Remarque 2.1.2. On parle souvent de « loi faible des grands nombres » lorsque seule la convergence en probabilité est énoncée. Pour les lois discrètes, cette version faible est ancienne et remonte sans doute aux travaux de Markov. D'autre part, le fait qu'il y ait convergence \mathbf{L}^1 dans la LGN est souvent désigné sous le nom de « loi des grands nombres de Kolmogorov ».

Remarque 2.1.3. Les hypothèses faites sur la suite des v.a. dans la LGN peuvent être considérablement affaiblies, au détriment de la preuve qui est alors plus difficile (par exemple le théorème d'Etemadi qui ne requière que l'indépendance deux à deux [Bor95]). De façon générale, la LGN reste encore vraie lorsque les v.a. n'ont pas la même loi – il faut toute fois contrôler certains moments. Nous avons évoqué une sorte de LGN pour le processus de Poisson (équation (1.2) page 25) et pour le mouvement brownien (équation (1.3) page 32). Nous verrons plus généralement dans le chapitre 5 page 71 que des théorèmes du type LGN et TCL existent pour des suites de variables aléatoires « dépendantes ».

2.1.1 Mesure empirique

Soit $(X_i, n \in \{1, \dots, n\})$ une suite finie de variables aléatoires i.i.d. de loi μ sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$. En statistique, on parle de *n-échantillon* de loi μ . On note $F : \mathbb{R} \rightarrow [0, 1]$ la fonction de répartition de la loi μ , donnée par $F(t) := \mu(-\infty, t]$. À l'échantillon $(X_i, n \in \{1, \dots, n\})$, on associe une *mesure empirique* $L_n(\omega)$ définie par

$$L_n(\omega) := \frac{1}{n} \sum_{i=1}^n \delta_{X_i(\omega)},$$

et une *fonction de répartition empirique* $F_n(\omega)(\cdot) : \mathbb{R} \rightarrow [0, 1]$ par

$$F_n(\omega)(t) := \frac{1}{n} \sum_{i=1}^n \mathbf{I}_{\{X_i \leq t\}}(\omega).$$

En regroupant suivant les valeurs des $X_i(\omega)$, on obtient les écritures suivantes faisant apparaître les *effectifs cumulés* et les *fréquences empiriques* :

$$L_n(\omega) = \sum_{x \in \{X_1(\omega), \dots, X_n(\omega)\}} \frac{\#\{1 \leq i \leq n, X_i(\omega) = x\}}{n} \delta_x,$$

et :

$$F_n(\omega)(t) = \frac{1}{n} \#\{1 \leq i \leq n, X_i(\omega) \leq t\}.$$

Notons que L_n est une variable aléatoire à valeurs dans l'ensemble des lois de probabilité sur \mathbb{R} , et que F_n est une variable aléatoire à valeurs dans l'ensemble des fonctions de répartition sur \mathbb{R} . Ainsi, pour tout $\omega \in \Omega$, $L_n(\omega)$ est une loi sur \mathbb{R} , de fonction de répartition $F_n(\omega)(\cdot)$.

Une conséquence de la LGN est que la mesure empirique $L_n(\omega)$ converge étroitement vers la loi μ . On rappelle qu'une suite de mesures de probabilités $(\mu_n)_n$ converge étroitement vers une mesure de probabilité μ lorsque pour toute fonction bornée f , $\mathbf{E}_{\mu_n}(f)$ converge vers $\mathbf{E}_{\mu}(f)$ quand $n \rightarrow \infty$. Une suite de variables aléatoires converge en loi si et seulement si la suite de leurs lois converge étroitement. Dans le cas des mesures empiriques, on a pour μ -presque tout ω :

$$L_n(\omega) \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mu.$$

Pour le voir, il suffit d'écrire, pour une fonction $f \in \mathcal{C}_b(\mathbb{R}, \mathbb{R})$,

$$\mathbf{E}_{L_n(\omega)}(f) = \frac{1}{n} \sum_{i=1}^n f(X_i(\omega)).$$

Les variables aléatoires $Y_i := f(X_i)$ sont i.i.d. et la loi des grands nombres nous assure la convergence presque sûre de $\mathbf{E}_{L_n(\omega)}(f)$ vers $\mathbf{E}(Y_1) = \mathbf{E}_{\mu}(f)$. La permutation des quantificateurs universels sur f et ω se fait en utilisant la séparabilité de \mathbb{R} . Ce résultat de convergence étroite des mesures empiriques peut être affiné en une convergence uniforme des fonctions de répartition, comme le précise le théorème de Glivenko-Cantelli :

Théorème 2.1.4 (Glivenko-Cantelli). *Soit $(X_n, n \in \mathbb{N}^*)$ une suite de variables aléatoires i.i.d. de loi μ sur \mathbb{R} , de fonction de répartition F , alors on a pour μ -presque-tout ω*

$$\lim_{n \rightarrow \infty} \left(\sup_{t \in \mathbb{R}} |F_n(\omega)(t) - F(t)| \right) = 0, \quad (2.1)$$

où $F_n(\omega)(\cdot)$ désigne la fonction de répartition empirique des $X_1(\omega), \dots, X_n(\omega)$.

Exercice 2.1.5 (Polynômes de Bernstein & théorème de Weierstrass). Le théorème de Weierstrass sur la densité de $\mathbb{R}[X]$ dans $\mathcal{C}([0, 1], \|\cdot\|_{\infty})$ peut-être établi en utilisant une forme faible de la loi des grands nombres (convergence en probabilité de S_n vers la moyenne de la loi μ).

En effet, si $f : [0, 1] \rightarrow \mathbb{R}$ est continue, le polynôme de Bernstein

$$P_n(f)(x) := \sum_{k=0}^n x^k (1-x)^{n-k} C_n^k f(k/n)$$

converge uniformément vers f sur $[0, 1]$. Pour le voir, il suffit d'exprimer $P_n(f)(x)$ comme la moyenne de f sous la loi binomiale de paramètre x sur $\{0, 1/n, \dots, 1\}$, puis de découper autour de la moyenne p et d'utiliser l'inégalité de Chebychev et l'absolue continuité de f (théorème de Heine). Pour illustrer ce qui précède, créer un fichier `ber.m` contenant le code suivant :

```

1 function Y=ber(X,n,f)
2 %BER Approximation de Bernstein sur [0,1]
3 % Y=ber(X,n,f)
4 % Renvoie l'approximation de Bernstein y=Pn(f)(X) de f(X).
5 % X est un vecteur quelconque a valeurs dans [0,1]
6 % Fait appel a StixBox pour les coef. binomiaux
7 %
8 % Exemple : X=[.1:.1:.9]; f=inline('x.^4.*sin(x)'); ber(X,100,f)
9 %
10 m=length(X);
11 D= repmat([0:n],m,1); % Réplication de D
12 B=bincoef(D,n); % Calcul des coefficients binomiaux
13 F= repmat(feval(f,[0:(1:n)/n]),m,1); % Calcul et réplication de f
14 X=reshape(X,1,m); % Remise en forme de X
15 XX= repmat(X',1,n+1); % Réplication de X
16 P=(XX.^D) .* ((1-XX).^(n-D)); % Calcul des coefficients en X
17 Y=sum((B.*F.*P)'); % Calcul de la valeur du polynôme

```

Voici maintenant un petit programme faisant appel à cette fonction, dont la sortie graphique est donnée par la figure 2.1 :

```

1 % Trace la fonction log(1+x)+x^4 sin(10*x) sur [0,1] ainsi que
2 % les polynômes de Berntein associes de degre 5, 10, et 20.
3 % Utilise la fonction ber pour le calcul des polynômes de Berstein
4 clf;
5 clear;
6 f=inline('log(1+x)+x.^4.*sin(10*x)');
7 S = [.01:.01:.99]; % subdivision de [0,1]
8 F=feval(f,S); % image de la subdivision par notre fonction
9 B1=ber(S,5,f);
10 B2=ber(S,10,f);
11 B3=ber(S,20,f);
12 plot(S,F,'k-',S,B1,'r.',S,B2,'b+',S,B3,'g*');
13 legend('f','B(5,f)','B(10,f)','B(20,f)');

```

La méthode de Monte-Carlo pour le calcul approché d'intégrales est également basée sur la loi des grands nombres, cf. section 3.2.5 page 60.

2.2 Théorème Central Limite & principe d'invariance

Poursuivant l'analogie avec les vecteurs orthogonaux d'un espace euclidien (cf. section 2.1 page 39), on s'attend à ce que la somme des n premiers vecteurs divisée par \sqrt{n} « n'explose pas » lorsque n tend vers l'infini. Cette intuition est confirmée par le Théorème Central Limite (TCL¹) :

Théorème 2.2.1 (Théorème Central Limite). *Si $(X_n, n \in \mathbb{N}^*)$ est une suite de v.a.r. i.i.d. de carré intégrable de moyenne m et de variance σ^2 , alors*

$$Y_n := \frac{X_1 + \dots + X_n - nm}{\sigma\sqrt{n}} \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, 1).$$

De manière équivalente, si Q_n désigne la fonction de répartition de la v.a.r. Y_n et Φ la fonction de répartition de la loi normale $\mathcal{N}(0, 1)$:

$$\Phi(t) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t \exp\left(-\frac{x^2}{2}\right) dx,$$

¹On parle également parfois de « Théorème de la Limite Centrale ». En anglais, on dit tout simplement « Central Limit Theorem », que l'on abrège en CLT.

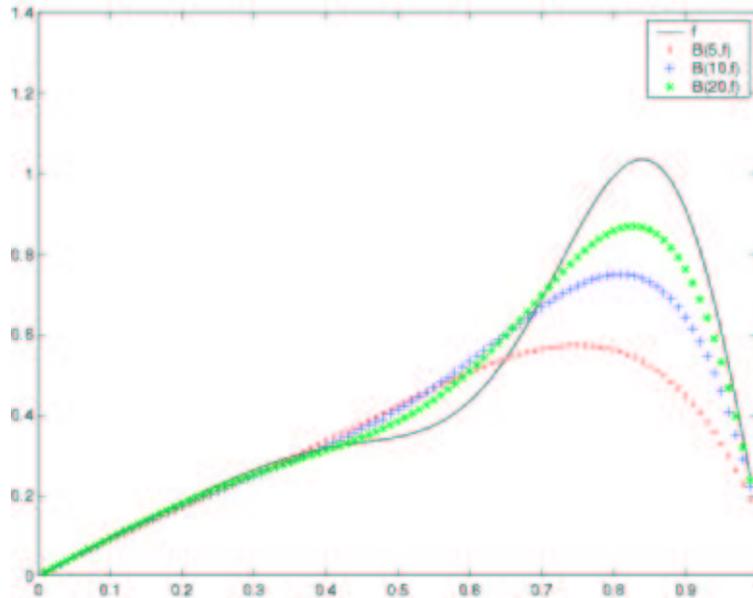


FIG. 2.1 – Illustration du théorème de Weierstrass avec les polynômes de Bernstein.

alors on a pour tout t dans \mathbb{R} :

$$Q_n(t) \xrightarrow[n \rightarrow +\infty]{} \Phi(t).$$

On peut également utiliser le théorème de Lévy de caractérisation de la convergence en loi par la transformée de Fourier :

$$(\forall s \in \mathbb{R}) \mathbf{E}(e^{isY_n}) \xrightarrow[n \rightarrow +\infty]{} e^{-s^2/2}.$$

Pour établir le TCL, on peut supposer sans perdre de généralité que les v.a. sont centrées réduites, on a alors :

$$\mathbf{E}(e^{isY_n}) = \left(\mathbf{E}(e^{isn^{-1/2} X_1}) \right)^n,$$

qui donne le résultat par un développement en série entière, mais il faut prendre garde au fait que l'on a affaire à une fonction complexe dont il faut choisir la détermination...

De même que pour la LGN, de nombreuses versions du TCL existent, notamment avec des hypothèses plus faibles sur les lois des v.a. On peut par exemple affaiblir l'indépendance requise ou encore se passer du fait que les v.a. aient la même loi, moyennant une hypothèse sur leurs moments. Historiquement, le premier TCL concernait les lois de Bernoulli :

Théorème 2.2.2 (Théorème de Moivre-Laplace). Soit $(X_n, n \in \mathbb{N}^*)$ des v.a. i.i.d. de loi de Bernoulli sur $\{0, 1\}$ et de paramètre $p \in [0, 1]$, alors, pour tout $a \in \mathbb{R}$ et $b \in \mathbb{R}$ avec $a < b$, on a :

$$\lim_{n \rightarrow +\infty} \mathbb{P}\left(a \leq \frac{S_n - np}{\sqrt{npq}} \leq b\right) = (2\pi)^{-1/2} \int_a^b e^{-x^2/2} dx,$$

où $q := 1 - p$.

On en trouvera une preuve dans [App96, cor. 8.9] et dans [BL98, expl. V.5.5(i) pages 145-146]. Voici un exemple de code Matlab illustrant le TCL, dont la sortie graphique est donnée par la figure 2.2 :

```

1 % Illustration du TLC avec un échantillon de lois uniformes
2 % Fait appel à StixBox pour pnorm
3 clear;
4 i=0;
5 p=input('Nombre_de_v.a._uniformes_(par_exemple_10)_=_' );
6 n=input('Taille_de_l''echantillon_(par_exemple_50)_=_' );
7 d=input('Discrétisation_(par_exemple_100)_=_' );
8 % On simule n fois la variable Z:=(U_1+...+U_p-p/2)/(p/12)

```

```

9  X=rand(p,n);
10 Z=sum(X)-p/2; % Centrage
11 Z=Z/sqrt(p/12); % Réduction
12 dz=range(Z)/d;
13 % Fonction de Répartition Simulée
14 [ Effectifs ,Classes]=hist(Z,d);
15 FRS=cumsum(Effectifs)/n;
16 FRS(d+1)=1;
17 % Tracé de la fonction de répartition simulée
18 % et de celle de la loi normale
19 Abscisses=[min(Z)+dz/2:dz:max(Z)+dz/2];
20 figure; clf; hold on;
21 plot(Abscisses,FRS,'b+-');
22 FRN=pnorm(Abscisses);
23 plot(Abscisses,FRN,'r-');
24 legend('Fonc._de_rep._simulee','Fonc._de_rep._Normale',4);

```

Le théorème suivant est une conséquence du TCL. Il affirme que les courbes affines par morceaux

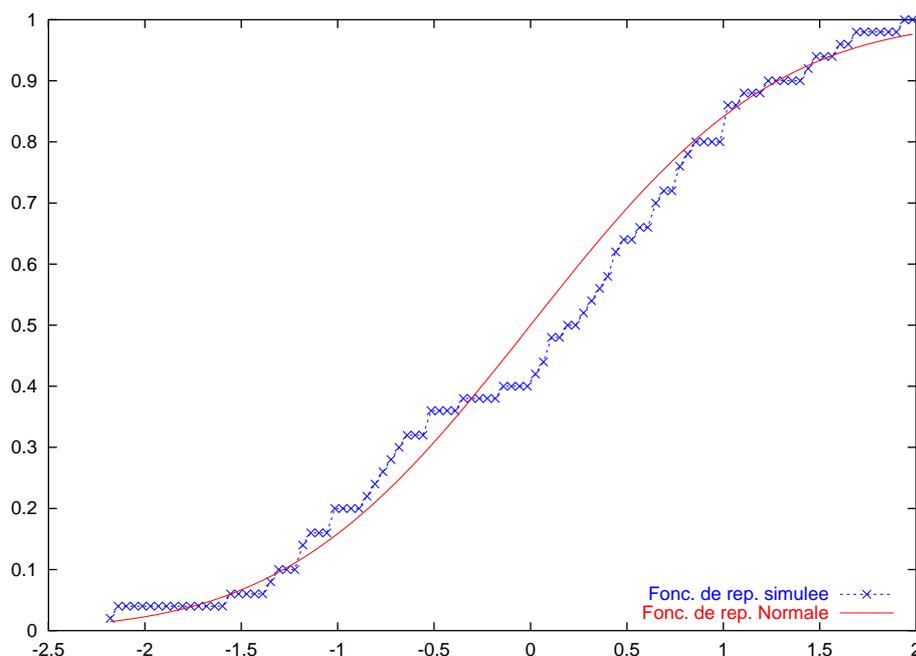


FIG. 2.2 – Illustration du TCL (lois uniformes).

construites à partir des sommes partielles renormalisées d'une suite de v.a. i.i.d., vues comme des courbes aléatoires, « convergent en loi vers la loi d'un mouvement brownien ». Il constitue donc une justification de l'intuition selon laquelle une marche aléatoire est un analogue discret du mouvement brownien :

Théorème 2.2.3 (Principe d'invariance de Donsker-Prohorov). Soit $(X_n, n \in \mathbb{N}^*)$ une suite de v.a.r. i.i.d. centrées de variance σ^2 . On définit la suite de v.a. $(Y_n, n \in \mathbb{N}^*)$ à valeurs dans $\mathcal{C}(\mathbb{R}_+, \mathbb{R})$ muni de sa tribu borélienne associée à la topologie cylindrique par :

$$Y_n(t) := \frac{X_1 + \cdots + X_{[nt]} + (nt - [nt])X_{[nt]+1}}{\sigma\sqrt{n}}.$$

où $[r]$ désigne la partie entière de r . Alors, la suite de v.a. $(Y_n, n \in \mathbb{N}^*)$ converge faiblement vers une loi de probabilité μ^* sur $\mathcal{C}(\mathbb{R}_+, \mathbb{R})$ telle que la v.a. $(B_t, t \geq 0)$ définie par

$$\begin{aligned} B_t : (\mathcal{C}(\mathbb{R}_+, \mathbb{R}), \mu^*) &\rightarrow \mathbb{R} \\ \omega &\mapsto B_t(\omega) := \omega(t) \end{aligned}$$

soit un mouvement brownien standard sous μ^* , c'est-à-dire un processus gaussien à trajectoires presque-sûrement continues, centré, à accroissements indépendants, tel que $B_0 = 0$ et $\mathbb{E}(B_t - B_s)^2 = |t - s|$. En particulier, pour tout $0 \leq t_1 < \dots < t_n$, on a :

$$(Y_n(t_1), \dots, Y_n(t_n)) \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} (B(t_1), \dots, B(t_n)).$$

Voici une illustration de ce théorème en Matlab, dont la sortie graphique est donnée par la figure 2.3 :

```

1 %% Illustration du théorème de Donsker.
2 clear;
3 % On veut simuler Y_n
4 n=1000;
5 mint=0; % Valeur maximale de t
6 maxt=1; % Valeur minimale de t
7 Dt=80; % Précision du tracé en t
8 % Approximation de Y_n
9 dt=(maxt-mint)/Dt; % Pas de la subdivision en t
10 T=[mint:dt:maxt]; % Subdivision en t de [mint,maxt]
11 X=rand(1,1+floor(n*maxt)); % Génération des v.a. X_1,...,X_n
12 X=X-mean(X); % Centrage empirique (théorique=1/2)
13 sigma=std(X); % Sigma empirique (théorique=1/12)
14 for i=1:Dt,
15     Y(i)=sum(X(1:floor(i*T(i))))+(i*T(i)-floor(i*T(i)))*X(1+floor(i*T(i)));
16 end
17 Y(Dt+1)=0;
18 Y=Y/(sigma*sqrt(n));
19 figure
20 clf;
21 hold on
22 plot(T,Y,'r-')
23 % Approximation au dixième Y_(n/10)
24 Dt=floor(Dt/10); % Précision du tracé en t
25 dt=(maxt-mint)/Dt; % Pas de la subdivision en t
26 T=[mint:dt:maxt]; % Subdivision en t
27 for i=1:Dt,
28     Z(i)=sum(X(1:floor(i*T(i))))+(i*T(i)-floor(i*T(i)))*X(1+floor(i*T(i)));
29 end
30 Z(Dt+1)=0;
31 Z=Z/(sigma*sqrt(n));
32 plot(T,Z,'b-@')
33 legend('Y(n)', 'Y(n/10)')
34 xlabel('t')
35 ylabel('Yn')
```

2.3 Vitesse de convergence dans le TCL

Considérons une suite $(X_n, n \in \mathbb{N}^*)$ de v.a. i.i.d. de carré intégrable de moyenne m et de variance σ^2 . Soit G_n la fonction de répartition de la v.a.r.

$$Y_n := \frac{X_1 + \dots + X_n - nm}{\sigma\sqrt{n}}$$

et Φ la fonction de répartition de la gaussienne standard $\mathcal{N}(0,1)$. Le théorème central limite (TCL) indique que pour tout $t \in \mathbb{R}$:

$$\lim_{n \rightarrow +\infty} |G_n(t) - \Phi(t)| = 0$$

Le théorème suivant donne une estimée uniforme sur la vitesse de cette convergence, cf. [Pet95] pour une preuve :

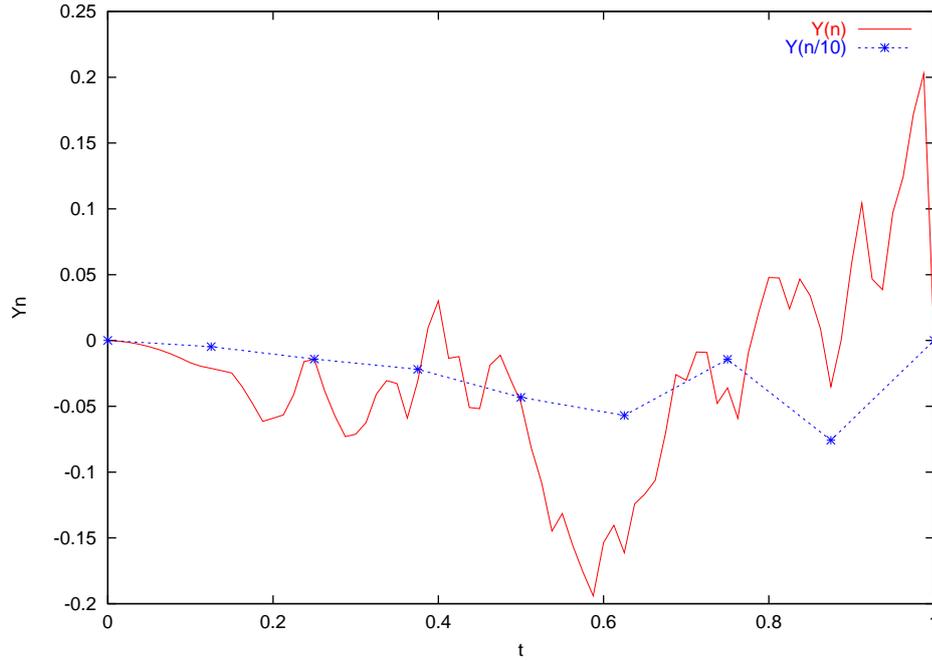


FIG. 2.3 – Illustration du théorème de Donsker.

Théorème 2.3.1 (Berry-Essen). Soit $(X_n, n \in \mathbb{N}^*)$ une suite de v.a.r. i.i.d. de variance $\sigma^2 > 0$ et possédant un moment d'ordre 3. Alors on a :

$$\sup_{t \in \mathbb{R}} |G_n(t) - \Phi(t)| \leq \frac{C\rho}{\sigma^3 \sqrt{n}},$$

où $\rho := \mathbf{E}(|X_1 - \mathbf{E}(X_1)|^3)$ et C est une constante positive qui ne dépend pas de la loi des X_n .

Il existe des versions dites « locales », du TCL, qui donnent des estimées sur les densités plutôt que sur les fonctions de répartition. Elles sont plus fortes que celles du type « Berry-Essen ». La plus élémentaire est sans doute la version forte du théorème de Moivre-Laplace 2.2.2, dont on trouvera une preuve faisant appel à la formule de Stirling dans [DCD82b, thm 2.2.4 page 35] :

Théorème 2.3.2 (Théorème de Moivre-Laplace « fort »). Soit $(S_n, n \in \mathbb{N}^*)$ des v.a. i.i.d. de loi binomiale $\mathcal{B}(n, p)$ de taille n et de paramètre $p \in]0, 1[$. On pose $q := 1 - p$. Alors, pour tout couple $(a, b) \in \mathbb{R}^2$ avec $a < b$, on a :

$$\sup_{x \in I_n \cap [a, b]} \left| \sqrt{npq} \mathbb{P} \left(\frac{S_n - np}{\sqrt{npq}} = x \right) - (2\pi)^{-1/2} e^{-x^2/2} \right| \xrightarrow{n \rightarrow +\infty} 0,$$

où I_n désigne le support de la v.a. discrète $(S_n - np)/\sqrt{npq}$.

2.4 Vitesse de convergence dans le théorème de Glivenko-Cantelli

Reprenons à présent le cadre du Théorème de Glivenko-Cantelli 2.1.4 page 40 et supposons que F est continue, et donc que μ n'a pas de masses ponctuelles. Alors, on montre que :

$$\mathcal{L} \left(\sqrt{n} \sup_{x \in \mathbb{R}} |F_n(x) - F(x)| \right) \xrightarrow[n \rightarrow +\infty]{\text{étr.}} \mathcal{L}_{\mathbf{F}}, \quad (2.2)$$

où $\mathcal{L}_{\mathbf{F}}$ est la loi de probabilité sur \mathbb{R}_+ , appelée « distribution de Kolmogorov-Smirnov », dont la fonction de répartition $\mathbf{F} : \mathbb{R} \rightarrow [0, 1]$ est donnée par :

$$\mathbf{F}(u) := \begin{cases} 0 & \text{si } u \leq 0, \\ 1 + 2 \sum_{k=1}^{+\infty} (-1)^k e^{-2k^2 u^2} & \text{si } u > 0. \end{cases} \quad (2.3)$$

La bibliothèque Stixbox fournit la fonction Matlab `pks` qui permet de la calculer. Comme nous allons le voir ultérieurement dans la section 3.3.3 page 65, cette convergence en loi permet de construire un test d'adéquation appelé *test de Kolmogorov-Smirnov*.

Si $(B_t, t \in [0, 1])$ est un mouvement brownien standard, on appelle *pont brownien sur* $[0, 1]$ le processus $(X_t, 0 \leq t \leq 1) := (B_t - tB_1, 0 \leq t \leq 1)$. Il doit son nom au fait qu'il s'interprète comme un « passage brownien » entre les deux extrémités de l'intervalle $[0, 1]$ en lesquelles il est nul. On montre que la v.a.r. $\sup_{0 \leq t \leq 1} |X_t|$ a pour loi \mathcal{L}_F . De plus, lorsque la fonction de répartition F de μ est continue, on a :

$$\sqrt{n} \sup_{x \in \mathbb{R}} (F_n(x) - F(x)) \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \sup_{0 \leq t \leq 1} X_t, \quad (2.4)$$

et que :

$$\sqrt{n} \sup_{x \in \mathbb{R}} |F_n(x) - F(x)| \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \sup_{0 \leq t \leq 1} |X_t|. \quad (2.5)$$

La convergence en loi (2.5) n'est qu'une réécriture de (2.2). En voici une illustration en Matlab, dont la sortie graphique est donnée par la figure 2.4 :

```

1  % Illustration de la vitesse de convergence dans
2  % le théorème de Glivenko-Cantelli.
3  % Tracé des fonctions de répartitions à n fixé.
4  % Fait appel à Stixbox pour rnorm, pnorm et pks
5  clf; clear;
6  n=input('Taille_des_échantillons_(par_exemple_30)_=_'?');
7  p=input('Nombre_d''échantillons_(par_exemple_30)_=_'?');
8  for i=1:p, % Parcours des échantillons
9      % On simule un échantillon de taille n de loi normale
10     X=rnorm(n);
11     % Fonction de Répartition Empirique de l'échantillon
12     [Effectifs,Classes]=hist(X,n);
13     FRE=cumsum(Effectifs)/n;
14     % Calcul de la norme uniforme de l'écart
15     % à la fonction de répartition théorique.
16     du=range(X)/n; % i.e. (max(X)-min(X))/n
17     Z(i)=sqrt(n)*max(abs(FRE-pnorm(Classes+du/2)))
18 end
19 % Tracé
20 % Fonction de Répartition Simulée
21 [Effectifs,Classes]=hist(Z,p);
22 FRS=cumsum(Effectifs)/p;
23 % Tracé de la fonction de répartition empirique
24 % et de celle du sup de la valeur absolue du pont brownien
25 % appelée aussi distribution de Kolmogorov-Smirnov.
26 figure
27 du=range(Z)/p;
28 plot(Classes+du/2,FRS,'b+-');
29 hold on;
30 Abscisses=[min(Z)+du/2:du:max(Z)+du/2];
31 plot(Abscisses,pks(Abscisses),'r-');
32 legend('Fonction_de_repartition_simulee',\
33        'Fonction_de_repartition_de_Kolmogorov-Smirnov',4);

```

2.5 Principes de Grandes Déviations

La loi des grands nombres indique que la moyenne arithmétique S_n de v.a. i.i.d. de loi μ , intégrables, à valeurs dans \mathbb{R}^n , se « concentre » autour de leur moyenne m_μ . Ainsi, on s'attend à ce que la probabilité de l'évènement $\{S_n \in A\}$ tende vers 0 lorsque A est « loin » de la moyenne m_μ . Prenons pour simplifier

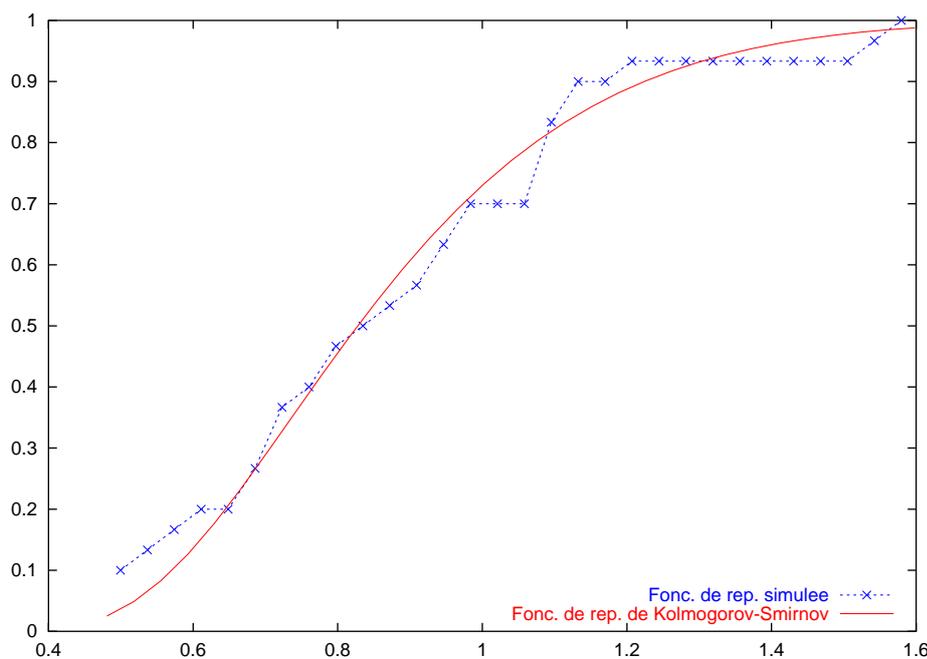


FIG. 2.4 – Convergence vers la distribution de Komogorov-Smirnov pour le théorème de Glivenko-Cantelli (loi normale).

$m_\mu = 0$, $n = 1$ et $A = [r, +\infty[$. On a pour tout $\theta \in \mathbb{R}^+$, d'après l'inégalité de Markov :

$$\begin{aligned} \mathbb{P}(S_n \in [r, +\infty[) &= \mathbb{P}(\exp(\theta(X_1 + \dots + X_n)) \geq nr\theta) \\ &\leq \exp(-nr\theta - \log \mathbf{E}(\exp(\theta(X_1 + \dots + X_n)))) \\ &= \exp(-n(r\theta - \log \mathbf{E}(\exp(\theta X_1)))). \end{aligned}$$

Cette inégalité étant valable pour tout $\theta > 0$, on s'attend donc à un comportement du type :

$$\mathbb{P}(S_n \in [r, +\infty[) \simeq \exp\left(-n \sup_{\theta} (r\theta - \log \mathbf{E}(\exp(\theta X_1)))\right).$$

Le théorème de Cramér confirme cette intuition. Il affirme que si μ possède des moments exponentiels de tous ordres, c'est-à-dire que $\mathbf{E}_\mu(\exp(\theta \cdot X_1)) < +\infty$ pour tout $\theta \in \mathbb{R}$, alors on a pour tout borélien A :

$$-\inf_{\text{int}(A)} \Lambda_\mu^* \leq \liminf_{n \rightarrow +\infty} \frac{1}{n} \log \mathbb{P}(S_n \in A) \leq \limsup_{n \rightarrow +\infty} \frac{1}{n} \log \mathbb{P}(S_n \in A) \leq -\inf_{\text{adh}(A)} \Lambda_\mu^*,$$

où Λ_μ^* désigne la transformée de Cramér² de μ , définie par :

$$\Lambda_\mu^*(x) := \sup_{\theta \in \mathbb{R}^n} \left[x \cdot \theta - \log \int e^{z \cdot \theta} d\mu(z) \right].$$

On dit alors que la suite de v.a. satisfait à *un principe de grandes déviations* par rapport à leur moyenne (PGD), de *vitesse* $1/n$ et de *fonction de taux* Λ_μ^* . On montre que la fonction de taux est positive, semi-continue inférieurement, convexe, qu'elle décroît sur $] -\infty, m_\mu]$, croît sur $[m_\mu, +\infty[$, qu'elle tend vers $+\infty$ en $\pm\infty$ et qu'elle admet un unique minimum en m_μ pour lequel elle est nulle. Quelques exemples de transformées de Cramér sont donnés dans la table 2.1.

De nombreux autres théorèmes autour des PGD existent, notamment celui de Sanov qui concerne la convergence de la mesure empirique vers la loi μ :

Théorème 2.5.1 (Théorème de Sanov). *Soit $(X_n, n \in \mathbb{N}^*)$ une suite de v.a. i.i.d. intégrables, de loi commune μ sur \mathbb{R}^n . On note $(L_n, n \in \mathbb{N}^*)$ les mesures empiriques associées et on munit l'ensemble*

²Ça n'est rien d'autre que la transformée de Legendre du logarithme de la transformée de Laplace de μ .

Bernoulli $p\delta_1 + (1-p)\delta_0$ avec $0 < p < 1$	$x \log \frac{x}{p} + (1-x) \log \frac{1-x}{1-p}$ si $0 \leq x \leq 1$ et $+\infty$ sinon
Poisson de paramètre $\lambda > 0$	$\lambda - x + x \log \frac{x}{\lambda}$ si $x > 0$ et $+\infty$ sinon
Exponentielle de paramètre $\lambda > 0$	$\lambda x - 1 - \log(\lambda x)$ si $x > 0$ et $+\infty$ sinon
Gaussienne de moyenne m et de variance $\sigma^2 > 0$	$(x-m)^2/2\sigma^2$

TAB. 2.1 – Quelques transformées de Cramér

$\mathcal{P}(\mathbb{R}^n)$ des lois de probabilité sur \mathbb{R}^n de la topologie de la convergence étroite (i.e. en loi). Alors, pour tout $A \subset \mathcal{P}(\mathbb{R}^n)$, on a :

$$-\inf_{\nu \in \text{int}(A)} \mathbf{Ent}(\nu | \mu) \leq \liminf_{n \rightarrow +\infty} \frac{1}{n} \log \mathbb{P}(L_n \in A) \leq \limsup_{n \rightarrow +\infty} \frac{1}{n} \log \mathbb{P}(L_n \in A) \leq -\inf_{\nu \in \text{adh}(A)} \mathbf{Ent}(\nu | \mu),$$

où $\mathbf{Ent}(\nu | \mu)$ désigne l'entropie relative de ν par rapport à μ , définie par :

$$\mathbf{Ent}(\nu | \mu) := \begin{cases} \mathbf{E}_\mu \left(\frac{d\nu}{d\mu} \log \frac{d\nu}{d\mu} \right) & \text{si } \nu \ll \mu, \\ +\infty & \text{sinon.} \end{cases}$$

Le symbole \ll désigne l'absolue continuité. On peut retrouver le théorème de Cramér à partir d'un théorème du « type Sanov » pour les lois de probabilité sur $\{1, \dots, n\}$. L'entropie relative se muant en transformée de Cramér.

Exemple 2.5.2. Considérons une suite $(X_n, n \in \mathbb{N}^*)$ de v.a. i.i.d. de loi de Bernoulli sur $\{0, 1\}$ de paramètre $p \in]0, 1[$. Soit r un réel positif. On note $S_n := (X_1 + \dots + X_n)/n$ leur moyenne arithmétique. La v.a. S_n est de moyenne p . Comme nS_n suit une loi binomiale $\mathcal{B}(n, p)$ de taille n et de paramètre p , on a, en notant F la fonction de répartition de $\mathcal{B}(n, p)$:

$$\log \mathbb{P}(S_n - p \geq r) = \log(1 - F(n(p+r))).$$

D'autre part, le TCL indique que la loi de $\sqrt{n}(S_n - p)/\sqrt{pq}$ est approximativement $\mathcal{N}(0, 1)$. On a donc, en notant Φ la fonction de répartition de la loi $\mathcal{N}(0, 1)$:

$$\log \mathbb{P}(S_n - p \geq r) \simeq \log \left(1 - \Phi \left(r \sqrt{\frac{n}{pq}} \right) \right) = \log \Phi \left(-r \sqrt{\frac{n}{pq}} \right).$$

Enfin, d'après le théorème de Cramér, on a par le PGD et par la formule explicite pour la transformée de Cramér pour la loi de Bernoulli donnée par la table 2.1 :

$$\log \mathbb{P}(S_n - p \geq r) \simeq -n \left((p+r) \log \frac{p+r}{p} + (1-p-r) \log \frac{1-p-r}{1-p} \right).$$

Voici un code Matlab illustrant ces trois estimées, dont la sortie graphique est donnée par la figure 2.5 :

```

1  %% Comparaison entre l'estimée de P(Sn - p >= r) par TLC et par
2  %% Principe de Grandes Déviations de Cramér lorsque Sn est la
3  %% moyenne arithmétique de v.a. de Bernoulli i.i.d.
4  %% de paramètre p. Les paramètres du programme sont p et r.
5  %% Il faut que 0<=p+r<=1
6  %% Exemple : pour (p,r)=(.975,.0125),
7  %% le PGD est meilleur que la LGN pour n >~ 3500
8  %% Fait appel à StixBox pour pnorm & pbinom
9  clf; clear
10 p=input('Paramètre_p_de_la_loi_de_Bernoulli_=_');
11 r=input('Paramètre_r_de_l''événement_=_');
12 q=1-p;
13 x=p+r;
14 N=[200:20:5000];
15 for n=N,
16     logpb(n)=log(1-pbinom(n*x,n,p))/n;
17     tcl(n)=log(pnorm(-r*sqrt(n/(p*q)),0,1))/n;
18     pgd(n)=-(x*log(x/p)+(1-x)*log((1-x)/q));

```

```

19 end
20 plot(N,logpb(N),'k',N,tcl(N),'r-',N,pgd(N),'b--');
21 legend('Log_proba','TCL','PGD',4);
22 xlabel('Taille_de_l''echatillon');
23 ylabel('Approximations_de_log(P(Sn > p+r))');

```

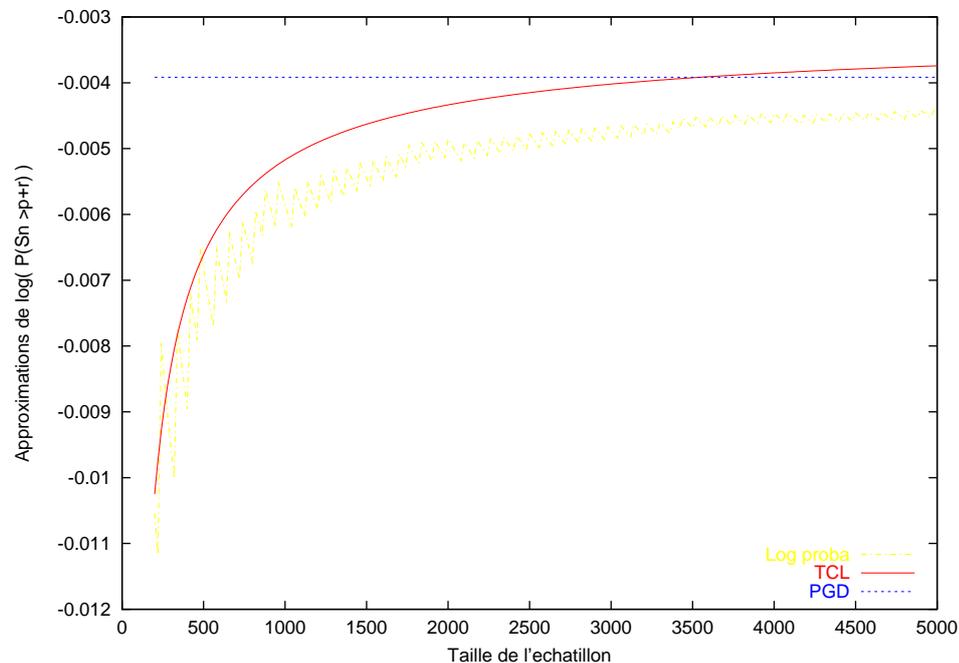


FIG. 2.5 – Comparaison des approximations par TCL et PGD pour un écart à la moyenne de la loi binomiale.

2.6 Loi du logarithme itéré

À mi-chemin entre la LGN et le TLC, la loi du logarithme itéré précise ce qui se passe lorsque l'on divise la somme partielle d'une suite de v.a.r. i.i.d. par une suite réelle de vitesse intermédiaire entre \sqrt{n} et n . En effet, si $(X_n, n \in \mathbb{N}^*)$ est une suite de v.a.r. i.i.d. de loi μ , intégrables, de moyenne m et de variance σ^2 , alors on a μ -presque sûrement :

$$\overline{\lim}_{n \rightarrow +\infty} Y_n = +\sigma \quad \text{et} \quad \underline{\lim}_{n \rightarrow +\infty} Y_n = -\sigma,$$

où

$$Y_n := \frac{X_1 + \cdots + X_n - nm}{\sqrt{2n \log \log n}}.$$

D'après Strassen, on a plus précisément :

$$\text{adh}((Y_n)_{n \in \mathbb{N}^*}) = [-\sigma, +\sigma] \quad \text{et} \quad \text{dist}(Y_n, [-\sigma, +\sigma]) \xrightarrow[n \rightarrow +\infty]{\text{p.s.}} 0.$$

De nombreuses sophistications existent, avec des suites $(a_n)_n$ quelconques. Il est sans doute vain de vouloir illustrer par une simulation la convergence des valeurs d'adhérence de $(Y_n, n \in \mathbb{N}^*)$ car la propriété est « trop asymptotique » pour être vue sur un échantillon de taille finie. En effet, ici, la suite $(Z_n, n \in \mathbb{N}^*)$ qui converge p.s. par exemple vers $+\sigma$ est donnée par la définition de $\overline{\lim}$:

$$Z_n := \inf_{n \geq 1} \sup_{k \geq n} \frac{X_1 + \cdots + X_k - km}{\sqrt{2k \log \log k}},$$

qui fait déjà intervenir un nombre infini de X_i et n'est donc pas accessible numériquement puisque les échantillons dont on dispose sont finis. Voici un petit bout de code pour s'en convaincre (à exécuter plusieurs fois...) :

```

1 %% Tentative vaine d'illustration de la loi du logarithme itéré
2 clear
3 n=100000;
4 moy=1/2; sig=sqrt(1/12);
5 N=[1:n];
6 X=rand(1,n);
7 S=cumsum(X);
8 Y=(S-N*moy)./sqrt(2*N.*log(log(N)));
9 clf; plot(N,Y,'b.',N,sig*ones(1,n),'r.-',-sig*ones(1,n),'r.-');
10 xlabel('Taille_de_l''echantillon');

```

En revanche, on peut visualiser la convergence vers 0 de la distance de la suite $(Y_n, n \in \mathbb{N}^*)$ à l'intervalle $[-\sigma, +\sigma]$. Voici un code Matlab dont la sortie graphique est donnée par la figure 2.6 :

```

1 %% Illustration de la décroissance vers zéro de la distance à
2 %% l'intervalle [-sig,+sig] dans le théorème de Strassen
3 %% (loi du logarithme itéré)
4 clear
5 n=1000;
6 n0=3;
7 moy=1/2; sig=sqrt(1/12);
8 N=[1:n];
9 S=cumsum(rand(1,n));
10 Y=(S(n0:n)-N(n0:n)*moy)./sqrt(2*N(n0:n).*log(log(N(n0:n))));
11 Z=((Y>sig).*(Y-sig)+(Y<-sig).*(-Y-sig))/sig;
12 clf;
13 plot(N(n0:n),Z,'b-.');
14 xlabel('Taille_n_de_l''echantillon');
15 ylabel('Distance(Y_n,[-sig,sig])/sig');
16 legend('Distance_dans_LLI_de_Strassen');

```

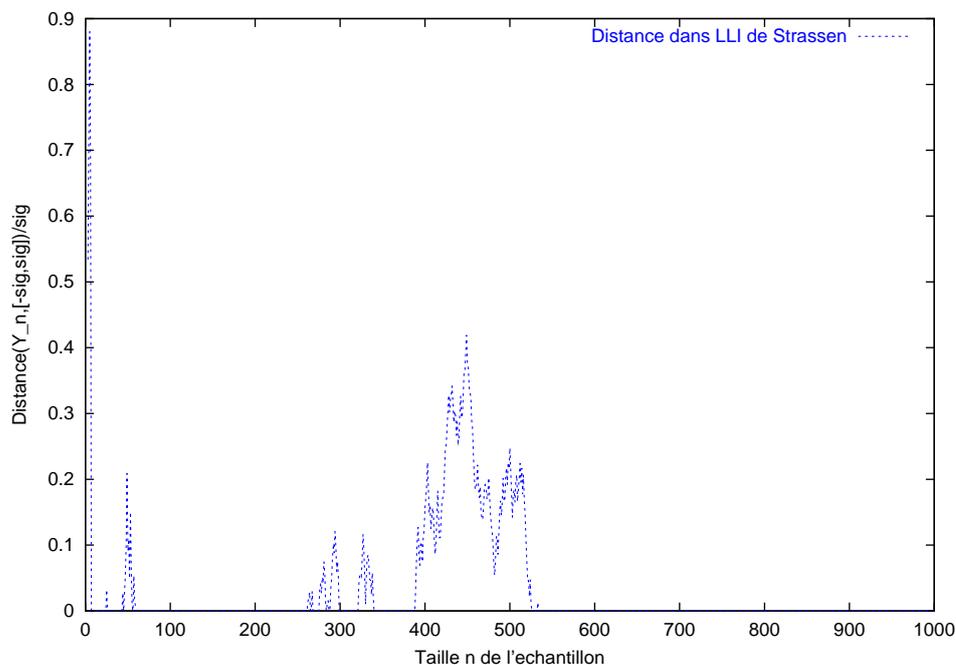


FIG. 2.6 – Illustration de la loi du log. itéré de Strassen (distance à $[-\sigma, +\sigma]$).

Remarque 2.6.1 (Sens des simulations). Le code précédent pose le problème de ce qu'apporte une simulation numérique d'un point de vue rationnel. Contrairement au calcul formel, une simulation numérique ne prouve rien. C'est plutôt la théorie qui permet de justifier son utilisation dans la pratique. En revanche, la simulation peut s'avérer très utile pour expérimenter et guider l'intuition. Enfin, elle permet parfois de mettre jour des comportements étranges ou inattendus qui font progresser la théorie en posant des questions nouvelles.

En fait, les « preuves » de théorèmes obtenues par calcul formel sur ordinateurs ne font pas l'unanimité dans la communauté des mathématiciens. Malgré les problèmes philosophiques posés, cela n'est peut-être après tout qu'une manifestation d'un certain conservatisme. Les progrès futurs de l'informatique nous promettent sans doute de nombreux débats sur le sujet. . . À la fin du dix-neuvième siècle, les preuves non constructives, fussent-elles dues à Hilbert, étaient jugées irrecevables par de nombreux mathématiciens, comme Kronecker par exemple. Les choses ont un peu changé depuis, comme vous l'avez constaté au cours de votre cursus mathématique !

2.7 Convergence des histogrammes

Simuler n réalisations indépendantes d'une loi μ sur \mathbb{R} correspond à la donnée, pour un $\omega \in \Omega$, de la suite $X_1(\omega), \dots, X_n(\omega)$ où les $(X_n, n \in \mathbb{N}^*)$ sont des variables aléatoires indépendantes de même loi μ . On note ici F la fonction de répartition de μ , L_n la mesure empirique et F_n la fonction de répartition empirique.

Si l'on considère un intervalle $I_n = [a_n, b_n]$ de \mathbb{R} dont les bornes peuvent dépendre a priori de n , le théorème de Glivenko-Cantelli (2.1) page 40 nous permet d'écrire, pour μ -presque tout ω :

$$\overline{\lim}_{n \rightarrow \infty} |L_n(I_n)(\omega) - \mu(I_n)| \leq \overline{\lim}_{n \rightarrow \infty} (|F_n(b_n) - F(b_n)| + |F_n(a_n) - F(a_n)|) = 0.$$

On rappelle que la quantité

$$L_n(I_n)(\omega) = \frac{\#\{i \leq n, X_i(\omega) \in I_n\}}{n}$$

représente la fréquence empirique de l'intervalle I_n . Partant des réalisations $X_1(\omega), \dots, X_n(\omega)$ obtenues par simulation, on subdivise l'intervalle $[\min_i X_i(\omega), \max_i X_i(\omega)]$ en m sous-intervalles disjoints $A_{i,n,m}(\omega)$, $i = 1, \dots, m$ de longueurs égales. L'histogramme en m classes associé aux valeurs simulées correspond au tracé de la fonction constante par morceaux $H_{n,m}(\omega)$ définie par

$$H_{n,m}(\omega) := \sum_{i=1}^m \#\{j, X_j(\omega) \in A_{i,n,m}(\omega)\} \mathbb{I}_{A_{i,n,m}(\omega)}.$$

Les $A_{i,n,m}(\omega)$ étant de même longueur $l_{n,m}(\omega)$, la surface de l'histogramme est donnée par la formule $n(\max_i X_i(\omega) - \min_i X_i(\omega))/m$. On définit à présent la fonction $D_{n,m}(\omega)$ comme la densité de probabilité sur \mathbb{R} obtenue en renormalisant l'histogramme

$$D_{n,m}(\omega) = \frac{m}{n(\max_i X_i(\omega) - \min_i X_i(\omega))} H_{n,m}(\omega).$$

Les $A_{i,n,m}(\omega)$ étant de même longueur, on a aussi

$$D_{n,m}(\omega) := \frac{1}{n} \sum_{i=1}^m \frac{\#\{j, X_j(\omega) \in A_{i,n,m}(\omega)\}}{l_{n,m}(\omega)} \mathbb{I}_{A_{i,n,m}(\omega)},$$

et $D_{n,m}(\omega)$ apparaît alors comme une combinaison linéaire convexe de densités uniformes sur les $A_{i,n,m}(\omega)$. On voit bien que la mesure de probabilité de densité $D_{n,m}(\omega)$ et la mesure empirique $L_n(\omega)$ affectent la même probabilité aux $A_{i,n,m}(\omega)$, et l'on peut montrer que pour tout n , la mesure de probabilité de densité $D_{n,m}(\omega)$ converge étroitement vers la mesure empirique $L_n(\omega)$.

Il existe donc une suite $m_n(\omega)$ telle que la mesure de probabilité de densité $D_{n,m_n(\omega)}(\omega)$ converge étroitement vers μ quand $n \rightarrow \infty$.

2.7.1 Estimées uniformes sur les densités

Nous avons vu que l'on avait convergence faible. Il est plus délicat d'obtenir des estimées uniformes sur la proximité des densités de μ et de $D_{n,m}$. Les $A_{i,n,m}(\omega)$ sont tous de même longueur $l_{n,m}(\omega)$, et l'on peut donc écrire, en omettant la dépendance à ω dans les notations

$$D_{n,m} = \frac{1}{l_{n,m}} \sum_{i=1}^m L_n(A_{i,n,m}) 1_{A_{i,n,m}}.$$

Si l'on définit à présent la densité approchée en m escaliers de μ par

$$f_m := \frac{1}{l_{n,m}} \sum_{i=1}^m \mu(A_{i,n,m}) 1_{A_{i,n,m}},$$

on peut écrire

$$|D_{n,m} - f_m| \leq \frac{2 \sup_{t \in I} |F_n(t) - F(t)|}{l_{n,m}} \sum_{i=1}^m 1_{A_{i,n,m}} \leq \frac{2 \sup_{t \in I} |F_n(t) - F(t)|}{l_{n,m}}.$$

Lorsque μ a pour support un intervalle compact I , $m l_{n,m} = \max_{1 \leq i \leq n} X_i - \min_{1 \leq i \leq n} X_i$ tend vers $|I|$ presque sûrement quand n tend vers l'infini (pourquoi?). Malheureusement, le théorème de Glivenko-Cantelli ne donne pas la vitesse de convergence de $\sup_{t \in I} |F_n(t) - F(t)|$ vers 0. Pour aller plus loin, il faudrait faire appel à un théorème plus fort du type Berry-Essen par exemple (hors programme), qui donne une vitesse de l'ordre de \sqrt{n} , ce qui conduit à prendre $m = m_n = \sqrt{n}$ pour obtenir la convergence uniforme de $|D_{n,m} - f_m|$ vers 0.

Notons que si μ admet une densité f sur I de classe \mathcal{C}^2 , alors, le théorème de Heine entraîne que f_m converge uniformément vers f lorsque m tendent vers l'infini (méthodes des rectangles!).

2.7.2 Un exemple avec une illustration du théorème central limite

Soient $(U_n, n \in \mathbb{N}^*)$ une suite de v.a. i.i.d. suivant une loi uniforme sur $[0, 1]$. On rappelle que la loi uniforme sur $[0, 1]$ a pour moyenne $1/2$ et variance $1/12$. En vertu du théorème central limite, la variable aléatoire

$$\frac{U_1 + \dots + U_p - p/2}{\sqrt{p/12}} \quad (2.6)$$

converge en loi vers la loi normale centrée réduite lorsque $p \rightarrow \infty$. Pour $p = 12$, la formule précédente a une expression très simple $(U_1 + \dots + U_{12} - 6)$ et fournit déjà une « bonne » approximation de la loi normale.

On effectue alors n simulations indépendantes $X_1(\omega), \dots, X_n(\omega)$ de la v.a. (2.6) et l'on trace sur un même graphique l'histogramme H_{n,m_n} et la densité de la loi normale centrée réduite correctement dilatée pour être comparée à l'histogramme. Pour obtenir un joli graphique, on choisit m_n plus petit que n , par exemple $m_n = \sqrt{n}$. Ce facteur \sqrt{n} ne représente pas la normalisation du Théorème Central Limite mais plutôt la vitesse de convergence dans le théorème de Glivenko-Cantelli.

```

1 % Illustration du TLC avec un échantillon de lois uniformes
2 % Fait appel à StixBox pour dnorm & histo
3 clf % Nettoyage des graphiques
4 clear % Nettoyage des variables
5 i=0;
6 p=input('Nombre_de_v.a._uniformes_(par_exple_12)_=_' );
7 n=input('Taille_de_l''echantillon_(par_exple_1000)_=_' );
8 % On simule n fois la variable Z:=U_1+...+U_p,
9 X=rand(p,n);
10 Z=sum(X)-p/2; % Centrage
11 Z=Z/sqrt(p/12); % Réduction
12 % Z est proche de la N(0,1) d'après le TCL
13 % Nombre de classes pour histogramme
14 nc=sqrt(n);
15 % On trace l'histogramme normalisé en nc classes
16 % E contient les effectifs pour nos classes

```

```

17 % C contient les extrémités des classes
18 [E,C]=histo(Z,nc,0,1);
19 hold on;
20 % On trace la densité normale
21 plot(C,dnorm(C),'r-');
22 legend('Histogramme','Loi normale');

```

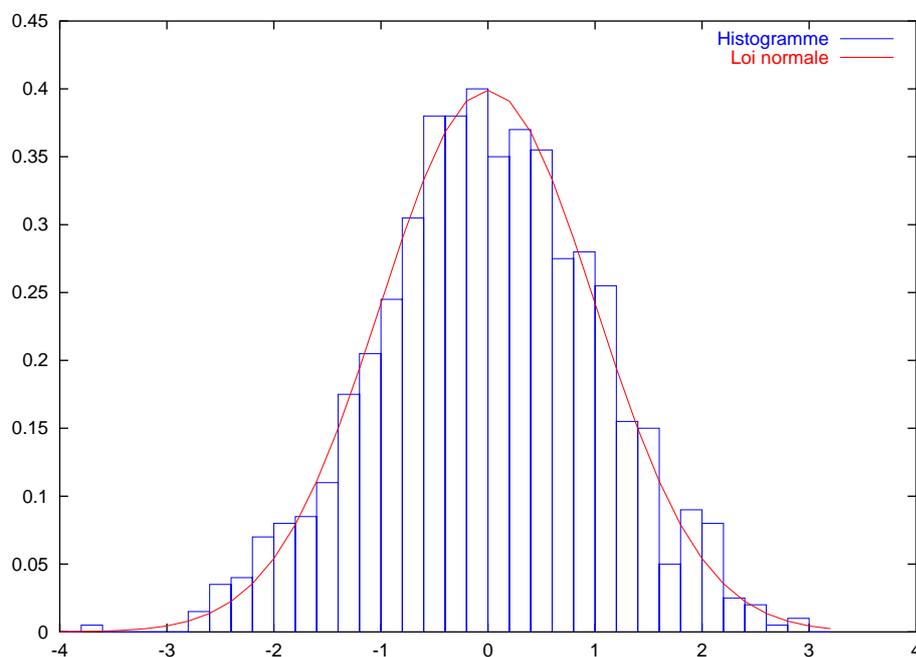


FIG. 2.7 – Approximation de la loi normale par TCL (lois uniformes).

2.8 Le théorème de Wigner

Le théorème de Wigner, qui ressemble à la loi des grands nombres, est ignoré par la plupart des cours de probabilités. Il est pourtant très simple à énoncer et sa démonstration n'est pas si difficile et fait appel à un peu de combinatoire. Il en existe de nombreuses versions avec des hypothèses plus faibles que celles requises ici. Considérons un tableau symétrique de taille infinie dont les entrées sont des v.a.r. :

$$\begin{pmatrix} w_{1,1} & w_{1,2} & \cdots \\ w_{2,1} & w_{2,2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

On note W_n la matrice aléatoire carrée symétrique réelle de taille n obtenue en tronquant le tableau à partir de son coin supérieur gauche. Les valeurs propres de W_n sont réelles d'après le théorème spectral.

Théorème 2.8.1 (Wigner). *Supposons que les v.a. $(w_{i,j}, 1 \leq i < j)$ sont i.i.d. de variance finie σ^2 et que les v.a. $(w_{i,i}, 1 \leq i)$ sont i.i.d. Si $(\lambda_{n,1}(\omega), \dots, \lambda_{n,n}(\omega))$ désigne le spectre de $n^{-1/2} W_n(\omega)$, alors, pour presque tout ω , la mesure empirique*

$$\frac{1}{n} \sum_{i=1}^n \delta_{\lambda_{i,n}(\omega)}$$

converge étroitement (i.e. en loi) vers la loi du demi-cercle de paramètre σ , de densité par rapport à la mesure de Lebesgue dx sur \mathbb{R} :

$$\frac{1}{2\pi\sigma^2} \sqrt{4\sigma^2 - x^2} \mathbf{1}_{\{-2\sigma \leq x \leq 2\sigma\}}(x).$$

Remarquons que les v.a. $\lambda_{n,i}$ ne sont pas i.i.d. et on ne peut donc pas appliquer la loi des grands nombres. Voici un code Matlab pour illustrer ce théorème, dont la sortie graphique est donnée par la figure 2.8 :

```

1 %% Illustration du théorème de Wigner
2 clf;
3 clear;
4 dim=100; % Dimension des matrices
5 nreal=100; % Nombre de réalisations i.i.d. de ces matrices
6 spectres=zeros(dim*nreal,1);
7 for i=[0:nreal-1],
8     M=randn(dim,dim);
9     spectres(1+i*dim:i*dim+dim)=eig(M-triu(M,1)+tril(M,-1)')/sqrt(dim);
10 end
11 nc=sqrt(nreal*dim);
12 % On utilise l'histogramme normalisé en nc classes
13 % E contient les effectifs pour nos classes
14 % C contient les extrémités des classes
15 [E,C]=histo(spectres,nc,0,1);
16 hold on;
17 % On trace la densité de la loi semi-circulaire
18 plot(C,sqrt(4-C.^2)/(2*pi),'r-');
19 legend('Histogramme','Loi_semi-circulaire');

```

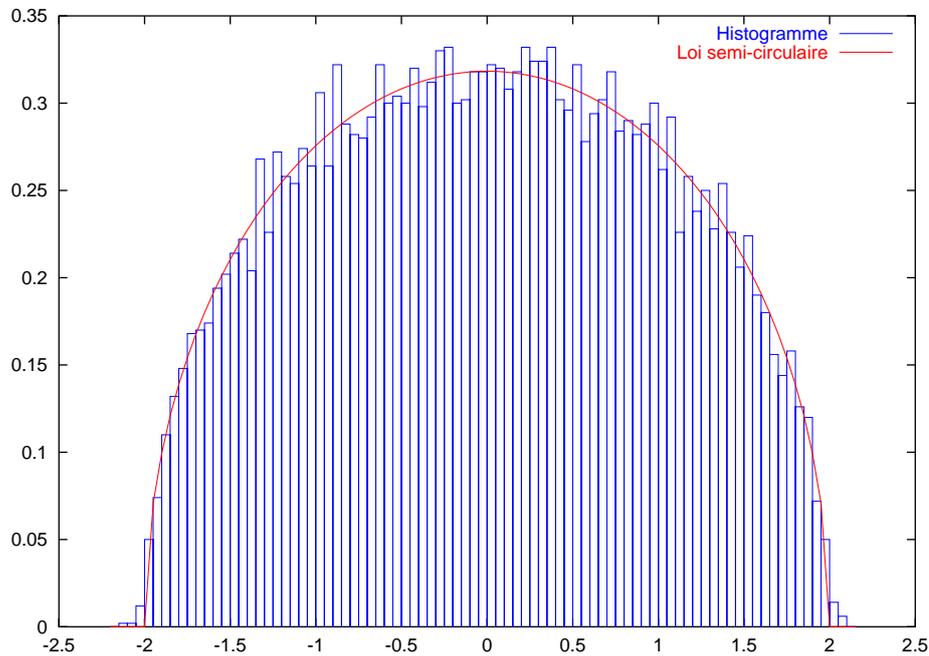


FIG. 2.8 – Illustration du théorème de Wigner.

Fragment 3

Intervalles de confiance & tests statistiques

La statistique paramétrique consiste à considérer qu'un phénomène aléatoire obéit à une loi de probabilité qui appartient à une famille de loi indexée par un paramètre. Le choix de la famille constitue la « modélisation ». La véritable valeur du paramètre est inconnue est l'on cherche à la déterminer à partir d'un « échantillon », c'est à dire d'une suite finie de réalisations i.i.d. qui suivent la loi inconnue cherchée. Les intervalles ou parallélépipèdes de confiance consistent alors en des « morceaux » de l'ensemble des paramètres qui contiennent la « véritable valeur » avec au moins une certaine probabilité p , appelée niveau de confiance. On parle de « risque » pour désigner la probabilité $1 - p$. Pour estimer le paramètre, on construit une fonction de l'échantillon, appelée « estimateur », qui converge, en un certain sens, vers la valeur du paramètre.

3.1 Rudiments de statistique paramétrique

Un modèle paramétrique de paramètre $\theta \in \Theta \subset \mathbb{R}^k$ consiste en la donnée d'un triplet $(\mathcal{E}, \mathcal{A}, (\mathbb{P}_\theta)_{\theta \in \Theta})$. En d'autres termes, on se donne une famille de lois de probabilité $\{\mathbb{P}_\theta, \theta \in \Theta\}$ sur l'espace mesurable $(\mathcal{E}, \mathcal{A})$ indexée par un ouvert Θ de \mathbb{R}^k . L'exemple le plus classique est donné par $\mathcal{E} = \mathbb{R}$ et

$$(\mathbb{P}_\theta)_{\theta \in \Theta} := \{\mathcal{N}(m, \sigma^2), (m, \sigma^2) \in \mathbb{R} \times \mathbb{R}^+\}.$$

En voici un autre, encore plus simple : $\mathcal{E} = \{0, 1\}$ et

$$(\mathbb{P}_\theta)_{\theta \in \Theta} := \{\mathcal{B}ern(\{0, 1\}, p), p \in [0, 1]\}.$$

On parle de statistique bayésienne lorsque l'on muni l'ensemble Θ des paramètres d'une loi de probabilités, sensée représenter une *information a priori* sur la valeur du paramètre.

Idéalement, on dispose de données x_1, \dots, x_n , qui constituent un *échantillon*, et que l'on suppose être des *réalisations* $X_1(\omega), \dots, X_n(\omega)$ de v.a. i.i.d. X_1, \dots, X_n de *loi inconnue* \mathbb{P}_{θ_0} qui appartient à la famille $\{\mathbb{P}_\theta, \theta \in \Theta\}$. On parle parfois d'*observations* pour désigner les données. Chaque $\omega \in \Omega$ correspond à une réalisation de la suite de v.a. i.i.d. $(X_i)_{1 \leq i \leq n}$, et donc à un jeu de données possible. Le terme « échantillon » de loi \mathbb{P}_{θ_0} est utilisé pour désigner aussi bien la suite finie de v.a. i.i.d. (X_1, \dots, X_n) que l'une de ses réalisations x_1, \dots, x_n .

L'un des objectifs de la statistique paramétrique est de déterminer au mieux le paramètre θ de la loi inconnue en utilisant les données. On construit pour ce faire une v.a. $Y_n := f_n(X_1, \dots, X_n)$, fonction des X_i , qui *approche le paramètre* recherché lorsque la taille n de l'échantillon tend vers l'infini. On s'intéresse en particulier à la *nature* de cette *convergence* et à sa *vitesse*. La fonction

$$f_n : (x_1, \dots, x_n) \in \mathbb{E}^n \mapsto f_n(x_1, \dots, x_n) \in \mathbb{E}$$

est appelée *estimateur* du paramètre θ . Le *biais* d'un estimateur est par définition l'écart $\mathbf{E}_{\mathbb{P}_{\theta_0}}(f_n) - \theta := \mathbf{E}(Y_n) - \theta$ entre sa moyenne et la véritable valeur du paramètre. On dit que l'estimateur est non biaisé lorsque son biais est nul. Lorsque $\Theta = \mathbb{R}$, on dit qu'un intervalle $I \subset \Theta$ qui dépend de Y_n est un *intervalle de confiance* de *niveau de confiance* $p \in [0, 1]$ lorsque la probabilité que I contienne θ est p .

Par exemple, pour le modèle $\mathcal{E} = \mathbb{R}$ et $(\mathbb{P}_\theta)_{\theta \in \Theta} := \{\mathcal{N}(m, 1), m \in \mathbb{R}\}$, on cherche à estimer la moyenne m . L'estimateur le plus naturel est constitué par la *moyenne empirique* :

$$Y := f_n(X_1, \dots, X_n) := \frac{X_1 + \dots + X_n}{n}.$$

La loi des grands nombres indique que *presque-sûrement* pour $\mathcal{N}(m, 1)$, l'estimateur Y_n converge vers θ lorsque n tend vers $+\infty$. Ainsi, pour presque-tout jeu de données (x_1, \dots, x_n) , la moyenne arithmétique $(x_1 + \dots + x_n)/n$ converge vers m lorsque la taille n de l'échantillon tend vers $+\infty$. Par ailleurs, Y_n est la moyenne arithmétique de n v.a. i.i.d. de loi $\mathcal{N}(m, 1)$, et donc a pour loi $\mathcal{N}(m, 1)$. Ainsi, pour tout $a > 0$,

$$\mathbb{P}(Y_n \in [m - a, m + a]) := p_a := (2\pi)^{-1/2} \int_{-a}^{+a} e^{-u^2/2} du.$$

Or on a

$$\{Y_n \in [m - a, m + a]\} = \{m \in [Y_n - a, Y_n + a]\}.$$

Ainsi, l'intervalle $[Y_n - a, Y_n + a]$ contient la moyenne m cherchée avec une probabilité p_a . On parle alors d'*intervalle de confiance* de *niveau de confiance* p_a . Dans des modèles non gaussiens, on ne connaît pas forcément la loi de l'estimateur, mais l'inégalité de Chebychev donne :

$$\mathbb{P}(|Y_n - m| \geq a) \leq \frac{\sigma^2}{na^2},$$

où σ^2 désigne la variance de la loi des X_i . Cette inégalité n'est rien d'autre que la loi faible des grands nombres, qui indique que Y_n converge vers m en probabilité. Les principes de grandes déviations introduits dans la section 2.5 page 46 indiquent que la probabilité que $|Y_n - m|$ dépasse le seuil $a > 0$ décroît exponentiellement vite. Dans la pratique, σ^2 est inconnu et l'on est parfois amené à le remplacer par la variance empirique.

De façon plus générale, lorsque le paramètre à estimer n'est pas la moyenne, on est parfois amené à utiliser des théorèmes de convergence en loi pour l'estimateur Y_n et des résultats de rapidité de convergence en loi pour déterminer une valeur acceptable de n pour laquelle la loi de l'estimateur est identifiée à la loi limite du théorème de convergence en loi.

3.1.1 L'exemple des sondages binaires : une approche naïve

On modélise l'intention de vote des électeurs à un référendum par 0 pour *non* et 1 pour *oui*. On cherche à estimer la probabilité $p \in [0, 1]$ qu'un électeur vote *oui*. Pour ce faire, on interroge n personnes choisies au hasard, dont les réponses x_1, \dots, x_n sont considérées comme des réalisations de v.a. X_1, \dots, X_n i.i.d. de même loi de Bernoulli $(1 - p)\delta_0 + p\delta_1$. On propose donc le modèle paramétrique $\mathcal{E} = \{0, 1\}$, $P_\theta = (1 - \theta)\delta_0 + \theta\delta_1$ avec $\theta \in \Theta = [0, 1]$. On désire donc estimer la moyenne p des X_i . Pour ce faire, on utilise l'estimateur « moyenne empirique » :

$$Y_n := \frac{X_1 + \dots + X_n}{n}.$$

D'après la LGN, Y_n converge p.s. vers la valeur cherchée p , et on peut donc considérer que $(x_1 + \dots + x_n)/n$ converge vers p . D'après l'inégalité de Chebychev,

$$\mathbb{P}(|Y_n - p| \geq a) \leq \frac{p(1-p)}{na^2}.$$

Comme $p \in [0, 1] \mapsto p(1-p)$ a pour maximum $1/4$, on a

$$\mathbb{P}(|Y_n - p| \geq a) \leq \frac{1}{4na^2},$$

et de façon équivalente

$$\mathbb{P}([Y_n - a, Y_n + a] \text{ contient } p) > 1 - \frac{1}{4na^2}.$$

On a donc un intervalle de confiance $[Y_n - a, Y_n + a]$ pour p de niveau de confiance au moins égal à $1 - (4na^2)^{-1}$. Autrement dit, $[Y_n - ((1 - \alpha)/(4n))^{-1/2}, Y_n + ((1 - \alpha)/(4n))^{-1/2}]$ est un intervalle de confiance pour p de niveau de confiance au moins égal à α .

Alternativement, la loi de l'estimateur Y_n n'est rien d'autre qu'une loi binomiale, et le théorème de Moivre-Laplace 2.2.2 page 42 indique que $Z_n := (X_1 + \dots + X_n - np)/\sqrt{np(1-p)}$ converge en loi vers $\mathcal{N}(0,1)$ lorsque n tend vers $+\infty$. Lorsque n est grand, il est préférable d'utiliser cette approximation gaussienne de la loi de Y_n , plus précise que celle obtenue par l'inégalité de Chebychev, et dont la vitesse en n est fournie par le théorème de Moivre-Laplace « fort » 2.3.2 page 45. Comme

$$Y_n = \frac{\sqrt{p(1-p)}}{\sqrt{n}} \left(\frac{X_1 + \dots + X_n - np}{\sqrt{np(1-p)}} \right) + p = \frac{\sqrt{p(1-p)}}{\sqrt{n}} Z_n + p,$$

on peut considérer que pour n « assez grand », la loi de Z_n est bien approximée par la loi $\mathcal{N}(0,1)$. On écrit alors pour tout $a > 0$:

$$\begin{aligned} \mathbb{P}(|Y_n - p| \leq a) &= \mathbb{P}\left(|Z_n| \leq a \frac{\sqrt{n}}{\sqrt{p(1-p)}}\right) \\ &\geq \mathbb{P}\left(|Z_n| \leq a\sqrt{4n}\right) \\ &\simeq \Phi(a\sqrt{4n}) - \Phi(-a\sqrt{4n}) = 2\Phi(a\sqrt{4n}) - 1, \end{aligned}$$

où Φ est la fonction de répartition de la loi gaussienne standard $\mathcal{N}(0,1)$. Ainsi, on obtient l'intervalle de confiance

$$\left[Y_n - (4n)^{-1/2} \Phi^{-1}((1+\alpha)/2), Y_n + (4n)^{-1/2} \Phi^{-1}((1+\alpha)/2) \right]$$

de niveau de confiance voisin de α .

Voici un petit code Matlab comparant les bornes obtenues par l'inégalité de Chebychev et par l'utilisation de la loi gaussienne. On voit bien sur la figure 3.1 que l'estimation gaussienne est meilleure lorsque le niveau de confiance est proche de 1, à condition bien sûr que l'approximation gaussienne soit justifiée (i.e. n grand). A contrario, la borne obtenue par l'inégalité de Chebychev est toujours valide.

```

1 clear; clf; hold on;
2 alpha = [.05:.05:.95];
3 gauss=qnorm((1+alpha)/2);
4 chebychev=(1-alpha).^(-1/2);
5 xlabel('Niveau_de_confiance');
6 ylabel('Borne');
7 plot(alpha,gauss,'r-');
8 plot(alpha,chebychev,'b-+');
9 title('Comparaison_des_bornes_des_intervalles_de_confiance');
10 legend('Via_Gaussienne','Via_Chebychev');
```

Remarque 3.1.1. En France, les instituts de sondages font souvent appel à des techniques de « fourchettes », qui permettent de construire un échantillon « représentatif » de la population par quotas en fonction de critères sociaux-culturels par exemple. Cette approche, sensée améliorer les intervalles de confiance, est discutable dans la mesure où elle fait intervenir une information a priori dont on mesure mal l'impact.

3.2 Divers procédés d'obtention d'intervalles de confiances

Pour un n -échantillon (X_1, \dots, X_n) constitué de v.a.r. i.i.d. de loi μ d'espérance m et de variance σ^2 , on note

$$\bar{X}_n := \frac{S_n}{n} := \frac{X_1 + \dots + X_n}{n} \quad \text{et} \quad \widehat{\sigma}_n^2 := \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2.$$

Ces deux valeurs \bar{X}_n et $\widehat{\sigma}_n$ peuvent être obtenues avec les fonctions `mean` et `std` de Matlab.

3.2.1 Intervalles sur la moyenne quand la variance est connue

On suppose que la variance σ^2 est connue. Alors:

$$\sqrt{n} \frac{\bar{X}_n - m}{\sigma} \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0,1).$$

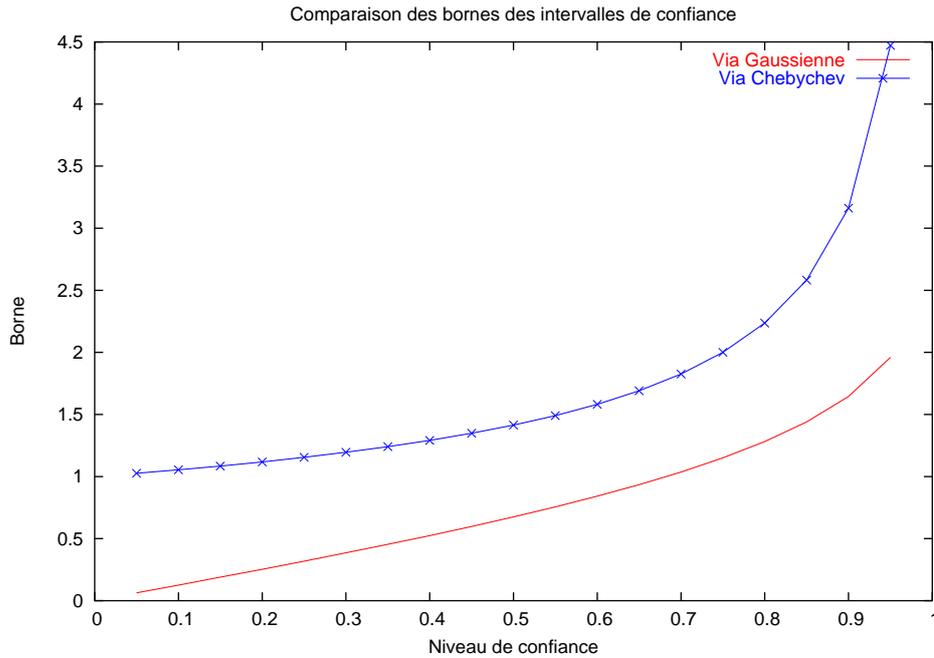


FIG. 3.1 – Comparaison des bornes des intervalles de confiance lors de l'estimation de la moyenne d'une loi de Bernoulli par la moyenne empirique.

Ce qui donne, lorsque l'on a une réalisation de (X_1, \dots, X_n) , l'intervalle de confiance:

$$I_\alpha = \left[\bar{X}_n - \frac{\sigma q_{(1-\alpha/2)}}{\sqrt{n}}, \bar{X}_n + \frac{\sigma q_{(1-\alpha/2)}}{\sqrt{n}} \right], \quad \text{pour un niveau de confiance voisin de } 1 - \alpha,$$

où q_a représente le *quantile* d'ordre a pour la loi $\mathcal{N}(0, 1)$. En pratique, on utilise ce résultat dès que $n \geq 30$, ce qui est justifié par des résultats de rapidité de convergence que nous ne donnons pas ici.

Cas particulier des lois gaussiennes

Si la loi μ de l'échantillon est gaussienne, alors on a

$$\mathcal{L}\left(\sqrt{n} \frac{\bar{X}_n - m}{\sigma}\right) = \mathcal{N}(0, 1),$$

et l'intervalle de confiance est le même que précédemment, sans limitations sur la valeur de n .

3.2.2 Intervalles sur la moyenne quand la variance est inconnue

On suppose que la variance σ^2 est inconnue. Alors on a :

$$\sqrt{n} \frac{\bar{X}_n - m}{\widehat{\sigma}_n} \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, 1),$$

ce qui donne l'intervalle de confiance

$$I_\alpha = \left[\bar{X}_n - \frac{\widehat{\sigma}_n q_{(1-\alpha/2)}}{\sqrt{n}}, \bar{X}_n + \frac{\widehat{\sigma}_n q_{(1-\alpha/2)}}{\sqrt{n}} \right], \quad \text{pour un niveau de confiance voisin de } 1 - \alpha,$$

où q_a représente le quantile d'ordre a pour la loi $\mathcal{N}(0, 1)$. Remarquons que ce résultat est vrai avec toute autre estimation convergente de σ . On l'utilise en pratique dès que $n \geq 30$.

Loi de Student

Si l'on suppose que la loi μ de l'échantillon est gaussienne, le résultat précédent reste vrai mais peut être amélioré pour les petites valeurs de n en utilisant le fait que pour tout $n \geq 2$, on a :

$$\mathcal{L}\left(\sqrt{n} \frac{\bar{X}_n - m}{\widehat{\sigma}_n}\right) = t(n-1),$$

où t est la loi de Student à $(n-1)$ degrés de liberté. Ceci donne l'intervalle de confiance suivant pour un niveau de confiance voisin de $1-\alpha$:

$$I_\alpha = \left[\bar{X}_n - \frac{\widehat{\sigma}_n t_{(1-\alpha/2)}(n-1)}{\sqrt{n}}, \bar{X}_n + \frac{\widehat{\sigma}_n t_{(1-\alpha/2)}(n-1)}{\sqrt{n}} \right],$$

où $t_a(n)$ représente le quantile d'ordre a pour la loi $t(n)$.

Loi binomiale

Si la loi μ de l'échantillon est une loi de Bernoulli de paramètre p , alors on a :

$$\sqrt{n} \frac{(\bar{X}_n - p)}{\sqrt{n \bar{X}_n (1 - \bar{X}_n)}} \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, 1).$$

Ce résultat est utilisé en pratique pour $np(1-p) \geq 20$.

3.2.3 Intervalles sur la variance

Moyenne connue

Si la loi μ de l'échantillon est *gaussienne* de moyenne m connue, alors on a pour tout $n \geq 2$:

$$\mathcal{L}\left(\frac{1}{\sigma^2} \sum_{i=1}^n (X_i - m)^2\right) = \chi^2(n).$$

On obtient donc l'intervalle de confiance suivant, sur la variance, pour le niveau de confiance $1-\alpha$:

$$I_\alpha = \left[\frac{1}{c_{(1-\alpha/2)}(n)} \sum_{i=1}^n (X_i - m)^2, \frac{1}{c_{\alpha/2}(n)} \sum_{i=1}^n (X_i - m)^2 \right],$$

où $c_a(n)$ représente le quantile d'ordre a pour la loi $\chi^2(n)$.

Moyenne inconnue

Si la loi μ de l'échantillon est *gaussienne*, alors pour tout $n \geq 2$, on a :

$$\mathcal{L}\left(\frac{(n-1)}{\sigma^2} \widehat{\sigma}_n^2\right) = \chi^2(n-1),$$

On obtient donc l'intervalle de confiance sur la variance suivant :

$$I_\alpha = \left[\frac{(n-1)}{c_{(1-\alpha/2)}(n)} \widehat{\sigma}_n^2, \frac{(n-1)}{c_{\alpha/2}(n)} \widehat{\sigma}_n^2 \right], \quad \text{pour le niveau de confiance } 1-\alpha,$$

où $c_a(n)$ représente le quantile d'ordre a de la loi $\chi^2(n)$.

3.2.4 Intervalles sur la différence de deux moyennes

On suppose que (X_1, \dots, X_n) et (Y_1, \dots, Y_m) sont deux échantillons indépendants de *loi gaussienne de même variance mais pas forcément de même moyenne*. On note $d := m_X - m_Y$ la différence entre les moyennes de X et de Y , alors pour tout $n \geq 2$ et $m \geq 2$ on a :

$$\mathcal{L}\left(\sqrt{\frac{nm(n+m-2)}{n+m}} \frac{(\bar{X}_n - \bar{Y}_m) - d}{\sqrt{(n-1)\widehat{\sigma}_n(X)^2 + (m-1)\widehat{\sigma}_n(Y)^2}}\right) = t(n+m-2),$$

où $t(n+m-2)$ est la loi de Student à $n+m-2$ degrés de liberté. On en déduit l'intervalle de confiance $I = [a_\alpha, b_\alpha]$ pour d avec :

$$a_\alpha := (\bar{X}_n - \bar{Y}_m) - t_{(1-\alpha/2)}(n+m-2) \sqrt{\frac{(n+m)((n-1)\widehat{\sigma}_n(X)^2 + (m-1)\widehat{\sigma}_n(Y)^2)}{nm(n+m-2)}}$$

et

$$b_\alpha := (\bar{X}_n - \bar{Y}_m) + t_{(1-\alpha/2)}(n+m-2) \sqrt{\frac{(n+m)((n-1)\widehat{\sigma}_n(X)^2 + (m-1)\widehat{\sigma}_n(Y)^2)}{nm(n+m-2)}}$$

et où $t_a(n)$ représente le le quantile d'ordre a de la loi $t(n)$.

3.2.5 Méthode de Monte-Carlo

La méthode de Monte-Carlo (cf. par exemple [DCD82b, p. 103]) permet le calcul de valeurs approchées d'intégrales multiples en utilisant des réalisations i.i.d. de loi uniforme. Si $(X_n)_{n \geq 1}$ est une suite de v.a. i.i.d. de loi uniforme sur $[0, 1]^m$ et si $f : [0, 1]^m \rightarrow \mathbb{R}$ est une fonction mesurable, alors la loi des grands nombre (cf. section 2.1 page 39) appliquée à la suite de v.a.r. i.i.d. $(f(X_n))_{n \geq 1}$ entraîne la convergence presque-sûre suivante :

$$\frac{1}{n} (f(X_1) + \dots + f(X_n)) \xrightarrow{n \rightarrow +\infty} \mathbf{E}(f(X_1)) = \int_{[0,1]^m} f(x) dx,$$

Si la variance de $f(X_1)$ est majorable, on est en mesure d'utiliser les intervalles de confiance pour la moyenne lorsque la variance est connue. On voit alors que la convergence à lieu en \sqrt{n} . Cette vitesse est assez lente, comparée aux méthodes déterministes, mais on observera qu'aucune régularité sur f n'est requise ici. La méthode de Monte-Carlo reste valable lorsque f est à support quelconque, en prenant par exemple des X_i de loi gaussienne.

Exemple 3.2.1. On se propose d'évaluer la valeur de π à partir des trois intégrales suivantes, en utilisant la méthode de Monte-Carlo :

1. $\int_0^1 4\sqrt{1-x^2} dx$ (aire du disque unité)
2. $\int_{[-1,+1]^2} 1_{\{x^2+y^2 \leq 1\}} dx dy$ (aire du disque unité)
3. $\int_{[-1,+1]^3} 1_{\{x^2+y^2+z^2 \leq 1\}} dx dy dz$ (volume de la boule unité).

```

1 clear; W=1;
2 while W==1
3   n=input('Taille des échantillons = ');
4   X1=4*(ones(1,n)-rand(1,n).^2).^(1/2);
5   X2=2*rand(1,n)-1;
6   Y2=2*rand(1,n)-1;
7   Z2=4*(X2.^2+Y2.^2<=1);
8   X3=2*rand(1,n)-1;
9   Y3=2*rand(1,n)-1;
10  Z3=2*rand(1,n)-1;
11  T3=3*2*(X3.^2+Y3.^2+Z3.^2<=1);
12  disp('La valeur de Pi est ');
13  [mean(X1),mean(Z2),mean(T3)]
14  W=input('Une autre simulation? (0=non, 1=oui) ');
15 end

```

Testez le avec $n = 10^4, 10^5, 10^6$.

```

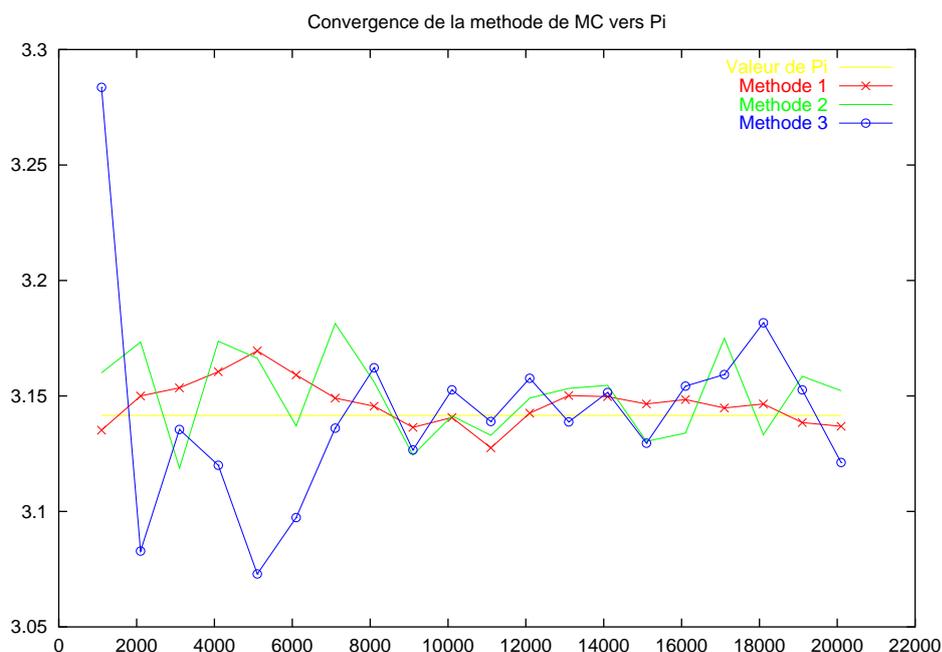
1 % Quelques graphiques pour faire joli
2 clear; clf; hold on; a=100; b=1000; step=20;
3 for k=1:step
4   n=a+b*k;

```

```

5  D(k)=n;
6  X1(k)=mean(4*(ones(1,n)-rand(1,n).^2).^(1/2));
7  X2=2*rand(1,n)-1;
8  Y2=2*rand(1,n)-1;
9  Z2(k)=mean(4*(X2.^2+Y2.^2<=1));
10 X3=2*rand(1,n)-1;
11 Y3=2*rand(1,n)-1;
12 Z3=2*rand(1,n)-1;
13 T3(k)=mean(3*2*(X3.^2+Y3.^2+Z3.^2<=1));
14 end
15 PI=pi*ones(1,k);
16 plot(D,PI,'k-')
17 plot(D,X1,'r-+')
18 plot(D,Z2,'g-.')
19 plot(D,T3,'b-*')
20 title('Convergence_de_la_methode_de_MC_vers_Pi')
21 legend('Valeur_de_Pi','Methode_1','Methode_2','Methode_3')

```

FIG. 3.2 – Approximation de π par méthodes de Monte-Carlo.

3.2.6 Exercices sur les intervalles de confiance

- `test1n` permet d'obtenir des intervalles de confiance sur la moyenne et la variance pour un échantillon simple (Student...). Pour n grand, on obtient également l'intervalle de confiance sur la moyenne dans le cas non-gaussien.
- `test2n` permet dans le cas de deux échantillons gaussiens de même variance d'obtenir des intervalles de confiance sur la différence des moyennes et sur la variance. Pour n grand, on obtient aussi l'intervalle de confiance sur la différence des moyennes dans le cas non-gaussien.
- `ciquant` fournit un intervalle de confiance sur le quantile d'ordre p de la loi d'un échantillon.

Exercice 3.2.2. Simuler un n -échantillon de variables de Bernoulli de paramètre p . Donner un intervalle de confiance à 95% pour p . Reproduire N fois la simulation précédente et déterminer le nombre de fois où l'intervalle de confiance proposé contient bien le véritable paramètre p . Reprendre cette étude avec des échantillons gaussiens (on calculera alors des intervalles de confiance pour la moyenne et pour la variance de ces variables).

Exercice 3.2.3. Dans le cadre de l'exercice précédent, comparer les intervalles de confiance exacts (obtenus avec la loi binomiale) à ceux donnés par le TCL.

Exercice 3.2.4. Générer un n -échantillon de loi gaussienne centré. Comparer les intervalles de confiance sur la variance obtenus avec la loi de la variance empirique et avec un TLC portant sur les X_i^2 . Utiliser d'autres lois. Le deuxième intervalle de confiance est-il utilisable si la moyenne est inconnue ?

Exercice 3.2.5. Générer deux n -échantillons de loi gaussiennes dont les moyennes sont données par deux réalisations de v.a. de loi uniforme sur $[-1, +1]$. Donner des intervalles de confiance de niveaux $1 - \alpha$ sur la différence des moyennes. Recommencer 100 fois et donner des intervalles de confiance sur la moyenne des différences à la moyenne.

Exercice 3.2.6. On considère 10 réalisations indépendantes de loi quelconque (par exemple, exponentielle, Poisson, Weibull, ...). Donnez des intervalles de niveau de confiance $1 - \alpha$ sur la moyenne et sur la variance en utilisant des quantiles empiriques calculés à partir des estimations des lois de la moyenne et de la variance.

3.3 Tests statistiques

La théorie des tests constitue un pan important de la statistique. Ce chapitre a pour but d'en donner quelques éléments. Dans l'optique de la préparation à l'oral, on pourra par exemple consulter les ouvrages en français [VK85], [DCD82b], [Tou99], [Sap89] et [TDM93]. Dans ce qui suit, nous commençons par introduire la notion de test en statistique paramétrique de façon succincte. Nous définissons ensuite les notions de niveau et de puissance d'un test. La section suivante présente brièvement le test du χ^2 , illustré par quelques exemples. Enfin, la dernière section est consacrée au test de Kolmogorov-Smirnov, qui fait appel au résultat sur la vitesse de convergence dans le théorème de Glivenko-Cantelli présenté dans la section 2.4 page 45.

3.3.1 Tests : niveau, puissance et notions asymptotiques

Dans le cadre de la statistique paramétrique, un test consiste à confronter les données à une hypothèse sur le paramètre. Ainsi, si des considérations extérieures (théorie, expérience historique, etc) nous font penser que la loi recherchée est \mathbb{P}_{θ_0} , on définit les sous-ensembles H_0 et H_1 de Θ par

$$H_0 := \{\theta_0\} \quad \text{et} \quad H_1 : \Theta \setminus \{\theta_0\}.$$

Un test T de niveau $\alpha \in [0, 1]$ (resp. au plus α) associé aux deux hypothèses antagonistes H_0 et H_1 est une application mesurable des données Y , à valeurs dans $\{0, 1\}$, telle que

$$\forall \theta \in H_0, \mathbb{P}_{\theta}(T(Y) = 1) = \alpha \text{ (rep. } \leq \alpha).$$

On parlera de *niveau asymptotique* lorsque cela est vrai quand la taille n de l'échantillon tend vers l'infini. La *région d'acceptation* de H_0 du test T est l'ensemble des valeurs des observations Y pour lesquelles $T(Y) = H_0$. On dit qu'un test T de niveau α est *sans biais* pour H_0 lorsque

$$\forall \theta \in H_1, \mathbb{P}_{\theta}(T(Y) = 1) \geq \alpha.$$

La *puissance* du test est par définition la fonction $\theta \in H_1 \mapsto \mathbb{P}_{\theta}(T(Y) \in H_1)$.

3.3.2 Test du khi deux

Considérons un phénomène aléatoire de loi inconnue $P = p_1\delta_1 + \dots + p_k\delta_k$ où l'on suppose que les p_i sont tous non nuls. Nous allons supposer que des considérations extérieures (théorie, etc) nous font penser que la loi du phénomène aléatoire est $Q = q_1\delta_1 + \dots + q_k\delta_k$, où les q_k sont tous non nuls. Partant d'observations supposées indépendantes du phénomène aléatoire, on désire tester l'hypothèse $H_0 : P = Q$ contre l'hypothèse antagoniste générale $H_1 : P \neq Q$. On définit la pseudo-distance du χ^2 entre deux distributions A et B discrètes sur $\{1, \dots, k\}$ par :

$$\chi^2(A, B) := \sum_{i=1}^k \frac{(A_i - B_i)^2}{B_i}.$$

Si les effectifs des n observations indépendantes sont donnés par N_1^n, \dots, N_k^n , on note

$$P_n := (p_1^n, \dots, p_k^n)$$

la distribution empirique associée donnée par $p_i^n := N_i^n/n$. On montre alors que :

1. La variable aléatoire $\sqrt{n} \left(\frac{p_1^n - p_1}{\sqrt{p_1}}, \dots, \frac{p_k^n - p_k}{\sqrt{p_k}} \right)$ converge en loi vers la loi normale centrée sur \mathbb{R}^k de matrice de covariance $I_k - \sqrt{P}(\sqrt{P})^T$.
2. Sous H_0 , $n\chi^2(P_n, Q)$ converge en loi vers la loi $\chi^2(k-1)$.
3. Sous H_1 , $n\chi^2(P_n, Q)$ converge en probabilité vers $+\infty$.

Le fait que la loi limite ne dépende du paramètre P qu'à travers sa taille k est très important ici. Un test du χ^2 de niveau (asymptotique) $\alpha \in [0, 1]$ consiste en ce qui suit, pour n « assez grand » :

- si $n\chi^2(P_n, Q) > \chi_\alpha^2(k-1)$, on choisit H_1 ,
- si $n\chi^2(P_n, Q) \leq \chi_\alpha^2(k-1)$, on choisit H_0 ,

où χ_α^2 est le fractile $1 - \alpha$ de la loi $\chi^2(k-1)$. On montre facilement que la puissance de ce test tend vers 1 quand n tend vers l'infini.

Test d'ajustement ou d'adéquation

Soit (X_1, \dots, X_n) un n -échantillon réel de loi inconnue μ . On suppose que les X_i prennent leurs valeurs dans k classes I_1, \dots, I_k constituées de k intervalles ou de k nombres selon que la loi μ est discrète ou continue. On note \hat{f}_i la fréquence empirique associée à la classe $i \in \{1, \dots, k\}$, définie par :

$$\hat{f}_i = \frac{1}{n} \sum_{m=1}^n \mathbb{I}_{\{X_m \in I_i\}}.$$

Soit ν une loi de probabilité connue, de même nature que μ . On désire tester l'hypothèse :

$$H_0 := \text{« La loi des } X_i \text{ est } \nu \text{ »}$$

contre l'hypothèse antagoniste

$$H_1 := \text{« La loi des } X_i \text{ n'est pas } \nu \text{ »}.$$

Les $f_i := \nu(I_i)$ sont calculables a priori, et on a alors sous H_0 :

$$n \sum_{i=1}^k \frac{(f_i - \hat{f}_i)^2}{f_i} \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \chi^2(k-1).$$

On utilise en pratique ce résultat lorsque pour tout i , $nf_i \geq 5$.

Test d'homogénéité

Soit (X_1, \dots, X_n) un n -échantillon réel de loi inconnue μ_1 et (Y_1, \dots, Y_m) un m -échantillon réel de loi inconnue μ_2 , les deux échantillons étant indépendants l'un de l'autre. On suppose que les X_i et Y_i prennent leurs valeurs dans k classes (I_1, \dots, I_k) . On note \hat{f}_i^X et \hat{f}_i^Y les fréquences empiriques associées à la classe $i \in \{1, \dots, k\}$. On désire tester l'hypothèse :

$$H_0 := \text{« } \mu_1 \text{ et } \mu_2 \text{ sont identiques »}$$

contre l'hypothèse antagoniste

$$H_1 := \text{« } \mu_1 \text{ et } \mu_2 \text{ sont différentes »}$$

On définit une sorte de fréquence empirique commune aux deux échantillons en posant, pour tout $i \in \{1, \dots, k\}$:

$$\hat{f}_i := \frac{n\hat{f}_i^X + m\hat{f}_i^Y}{n+m}.$$

On a alors sous H_0 :

$$n \sum_{i=1}^k \frac{(\hat{f}_i - \hat{f}_i^X)^2}{\hat{f}_i} + m \sum_{i=1}^k \frac{(\hat{f}_i - \hat{f}_i^Y)^2}{\hat{f}_i} \xrightarrow[n, m \rightarrow +\infty]{\mathcal{L}} \chi^2(k-1).$$

Test d'indépendance

Soit $((X_1, Y_1), \dots, (X_n, Y_n))$ un n -échantillon réel de loi inconnue μ . On note μ_1 et μ_2 les lois marginales de μ , et l'on suppose que les (X_i, Y_i) prennent respectivement leurs valeurs sur k classes (I_1, \dots, I_k) et l classes (J_1, \dots, J_l) . On note \hat{f}_{ij} la fréquence empirique associée à (I_i, J_j) , définie par :

$$\hat{f}_{ij} := \frac{1}{n} \sum_{k=1}^n \mathbf{I}_{\{(X_k \in I_i) \cap (Y_k \in J_j)\}}.$$

On désire tester l'hypothèse

$$H_0 := \ll \mu = \mu_1 \otimes \mu_2 \gg$$

contre l'hypothèse antagoniste

$$H_1 := \ll \mu \neq \mu_1 \otimes \mu_2 \gg.$$

Pour cela, on définit f_{ij} par :

$$f_{ij} := \frac{1}{n} \sum_{k=1}^n \mathbf{I}_{\{X_k \in I_i\}} \frac{1}{n} \sum_{k=1}^n \mathbf{I}_{\{Y_k \in J_j\}}.$$

On a alors sous H_0 :

$$n \sum_{i=1}^k \sum_{j=1}^l \frac{(f_{ij} - \hat{f}_{ij})^2}{\hat{f}_{ij}} \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \chi^2(k-1)(l-1).$$

Exercices sur les tests du Khi Deux

Exercice 3.3.1. Générez un échantillon de taille 100 de loi uniforme sur $[0, 1]$, décomposez le en k classes de même longueur et faites un test du χ^2 d'ajustement à la loi uniforme. Recommencez en augmentant la taille de l'échantillon, le nombre de classes, puis en changeant de loi : normale, binomiale, Poisson par exemple.

Exercice 3.3.2. Générez un échantillon de taille 100 de loi exponentielle et testez si l'échantillon est de loi normale, de Poisson ou binomiale.

Exercice 3.3.3. Générez deux échantillons de même loi – par exemple uniforme ou normale ou exponentielle ou binomiale – et de tailles différentes, par exemple 1000 et 10000. Testez ensuite l'homogénéité de ces deux échantillons. Faire de même avec deux échantillons de lois distinctes mais de même support.

Exercice 3.3.4 (Les mathéux sont-ils philosophes ?). On a relevé sur 100 bacheliers les notes obtenues en mathématiques X et en philosophie Y :

X \ Y	[0,4[[4,8[[8,12[[12,16[[16,20]
[0,4[3	4	2	0	0
[4,8[6	10	8	2	0
[8,12[1	8	20	12	3
[12,16[0	0	8	7	3
[16,20]	0	0	1	0	2

Testez l'hypothèse d'indépendance entre les notes obtenues dans les deux disciplines.

Exercice 3.3.5. Générez un échantillon (X_1, \dots, X_n) de taille $n = 1000$ de loi normale centrée réduite. Considérer $Y_i := X_i^2$, puis testez l'indépendance de X et de Y . Faire de même avec d'autres lois : Bernoulli, exponentielle, uniforme. . .

Exercice 3.3.6 (Cas où l'asymptotique du χ^2 ne s'applique pas). On reprend l'exercice [TDM93, page 97]. On désire savoir si la couleur du verre a une influence sur l'appréciation de la qualité d'un vin. On dispose des données suivantes :

Couleur du verre (du moins foncé au plus foncé)	1	2	3	4	5
Nombre de personnes désignant le vin comme le meilleur	0	1	0	5	4

Simulez la loi de la distance du χ^2 entre la loi uniforme sur $\{1, \dots, 5\}$ et la loi d'un échantillon équidistribué suivant une loi uniforme sur $\{1, \dots, 5\}$. En déterminant les quantiles à partir de l'histogramme de cette distance, testez l'hypothèse d'influence (H_0) pour un niveau de confiance quelconque.

Exercice 3.3.7. En reprenant les données de Stixbox obtenues par `getdata`, faites des tests d'ajustement pour les modéliser par des lois connues, que vous proposerez en traçant des histogrammes.

3.3.3 Test de Kolmogorov-Smirnov

Comme nous l'avons vu dans la section 2.4 page 45, la distribution de Kolmogorov-Smirnov \mathbf{F} définie en (2.3) page 45 intervient pour exprimer la vitesse de convergence dans le théorème de Glivenko-Cantelli 2.1.4 page 40.

Test d'adéquation

Soit (X_1, \dots, X_n) un n -échantillon de loi μ de fonction de répartition F continue. Si F_n désigne la fonction de répartition empirique de l'échantillon, on identifie la loi de $\sqrt{n} \sup_{x \in \mathbb{R}} |F_n(x) - F(x)|$ à celle de Kolmogorov-Smirnov pour n assez grand, $n \geq 80$ par exemple. On peut alors utiliser les quantiles à 0.95 et 0.99 correspondants à $u = 1.3581$ et $u = 1.629$, pour obtenir un test de risque 0.05 et 0.01 sur l'adéquation de la loi de l'échantillon à μ . Pour n petit, on peut utiliser des tables, cf. [Sap89]. La bibliothèque Stixbox fournit la fonction `pks` qui donne la valeur de la fonction de répartition de Kolmogorov-Smirnov et la fonction `kstwo` qui permet de tester l'homogénéité de deux lois continues. Il est très important de se souvenir que la convergence vers la loi de Kolmogorov-Smirnov n'est garantie que lorsque la loi μ n'a pas d'atomes.

Exercice 3.3.8 (Test d'adéquation pour les générateurs aléatoires). Voici deux petits programmes Matlab qui font appel au test de Kolmogorov-Smirnov pour tester les générateurs aléatoires gaussiens. Ils sont aisément adaptables aux autres générateurs aléatoires de distributions continues.

```

1  %%% Test de Kolmogorov-Smirnov pour générateurs gaussiens
2  clear
3  %
4  n=1000; alpha=0.05;
5  %
6  U=rand(1,n); V=rand(1,n);
7  G=sqrt(-2*log(U)).*cos(2*pi*V);
8  GG=sort(G);
9  % Ici, les GG(i) sont tous différents presque sûrement
10 % car la loi gaussienne est diffuse. C'est par ailleurs nécessaire
11 % à l'application du test de K-S.
12 x=sqrt(n)*max(abs([1:n]/n - pnorm(GG)));
13 p=pks(x);
14 if p<1.-alpha,
15     sprintf('On accepte H0. Niveau = %4.2f, n=%d, KS=%f.',alpha,n,p)
16 else
17     sprintf('On rejette H0. Niveau = %4.2f, n=%d, KS=%f.',alpha,n,p)
18 end

```

```

1  %%% Test de Kolmogorov-Smirnov pour générateurs gaussiens
2  clear; clf
3  %
4  n=1000; r=1000; X=zeros(1,r);
5  %
6  h = waitbar(0,'Patience ...');
7  for i=1:r,
8      U=rand(1,n); V=rand(1,n);
9      G=sqrt(-2*log(U)).*cos(2*pi*V);
10     %G=randn(1,n);
11     GG=sort(G);
12     % Ici, les GG(i) sont tous différents presque sûrement
13     % car la loi gaussienne est diffuse. C'est par ailleurs nécessaire
14     % à l'application du test de K-S.
15     X(i)=sqrt(n)*max(abs([1:n]/n - pnorm(GG)));
16     waitbar(i/r)
17 end
18 close(h)
19 X=sort(X);
20 hold on
21 plot(X,pks(X),'g-')

```

```

22 stairs(X,[1:r]/r)
23 legend('PKS','Femp')

```

Test d'homogénéité

Soient (X_1, \dots, X_n) et (Y_1, \dots, Y_m) deux échantillons de lois continues inconnues. On a la convergence en loi suivante pour tester si leurs lois sont identiques :

$$\sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n^{(X)}(x) - F_m^{(Y)}(x)| \xrightarrow[n, m \rightarrow +\infty]{\mathcal{L}} \mathbf{F},$$

où \mathbf{F} désigne la fonction de répartition de Kolmogorov-Smirnov.

Exercices faisant appel aux processus empiriques

Exercice 3.3.9. Générez un échantillon de taille 100 de loi uniforme sur $[0, 1]$ puis tracez sur le même graphique la fonction de répartition empirique et la fonction de répartition théorique. Faites un test de Kolmogorov-Smirnov sur cet échantillon pour un risque de 0.05, recommencer 100 fois le test. Combien de fois l'hypothèse H_0 est-elle acceptée? Faites la même chose pour la loi normale centrée réduite et comparez au résultat obtenu avec un test du χ^2 .

Exercice 3.3.10. Reprenez l'exercice précédent avec un échantillon de taille 10 de loi uniforme, et tracer les fonctions de répartitions empirique et théorique. Pour utiliser le test, déterminez la distribution empirique de $\sqrt{10} \sup_{x \in [0, 1]} |F_n(x) - F(x)|$ en utilisant 1000 échantillons de taille 10 de loi uniforme sur $[0, 1]$. En déduire le quantile empirique à 0.95 et conclure pour le test du premier échantillon. Refaites la même chose avec la loi normale.

Exercice 3.3.11. Faire un test d'homogénéité de Kolmogorov-Smirnov pour deux échantillons de loi uniformes sur $[0, 1]$ et de tailles respectives 100 et 1000.

Exercice 3.3.12. La mesure empirique d'un échantillon (X_1, \dots, X_n) constitue une approximation discrète (atomique) de la densité de la loi commune de l'échantillon. On peut la lisser en remplaçant les masses de Dirac par des approximations lisses : on parle alors de *méthode d'estimation par noyaux*. Le plus souvent, le lissage est obtenu en prenant la convolée de la distribution empirique par une fonction lisse appelée *noyau*. Plus précisément, soit $K : \mathbb{R} \rightarrow \mathbb{R}^+$ une fonction bornée, intégrable pour la mesure de Lebesgue et nulle à l'infini, par exemple $K = \mathbb{I}_{[-5, 5]}$ ou encore $K(x) = (2\pi)^{-1/2} \exp(-x^2/2)$. On considère alors

$$\hat{f}_{n,h}(x)(\omega) := \frac{1}{n} \sum_{k=1}^n \frac{1}{h} K\left(\frac{x - X_k(\omega)}{h}\right).$$

Si h est une fonction de n telle que $\lim_{n \rightarrow +\infty} h(n) = 0$ et $\lim_{n \rightarrow +\infty} (n/\log n)h(n) = +\infty$, si f est continue en x avec $f(x) \neq 0$, et enfin si K est à support compact, alors

$$\left| \hat{f}_{n,h}(x) - f(x) \right| \xrightarrow[n \rightarrow +\infty]{\text{P.-S.}} 0.$$

$\hat{f}_{n,h}(x)$ est donc un estimateur convergent de $f(x)$. Ecrivez un code Matlab qui implémente cette méthode d'estimation de la densité par noyaux.

Fragment 4

Vecteurs aléatoires gaussiens

Un vecteur aléatoire gaussien de dimension n est un vecteur aléatoire $X : (\Omega, \mathcal{A}, \mathbb{P}) \rightarrow \mathbb{R}^n$ tel que pour tout $\alpha \in \mathbb{R}^n$, la variable aléatoire réelle $\alpha^\top X := \alpha_1 X_1 + \dots + \alpha_n X_n$ suit une loi gaussienne sur \mathbb{R} . La loi d'un vecteur gaussien est caractérisée par son vecteur moyenne $m := (\mathbb{E}(X_1), \dots, \mathbb{E}(X_n)) \in \mathbb{R}^n$ et sa matrice de covariance $\Gamma \in \mathcal{S}_n^+$ définie par

$$\Gamma := (\mathbb{E}[(X_i - m_i)(X_j - m_j)])_{1 \leq i, j \leq n} = (\mathbb{E}(X_i X_j) - m_i m_j)_{1 \leq i, j \leq n}.$$

La loi d'un tel vecteur, notée $\mathcal{N}(m, \Gamma)$, a pour densité sur \mathbb{R}^n

$$(2\pi)^{-\frac{n}{2}} \det \Gamma^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x - m)^\top \Gamma^{-1}(x - m)\right) dx.$$

On parle de loi gaussienne standard lorsque $(m, \Gamma) = (0, \mathbf{Id}_n)$. Les composantes d'un vecteur gaussien sont indépendantes si et seulement si sa matrice de covariance est diagonale. En conséquence, un vecteur gaussien *centré* a des composantes indépendantes si et seulement si elles sont orthogonales dans $\mathbf{L}^2(\Omega, \mathcal{A}, \mathbb{P})$, c'est-à-dire que pour tous $1 \leq i, j \leq n$, on a $\mathbb{E}(X_i X_j) = 0$.

De plus, on montre, que pour un vecteur gaussien *centré* X de \mathbb{R}^n , le sous espace vectoriel de $\mathbf{L}^2(\Omega, \mathcal{A}, \mathbb{P})$ constitué par les variables aléatoires centrées de carré intégrable $\sigma(X_{k+1}, \dots, X_n)$ -mesurables est constitué par les combinaisons linéaires des v.a.r. X_{k+1}, \dots, X_n . En conséquence, $\mathbb{E}(X_i | X_{k+1}, \dots, X_n)$ est la projection orthogonale dans $\mathbf{L}^2(\Omega, \mathcal{A}, \mathbb{P})$ de X_i sur l'espace vectoriel engendré par les X_{k+1}, \dots, X_n .

4.1 Lois conditionnelles

Soit X un vecteur gaussien *centré* de \mathbb{R}^n de matrice de covariance $\Gamma \in \mathcal{S}_n^+$. Soit $k \in \{1, \dots, n\}$ et soit $Z := (X_1, \dots, X_k)$ et $Y := (X_{k+1}, \dots, X_n)$. On démontre en utilisant la transformée de Fourier que la loi conditionnelle $\mathcal{L}(Z|Y = y)$ de Z sachant que $Y = y$ est encore une loi gaussienne, de moyenne $\mathbb{E}(Z|Y = y) \in \mathbb{R}^k$ et de matrice de covariance $\Sigma \in \mathcal{S}_k^+$ donnée par

$$\Sigma_{i,j} := \mathbb{E}((X_i - \mathbb{E}(X_i|Y))(X_j - \mathbb{E}(X_j|Y))) = \Gamma_{ij} - \mathbb{E}(\mathbb{E}(X_i|Y)\mathbb{E}(X_j|Y)),$$

qui ne dépend pas de y . Or on sait que $\mathbb{E}(X_i|Y) = \lambda_{i,k+1} X_{k+1} + \dots + \lambda_{i,n} X_n$, d'où, en multipliant par X_{k+j} puis en prenant l'espérance : $\Gamma_{i,k+j} = \lambda_{i,k+1} \Gamma_{k+1,k+j} + \dots + \lambda_{i,n} \Gamma_{n,k+j}$ car comme X est centré on a $\Gamma = (\mathbb{E}(X_i X_j))_{1 \leq i, j \leq n}$. On en déduit qu'il suffit de résoudre les k systèmes linéaires

$$\begin{pmatrix} \Gamma_{k+1,k+1} & \cdots & \Gamma_{n,k+1} \\ \vdots & & \vdots \\ \Gamma_{k+1,n} & \cdots & \Gamma_{n,n} \end{pmatrix} \begin{pmatrix} \lambda_{i,k+1} \\ \vdots \\ \lambda_{i,n} \end{pmatrix} = \begin{pmatrix} \Gamma_{i,k+1} \\ \vdots \\ \Gamma_{i,n} \end{pmatrix}$$

Pour cela, il suffit d'inverser la matrice $\Gamma^{(k)} := (\Gamma_{i,j})_{k+1 \leq i, j \leq n}$ une fois pour toute. On a alors :

$$\Sigma_{ij} = \Gamma_{ij} - \lambda_i^\top \Gamma^{(k)} \lambda_j = \Gamma_{ij} - \Gamma_{(k,i)}^\top \Gamma_{(k)}^{-1} \Gamma_{(k,i)},$$

où $\Gamma_{(k,i)} := (\Gamma_{i,k+1}, \dots, \Gamma_{i,n})^\top$. Voici un exemple d'illustration en Matlab :

```

1 % Lois conditionnelles d'un vecteur gaussien
2 clear
3
4 n=10; k=6; % (dim X,dim Z)=(n,k) 0<k<n
5 y=rand(n-k,1); % vecteur des valeurs de Y pour le calcul de L(Z|Y=y)
6
7 % G = matrice de covariance de X, doit etre >0
8 X=rand(n,n);
9 G=X'*X+.1*eye(n);
10
11 % Calcul des coefficients lambda(i,j)
12 L=zeros(k,n-k);
13 for i=1:k,
14     L(i,:)=G(i,k+1:n)/G(k+1:n,k+1:n);
15 end
16
17 % Calcul de E(Z|Y=y)
18 moyenne = dot(L(i,:),y)
19 % Calcul de cov = matrice de cov de L(Z|Y=y)
20 cov=zeros(k,k);
21 for i=1:k,
22     for j=1:k,
23         cov(i,j)=-L(i,:)*G(k+1:n,k+1:n)*L(i,:);
24     end
25 end
26 cov=G(1:k,1:k)-cov

```

4.2 Entropie de Shannon d'un vecteur aléatoire

L'entropie de Shannon de base $b > 0$ d'une loi discrète $p_1\delta_1 + \dots + p_n\delta_n$ est définie par $-\sum_{i=1}^n p_i \log_b p_i$. Elle mesure le nombre moyen de chiffres en base b nécessaires au codage de X . L'entropie de base 2 d'une loi de Bernoulli de paramètre $1/2$ vaut 1 (il ne faut qu'un bit). L'entropie discrète est toujours positive (inégalité de Jensen) et est maximisée à n et b fixés par la loi uniforme. De la même manière, l'entropie de Shannon d'un vecteur aléatoire X de \mathbb{R}^n de loi de densité $f : \mathbb{R}^n \rightarrow \mathbb{R}^+$ est définie par

$$\mathbf{H}(X) := -\mathbb{E}(\log f(X)) = -\int f \log f \, dx.$$

Elle prend ses valeurs dans tout \mathbb{R} , contrairement à l'entropie discrète. Calculée avec un logarithme de base b , elle mesure la précision nécessaire au codage de X en base b . La quantité $\mathbf{H}(X)$ ne dépend que de la loi $\mathcal{L}(X)$ de X . Elle ne dépend pas de sa moyenne. L'entropie d'une loi gaussienne $\mathcal{N}(m, \Gamma)$ est donnée par $(1/2) \log((2\pi e)^n \det \Gamma)$. Si l'on note K_X la matrice des corrélations de X , qui n'est rien d'autre que la matrice de covariance quand X suit une loi gaussienne. On montre alors en utilisant l'inégalité de Jensen que pour toute matrice symétrique positive $K \in \mathcal{S}_n^+$, on a

$$\sup_{K_X=K} \mathbf{H}(X) = \mathbf{H}(\mathcal{N}(0, K)) = \frac{1}{2} \log((2\pi e)^n \det K).$$

Ainsi, les lois gaussiennes maximisent l'entropie à covariance fixée. Pour mesurer la perte d'information entre deux vecteurs gaussiens de matrices de covariances respectives Γ_1 et Γ_2 , on peut calculer la réduction d'entropie (ER), qui vaut $(-1/2) \log(\det(\Gamma_1 \Gamma_2^{-1}))$. Une autre méthode consiste utiliser ce que l'on appelle le degrés de liberté (DF), qui est donné par $\text{Tr}(\mathbf{Id}_n - \Gamma_1 \Gamma_2^{-1})$. En diagonalisant $\Gamma_1 \Gamma_2^{-1}$, la comparaison classique entre moyenne arithmétique et géométrique $(\lambda_1 \cdots \lambda_n)^{1/n} \leq (\lambda_1 + \cdots + \lambda_n)/n$, appliquée au spectre de $\Gamma_1 \Gamma_2^{-1}$ donne que $ER \geq -(n/2) \log_2(1 - DF/n)$. Voici un exemple écrit en Matlab illustrant la perte en degrés de liberté et la réduction d'entropie dans le programme précédent. Sa sortie graphique est donnée par la figure 4.1 :

```

1 %%% Tracé de l'écart en entropie et en degrés de liberté
2 %%% lors de la simulation d'une loi gaussienne multivariée
3 clear

```

```

4 X = rand(4,4); G = 1*eye(4)+X'*X/4; % G = matrice de cov.
5 A = sqrtm(G); % A = Racine carrée de G
6 n = 1000; % Nombre de réalisations
7 realis = zeros(4,n);
8 % Calcul matrice de cov. empirique au fil des n réalisations.
9 covemp = zeros(4,4); moyemp = zeros(4,n);
10 df = zeros(1,n); er = zeros(1,n);
11 for k=1:n,
12     realis(:,k) = A*randn(4,1);
13     moyemp(:,k) = sum(realis(:,1:k)')';
14     for i=1:4,
15         covemp(i,i) = sum(realis(i,1:k).^2)/k - moyemp(i,k)^2/k;
16         for j=i+1:4,
17             covemp(i,j) = dot(realis(i,1:k),realis(j,1:k))/k;
18             covemp(i,j) = covemp(i,j) - moyemp(i,k)*moyemp(j,k)/k;
19             covemp(j,i) = covemp(i,j);
20         end
21     end
22     X=covemp/G;
23     df(k) = 4-sum(trace(X));
24     er(k) = -.5*log2(det(X));
25 end
26 figure(1); clg
27 hold on
28 xlabel('Nombre_de_réalisations')
29 ylabel('DL_&_RE')
30 plot(df(10:n),'r-','Degrés_de_liberte;')
31 plot(er(10:n),'b-','Reduction_d''entropie;')
32 title('Comparaison_matrices_de_cov._empiriques_et_théoriques')
33 %Idee: utiliser la fonc cov ou cor de octave (et MatLab ?)

```

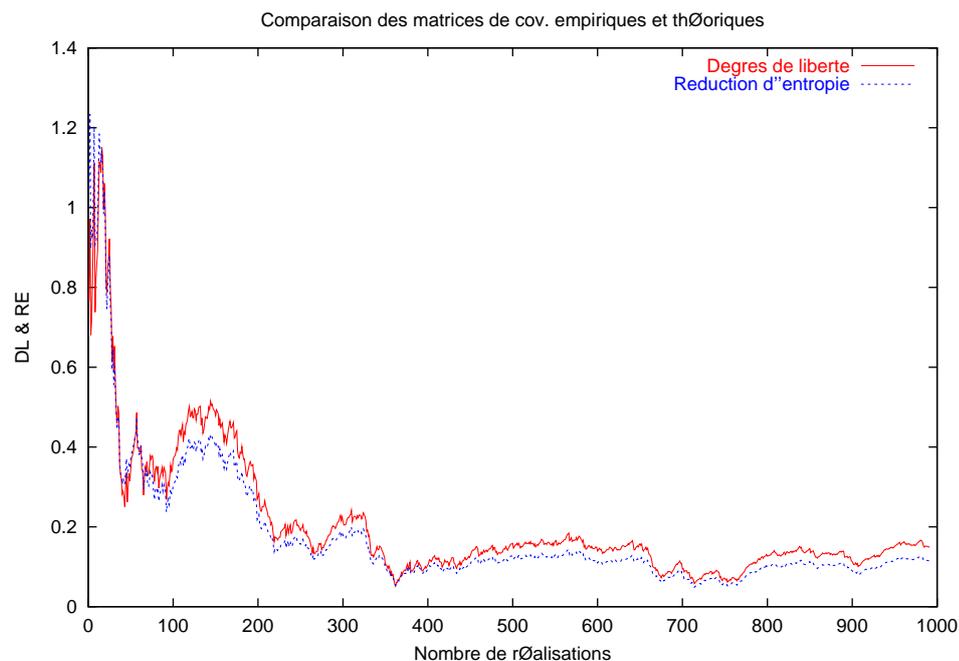


FIG. 4.1 – Réduction d'entropie et de DFS empirique/théorique lors d'une simulation d'un vecteur gaussien.

4.3 Le filtre de Kalman

Partie à intégrer

4.4 Le modèle linéaire gaussien

Partie à intégrer

Fragment 5

Chaînes de Markov & martingales

Les processus à accroissements indépendants comme la marche aléatoire simple, le processus de Poisson et le mouvement brownien sont à la fois des martingales et des processus de Markov. Ces deux propriétés correspondent à deux descriptions différentes de la façon dont le passé du processus intervient dans son évolution future. Les chaînes de Markov et les martingales constituent des généralisations naturelles de la notion de suites de v.a. à accroissements indépendants.

5.1 Chaînes de Markov à espace d'état fini

Une chaîne de Markov sur un espace d'état fini $E := \{x_1, \dots, x_n\}$ est une suite de variables aléatoires $(X_n)_{n \geq 0}$ à valeurs dans E telles que pour tout n ,

$$\mathcal{L}(X_{n+1}|X_0, \dots, X_n) = \mathcal{L}(X_{n+1}|X_n).$$

Le « futur » et le « passé » de la chaîne sont conditionnellement indépendants par rapport au « présent ».

Lorsque $\mathcal{L}(X_{n+1}|X_n)$ ne dépend pas de n , on dit que la chaîne est homogène et il suffit alors, pour caractériser sa loi, de donner la loi μ sur E de X_0 ainsi que la « matrice de transition » P donnée par :

$$(P_{ij})_{1 \leq i, j \leq n} := (P(X_{n+1} = x_j | X_n = x_i))_{1 \leq i, j \leq n}.$$

Ainsi, les entrées de cette matrice sont dans $[0, 1]$ et chaque ligne est une loi conditionnelle (donc de somme égale à 1). On parle alors de matrice markovienne ou stochastique. Si l'on assimile les lois sur E à des vecteurs ligne et les fonctions de E dans \mathbb{R} à des vecteurs colonne, on a, en notant μ la loi de X_0 et f une fonction quelconque de E dans \mathbb{R} :

$$\mathcal{L}(X_n) = \mu P^n \quad \text{et} \quad \mathbf{E}(f(X_{n+k})|X_n = x_i) = (P^k f)_i.$$

Dans la suite, nous nous intéresseront qu'aux chaînes de Markov homogènes. On dit qu'une loi μ sur E est invariante pour une chaîne de Markov de matrice de transition P ssi on a $\mu P = \mu$. En d'autres termes, le vecteur μ^\top est un vecteur propre de P^\top associé à la valeur propre 1. Un théorème du cours affirme qu'il existe toujours une loi invariante. Si X_0 suit la mesure invariante, il en sera de même de tous les X_n .

5.1.1 Récurrence et irréductibilité

Pour tout x_i, y_j dans E , on dit que $x_i > x_j$ ssi $P_{ij} > 0$ et que $x_i \gg x_j$ lorsqu'il existe une suite finie de points x_{k_1}, \dots, x_{k_n} de E avec $x_i > x_{k_1} > \dots > x_{k_n} > x_j$. Un point x_i de E est dit récurrent lorsque pour tout x_j de E , $x_i \gg x_j$ entraîne $x_j \gg x_i$. Les points non récurrents sont dit transitoires. On montre que sur l'ensemble des points récurrents, la relation de récurrence \gg est une relation d'équivalence.

On peut montrer qu'une mesure invariante ne charge pas les points transitoires. On dit qu'une matrice markovienne est récurrente-irréductible lorsqu'il n'y pas de points transitoires et qu'il n'y a qu'une seule classe de récurrence. Dans ce cas, on peut montrer l'unicité de la mesure invariante.

Un théorème du cours montre que si $(X_n)_{n \geq 0}$ est une chaîne de Markov dont la matrice de transition P est récurrente-irréductible, alors, quelque soit la loi μ de X_0 , la moyenne arithmétique des lois $\mathcal{L}(X_0), \dots, \mathcal{L}(X_{n-1})$ converge vers l'unique mesure invariante ν de la chaîne. En d'autres termes, pour tout vecteur ligne μ sur E , $(\mu + \mu P + \dots + \mu P^{n-1})/n$ converge vers ν .

5.1.2 Apériodicité

On appelle période d'un point x_i de E le PGCD des entiers n tels que $P_{ii}^n > 0$. On peut montrer que la période est constante sur chaque classe de récurrence. On dit qu'une matrice markovienne P récurrente-irréductible est apériodique lorsque la période est égale à 1. On peut montrer que si P est récurrente-irréductible, alors il y a équivalence entre son apériodicité, l'existence d'un entier m tel que P^m a toutes ses entrées strictement positives et enfin le fait que 1 est la seule valeur propre de module 1 de P .

On montre que si $(X_n)_{n \geq 0}$ est une chaîne de Markov de matrice de transition P récurrente-irréductible, alors elle est apériodique ssi pour toute loi initiale μ (de X_0), la suite $(X_n)_{n \geq 0}$ converge en loi vers l'unique mesure invariante ν (i.e. μP^n converge vers ν pour tout vecteur ligne μ).

5.1.3 Théorème ergodique

Si $(X_n)_{n \geq 0}$ est une chaîne de Markov de matrice de transition P récurrente-irréductible et de mesure invariante ν , alors pour toute fonction f de E dans \mathbb{R} et toute loi initiale μ (loi de X_0), on a :

$$\lim_{n \rightarrow +\infty} \frac{f(X_0) + \dots + f(X_{n-1})}{n} = \nu(f) \text{ p.s.}$$

En d'autres termes, la mesure empirique converge presque sûrement vers la mesure invariante. Appliqué à $f(x) = \delta_x$ avec $x \in E$, on obtient que $\#\{k \leq n, X_k = x\}/n$ converge presque sûrement vers $\nu(x)$.

Les trois théorèmes de convergence que nous avons énoncés expriment le fait que la mesure initiale (loi de X_0) est « oubliée » au profit de la mesure invariante.

Exercice 5.1.1. Déterminez la matrice de transition de la chaîne de Markov sur $\{0, 1\}$ telle que $P(X_{n+1} = 1|X_n = 1) = p$ et $P(X_{n+1} = 0|X_n = 0) = q$. Dans quels cas est-elle récurrente-irréductible? apériodique? Dans ce dernier cas, déterminer sa mesure invariante théorique et la vérifier numériquement, d'abord en considérant les puissances de la matrice de transition pour différentes valeurs de p , puis en diagonalisant la matrice de transition.

Exercice 5.1.2. En utilisant la méthode décrite dans la feuille « Simulation », écrire une fonction Matlab *rtps* de simulation d'une loi discrète $p\delta_0 + q\delta_1 + r\delta_2 + s\delta_3$ sur $\{0, 1, 2, 3\}$ à partir de la loi uniforme, qui prend comme paramètre les probabilités p, q et r . S'en servir pour écrire une fonction *rmtps* qui permet de simuler la loi d'une chaîne de Markov à l'étape n sur $\{0, 1, 2, 3\}$ à partir de sa matrice de transition et du point initial.

Exercice 5.1.3. Construire une matrice de transition sur $\{0, 1, 2, 3\}$ récurrente-irréductible, d'abord apériodique, puis non-apériodique. En utilisant la fonction *rmtps*, regarder dans les deux cas les différents régimes asymptotiques suivant les valeurs initiales de la chaîne.

5.1.4 L'exemple du modèle d'Ehrenfest

On considère deux urnes contenant respectivement k et $n - k$ boules. À chaque étape, on choisit aléatoirement une des n boules et on la change d'urne. Si X_n représente le nombre de boules dans la première urne à l'étape n , et si $X_i = j$, alors on déplace une boule de la première urne vers la seconde avec probabilité j/n et réciproquement avec la probabilité $1 - j/n$. Ainsi définie, la suite $(X_n)_{n \geq 0}$ est une chaîne de Markov homogène.

Déterminez la matrice de transition, puis montrez qu'elle est récurrente-irréductible. Vérifier que la loi binomiale de taille n et de paramètre $1/2$ est invariante (car symétrique, i.e. $\nu(i)P_{ij} = \nu(j)P_{ji}$ pour tout i et j).

Voici un programme Matlab qui compare la distribution empirique à la loi binomiale (on a convergence presque sûre d'après le théorème ergodique) :

```

1  clf; % On efface les figures
2  clear; % On nettoie la memoire
3  n=input('Nombre_total_de_boules_(ex_10)_=_' );
4  k=input('Nombre_initial_de_boules_dans_la_premiere_urne_(ex_2)_=_' );
5  m=input('Nombre_de_termes_dans_le_thm_ergodique_(ex_300)_=_' );
6  X(1)=k; % X(i)= nombre de particules dans la premiere urne a l'etape i
7  for i=1:m-1
8  % Notez que la Bernoulli de parametre p est une binomiale B(1,p)

```

```

9 % Ici, on simule une bernoulli de parametre p=1-X(i)/n sur -1,+1 a partir
10 % d'une bernoulli de meme parametre sur 0,1.
11 u=2*rbinom(1,1,1-(X(i)/n))-1;
12 X(i+1)=X(i)+u;
13 end
14 histo(X,n+1,1);
15 hold on;
16 Y=dbinom(0:n,n,1/2);
17 plot([0:n], Y*m,'r');
18 title('Mesures_invariantes_empirique_et_theorique');
19 legend('Mesure_invariante_simulee','Loi_binomiale')

```

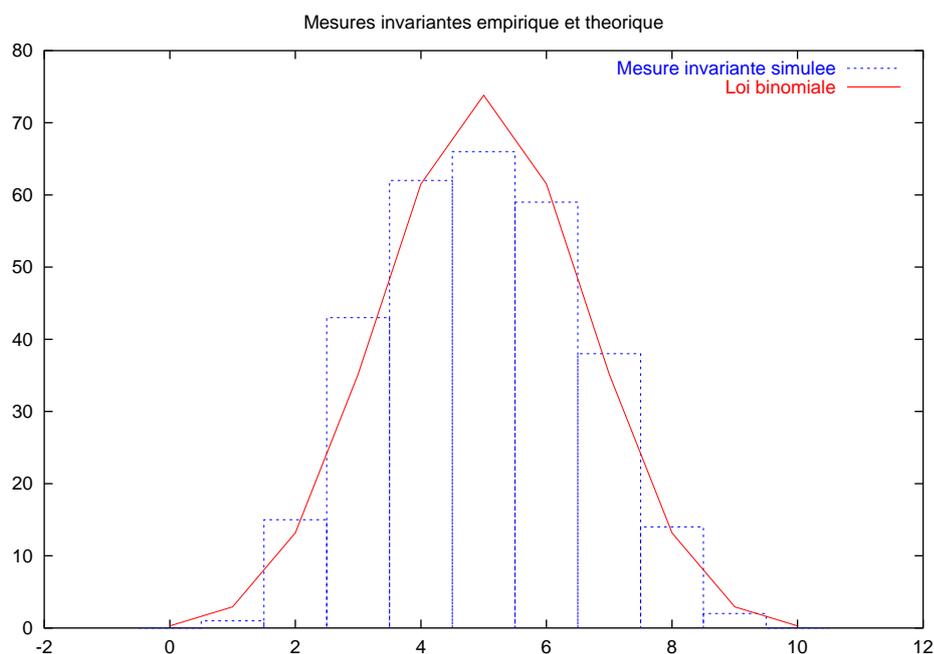


FIG. 5.1 – Comparaison des mesures ivariantes théorique et simulée pour le modèle d'Ehrenfest.

Exercice 5.1.4. On pourrait adapter la fonction *rmtpsim* pour simuler une loi $p_1\delta_0, \dots, p_n\delta_n$ sur $\{0, \dots, n\}$ à partir de la donnée des probabilités p_i , ce qui permettrait de simuler une chaîne de Markov quelconque à partir de la donnée de sa matrice de transition et de son état initial. Cela pourrait faire une illustration amusante pour la leçon sur la simulation de v.a.

5.2 Martingales et temps d'arrêt

5.2.1 Temps d'arrêt

Le théorème de convergence des martingales est parfois implicitement utilisé. On notera que certaines propriétés, comme la finitude p.s. des temps d'arrêt, ne peuvent pas se constater numériquement.

Un exemple simple de temps d'arrêt: la ruine du joueur

Le jeu consiste à partir avec une somme d'argent initiale a , puis à jouer à plusieurs parties indépendantes dont le gain est $+1$ avec probabilité p et -1 avec probabilité $q := 1 - p$. Le joueur s'arrête dès qu'il a atteint une somme b avec $b > a$ fixée d'avance (il a gagné) ou dès qu'il est ruiné (il a perdu). On modélise sa fortune à l'issue de la n -ième partie par la variable aléatoire

$$S_n := a + X_1 + \dots + X_n,$$

où $(X_i)_{i \in \mathbb{N}^*}$ est une suite de v.a. i.i.d. de loi de Bernoulli $p\delta_{+1} + q\delta_{-1}$. Par convention, $S_0 := a$. L'instant de fin du jeu T est alors donné par :

$$T := \inf \{n \in \mathbb{N}, S_n = b \text{ ou } S_n = 0\}.$$

On désire calculer la probabilité de gain, qui n'est rien d'autre que $\mathbb{P}(S_T = b)$, et également la durée moyenne du jeu $\mathbb{E}(T)$. Il nous faut cependant vérifier que le jeu s'arrête bien presque-sûrement. Traitons le cas où $m = p - q \neq 0$. Nous savons que $(M_n)_{n \in \mathbb{N}} := (S_n - nm)_{n \in \mathbb{N}}$ est une martingale pour $(\sigma(X_k, k \leq n))_{n \in \mathbb{N}^*}$. La v.a.r. T est un temps d'arrêt mais n'est pas finie p.s. a priori. En appliquant le théorème d'arrêt à la martingale $(M_k)_{k \in \mathbb{N}}$ et au temps d'arrêt $T \wedge n$, on obtient :

$$\mathbb{E}(M_0) = \mathbb{E}(M_{T \wedge n}) = \mathbb{E}((b - Tm) 1_{T \leq n}) + \mathbb{E}((S_n - nm) 1_{T > n}).$$

Or sur l'évènement $\{T > n\}$, on a $S_n < b$ d'où :

$$a \leq b - m\mathbb{E}(T 1_{T \leq n}).$$

Si $m > 0$, on en déduit par convergence monotone $\mathbb{E}(T) \leq (b - a)/m$, et donc T est fini presque-sûrement. Le cas $m < 0$ se déduit par symétrie en permutant les rôles de p et q et de a et $b - a$, ce qui donne la majoration $\mathbb{E}(T) \leq -a/m$. Il nous reste maintenant à calculer $\mathbb{P}(S_T = b)$ et $\mathbb{E}(T)$. Remarquons que la variable aléatoire S_T ne prend que deux valeurs distinctes b et 0 , et on a donc toujours $\mathbb{E}(S_T) = b\mathbb{P}(S_T = b)$. La suite $(r^{S_n})_{n \in \mathbb{N}^*}$ est une martingale pour $r = q/p$, et le théorème d'arrêt appliqué avec le t.d.a. fini p.s. T donne alors :

$$r^a = \mathbb{E}(r^{S_0}) = \mathbb{E}(r^{S_T}) = r^b\mathbb{P}(S_T = b) + r^0\mathbb{P}(S_T = 0),$$

d'où enfin :

$$\mathbb{P}(S_T = b) = \frac{r^a - 1}{r^b - 1}.$$

D'autre part, le théorème d'arrêt appliqué à la martingale $(S_n - nm)_{n \in \mathbb{N}}$ et au t.d.a. fini p.s. T donne :

$$a = \mathbb{E}(S_0) - 0m = \mathbb{E}(S_T) - \mathbb{E}(T)m,$$

d'où enfin puisque¹ $\mathbb{E}(S_T) = b\mathbb{P}(S_T = b)$:

$$\mathbb{E}(T) = \frac{1}{m} \left(\frac{r^a - 1}{r^b - 1} b - a \right).$$

Lorsque $m = 0$, on peut utiliser la martingale $(S_n^2 - n)_{n \in \mathbb{N}}$ pour montrer que T est fini p.s., puis pour montrer que $a^2 = \mathbb{E}(S_T^2) - d\mathbb{E}(T)$, puis la martingale $(S_n)_{n \in \mathbb{N}}$, qui donne $a = \mathbb{E}(S_T)$ comme pour le cas $m \neq 0$. En définitive, puisque $\mathbb{E}(S_T) = b\mathbb{P}(S_T = b)$ et $\mathbb{E}(S_T^2) = b^2\mathbb{P}(S_T = b)$, on obtient :

$$\mathbb{P}(S_T = b) = \frac{a}{b} \quad \text{et} \quad \mathbb{E}(T) = a(b - a).$$

Ces formules peuvent également s'obtenir en passant à la limite $m \rightarrow 0$ dans les formules obtenues pour $m \neq 0$.

Voici un exemple de code Matlab illustrant ce qui précède. Le fait que T soit fini presque sûrement et même intégrable assure théoriquement que le programme ne fera pas de boucle infinie ! Cela dit, l'infini pratique n'est pas l'infini théorique, et nous savons juste qu'en moyenne, une boucle aura $\mathbb{E}(T)$ tours, ce qui peut être énorme selon les valeurs de a , b et p . La connaissance de la variance de T serait alors utile. On peut calculer la transformée de Laplace $\mathbb{E}(\exp(\beta T))$ de T en utilisant la martingale $(\exp(\alpha S_n - \beta nm))_{n \in \mathbb{N}}$ où α est bien choisit en fonction de β :

$$\alpha = \frac{e^{\beta m} \pm \sqrt{e^{2\beta m} - 4pq}}{2p},$$

cf. [CDGC80, exercice 8.8 page 247].

¹De façon plus générale, le théorème d'arrêt permet de montrer que si $(X_n)_{n \in \mathbb{N}}$ est une suite de v.a. i.i.d. et si T est un t.d.a. intégrable relatif à la filtration naturelle associée à la suite précédente, alors $\mathbb{E}(S_T) = \mathbb{E}(X_1)\mathbb{E}(T)$ où $S_n := X_1 + \dots + X_n$. On parle alors d'identité ou de Lemme de Wald.

```

1  clf; clear
2
3  %Mettre plutôt une fortune entière (en Fr).
4  M=input('Fortune_initiale_du_joueur')
5
6  % Initialisations (à ne pas oublier)
7  G(1)=0;t=0;
8
9  while (G(t+1)>-M) t=t+1;
10     % Boucle conditionnée
11     u=rand(1);
12     if (u<0.5) G(t+1)=G(t)-1;
13         % Le joueur a une probabilité 1/2 de perdre 1 Fr.
14     end % Fin du if.
15     if (u>=0.5) G(t+1)=G(t)+1;
16         % Le joueur a une probabilité 1/2 de gagner 1 Fr.
17     end % Fin du if.
18 end % Fin du while
19
20 plot(G) % Tracé de l'évolution du gain.

```

Tentez de retrouver les formules précédentes pour $\mathbb{E}(T)$ et $\mathbb{P}(S_T = b)$ par la simulation, aussi bien pour $m = 0$ que pour $m \neq 0$. Tracez un histogramme de la loi approchée de T obtenu à partir de plusieurs réalisations indépendantes de T .

Autre exemple : enveloppe de Snell et arrêt optimal

Soit X_0, \dots, X_N une suite finie de variables aléatoires adaptée pour une filtration $\mathcal{F}_0, \dots, \mathcal{F}_N$. On cherche un temps d'arrêt T pour cette filtration, à valeurs dans $\{0, \dots, N\}$, qui maximise la quantité $\mathbb{E}(X_T)$. On parle alors de problème d'arrêt optimal. La solution est fournie par l'*enveloppe de Snell* S_0, \dots, S_N de la suite X_0, \dots, X_N , définie par récurrence décroissante en partant de N de la façon suivante :

$$S_N = X_N,$$

et pour $k \in \{0, \dots, N-1\}$

$$S_k = \max(X_k, \mathbb{E}(S_{k+1} | \mathcal{F}_k))$$

Le temps d'arrêt recherché s'exprime alors comme le plus petit entier pour lequel la suite se confond avec son enveloppe de Snell :

$$T := \inf \{k, X_k = S_k\}.$$

L'enveloppe de Snell est également la plus petite sur-martingale qui majore la suite X_1, \dots, X_n . Le théorème d'arrêt permet de montrer que T a bien la propriété recherchée. On pourra consulter [LL97] pour une application à la finance (call pour option américaine).

Temps d'atteinte pour une marche aléatoire

Soit A un sous-ensemble *fini* de \mathbb{Z}^d , $x \in A$ et $(S_n)_{n \in \mathbb{N}}$ la marche aléatoire simple issue de x . On considère alors le temps de sortie T de A , défini par :

$$T := \inf \{n, S_n \notin A\}.$$

Bien entendu, T dépend à la fois de A et de x et est un temps d'arrêt pour la filtration naturelle de la marche aléatoire. Il est facile de montrer qu'il existe un $C < \infty$ et un $\rho < 1$ dépendants de A tels que pour tout n et pour tout $x \in A$:

$$\mathbb{P}^x(T \geq n) \leq C \rho^n.$$

En conséquence T est fini p.s. et a tous ses moments finis.

Prouvons le. Soit $R = \sup \{|x|, x \in A\}$. Alors, pour tout x dans A , il existe un chemin de longueur $R+1$ partant de x et se terminant à l'extérieur de A , et donc :

$$\mathbb{P}^x(T \leq R+1) \geq \left(\frac{1}{2d}\right)^{R+1}.$$

Par la propriété de Markov, on a pour tout entier k :

$$\begin{aligned} \mathbb{P}^x(T > k(R+1)) &= \mathbb{P}^x(T > k(R+1) | T > (k-1)(R+1)) \mathbb{P}^x(T > (k-1)(R+1)) \\ &\leq \left(1 - \left(\frac{1}{2d}\right)^{R+1}\right) \mathbb{P}^x(T > (k-1)(R+1)), \end{aligned}$$

et donc, si $\rho := \left(1 - \left(\frac{1}{2d}\right)^{R+1}\right)^{1/(R+1)}$:

$$\mathbb{P}^x(T > k(R+1)) \leq \rho^{k(R+1)}.$$

Enfin, pour un entier n , il suffit d'écrire $n = k(R+1) + i$ avec $j \in \{1, \dots, R+1\}$, et on a alors :

$$\mathbb{P}^x(T \geq n) \leq \mathbb{P}^x(T > k(R+1)) \leq \rho^{k(R+1)} \leq \rho^{-(R+1)} \rho^n.$$

Il suffit donc de prendre $C := \rho^{-(R+1)}$. Cette démonstration se trouve dans [Law91, Lemme 1.4.4]. De façon générale, le théorème central limite permet de voir que le temps de sortie d'un ensemble fini (ou compact) est toujours fini p.s. pour une marche aléatoire dont l'accroissement a par exemple pour loi $p\delta_{+1} + q\delta_{-1} + r\delta_0$ (avec $0 \leq p, q, r < 1$) dans \mathbb{Z}^1 , cf. [CDGC80, exercice 8.8 page 247].

Exercice 5.2.1. Utilisez ce qui précède pour résoudre le problème de Dirichlet discret.

Exercices sur les temps d'arrêt

1. On considère $(X_n)_n$ des v.a indépendantes à valeurs dans $\{0, 1, -1\}$, telles que :

$$P(X_n = 1) = P(X_n = -1) = (1 - p_n)/2 \text{ et } P(X_n = 0) = p_n.$$

On suppose que $\sum_{n=1}^{+\infty} (1 - p_n) = +\infty$. Soit $S_n = X_1 + \dots + X_n$ et soit $T = \inf\{n \mid S_n \notin]-a, b[\}$, où $(a, b) \in \mathbf{N}^{*2}$. Vérifier expérimentalement que T est presque sûrement fini (prendre diverses valeurs de a et b). Montrer numériquement que $P(S_T = -a) = \frac{b}{a+b}$. Que se passe-t-il si $\sum_{n=1}^{+\infty} (1 - p_n) < +\infty$?

2. Soit (X_n) des v.a. i.i.d. suivant une loi de Bernoulli de paramètre p , et soit $S_n = X_1 + \dots + X_n$. Soit $m \in \mathbf{N}^*$ et soit $\tau = \inf\{n \mid S_n = m\}$. Vérifier que τ est fini presque sûrement, représenter la loi de $\tau - m$ (histogramme) et donner une approximation de $\mathbf{E}(\tau)$ pour différentes valeurs de m et de p . La loi de τ est appelée « loi binomiale négative BN(p,m) », et correspond au nombre d'essais indépendants nécessaires pour obtenir m succès lorsque la probabilité de réussite est p .
3. On considère des objets produits par une machine, dont chacun a une probabilité p d'être défectueux ($0 \leq p < 1$). On choisit 100 objets et on note $X_i = 1$ si l'objet i est défectueux, $X_i = 0$ sinon. Soit $\tau = \inf\{n \mid X_n = 1\}$. Vérifier que $\nu = \inf(\tau, 100)$ est fini, donner des approximations de $P(\tau \leq 100)$ et de $\mathbf{E}(\nu)$ (en fonction de p que vous ferez varier).
4. Soit (X_n) des v.a.i.i.d. suivant une loi de Poisson de paramètre $\lambda \leq 1$. Soit $Y_n = X_n - 1$ pour $n \in \mathbf{N}^*$ et soit $S_n = X_1 + \dots + X_n$. On note $T = \inf\{n \mid S_n = -1\}$. Vérifier que T est presque sûrement fini et donner une approximation de son espérance. Qu'en pensez-vous? Peut-on obtenir par simulation la loi de T ? Faire le même exercice avec pour loi des X_i une loi géométrique de paramètre $1/2$ et $Y_n = X_n - 2$. Conclusion?
5. Reprendre le modèle d'Ehrenfest. En notant n le nombre total de boules, et en supposant que l'on ait aucune boule initialement dans l'urne 1, donner une approximation de l'espérance du temps nécessaire pour avoir à nouveau aucune boule dans l'urne 1 (vérifier que cette espérance est 2^n).

5.2.2 Convergence des martingales

1. On considère (X_n) des v.a. i.i.d. suivant une loi normale centrée réduite. On pose $Y_i = X_i X_{i+1}$. Vérifier que les Y_i ne sont pas indépendantes. Vérifier que $M_n = \sum_{i=0}^n \frac{Y_i}{a_i}$ est une martingale, avec a_i suite de réels strictement positifs. Dans le cas où $a_n = n$, faire un programme permettant d'observer la convergence de M_n . Quelle est la distribution de la limite M_∞ . Recommencer avec $a_n = \sqrt{n}$. Qu'observe-t-on?

2. L'urne de Polya : une urne contient des boules blanches et noires. À l'étape 1, il y a une boule blanche et une boule noire. À l'étape n , on tire une boule au hasard, on la remet ensuite dans l'urne et on rajoute une boule de la même couleur. On appelle X_n le nombre de boules blanches et Y_n le nombre de boules noires à l'étape n . Montrer que $X_n + Y_n = n + 1$ et que $P(X_n = j) = 1/n$, $1 \leq j \leq n$. Simuler plusieurs expériences, et tracer sur un même graphique les différentes trajectoires (X_n) obtenues. Soit $M_n = \frac{X_n}{n+1}$. Montrer que M_n est une martingale convergente. Tracer sur un même graphique les différentes trajectoires de (M_n) . Quelle est la distribution de M_∞ ?
3. On considère (X_n) des v.a.i.i.d. suivant une loi normale centrée réduite et $S_n = X_1 + \dots + X_n$. On pose $Z_n^u = \exp(uS_n - \frac{nu^2}{2})$. Vérifier que Z_n^u est une martingale pour toute valeur de u . Pour différentes valeurs de u , faire un programme permettant d'observer la convergence de Z_n^u et la distribution de Z_∞^u . Déterminer $\mathbf{E}Z_n^u$, et en déduire que la convergence n'a lieu dans \mathbf{L}^1 que si $u = 0$. Est-ce observable numériquement ?
4. On considère (X_n) des v.a.i.i.d. suivant une loi normale centrée réduite et on note $Y_n = X_n^2$. Que vaut $\mathbf{E}Y_n$? Vérifier numériquement que $\mathbf{E}\sqrt{Y_n} < 1$. On pose $Z_n = \prod_{k=1}^n Y_k$. Montrer que (Z_n) est une martingale et que $(\sqrt{Z_n})$ est une surmartingale. Etudier la convergence et la limite de $(\sqrt{Z_n})$, puis de (Z_n) . A-t-on convergence dans \mathbf{L}^1 ?
5. Soit (X_n) des v.a.i.i.d. à valeurs dans $\{-1, 0, 1\}$, telles que $P(X_i = 1) = p$ et $P(X_i = -1) = q$ avec $q > p$. On note $S_n = X_1 + \dots + X_n$ ($S_0 = 0$) et $W = \sup_{n \geq 0} S_n$. Vérifier sur de nombreuses trajectoires que $W < +\infty$ p.s. Calculer explicitement $\psi(\lambda) = \log \mathbf{E}(e^{\lambda X_i})$. Résoudre numériquement l'équation $\psi(\lambda_0) = 0$ pour différentes valeurs de p et q . Vérifier numériquement que W est distribuée suivant une loi géométrique de paramètre $(1 - e^{\lambda_0})$ (pour des explications théoriques et des généralisations, voir [MPB98]).

Fragment 6

Files d'attente

On pourra consulter par exemple [Bon95], [Bou88], [DCD83] et [Ros00].

6.1 Notion de générateur infinitésimal

Soit $(N_t)_{t \geq 0}$ un processus de Poisson d'intensité λ . Si l'on pose $p_{t,m}(n) := \mathbb{P}(N_t = m \mid N_0 = n)$, un simple calcul fournit :

$$\begin{aligned} \frac{d}{dt} p_{t,m}(n) &= \lambda(p_{t,m}(n+1) - p_{t,m}(n)) \\ &= (\mathbf{L}p_{t,m})(n) \end{aligned}$$

où la matrice infinie \mathbf{L} est donnée pour tout $(i, j) \in \mathbb{N}^2$ par

$$\mathbf{L}_{i,j} := \begin{cases} -\lambda & \text{si } j = i \\ \lambda & \text{si } j = i + 1 \\ 0 & \text{sinon} \end{cases}$$

La matrice \mathbf{L} joue le même rôle que la matrice de transition des chaînes de Markov, et l'on parle de générateur infinitésimal du processus $(N_t)_{t \geq 0}$. Sa donnée caractérise la loi du processus de Markov $(N_t)_{t \geq 0}$ et l'on a $p_{t,m}(n) = (\exp(t\mathbf{L}))_{m,n}$. Si N_0 a pour loi σ , vue comme un vecteur colonne de longueur infinie, alors N_t a pour loi $\exp(t\mathbf{L})\sigma$ où le produit est « matriciel ».

Réciproquement, donnons nous une matrice infinie $(\mathbf{L}_{i,j}, (i, j) \in \mathbb{N}^2)$, telle que $\mathbf{L}_{i,j} \geq 0$ si $i \neq j$ et dont la somme de chaque ligne est nulle. Alors elle constitue le générateur infinitésimal d'un processus de Markov $(N_t)_{t \geq 0}$ sur \mathbb{N} , qui reste en $i \in \mathbb{N}$ un temps exponentiel de paramètre

$$\sum_{j \neq i} \mathbf{L}_{i,j},$$

puis saute en $j \neq i$ avec probabilité

$$\frac{\mathbf{L}_{i,j}}{\sum_{j \neq i} \mathbf{L}_{i,j}}.$$

Un cas particulier important est constitué par les processus de *vie et de mort* :

$$\mathbf{L}_{i,j} := \begin{cases} -\lambda_i - \mu_j & \text{si } j = i \text{ (attente)} \\ \lambda_i & \text{si } j = i + 1 \text{ (vie)} \\ \mu_i & \text{si } j = i - 1 \text{ (mort)} \\ 0 & \text{sinon} \end{cases}$$

où $(\lambda_n)_{n \in \mathbb{N}}$ et $(\mu_n)_{n \in \mathbb{N}}$ sont deux suites de réels strictement positifs avec $\mu_0 = 0$. Ces processus ne peuvent sauter que de +1 ou -1. De tels processus sont très faciles à simuler. Ils modélisent bien un certain type de files d'attente : les incréments correspondant à l'arrivée poissonnienne des clients et les décréments aux résultats du travail des serveurs qui réduisent la file d'attente. La dénomination

« vie ou mort » provient de l'interprétation en terme de population de particules qui augmente ou diminue d'une unité. Le processus de Poisson d'intensité λ correspond à $\lambda_n = \lambda$ et $\mu_n = 0$ pour tout $n \in \mathbb{N}$.

Pour certaines suites $(\lambda_n)_{n \in \mathbb{N}}$ et $(\mu_n)_{n \in \mathbb{N}}$, la loi du processus à l'instant t converge vers une loi limite sur \mathbb{N} lorsque t est grand, qui correspond à un état d'équilibre. Formellement, cette loi stationnaire est solution de l'équation $\mathbf{L}\sigma = 0$. Pour le processus de Poisson, les mesures stationnaires sont les mesures constantes sur \mathbb{N} , qui ne sont pas des probabilités. En revanche, nous allons voir que certains processus de vie ou de mort admettent des lois de probabilités stationnaires, et l'on comprend que le rapport entre les coefficients de vie λ_n et de mort μ_n va jouer un rôle.

6.2 File d'attente M/M/1

En anglais, on parle de queues. Les files d'attente apparaissent naturellement dans beaucoup de situations : un guichet déservant des usagers, une piste d'aéroport sur laquelle les avions atterissent, un serveur informatique répondant à des requêtes, etc. Les clients arrivent à des instants aléatoires et attendent leur tour devant le guichet (serveur). Le serveur met un temps aléatoire pour servir chaque client. On suppose ici que les clients sont servis dans leur ordre d'arrivée (on parle alors de file FIFO : « First In First Out »).

On suppose que les instants d'arrivée des clients correspondent à un processus de Poisson d'intensité λ et que les instants de fin des services correspondent à un autre processus de Poisson, indépendant du premier, et de paramètre μ . Ainsi, si A_t désigne le nombre d'arrivées de clients dans l'intervalle $[0, t]$, alors $(A_t)_{t \geq 0}$ est un processus de Poisson et A_t/t tend presque sûrement vers λ .

On s'intéresse par exemple au nombre de clients N_t à l'instant t dans la file d'attente, en comptant l'éventuel client qui est en train d'être servi, et on suppose que $N_0 = 0$. Le processus $(N_t)_{t \geq 0}$ est un processus de Markov qui appartient à la classe des processus de « vie ou de mort » : il s'agit d'un processus à saut, dont les sauts ne peuvent valoir que $+1$ ou -1 , et dont les durées inter-sauts suivent des lois exponentielles. Ce processus est plus complexe que le processus de Poisson car les temps inter-sauts et la loi des sauts dépendent de la valeur du processus. À ce stade, il est important d'observer que le minimum de deux v.a.r. de loi exponentielles de paramètres respectifs λ et μ suit encore une loi exponentielle de paramètre $\lambda + \mu$ et que la loi exponentielle est sans mémoire. On peut alors simuler la valeur de N_t de la manière suivante :

$$N_t(\omega) = \sum_{n=0}^{+\infty} X_n(\omega) I_{[T_n(\omega), T_{n+1}(\omega)]}(t),$$

où $X_0 = 0$, $T_0 = 0$ et pour $n > 0$:

$$\mathcal{L}(T_{n+1} - T_n | X_n) = \begin{cases} \mathcal{E}(\lambda) & \text{si } X_n = 0 \\ \mathcal{E}(\lambda + \mu) & \text{si } X_n > 0 \end{cases}$$

et

$$\mathcal{L}(X_{n+1} - X_n | X_n) = \begin{cases} \delta_{+1} & \text{si } X_n = 0 \\ \frac{\mu}{\lambda + \mu} \delta_{-1} + \frac{\lambda}{\lambda + \mu} \delta_{+1} & \text{si } X_n > 0 \end{cases}$$

Sur chaque intervalle de temps $[T_n, T_{n+1}[$, N_t vaut X_n et chaque instant de saut T_n correspond à l'arrivée d'un nouveau client ou à la fin d'un service (départ d'un client). Voici un code Matlab illustrant ce procédé :

```

1 % File d'attente M/M/1
2 % Les clients arrivent avec un temps expo de paramètre l
3 % Le serveur sert avec un temps expo de paramètre m
4 % N(t) = nombre de clients dans la file au temps t
5 % y compris ceux qui sont en train d'être servis
6
7 clear; clf;
8
9 l=1.5; % intensité d'arrivée des clients
10 m=1.6; % intensité du traitement par le serveur
11
12 t=2; % on cherche la longueur de la file à cet instant

```

```

13 r=1000; % nombre de réalisations i.i.d.
14
15 for i=1:r,
16     n=0; % la file est vide au depart
17     T=-log(rand)/l; % on attend le premier client
18     while (T<t) % on boucle tant que le temps est < t
19         % il s'est passe quelque chose
20         if n==0 % la file etait vide
21             n=1; % un client de plus
22         else % la file n'etait pas vide
23             % un client de plus avec proba l/(l+m)
24             % un client de moins avec proba m/(l+m)
25             n=n+2*(rand<l/(l+m))-1;
26         end
27         % on attend le prochain evenement
28         % on tient compte du fait que la file est vide ou pas
29         T=T-log(rand)/(l+m*(n>0));
30     end
31     N(i)=n;
32 end
33 histo(N)
34 [mean(N),std(N)]

```

Lorsque $\rho := \lambda/\mu < 1$, N_t converge en loi vers la loi géométrique de paramètre ρ lorsque t tend vers $+\infty$. La file d'attente atteint alors un équilibre asymptotique, et la moyenne du nombre de clients dans la file lorsque t est grand devient proche de la moyenne $\rho/(1-\rho)$ de la loi géométrique limite. La probabilité que la file d'attente dépasse le seuil K à l'équilibre est alors de ρ^K . Le paramètre ρ représente la *charge du système*. Lorsque $\rho > 1$, les clients arrivent trop vite pour qu'un équilibre s'installe. Si W_n représente le temps de séjour dans la file du n -ième client, on montre que lorsque $\rho < 1$, les quantités suivantes sont :

$$\lambda \mathbb{E} \left(\lim_{n \rightarrow +\infty} \frac{W_1 + \dots + W_n}{n} \right) = \mathbb{E} \left(\lim_{t \rightarrow +\infty} \frac{1}{t} \int_0^t N_s ds \right) = \frac{\lambda}{\mu - \lambda}.$$

En fait, on peut même montrer que $T := \lim_{n \rightarrow +\infty} (W_1 + \dots + W_n)/n$ suit une loi exponentielle de paramètre $\mu - \lambda$. La variable aléatoire T représente le *temps de réponse du système* à l'équilibre.

Exercice 6.2.1. En utilisant le programme Matlab précédent, vérifier que tout ceci a bien lieu par des simulations. Qu'elle est la variance asymptotique de N_t lorsque t tend vers $+\infty$? Que se passe-t-il lorsque $\rho = 1$? Que représente la quantité $\log \varepsilon / \log \rho$ pour $\varepsilon \in]0, 1[$?

La dénomination M/M/1 correspond à un cas particulier de la nomenclature suivante utilisée pour les files d'attente : **A/B/s/K/n/d** où

- **A** indique la loi des temps d'arrivée des clients
- **B** indique la loi des temps de service
- **s** indique le nombre de serveurs
- **K** indique la capacité maximale de la salle d'attente ($+\infty$ si omis)

Voici quelques codes utilisés pour **A** et **B** :

- M pour des v.a. i.i.d. de loi exponentielles (« memoryless »)
- U pour des v.a. i.i.d. de loi uniforme
- D pour des temps constants (déterministes)
- GI pour des v.a. i.i.d. de même loi
- G pour des loi quelconques

Exercice 6.2.2 (File d'attente M/M/s). Simulez une file M/M/s (i.e. avec s serveurs). On pourra remarquer que dans ce cas, $\lambda_n = \lambda$ et $\mu_n = n\mu$ si $n \leq s$ et $\mu_n = ns$ si $n = s$. La condition de stationnarité est donnée par $\rho := \lambda/s\mu < 1$. Qu'elle est la loi stationnaire?

Exercice 6.2.3 (File d'attente M/M/1/K). Simulez une file M/M/1/K (i.e. un seul serveur et une salle d'attente de capacité maximale K). Dans ce cas, $\lambda_n = \lambda \mathbb{1}_{0 \leq n < K}$ et $\mu_n = \mu \mathbb{1}_{1 \leq n \leq K}$. La condition de stationnarité est donnée par $\rho := \lambda/\mu < 1$. Vérifiez que la loi stationnaire est donnée par

$$\frac{1 - \rho}{1 - \rho^K} \sum_{n=0}^K \rho^n \delta_n.$$

Exercice 6.2.4 (File d'attente M/M/∞). Simulez une file M/M/∞ (i.e. une infinité de serveurs). Dans ce cas, $\lambda_n = \lambda$ et $\mu_n = n\mu$. Vérifiez que la loi stationnaire est la loi de Poisson de paramètre $\rho := \lambda/\mu$.

Exercice 6.2.5 (File d'attente avec impatience). Dans ce cas, chaque nouveau client décide d'entrer dans la file avec une probabilité r_n où n désigne le nombre de clients déjà présents dans la file. Simulez une telle file (on a $\lambda_n = r_n\lambda$ et $\mu_n = \mu$).

6.3 Exercices

1. Simuler un processus de Poisson à l'aide d'exponentielles de paramètre λ . Tracer la trajectoire de ce processus. Pour une valeur de t quelconque, donner N_t . Déterminer un estimateur de λ et donner un intervalle de confiance sur la valeur obtenue pour une trajectoire.
2. On considère une machine qui peut tomber en panne, mais qui est réparée instantanément. Par exemple, une machine avec une pile, une lampe avec une ampoule, etc. On suppose que la durée de vie moyenne entre chaque panne est de m . On note par N le processus de renouvellement correspondant au nombre de pannes, et par T_k la durée de vie après la panne k (possédant un moment d'ordre 2). On choisit pour loi des T_k une loi exponentielle de paramètre 0.5 (unité de temps = le mois?). Représenter N_t , puis N_t/t pour $t \in [0, 100]$.
3. Simuler un processus de renouvellement pour lequel T_1 a une loi uniforme et les durées d'arrivées une loi de Weibull. Vérifier que N_t/t tend vers une constante. Quelle est la loi asymptotique de N_t/t ?
4. On suppose que (T_n) est un processus de renouvellement avec $T_n = \sum_{k=1}^n \tau_k$, les v.a.i.i.d. τ_k étant distribuées suivant une loi sur \mathbb{R}^+ possédant un moment d'ordre 2. On considère ensuite un processus de Poisson $N := (N_t \geq 0)$ d'intensité θ , indépendant du processus T_n . On pose $X_n := N_{T_n}$ pour $n \in \mathbb{N}$. Construire T_n , N , et en déduire X_n (choisir différentes lois pour τ_k). Vérifier que la suite $(X_n/n)_n$ possède une limite lorsque n tend vers $+\infty$ et qu'il existe a_n et b_n tels que $(X_n - a_n)/b_n$ converge vers une loi $\mathcal{N}(0, 1)$.
5. On considère une machine qui peut tomber en panne pendant un certain temps. Soient $(X_n)_n$ (resp. $(Y_n)_n$) des v.a.i.i.d. à valeurs dans \mathbb{R}^+ , de loi de densité f_X (resp. f_Y), d'espérance m_X finie (resp. m_Y finie). Les $(X_n)_n$ sont les durées de bon fonctionnement et sont indépendantes des $(Y_n)_n$, qui sont les durées de pannes successives. On pose $T_0 := 0$ et pour $n \geq 1$, $S_n := T_{n-1} + X_n$ et $T_n := S_n + Y_n$. Ainsi, les intervalles $[T_n, S_{n+1}[$ sont les intervalles de bon fonctionnement et les $[S_n, T_n[$ les intervalles de panne.
 - (a) On prend pour f_X une densité de Weibull $(5, 1)$ et pour f_Y une densité exponentielle de paramètre 3. Représenter $(T_n)_n$ dont on montre que c'est un processus de renouvellement.
 - (b) Montrer que :

$$\lim_{t \rightarrow +\infty} \mathbb{P} \left(t \in \bigcup_{n \in \mathbb{N}} [T_n, S_{n+1}[\right) = \frac{m_X}{m_X + m_Y}.$$

6. On considère un processus de Poisson d'intensité λ . À chaque saut du processus, on note A le saut avec une probabilité p et B avec une probabilité $1 - p$. Vérifier que les processus de comptage des A et des B , notés N^A et N^B , sont des processus de Poisson. Quelles sont leurs intensités? Sont-ils indépendants? Le vérifier à l'aide d'un test du χ^2 .

Fragment 7

Évolution de la taille d'une population lorsque la loi de reproduction est homogène

On pourra consulter par exemple [Har63], [Har69] et [Wil91] pour les preuves des résultats énoncés. On se propose d'étudier l'évolution d'une population de taille Z_n à la génération n , modélisée par le processus suivant, appelé *processus de branchement* ou processus de Galton-Watson :

– $Z_0 = 1$.

– Chaque individu a pour probabilité p_k d'avoir k descendants, avec $k \in \mathbb{N}$.

En supposant que les enfantements se font de manière indépendante, on a la formule de récurrence suivante pour $n \in \mathbb{N}^*$:

$$Z_{n+1} = \sum_{i=1}^{Z_n} X_{n,i},$$

où les variables aléatoires $(X_{n,i})_{n,i \in \mathbb{N}}$ sont i.i.d. et de loi

$$P := \sum_{k=0}^{+\infty} p_k \delta_k.$$

Cette loi de probabilité discrète sur \mathbb{N} est appelée *loi de fécondité* du processus. On note m son espérance et σ sa variance. On supposera toujours pour simplifier que :

– La loi P a tout ses moments finis (donc $m < \infty$ et $\sigma < \infty$)

– Pour tout k , $p_k \neq 1$, et $p_0 + p_1 < 1$

On vérifie facilement que $\mathbb{E}(Z_n) = m^n$ et

$$\text{Var}(Z_n) = \begin{cases} n\sigma^2 & \text{si } m = 1 \\ \sigma^2 m^{n-1} (m^n - 1)(m - 1)^{-1} & \text{si } m \neq 1. \end{cases}$$

D'autre part, si f_n est la fonction génératrice de Z_n , on vérifie facilement que

$$f_n = \underbrace{f \circ \dots \circ f}_{n \text{ fois}},$$

où f est la fonction génératrice de la loi P . Les hypothèses faites sur P entraînent alors que f est strictement convexe sur $[0, 1]$ et que $f'(1)$ est finie.

On s'intéresse naturellement à la *probabilité d'extinction* de la population modélisée par $(Z_n)_{n \in \mathbb{N}}$. La suite $\{Z_n = 0\}$ est croissante, et la probabilité d'extinction q est donnée par :

$$q := \mathbb{P}(\exists n, Z_n = 0) = \lim_{n \rightarrow +\infty} \mathbb{P}(Z_n = 0) = \lim_{n \rightarrow +\infty} f(0),$$

On peut alors montrer que $q = 1$ si $m \leq 1$ et que si $m > 1$, q est point fixe de f . En fait, la suite $(Z_n)_{n \in \mathbb{N}}$ est instable quel que soit la valeur de m :

$$\forall k > 0, \lim_{n \rightarrow +\infty} \mathbb{P}(Z_n = k) = 0,$$

et la suite $(Z_n)_{n \in \mathbb{N}}$ converge vers 0 avec probabilité q et vers $+\infty$ avec probabilité $1 - q$. Le processus $(Z_n)_{n \in \mathbb{N}}$ est une chaîne de Markov de probabilité de transition

$$\mathcal{L}(Z_{n+1} | Z_n = k) = P^{*k}.$$

D'autre part, on vérifie facilement que $(Z_n/m^n)_{n \in \mathbb{N}}$ est une martingale, bornée dans \mathbf{L}^2 (car $\sigma < +\infty$).

Suivant la valeur de m , trois comportements asymptotiques distincts pour $(Z_n)_{n \in \mathbb{N}}$ vont se présenter :

1. Si $m < 1$, on est dans le cas *sous-critique* et la population va s'éteindre presque sûrement (i.e. Z_n tend vers 0 p.s.). De plus la loi conditionnelle de Z_n sachant que $Z_n > 0$ admet une loi limite $B = b_1 \delta_1 + b_2 \delta_2 + \dots$:

$$\forall k \in \mathbb{N}^*, \lim_{n \rightarrow +\infty} \mathbb{P}(Z_n = k | Z_n > 0) = b_k,$$

et la fonction génératrice g de B vérifie $g \circ f = mg + 1 - m$.

2. Si $m > 1$, on est dans le cas *sur-critique* et la probabilité d'extinction q est l'unique point fixe de f . De plus, $m^{-n} Z_n$ converge presque-sûrement et dans \mathbf{L}^2 vers la variable aléatoire W , de moyenne 1 et de variance $\sigma^2/(m^2 - m)$, $\mathbb{P}(W = 0) = q$ et la transformée de Laplace $\varphi(s) := \mathbb{E}(\exp(-sW))$ de W vérifie

$$\forall s \geq 0, \varphi(ms) = f(\varphi(s)).$$

3. Si $m = 1$, on est dans le cas *critique*, et Z_n est une martingale qui tend presque sûrement vers 0 (elle ne converge donc pas en moyenne quadratique). On a de plus :

$$n \mathbb{P}(Z_n > 0) \xrightarrow{n \rightarrow +\infty} \frac{2}{\sigma^2} \quad \text{et} \quad \frac{1}{n} \mathbb{E}(Z_n | Z_n > 0) \xrightarrow{n \rightarrow +\infty} \frac{\sigma^2}{2}$$

et enfin, pour tout $u \geq 0$

$$\mathbb{P}\left(\frac{Z_n}{n} \geq u | Z_n > 0\right) \xrightarrow{n \rightarrow +\infty} \exp\left(-\frac{2u}{\sigma^2}\right).$$

Pour $P = p_0 \delta_0 + p_1 \delta_1 + p_2 \delta_2$, on a $m = p_1 + 2 * p_2$, $f(s) = p_0 + p_1 s + p_2 s^2$ et $f(s) = s$ si et seulement si $s \in \{1, p_0/p_2\}$, d'où $q = p_0/p_2$.

```

1  %% Processus de Galton-Watson avec loi de fécondité portée par {0,...,n}.
2
3  clear
4  p=[1/4,1/4,1/2]; % Loi de reproduction sur {0,1,2,...}
5  ng=20;          % Nombre de générations
6  nr=100;        % Nombre de réalisations
7
8  n=length(p)-1; % Nombre maximal d'enfants possibles
9  pr=[0,cumsum(p)];
10 m=sum([0:n].*p);
11 si=sqrt(sum([0:n]^2.*p)-m^2)
12 sprintf('Moyenne_m_=%f\n',m)
13 sprintf('Écart_type_sigma_=%f\n',si)
14
15 Z=zeros(nr,ng+1);
16 Z(1:nr,1)=1;
17
18 for i=1:nr % réalisations
19     for j=1:ng % générations
20         U=rand(1,Z(i,j));
21         X=zeros(1,Z(i,j));
22         for k=1:Z(i,j) % simulation des enfantements de la génération j
23             for l=1:n+1 % Z(i,j) réalisations i.i.d. de loi P
24                 if ( (U(k)>=pr(l)) & (U(k)<pr(l+1)) )
25                     X(k)=l-1;
26                     break
27                 end
28             end % for l
29         end % for k
30         Z(i,j+1)=sum(X);

```

```

31     if Z(i,j+1)==0
32         break % On sort de la boucle for j si la population s'éteint
33     end
34 end % for j
35 end % for i

```

Pour $P = p \sum_{k=0}^{\infty} (1-p)^k \delta_k$, on a $m = (1-p)/p$, $f(s) = p/(1 - (1-p)s)$ et $f(s) = s$ si et seulement si $s \in \{1, 1/m\}$, d'où $q = p/(1-p)$.

```

1  %% Processus de Galton-Watson avec loi de fécondité géométrique
2  %
3  clear
4  p=9/20; % Parametre de la loi de fécondité géométrique
5      % p*(de(0)+q*de(1)+q^2*de(2)+...)
6  ng=30; % Nombre de générations
7  nr=100; % Nombre de réalisations
8  %
9  m=(1-p)/p;
10 si=sqrt(1-p)/p;
11 sprintf('Moyenne_m_=_%f\n',m)
12 sprintf('Écart_type_sigma_=_%f\n',si)
13 %
14 Z=zeros(nr,ng+1);
15 Z(1:nr,1)=1;
16 %
17 for i=1:nr % réalisations
18     for j=1:ng % générations
19         Z(i,j+1)=sum(rgeom([1,Z(i,j)],p));
20         if Z(i,j+1)==0
21             break % On sort de la boucle for j si la population s'éteint
22         end
23     end % for j
24 end % for i

```

```

1  %% Processus de Galton-Watson : tracé de la loi empirique de Z_n/m^n
2
3  clf
4  hold on
5  histo(Z(1:nr,ng+1)/m^ng)

```

```

1  %% Processus de Galton-Watson avec loi de fécondité géométrique
2  %% Probabilité d'extinction empirique
3
4  clf; plot([0:ng],sum(Z==0)/nr)

```

Exercice 7.0.1. Dans le cas d'une loi de reproduction géométrique de paramètre p , calculez la probabilité d'extinction en fonction de p . Choisir p de façon à obtenir les cas critique, sur-critique et sous-critique. Tracez des histogrammes et des trajectoires simulées dans chacun de ces cas et confrontez vos résultats aux comportements prévus par la théorie.

On trouvera dans [Rug00] deux chapitres consacrés à des modèles de Galton-Watson. Le modèle avec immigration est très simple à simuler puisqu'il suffit d'ajouter à Z_n une réalisation d'une loi fixée à l'avance (par exemple une loi de Poisson) qui modélise le flux migratoire à la n -ième génération.

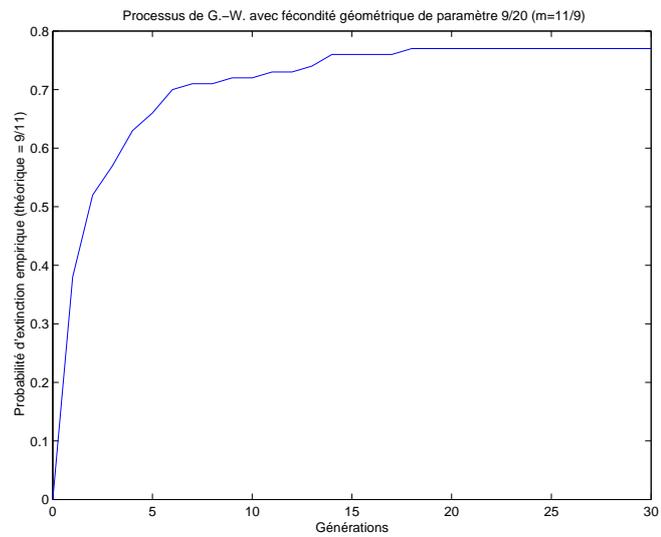


FIG. 7.1 – Probabilité d'extinction empirique

Fragment 8

Quelques lois classiques

On renvoie principalement à [Bor95], [Rud75], [Bou86] et [DCD82b]. Dans toute la suite, on note $\mathcal{L}(X)$ la loi de la v.a. X et $*$ la convolution des fonctions ou des lois de probabilité, définies par :

$$(f * g)(x) := \int f(x-y)g(y) dy = \int f(z)g(x-z) dz = (g * f)(x),$$

et

$$\mathbf{E}_{\mu * \nu}(\varphi) := \int \varphi(x+y) d(\mu \otimes \nu)(x,y) = \int \varphi(x+y) d(\nu \otimes \mu)(x,y) = \mathbf{E}_{\nu * \mu}(\varphi).$$

Si μ et ν ont pour densités respectives f et g par rapport à la mesure de Lebesgue, alors $\mu * \nu$ a pour densité $f * g$. On rappelle que la convolée $\mu * \nu$ de deux mesures de probabilité représente la loi de la somme de deux v.a. indépendantes X, Y de loi respectives μ et ν :

$$X, Y \text{ indépendantes} \Rightarrow \mathcal{L}(X + Y) = \mathcal{L}(X) * \mathcal{L}(Y).$$

La réciproque est fautive en général. On note μ^{*n} la loi obtenue en convolant n fois μ :

$$\mu^{*n} := \underbrace{\mu * \dots * \mu}_{n \text{ fois}}.$$

D'autre part :

$$X, Y \text{ indépendantes} \Leftrightarrow \mathcal{L}((X, Y)) = \mathcal{L}(X) \otimes \mathcal{L}(Y).$$

On note δ_a la masse de Dirac en a , c'est à dire la mesure de probabilité qui affecte la masse 1 aux ensembles mesurables contenant le point a et 0 aux autres. C'est la loi d'une v.a. constante de valeur a . Sa moyenne vaut a et sa variance est nulle. Pour tout μ et ν , on a $\mu * \nu = \nu * \mu$ et $\mu * \delta_0 = \delta_0 * \mu = \mu$ et $\delta_a * \delta_b = \delta_{a+b}$. La masse de Dirac en 0 est donc l'élément neutre de la convolution.

Heuristique 8.0.2 (Convolution = lissage). Convoler une fonction f par une fonction g revient à prendre en x une moyenne des valeurs de f pondérées par la translation de g en x . Il est donc naturel que la masse de Dirac en 0 soit l'élément neutre de la convolution. Si g est lisse, on fait donc du lissage de f . Pour ce faire, on prend souvent pour g une fonction \mathcal{C}^∞ paire et à décroissance rapide voire même à support compact de façon à remplacer $f(x)$ par une « g -moyenne » de f au voisinage de x . Penser à la technique de régularisation par convolution utilisée en analyse, en particulier pour montrer la densité de \mathcal{C}_c^∞ dans les espaces \mathbf{L}^p de Lebesgue.

Heuristique 8.0.3 (Convolution des densités = addition d'un bruit indépendant sur la v.a.). Soit X et Y des v.a. de densités respectives f et g . Dire que g est lisse signifie que la v.a. Y est purement continue, c'est à dire que les valeurs prises sont bien réparties et sans sauts. Ainsi, si X est indépendante de Y , la v.a. $X + Y$ aura en général des valeurs mieux réparties que celles de X lorsque g est lisse. La densité de $X + Y$ est la convolée de la densité f de la v.a. X par la densité g de la v.a. Y . Si g est lisse, il en sera de même pour $f * g$. Ainsi, lisser une densité se traduit sur les v.a. par l'addition d'une v.a. indépendante à densité lisse (penser par exemple au lissage d'une v.a. discrète X par addition d'une v.a. indépendante Y de loi uniforme ou gaussienne, centrée et de faible variance par rapport à l'écart entre les valeurs discrètes prises par X).

Remarque 8.0.4 (Fonctions de répartition et densité). Considérons une loi de probabilité μ sur \mathbb{R} , de fonction de répartition F . En tant que fonction de répartition, la fonction F est croissante, continue à droite, et $F(\mathbb{R}) = [0, 1]$. On montre alors qu'elle ne peut avoir qu'un nombre au plus infini dénombrable de points de discontinuité, notés $(s_n)_n \subset \mathbb{R}$, cf. [BL98, prop. III.2.2 page 48]. On montre de plus, [BL98, III, pages 49-50], qu'il existe trois réels positifs ou nuls α, β, γ vérifiant $\alpha + \beta + \gamma = 1$, tels que F possède la décomposition suivante :

$$F = \alpha F_{\text{disc}} + \beta F_{\text{sing}} + \gamma F_{\text{cont}},$$

où les fonctions F_\bullet , qui vérifient les propriétés suivantes, n'existent que lorsque le coefficient associé est non nul :

1. F_{disc} est constante par morceaux, et c'est la fonction de répartition de la loi de probabilité discrète $\sum_n \delta_{s_n}$
2. F_{cont} est absolument continue et c'est la fonction de répartition d'une loi de probabilité à densité par rapport à la mesure de Lebesgue sur \mathbb{R}
3. F_{sing} est continue et c'est la fonction de répartition d'une mesure de probabilité singulière par rapport à la mesure de Lebesgue sur \mathbb{R}

F est continue si et seulement si $\alpha = 0$ et μ admet une densité si et seulement si $\alpha = \beta = 0$. Il ne suffit donc pas que F soit continue pour que μ admette une densité par rapport à la mesure de Lebesgue. En d'autres termes, la composante étrangère à la mesure de Lebesgue de μ dans sa décomposition de Lebesgue-Radon-Nikodym (cf. [Rud75, thm 6.9, page 117]) ne se réduit pas aux sauts de la fonction F . Un exemple simple est fourni par les lois fractales de la section 1.5.1 page 21. Bien entendu, si F est C^1 , alors μ admet une densité qui n'est rien d'autre que F' .

Remarque 8.0.5 (Densité au sens de la théorie des distributions). Une loi de probabilité est, en tant que mesure de Radon, une distribution. Elle s'identifie à sa densité lorsque cette dernière existe. La fonction de répartition F_X d'une v.a.r. X existe toujours. Sa dérivée F'_X au sens des distributions a donc toujours un sens. Lorsque la loi $\mathcal{L}(X)$ admet une densité f , cette dernière n'est rien d'autre que F'_X . Les points de discontinuité de F_X se traduisent sur F'_X par des masses de Dirac au sens des distributions, tandis que les « plats » du graphe de F_X correspondent à des intervalles qui ne sont pas portés par $\mathcal{L}(X)$. Il ne suffit pas que F_X soit continue pour que $\mathcal{L}(X)$ admette une densité, i.e. que F'_X soit une fonction en tant que distribution. Les F_X pour lesquelles c'est le cas sont dites « absolument continues » et s'écrivent comme l'intégrale d'une fonction \mathbf{L}^1 (qui est alors la densité associée). Une condition suffisante d'absolue continuité est que F_X soit C^1 .

Remarque 8.0.6. Les générateurs aléatoires de la bibliothèque Stixbox pour Matlab sont listés dans la table 1.1 page 38. Le tableau 8.1 page 98 recense quelques lois usuelles. Pour les démonstrations et des extensions, on renvoie principalement à [DCD82b] et [Bou86].

8.1 Fonctions génératrices – la transformée du pauvre

Si X est une variable aléatoire entière (i.e. à valeurs dans $\bar{\mathbb{N}} := \mathbb{N} \cup \{+\infty\}$), on définit sa fonction génératrice par :

$$G_X : [0, 1[\rightarrow \mathbb{R}$$

$$s \mapsto G_X(s) := \mathbf{E}(s^X) = \sum_{k=0}^{+\infty} s^k \mathbb{P}(X = k).$$

Elle ne dépend que de la loi de X , qu'elle caractérise. En d'autres termes, si X et Y sont deux v.a. entières, alors $\mathcal{L}(X) = \mathcal{L}(Y)$ si et seulement si $G_X = G_Y$. De plus, $G_{X+Y} = G_X G_Y$ lorsque X et Y sont indépendantes (la réciproque est fautive). On peut définir la fonction génératrice du couple de v.a. entières (X, Y) en $(s_1, s_2) \in [0, 1]^2$ par : $G_{(X,Y)}(s_1, s_2) := \mathbf{E}(s_1^X s_2^Y)$. Les v.a. X et Y sont alors indépendantes si et seulement si $G_{(X,Y)}(s_1, s_2) = G_X(s_1)G_Y(s_2)$ pour tout $(s_1, s_2) \in [0, 1]^2$.

La fonction génératrice permet également de calculer les moments puisque $\mathbb{P}(X = n) = \frac{1}{n!} G_X^n(0)$ et $G'_X(1) = \mathbf{E}(X)$ et plus généralement : $G_X^{(n)}(1) = \mathbf{E}(X(X-1)\cdots(X-n+1))$.

8.2 Lois discrètes à support sur entiers naturels

On rappelle que si X et Y sont deux v.a. entières (i.e. à valeurs dans $\bar{\mathbb{N}}$) indépendantes alors :

$$\mathbb{P}(X + Y = n) = \sum_{k_1+k_2=n} \mathbb{P}(X = k_1)\mathbb{P}(Y = k_2).$$

En terme de lois, cela donne :

$$(\mu * \nu) = \sum_{k_1+k_2=n} \mu(\{k_1\})\nu(\{k_2\})\delta_n.$$

8.2.1 Loi uniforme discrète

C'est la loi donnée par : $(\delta_{a_1} + \dots + \delta_{a_n})/n$, où les a_i sont tous différents. Elle a pour moyenne $(a_1 + \dots + a_n)/n$ et pour variance $(n\|a\|_2^2/n - (a_1 + \dots + a_n)^2)/n^2$. On l'utilise pour modéliser un phénomène pouvant prendre n valeurs différentes a_1, \dots, a_n avec la même probabilité $1/n$. Elle a pour propriété remarquable de maximiser, à n fixé, l'entropie de Shannon $\mathbf{H}(p_1, \dots, p_n)$ définie pour une loi de probabilité $p_1\delta_{a_1} + \dots + p_n\delta_{a_n}$ par :

$$\mathbf{H}(p_1, \dots, p_n) := - \sum_{k=1}^n p_i \log p_i.$$

8.2.2 Loi de Bernoulli

C'est la loi $q\delta_a + p\delta_b$ où $p \in [0, 1]$, $q := 1 - p$ et $a \neq b$. Le plus souvent, on utilise $q\delta_{-1} + p\delta_1$ ou $q\delta_0 + p\delta_1$. La loi $q\delta_0 + p\delta_1$, notée $\mathcal{B}(1, p)$, est de moyenne p et de variance pq . Elle permet de modéliser le résultat d'une expérience à deux issues possibles, comme par exemple un jet de pièce dans un jeu de pile ou face, avec probabilité p de gagner (codé 1) et probabilité $1 - p$ de perdre (codé 0). Elle a pour fonction génératrice $q + ps$.

8.2.3 Loi binomiale

Notée $\mathcal{B}(n, p)$, c'est la loi :

$$\sum_{k=0}^n C_n^k p^k (1-p)^{n-k} \delta_k = (q\delta_0 + p\delta_1)^{*n}.$$

Elle permet de modéliser le nombre de gains pour n lancers au jeu de pile ou face avec probabilité de gain de p . Si X_1, \dots, X_n sont i.i.d. de loi de Bernoulli $q\delta_0 + p\delta_1$, alors $X_1 + \dots + X_n$ suit la loi $\mathcal{B}(n, p)$. La probabilité de gagner k fois en n lancers vaut $C_n^k p^k (1-p)^{n-k}$. Sa moyenne est np et sa variance npq . Elle a pour fonction génératrice :

$$G_{\mathcal{B}(n,p)}(s) := \mathbf{E}_{\mathcal{B}(n,p)}(s^\bullet) = (q + ps)^n.$$

La loi gaussienne peut être obtenue par passage à la limite, cf. théorème 2.2.2, page 42.

8.2.4 Loi géométrique

La loi géométrique $\mathcal{G}(p)$ de paramètre $p \in [0, 1]$, ou loi de Pascal, correspond à la loi du premier succès à un jeu de pile ou face avec probabilité de gain p . Si $(X_n, n \in \mathbb{N}^*)$ est une suite de v.a. i.i.d. de loi de Bernoulli $q\delta_0 + p\delta_1$, la loi géométrique de paramètre p est la loi de la v.a. « premier succès » définie par :

$$T(p) := \inf \{n \in \mathbb{N}^*, X_n = 1\}.$$

Elle est donnée par :

$$\mathcal{G}(p) := pq^{-1} \sum_{k=1}^{+\infty} q^k \delta_k.$$

Sa moyenne vaut $1/p$, sa variance q/p^2 et sa fonction génératrice est $sp/(1-sq)$. On préfère parfois numéroter les lancers de pile ou face à partir de 0, on a alors la loi géométrique de base 0 :

$$p \sum_{k=0}^{+\infty} q^k \delta_k,$$

qui a pour moyenne q/p et pour variance q/p^2 . Sous certains aspects, la loi géométrique peut être vue comme l'analogie discrète de la loi exponentielle.

8.2.5 Loi négative-binomiale

C'est la loi du nombre de lancers nécessaires à un jeu de pile ou face de probabilité de gain p pour obtenir exactement m succès. C'est donc la loi de la v.a. $T_m(p)$ où $T_1(p) = T(p)$ (de loi géométrique) et $T_i(p)$ ($i \geq 1$) est défini par récurrence par :

$$T_i(p) := \inf \{n > T_{i-1}, X_n = 1\}.$$

Il est facile de voir que $T_m(p)$ a la même loi que la somme de m v.a. i.i.d. de loi géométrique de paramètre p . On a donc :

$$\mathcal{G}_m(p) = \sum_{k=m}^{+\infty} C_{n-1}^{m-1} p^m q^{k-m} \delta_k = \mathcal{G}(p)^{*m}.$$

On retrouve bien la loi géométrique pour $m = 1$: $\mathcal{G}(p) = \mathcal{G}_1(p)$. La loi $\mathcal{G}_m(p)$ a pour moyenne m/p , pour variance mq/p^2 et pour fonction génératrice $(ps/(1-sq))^m$. Elle tire son nom du fait que les coefficients de sa fonction génératrice proviennent d'une sorte de « formule du binôme inversé ».

Remarque 8.2.1 (Schéma de Bernoulli). Le schéma de Bernoulli correspond à un jeu de pile ou face infini, avec probabilité $p \in]0, 1[$ de succès à chaque lancé (les cas extrêmes sont triviaux). On le modélise par la donnée d'une suite $(X_n, n \in \mathbb{N}^*)$ de v.a. i.i.d. de loi de Bernoulli de paramètre p sur $\{0, 1\}$: à chaque lancé, on code 0 un échec (probabilité $q := 1 - p$) et 1 un succès (probabilité p). Le théorème de Caratheodory permet de définir une tribu et une loi de probabilité \mathcal{L} sur l'ensemble $\{0, 1\}^{\mathbb{N}^*}$ des suites infinies de 0 et de 1 telle que (X_1, X_2, \dots) soit de loi \mathcal{L} . Bien entendu, pour toute suite s dans cet espace, on a $\mathcal{L}(\{s\}) = 0$. En revanche, on a pour tout $n \in \mathbb{N}^*$ et $a \in \{0, 1\}^n$, par indépendance des X_1, \dots, X_n : $\mathbb{P}(X_1 = a_1, \dots, X_n = a_n) = \mathbb{P}(X_1 = a_1) \cdots \mathbb{P}(X_n = a_n)$. Cela correspond à l'événement cylindrique $\{a_1\} \times \cdots \times \{a_n\} \times \{0, 1\} \times \{0, 1\} \times \cdots$. La plupart des questions que l'on peut se poser sur le jeu de pile ou face reviennent à étudier certaines propriétés de la loi \mathcal{L} . Ses marginales sont des lois de Bernoulli $\mathcal{B}(1, p)$, et la loi de la somme de k termes est binomiale $\mathcal{B}(k, p)$. Sous \mathcal{L} , la loi du rang du premier 1 dans la suite infinie est géométrique $\mathcal{G}(p)$, et plus généralement la loi du rang du $m^{\text{ième}}$ 1 est négative-binomiale $\mathcal{G}_m(p)$. Les schémas de Bernoulli sont beaucoup moins triviaux qu'on le pense, et fournissent un très bon terrain d'investigation (y compris pour les systèmes dynamiques...). La section 1.5.1 page 21 donne une propriété fractale amusante de certains schémas de Bernoulli.

8.2.6 Loi de Poisson

La loi de Poisson $\mathcal{P}(\lambda)$ de paramètre $\lambda > 0$ est définie par :

$$e^{-\lambda} \sum_{n=0}^{+\infty} \frac{\lambda^n}{n!} \delta_n,$$

C'est la loi du nombre de v.a.r. i.i.d. de loi exponentielle de paramètre λ qu'il faut additionner pour dépasser le seuil 1. Elle a pour moyenne et variance λ et pour fonction génératrice $e^{\lambda(s-1)}$. La loi de Poisson est en quelque sorte l'analogie discret de la loi gaussienne. On a par exemple $\mathcal{P}(\lambda_1) * \mathcal{P}(\lambda_2) = \mathcal{P}(\lambda_1 + \lambda_2)$. On dispose même d'un « théorème central limite poissonien » : $\mathcal{B}(n, \lambda/n)$ converge étroitement vers $\mathcal{P}(\lambda)$ lorsque $n \rightarrow +\infty$, cf. [BL98, th. V.5.6 p. 148].

8.2.7 Loi multinomiale

La loi multinomiale $\mathcal{M}(n, k, p_1, \dots, p_k)$ de taille $n \in \mathbb{N}^*$, de dimension $k \in \mathbb{N}^*$ et de paramètre $(p_1, \dots, p_k) \in \mathbb{R}_+^k$ avec $p_1 + \cdots + p_k = 1$ est la loi portée par l'ensemble

$$\{(n_1, \dots, n_k) \in \mathbb{N}^k, \text{ t.q. } n_1 + \cdots + n_k = n\}$$

et donnée par la formule :

$$\mathcal{M}(n, k, p_1, \dots, p_k)(\{n_1, \dots, n_k\}) := \frac{n!}{n_1! \dots n_k!} p_1^{n_1} \dots p_k^{n_k}.$$

Elle tire son nom de la formule du multinôme :

$$(a_1 + \dots + a_k)^n = \sum_{\substack{0 \leq n_1, \dots, n_k \leq n \\ n_1 + \dots + n_k = n}} \frac{n!}{n_1! \dots n_k!} a_1^{n_1} \dots a_k^{n_k},$$

qui donne :

$$\mathcal{M}(n, k, p_1, \dots, p_k) = \sum_{\substack{0 \leq n_1, \dots, n_k \leq n \\ n_1 + \dots + n_k = n}} \frac{n!}{n_1! \dots n_k!} p_1^{n_1} \dots p_k^{n_k} \delta_{(n_1, \dots, n_k)} = (p_1 \delta_{e_1} + \dots + p_k \delta_{e_k})^{*n},$$

où $\{e_1, \dots, e_k\}$ est la base canonique de \mathbb{R}^k . On retrouve la loi binomiale pour $k = 1$:

$$\mathcal{M}(n, 1, p) = \mathcal{B}(n, p).$$

Si (X_1, \dots, X_n) sont des v.a. i.i.d. de loi (p_1, \dots, p_k) sur $\{1, \dots, k\}$, et si $N_i := \#\{j = 1, \dots, n, \text{ t.q. } X_j = i\}$ alors (N_1, \dots, N_k) suit la loi $\mathcal{M}(n, k, p_1, \dots, p_k)$. Remarquons que par définition, $N_1 + \dots + N_k = n$. La loi multinomiale permet par exemple de modéliser les résultats de n lancers indépendants d'un dé à k faces, N_i représentant alors le nombre de fois où la face numéro i a été obtenue. On a enfin la convergence étroite suivante vers une loi normale :

$$\left(\frac{N_1 - np_1}{\sqrt{np_1}}, \dots, \frac{N_k - np_k}{\sqrt{np_k}} \right) \xrightarrow[n \rightarrow +\infty]{\text{étr.}} \mathcal{N}(0, \Sigma_{p_1, \dots, p_k}),$$

où

$$\Sigma_{p_1, \dots, p_k} := \mathbf{I}_k - (\sqrt{p_1}, \dots, \sqrt{p_k})^\top (\sqrt{p_1}, \dots, \sqrt{p_k}).$$

La loi normale $\mathcal{N}(0, \Sigma(p_1, \dots, p_k))$ est la loi de la projection d'un vecteur aléatoire de loi $\mathcal{N}(0, \mathbf{I}_k)$ sur l'hyperplan orthogonal au vecteur $(\sqrt{p_1}, \dots, \sqrt{p_k})^\top$. Sa matrice de covariance est de rang $k - 1$, elle est donc dégénérée (i.e. singulière).

8.2.8 Loi hypergéométrique

La loi hypergéométrique de paramètres $(N, N_1, n) \in \mathbb{N}^* \times \mathbb{N}^* \times \mathbb{N}^*$ avec $N_1 \leq N$ et $n \leq N$ est la loi sur $\{0, \dots, n\}$ donnée par :

$$\sum_{k=0}^n \frac{C_{N_1}^k C_{N-N_1}^{n-k}}{C_N^n} \delta_k.$$

Si l'on considère une population de N individus dont N_1 possèdent un caractère particulier, par exemple une maladie, alors la loi hypergéométrique de paramètre (N, N_1, n) représente la loi du nombre d'individus possédant ce caractère dans un échantillon de taille n de la population. C'est donc la loi de $X_1 + \dots + X_n$ où les v.a. X_i sont i.i.d. et ont pour loi $((N - N_1)/N)\delta_0 + (N_1/N)\delta_1$.

Elle a pour moyenne nN_1/N et pour variance $n(N - n)N_1(N - N_1)(N - 1)^{-1}N^{-2}$. Bien sûr, de la même façon que la loi multinomiale généralise la loi binomiale, on pourrait définir une loi hypergéométrique multiple lorsque l'on s'intéresse à un nombre $m \geq 1$ de caractères dans la population.

8.3 Transformées de Fourier & de Laplace

Pour les lois continues, la fonction génératrice est remplacée par la transformée de Fourier ou de Laplace, qui sont plus puissantes et gardent leur sens pour les lois discrètes. Soit μ une mesure de probabilité sur (Ω, \mathcal{A}) où Ω est un espace vectoriel topologique et \mathcal{A} sa tribu borélienne. On note Ω' l'espace dual topologique de Ω , constitué des formes linéaires continues sur Ω . La transformée de Fourier de μ est définie par :

$$\begin{aligned} \mathcal{F}(\mu) : \Omega' &\rightarrow \mathbb{R} \\ f &\mapsto \mathbf{E}_\mu(\exp(if)) \end{aligned}$$

Cette fonction est bien définie puisque pour tout $f \in \Omega'$, la fonction $\exp(if) : \Omega \rightarrow \mathbb{C}$ est continue et bornée. Lorsque Ω est un espace Hilbert \mathbb{H} (\mathbb{R}^n par exemple), on identifie Ω' à \mathbb{H} et on pose :

$$\begin{aligned} \text{Fourier}(\mu) : \mathbb{H} &\rightarrow \mathbb{C} \\ s &\mapsto \int_{\Omega} \exp(i \langle s, x \rangle) d\mu(x). \end{aligned}$$

La transformée de Laplace, quant à elle, est définie lorsque cela a un sens par :

$$\text{Laplace}(\mu)(u) := \int_{\Omega} \exp(-\langle u, x \rangle) d\mu(x).$$

Formellement, il suffit de prendre $s = iu$ dans la définition de la transformée de Fourier pour obtenir la transformée de Laplace, lorsqu'elle existe.

La transformée de Fourier (ou de Laplace) caractérise la loi et l'indépendance. Ainsi, deux v.a. ont la même loi si et seulement si elles ont même transformée et X et Y sont indépendantes si et seulement si la transformée du couple (X, Y) en (s_1, s_2) est égale au produit des transformées respectives en s_1 et s_2 pour tout (s_1, s_2) . bien entendu, la transformée en s de la somme de deux v.a. indépendantes est égale au produit des transformées respectives en s (la réciproque est fautive).

La transformée de Fourier d'une loi symétrique (i.e. invariante par la symétrie centrale de centre 0) est réelle.

La transformée de Fourier de δ_a avec $a \in \mathbb{R}^n$ est donnée par $e^{i\langle s, a \rangle}$, elle permet de calculer les transformées de Fourier des lois discrètes (i.e. à support au plus dénombrable sans point d'accumulation). Ainsi, les lois binomiales $\mathcal{B}(n, p)$ et de Poisson $\mathcal{P}(\lambda)$ ont pour transformée de Fourier respectives :

$$(pe^{is} + q)^n \quad \text{et} \quad e^{\lambda(e^{is} - 1)}.$$

Quant à la loi géométrique $\mathcal{G}(p)$, sur \mathbb{N}^* , et à la loi négative-binomiale $\mathcal{G}_m(p)$, elles ont pour transformée de Fourier respectives :

$$\frac{pe^{is}}{1 - qe^{is}} \quad \text{et} \quad \frac{p^m e^{mis}}{(1 - qe^{is})^m}.$$

Heuristique 8.3.1 (Transformation de Fourier = analyse en fréquences). La transformation de Fourier d'une fonction f exprime les composantes en fréquences (on dit aussi spectre), indexées par s , de son argument f . La fonction $x \mapsto e^{isx}$ représente une fonction périodique élémentaire, dite *harmonique de fréquence* $2\pi/s$. Il faut beaucoup de spectre pour reconstituer une fonction f constante ou plus généralement « peu périodique » à partir des harmoniques. La transformation de Fourier change la convolution en un produit. Pour tronquer le spectre d'un signal f , il suffit de le convoluer avec la transformée de Fourier inverse de la fonction indicatrice de troncature.

8.4 Lois continues

8.4.1 Loi uniforme

La loi uniforme $\mathcal{U}(a, b)$ sur l'intervalle compact non vide $[a, b] \subset \mathbb{R}^n$ a pour densité par rapport à la mesure de Lebesgue dx sur \mathbb{R} :

$$\frac{1}{b-a} \mathbb{I}_{[a,b]}(x).$$

Elle a pour moyenne $(a+b)/2$ et pour variance $(b-a)^2/12$. Sa transformée de Fourier est :

$$e^{is(a+b)/2} \frac{\sin(s(b-a)/2)}{s(b-a)/2}.$$

Tout ceci se généralise sans difficultés à un pavé $[a_1, b_1] \times \dots \times [a_n, b_n]$ de \mathbb{R}^n .

8.4.2 Loi triangulaire

C'est la loi « uniforme » sur le triangle du plan de base $[-a, a]$ et de sommet de coordonnées $(0, 1)$. Elle a pour densité par rapport à la mesure de Lebesgue dx sur \mathbb{R} :

$$\frac{1}{a} \left(1 - \frac{|x|}{a}\right) \mathbb{I}_{[-a,a]}(x),$$

pour moyenne 0, pour variance $a^2/6$ et pour transformée de Fourier :

$$\frac{2(1 - \cos(as))}{a^2 s^2}.$$

8.4.3 Loi gaussienne

On parle aussi de loi normale, de Gauss, ou encore de Maxwell! En Amérique du nord, on parle souvent de « bell curve », i.e. de courbe en cloche. Ces multiples appellations témoignent de l'importance de cette loi. On pourrait définir les lois gaussiennes sur des espaces très généraux, mais cela nous mènerait bien trop loin¹. La loi gaussienne $\mathcal{N}(m, \Gamma)$, sur \mathbb{R}^n , de moyenne $m \in \mathbb{R}^n$ et de matrice de covariance $\Sigma \in \mathbb{S}_n^+(\mathbb{R}) \subset \mathcal{M}_n(\mathbb{R})$, a pour densité par rapport à la mesure de Lebesgue $dx_1 \cdots dx_n$ sur \mathbb{R}^n :

$$(2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2} \langle x - m, \Sigma(x - m) \rangle\right).$$

Elle a pour transformée de Fourier :

$$\exp\left(i \langle s, m \rangle - \frac{1}{2} \langle \Sigma s, s \rangle\right).$$

Elle a pour propriété de maximiser l'entropie de Shannon définie sur les lois des vecteurs aléatoires à densité f de covariance fixée Σ par :

$$\mathbf{H}(f) := - \int_{\mathbb{R}^n} f(x) \log f(x) dx.$$

8.4.4 Loi exponentielle et loi gamma

La loi gamma $\Gamma(a, \lambda)$ de paramètres $(a, \lambda) \in \mathbb{R}_+^* \times \mathbb{R}_+^*$ a pour densité sur \mathbb{R} par rapport à la mesure de Lebesgue dx sur \mathbb{R} :

$$\frac{\lambda^a}{\Gamma(a)} x^{a-1} e^{-\lambda x} \mathbf{I}_{\mathbb{R}^+}(x),$$

où $\Gamma(a)$ est la fonction Gamma d'Euler définie par :

$$\Gamma(a) := \int_0^{+\infty} x^{a-1} e^{-x} dx, \quad \text{en particulier } \Gamma(n) = (n-1)!.$$

Parfois, on choisit un paramétrage différent de la loi gamma, en posant $b = 1/\lambda$, et l'on note alors $G(a, b) = \Gamma(a, 1/b)$. Pour $a = 1$, on obtient la loi exponentielle :

$$\Gamma(1, \lambda) = G(1, 1/\lambda) = \mathcal{E}(\lambda).$$

La loi exponentielle est l'analogue continu de la loi géométrique, elle apparaît comme loi du temps inter-sauts du processus de Poisson. La loi exponentielle est également sans mémoire : si $\mathcal{L}(X) = \mathcal{E}(\lambda)$, alors $\mathbb{P}(X \geq t + s \mid X \geq t) = \exp(-\lambda s)$ ne dépend que de s et pas de t . Cette propriété la caractérise dans l'ensemble des lois continues, et traduit bien le comportement de durées de vie en fiabilité, du temps d'arrivée du prochain client dans une file d'attente, etc. Elle explique également la stationnarité des accroissements du processus de Poisson (renouvellement).

La loi gamma $\Gamma(a, \lambda)$ a pour moyenne a/λ , pour variance a/λ^2 et pour transformée de Fourier $(1 - is\lambda^{-1})^{-a}$. La loi $\Gamma(n, \lambda)$ avec $n \in \mathbb{N}^*$ est parfois appelée loi de Erlang. Ces lois sont très importantes puisqu'elles apparaissent lorsque l'on fait la somme de v.a.r. i.i.d. de loi exponentielle. La famille des lois gamma est stable par convolution et constitue donc en quelque sorte l'analogue continu de la famille des lois négatives-binomiales. La loi du temps de dépassement d'un seuil pour un processus de Poisson est une loi gamma, car convolution de lois exponentielles.

¹Sur un espace préhilbertien \mathbb{H} , les choses sont simples : une v.a. à valeurs dans \mathbb{H} est gaussienne si et seulement si pour tout v dans \mathbb{H} , la v.a.r. $\langle X, v \rangle_{\mathbb{H}}$ est gaussienne réelle. On retrouve facilement les vecteurs gaussiens usuels en prenant $\mathbb{H} = \mathbb{R}^n$.

8.4.5 Loi double exponentielle

Également appelée loi de Laplace. Elle a pour densité sur \mathbb{R} par rapport à la mesure de Lebesgue dx sur \mathbb{R} : $\lambda \exp(-\lambda|x|)/2$, où λ est un paramètre dans \mathbb{R}_+ . Sa moyenne vaut 0, sa variance $2/\lambda^2$ et sa transformée de Fourier est $1/(1+s^2/\lambda^2)$.

8.4.6 Loi de Dirichlet

La loi de Dirichlet de taille $n \in \mathbb{N}^*$ et de paramètre $\alpha \in (\mathbb{R}_+^*)^n$ est la loi du vecteur aléatoire $(X_1, \dots, X_n)/(X_1 + \dots + X_n)$ lorsque les variables aléatoires X_1, \dots, X_n sont i.i.d. et suivent des lois exponentielles de paramètres respectifs $\alpha_1, \dots, \alpha_n$, cf. [DCD82b, exercice 8.2.15, page 192]. Elle est portée par le simplexe $\mathbb{S}(n, 1) := \{y \in \mathbb{R}^n, y_i \geq 0, y_1 + \dots + y_n = 1\}$. La densité de (Y_1, \dots, Y_{n-1}) est donnée par :

$$\frac{\Gamma(\alpha_1 + \dots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \left(\prod_{i=1}^{n-1} y_i^{\alpha_i - 1} \right) \left(1 - \sum_{i=1}^{n-1} y_i \right)^{\alpha_n - 1}.$$

La loi de Dirichlet de taille n et de paramètre $\alpha = (1, \dots, 1)$ n'est rien d'autre que la loi uniforme sur le simplexe $\mathbb{S}(n, 1)$, cf. section 1.7 page 32. Elle permet par exemple de choisir « uniformément » une loi de probabilité discrète.

8.4.7 Lois du chi-deux

La loi du $\chi^2(n)$ à n degrés de liberté n'est rien d'autre que la loi de la somme des carrés de n v.a.r. i.i.d. de loi normale centrées réduites sur \mathbb{R} . C'est aussi la loi gamma $G(n/2, 2)$ de paramètres $(a, b) := (n/2, 2)$, i.e. la loi gamma $\Gamma(n/2, 1/2)$ de paramètres $(a, \lambda) := (n/2, 1/2)$. La loi du $\chi^2(n)$ intervient dans le test du même nom (cf. 3.3.2 page 62).

8.4.8 Loi exponentielle et loi de Weibull

La loi de Weibull $W(\alpha, \lambda)$ de paramètres $(\alpha, \lambda) \in \mathbb{R}_+^* \times \mathbb{R}_+^*$ est la loi sur \mathbb{R} dont la densité par rapport à la mesure de Lebesgue dx sur \mathbb{R} est :

$$\alpha \lambda x^{\alpha-1} \exp(-\lambda x^\alpha) \mathbb{I}_{\mathbb{R}_+}(x).$$

Elle est plus communément donnée par sa fonction de répartition :

$$F_{\alpha, \lambda}(u) = (1 - \exp(-\lambda u^\alpha)) \mathbb{I}_{\mathbb{R}_+}(u).$$

Sa moyenne vaut $\lambda^{-1/\alpha} \Gamma(1 + \alpha^{-1})$ et sa variance $\lambda^{-2/\alpha} \left(\Gamma(1 + 2\alpha^{-1}) - (\Gamma(1 + \alpha^{-1}))^2 \right)$. Cette loi intervient en fiabilité des systèmes. Notons que l'on retrouve la loi exponentielle de paramètre λ lorsque $\alpha = 1$:

$$W(1, \lambda) = \mathcal{E}(\lambda).$$

Pour $\alpha = 2$, on parle parfois de loi de Rayleigh.

8.4.9 Loi de Cauchy

C'est la loi sur \mathbb{R} de densité par rapport à la mesure de Lebesgue dx sur \mathbb{R} :

$$\frac{c}{\pi} \frac{1}{c^2 + x^2},$$

où c est un paramètre positif. Elle n'a pas de moyenne et donc pas de variance. Sa transformée de Fourier est $\exp(-c|s|^2)$.

8.4.10 Loi Beta

La loi bêta de paramètres $(a, b) \in \mathbb{R}_+^* \times \mathbb{R}_+^*$ est la loi sur \mathbb{R} de densité par rapport à la mesure de Lebesgue dx sur \mathbb{R} :

$$\frac{1}{B(a, b)} x^{a-1} (1-x)^{b-1} \mathbf{I}_{\{0 < x < 1\}}(x),$$

où $B(a, b)$ est la fonction bêta d'Euler définie par :

$$B(a, b) := \int_0^1 x^{a-1} (1-x)^{b-1} dx = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}.$$

La loi bêta de paramètres (a, b) a pour moyenne $a/(a+b)$ et pour variance $ab(a+b)^{-2}(a+b+1)^{-1}$.

8.4.11 Loi de Pareto

Si $\mathcal{L}(X) = \mathcal{E}(\lambda)$ et si $r \in \mathbb{R}$, la v.a. $r \exp(X)$ suit par définition une loi de Pareto de paramètres λ et r . Sa densité par rapport à la mesure de Lebesgue sur \mathbb{R} vaut :

$$\frac{\lambda r^\lambda}{x^{1+\lambda}} \mathbf{I}_{\{x > r\}}(x).$$

Elle a pour moyenne $\alpha r/(\alpha - 1)$ si $\alpha > 1$ et n'a pas moyenne sinon. Si $\alpha > 2$, elle a pour variance $\alpha r^2(\alpha - 1)^{-2}(\alpha - 2)^{-1}$, et elle n'a pas de moment d'ordre deux sinon.

8.4.12 Loi log-normale

C'est la loi de l'exponentielle d'une variable aléatoire de loi gaussienne $\mathcal{N}(m, \sigma^2)$. Elle a pour densité par rapport à la mesure de Lebesgue dx sur \mathbb{R} :

$$(2\pi\sigma^2)^{-1/2} x^{-1} \exp\left(-\frac{(\log(x) - m)^2}{2\sigma^2}\right) \mathbf{I}_{\{x > 0\}}(x),$$

pour moyenne $\exp(m + \sigma^2/2)$ et pour variance $\exp(2m + \sigma^2)(e^{\sigma^2} - 1)$.

8.4.13 Loi de Fisher-Snedecor

La loi de Fisher-Snedecor $\mathcal{F}(n, m)$ à n et m degrés de liberté est la loi de

$$\frac{X/n}{Y/m},$$

où X et Y sont indépendantes et ont pour lois respectives $\chi^2(n)$ et $\chi^2(m)$. Elle a pour densité par rapport à la mesure de Lebesgue dx sur \mathbb{R} :

$$\frac{\Gamma((n+m)/2)n^{n/2}m^{m/2}}{\Gamma(n/2)\Gamma(m/2)} \frac{x^{n/2-1}}{(m+nx)^{(n+m)/2}}.$$

Pour $m > 4$, sa moyenne vaut $m/(m-2)$ et sa variance $2m^2(m+n-2)n^{-1}(m-4)^{-1}(m-2)^{-2}$.

8.4.14 Loi de Student

La loi de Student \mathcal{T}_n à n degrés de liberté est la loi de

$$\frac{\sqrt{n} X}{\sqrt{Y}},$$

où X et Y sont indépendantes et ont pour lois respectives $\mathcal{N}(0, 1)$ et $\chi^2(n)$. Elle a pour densité par rapport à la mesure de Lebesgue dx sur \mathbb{R} :

$$\frac{\Gamma((n+1)/2)}{\Gamma(n/2)\sqrt{n\pi}} \left(1 + \frac{x^2}{n}\right)^{-(n+1)/2}.$$

Elle est centrée et sa variance vaut $n/(n-2)$ pour $n > 2$.

8.4.15 Loi de Kolmogorov-Smirnov

C'est la loi sur \mathbb{R} de fonction de répartition :

$$F(u) := \begin{cases} 0 & \text{si } u \leq 0, \\ 1 + 2 \sum_{k=1}^{+\infty} (-1)^k e^{-2k^2 u^2} & \text{si } u > 0. \end{cases}$$

Elle intervient dans le test d'adéquation de Kolmogorov-Smirnov (cf. 3.3.3 page 65). C'est aussi la loi sur supremum de la valeur absolue d'un pont brownien sur $[0, 1]$ (cf. section 2.4 page 45).

8.4.16 Statistique d'ordre, quantiles, médiane

Soit μ une loi de probabilité sur \mathbb{R} , de fonction de répartition F . On note $F(x^-)$ sa limite à gauche en x . On dit qu'un réel m est une *médiane* pour μ lorsque $F(m^-) \leq 1/2 \leq F(m)$. Plus généralement, pour tout $\alpha \in]0, 1[$, on dit qu'un réel q_α est un *quantile* d'ordre α pour μ lorsque $F(q_\alpha^-) \leq \alpha \leq F(q_\alpha)$. Pour tout ordre α , un quantile d'ordre α existe toujours, car la fonction de répartition F est croissante, et a pour limite 0 en $-\infty$ et 1 en $+\infty$. La croissance de F entraîne également que l'ensemble des quantiles d'ordre α est un intervalle. Si q est un quantile, l'intervalle des quantiles du même ordre est réduit à $\{q\}$ si et seulement si $\mu(\{q_\alpha\}) = 0$, et alors F est continue en α (on rappelle qu'une fonction de répartition est toujours continue à droite). Souvent, on appelle encore quantile d'ordre α le milieu de l'intervalle des quantiles d'ordre α , ce qui permet de parler *du* quantile d'ordre α , même quand il y en a plusieurs. Enfin, lorsque la loi μ possède une densité, sa fonction de répartition est continue et les intervalles quantiles sont tous des singletons.

On parle de *déciles* pour les quantiles d'ordre $1/10$, de *quartiles* pour ceux d'ordre $1/4$, etc. On parle aussi parfois de *percentiles* pour les quantiles. Une médiane est un quantile d'ordre $1/2$. La fonction Matlab `median` permet d'obtenir la médiane empirique d'un échantillon (i.e. la médiane de la fonction de répartition empirique), alors que la fonction Stixbox `quantile` permet d'obtenir un quantile empirique d'ordre quelconque d'un échantillon.

Si (x_1, \dots, x_n) est un vecteur de \mathbb{R}^n , on note $(x_{(1)}, \dots, x_{(n)})$ le vecteur de \mathbb{R}^n obtenu en réordonnant de façon croissante les composantes de x . Si $X := (X_1, \dots, X_n)$ est un vecteur aléatoire de \mathbb{R}^n , on appelle statistique d'ordre de X le vecteur aléatoire $\tilde{X} := (X_{(1)}, \dots, X_{(n)})$ obtenu en réordonnant les coordonnées de X à ω fixé. La fonction Matlab `sort` permet de trier de façon croissante un vecteur.

Si $X := (X_1, \dots, X_n)$ est un échantillon d'une loi μ sur \mathbb{R} , alors la médiane empirique n'est rien d'autre que $X_{((n+1)/2)}$ si n est impair et $(X_{(n/2)} + X_{((1+n)/2)})/2$ si n est pair. De manière générale, si α n'est pas un multiple de $1/n$, le quantile d'ordre α empirique vaut $(X_{([n\alpha])} + X_{([n\alpha]+1)})/2$. Enfin, on peut montrer que si μ a une densité f par rapport à la mesure de Lebesgue, alors la loi de $X_{(i)}$ a pour densité :

$$f_i(x) = \frac{n!}{(i-1)!(n-i)!} F^{i-1}(x)(1-F(x))^{n-i} f(x),$$

où F est la fonction de répartition de μ . On retrouve bien f pour $n = 1$. Si $x := (x_1, \dots, x_n) \in \mathbb{R}^n$, on appelle *rang* de x_i dans la suite x la quantité

$$r_i(x) := \sum_{j=1}^n \mathbf{1}_{x_j \leq x_i},$$

qui représente le nombre de composantes de x qui sont inférieures ou égales à x_i . Il peut arriver que $r_i = r_j$, mais on a toujours $x_{r_i(x)} = x_{(i)}$. Lorsque les composantes de x sont toutes différentes, l'application $r : i \in \{1, \dots, n\} \mapsto r_i(x) \in \{1, \dots, n\}$ est une permutation. Le vecteur $r(x) := (r_1(x), \dots, r_n(x))$ est appelé *vecteur rang* de x . On définit naturellement le vecteur aléatoire rang $R(X)$ d'un vecteur aléatoire $X := (X_1, \dots, X_n)$ par $R(X)(\omega) := r(X(\omega))$. On montre, cf. [DCD82b, thm 4.4.29 page 107] que pour un échantillon $X := (X_1, \dots, X_n)$ d'une loi μ sur \mathbb{R} , continue :

- le vecteur rang $R(X)$ est presque-sûrement une permutation
- le vecteur rang $R(X)$ suit la loi uniforme sur l'ensemble des permutations de $\{1, \dots, n\}$
- la statistique d'ordre \tilde{X} de X et son rang $R(X)$ sont indépendants
- la statistique d'ordre \tilde{X} a pour loi la loi trace de $n! \mu \otimes \dots \otimes \mu$ sur $\{x \in \mathbb{R}^n, x_1 \leq \dots \leq x_n\}$.

8.5 Convolutions & convergences

On a les propriétés suivantes :

1. $\text{Bern}(p, \{0, 1\})^{*n} = \mathcal{B}(n, p)$.
2. $\mathcal{G}(p)^{*m} = \mathcal{G}_m(p)$.
3. $\mathcal{G}_{m_1}(p) * \cdots * \mathcal{G}_{m_n}(p) = \mathcal{G}_{m_1 + \cdots + m_n}(p)$.
4. $\mathcal{E}(\lambda)^{*n} = \Gamma(n, \lambda)$.
5. $\mathcal{N}(m_1, \Sigma_1) * \cdots * \mathcal{N}(m_n, \Sigma_n) = \mathcal{N}(m_1 + \cdots + m_n, \Sigma_1 + \cdots + \Sigma_n)$.
6. $\mathcal{P}(\lambda_1) * \cdots * \mathcal{P}(\lambda_n) = \mathcal{P}(\lambda_1 + \cdots + \lambda_n)$.
7. Si $\mathcal{L}(X) = \mathcal{N}(0, \mathbf{I}_n)$ alors $\mathcal{L}(X_1^2 + \cdots + X_n^2) = G(n/2, 2) = \chi^2(n)$.
8. Si $\mathcal{L}((X, Y)) = \Gamma(a, 1) \otimes \Gamma(b, 1)$ alors $\mathcal{L}(X/(X+Y)) = \text{Beta}(a, b)$.
9. Si $\mathcal{L}(X) = \Gamma(1, \lambda) = \mathcal{E}(\lambda)$ alors $\mathcal{L}(X^{-\alpha}) = W(\alpha, \lambda)$.
10. Si $\mathcal{L}((X, Y)) = \chi^2(n) \otimes \chi^2(m)$, alors $\mathcal{L}((mX)/(nY)) = \mathcal{F}(n, m)$.
11. Si $\mathcal{L}((X, Y)) = \mathcal{N}(0, 1) \otimes \chi^2(n)$, alors $\mathcal{L}(\sqrt{n}X/\sqrt{Y}) = \mathcal{T}(n)$.
12. Si $\mathcal{L}((X_n, n \in \mathbb{N}^*)) = \mathcal{E}(\lambda)^{\mathbb{N}^*}$ et

$$N_t := \sum_{n=1}^{+\infty} \mathbf{1}_{X_1 + \cdots + X_n \leq t} = \inf \{n \in \mathbb{N}, X_1 + \cdots + X_n > t\},$$

alors pour tout $t \in \mathbb{R}_+$, $\mathcal{L}(N_t) = \mathcal{P}(\lambda t)$ (processus de Poisson simple d'intensité λ).

13. $\mathcal{B}(n, \lambda/n) \xrightarrow[n \rightarrow +\infty]{\text{étr.}} \mathcal{P}(\lambda)$

1. $X_n \xrightarrow[n \rightarrow +\infty]{\mathbb{P}} X$

(a) ssi $(\forall \varepsilon > 0) \mathbb{P}(\{|X_n - X| \geq \varepsilon\}) \xrightarrow[n \rightarrow +\infty]{} 0$

(b) ssi $\mathbf{E}(|X_n - X| \wedge 1) \xrightarrow[n \rightarrow +\infty]{} 0$

(c) ssi $\mathbf{E}(|X_n - X|/(1 + |X_n - X|)) \xrightarrow[n \rightarrow +\infty]{} 0$

(d) ssi de toute sous suite $(X_{n_k}, k \in \mathbb{N}^*)$, on peut extraire une sous-suite qui converge p.s. vers X

2. $X_n \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mu$ ssi $\mathcal{L}(X_n) \xrightarrow[n \rightarrow +\infty]{\text{étr.}} \mu$.

3. $\mu_n \xrightarrow[n \rightarrow +\infty]{\text{étr.}} \mu$ (avec μ_n et μ probas)

(a) ssi $(\forall f \in \mathcal{C}_b) \mathbf{E}_{\mu_n}(f) \xrightarrow[n \rightarrow +\infty]{} \mathbf{E}_\mu(f)$

(b) ssi $(\forall s \in \mathbb{R}) \mathbf{E}_{\mu_n}(e^{is\bullet}) \xrightarrow[n \rightarrow +\infty]{} \mathbf{E}_\mu(e^{is\bullet})$

(c) ssi $(\forall t \in \text{Cont}(\mathbb{F}_\mu)) F_{\mu_n}(t) \xrightarrow[n \rightarrow +\infty]{} F_\mu(t)$

4. La famille de v.a. $(X_i, i \in I) \subset \mathbf{L}^1$ est uniformément intégrable (UI)

(a) ssi $\sup_{i \in I} \mathbf{E}(|X_i| \mathbf{1}_{\{|X_i| > c\}}) \xrightarrow[c \rightarrow +\infty]{} 0$

(b) ssi² $\sup_{i \in I} \mathbf{E}(|X_i|) < +\infty$ et $(\forall \varepsilon > 0)(\exists \delta > 0)(\forall A)(\mathbb{P}(A) \leq \delta \Rightarrow \sup_{i \in I} \mathbf{E}(|X_i| \mathbf{1}_A) \leq \varepsilon)$

(c) ssi³ $\exists \Psi : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ telle que $\sup_{i \in I} \mathbf{E}(\Psi(|X_i|)) < +\infty$ et $\Psi(x)/x \xrightarrow[x \rightarrow +\infty]{} +\infty$.

5. Une famille bornée dans \mathbf{L}^1 est UI.

6. Soit $(X_n, n \in \mathbb{N}^*) \subset \mathbf{L}^1$, il y a équivalence entre :

(a) $X_n \xrightarrow[n \rightarrow +\infty]{\mathbf{L}^1} X$ avec $X \in \mathbf{L}^1$

(b) $(X_n, n \in \mathbb{N}^*)$ est UI et $X_n \xrightarrow[n \rightarrow +\infty]{\mathbb{P}} X$

²Critère « epsilon-delta » ou encore d'équi-intégrabilité.

³Critère de Lavallée-Poussin. cf. [Bor95, p. 10]

7. Si $X_n \xrightarrow[n \rightarrow +\infty]{\mathbf{L}^p} X$ avec $p \geq 1$, alors de toute sous-suite $(X_{n_k}, k \in \mathbb{N}^*)$, on peut extraire une sous-suite qui converge p.s. vers X
8. Si $X_n \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mu$ alors⁴ $s \mapsto \mathbf{E}(e^{isX_n})$ converge vers $s \mapsto \mathbf{E}_\mu(e^{is\bullet})$ uniformément sur tout compact
9. Soit $(\mu_n, n \in \mathbb{N}^*)$ des probas. Si $s \mapsto \mathbf{E}_{\mu_n}(e^{is\bullet})$ converge simplement vers $s \mapsto G(s)$ continue en 0 alors⁵ G est la transformée de Fourier d'une loi μ et $\mu_n \xrightarrow[n \rightarrow +\infty]{\text{étr.}} \mu$.
10. Soit $(X_n, n \in \mathbb{N}^*)$ v.a.r. indép. et $S_n := X_1 + \dots + X_n$. Il y a équivalence entre⁶ :
 - (a) $(S_n, n \in \mathbb{N}^*)$ converge p.s.
 - (b) $(S_n, n \in \mathbb{N}^*)$ converge en \mathbb{P}
 - (c) $(S_n, n \in \mathbb{N}^*)$ converge en \mathcal{L}
 De plus⁷, si $c > 0$ et $Y_n := X_n \mathbf{1}_{\{|X_n| \leq c\}}$ alors $(S_n, n \in \mathbb{N}^*)$ converge p.s. ssi on a à la fois :
 - (a) $\sum_{n=1}^{+\infty} \mathbb{P}(\{X_n \neq Y_n\}) < +\infty$
 - (b) $\sum_{n=1}^{+\infty} \mathbf{E}(Y_n) < +\infty$
 - (c) $\sum_{n=1}^{+\infty} \mathbf{Var}(Y_n) < +\infty$
11. Si $X_n \xrightarrow[n \rightarrow +\infty]{\text{p.s.}} X$ et $Y_n \xrightarrow[n \rightarrow +\infty]{\text{p.s.}} Y$ et $(a, b) \in \mathbb{R}^2$ alors $aX_n + bY_n \xrightarrow[n \rightarrow +\infty]{\text{p.s.}} aX + bY$ et $X_n Y_n \xrightarrow[n \rightarrow +\infty]{\text{p.s.}} XY$
12. L'espace \mathbf{L}^0 des fonctions mesurables muni de la topologie de la CV en \mathbb{P} associée à la distance $d(X, Y) := \mathbf{E}(|X - Y|/(1 + |X - Y|))$ est un espace de Banach⁸.
13. Si $X_n \xrightarrow[n \rightarrow +\infty]{\mathbb{P}} X$ et $Y_n \xrightarrow[n \rightarrow +\infty]{\mathbb{P}} Y$ alors $X_n Y_n \xrightarrow[n \rightarrow +\infty]{\mathbb{P}} XY$
14. Si $X_n \xrightarrow[n \rightarrow +\infty]{\mathbf{L}^1} X$ et $Y_n \xrightarrow[n \rightarrow +\infty]{\mathbb{P}} Y$ et $(Y_n, n \in \mathbb{N}^*)$ unif. bornée alors $X_n Y_n \xrightarrow[n \rightarrow +\infty]{\mathbf{L}^1} XY$
15. Si $(X_n, Y_n) \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} (X, Y)$ alors $X_n + Y_n \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} X + Y$ et $X_n \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} X$ et $Y_n \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} Y$
16. Si $X_n \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} X$ et $Y_n \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} Y$ et $(X_n, Y_n, n \in \mathbb{N}^*)$ indép. alors X et Y indép. et $X_n + Y_n \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} X + Y$

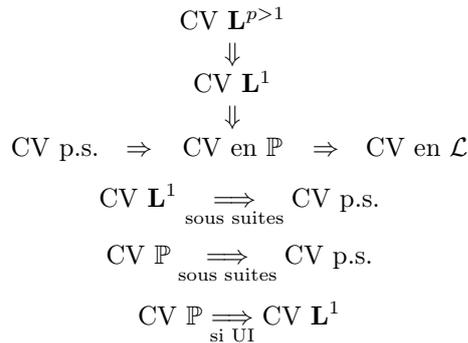


FIG. 8.1 – Relations entre les différents types de convergences de v.a.

⁴Théorème de continuité de Paul Lévy, cf. [Bor95, p. 32].

⁵Théorème de continuité de Paul Lévy, cf. [Bor95, p. 32].

⁶Théorème de Paul Lévy, cf. [Bor95, p. 83]

⁷Théorème des trois séries de Kolmogorov, cf. [Bor95, p. 85]

⁸Donc un espace vectoriel normé, d'où l'additivité des suites convergentes en \mathbb{P} .

Loi	Moyenne	Variance	Transformée de Fourier en $s \in \mathbb{C}$
δ_a	a	0	e^{ias}
Bern. $q\delta_0 + p\delta_1$	p	pq	$1 + pe^{is}$
Bern. $q\delta_{-1} + p\delta_1$	$q - p$	pq	$qe^{-is} + pe^{is}$
Unif. sur $\{a_1, \dots, a_n\}$	$(a_1 + \dots + a_n)/n$	$(\sum_k a_k^2)/n - (\sum_k a_k)^2/n^2$	$\sum_k e^{ia_k s}/n$
$\mathcal{B}(n, p)$	np	npq	$(1 + pe^{is})^n$
Géom(p) (sur \mathbb{N})	q/p	q/p^2	$pe^{is}(1 - qe^{is})^{-1}$
$\mathcal{G}(p)$ (sur \mathbb{N}^*)	$1/p$	q/p^2	$p^n e^{nis}(1 - qe^{is})^{-n}$
$\mathcal{G}_m(p)$	m/p	q/p^2	$\exp(\lambda(e^{is} - 1))$
$\mathcal{P}(\lambda)$	λ	λ	$2s^{-1}(b - a)^{-1} \exp(is(a + b)/2) \sin(s(b - a)/2)$
Unif. $\mathcal{U}(a, b)$	$(a + b)/2$	$(b - a)^2/12$	$(1 - is\lambda^{-1})^{-1}$
$\mathcal{E}(\lambda)$	$1/\lambda$	$1/\lambda^2$	$(1 - is\lambda^{-1})^{-a}$
$\Gamma(a, \lambda)$	a/λ	a/λ^2	$(1 - ibs)^{-a}$
$G(a, b)$	ab	ab^2	$\exp(i < m, s > - < \Sigma s, s > /2)$
$\mathcal{N}(m, \Sigma)$	m	Σ	$(1 - 2is)^{-n/2}$
$\chi^2(n)$	n	$2n$	
$W(\alpha, \lambda)$	$\lambda^{-1/\alpha} \Gamma(1 + \alpha - 1)$	$\lambda^{-2/\alpha} (\Gamma(1 + 2\alpha^{-1}) - (\Gamma(1 + \alpha^{-1}))^2)$	
$\mathcal{F}(n, m)$	$m/(m - 2)$ si $m > 4$	$2m^2(m + n - 2)n^{-1}(m - 4)^{-1}(m - 2)^{-2}$	
$\mathcal{T}(n)$	0	$n/(n - 2)$ si $n > 2$	

TAB. 8.1 – Tableau récapitulatif de quelques lois usuelles

Loi μ	Fonction de répartition $F_\mu(t) := \mu([t, +\infty[)$
Discrète $\sum_{n \geq 0} p_n \delta_{a_n}$ ($a_n < a_{n+1}$)	$\sum_{n \geq 0} p_n \mathbb{I}_{[a_n, a_{n+1}[}(t)$
Unif. $\mathcal{U}(a, b)$	$(b - a)^{-1}(t - a) \mathbb{I}_{[a, b]}(t) + \mathbb{I}_{]b, +\infty[}(t)$
$\mathcal{E}(\lambda)$	$(1 - \exp(-\lambda t)) \mathbb{I}_{\mathbb{R}_+}(t)$
$W(\alpha, \lambda)$	$(1 - \exp(-\lambda t^\alpha)) \mathbb{I}_{\mathbb{R}_+}(t)$
$\mathcal{N}(0, 1)$	$\Phi(t) \sim_{+\infty} 1 - t^{-1} \exp(-t^2/2)$
Kolmogorov-Smirnov	$\left(1 + 2 \sum_{k=1}^{+\infty} (-1)^k e^{-2k^2 u^2}\right) \mathbb{I}_{\mathbb{R}_+^*}(t)$

TAB. 8.2 – Quelques fonctions de répartition

Annexe A

Informations utiles (et volatiles)

A.1 Liste des leçons d'oral

- 200001. Loi des grands nombres ; application à l'estimation.
- 200002. Le théorème de la limite centrale et son utilisation.
- 200003. Convergence d'une chaîne de Markov vers une loi stationnaire.
- 200004. Décrire une méthode probabiliste ou déterministe pour le calcul approché d'une intégrale.
- 200005. Théorème d'arrêt pour une martingale ; applications.
- 200006. Martingales : convergence et applications.
- 200007. Vecteurs aléatoires gaussiens.
- 200008. Modèle linéaire gaussien en statistique.
- 200009. Test d'adéquation du χ^2 .
- 200010. Tests statistiques : principes et exemples.
- 200011. Intervalles de confiance : principe et exemples.
- 200012. Écarts à la loi des grands nombres ; événements de faibles probabilité, applications.
- 200013. Fonction de répartition empirique : comportement asymptotique, usage en statistique.
- 200014. Exemples de chaînes de Markov récurrentes ou transientes à espace d'état au plus dénombrable.
- 200015. Évolution de la taille d'une population lorsque la loi de reproduction est homogène.
- 200016. Modélisation d'une durée de vie, renouvellement.
- 200017. Simulation de variables aléatoires; applications.
- 200101. Loi des grands nombres ; applications à l'estimation.
- 200102. Le théorème de la limite centrale ; application(s).
- 200103. Décrire une méthode probabiliste pour le calcul approché d'une intégrale.
- 200104. Espérance conditionnelle et applications (par exemple aux martingales).
- 200105. Exemples d'utilisation(s) de l'espérance conditionnelle en modélisation.
- 200106. Exemple d'utilisation(s) des martingales.
- 200107. Vecteur aléatoires gaussiens, utilisation.
- 200108. Modèle linéaire gaussien en statistique.
- 200109. Test du Khi-deux.
- 200110. Tests statistiques : principes et exemples.
- 200111. Intervalles de confiance : principe et exemples.
- 200112. Application de la transformée de Laplace (par exemple, à l'inégalité de Cramér-Chernoff, au problème de ruine du joueur, etc)
- 200113. Usage de la fonction de répartition empirique en statistique.
- 200114. Utilisation d'une loi exponentielle (par exemple, pour l'étude d'une durée de vie, etc).

http://www.education.gouv.fr/siac/siac2/	Infos concours au Ministère
http://www.math.u-psud.fr/~agreg/	Infos en provenance du jury (Orsay)
http://smf.emath.fr/Enseignements/	Rubrique enseignements de la SMF
http://www.umpa.ens-lyon.fr/~agregmath/	Préparation de l'ENS-Lyon
http://www.maths.lth.se/matstat/stibox/	Page officielle de Stibox
http://www.octave.org/	Page officielle d'Octave
http://www.mathworks.com/products/matlab/	Page officielle de Matlab
http://www-rocq.inria.fr/scilab/	Page officielle de Scilab
http://www.mupad.de/	Page officielle de Mupad
http://www.math-info.univ-paris5.fr/~ycart/polys/	Polycopiés de Bernard Ycart
http://www.cmla.ens-cachan.fr/Utilisateurs/scopos/ruget.html	Page du livre « Mathématiques en situation »
http://www.cmla.ens-cachan.fr/Utilisateurs/scopos/collection.html	Page de la collection Scopos
http://www.math.u-psud.fr/~lichnew/	Page d'Alain Lichniewsky
http://www.dma.ens.fr/~ldumas/	Page de Laurent Dumas
http://www.univ-angers.fr/cufco/matlab/	Livre de Eric Schrafstetter
http://www.lsp.ups-tlse.fr/Chafai/Agreg/	Page de l'auteur pour l'agrégation
http://www.lsp.ups-tlse.fr/Bakry/	Cours de Maîtrise de Dominique Bakry
http://www.lsp.ups-tlse.fr/Fp/Letac/	Cours de DEUG de Gérard Letac
http://www.lsp.ups-tlse.fr/Besse/	Cours de modélisation statistique de Ph. Besse

TAB. A.1 – Quelques liens sur la toile mondiale

- 200115. Simulation de variables et de vecteurs aléatoires; applications.
- 200116. Sommes de variables aléatoires; application (par exemple aux marches aléatoires, à la ruine du joueur etc.)
- 200117. Exemple de chaînes de Markov récurrentes ou transiente à espace d'états au plus dénombrable.
- 200118. Convergence d'une chaîne de Markov vers une loi stationnaire.
- 200119. Utilisation de la transformée de Laplace, de la fonction génératrice (par exemple, marches aléatoires, ruine du joueur, branchements, etc).
- 200120. Utilisations de la loi de Poisson en modélisation (par exemple, processus de comptage, etc).

A.2 Thèmes applicatifs 2001

- Contrôle de qualité et fiabilité.
- Géométrie effective et appliquée.
- Modèles en économie et finance.
- Propagation d'ondes.
- Télécommunications.

A.3 Textes 2001

- 601. Un modèle d'actifs financiers à longue mémoire
- 602. Le coussin financier
- 603. Convergence en loi des prix des actifs financiers
- 604. Minimisation du risque quadratique en finance
- 605. Instants de records
- 606. Aversion au risque dans les choix économiques
- 607. Processus ARMA et ARFIMA en finance
- 608. Processus à volatilité stochastique
- 609. Problème d'arrêt optimal et application aux options américaines.
- 610. Gestion optimale de portefeuilles dans un marché financier discret.
- 611. Modélisation ARCH
- 612. Modèle binomial
- 613. Rendement d'une ligne de transmission
- 614. Files d'attente
- 615. Indépendance conditionnelle et réseaux bayésiens
- 616. Fonctions de Lyapounov et stabilité de réseaux de télécommunications
- 617. Étude d'une liaison satellite avec erreurs aléatoires
- 618. Modélisation d'un central téléphonique avec répétitions d'appels
- 619. Réduction de dimension par analyse en composantes principales
- 620. Compression de données par quantification
- 621. Modélisation du système ALOHA
- 622. Codage et entropie
- 623. Modélisation du tracé d'un réseau de communication
- 624. Détection d'un signal en télécommunications
- 625. Indice de Pareto
- 626. Modèle de Cox et statistique bayésienne en fiabilité
- 627. Processus de Poisson et valeurs records
- 628. Echantillonnage préférentiel dans les calculs de fiabilité
- 629. Approximation exponentielle de la durée de vie d'un système fiable
- 630. Analyse d'un système multicomposants
- 631. Politique de remplacement à l'âge tau
- 632. Inférence statistique pour les quantiles

A.4 Information en provenance du jury

Ce qui suit est extrait du site <http://www.math.u-psud.fr/~agreg/> :

A.4.1 Utilisation des ordinateurs lors de l'épreuve Modélisation

« Après avoir tiré une feuille comportant un titre de leçon et un titre de texte, le candidat rejoint la salle de préparation. À l'accueil, il reçoit un nom de login et un mot de passe ainsi que le texte qu'il a tiré au sort. Le candidat est placé dans la salle de préparation où une machine est mise à sa disposition. Il devra indiquer au responsable son choix du système d'exploitation (environnement Linux ou Windows). Une fois le système configuré et lancé par le responsable, des icônes permettront au candidat de lancer les logiciels qu'il souhaite utiliser. La documentation sera aussi disponible en ligne. Une imprimante est mise à la disposition des candidats ; il est conseillé de ne pas attendre le dernier moment pour imprimer.

« Environ un quart d'heure avant la fin de sa préparation, il sera demandé au candidat de sauvegarder ses fichiers de travail (la manipulation de sauvegarde sera expliquée). Le responsable de la salle de préparation pourra être appelé en cas de problème de fonctionnement du système ; en revanche, il ne donnera pas d'explications sur le fonctionnement des logiciels mathématiques eux-mêmes qui sont supposés connus par le candidat. »

« La présentation du travail sur une machine devant le jury se fera à partir des fichiers sauvegardés. Cette machine aura la même configuration que celle utilisée pour la préparation et elle sera équipée d'un dispositif permettant à l'ensemble du jury d'en visionner l'écran. »

A.4.2 Matériels et Logiciels disponibles

Matériel : PC avec processeur Pentium et clavier français.

Logiciels système : Windows 98 SE, Red Hat Linux 6.1 + fvwm95

Logiciels applicatifs :

- Maple Version V 5.1,
- Matlab Version 5.2 avec Stixbox,
- MuPAD Pro Version 1.4.2,
- Scilab Version 2.5 avec Stixbox

Autres logiciels sous Linux : GNU Emacs, vi et Netscape.

Autres logiciels sous Windows : GNU Emacs, Notepad et Internet Explorer.

A.5 Boite à outils Stixbox pour Matlab

```
%
% A rudimentary statistics toolbox.
% Version 1.10, 9-Sep-98
% Copyright (c) Anders Holtsberg.
% Comments and suggestions to andersh@maths.lth.se.
%
% Revision 01-10-98 Mathematique Universite de Paris-Sud
%
% Distribution functions.
% dbeta      - Beta density function.
% dbinom    - Binomial probability function.
% dchisq    - Chisquare density function.
% df        - F density function.
% dgamma    - Gamma density function.
% dhypg     - Hypergeometric probability function.
% dnorm     - Normal density function.
% dt        - Student t density function.
%
% pbeta     - Beta distribution function.
% pbinom    - Binomial cumulative probability function.
% pchisq    - Chisquare distribution function.
```

```

% pf      - F distribution function.
% pgamma  - Gamma distribution function.
% phypg   - Hypergeometric cumulative probab. function.
% pks     - Kolmogorov Smirnov distribution function.
% pnorm   - Normal distribution function.
% pt      - Student t cdf.
%
% qbeta   - Beta inverse distribution function.
% qbinom  - Binomial inverse cdf.
% qchisq  - Chisquare inverse distribution function.
% qf      - F inverse distribution function.
% qgamma  - Gamma inverse distribution function.
% qhypg   - Hypergeometric inverse cdf.
% qnorm   - Normal inverse distribution function.
% qt      - Student t inverse distribution function.
%
% rbeta   - Rand. num. , beta distribution.
% rbinom  - Rand. num. , binomial distribution.
% rchisq  - Rand. num. , chisquare distribution.
% rexpweib - Rand. num. , exp and weibull distributions.
% rf      - Rand. num. , F distribution
% rgamma  - Rand. num. , gamma distribution.
% rgeom   - Rand. num. , geometric distribution.
% rhypg   - Rand. num. , hypergeometric distribution.
% rnorm   - Norm. rand. num. (use randn instead).
% rpoiss  - Rand. rand. num. , poisson distribution.
% rt      - Rand. num. , student t distribution.
%
%simulation of random variate by reject method
% rjbinom.m - Rand. num. , binomial distribution.
% rjgamma.m - Rand. numbers , gamma distribution.
% rjpoiss.m - Rand. rand. num. , poisson distribution.
%
%Logistic regression.
% logitfit - Fit a logistic regression model.
% lodds    - Log odds function.
% loddsinv - Inverse of log odds function.
%
% Various functions.
% bincoef  - Binomial coefficients.
% getdata  - Some famous multivariate data sets.
% quantile - Empirical quantile (percentile).
%
% Tests, confidence intervals, and model estimation.
% cmpmod   - Compare small linear model versus large one.
% ciquant  - Nonparametric confidence interval for quantile.
% kstwo    - Smirnov test for two samples.
% lsfit    - Fit a least squares model.
% lsselect - Select a predictor subset for regression.
% test1n   - Tests and confid. interv., 1 norm. sample.
% test2n   - Tests and confid. interv., 2 norm. samples.
%
% Graphics.
% qqnorm   - Normal probability paper.
% qqplot   - Plot empirical quantile vs empirical quantile.
% linreg   - Linear or polynomial regression, including plot.
% histo    - Plot a histogram (alternative to hist).

```

```
% plotsym - Plot with symbols.
% plotdens - Draw a nonparametric density estimate.
% identify - Identify points on a plot by mouse clicks.
% pairs - Pairwise scatter plots.
```

A.6 Boite à outils statistique de Matlab

Cette boite à outils de Matlab, bien plus complète que Stixbox, n'est pas sensée être disponible le jour de l'oral malheureusement...

```
Statistics Toolbox.
Version 3.0 (R12.1) 1-Sep-2000
```

```
Distributions.
```

```
Parameter estimation.
```

```
betafit - Beta parameter estimation.
binofit - Binomial parameter estimation.
expfit - Exponential parameter estimation.
gamfit - Gamma parameter estimation.
mle - Maximum likelihood estimation (MLE).
normfit - Normal parameter estimation.
poissfit - Poisson parameter estimation.
raylfit - Rayleigh parameter estimation.
unifit - Uniform parameter estimation.
weibfit - Weibull parametir estimation.
```

```
Probability density functions (pdf).
```

```
betapdf - Beta density.
binopdf - Binomial density.
chi2pdf - Chi square density.
exppdf - Exponential density.
fpdf - F density.
gampdf - Gamma density.
geopdf - Geometric density.
hygepdf - Hypergeometric density.
lognpdf - Lognormal density.
nbinpdf - Negative binomial density.
ncfpdf - Noncentral F density.
nctpdf - Noncentral t density.
ncx2pdf - Noncentral Chi-square density.
normpdf - Normal (Gaussian) density.
pdf - Density func. for a specified distrib.
poisspdf - Poisson density.
raylpdf - Rayleigh density.
tpdf - T density.
unidpdf - Discrete uniform density.
unifpdf - Uniform density.
weibpdf - Weibull density.
```

```
Cumulative Distribution functions (cdf).
```

```
betacdf - Beta cdf.
binocdf - Binomial cdf.
cdf - Specified cumulative distrib. function.
chi2cdf - Chi square cdf.
expcdf - Exponential cdf.
fcdf - F cdf.
gamcdf - Gamma cdf.
```

geocdf	- Geometric cdf.
hygecdf	- Hypergeometric cdf.
logncdf	- Lognormal cdf.
nbincdf	- Negative binomial cdf.
ncfcdf	- Noncentral F cdf.
nctcdf	- Noncentral t cdf.
ncx2cdf	- Noncentral Chi-square cdf.
normcdf	- Normal (Gaussian) cdf.
poisscdf	- Poisson cdf.
raylcdf	- Rayleigh cdf.
tcdf	- T cdf.
unidcdf	- Discrete uniform cdf.
unifcdf	- Uniform cdf.
weibcdf	- Weibull cdf.

Critical Values of Distribution functions.

betainv	- Beta inverse cumulative distrib. func.
binoinv	- Binomial inverse cumulative distrib. func.
chi2inv	- Chi square inverse cumulative distrib. func.
expinv	- Exponential inverse cumulative distrib. func.
finv	- F inverse cumulative distrib. func.
gaminv	- Gamma inverse cumulative distrib. func.
geoinv	- Geometric inverse cumulative distrib. func.
hygeinv	- Hypergeometric inverse cumulative distrib. func.
icdf	- Specified inverse cdf.
logninv	- Lognormal inverse cumulative distrib. func.
nbininv	- Negative binomial inverse distrib. func.
ncfinv	- Noncentral F inverse cumulative distrib. func.
nctinv	- Noncentral t inverse cumulative distrib. func.
ncx2inv	- Noncentral Chi-square inverse distrib. func.
norminv	- Normal (Gaussian) inv. cumulative distrib. func.
poissinv	- Poisson inverse cumulative distrib. func.
raylinv	- Rayleigh inverse cumulative distrib. func.
tin	- T inverse cumulative distrib. func.
unidinv	- Discrete unif. inv. cumulative distrib. func.
unifinv	- Uniform inverse cumulative distrib. func.
weibinv	- Weibull inverse cumulative distrib. func.

Random Number Generators.

betarnd	- Beta random numbers.
binornd	- Binomial random numbers.
chi2rnd	- Chi square random numbers.
exprnd	- Exponential random numbers.
frnd	- F random numbers.
gamrnd	- Gamma random numbers.
geornd	- Geometric random numbers.
hygernd	- Hypergeometric random numbers.
lognrnd	- Lognormal random numbers.
mvnrnd	- Multivariate normal random numbers.
mvtrnd	- Multivariate t random numbers.
nbinrnd	- Negative binomial random numbers.
ncfrnd	- Noncentral F random numbers.
nctrnd	- Noncentral t random numbers.
ncx2rnd	- Noncentral Chi-square random numbers.
normrnd	- Normal (Gaussian) random numbers.
poissrnd	- Poisson random numbers.
random	- Random numbers from specified distribution.

raylrnd - Rayleigh random numbers.
 trnd - T random numbers.
 unidrnd - Discrete uniform random numbers.
 unifrnd - Uniform random numbers.
 weibrnd - Weibull random numbers.

Statistics.

betastat - Beta mean and variance.
 binostat - Binomial mean and variance.
 chi2stat - Chi square mean and variance.
 expstat - Exponential mean and variance.
 fstat - F mean and variance.
 gamstat - Gamma mean and variance.
 geostat - Geometric mean and variance.
 hygestat - Hypergeometric mean and variance.
 lognstat - Lognormal mean and variance.
 nbinstat - Negative binomial mean and variance.
 ncfstat - Noncentral F mean and variance.
 nctstat - Noncentral t mean and variance.
 ncx2stat - Noncentral Chi-square mean and variance.
 normstat - Normal (Gaussian) mean and variance.
 poisstat - Poisson mean and variance.
 raylstat - Rayleigh mean and variance.
 tstat - T mean and variance.
 unidstat - Discrete uniform mean and variance.
 unifstat - Uniform mean and variance.
 weibstat - Weibull mean and variance.

Descriptive Statistics.

bootstrp - Bootstrap statistics for any function.
 corrcoef - Correlation coefficient.
 cov - Covariance
 crosstab - Cross tabulation.
 geomean - Geometric mean.
 grpstats - Summary statistics by group.
 harmmean - Harmonic mean.
 iqr - Interquartile range.
 kurtosis - Kurtosis.
 mad - Median Absolute Deviation.
 mean - Sample average (in matlab toolbox).
 median - 50th percentile of a sample.
 moment - Moments of a sample.
 nanmax - Maximum ignoring NaNs.
 nanmean - Mean ignoring NaNs.
 nanmedian - Median ignoring NaNs.
 nanmin - Minimum ignoring NaNs.
 nanstd - Standard deviation ignoring NaNs.
 nansum - Sum ignoring NaNs.
 prctile - Percentiles.
 range - Range.
 skewness - Skewness.
 std - Standard deviation ((in matlab toolbox).
 tabulate - Frequency table.
 trimmean - Trimmed mean.
 var - Variance (in matlab toolbox).

Linear Models.

anova1 - One-way analysis of variance.
 anova2 - Two-way analysis of variance.
 anovan - n-way analysis of variance.
 aoctool - Interactive tool for analysis of covariance.
 dummyvar - Dummy-variable coding.
 friedman - Friedman's test (nonparametric two-way anova).
 glmfit - Generalized linear model fitting.
 kruskalwallis - Kruskal-Wallis test (nonparametric one-way anova).
 leverage - Regression diagnostic.
 lscov - Least-squares estimates with known cov. matrix.
 manova1 - One-way multivariate analysis of variance.
 manovacluster - Draw clusters of group means for manova1.
 multcompare - Multiple comparisons of means and other estimates.
 polyconf - Polynomial eval. and confid. interv. estim.
 polyfit - Least-squares polynomial fitting.
 polyval - Predicted values for polynomial functions.
 rcoplot - Residuals case order plot.
 regress - Multivariate linear regression.
 regstats - Regression diagnostics.
 ridge - Ridge regression.
 robustfit - Robust regression model fitting.
 rstool - Multidimensional response surface visualization (RSM).
 stepwise - Interactive tool for stepwise regression.
 x2fx - Factor settings matrix (x) to design matrix (fx).

Nonlinear Models

nlinfit - Nonlinear least-squares data fitting (Newton).
 nlintool - Interactive gfx tool for prediction in nonlin. models.
 nlpredci - Confidence intervals for prediction.
 nlparci - Confidence intervals for parameters.
 nnls - Non-negative least-squares.

Cluster Analysis

pdist - Pairwise distance between observations.
 squareform - Square matrix formatted distance.
 linkage - Hierarchical cluster information.
 dendrogram - Generate dendrogram plot.
 inconsistent - Inconsistent values of a cluster tree.
 cophenet - Cophenetic coefficient.
 cluster - Construct clusters from LINKAGE output.
 clusterdata - Construct clusters from data.

Design of Experiments (DOE)

cordexch - D-optimal design (coordinate exchange algorithm).
 daugment - Augment D-optimal design.
 dcovary - D-optimal design with fixed covariates.
 ff2n - Two-level full-factorial design.
 fracfact - Two-level fractional factorial design.
 fullfact - Mixed-level full-factorial design.
 hadamard - Hadamard matrices (orthogonal arrays).
 rowexch - D-optimal design (row exchange algorithm).

Statistical Process Control (SPC)

capable - Capability indices.
 capaplot - Capability plot.
 ewmaplot - Exponentially weighted moving average plot.
 histfit - Histogram with superimposed normal density.

normspec - Plot normal density between specification limits.
 schart - S chart for monitoring variability.
 xbarplot - Xbar chart for monitoring the mean.

Principal Components Analysis

barttest - Bartlett's test for dimensionality.
 pcacov - Principal components from covariance matrix.
 pcares - Residuals from principal components.
 princomp - Principal components analysis from raw data.

Multivariate Statistics.

classify - Linear Discriminant Analysis.
 mahal - Mahalanobis distance.
 manova1 - One-way multivariate analysis of variance.

Hypothesis Tests.

ranksum - Wilcoxon rank sum test (independent samples).
 signrank - Wilcoxon sign rank test (paired samples).
 signtest - Sign test (paired samples).
 ztest - Z test.
 ttest - One sample t test.
 ttest2 - Two sample t test.

Distribution Testing

jbtest - Jarque-Bera test of normality
 kstest - Kolmogorov-Smirnov test for one sample
 kstest2 - Kolmogorov-Smirnov test for two samples
 lillietest - Lilliefors test of normality

Nonparametric Testing

friedman - Friedman's test (nonparametric two-way anova).
 kruskalwallis - Kruskal-Wallis test (nonparametric one-way anova).
 ranksum - Wilcoxon rank sum test (independent samples).
 signrank - Wilcoxon sign rank test (paired samples).
 signtest - Sign test (paired samples).

Statistical Plotting.

boxplot - Boxplots of a data matrix (one per column).
 cdfplot - Plot of empirical cumulative distribution function.
 fsurfht - Interactive contour plot of a function.
 gline - Point, drag and click line drawing on figures.
 gname - Interactive point labeling in x-y plots.
 gplotmatrix - Matrix of scatter plots grouped by a common variable.
 gscatter - Scatter plot of two variables grouped by a third.
 lsline - Add least-square fit line to scatter plot.
 normplot - Normal probability plot.
 qqplot - Quantile-Quantile plot.
 refcurve - Reference polynomial curve.
 refline - Reference line.
 surfht - Interactive contour plot of a data grid.
 weibplot - Weibull probability plot.

Statistics Demos.

aocool - Interactive tool for analysis of covariance.
 disttool - GUI tool for exploring probability distribution functions.
 glmdemo - Generalized linear model slide show.
 polytool - Interactive graph for prediction of fitted polynomials.

randtool - GUI tool for generating random numbers.
rsmdemo - Reaction simulation (DOE, RSM, nonlinear curve fitting).
robustdemo - Interactive tool to compare robust and least squares fits.

File Based I/O

tblread - Read in data in tabular format.
tblwrite - Write out data in tabular format to file.
tdfread - Read in text and numeric data from tab-delimited file.
caseread - Read in case names.
casewrite - Write out case names to file.

Annexe B

Petite galerie de portraits

Ce petit chapitre atypique est une invitation à la découverte de l'histoire de l'informatique et des probabilités. Tout le monde connaît la maxime de RABELAIS : « science sans conscience n'est que ruine de l'âme », j'ajouterais pour ma part que « sciences sans Histoire n'est que vide de l'âme ». Il me semble indispensable que les enseignants possèdent un minimum de culture scientifique et historique, génératrice d'enthousiasme et de structuration des esprits. Bien sûr, il faut se faisant se méfier des stéréotypes et des mythes trop caricaturaux, mais un cynique dirait qu'ils sont un mal nécessaire.

Le mot *hasard* vient de la racine arabe *zahara* de laquelle dérivent en arabe les mots « brille », « étincelle », « fleur », mais aussi « dé à jouer », « chance ». En arabe dialectal maghrébin, un « bou zahroun » est un chanceux. . . Le concept de hasard fascine depuis très longtemps les esprits. C'est sans doute au cours de la renaissance européenne que les probabilités combinatoires prennent la forme mathématisée que nous leur connaissons aujourd'hui avec PASCAL, BERNOULLI (Jacob) MOIVRE et BAYES par exemple – la combinatoire elle-même ayant fait l'objet de développements antérieurs par les civilisations chinoise et arabe tout au moins. Le dix-neuvième siècle, mécaniste et positiviste, ne verra souvent dans le hasard que notre ignorance des causes, comme l'exprime clairement LAPLACE dans ses écrits [Lap95a, Lap95b]. Il faudra attendre la fin du dix-neuvième siècle pour que la théorie moderne des probabilités prenne forme peu à peu, avec par exemple les travaux de MARKOV sur les traces de son maître CHEBYCHEV, de BERNSTEIN (Segi), et aussi des statisticiens EDGEWORTH, FISHER, PIERSON et NEYMAN. La théorie actuelle, construite à partir de la théorie de la mesure, prend progressivement corps pendant le vingtième siècle grâce notamment aux travaux fondamentaux de la « seconde vague », dont les plus illustres représentants sont sans doute WIENER, KOLMOGOROV, DOOB et LÉVY. Cette arrivée tardive dans les mathématiques « supérieures » explique probablement en partie le délaissement dont les probabilités ont fait l'objet de la part de toute une génération de mathématiciens, dont le groupe BOURBAKI faisait partie. La théorie de la mesure elle-même a mis du temps à s'imposer à la communauté mathématique. Cette crise d'adolescence fait désormais partie du passé. Les probabilités constituent aujourd'hui un pan important des mathématiques, riche et vivant, en connexion avec l'analyse, la topologie, la géométrie, la théorie des nombres et enfin la statistique mathématique, dont elle constitue le socle. De nos jours, on peut voir les probabilités appliquées comme une façon de mathématiser et quantifier l'information dont on dispose sur un problème donné, comme le font des traités récents, [App96] par exemple. Certains fondateurs de la statique du début du vingtième siècle partageaient déjà cette vision des choses. Enfin, la théorie de l'information de SHANNON, qui naît après la seconde guerre mondiale, fait le pont entre théorie des probabilités et informatique, cf. [App96, Dio97, CT91]. KOLMOGOROV et ses élèves se sont intéressés ensuite aux liens entre algorithmique, informatique, et probabilités.

Quant au mot *algorithme*, il vient du nom du mathématicien Abu Ja'far Muhammad ibn Musa AL-KHWARIZMI, l'un des pères de l'*algèbre*, autre mot d'origine arabe, cf. [Dje01]. Les algorithmes de calcul sont connus depuis l'antiquité, et l'on pense naturellement à EUCLIDE et à DIOPHANTE. Les systèmes mécaniques complexes ont été utilisés depuis très longtemps en astronomie. Cependant, les horloges, les automates et même la machine à calculer de PASCAL – la fameuse pascaline – ne sont pas vraiment programmables et n'exécutent que des algorithmes figés. Un grand pas est franchi avec le métier à tisser à cartes perforées que met au point JACQUARD à la toute fin du dix-huitième siècle. Charles BABBAGE conçoit quelques dizaines d'années plus tard sa machine à différences puis sa machine analytique – qui est même dotée d'une imprimante ! – que l'on peut considérer comme l'ancêtre des ordinateurs modernes. Ada LOVELACE a sans doute été la première à concevoir des programmes complexes pour ces machines, en collaboration avec BABBAGE. Mais ces chefs d'œuvres d'ingéniosité et de technologie sont beaucoup

trop coûteux à construire pour l'époque et resteront des curiosités inusitées. En revanche, des machines à calculer mécaniques puis électromécaniques plus simples vont faire progressivement leur apparition dans le commerce, la société IBM voit le jour à cette époque. Le début du vingtième siècle est marqué à la fois par un développement de l'électronique et des télécommunications et par un mouvement de refonte des mathématiques et de leur socle logique et axiomatique, on peut par exemple penser aux travaux fondamentaux de PEANO, RUSSEL, WHITEHEAD et GÖDEL [NNGG97]. Pendant la seconde guerre mondiale, des dizaines de mathématiciens, cruciverbistes et calculateurs prodiges sont mobilisés par les Alliés en Angleterre pour décrypter le code Enigma utilisé par les Nazis [Sin99]. À leur tête, un mathématicien de génie : Alan TURING. On lui doit en particulier la conceptualisation de la notion d'automate programmable, cf. [TG95]. Au carrefour de toutes ces avancées technologiques et conceptuelles, John Von NEUMANN, mathématicien au large spectre intellectuel, est sans doute l'un des précurseurs du premier ordinateur électronique ENIAC mis au point en 1947. La révolution électronique du transistor et de la miniaturisation des années suivantes va donner les machines informatiques que nous connaissons aujourd'hui, d'une complexité fascinante.

Les noms de fichiers HTML indiqués sous chaque illustration ci-dessous sont relatifs au répertoire /history/Mathematicians/ du serveur <http://www-groups.dcs.st-andrews.ac.uk> très riche sur l'histoire des mathématiques, et que je vous incite vivement à explorer.



Abu Ja'far Muhammad ibn Musa AL-KHWARIZMI
Al-Khwarizmi.html



Pierre-Simon de LAPLACE
Laplace.html



Blaise PASCAL
Pascal.html



Andrei Andreyevich MARKOV
Markov.html



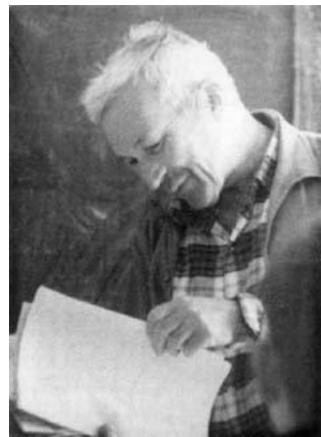
Sir Ronald Aylmer FISHER
Fisher.html



Claude Elwood SHANNON
Shannon.html



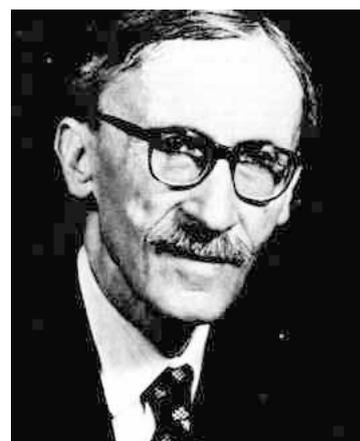
Norbert WIENER
Wiener_Norbert.html



Andrey Nikolaevich KOLMOGOROV
Kolmogorov.html



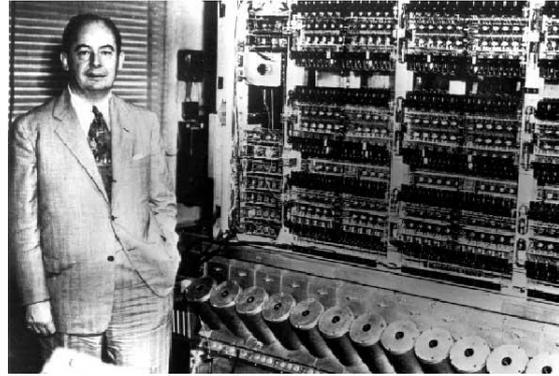
Alan Mathison TURING
Turing.html



Paul LÉVY
Levy_Paul.html



Joseph Leo DOOB
Doob.html



John VON NEUMANN aux côtés de l'ENIAC.
Von_Neumann.html

Annexe C

GNU Free Documentation License

À titre indicatif, une traduction française est fournie page 118).

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

C.1 Preamble

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

C.2 Applicability and definitions

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is

available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

C.3 Verbatim copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

C.4 Copying in quantity

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of

the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

C.5 Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

C.6 Combining documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

C.7 Collections of documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

C.8 Aggregation with independent works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on

account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

C.9 Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

C.10 Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

C.11 Future revisions of this licence

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time.

Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Addendum: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Annexe D

Licence de documentation libre GNU

Voici une traduction française indicative de la « GFDL » anglophone (cf. page 115).

Avertissement / Disclaimer

This is an unofficial translation of the GNU Free Documentation License into French. It was not published by the Free Software Foundation, and does not legally state the distribution terms for documentation that uses the GNU FDL—only the original English text of the GNU FDL does that. However, we hope that this translation will help French speakers understand the GNU FDL better.

Ceci est une traduction française non officielle de la Licence de documentation libre GNU. Elle n'a pas été publiée par la Free Software Foundation, et ne fixe pas légalement les conditions de redistribution des documents qui l'utilisent – seul le texte original en anglais le fait. Nous espérons toutefois que cette traduction aidera les francophones à mieux comprendre la FDL GNU.

Traduction française non officielle de la GFDL Version 1.1 (Mars 2000)

Copyright original : Copyright © 2000 Free Software Foundation, inc 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Pour la traduction :

- Version 1.0 FR (Jean-Luc Fortin, juillet 2000)
- Version 1.1 FR (Christian Casteyde, mars 2001)
- Version 1.1.1 FR (César Alexanian, mars 2001)
- Version 1.1.1r2 FR (Christian Casteyde et César Alexanian, juin 2001)

Chacun est libre de copier et de distribuer des copies conformes de cette Licence, mais nul n'est autorisé à la modifier.

D.1 Préambule

L'objet de cette Licence est de rendre tout manuel, livre ou autre document écrit « libre » au sens de la liberté d'utilisation, à savoir : assurer à chacun la liberté effective de le copier ou de le redistribuer, avec ou sans modifications, commercialement ou non. En outre, cette Licence garantit à l'auteur et à l'éditeur la reconnaissance de leur travail, sans qu'ils soient pour autant considérés comme responsables des modifications réalisées par des tiers.

Cette Licence est une sorte de « copyleft », ce qui signifie que les travaux dérivés du document d'origine sont eux-mêmes « libres » selon les mêmes termes. Elle complète la Licence Publique Générale GNU, qui est également une Licence copyleft, conçue pour les logiciels libres.

Nous avons conçu cette Licence pour la documentation des logiciels libres, car les logiciels libres ont besoin d'une documentation elle-même libre : un logiciel libre doit être accompagné d'un manuel garantissant les mêmes libertés que celles accordées par le logiciel lui-même. Mais cette Licence n'est pas limitée aux seuls manuels des logiciels ; elle peut être utilisée pour tous les documents écrits, sans distinction particulière relative au sujet traité ou au mode de publication. Nous recommandons l'usage de cette Licence principalement pour les travaux destinés à des fins d'enseignement ou devant servir de documents de référence.

D.2 Applicabilité et définitions

Cette Licence couvre tout manuel ou tout autre travail écrit contenant une notice de copyright autorisant la redistribution selon les termes de cette Licence. Le mot « Document » se réfère ci-après à un tel manuel ou travail. Toute personne en est par définition concessionnaire et est référencée ci-après par le terme « Vous ».

Une « Version modifiée » du Document désigne tout travail en contenant la totalité ou seulement une portion de celui-ci, copiée mot pour mot, modifiée et/ou traduite dans une autre langue.

Une « Section secondaire » désigne une annexe au Document, ou toute information indiquant les rapports entre l'auteur ou l'éditeur et le sujet (ou tout autre sujet connexe) du document, sans toutefois être en rapport direct avec le sujet lui-même (par exemple, si le Document est un manuel de mathématiques, une Section secondaire ne traitera d'aucune notion mathématique). Cette section peut contenir des informations relatives à l'historique du Document, des sources documentaires, des dispositions légales, commerciales, philosophiques, ou des positions éthiques ou politiques susceptibles de concerner le sujet traité.

Les « Sections inaltérables » sont des sections secondaires considérées comme ne pouvant être modifiées et citées comme telles dans la notice légale qui place le Document sous cette Licence.

Les « Textes de couverture » sont les textes courts situés sur les pages de couverture avant et arrière du Document, et cités comme tels dans la mention légale de ce Document.

Le terme « Copie transparente » désigne une version numérique du Document représentée dans un format dont les spécifications sont publiquement disponibles et dont le contenu peut être visualisé et édité directement et immédiatement par un éditeur de texte quelconque, ou (pour les images composées de pixels) par un programme de traitement d'images quelconque, ou (pour les dessins) par un éditeur de dessins courant. Ce format doit pouvoir être accepté directement ou être convertible facilement dans des formats utilisables directement par des logiciels de formatage de texte. Une copie publiée dans un quelconque format numérique ouvert mais dont la structure a été conçue dans le but exprès de prévenir les modifications ultérieures du Document ou dans le but d'en décourager les lecteurs n'est pas considérée comme une Copie Transparente. Une copie qui n'est pas « Transparente » est considérée, par opposition, comme « Opaque ».

Le format de fichier texte codé en ASCII générique et n'utilisant pas de balises, les formats de fichiers Texinfo ou LaTeX, les formats de fichiers SGML ou XML utilisant une DTD publiquement accessible, ainsi que les formats de fichiers HTML simple et standard, écrits de telle sorte qu'ils sont modifiables sans outil spécifique, sont des exemples de formats acceptables pour la réalisation de Copies Transparentes. Les formats suivants sont opaques : PostScript, PDF, formats de fichiers propriétaires qui ne peuvent être visualisés ou édités que par des traitements de textes propriétaires, SGML et XML utilisant des DTD et/ou des outils de formatage qui ne sont pas disponibles publiquement, et du code HTML généré par une machine à l'aide d'un traitement de texte quelconque et dans le seul but de la génération d'un format de sortie.

La « Page de titre » désigne, pour les ouvrages imprimés,

la page de titre elle-même, ainsi que les pages supplémentaires nécessaires pour fournir clairement les informations dont cette Licence impose la présence sur la page de titre. Pour les travaux n'ayant pas de Page de titre comme décrit ci-dessus, la « Page de titre » désigne le texte qui s'apparente le plus au titre du document et situé avant le texte principal.

D.3 Copies conformes

Vous pouvez copier et distribuer le Document sur tout type de support, commercialement ou non, à condition que cette Licence, la notice de copyright et la notice de la Licence indiquant que cette Licence s'applique à ce Document soient reproduits dans toutes les copies, et que vous n'y ajoutiez aucune condition restrictive supplémentaire. Vous ne pouvez pas utiliser un quelconque moyen technique visant à empêcher ou à contrôler la lecture ou la reproduction ultérieure des copies que vous avez créées ou distribuées. Toutefois, vous pouvez solliciter une rétribution en échange des copies. Si vous distribuez une grande quantité de copies, référez-vous aux dispositions de la section 3.

Vous pouvez également prêter des copies, sous les mêmes conditions que celles suscitées, et vous pouvez afficher publiquement des copies de ce Document.

D.4 Copies en nombre

Si vous publiez des copies imprimées de ce Document à plus de 100 exemplaires et que la Licence du Document indique la présence de Textes de couverture, vous devez fournir une couverture pour chaque copie, qui présente les Textes de couverture des première et dernière pages de couverture du Document. Les première et dernière pages de couverture doivent également vous identifier clairement et sans ambiguïté comme étant l'éditeur de ces copies. La première page de couverture doit comporter le titre du Document en mots d'importance et de visibilité égales. Vous pouvez ajouter des informations complémentaires sur les pages de couverture. Les copies du Document dont seule la couverture a été modifiée peuvent être considérées comme des copies conformes, à condition que le titre du Document soit préservé et que les conditions indiquées précédemment soient respectées.

Si les textes devant se trouver sur la couverture sont trop importants pour y tenir de manière claire, vous pouvez ne placer que les premiers sur la première page et placer les suivants sur les pages consécutives.

Si vous publiez plus de 100 Copies opaques du Document, vous devez soit fournir une Copie transparente pour chaque Copie opaque, soit préciser ou fournir avec chaque Copie opaque une adresse réseau publiquement accessible d'une Copie transparente et complète du Document, sans aucun ajout ou modification, et à laquelle tout le monde peut accéder en téléchargement anonyme et sans frais, selon des protocoles réseau communs et standards. Si vous choisissez cette dernière option, vous devez prendre les dispositions nécessaires, dans la limite du raisonnable, afin de garantir l'accès non restrictif à la Copie transparente durant une année pleine après la diffusion publique de la dernière Copie opaque (directement ou via vos revendeurs).

Nous recommandons, mais ce n'est pas obligatoire, que vous contactiez l'auteur du Document suffisamment tôt avant toute publication d'un grand nombre de copies, afin de lui permettre de vous donner une version à jour du Document.

D.5 Modifications

Vous pouvez copier et distribuer une Version modifiée du Document en respectant les conditions des sections 2 et 3 précédentes, à condition de placer cette Version modifiée sous la présente Licence, dans laquelle le terme « Document » doit être remplacé par les termes « Version modifiée », donnant ainsi l'autorisation de redistribuer et de modifier cette Version modifiée à quiconque en possède une copie. De plus, vous devez effectuer les actions suivantes dans la Version modifiée :

1. Utiliser sur la Page de titre (et sur la page de couverture éventuellement présente) un titre distinct de celui du Document d'origine et de toutes ses versions antérieures (qui, si elles existent, doivent être mentionnées dans la section « Historique » du Document). Vous pouvez utiliser le même titre si l'éditeur d'origine vous en a donné expressément la permission.
2. Mentionner sur la Page de titre en tant qu'auteurs une ou plusieurs des personnes ou entités responsables des modifications de la Version modifiée, avec au moins les cinq

principaux auteurs du Document (ou tous les auteurs s'il y en a moins de cinq).

3. Préciser sur la Page de titre le nom de l'éditeur de la Version modifiée, en tant qu'éditeur du Document.
4. Préserver intégralement toutes les notices de copyright du Document.
5. Ajouter une notice de copyright adjacente aux autres notices pour vos propres modifications.
6. Inclure immédiatement après les notices de copyright une notice donnant à quiconque l'autorisation d'utiliser la Version modifiée selon les termes de cette Licence, sous la forme présentée dans l'annexe indiquée ci-dessous.
7. Préserver dans cette notice la liste complète des Sections inaltérables et les Textes de couverture donnés avec la notice de la Licence du Document.
8. Inclure une copie non modifiée de cette Licence.
9. Préserver la section nommée « Historique » et son titre, et y ajouter une nouvelle entrée décrivant le titre, l'année, les nouveaux auteurs et l'éditeur de la Version modifiée, tels que décrits sur la Page de titre, ainsi qu'un descriptif des modifications apportées depuis la précédente version.
10. Conserver l'adresse réseau éventuellement indiquée dans le Document permettant à quiconque d'accéder à une Copie transparente du Document, ainsi que les adresses réseau indiquées dans le Document pour les versions précédentes sur lesquelles le Document se base. Ces liens peuvent être placés dans la section « Historique ». Vous pouvez ne pas conserver les liens pour un travail datant de plus de quatre ans avant la version courante ou si l'éditeur d'origine vous en accorde la permission.
11. Si une section « Dédicaces » ou une section « Remerciements » sont présentes, les informations et les appréciations concernant les contributeurs et les personnes auxquelles s'adressent ces remerciements doivent être conservées, ainsi que le titre de ces sections.
12. Conserver sans modification les Sections inaltérables du Document, ni dans leurs textes, ni dans leurs titres. Les numéros de sections ne sont pas considérés comme faisant partie du texte des sections.
13. Effacer toute section intitulée « Approbations ». Une telle section ne peut pas être incluse dans une Version modifiée.
14. Ne pas renommer une section existante sous le titre « Approbations » ou sous un autre titre entrant en conflit avec le titre d'une Section inaltérable.

Si la Version modifiée contient de nouvelles sections préliminaires ou de nouvelles annexes considérées comme des Sections secondaires et que celles-ci ne contiennent aucun élément copié à partir du Document, vous pouvez à votre convenance en désigner une ou plusieurs comme étant des Sections inaltérables. Pour ce faire, ajoutez leurs titres dans la liste des Sections inaltérables au sein de la notice de Licence de la version Modifiée. Ces titres doivent être distincts des titres des autres sections.

Vous pouvez ajouter une section nommée « Approbations » à condition que ces approbations ne concernent que les modifications ayant donné naissance à la Version modifiée (par exemple, comptes rendus de revue du document ou acceptation du texte par une organisation le reconnaissant comme étant la définition d'un standard).

Vous pouvez ajouter un passage comprenant jusqu'à cinq mots en première page de couverture, et jusqu'à vingt-cinq mots en dernière page de couverture, à la liste des Textes de couverture de la Version modifiée. Il n'est autorisé d'ajouter qu'un seul passage en première et en dernière pages de couverture par personne ou groupe de personnes ou organisation ayant contribué à la modification du Document. Si le Document comporte déjà un passage sur la même couverture, ajouté en votre nom ou au nom de l'organisation au nom de laquelle vous agissez, vous ne pouvez pas ajouter de passage supplémentaire ; mais vous pouvez remplacer un ancien passage si vous avez expressément obtenu l'autorisation de l'éditeur de celui-ci.

Cette Licence ne vous donne pas le droit d'utiliser le nom des auteurs et des éditeurs de ce Document à des fins publicitaires ou pour prétendre à l'approbation d'une Version modifiée.

D.6 Fusion de documents

Vous pouvez fusionner le Document avec d'autres documents soumis à cette Licence, suivant les spécifications de la section 4 pour les Versions modifiées, à condition d'inclure dans le document résultant toutes les Sections inaltérables des documents

originaux sans modification, et de toutes les lister dans la liste des Sections inaltérables de la notice de Licence du document résultant de la fusion.

Le document résultant de la fusion n'a besoin que d'une seule copie de cette Licence, et les Sections inaltérables existant en multiples exemplaires peuvent être remplacées par une copie unique. S'il existe plusieurs Sections inaltérables portant le même nom mais de contenu différent, rendez unique le titre de chaque section en ajoutant, à la fin de celui-ci, entre parenthèses, le nom de l'auteur ou de l'éditeur d'origine, ou, à défaut, un numéro unique. Les mêmes modifications doivent être réalisées dans la liste des Sections inaltérables de la notice de Licence du document final.

Dans le document résultant de la fusion, vous devez rassembler en une seule toutes les sections « Historique » des documents d'origine. De même, vous devez rassembler les sections « Remerciements » et « Dédicaces ». Vous devez supprimer toutes les sections « Approbations ».

D.7 Regroupement de documents

Vous pouvez créer un regroupement de documents comprenant le Document et d'autres documents soumis à cette Licence, et remplacer les copies individuelles de cette Licence des différents documents par une unique copie incluse dans le regroupement de documents, à condition de respecter pour chacun de ces documents l'ensemble des règles de cette Licence concernant les copies conformes.

Vous pouvez extraire un document d'un tel regroupement et le distribuer individuellement sous couvert de cette Licence, à condition d'y inclure une copie de cette Licence et d'en respecter l'ensemble des règles concernant les copies conformes.

D.8 Agrégation avec des travaux indépendants

La compilation du Document ou de ses dérivés avec d'autres documents ou travaux séparés et indépendants sur un support de stockage ou sur un média de distribution quelconque ne représente pas une Version modifiée du Document tant qu'aucun copyright n'est déposé pour cette compilation. Une telle compilation est appelée « agrégat » et cette Licence ne s'applique pas aux autres travaux indépendants compilés avec le Document s'ils ne sont pas eux-mêmes des travaux dérivés du Document.

Si les exigences de la section 3 concernant les Textes de couverture sont applicables à ces copies du Document, et si le Document représente un volume inférieur à un quart du volume total de l'agrégat, les Textes de couverture du Document peuvent être placés sur des pages de couverture qui n'encadrent que le Document au sein de l'agrégat. Dans le cas contraire, ils doivent apparaître sur les pages de couverture de l'agrégat complet.

D.9 Traduction

La traduction est considérée comme une forme de modification, vous pouvez donc distribuer les traductions du Document selon les termes de la section 4. Vous devez obtenir l'autorisation spéciale des auteurs des Sections inaltérables pour les remplacer par des traductions, mais vous pouvez inclure les traductions des Sections inaltérables en plus des textes originaux. Vous pouvez inclure une traduction de cette Licence à condition d'inclure également la version originale en anglais. En cas de contradiction entre la traduction et la version originale en anglais, c'est cette dernière qui prévaut.

D.10 Révocation

Vous ne pouvez pas copier, modifier, sous-licencier ou distribuer le Document autrement que selon les termes de cette Licence. Tout autre acte de copie, modification, sous-Licence ou distribution du Document est sans objet et vous prive automatiquement des droits que cette Licence vous accorde. En revanche, les personnes qui ont reçu de votre part des copies ou les droits sur le document sous couvert de cette Licence ne voient pas leurs droits révoqués tant qu'elles en respectent les principes.

D.11 Révisions futures de cette licence

La Free Software Foundation peut publier de temps en temps de nouvelles versions révisées de cette Licence. Ces nouvelles versions seront semblables à la présente version dans l'esprit, mais pourront différer sur des points particuliers en fonction de nouvelles questions ou nouveaux problèmes. Voyez <http://www.gnu.org/copyleft/> pour plus de détails.

Chaque version de cette Licence est dotée d'un numéro de version distinct. Si un Document spécifie un numéro de version particulier de cette Licence, et porte la mention « ou toute autre version ultérieure », vous pouvez choisir de suivre les termes de la version spécifiée ou ceux de n'importe quelle version ultérieure publiée par la Free Software Foundation. Si aucun numéro de version n'est spécifié, vous pouvez choisir n'importe quelle version officielle publiée par la Free Software Foundation.

Comment utiliser cette Licence pour vos documents

Pour utiliser cette Licence avec un document que vous avez écrit, incorporez une copie du texte de cette Licence en anglais et placez le texte ci-dessous juste après la page de titre :

Copyright © ANNÉE VOTRE NOM

Permission vous est donnée de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU Free Documentation License, Version 1.1 ou ultérieure publiée par la Free Software Foundation ; avec les sections inaltérables suivantes :

LISTE DES TITRES DES SECTIONS INALTÉRABLES

Avec le texte de première page de couverture suivant :

TEXTE DE PREMIÈRE PAGE DE COUVERTURE

Avec le texte de dernière page de couverture suivant :

TEXTE DE DERNIÈRE PAGE DE COUVERTURE

Une copie de cette Licence est incluse dans la section appelée GNU Free Documentation License de ce document.

Si votre Document ne comporte pas de section inaltérable, de textes de première et dernière pages de couverture, veuillez insérer les mentions suivantes dans les sections adéquates :

- pas de section inaltérable
- pas de texte de première page de couverture
- pas de texte de dernière page de couverture

Vous pouvez également fournir une traduction de la Licence GNU FDL dans votre document, mais celle-ci ne doit pas remplacer la version anglaise. La section intitulée GNU Free Documentation License doit contenir la version anglaise de la Licence GNU FDL, c'est la seule qui fait foi.

Si votre Document contient des exemples non triviaux de code programme, nous recommandons de distribuer ces exemples en parallèle sous Licence GNU General Public License, qui permet leur usage dans les logiciels libres.

Table des figures

1	Ce que donne <code>plot(cumsum(randn(1,1000))./[1:1000], 'b-')</code>	7
2	Comparaison des temps de calcul de la fonction factorielle (Octave 2.1).	14
1.1	Simulation de la loi exponentielle de paramètre 1/2 au moyen de sa fonction de répartition.	20
1.2	Lois fractales associées à un schéma de Bernoulli.	23
1.3	Simulation d'une trajectoire d'un processus de Poisson simple.	25
1.4	Simulation d'une trajectoire renormalisée d'un processus de Poisson simple.	26
1.5	Simulation de la loi normale par méthode « polaire »	27
1.6	Simulation d'une trajectoire d'un processus gaussien à partir de sa fonction de covariance (mouvement brownien ici).	31
1.7	Simulation d'une trajectoire d'un mouvement brownien.	32
1.8	Simulation d'une trajectoire d'un processus à temps de sauts exponentiels et à amplitudes de sauts gaussiennes.	35
1.9	Schéma d'un réseau	36
2.1	Illustration du théorème de Weierstrass avec les polynômes de Bernstein.	42
2.2	Illustration du TCL (lois uniformes).	43
2.3	Illustration du théorème de Donsker.	45
2.4	Convergence vers la distribution de Komogorov-Smirnov pour le théorème de Glivenko-Cantelli (loi normale).	47
2.5	Comparaison des approximations par TCL et PGD pour un écart à la moyenne de la loi binomiale.	49
2.6	Illustration de la loi du log. itéré de Strassen (distance à $[-\sigma, +\sigma]$).	50
2.7	Approximation de la loi normale par TCL (lois uniformes).	53
2.8	Illustration du théorème de Wigner.	54
3.1	Comparaison des bornes des intervalles de confiance lors de l'estimation de la moyenne d'une loi de Bernoulli par la moyenne empirique.	58
3.2	Approximation de π par méthodes de Monte-Carlo.	61
4.1	Reduction d'entropie et de DFS empirique/théorique lors d'une simulation d'un vecteur gaussien.	69
5.1	Comparaison des mesures ivariantes théorique et simulée pour le modèle d'Ehrenfest.	73
7.1	Probabilité d'extinction empirique	85
8.1	Relations entre les différents types de convergences de v.a.	97

Liste des tableaux

1	Quelques variables spéciales de Matlab.	2
1.1	Générateurs aléatoires de la bibliothèque Stixbox.	38
2.1	Quelques transformées de Cramér	48
8.1	Tableau récapitulatif de quelques lois usuelles	98
8.2	Quelques fonctions de répartition	99
A.1	Quelques liens sur la toile mondiale	101

Liste des programmes

tut-elem.m	2
tut-elem2.m	3
tut-mat.m	3
tut-mat2.m	5
tut-rndgfx.m	6
chaines	7
tut-printf.m	8
tut-io.m	8
tut-iof.m	8
tut-logics.m	9
tut-ctrl.m	10
tut-fun.m	11
firstdigit.m	11
empcdf.m	12
fact.m	13
tut-ifun.m	13
tictoc.m	13
tut-rond.m	15
simexp.m	19
rdist.m	20
randiscr.m	21
loisfrac.m	22
simchma.m	24
simps-short.m	25
simnorm.m	26
simvg.m	28
simmb.m	30
simmb2.m	31
rdirichlet.m	34
rmatmar.m	34
simpscg-short.m	34
fiab.m	36
ber.m	41
bertest.m	41
tcl1.m	42
donsker.m	44
pontbrown.m	46
tclvspgd.m	48
lli.m	50
llid.m	50
tcl2.m	52
wigner.m	54
iconfcompare.m	57
montecarlo.m	60
montecarlo2.m	60
gauss.m	65

gauss2.m	65
normproj.m	68
erdfs.m	68
ehrenfest.m	72
ruine.m	75
fileattente.m	79
gwpop-discr.m	83
gwpop-geom.m	84
gwpop-histo.m	84
gwpop-pevt.m	84

Bibliographie

- [AN72] K. B. ATHREYA et P. E. NEY – *Branching processes*, Springer-Verlag, New York, 1972, Die Grundlehren der mathematischen Wissenschaften, Band 196.
- [App96] D. APPLEBAUM – *Probability and information, an integrated approach*, Cambridge University Press, Cambridge, 1996.
- [Bak] D. BAKRY – « Cours de probabilités de maîtrise », <http://www.lsp.ups-tlse.fr/Bakry/>.
- [Bas95] R. F. BASS – *Probabilistic techniques in analysis*, Springer-Verlag, New York, 1995.
- [BC98] J.-C. BERTEIN et R. CESCHI – *Processus stochastiques et filtrage de Kalman*, Editions Hermès, Paris, 1998.
- [BDY00] B. BRU, W. DOEBLIN et M. YOR – *Sur l'équation de Kolmogoroff, par W. Doebelin*, Éditions Elsevier, Paris, 2000, C. R. Acad. Sci. Paris Sér. I Math. **331** (2000), Special Issue.
- [Bes] P. C. BESSE – « Cours de modélisation statistique », <http://www.lsp.ups-tlse.fr/Besse/>.
- [BL98] P. BARBE et M. LEDOUX – *Probabilités*, De la licence à l'agrégation, Belin, 1998.
- [Bon95] J.-L. BON – *Fiabilité des systèmes - Méthodes mathématiques*, Masson, 1995.
- [Bor95] V. S. BORKAR – *Probability theory, an advanced course*, Springer-Verlag, New York, 1995.
- [Bou86] N. BOULEAU – *Probabilités de l'ingénieur, variables aléatoires et simulation*, Hermann, 1986.
- [Bou88] — , *Processus stochastiques et applications*, Hermann, 1988.
- [CDGC80] M. COTTRELL, C. DUHAMEL et V. GENON-CATALOT – *Exercices de probabilités*, Collections DIA, 1980, Avec rappels de cours. Préface de Gabriel Ruget.
- [CT91] T. M. COVER et J. A. THOMAS – *Elements of information theory*, John Wiley & Sons Inc., New York, 1991, A Wiley-Interscience Publication.
- [CT97] C. COCOZZA-THIVENT – *Processus stochastiques et fiabilité des systèmes*, Springer-Verlag, Berlin, 1997.
- [DCD82a] D. DACUNHA-CASTELLE et M. DUFLO – *Exercices de probabilités et statistiques. Tome 1*, Masson, Paris, 1982, Problèmes à temps fixe.
- [DCD82b] — , *Probabilités et statistiques. Tome 1*, Masson, Paris, 1982, Problèmes à temps fixe.
- [DCD83] — , *Probabilités et statistiques. Tome 2*, Masson, Paris, 1983, Problèmes à temps mobile.
- [Dio97] E. DION – *Invitation à la théorie de l'information*, Points sciences, Éditions du seuil, 1997.
- [Dje01] A. DJEBBAR – *Une histoire de la science arabe*, Points sciences, vol. 144, Éditions du seuil, 2001, Entretien avec Jean Rosmorduc.
- [Duf96] M. DUFLO – *Algorithmes stochastiques*, Springer-Verlag, Berlin, 1996.
- [Fel68] W. FELLER – *An introduction to probability theory and its applications. Vol. I*, John Wiley & Sons Inc., New York, 1968.
- [Fel71] — , *An introduction to probability theory and its applications. Vol. II.*, John Wiley & Sons Inc., New York, 1971.
- [GBS69] B. V. GNEDENKO, Y. K. BELYAYEV et A. D. SOLOVYEV – *Mathematical methods of reliability theory*, Academic Press, New York, 1969, Translated by Scripta Technica, Inc, Translation edited by Richard E. Barlow, Probability and Mathematical Statistics, Vol. 6.
- [Har63] T. E. HARRIS – *The theory of branching processes*, Springer-Verlag, Berlin, 1963.
- [Har69] T. E. HARRIS – *Les processus de ramification. Applications à la physique et à la biologie*, Dunod, Paris, 1969.

- [Knu73] D. E. KNUTH – *The art of computer programming. Volume 3*, Computer Science and Information Processing, Addison-Wesley, 1973, Sorting and searching.
- [Knu75] — , *The art of computer programming*, Computer Science and Information Processing, Addison-Wesley, 1975, Volume 1: Fundamental algorithms.
- [Knu76] — , *The state of the art of computer programming*, Computer Science Department, School of Humanities and Sciences, Stanford University, Stanford, Calif., 1976, Errata to *The art of computer programming, Vols. 1 and 2*, Addison-Wesley.
- [Knu81] — , *The art of computer programming. Vol. 2*, Computer Science and Information Processing, Addison-Wesley, 1981, Seminumerical algorithms.
- [KS91] I. KARATZAS et S. E. SHREVE – *Brownian motion and stochastic calculus*, Springer-Verlag, New York, 1991.
- [Lap95a] P.-S. LAPLACE – *Théorie analytique des probabilités. Vol. I*, Éditions Jacques Gabay, Paris, 1995, Introduction: Essai philosophique sur les probabilités. Livre I: Du calcul des fonctions génératrices., Réédition de la quatrième édition de 1819.
- [Lap95b] — , *Théorie analytique des probabilités. Vol. II*, Éditions Jacques Gabay, Paris, 1995, Livre II: Théorie générale des probabilités. Suppléments., Réédition de la troisième édition de 1820.
- [Law91] G. F. LAWLER – *Intersections of random walks*, Birkhäuser Boston Inc., Boston, MA, 1991.
- [Let] G. LETAC – « Cours de probabilités de premier cycle », <http://www.lsp.ups-tlse.fr/Fp/Letac/>.
- [Lic02] A. LICHNEVSKI (éd.) – *Modélisation mathématique : un autre regard – Issu de l'épreuve de modélisation de l'agrégation*, Scopos, no. 16, Springer, 2002.
- [LL97] D. LAMBERTON et B. LAPEYRE – *Introduction au calcul stochastique appliqué à la finance*, second éd., Ellipses Édition Marketing, Paris, 1997.
- [LPS98] B. LAPEYRE, É. PARDOUX et R. SENTIS – *Méthodes de Monte-Carlo pour les équations de transport et de diffusion*, Springer-Verlag, Berlin, 1998.
- [Mil01] X. MILHAUD – *Statistique*, De la licence à l'agrégation, Belin, 2001.
- [MPB98] L. MAZLIAK, P. PRIOURET et P. BALDI – *Martingales et chaînes de Markov*, Hermann, 1998.
- [NNGG97] E. NAGEL, J. NEWMAN, K. GÖDEL et J.-Y. GIRARD – *Le théorème de Gödel*, Points - Sciences, Éditions du Seuil, 1997.
- [Nor98] J. R. NORRIS – *Markov chains*, Cambridge University Press, Cambridge, 1998, Reprint of 1997 original.
- [Pet95] V. V. PETROV – *Limit theorems of probability theory*, The Clarendon Press Oxford University Press, New York, 1995, Sequences of independent random variables, Oxford Science Publications.
- [Rev97] D. REVUZ – *Probabilités*, Hermann, 1997.
- [RM96] R. RASHED et R. MORELON (éds.) – *Encyclopedia of the history of Arabic science. Vol. 1–3*, Routledge, London, 1996.
- [RM98] R. Y. RUBINSTEIN et B. MELAMED – *Modern simulation and modeling*, John Wiley & Sons Inc., New York, 1998, A Wiley-Interscience Publication.
- [Ros00] S. M. ROSS – *Introduction to probability models*, seventh éd., Harcourt/Academic Press, San Diego, CA, 2000.
- [Rub81] R. Y. RUBINSTEIN – *Simulation and the Monte Carlo method*, John Wiley & Sons Inc., New York, 1981, Wiley Series in Probability and Mathematical Statistics.
- [Rud75] W. RUDIN – *Analyse réelle et complexe*, Masson, Paris, 1975.
- [Rue89] A. RUEGG – *Processus stochastiques*, Presses Polytechniques Romandes, Lausanne, 1989, Avec applications aux phénomènes d'attente et de fiabilité. [With applications to queueing and reliability].
- [Rug00] C. RUGET (éd.) – *Mathématiques en situation – Issues de l'épreuve de modélisation de l'agrégation*, Scopos, Springer, 2000.
- [Sap89] G. SAPORTA – *Probabilités, analyse des données et statistique*, Technip, 1989.

- [Sin99] S. SINGH – *Histoire des codes secrets*, Jean-Claude Latès, 1999.
- [TDM93] R. TOMASSONE, C. DERVIN et J.-P. MASSON – *Biométrie, modélisation de phénomènes biologiques*, Masson, 1993.
- [TG95] A. TURING et J.-Y. GIRARD – *La machine de Turing*, Points - Sciences, Éditions du Seuil, 1995.
- [TK98] H. M. TAYLOR et S. KARLIN – *An introduction to stochastic modeling*, third éd., Academic Press Inc., San Diego, CA, 1998.
- [TL90] P. TASSI et S. LEGAIT – *Théorie des probabilités en vue des applications statistiques*, Éditions Technip, Paris, 1990.
- [Tou99] P. TOULOUSE – *Thèmes de probabilités et statistique*, Dunod, 1999, INTERDIT À L'ORAL.
- [VK85] K. VO KAC – *Estimation et tests*, Ellipses, 1985.
- [Wil91] D. WILLIAMS – *Probability with martingales*, Cambridge University Press, Cambridge, 1991.

