

Simulation de variables aléatoires

4 avril 2018

Pierre Rousselin

La dernière version de ce document est disponible sur le site :

<https://www.math.univ-paris13.fr/~rousselin/>

N'hésitez pas à me faire part de vos commentaires et à me signaler les coquilles, erreurs et imprécisions à l'adresse indiquée sur ce site.

1 Introduction

Bien que le sujet soit intéressant, nous ne nous attarderons pas trop sur les les questions que posent la simulation du hasard par un ordinateur. Pour en savoir plus sur ce sujet, voir par exemple le début du chapitre 1 de [BC07].

On suppose donc disposer d'un « bon » programme renvoyant lors de ses appels successifs des nombres réels aléatoires *indépendants* suivant la loi uniforme sur $[0; 1]$. Les problèmes surgissent dès la définition : d'une part les nombres réels ne sont pas représentables en machine. On travaille le plus souvent avec des nombres à virgule flottante en double précision qui sont en nombre fini, bien qu'important. D'autre part, l'ordinateur est une machine *déterministe*.

On dispose donc d'algorithmes permettant de générer des suites de nombres, de façon déterministe, qui « ressemblent » aux premières décimales d'une suite de réalisations indépendantes de la loi uniforme sur $[0; 1]$. Le choix de tel ou tel algorithme dépend de l'utilisation que l'on en fait.

Pour la simulation, l'algorithme le plus utilisé aujourd'hui est sans doute « Mersenne Twister 19937 », mais son utilisation est fortement déconseillée lorsqu'on souhaite, par exemple, fabriquer une clé privée aléatoirement car il est jugé trop « prévisible ». En revanche, il passe un grand nombre de tests statistiques et est assez performant.

Le logiciel Scilab est assez généreux en ce qui concerne la simulation. Les deux principales fonctions que nous utiliserons seront `rand`, le générateur basique de Scilab, qui utilise un générateur congruentiel simple et permet de simuler la loi normale centrée réduite et `grand` une gigantesque fonction permettant de choisir son générateur parmi une liste assez longue (avec Mersenne Twister 19937 comme choix par défaut) et de simuler un nombre important de lois de probabilités.

Pour la suite de ce chapitre, nous utiliserons uniquement la fonction prédéfinie `rand` et nous travaillerons sous l'hypothèse que ses appels successifs fournissent des réalisations indépendantes de variables uniformes sur $[0; 1]$. En général, on peut imposer un état initial au générateur de nombres aléatoires en choisissant une *graine* (en anglais, *seed*). Voyons le comportement par défaut de Scilab.

Exercice 1.1. `rand`, `rand("seed", s)`, `rand(m, n)`, `histplot`

1. Lancer Scilab et taper `rand` puis « Entrée » trois fois. Noter les résultats.
2. Quitter Scilab et recommencer l'expérience précédente.
3. Quitter Scilab (promis c'est la dernière fois), et entrer la commande `rand("seed")` dans la console. Taper `rand` puis redemander la graine comme précédemment.
4. Entrer la commande `rand("seed", 0)`, puis refaire un appel à `rand`
5. Faire le bilan. Quelle est l'influence de la graine ? Comment la modifier ? Comment retrouver l'état initial par défaut du générateur `rand` ?
6. Que fait la commande `rand(3, 5)` ?

7. Que fait la commande `histplot(linspace(0, 1, 10), rand(1, 10000))` ? Consulter l'aide de `histplot`
8. Entrer plusieurs fois de suite la commande `getdate()`, et aller voir sa documentation.
9. Que fait le script suivant ?

```
x = getdate("s");
rand("seed", x);
```

2 Lois uniformes

Nous commençons par simuler la loi uniforme sur un intervalle réel ou entier, en donnant une définition qui englobe ces deux cas et bien d'autres encore. Nous en profitons au passage pour donner quelques rappels sur le calcul des moments de ces lois.

Définition 1. Soit (E, \mathcal{E}, μ) un espace mesuré, et $A \in \mathcal{E}$ tel que $0 < \mu(A) < \infty$. La loi uniforme sur A (relativement à la mesure μ) est la loi de probabilité ν_A définie par :

$$\forall B \in \mathcal{E}, \quad \nu_A(B) = \frac{\mu(A \cap B)}{\mu(A)}.$$

Exemple 2. 1. Prenons $E = \mathbb{R}$, $\mathcal{E} = \mathcal{B}(\mathbb{R})$ et pour μ , la mesure de Lebesgue. La loi uniforme sur l'intervalle $[a; b]$ (ou $]a; b]$, ou $[a; b[$, ou $]a; b[$, cela ne change rien) est la loi de densité $\frac{1_{[a;b]}}{b-a}$ par rapport à la mesure de Lebesgue.

2. On peut, de même, caractériser la loi uniforme sur un pavé de \mathbb{R}^d de la forme $\prod_{i=1}^d [a_i; b_i]$ par sa densité par rapport à la mesure de Lebesgue :

$$(x_1, x_2, \dots, x_d) \mapsto \prod_{i=1}^d \frac{1_{[a_i; b_i]}(x_i)}{b_i - a_i}.$$

3. Prenons $E = \mathbb{Z}$, muni de la tribu discrète $2^{\mathbb{Z}}$ et pour μ , la mesure de comptage (c'est-à-dire, pour tout $A \subset \mathbb{Z}$, $\mu(A) = \text{Card}(A)$). On peut définir ainsi la mesure uniforme sur toute partie finie non vide de \mathbb{Z} . Ceci est encore valable en remplaçant \mathbb{Z} par tout ensemble dénombrable.

Exercice 2.1.

Soient a et b deux réels avec $a < b$. Soit U une variable aléatoire de loi uniforme sur $[a; b]$. Montrer que l'espérance de U vaut $\frac{a+b}{2}$ et que sa variance vaut $\frac{(b-a)^2}{12}$.

Exercice 2.2.

En utilisant la fonction `rand`, écrire une fonction `runif(m,n,a,b)` qui renvoie une matrice de taille $m \times n$ dont les coefficients sont des réalisations de la loi uniforme sur $[a; b]$.

Exercice 2.3.

Soient a et b deux entiers, avec $a < b$. Notons $\llbracket a; b \rrbracket$ l'ensemble $\{a, a+1, \dots, b\}$ et $n = b - a + 1$. Soit X une variable aléatoire de loi uniforme sur $\llbracket a; b \rrbracket$.

1. Montrer que la fonction génératrice des moments (aussi appelée transformée de Laplace) de X est égale à

$$M_X(t) := \mathbb{E} \left[e^{tX} \right] = \frac{1}{n} \exp \left(\frac{a+b}{2} t \right) \frac{\sinh \left(\frac{n}{2} t \right)}{\sinh \left(\frac{1}{2} t \right)}.$$

2. En faisant un développement limité de M_X à l'ordre 2 en 0, calculer les moments d'ordre 1 et 2 de X (c'est-à-dire $\mathbb{E}[X]$ et $\mathbb{E}[X^2]$), puis montrer que la variance de X vaut $\frac{n^2-1}{12}$.

3. En déduire une formule permettant de calculer la somme des carrés des n premiers entiers, pour $n \geq 1$. Donner une méthode pour calculer la somme des n premiers cubes.

Remarque 3. On pouvait aussi utiliser la fonction génératrice pour résoudre l'exercice 2.1. En utilisant la fonction sinus cardinal hyperbolique, définie par

$$\operatorname{sinc}(x) = \frac{\sinh(x)}{x} = \frac{e^x - e^{-x}}{2x},$$

on trouve, pour U uniforme sur $[a; b]$,

$$M_U(t) = \exp\left(\frac{a+b}{2}t\right) \operatorname{sinc}\left(\frac{b-a}{2}t\right).$$

Exercice 2.4.

On considère deux entiers a et b avec $a < b$. En utilisant la fonction `rand`, écrire une fonction `runif_int(m, n, a, b)` qui renvoie une matrice de taille $m \times n$ dont les coefficients sont des réalisations indépendantes de la loi uniforme sur $\{a, a+1, \dots, b\}$.

3 Lois discrètes

Pour continuer, nous allons simuler les principales lois de référence sur \mathbb{N} et en rappeler les définitions et les principales propriétés.

Définition 4 (Bernoulli, binomiale, géométrique). Soit $p \in [0; 1]$, $n \in \{1, 2, \dots\}$ et X une variable aléatoire. On dit que X suit :

1. la loi de Bernoulli de paramètre p (notation : $X \sim \mathcal{B}(1, p)$) si

$$\mathbb{P}(X = 1) = 1 - \mathbb{P}(X = 0) = p;$$

2. la loi binomiale de paramètres n et p (notation : $X \sim \mathcal{B}(n, p)$) si X a la même loi que

$$Y_1 + Y_2 + \dots + Y_n,$$

où Y_1, Y_2, \dots, Y_n , sont n variables de Bernoulli indépendantes de même paramètre p ;

3. la loi géométrique de paramètre p si X a la même loi que

$$\inf \{n \geq 1 \mid Y_n = 1\},$$

où $(Y_n)_{n \geq 1}$ est une famille de variables de Bernoulli indépendantes de même paramètre p .

Exercice 3.1. Loi binomiale

- Soient X et Y deux variables aléatoires indépendantes de carré intégrable. Montrer que $\operatorname{Var}[X + Y] = \operatorname{Var}[X] + \operatorname{Var}[Y]$ et que pour tout réel a , $\operatorname{Var}[aX] = a^2 \operatorname{Var}[X]$.
- Soit X une variable aléatoire de loi $\mathcal{B}(n, p)$, où n est un entier naturel non nul et $p \in [0; 1]$. Démontrer les formules suivantes :

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k = 0, 1, \dots, n, \quad (1)$$

$$\mathbb{E}[X] = np, \quad (2)$$

$$\operatorname{Var}[X] = np(1-p). \quad (3)$$

Remarque 5. La réciproque de la première formule de la question 1 est, sauf cas très particulier (voir les vecteurs gaussiens) complètement fautive. Pour un contre-exemple, considérer le cas où ε est une variable uniforme sur $\{-1; 1\}$ et X suit la loi normale centrée réduite. Alors, si $Y = \varepsilon \times X$, on peut montrer (exercice) que Y suit également la loi normale centrée réduite et que $\operatorname{Var}[X + Y] = \operatorname{Var}[X] + \operatorname{Var}[Y]$ alors que X et Y ne sont pas indépendantes.

Exercice 3.2. Simulation de la loi binomiale

1. En utilisant la fonction `rand`, écrire une fonction `rbernoulli(m,n, p)` qui renvoie une matrice de taille $m \times n$ dont les coefficients sont des réalisations indépendantes de la loi de Bernoulli de paramètre p .
2. En utilisant la définition de la loi binomiale, écrire une fonction `rbinomiale(m,n,r,p)` qui renvoie une matrice de taille $m \times n$ dont les coefficients sont des réalisations indépendantes de la loi $\mathcal{B}(r, p)$.
3. En utilisant la formule (1), écrire une fonction `rbinomiale2(m,n,r,p)` ayant le même objectif que la fonction précédente mais n'utilisant que $m \times n$ appels à `rand`.

Exercice 3.3. Fonction génératrice des probabilités

Soit X une variable aléatoire à valeurs dans \mathbb{N} et G_X , sa fonction génératrice des probabilités, définie pour $s \in [0; 1]$ par

$$G_X(s) = \mathbb{E}[s^X] = \sum_{k=0}^{\infty} \mathbb{P}(X = k) s^k.$$

1. Calculer $G_X(1)$ et $G'_X(1)$. Pour le calcul de $G'_X(1)$, on pourra, pour intervertir les limites utiliser le théorème de convergence monotone avec la mesure de comptage.
2. Soit n un entier supérieur ou égal à 1. On suppose que $\mathbb{E}[X^n] < \infty$. Montrer que

$$\mathbb{E}[X(X-1)(X-2)\cdots(X-n+1)] = G_X^{(n)}(1),$$

où $G_X^{(n)}$ désigne la dérivée $n^{\text{ème}}$ de G_X .

3. Montrer que, si X est de carré intégrable, on a

$$\text{Var}[X] = G''_X(1) + G'_X(1) - (G'_X(1))^2.$$

Exercice 3.4. Loi géométrique

Soit $p \in]0; 1[$ et T une variable aléatoire de loi géométrique de paramètre p . Soit $q = 1 - p$.

1. En utilisant la définition de la loi géométrique, démontrer que

$$\mathbb{P}(T = k) = q^{k-1}p, \quad k = 1, 2, \dots \quad (4)$$

2. Montrer que la fonction génératrice des probabilités de T vaut

$$G_T(s) = \frac{ps}{1 - qs}, \quad s \in [0; 1]. \quad (5)$$

3. Démontrer que l'espérance et la variance de T sont données par

$$\mathbb{E}[T] = \frac{1}{p} \quad \text{et} \quad \text{Var}[T] = \frac{q}{p^2}. \quad (6)$$

Exercice 3.5. Loi géométrique avec chaîne de Markov

Soit $p \in]0; 1[$ et $(X_k)_{k \geq 0}$ la chaîne de Markov dont l'ensemble d'états est $\{0; 1\}$ telle que : $X_0 = 0$ p.s. et pour tout $k \geq 0$,

$$\begin{aligned} \mathbb{P}(X_{k+1} = 1 \mid X_k = 0) &= p \\ \mathbb{P}(X_{k+1} = 0 \mid X_k = 0) &= 1 - p \\ \mathbb{P}(X_{k+1} = 1 \mid X_k = 1) &= 1 \end{aligned}$$

Soit T le temps d'atteinte de l'état 1. On note $q = 1 - p$.

1. Montrer que T suit la loi géométrique de paramètre p .
2. En utilisant la propriété de Markov faible au temps 1, montrer que pour tout entier naturel non nul k ,

$$\mathbb{E}[T^k] = 1 + \frac{q}{p} \sum_{i=0}^{k-1} \binom{k}{i} \mathbb{E}[T^i].$$

3. Retrouver $\mathbb{E}[T]$ et $\text{Var}[T]$.

Exercice 3.6. Simulation de la loi géométrique

Soit $p \in]0; 1[$ et $q = 1 - p$.

1. En utilisant la définition, écrire une fonction `rgeom(m,n,p)` qui renvoie une matrice de taille $m \times n$ dont les coefficients sont des réalisations indépendantes de la loi géométrique de paramètre p .
2. Soit U une variable aléatoire de loi uniforme sur $]0; 1[$. Montrer que la variable aléatoire

$$\left\lfloor \frac{\ln(U)}{\ln(q)} \right\rfloor + 1,$$

où $\lfloor x \rfloor$ désigne la partie entière du réel x , suit la loi géométrique de paramètre p .

3. En utilisant ce résultat, écrire une fonction `rgeom2(m,n,p)` ayant le même rôle que la fonction précédente.

Définition 6. Soit $\lambda > 0$. On dit que la variable aléatoire N suit la loi de Poisson lorsque, pour tout $k \in \mathbb{N}$,

$$\mathbb{P}(N = k) = e^{-\lambda} \frac{\lambda^k}{k!}.$$

Exercice 3.7. Fonction génératrice des probabilités de la loi de Poisson

1. Montrer que la fonction génératrice des probabilités de N est

$$G_N(s) = e^{\lambda(s-1)}.$$

2. En déduire l'égalité

$$\mathbb{E}[N] = \text{Var}[N] = \lambda.$$

3. Soit X_n de loi $\mathcal{B}\left(n, \frac{\lambda}{n}\right)$, pour n assez grand. Montrer que X_n converge en loi vers N .

Exercice 3.8. Simulation de la loi de Poisson

La loi $\Gamma(\alpha, \lambda)$, où α et λ sont deux réels strictement positifs est la loi de densité

$$f_{\alpha, \lambda}(x) = \mathbf{1}_{]0; \infty[} \frac{x^{\alpha-1}}{\Gamma(\alpha)} \lambda^\alpha e^{-\lambda x}.$$

On rappelle que la fonction Γ est définie sur $]0; \infty[$ par

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

et que, si $n \in \mathbb{N}^*$, $\Gamma(n) = (n-1)!$. On fixe $\lambda > 0$.

1. Soient X_1, X_2, \dots des variables aléatoires indépendantes de même loi exponentielle de paramètre λ . Montrer que, pour tout $n \in \mathbb{N}^*$, la variable aléatoire $X_1 + X_2 + \dots + X_n$ suit la loi $\Gamma(n, \lambda)$.
2. Soit $(U_k)_{k \geq 1}$ une famille de variables aléatoires indépendantes de loi uniforme sur $[0; 1]$. Soit

$$N = \max \left\{ n \geq 1 \mid U_1 U_2 \cdots U_n > e^{-\lambda} \right\},$$

avec ici $\max \emptyset = 0$. Justifier que $\mathbb{P}(N \geq n) = \mathbb{P}(X_1 + X_2 + \dots + X_n \leq 1)$.

3. En déduire que N suit la loi de Poisson de paramètre λ .
4. Écrire une fonction `rpoisson(m,n, lambda)` qui simule la loi de poisson de paramètre λ .

4 Inversion de la fonction de répartition

Nous passons maintenant à une méthode générale, simple et puissante pour simuler des lois de probabilité sur \mathbb{R} . Mais d'abord, quelques rappels sur la fonction de répartition sont de mise.

4.1 Fonction de répartition

Soit X une variable aléatoire à valeurs réelles. La fonction de répartition de X (en anglais, *cumulative distribution function*) est définie sur \mathbb{R} par

$$F_X(x) = \mathbb{P}(X \leq x).$$

En Scilab, la plupart des fonctions de répartition des lois de références sont déjà programmées. Elles sont en général préfixées par `cdf`, par exemple, `cdfnor`, `cdfpoi`, ... Voir l'exemple ci-dessous (en particulier la bizarrerie consistant à entrer une chaîne de caractère « PQ » pour dire que l'on veut la fonction de répartition).

```
// Exemple d'utilisation de cdfnor
// pour la fonction de répartition de la loi normale
mu = 0 ; sigma = 1;
// Calcul de F(1,96)
cdfnor("PQ", 1.96, mu, sigma)
// résultat (bien connu) : 0.9750021
```

Exercice 4.1. Rappels sur les fonctions croissantes et la fonction de répartition

1. Soit I un intervalle réel d'intérieur non vide et $f: I \rightarrow \mathbb{R}$ une fonction croissante. Montrer qu'en tout point x de l'intérieur de I , f admet une limite à gauche $f(x-)$ et une limite à droite $f(x+)$, lesquelles vérifient

$$f(x-) \leq f(x) \leq f(x+).$$

2. En déduire que l'ensemble des points de discontinuité de f est au plus dénombrable.
3. Soit X une variable aléatoire à valeurs réelles et F sa fonction de répartition. Montrer que F est croissante, a pour limite 0 en $-\infty$, 1 en $+\infty$, admet des limites à gauche et est continue à droite en tout réel.
4. Justifier que tout ouvert de \mathbb{R} est une union dénombrable d'intervalles ouverts.
5. Soient $a < b$. Exprimer avec la fonction F la probabilité $\mathbb{P}(X \in]a; b])$.
6. Déduire des deux points précédents que la fonction F caractérise la loi de X .

4.2 Inverse de Lévy

Notons maintenant, pour $0 < u \leq 1$,

$$F_X^{-1}(u) = \inf \{x \in \mathbb{R} \mid F_X(x) \geq u\}.$$

Cette fonction s'appelle *l'inverse généralisée* ou *l'inverse de Lévy* de F_X ou encore *fonction quantile* de X .

En Scilab, on utilise encore les fonctions `cdf` avec des paramètres différents. Voir l'exemple ci-dessous (en particulier la bizarrerie consistant à entrer une chaîne de caractère « X » pour dire que l'on veut l'inverse de la fonction de répartition et l'entrée des deux nombres `p` et `1-p`).

```
// Inversement, je veux connaître x
// tel que F(x) = 0.975
mu = 0 ; sigma = 1;
p = 0.975; q = 1-p;
cdfnor("X", mu, sigma, p, q)
// résultat : 1.959964
```

Consulter l'aide et s'entraîner *avant l'épreuve* semble indispensable.

Théorème 7 (Inverse de la fonction de répartition). *La fonction F^{-1} est croissante et continue à gauche. De plus, si U est une variable aléatoire de loi uniforme sur $[0; 1]$. Alors la variable aléatoire $F_X^{-1}(U)$ a la même loi que X .*

Exercice 4.2. Démonstration du théorème

Pour alléger les notations, on écrit $F := F_X$ et $F^{-1} := F_X^{-1}$.

1. Montrer que F^{-1} est croissante.
2. Montrer que pour tout $u \in]0; 1[$,

$$F(F^{-1}(u)) \geq u. \quad (7)$$

3. Montrer que F^{-1} est continue à gauche.
4. Montrer que pour tous $x \in \mathbb{R}$ et $0 < u \leq 1$,

$$u \leq F(x) \iff F^{-1}(u) \leq x.$$

5. Calculer $\mathbb{P}(F^{-1}(U) \leq x)$ puis conclure.

4.3 Loi prenant un nombre dénombrable de valeurs

Ceci n'entre pas exactement dans le cadre précédent mais suit le même principe. Soit X une variable aléatoire à valeurs dans un ensemble (fini ou) dénombrable

$$E = \{x_1, x_2, \dots\}.$$

Soit, pour tout $k \geq 1$, $p_k = \mathbb{P}(X = x_k)$. Soit la suite $(s_k)_{k \geq 0}$ définie par $s_0 = 0$ et, pour tout $k \geq 0$,

$$s_{k+1} = s_k + p_k. \quad 1$$

Alors, si U suit la loi uniforme sur $[0; 1]$, la variable aléatoire Z définie par

$$Z = x_k \iff s_k < U \leq s_{k+1},$$

a la même loi que X . On remarquera que cette méthode fait intervenir *un nombre aléatoire de comparaisons*. On a parfois intérêt à indexer les valeurs x_k par ordre décroissant de probabilités pour diminuer le nombre de tests.

En Scilab, on pensera dans ce genre de situation à utiliser les fonctions `cumsum` (pour la somme cumulative) et `dsearch` (qui localise la classe d'une valeur, les classes étant données par un vecteur trié).

Exercice 4.3. 1. En utilisant la méthode ci-dessus, proposer une nouvelle fonction pour la simulation de la loi de Poisson.

2. Soit X une variable aléatoire à valeurs dans $\{0, 1, 2, \dots, N-1\}$. Écrire une fonction en Scilab `rfinite(m,n,P)` qui renvoie un échantillon de taille $m \times n$ suivant la loi de X , où la matrice P contient les valeurs $p_k = \mathbb{P}(X = k)$, pour $k = 0, 1, \dots, N-1$.

1. Bref, les s_k sont les sommes cumulatives des p_k .

4.4 Application à la loi de Cauchy et la loi exponentielle

Exercice 4.4. 1. Calculer l'inverse de la fonction de répartition de la fonction exponentielle de paramètre $\lambda > 0$, de densité

$$\lambda \exp(-\lambda x).$$

2. En déduire une fonction `rexp(m,n,lambda)` pour simuler la loi exponentielle.
3. Faire de même pour la loi de Cauchy centrée réduite, de densité

$$f(x) = \frac{1}{\pi(1+x^2)}.$$

4. La loi de Cauchy admet-elle une espérance ?

5 Méthode du rejet

Ce chapitre s'inspire, plus encore que les autres, du livre [BC07] (partie 1.10).

5.1 Loi uniforme

Proposition 8 (Méthode du rejet). *Soit μ une mesure sur un espace mesurable (E, \mathcal{A}) et $B \subset A$ deux mesurables tels que*

$$0 < \mu(B) \leq \mu(A) < \infty.$$

Soit $p = \frac{\mu(B)}{\mu(A)}$. Soit $(U_k)_{k \geq 1}$ une suite de variables aléatoires i.i.d. de loi (μ) -uniforme sur A et $V_k = \mathbf{1}_{\{U_k \in B\}}$. Alors les variables aléatoires V_k sont i.i.d. de loi de Bernoulli de paramètre p . Si de plus, $T = \inf \{k \geq 1 \mid Y_k = 1\}$, alors T suit la loi géométrique de paramètre p et est indépendante de X_T , laquelle suit la loi uniforme sur B .

Démonstration. Par définition de la loi uniforme sur A , on a pour tout $k \geq 1$,

$$\mathbb{P}(X_k \in B) = \frac{\mu(B)}{\mu(A)},$$

donc Y_k suit la loi de Bernoulli de paramètre p et T suit la loi géométrique de paramètre p . Pour vérifier la deuxième partie du théorème, on calcule, étant donné une partie mesurable C incluse dans B , et un entier n , en utilisant l'indépendance des X_k :

$$\begin{aligned} \mathbb{P}(T = n, X_T \in C) &= \mathbb{P}(X_1 \notin C, \dots, X_{n-1} \notin C, X_n \in C) \\ &= \mathbb{P}(X_1 \notin C) \cdots \mathbb{P}(X_{n-1} \notin C) \mathbb{P}(X_n \in C) \\ &= (1-p)^{n-1} \frac{\mu(C)}{\mu(A)} = (1-p)^{n-1} \frac{\mu(C)}{\mu(B)} \frac{\mu(B)}{\mu(A)} \\ &= (1-p)^{n-1} p \frac{\mu(C)}{\mu(B)} = \mathbb{P}(T = n) \mathbb{P}(Z \in C), \end{aligned}$$

où Z suit la loi uniforme sur B . □

L'exercice suivant, très connu (bien que peu efficace), permet de calculer π en utilisant la méthode de Monte-Carlo.

Exercice 5.1. Calcul de π par la méthode de Monte-Carlo

1. En utilisant la loi uniforme sur le carré $C = [0; 1]^2$, simuler la loi uniforme sur le quart de disque

$$\mathcal{Q} = \left\{ (x, y) \in C \mid x^2 + y^2 \leq 1 \right\}.$$

Combien faut-il, en moyenne, simuler de variables uniformes C pour obtenir une uniforme sur \mathcal{Q} ?

2. Soient U_1, U_2, \dots i.i.d. de lois uniformes sur C . Justifier que

$$\frac{1}{n} \text{Card} \{1 \leq k \leq n \mid U_k \in \mathcal{Q}\} \xrightarrow[n \rightarrow \infty]{} \frac{\pi}{4}.$$

3. Écrire un programme utilisant cette méthode pour calculer π .

4. En utilisant l'approximation donnée par le théorème central limite, estimer le nombre minimal de lancers à faire pour que la valeur donnée de $\frac{\pi}{4}$ soit juste à 10^{-2} près, avec une probabilité supérieure à 95%.

5.2 Application aux lois à densités

Pour une approche plus intuitive du théorème suivant, se référer à [BC07], partie 1.11.

Théorème 9 (Méthode du rejet comparatif). Soient μ et ν deux lois à densité sur \mathbb{R} , de fonctions de densité respectives f et g . On suppose qu'il existe une constante $c > 0$ telle $f \leq cg$. Soient X_1, X_2, \dots des variables i.i.d. de loi μ et soient U_1, U_2, \dots des variables i.i.d. de loi uniforme sur $[0; 1]$, indépendantes des X_k . Soit

$$T := \inf \left\{ k \geq 1 \mid U_k < \frac{f(X_k)}{cg(X_k)} \right\}. \quad (8)$$

Alors, T suit la loi géométrique de paramètre $\frac{1}{c}$ et est indépendante de X_T , laquelle a pour loi ν .

Démonstration. On commence par remarquer que l'expression 8 a un sens avec probabilité 1, car

$$\mathbb{P}(g(X_k) = 0) = \int \mathbf{1}_{\{g(x)=0\}} g(x) dx = 0.$$

Les événements $U_k < \frac{f(X_k)}{cg(X_k)}$ sont clairement indépendants de même probabilité égale à

$$\mathbb{E} \left[\mathbb{P} \left[U_k < \frac{f(X_k)}{cg(X_k)} \mid X_k \right] \right] = \mathbb{E} \left[\frac{f(X_k)}{cg(X_k)} \right] = \int \frac{f(x)}{cg(x)} g(x) dx = \frac{1}{c} \int f(x) dx = \frac{1}{c}.$$

Cela montre que T suit la loi géométrique de paramètre $\frac{1}{c}$. Ensuite, soit A un borélien et un entier $n \geq 1$. On a (en considérant X et U indépendantes de loi respectives μ et $\mathcal{U}([0; 1])$ pour alléger les notations) :

$$\begin{aligned} \mathbb{P}(T = n, X_T \in A) &= \mathbb{P}(T = n, X_n \in A) = \mathbb{P}(T = n) \mathbb{P}[X_n \in A \mid T = n] \\ &= \mathbb{P}(T = n) \mathbb{P} \left[X \in A \mid U \leq \frac{f(X)}{cg(X)} \right] \\ &= \mathbb{P}(T = n) \frac{\mathbb{P} \left(X \in A, U \leq \frac{f(X)}{cg(X)} \right)}{\mathbb{P} \left(U \leq \frac{f(X)}{cg(X)} \right)} \\ &= \mathbb{P}(T = n) \times c \int_{u=0}^{\frac{f(x)}{cg(x)}} \int_{x \in A} g(x) dx du. \end{aligned}$$

où, pour passer de la première à la deuxième ligne, on a décomposé l'événement $T = n$ en

$$\{T = n\} = \left\{ U_1 \geq \frac{f(X_1)}{g(X_1)}, U_2 \geq \frac{f(X_2)}{g(X_2)}, \dots, U_{n-1} \geq \frac{f(X_{n-1})}{g(X_{n-1})}, U_n < \frac{f(X_n)}{g(X_n)} \right\},$$

et l'on a remarqué que tous ces événements à l'exception du dernier étaient indépendants de l'événement $X_n \in A$. En utilisant le théorème de Fubini et en intégrant d'abord suivant u la dernière intégrale devient

$$\int_{x \in A} \frac{f(x)}{cg(x)} g(x) dx = \int_{x \in A} \frac{1}{c} f(x) dx = \frac{1}{c} \nu(A),$$

ce qui achève cette démonstration. □

L'exercice suivant montre une application (un peu trop facile et maladroite) de cette méthode.

Exercice 5.2. On considère la fonction f définie par :

$$f(x) = \begin{cases} 3x^2 & \text{si } x \in [0; 1], \\ 0 & \text{sinon.} \end{cases}$$

1. Justifier que f est une fonction de densité de probabilité, calculer sa fonction de répartition et l'inverse de sa fonction de répartition. En déduire un programme `inv_rep` permettant de simuler la loi de densité f .
2. Expliquer comment, en utilisant la méthode du rejet comparatif, on peut simuler la loi de densité f puis mettre en œuvre cette méthode dans une fonction `rejet`.
3. Pour un grand nombre de valeurs simulées, combien d'appels à la fonction `rand` votre programme `rejet` fait-il en moyenne ?

6 Loi normale, vecteurs gaussiens

Sur ce sujet, on pourra consulter par exemple les premières pages de [LG12], le chapitre 6 de [BC07], le chapitre 7 de [Can13] et bien d'autres références encore.

6.1 Variables gaussiennes

Proposition 10. Soit $f: \mathbb{R} \rightarrow \mathbb{R}$ définie par

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

Alors f est une densité de probabilité.

Exercice 6.1. Soit $I = \int e^{-\frac{x^2}{2}} dx$. Justifier que

$$I^2 = \iint e^{-\frac{x^2+y^2}{2}} dx dy.$$

Faire un changement de variable en coordonnées polaires, puis calculer I .

Définition 11. La loi normale centrée réduite, notée $\mathcal{N}(0, 1)$ sur \mathbb{R} est la loi de densité f . Si $\mu \in \mathbb{R}$ et $\sigma \geq 0$, on définit la loi normale $\mathcal{N}(\mu, \sigma^2)$ comme la loi de la variable aléatoire $X = \mu + \sigma N$, où N suit la loi $\mathcal{N}(0, 1)$. Lorsqu'une variable aléatoire X suit une loi normale, on dit que cette variable aléatoire est gaussienne.

Proposition 12. Soit $N \sim \mathcal{N}(0, 1)$, $m \in \mathbb{R}$, $\sigma \geq 0$ et $X \sim \mathcal{N}(m, \sigma^2)$.

1. L'espérance et la variance de N valent respectivement 0 et 1.
2. L'espérance et la variance de X valent respectivement m et σ^2 .
3. Si $\sigma = 0$, alors X a pour loi la mesure de Dirac en m , sinon X a pour densité la fonction

$$x \mapsto \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}}.$$

4. Les fonctions caractéristiques de N et de X valent respectivement

$$\varphi_N(t) := \mathbb{E} \left[e^{itN} \right] = e^{-\frac{t^2}{2}}, \quad (9)$$

$$\varphi_X(t) := \mathbb{E} \left[e^{itX} \right] = e^{imt} e^{-\frac{\sigma^2 t^2}{2}}. \quad (10)$$

5. Les fonctions génératrices des moments de N et de X sont finies sur \mathbb{R} et valent respectivement

$$M_N(t) := \mathbb{E} \left[e^{tN} \right] = e^{\frac{t^2}{2}} \quad (11)$$

$$M_X(t) = e^{tm} e^{\frac{\sigma^2 t^2}{2}}. \quad (12)$$

6. La loi normale centrée réduite admet des moments de tous ordre. Ses moments d'ordre impair sont tous nuls, tandis que ses moments d'ordre pair sont donnés par

$$\mathbb{E} \left[N^{2n} \right] = \frac{(2n)!}{2^n n!}.$$

Exercice 6.2. Démonstration des résultats.

On indique pour chacun des points une stratégie de démonstration.

1. Pour l'espérance, utiliser un argument de symétrie. Pour la variance, faire une intégration par parties.
2. Surtout, ne pas refaire le même calcul, utiliser la définition.
3. Considérer une fonction f mesurable positive, écrire

$$\mathbb{E} [f(X)] = \mathbb{E} [f(m + \sigma N)],$$

et faire un changement de variable dans l'intégrale.

4. Commencer par remarquer que φ_N est à valeurs réelles pour des raisons de symétries et donc que

$$\varphi_N(t) = \int \cos(tx) e^{-\frac{x^2}{2}} dx.$$

Dériver sous le signe intégrale puis intégrer par parties pour montrer que φ_N vérifie l'équation différentielle $y' = -ty$ puis conclure.

5. Faire un calcul direct (avec un changement de variables sous la forme d'une translation).
6. Utiliser, par exemple, un développement limité de la fonction caractéristique au voisinage de 0.

Corollaire 13. Toute combinaison linéaire de variables gaussiennes indépendantes est encore gaussienne.

Exercice 6.3. Démontrer ce corollaire en utilisant la fonction caractéristique.

Le rôle central de la loi normale, tant en probabilités qu'en statistiques, vient de son caractère universel donné par le théorème central limite. La démonstration utilise les fonctions caractéristiques. Elle n'est pas très difficile et est bien détaillée dans [Tou99].

Théorème 14 (théorème central limite). Soit μ une mesure de probabilité sur \mathbb{R} , de variance finie σ^2 et de moyenne m . Soient X_1, X_2, \dots des variables aléatoires i.i.d. de loi μ et pour tout $n \in \mathbb{N}^*$ $S_n = X_1 + \dots + X_n$. Alors on a la convergence en loi

$$\sqrt{n} \left(\frac{S_n}{n} - m \right) \xrightarrow[n \rightarrow \infty]{(d)} \mathcal{N} \left(0, \sigma^2 \right).$$

6.2 Méthode de Box-Muller

Proposition 15 (Box, Muller). *Soient Y et θ deux variables aléatoires indépendantes, avec Y de loi exponentielle de paramètre $\frac{1}{2}$ et θ uniforme sur $[0; 2\pi]$. Soit*

$$(X, Y) = \left(\sqrt{Y} \cos \theta, \sqrt{Y} \sin \theta \right).$$

Alors, les variables aléatoires X et Y sont indépendantes, de même loi $\mathcal{N}(0, 1)$.

Exercice 6.4.

1. Démontrer la proposition précédente. On pourra considérer une fonction f mesurable positive, et calculer

$$\mathbb{E} \left[f \left(\sqrt{Y} \cos \theta, \sqrt{Y} \sin \theta \right) \right]$$

en faisant un premier changement de variable $y = r^2$, puis un passage en coordonnées polaires.

2. En déduire une fonction Scilab `rnorm(m, n)` pour simuler la loi normale centrée réduite.

Remarque 16. La méthode de Box-Muller est assez classique et utilisée en pratique, mais on lui préfère souvent la méthode de la ziggurat, inventée par Marsaglia (un grand nom de la simulation) et ses collaborateurs. Elle est basée sur une méthode de rejet très astucieuse. Voir l'article original [MT⁺00].

6.3 Vecteurs aléatoires

On considère toujours un espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$. Un vecteur aléatoire $X = (X_1, \dots, X_n)^T$ est une fonction mesurable de $(\Omega, \mathcal{F}, \mathbb{P})$ vers un espace $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$. Ses composantes sont des variables aléatoires et réciproquement, la donnée de n variables aléatoires définit un vecteur aléatoire dans \mathbb{R}^n .

On dit que le vecteur aléatoire X est carré-intégrable, lorsque chacune de ses composantes l'est, c'est-à-dire lorsque

$$\forall i \in \llbracket 1; n \rrbracket, \quad \mathbb{E} \left[X_i^2 \right] < \infty.$$

Étant donné un vecteur aléatoire X carré-intégrable, on définit l'espérance de X comme le vecteur (déterministe) de \mathbb{R}^n ,

$$\mathbb{E} [X] = (\mathbb{E} [X_1], \mathbb{E} [X_2], \dots, \mathbb{E} [X_n])^T,$$

et la matrice de covariance de X par

$$\text{Cov} (X) = (\text{Cov} (X_i, X_j))_{1 \leq i, j \leq n},$$

où la covariance des deux variables aléatoires X_i et X_j est définie par

$$\text{Cov} (X_i, X_j) = \mathbb{E} [(X_i - \mathbb{E} [X_i]) (X_j - \mathbb{E} [X_j])] = \mathbb{E} [X_i X_j] - \mathbb{E} [X_i] \mathbb{E} [X_j].$$

La covariance est ici bien définie car, d'après l'inégalité de Cauchy-Schwarz²,

$$|\mathbb{E} [X_i X_j]|^2 \leq \mathbb{E} [X_i^2] \mathbb{E} [X_j^2] < \infty.$$

Bien sûr, $\text{Cov} (X_i, X_i) = \text{Var} [X_i]$. Noter que la covariance est, comme la variance, invariante par translation.

Voyons maintenant l'effet du produit par une matrice A .

2. Hermann Schwarz (1843 - 1921), mathématicien allemand, auteur également du théorème de Schwarz sur les dérivées partielles, à ne pas confondre avec Laurent Schwartz (1915 - 2002), mathématicien français célèbre pour ses travaux sur la théorie des distributions.

Proposition 17. Soit X un vecteur aléatoire carré-intégrable de \mathbb{R}^n et A une matrice réelle de taille $m \times n$. Alors AX est un vecteur aléatoire de \mathbb{R}^m d'espérance et de covariance

$$\mathbb{E}[AX] = A\mathbb{E}[X] \quad \text{et} \quad \text{Cov}(AX) = A \text{Cov}(X) A^T.$$

Démonstration. La première égalité est très facile en utilisant la linéarité de l'espérance. Pour la seconde, on peut, quitte à lui soustraire son espérance, supposer que $\mathbb{E}[X]$ est le vecteur nul, sans changer ni $\text{Cov}(X)$, ni $\text{Cov}(AX)$ (par invariance par translation). Notons

$$A = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n} \quad \text{et} \quad D = (d_{ij})_{1 \leq i, j \leq m} = \text{Cov}(AX).$$

On a alors, pour tous $i, j \in \llbracket 1; m \rrbracket$,

$$d_{ij} = \mathbb{E} \left[\left(\sum_{r=1}^n a_{ir} X_r \right) \left(\sum_{s=1}^n a_{js} X_s \right) \right] = \sum_{1 \leq r, s \leq n} a_{ir} \mathbb{E}[X_r X_s] a_{js},$$

qui est précisément le coefficient d'indice (i, j) dans la matrice

$$A \text{Cov}(X) A^T. \quad \square$$

Proposition 18. Une matrice de covariance est toujours symétrique et (semi-définie) positive.

Démonstration. La symétrie est évidente. Soit u un vecteur colonne de \mathbb{R}^n . D'après la proposition précédente, on a

$$u^T \text{Cov}(X) u = \text{Cov}(u^T X) = \text{Var}(u^T X) \geq 0. \quad \square$$

Comme dans le cas des variables aléatoires (réelles), les vecteurs aléatoires (de \mathbb{R}^n) ont une loi caractérisée par leurs fonctions caractéristiques (leurs transformées de Fourier), définies pour tout $t = (t_1, t_2, \dots, t_n) \in \mathbb{R}^n$ par

$$\varphi_X(t) = \mathbb{E} \left[e^{i\langle t, X \rangle} \right],$$

où $\langle \cdot, \cdot \rangle$ désigne le produit scalaire canonique sur \mathbb{R}^n .

Avec les mêmes notations que précédemment, on obtient immédiatement,

$$\varphi_{AX}(t) = \varphi_X(A^T t). \quad (13)$$

6.4 Vecteurs gaussiens

Définition 19. Soit n un entier naturel non nul. Un vecteur gaussien de \mathbb{R}^n est un vecteur aléatoire X de \mathbb{R}^n , tel que toute combinaison linéaire de ses composantes est une variable gaussienne.

Commençons par donner quelques conséquences immédiates de cette définition.

Proposition 20. Soit n un entier naturel non nul et X un vecteur aléatoire de \mathbb{R}^n . Les propositions suivantes sont équivalentes.

1. Le vecteur X est un vecteur gaussien.
2. Pour tout vecteur (déterministe) u de \mathbb{R}^n , le produit scalaire $\langle u, X \rangle$ est une variable gaussienne.
3. Pour tout entier naturel non nul m , toute matrice réelle A de taille $m \times n$ et tout vecteur déterministe b de \mathbb{R}^m , le vecteur aléatoire de \mathbb{R}^m $AX + b$ est un vecteur gaussien.

Exemple 21. Si X_1, X_2, \dots, X_n sont n variables gaussiennes indépendantes, alors $X = (X_1, \dots, X_n)$ est un vecteur gaussien d'après le corollaire 13. D'après la proposition précédente, et en utilisant les mêmes notations, il en est donc de même pour tous les vecteurs aléatoire de la forme $AX + b$.

Nous allons voir qu'en fait, tous les vecteurs gaussiens peuvent s'écrire de cette façon.

Remarque 22. Si X est un vecteur gaussien, alors ses composantes sont des variables gaussiennes. **La réciproque est fautive !** La remarque 5 fournit un contre-exemple.

Le calcul de la fonction caractéristique d'un vecteur gaussien est immédiat, mais d'une importance cruciale.

Proposition 23 (fonction caractéristique d'un vecteur gaussien). *Soit X un vecteur aléatoire de \mathbb{R}^n . Alors, X est un vecteur gaussien si et seulement si, il existe un vecteur déterministe μ de \mathbb{R}^n et une matrice carrée de taille n , C symétrique et positive tels que que pour tout vecteur v de \mathbb{R}^n ,*

$$\varphi_X(v) := \mathbb{E} \left[e^{i\langle v, X \rangle} \right] = e^{i\langle v, \mu \rangle} e^{-\frac{1}{2}\langle v, Cv \rangle}.$$

Dans ce cas, μ est l'espérance de X et C est sa matrice de covariance.

Démonstration. Pour l'implication directe, on pose $\mu = \mathbb{E}[X]$ et $C = \text{Cov}(X)$. La variable aléatoire $\langle v, X \rangle$ est gaussienne, d'espérance $\langle v, \mu \rangle$ et de variance $\langle v, Cv \rangle$, d'après la proposition 17 (le réel $\langle v, Cv \rangle$ est bien positif car C est symétrique positive). D'après la formule 10, sa fonction caractéristique est

$$\varphi_{\langle v, X \rangle}(t) = e^{it\langle v, X \rangle} e^{-\frac{1}{2}t^2\langle v, Cv \rangle}.$$

On remarque pour conclure que

$$\varphi_X(v) = \varphi_{\langle v, X \rangle}(1).$$

Pour l'implication réciproque, on commence par considérer un vecteur v de \mathbb{R}^n . Comme

$$\varphi_{\langle v, X \rangle}(t) = \varphi_X(tv) = e^{it\langle v, \mu \rangle} e^{-\frac{1}{2}t^2\langle v, Cv \rangle},$$

la variable $\langle v, X \rangle$ est gaussienne. Ceci étant valable pour tout vecteur v , X est bien un vecteur gaussien.

Pour calculer son espérance, il suffit d'appliquer la formule précédents aux éléments e_1, \dots, e_n de la base canonique de \mathbb{R}^n . Enfin pour trouver sa matrice de covariance, on peut considérer les vecteurs $e_i + e_j$ et utiliser la symétrie de la matrice C . \square

Corollaire 24. 1. *La loi d'un vecteur gaussien est entièrement caractérisée par son espérance et sa matrice de covariance.*

2. *Si $X = (X_1, \dots, X_n)$ est un vecteur gaussien, alors les variables gaussiennes X_1, \dots, X_n sont indépendantes si et seulement si la matrice de covariance de X est diagonale, c'est-à-dire si et seulement si les variables (X_1, \dots, X_n) sont deux-à-deux non corellées.*

Proposition 25. *Soit C une matrice symétrique positive de taille n et μ un vecteur déterministe de \mathbb{R}^n . Soit N un vecteur gaussien de \mathbb{R}^n dont toutes les composantes sont indépendantes de loi normale centrée réduite.*

Alors il existe des matrices A telles que $AA^T = C$ et pour toute telle matrice, le vecteur aléatoire $AN + \mu$ est un vecteur gaussien de matrice de covariance C et d'espérance μ .

Remarque 26. En vertu des deux résultats précédent, on peut parler, pour tout vecteur μ et toute matrice symétrique positive C de « la loi normale de matrice de covariance C et d'espérance μ ». On notera cette dernière $\mathcal{N}(\mu, C)$.

Démonstration. La seule chose à montrer est l'existence de telles matrices A . Une première possibilité est d'utiliser le fait que C , symétrique, est diagonalisable par le groupe orthogonale, donc il existe Q orthogonale et Δ diagonale telles que

$$C = Q^{-1}\Delta Q = Q^T\Delta Q.$$

Comme C est positive par hypothèse, tous les éléments de Δ sont positifs, de sorte que la matrice formée par leurs racines carrées, notée ici $\sqrt{\Delta}$ est bien définie. On peut alors poser

$$A = Q^T\sqrt{\Delta}. \quad \square$$

Remarque 27. Nous verrons plus loin avec la décomposition de Cholesky qu'il n'est nullement besoin de diagonaliser C pour trouver une telle matrice A .

Nous justifions ensuite la remarque qui suit l'exemple 21.

Proposition 28. *Soit X un vecteur gaussien de \mathbb{R}^n , de moyenne μ et de matrice de covariance C . Soit $r \leq n$ le rang de la matrice C . Alors il existe r variables gaussiennes indépendantes N_1, N_2, \dots, N_r indépendantes centrées et réduites et une matrice A de taille $n \times r$ telles que*

$$X = A(N_1, N_2, \dots, N_r)^T + \mu.$$

Démonstration. Quitte à raisonner sur $X - \mu$, on peut supposer que μ est le vecteur nul. Soient

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0 = \lambda_{r+1} = \dots = \lambda_n$$

les valeurs propres de C comptées avec leurs multiplicités. Soient

$$\Delta = \begin{pmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & \lambda_r & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix} \text{ et } D = \begin{pmatrix} \lambda_1^{-\frac{1}{2}} & & & & \\ & \ddots & & & \\ & & \lambda_r^{-\frac{1}{2}} & & \\ & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix}.$$

Soit Q une matrice orthogonale telle que $C = Q^T \Delta Q$. Le vecteur gaussien

$$N := DQX$$

a pour matrice de covariance

$$\text{Cov}(N) = DQCC^T D = DQQ^T \Delta QQ^T D = D\Delta D = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix}.$$

Par conséquent ses r premières composantes N_1, \dots, N_r sont bien gaussiennes centrées réduites et indépendantes, d'après le corollaire 24, tandis que ses $n - r$ dernières composantes sont presque sûrement égales à 0.

Pour finir, on note que

$$X = Q^{-1} D^{-1} N = Q^{-1} \begin{pmatrix} \sqrt{\lambda_1} & & & \\ & \ddots & & \\ & & \sqrt{\lambda_r} & \\ & & & 0 \\ & & & \vdots \\ & & & & 0 \end{pmatrix} \begin{pmatrix} N_1 \\ \vdots \\ N_r \end{pmatrix}. \quad \square$$

Remarque 29. En Scilab, la fonction `spec` permet de diagonaliser une matrice. Ce n'est pas totalement explicite dans la documentation, mais dans le cas d'une matrice symétrique, la base de vecteurs propres renvoyée est bien orthogonale³.

3. La sous-routine fortran utilisée dans ce cas est `DSYEV`, dont la documentation est claire.

6.5 Décomposition de Cholesky

Comme promis plus haut, la décomposition de Cholesky permet d'obtenir une matrice C , symétrique et positive comme un produit AA^T avec la propriété très pratique que A est triangulaire. Étonnamment, il est plus simple de démontrer d'abord l'existence d'une telle décomposition avant d'en donner l'algorithme.

Lemme 30 (Complément de Schur). *Soit C une matrice symétrique définie positive d'ordre $n = p + q$ dont une écriture par blocs est*

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

Avec C_{11} carrée d'ordre p et C_{22} carrée d'ordre q . Alors C_{11} est inversible et le complément de Schur

$$S = C_{22} - C_{21}C_{11}^{-1}C_{12}$$

est une matrice carrée d'ordre p , symétrique définie positive.

Exercice 6.5. Démonstration du lemme 30

1. Soit (e_1, \dots, e_n) la base canonique de \mathbb{R}^n et $E = \text{Vect}(e_1, \dots, e_p)$. Soit q la forme quadratique associée à la matrice C . Quelle est la matrice associée à la restriction de q à $E \times E$? En déduire que C_{11} est inversible.
2. Soit x un vecteur non nul du sous-espace E . Soit y le vecteur de \mathbb{R}^n dont une écriture par blocs est

$$y = \begin{pmatrix} -C_{11}^{-1}C_{12}x \\ x \end{pmatrix}$$

Calculer $q(y)$. Conclure.

Théorème 31 (Décomposition de Cholesky). *Soit C une matrice symétrique définie positive d'ordre n . Alors il existe une unique matrice triangulaire inférieure R telle que*

$$C = RR^T.$$

Démonstration. On commence par démontrer l'existence, par récurrence sur la dimension n . Le résultat est évident si $n = 1$. Supposons que $n \geq 2$ et que le théorème est vrai pour les matrices d'ordre $n - 1$. Notons

$$C = \begin{pmatrix} c_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix},$$

où c_{11} est un scalaire (strictement positif), C_{12} est une ligne de taille $n - 1$, $C_{21} = C_{12}^T$ et C_{22} est carrée d'ordre $n - 1$. Soit $S = C_{22} - C_{21}c_{11}^{-1}C_{12}$ le complément de Schur associé à cette écriture par blocs. D'après le lemme précédent, la matrice S est encore définie positive. Par hypothèse de récurrence, il existe une matrice triangulaire inférieure R_S telle que $S = R_S R_S^T$.

Il suffit de maintenant de poser (toujours en écriture par blocs)

$$R = \begin{pmatrix} \sqrt{c_{11}} & 0 \\ \frac{1}{\sqrt{c_{11}}}C_{21} & R_S \end{pmatrix}$$

pour avoir $C = RR^T$.

Pour l'unicité, on écrit $R = (r_{ij})_{1 \leq i, j \leq n}$ et l'on déduit de $RR^T = C$ les égalités, pour $1 \leq i \leq j \leq n$:

$$c_{ij} = \sum_{k=1}^i r_{ik}r_{jk} \quad (14)$$

$$c_{ii} = \sum_{k=1}^i r_{ik}^2. \quad (15)$$

On calcule donc d'abord le terme $r_{11} = \sqrt{c_{11}}$, puis ceux de la première colonne $r_{j1} = \frac{c_{1j}}{r_{11}}$. Puis, si les $i - 1$ premières colonnes ont été calculées, on calcule la colonne d'indice i par :

$$r_{ii} = \sqrt{c_{ii} - \sum_{k=1}^{i-1} r_{ik}^2}$$

$$r_{ji} = \frac{c_{ij} - \left(\sum_{k=1}^{i-1} r_{ik}r_{jk}\right)}{r_{ii}}. \quad \square$$

Exercice 6.6.

1. Vérifier que $C = RR^T$ dans la démonstration précédente.
2. Mettre en œuvre la décomposition de Cholesky dans une fonction Scilab.
3. Tester votre fonction.
4. Ne plus jamais utiliser votre fonction, utiliser la fonction prédéfinie `chol`, bien plus rapide (mais lire la documentation de cette fonction).

6.6 Simulation des vecteurs gaussiens

Tout a déjà été dit dans les sous-parties précédentes. Si l'on veut :

- simuler la loi $\mathcal{N}(\mu, C)$, où C est symétrique définie positive d'ordre n : on utilise la décomposition de Cholesky pour obtenir une matrice A telle que $C = AA^T$, puis on utilise un tirage N de n variables gaussiennes centrées réduites. En Scilab, on peut donc utiliser les commandes `rand(n,1,"normal")` ou bien `grand(n,1,"nor",0,1)` ;
- diagonaliser la matrice C dans le groupe orthogonal : on utilisera la fonction `spec` (avec deux arguments de retour).

Exercice 6.7. Soit $C = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$ et $\mu = (-2 \quad -3)$.

1. En utilisant la décomposition de Cholesky, écrire un script permettant d'obtenir un échantillon X , de 1000 points de \mathbb{R}^2 de loi $\mathcal{N}(\mu, C)$.
2. Compléter votre script de façon à tracer ces 1000 points dans un repère, en utilisant, par exemple la commande `plot(X(1,:), X(2,:), "r.")`.
3. Diagonaliser la matrice C , puis tracer (par exemple avec `xarrows`) les vecteurs $u_i = \lambda_i e_i$, pour $i = 1, 2$, où les λ_i sont les valeurs propres de la matrice C et e_i des vecteurs propres (de norme 1) associés.

7 Quelques exemples de processus

7.1 Chaîne de Markov à temps discret et espace d'états fini

On considère un ensemble dénombrable d'états E , muni de la tribu discrète (c'est-à-dire l'ensemble de ses parties 2^E) et une matrice stochastique P sur E , c'est-à-dire une application

$$P : E \times E \longrightarrow [0; 1]$$

telle que

$$\forall x \in E, \quad \sum_{y \in E} P(x, y) = 1.$$

On dit que P est la *matrice de transitions* de la chaîne de Markov. Soit μ une mesure de probabilité sur E appelée loi initiale de la chaîne.

On définit une variable aléatoire X à valeurs dans l'ensemble des suites $E^{\mathbb{N}}$ muni de sa tribu produit⁴ par :

1. la loi de X_0 est μ ;
2. pour tout $n \geq 0$ et tous $x, y \in E$,

$$\mathbb{P}(X_{n+1} = y \mid X_n = x) = P(x, y).$$

La mesure de probabilité ainsi définie sur l'espace des suites $E^{\mathbb{N}}$ est notée \mathbb{P}_μ .

En conditionnant par rapport aux différentes valeurs de X_0 , on voit que la loi de X_1 sous \mathbb{P}_μ est donnée par le produit μP :

$$\forall y \in E, \quad \mathbb{P}_\mu(X_1 = y) = \mu P(y) = \sum_{x \in E} \mu(x) P(x, y).$$

On définit les puissances de la matrice stochastique P de la façon usuelle :

$$\forall x, y \in E, \quad P^{k+1}(x, y) = \sum_{z \in E} P^k(x, z) P(z, y).$$

Le même raisonnement que ci-dessus montre alors que la loi de X_n sous \mathbb{P}_μ est donnée par le produit μP^n .

Supposons maintenant que l'espace d'états E est fini. À l'aide des techniques vues précédemment, on voit qu'à n fixé, il n'est pas difficile de simuler une trajectoire X_0, X_1, \dots, X_n de la chaîne de Markov. On peut par exemple :

1. utiliser la méthode de l'inverse de la fonction de répartition pour simuler la loi μ pour obtenir ainsi un certain x_0 dans E ;
2. puis simuler de la même manière la loi de probabilité $y \mapsto P(x_0, y)$ pour obtenir un certain x_1 ;
3. et recommencer l'étape précédente, en remplaçant x_0 par x_1 , etc.

On peut bien sûr trouver de nombreuses améliorations à l'algorithme précédent (calculer toutes les sommes cumulatives des lignes de P dès le départ, etc). Tout ceci est déjà programmé dans Scilab, on utilise `grand(n, "markov", P, x0)` pour le cas où X_0 est presque sûrement constante égale à x_0 . Si l'on donne un vecteur $X_0 = (x_0^1, \dots, x_0^N)$ comme dernier argument, alors `grand(N, "markov", P, X_0)` renvoie N trajectoires simulées, d'états initiaux respectifs x_0^1, x_0^2 , etc.

Signalons aussi la fonction `genmarkov` qui permet d'obtenir des matrices de transitions ayant certaines propriétés.

Exercice 7.1. Le chat bagarreur

Pito le chat n'a que trois activités : dormir, manger et se battre avec les autres chats. À chaque heure, il peut décider, ou non de changer d'activité. On note X_n l'activité du chat après n heures.

1. Après avoir dormi une heure, il continue à dormir avec probabilité 0,7 ou bien va manger avec probabilité 0,2 et se bat avec probabilité 0,1.
2. Après avoir mangé, il va se battre avec probabilité 0,3 ou bien dort avec probabilité 0,7.
3. Après s'être battu, il va manger avec probabilité 0,5 ou dormir avec probabilité 0,5.

On suppose qu'au début de l'observation, Pito dort.

1. Tracer le graphe, puis écrire la matrice de transition de cette chaîne de Markov.
2. Calculer numériquement avec Scilab la loi de l'activité de Pito après 5 heures.
3. Simuler avec Scilab 1000 observations d'une durée de 15 heures chacune. Donner les fréquences empiriques associées à X_{15} .
4. Justifier que la chaîne de Markov a une unique probabilité invariante, la calculer et justifier que la loi de X_n converge vers cette probabilité invariante.

4. c'est-à-dire la plus petite tribu qui rend les applications coordonnées $\varphi_k : (x_n)_{n \geq 0} \mapsto x_k$ mesurables

7.2 Marche aléatoire

7.2.1 Marche aléatoire dans \mathbb{R}

Soit η une mesure de probabilité sur \mathbb{R} et Z_1, Z_2, \dots des variables aléatoires i.i.d. de loi η . On définit la marche aléatoire de position initiale $x_0 \in \mathbb{R}$ et de loi d'incrément η par $X_0 = x_0$ et, pour tout $k \in \mathbb{N}$,

$$X_{k+1} = X_k + Z_{k+1}.$$

Si l'on sait simuler la loi η , la simulation de cette marche aléatoire se fait de façon immédiate à l'aide de la fonction `cumsum`.

Exercice 7.2. Soit η définie par

$$\eta(\{-1\}) = \eta(\{1\}) = \frac{1}{2},$$

et $(X_k)_{k \geq 0}$ la marche aléatoire de loi d'incrément η et de position initiale 0.

1. Simuler une trajectoire de X entre les temps 0 et 100.
2. On étend X à tous les temps $t \in [0; +\infty[$ par interpolation linéaire. Tracer une trajectoire de X .
3. Soit $n \geq 1$ et $t \in [0; +\infty[$. On définit le changement d'échelle Y_t^n de la marche par

$$Y_t^n = \frac{X_{nt}}{\sigma\sqrt{n}},$$

où σ est l'écart-type de la loi η . Tracer une trajectoire de Y_t^n , pour $n = 1000$ et $t \in [0; 1]$.

4. À t fixé, est-ce que Y_t^n converge en loi lorsque n tend vers l'infini? Si oui, vers quelle loi?

7.2.2 Marche aléatoire dans \mathbb{R}^2

Exercice 7.3. Soit $(X_k)_{k \in \mathbb{N}}$ une suite de points aléatoires du plan \mathbb{R}^2 telle que $X_0 = (0, 0)$ et pour tout entier k ,

$$X_{k+1} - X_k \in \{(1; 0), (0; 1), (-1; 0), (0; -1)\},$$

et ces 4 possibilités sont équiprobables.

Tracer avec Scilab une trajectoire de (X_k) pour $0 \leq k \leq 10000$.

7.3 Ruine du joueur

Soit $p \in]0; 1[$ et $q = 1 - p$. Un joueur joue à un jeu de hasard avec une fortune initiale $a \in \{1, 2, \dots\}$. À chaque tour, il gagne 1 euro avec probabilité p et perd un euro avec probabilité q . Soit c un entier supérieur à a . Le joueur décide de continuer à jouer jusqu'à ce que sa fortune soit égale à c ou bien qu'il soit ruiné.

On modélise cette situation par une chaîne de Markov X_n à espace d'états \mathbb{Z} avec $X_0 = 0$ et

$$\mathbb{P}(X_{n+1} = y \mid X_n = x) = \begin{cases} p & \text{si } y = x + 1, \\ q & \text{si } y = x - 1, \\ 0 & \text{sinon.} \end{cases}$$

Pour $x \in \mathbb{Z}$, on note

$$T_x = \inf \{k \geq 0 \mid X_k = x\},$$

avec $\inf \emptyset = +\infty$. Soient a et b dans $\{1, 2, \dots\}$ et

$$\tau = \min \{T_{-a}, T_b\} \stackrel{\text{not.}}{=} T_{-a} \wedge T_b.$$

On s'intéresse à la probabilité que le joueur s'arrête en ayant gagné b euros par rapport à sa mise de départ :

$$\alpha = \mathbb{P}(\tau = T_b).$$

On note pour tout $n \geq 0$, $\mathcal{F}_n = \sigma(X_0, X_1, X_2, \dots, X_n)$ la tribu engendrée par les variables aléatoires X_0, X_1, \dots, X_n .

Exercice 7.4. Cas où $p = q = \frac{1}{2}$.

1. Faire en Scilab un nombre important de simulation dans le cas où $a = b = 10$ pour donner une estimation de α .
2. Montrer que pour tout $x \in \mathbb{Z}$, T_x est un temps d'arrêt par rapport à la filtration $(\mathcal{F}_n)_{n \geq 0}$. En déduire que τ est également un temps d'arrêt.
3. Montrer que τ est presque sûrement fini (on pourra par exemple considérer la chaîne de Markov à espace d'états $\{-a, -a + 1, \dots, b - 1, b\}$ ayant les mêmes probabilités de transition que la chaîne de départ sauf en b et en $-a$ où elle est réfléchie, c'est-à-dire va respectivement en $b - 1$ et en $-a + 1$ avec probabilité 1).
4. Montrer que $(X_n)_{n \geq 0}$ est une martingale par rapport à la filtration \mathcal{F}_n .
5. En déduire que $(X_{\tau \wedge n})$ est une martingale. Montrer que cette dernière est uniformément intégrable.
6. En déduire la valeur de α en fonction de a et b .

Exercice 7.5. Cas où $p \neq q$

On reprend les notations de l'exercice précédent.

1. Faire d'autres simulations, par exemple pour $p = 0,4$ et $a = b = 10$.
2. Montrer que le processus, défini pour $n \geq 0$ par

$$M_n = \left(\frac{q}{p}\right)^{X_n}$$

est une martingale.

3. En déduire la valeur de α dans ce cas.

7.4 Processus de Galton-Watson

On commence par se donner une loi de probabilité η sur \mathbb{N} , appelée *loi de reproduction*. Le processus de Galton-Watson a été introduit au départ pour étudier la probabilité de survie des noms de famille. Il modélise de façon assez simple l'évolution d'une population.

On considère qu'il n'y a, au temps 0, qu'un seul individu. Celui-ci se reproduit suivant une certaine loi de probabilité appelée *loi de reproduction*. S'il n'a pas d'enfants, il y a extinction du processus. S'il a des enfants, ceux-ci se reproduisent de façon indépendante, suivant la même loi de reproduction.

On peut formaliser le modèle de la façon suivante. Soit $\eta = (\eta_k)_{k \geq 0}$ une suite de réels positifs de somme égale à 1. Avec un léger abus de notation, on assimile la suite η à la loi de probabilités qu'elle définit sur \mathbb{N} . Soit ξ une variable aléatoire de loi η , de sorte que, pour tout entier naturel k ,

$$\mathbb{P}(\xi = k) = \eta_k.$$

On note, pour n dans \mathbb{N} , Z_n , le nombre d'individus à la génération n . Ainsi, $Z_0 = 1$. On se donne une famille $(\xi_{n,i})_{n \geq 0, i \geq 1}$ de variables i.i.d. de loi η , puis on définit par récurrence

$$Z_{n+1} = \sum_{i=1}^{Z_n} \xi_{i,n} \quad n = 1, 2, \dots,$$

où, bien sûr la somme sur un ensemble d'indices vide vaut 0.

Définition 32. Le processus $(Z_n)_{n \geq 0}$ ainsi défini est appelé *processus de Galton-Watson* de loi de reproduction η .

Exercice 7.6. Simulation d'un processus de Galton-Watson

1. Nous commençons par simuler quelques lois de reproduction. Prenons le cas où η est à support fini, c'est-à-dire qu'il existe un entier N tel que pour tout $k \geq 1$, $\eta_{N+k} = 0$. Écrire une fonction `support_fini(m,n,eta)` simulant une matrice aléatoire de taille $m \times n$ dont les coefficients sont i.i.d. de loi η , et dont le paramètre `eta` est le vecteur (η_0, \dots, η_N) .
2. Soit maintenant η une loi de probabilité quelconque sur \mathbb{N} . On suppose qu'on dispose d'une fonction `xi` qui est telle que pour tout entier naturel n , `xi(n)` est un vecteur ligne dont les coefficients sont des variables aléatoires i.i.d. de loi η .
Écrire une fonction `galton_watson(n, xi)` qui renvoie un échantillon du vecteur aléatoire $(Z_0, Z_1, Z_2, \dots, Z_n)$ de loi de reproduction η .
3. À l'aide des deux questions précédentes, faire une dizaine de simulations du processus (Z_0, \dots, Z_{15}) et les tracer dans le cas

$$\eta_0 = \frac{1}{5}, \quad \eta_1 = \frac{2}{5}, \quad \eta_2 = \frac{2}{5}.$$

Estimer (avec un grand nombre de simulations) la probabilité $\mathbb{P}(Z_{15} = 0)$.

4. Soit $f(s) = \frac{1+2s+2s^2}{5}$ la fonction génératrice des probabilités de la loi η . Calculer à l'aide de Scilab le réel

$$\underbrace{f \circ f \circ \dots \circ f}_{15 \text{ fois}}(0),$$

et comparer avec le résultat obtenu à la question précédente. *Pour une justification de ce résultat, voir l'exercice suivant.*

5. Reprendre (si l'envie vous en prend) tout ou partie des deux questions précédentes, dans les cas suivants :
 - $\eta_0 = 0, \eta_1 = \eta_2 = 1/2$;
 - $\eta_0 = \frac{1}{4}, \eta_1 = \frac{1}{2}, \eta_2 = \frac{1}{4}$;
 - η est la loi géométrique de paramètre $1/2$;
 - η est la loi de poisson de paramètre 3.

Dégageons tout de même quelques résultats plus théoriques sur ce modèle. On garde les mêmes notations que précédemment. Nous nous intéressons tout d'abord à la probabilité que le processus s'éteigne, c'est-à-dire qu'il existe un entier n pour lequel $Z_n = 0$.

Le personnage principal de cette histoire est la fonction génératrice des probabilités de la loi η , que nous noterons f . Celle-ci est définie au moins sur l'intervalle $[0; 1]$ par :

$$f(s) = \sum_{k=0}^{\infty} \eta_k s^k.$$

Ses principales propriétés ont déjà été étudiées lors de l'exercice 3.3.

Lemme 33. *Pour tout entier naturel n , la fonction*

$$f_n := \underbrace{f \circ f \circ \dots \circ f}_{n \text{ fois}}$$

est la fonction génératrice des probabilités de la variable aléatoire Z_n .

Démonstration. Le résultat est vrai au rang $n = 1$, car la loi de Z_1 est η . On raisonne ensuite par récurrence. Soit n un entier naturel. Par définition, la fonction génératrice g_{n+1} de Z_{n+1} vaut

$$g_{n+1}(s) = \mathbb{E} \left[s^{Z_{n+1}} \right].$$

On utilise ensuite le fait que

$$Z_{n+1} = \sum_{i=1}^{Z_n} \xi_{n,i} = \sum_{k=1}^{\infty} \mathbf{1}_{\{Z_n=k\}} \sum_{i=1}^k \xi_{n,i},$$

pour obtenir en conditionnant par rapport aux valeurs possibles de Z_n et en utilisant que les variables aléatoires $\xi_{n,i}$, pour i entier naturel non nul, sont indépendantes et indépendantes de Z_n :

$$\begin{aligned} \mathbb{E} \left[s^{Z_{n+1}} \right] &= \sum_{k=0}^{\infty} \mathbb{P}(Z_n = k) \mathbb{E} \left[s^{\sum_{i=1}^k \xi_{n,i}} \mid Z_n = k \right] \\ &= \sum_{k=0}^{\infty} \mathbb{P}(Z_n = k) \prod_{i=1}^k \mathbb{E} \left[s^{\xi_{n,i}} \right] \\ &= \sum_{k=0}^{\infty} \mathbb{P}(Z_n = k) (f(s))^k \\ &= f_n(f(s)), \end{aligned}$$

la dernière égalité découlant de l'hypothèse de récurrence car on a fait apparaître la fonction génératrice de Z_n . \square

A Rappels d'intégration et de probabilités : l'appendice dont vous êtes le héros

Pour se rafraîchir la mémoire, on pourra consulter les ouvrages suivants :

- [**LG06**] Polycopié sur l'intégration et les probabilités de Jean-François Le Gall, disponible en ligne. Une référence.
- [**BP12**] Ouvrage de référence pour la théorie de la mesure, l'intégration et la transformée de Fourier aux niveaux L3 et M1. Très détaillé.
- [**Rud87**] Pas toujours facile d'accès, mais un ouvrage d'analyse très complet et profond.
- [**Can13**] Un excellent livre d'introduction aux probabilités, de niveau L3, très détaillé, avec des exercices souvent abordables et corrigés.
- [**Dur10**] Une très bonne référence en anglais qui aborde certains points parfois absents des références précédentes.
- [**RS12**] Le chapitre 10 liste, souvent sans démonstrations, les théorèmes classiques les plus méconnus des candidats à l'agrégation.
- [**Tou99**] Le premier chapitre est une bonne liste des résultats importants d'intégration et de probabilité.

1. Dans la théorie de l'intégration de Lebesgue, il y a trois théorèmes sur l'interversion limite-intégrale pour les suites de fonctions. Les énoncer. Donner un ordre possible de démonstration.
2. Donner un énoncé possible et une démonstration pour le théorème de continuité sous le signe intégrale.
3. Faire de même pour le théorème de dérivabilité sous le signe intégrale.

4. Énoncer le théorème de Fubini-Tonelli et le théorème de Fubini-Lebesgue. Rappeler comment on utilise en pratique le théorème de Fubini-Lebesgue.
5. Donner des conditions pratiques pour qu'une application C^1 entre deux ouverts de \mathbb{R}^n soit un C^1 -difféomorphisme.
6. Énoncer le théorème de changement de variable dans \mathbb{R}^n pour l'intégrale de Lebesgue.
7. Énoncer et démontrer le changement de coordonnées en polaires.
8. Rappeler la définition de la convergence en loi. Donner un critère pour une suite de variables aléatoires à valeurs discrètes. Donner un critère utilisant la fonction caractéristique. Donner un critère utilisant la fonction de répartition.
9. Comment peut-on utiliser la fonction caractéristique pour démontrer que deux variables aléatoires sont indépendantes ?
10. Même question avec la fonction de répartition.

Sur la transformée de Laplace, une bonne référence est la partie 3.1 de [Tou99]. La transformée de Fourier est abondamment traitée dans [Can13] et [LG06]. Pour la fonction génératrice, voir par exemple, [Can13] partie VI.3 et le petit paragraphe 8.2.4 dans [LG06].

B Solutions des exercices

Corrigé de l'exercice 1.1 Le générateur de nombre aléatoire par défaut de Scilab `rand` a un état totalement décrit par un entier appelé la graine. Par défaut, Scilab initialise la graine à 0. Ainsi, chaque fois que l'on redémarre le programme, on obtient exactement les mêmes résultats. Contrairement à ce que l'on pourrait penser, c'est un très bon comportement par défaut car il permet de déboguer les simulations en permettant la reproductibilité des expériences. À chaque appel à la fonction `rand`, l'état du générateur change et donc la graine avec lui.

On peut choisir la graine avec la syntaxe `rand("seed", n)`, où `n` est une variable contenant un entier.

On peut sauvegarder l'état actuel du générateur dans une variable avec la syntaxe `n = rand("seed")`.

Le dernier script, utilisant la fonction `getdate` se sert de l'horloge interne de la machine pour choisir « au hasard » une graine.

Corrigé de l'exercice 2.1 Il y a plusieurs façons de procéder. La plus simple est bien sûr un calcul direct.

$$\mathbb{E}[U] = \int_{\mathbb{R}} x \frac{\mathbf{1}_{[a;b]}(x)}{b-a} dx = \int_a^b x \frac{dx}{b-a} = \frac{1}{b-a} \left[\frac{x^2}{2} \right]_a^b = \frac{a+b}{2}.$$

On calcule de la même manière

$$\mathbb{E}[U^2] = \int_a^b x^2 \frac{dx}{b-a} = \frac{1}{3} \frac{b^3 - a^3}{b-a} = \frac{b^2 + ab + a^2}{b-a}.$$

Enfin,

$$\text{Var}[U] = \mathbb{E}[U^2] - (\mathbb{E}[U])^2 = \frac{(b-a)^2}{12}.$$

Une remarque au passage : la variance est invariante par translation, ce qui permet de détecter d'éventuelles erreurs. Une autre possibilité est le calcul de la transformée de Laplace, voir la remarque 3.

Corrigé de l'exercice 2.2 On utilise simplement que si V suit la loi uniforme sur $[0; 1]$, alors $U := a + (b-a)V$ suit la loi uniforme sur $[a; b]$.

```
function U = runif(m, n, a, b)
    y = a + (b - a)*rand(m,n);
endfunction
```

Corrigé de l'exercice 2.3 Comme X est bornée, sa fonction génératrice des moments est définie sur \mathbb{R} . Le calcul suivant est classique, il utilise la somme des termes d'une suite géométrique et une méthode de symétrisation des quotients bien connue.

$$\mathbb{E} \left[e^{tX} \right] = \frac{1}{n} \sum_{k=a}^b e^{tk} = \frac{1}{n} \frac{e^{at} - e^{t(b+1)}}{1 - e^t} = \frac{1}{n} \frac{e^{\frac{a+b+1}{2}t} \sinh \frac{nt}{2}}{e^{\frac{t}{2}} \sinh \frac{t}{2}}.$$

Comme la transformée de Laplace est finie sur \mathbb{R} , elle est analytique (et même entière) sur \mathbb{R} et admet pour développement en série entière

$$\mathbb{E} \left[e^{tX} \right] = 1 + \sum_{k=1}^{\infty} \mathbb{E} \left[X^k \right] \frac{t^k}{k!}.$$

Ainsi, pour connaître les moments (c'est-à-dire les valeurs $\mathbb{E} \left[X^k \right]$) de X , il suffit de faire un développement limité de $M_X(t)$ au voisinage de 0.

On pose $\alpha = \frac{a+b}{2}$. Les formules classiques sur les développements limités donnent :

$$\begin{aligned} M_X(t) &= \frac{1}{n} \left(1 + \alpha t + \frac{\alpha^2}{2} t^2 + o(t^2) \right) \left(\frac{n}{2} t + \frac{n^3}{8 \times 6} t^3 + o(t^3) \right) \left(\frac{1}{2} t + \frac{1}{8 \times 6} t^3 + o(t^3) \right)^{-1} \\ &= \left(1 + \alpha t + \frac{\alpha^2}{2} t^2 + o(t^2) \right) \left(1 + \frac{n^2}{24} t^2 + o(t^2) \right) \left(1 - \frac{1}{24} t^2 + o(t^2) \right) \\ &= 1 + \alpha t + \frac{t^2}{2} \left(\frac{(a+b)^2}{2} + \frac{n^2 - 1}{12} \right) + o(t^2). \end{aligned}$$

En identifiant les coefficients, on trouve bien

$$\mathbb{E} [X] = \frac{a+b}{2} \quad \text{et} \quad \mathbb{E} [X^2] = \left(\frac{a+b}{2} \right)^2 + \frac{n^2 - 1}{12}.$$

Donc la variance est bien celle donnée dans l'exercice.

On peut retrouver la formule donnant la somme des carrés des n premiers entiers, en prenant $a = 1$, $b = n$ et en remarquant que dans ce cas,

$$\sum_{k=1}^n k^2 = n \times \mathbb{E} [X^2] = n \left(\left(\frac{n+1}{2} \right)^2 + \frac{n^2 - 1}{12} \right) = \frac{n(2n^2 + 3n + 1)}{6} = \frac{n(n+1)(2n+1)}{6}.$$

On pourrait, bien sûr procéder de la même manière pour calculer la somme des cubes, il faudrait alors faire un développement limité à l'ordre 3 de $M_X(t)$ (ce qui n'est pas beaucoup plus long car il suffit de rajouter le terme d'ordre 3 dans l'exponentielle).

Remarque 34. Il y a bien sûr d'autres façons de faire. Rien qu'en restant avec ce point de vue probabiliste, on aurait ici pu calculer la fonction génératrice des probabilités (car X est à valeurs entières) et trouver son moment d'ordre 2 en utilisant sa dérivée seconde en 1.

Sans utiliser de probabilité, il existe beaucoup de méthodes élémentaires. Mais en général le problème de la somme des puissances des entiers consécutifs est un problème qui n'est pas si simple. Vous pouvez, par exemple, consulter la page Wikipedia sur les nombres de Bernoulli si cela vous intéresse.

Corrigé de l'exercice 2.4 Il suffit de remarquer que, si U a la loi uniforme sur $[0; 1]$, alors $V := a + \lfloor (b - a + 1)U \rfloor$ a la loi uniforme sur $\llbracket a; b \rrbracket$. Une possibilité de mise en œuvre en Scilab est la suivante.


```
function U = runif_int(m,n,a,b)
    U = a + floor((b-a+1) * rand(m,n));
endfunction
```

Corrigé de l'exercice 3.1

1. Il suffit de remarquer que si X et Y sont indépendantes, de carré intégrable, alors XY est intégrable (par l'inégalité de Cauchy-Schwarz) et $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$, donc

$$\begin{aligned}\text{Var}[X+Y] &= \mathbb{E}[(X+Y)^2] - (\mathbb{E}[X] + \mathbb{E}[Y])^2 \\ &= \mathbb{E}[X^2] + \mathbb{E}[Y^2] + 2\mathbb{E}[X]\mathbb{E}[Y] - \mathbb{E}[X^2] - \mathbb{E}[Y^2] - 2\mathbb{E}[X]\mathbb{E}[Y]. \\ &= \text{Var}[X] + \text{Var}[Y].\end{aligned}$$

2. Soient Y_1, \dots, Y_n indépendantes de loi de Bernoulli de paramètre p et $X = \sum_{i=1}^n Y_i$. Alors X a bien pour loi $\mathcal{B}(n, p)$. Par définition, on a l'égalité entre événements

$$\{X = k\} = \bigsqcup_{A \subset \llbracket 1;n \rrbracket, \text{Card}(A)=k} \{\forall i \in A, Y_i = 1 \text{ et } \forall j \notin A, Y_j = 0\}.$$

Chacun des événements de l'union disjointe a pour probabilité $p^k(1-p)^{n-k}$. Il ne reste qu'à compter le nombre de parties à k éléments de l'ensemble à n éléments $\llbracket 1;n \rrbracket$, ce qui, par définition est $\binom{n}{k}$. Pour l'espérance de X , il suffit d'utiliser la linéarité :

$$\mathbb{E}[X] = \sum_{k=1}^n \mathbb{E}[Y_k] = np.$$

Pour la variance, par indépendance,

$$\text{Var}[X] = \sum_{k=1}^n \text{Var}[Y_k] = np(1-p).$$

Corrigé de l'exercice 3.2

1. Il suffit de remarquer que, si U suit la loi uniforme sur $[0; 1]$, alors la variable aléatoire $\mathbf{1}_{\{U \leq p\}}$ suit la loi de Bernoulli de paramètre p .

```
function Y = rbernoulli(m,n,p)
    Y = double(rand(m,n) < p);
endfunction
```

Ici, on a utilisé la fonction `double` pour convertir les booléens en flottants à double précision. Ce n'est pas obligatoire et même superflu dans la plupart des cas.

2. On utilise la définition. Si l'on ne cherche pas à vectoriser, voici une première solution avec boucles, qui n'est pas très efficace en Scilab mais a le mérite d'être assez simple.

```
function Y = rbinomiale_avec_boucles(m,n,r,p)
    Y = zeros(m,n); // préallocation
    // parcours de Y
    for j = [1:n]
        for i = [1:m]
            Y(i,j) = sum( rand(1, r) < p );
        end
    end
endfunction
```

Si l'on cherche à vectoriser, il faut en fait créer un tableau de variables de Bernoulli en 3 dimensions, sommer suivant la première dimension, et effacer la dimension inutile (la première) en utilisant la fonction `matrix` qui sert à redimensionner une matrice.

```
function Y = rbinomiale(m,n,r,p)
    bern = (rand(r,m,n) < p);
    Y = matrix(sum(bern, "r"), m,n);
endfunction
```

3. On commence par calculer la fonction de répartition F de la loi binomiale, avec le calcul de sa loi d'abord⁵ dans la fonction `binomiale_probas` puis un appel à `cumsum`. On utilise la méthode de l'inverse de la fonction de répartition. Étant donné une variable aléatoire uniforme sur $[0; 1]$, on cherche (avec la fonction `gsearch`) le plus petit entier k tel que

$$U \leq F(k).$$

Cet entier suit la loi binomiale de paramètres r et p . En raisons de problèmes numériques, ce programme ne fonctionne pas pour des valeurs trop grandes de r . Dans ce cas, utiliser la méthode directe ci-dessus ou bien l'approximation par la loi normale.

```
function P = binomiale_probas(p,r)
    // Calculer la loi B(r, p) sous la forme d'un
    // tableau à r+1 éléments
    P = zeros(1,r+1);
    P(1) = (1-p)^r;
    for i = [1:r]
        P(i+1)=P(i) * (r-i+1)/i * p/(1-p);
    end
endfunction

function Y = rbinomiale2(m,n,r,p)
    // simulation de la loi binomiale par l'inverse
    // de la fonction de répartition
    P = cumsum( binomiale_probas(p,r) );
    // !!\ Des valeurs trop elevees de r
    // entraînent des valeurs "egales" dans P
    // et donc une erreur dans dsearch
    Y = dsearch( rand(m,n), P );
endfunction
```

Corrigé de l'exercice 3.3 On note, pour $k \in \mathbb{N}$, $p_k = \mathbb{P}(X = k)$.

1. Par définition,

$$G_X(1) = \sum_{k=0}^n p_k = 1,$$

ce qui prouve que le rayon de convergence de la série entière G_X est au moins égal à 1. Ceci assure que pour tout $s \in [0; 1[$, G_X est dérivable en s et

$$G'_X(s) = \sum_{k=1}^{\infty} k p_k s^{k-1}.$$

5. On aurait aussi pu appeler directement la fonction `binomial` de Scilab.

Pour étudier la dérivée à gauche en 1, on écrit

$$\frac{G_X(s) - 1}{s - 1} = \sum_{k=1}^{\infty} p_k \frac{s^k - 1}{s - 1} = \sum_{k=1}^{\infty} p_k (1 + s + \dots + s^{k-1}).$$

Comme la fonction

$$s \mapsto 1 + s + \dots + s^{k-1}$$

est *croissante* (pour $s > 0$), on peut utiliser le théorème de convergence monotone avec la mesure de comptage :

$$\begin{aligned} \lim_{s \uparrow 1} \frac{G_X(s) - 1}{s - 1} &= \lim_{s \uparrow 1} \sum_{k=1}^{\infty} p_k (1 + s + \dots + s^{k-1}) \\ &= \sum_{k=1}^{\infty} \lim_{s \uparrow 1} p_k (1 + s + \dots + s^{k-1}) \\ &= \sum_{k=1}^{\infty} k p_k. \end{aligned}$$

Ce raisonnement reste valable lorsque $\mathbb{E}[X] = +\infty$.

2. On raisonne par récurrence. Soit $n \geq 1$. Supposons le résultat vrai au rang n et $\mathbb{E}[X^{n+1}] < \infty$. Comme le rayon de convergence de G_X est au moins égal à 1, pour tout $s \in [0; s[$,

$$G_X^{(n)}(s) = \sum_{k=n}^{+\infty} k(k-1)\dots(k-n+1) p_k s^{n-k+1}$$

et par hypothèse de récurrence, cette formule est encore vraie en $s = 1$. On raisonne ensuite comme ci-dessus, par convergence monotone :

$$\begin{aligned} \lim_{s \uparrow 1} \frac{G_X^{(n)}(s) - G_X^{(n)}(1)}{s - 1} &= \lim_{s \uparrow 1} \sum_{k=n+1}^{\infty} p_k (k(k-1)\dots(k-n+1)) \frac{s^{k-n+1} - 1}{s - 1} \\ &= \lim_{s \uparrow 1} \sum_{k=n+1}^{\infty} p_k (k(k-1)\dots(k-n+1)) (1 + s + \dots + s^{k-n}) \\ &= \sum_{k=1}^{\infty} \lim_{s \uparrow 1} p_k (k(k-1)\dots(k-n+1)) (1 + s + \dots + s^{k-n}) \\ &= \sum_{k=1}^{\infty} p_k (k(k-1)\dots(k-n+1)(k-n)). \end{aligned}$$

Au passage, ceci est encore valable si tous les moments $\mathbb{E}[X], \mathbb{E}[X^2], \dots, \mathbb{E}[X^n]$ sont finis mais $\mathbb{E}[X^{n+1}] = +\infty$.

3. Le calcul de la variance se fait très simplement.

$$\text{Var}[X] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \mathbb{E}[X(X-1)] + E[X] - E[X]^2.$$

Puis il suffit de remplacer $\mathbb{E}[X]$ par $G'_X(1)$ et $\mathbb{E}[X(X-1)]$ par $G''_X(1)$.

Corrigé de l'exercice 3.4 Soit $p \in]0; 1[$ et $q = 1 - p$. Soient Y_1, Y_2, \dots des variables aléatoires indépendantes de loi Bernoulli de paramètre p et

$$T = \inf \{k \geq 1 \mid Y_k = 1\}.$$

Par définition, T suit la loi géométrique de paramètre p .

1. Soit $k \geq 1$. On a l'égalité d'événements

$$\{T = k\} = \{Y_1 = 0, Y_2 = 0, \dots, Y_{k-1} = 0, Y_k = 1\},$$

d'où le résultat, par indépendance. Au passage, il serait en théorie possible que T soit égale à $+\infty$, si toutes les variables de Bernoulli étaient nulles. Comme $\sum_{k=1}^{\infty} pq^{k-1} = 1$, ceci n'arrive presque sûrement jamais.

2. On calcule la fonction génératrice avec la série géométrique. Pour tout $s \in [0; 1]$,

$$G_T(s) = \sum_{k=1}^{\infty} pq^{k-1} s^k = \frac{ps}{1 - qs}.$$

3. On en déduit l'espérance et la variance par dérivations successives.

$$G'_T(s) = \frac{p}{(1 - qs)^2},$$

donc $\mathbb{E}[T] = \frac{p}{(1-q)^2} = \frac{1}{p}$ (résultat très intuitif!).

$$G''_T(s) = \frac{2pq}{(1 - qs)^3},$$

donc

$$\mathbb{E}[T(T-1)] = \frac{2q}{p^2},$$

puis

$$\text{Var}[T] = \frac{2q}{p^2} + \frac{1}{p} - \frac{1}{p^2} = \frac{q}{p^2}.$$

Corrigé de l'exercice 3.5

1. Soit $k \geq 1$. En conditionnant successivement et en utilisant la propriété de Markov, on trouve :

$$\mathbb{P}(T = k) = \mathbb{P}(X_1 = 0, X_2 = 0, \dots, X_{k-1} = 0, X_k = 1) = q \times q \times \dots \times q \times p = q^{k-1}p,$$

donc T a bien la loi géométrique de paramètre p .

2. On travaille avec la chaîne canonique et on note Θ l'opérateur de décalage. On note que sur l'événement $\{X_1 = 0\}$, on a $T = 1 + T \circ \Theta$, puis on écrit

$$\begin{aligned} \mathbb{E}[T^k \mid X_1] &= \mathbb{E}\left[1 \times \mathbf{1}_{\{X_1=1\}} + T^k \mathbf{1}_{\{X_1=0\}} \mid X_1\right] \\ &= \mathbf{1}_{\{X_1=1\}} + \mathbf{1}_{\{X_1=0\}} \mathbb{E}\left[(1 + T \circ \Theta)^k \mid X_1\right]. \end{aligned}$$

D'après la propriété de Markov faible au temps 1,

$$\mathbf{1}_{\{X_1=0\}} \mathbb{E}\left[(1 + T \circ \Theta)^k \mid X_1\right] = \mathbf{1}_{\{X_1=0\}} \mathbb{E}\left[(1 + T)^k\right].$$

En prenant l'espérance, on trouve

$$\mathbb{E}[T^k] = p + q \mathbb{E}\left[(1 + T)^k\right],$$

d'où le résultat, en isolant $\mathbb{E}[T^k]$ et en divisant par p .

3. Pour $k = 1$, on trouve directement

$$\mathbb{E}[T] = 1 + \frac{q}{p} = \frac{1}{p}.$$

Pour $k = 2$, on trouve

$$\mathbb{E}[T^2] = 1 + \frac{q}{p}(1 + 2\mathbb{E}[T]) = \frac{p^2 + qp + 2q}{p^2} = \frac{p^2 + (1-p)p + 2q}{p^2} = \frac{p + 2q}{p^2}.$$

La variance vaut donc

$$\mathbb{E}[T^2] - (\mathbb{E}[T])^2 = \frac{p + 2q}{p^2} - \frac{1}{p^2} = \frac{q}{p^2}.$$

Corrigé de l'exercice 3.6

1. Une mise en œuvre possible du premier algorithme est la suivante.

```
function Y = rgeom(m,n,p)
    // Pre-allocation de Y
    Y = zeros(m,n);
    // Parcours de Y
    for j = [1:n]
        for i = [1:m]
            //simulation avec boucle tant que
            k = 0; // nombre d'essais
            succes = %f; //false
            while (~succes) // ~ pour négation
                k = k + 1;
                if (rand() < p) //succes
                    Y(i,j) = k;
                    succes = %t; //true
                end
            end
        end
    end
endfunction
```

2. On a l'égalité des événements :

$$\left\{ \lfloor \frac{\ln(U)}{\ln(q)} \rfloor + 1 = k \right\} = \left\{ k-1 \leq \frac{\ln(U)}{\ln(q)} < k \right\} = \left\{ q^{k-1} \geq U > q^k \right\}$$

Ainsi,

$$\mathbb{P} \left(\lfloor \frac{\ln(U)}{\ln(q)} \rfloor + 1 = k \right) = q^{k-1} - q^k = q^{k-1}p,$$

donc $\lfloor \frac{\ln(U)}{\ln(q)} \rfloor + 1$ suit bien la loi géométrique de paramètre p .

3.

```
function Y = rgeom2(m,n,p)
    Y = floor( log(rand(m,n))/log(1-p) ) + 1;
endfunction
```

Corrigé de l'exercice 3.7

1. Soit $s \in [0; 1]$.

$$\begin{aligned} G_N(s) &= \sum_{k=0}^{\infty} e^{-\lambda} \frac{\lambda^k}{k!} s^k \\ &= e^{-\lambda} \sum_{k=0}^{\infty} \frac{(\lambda s)^k}{k!} = e^{-\lambda} e^{\lambda s} = e^{\lambda(s-1)}. \end{aligned}$$

2. On dérive deux fois la fonction génératrice :

$$\begin{aligned} G'_N(s) &= \lambda e^{\lambda(s-1)}, \\ G''_N(s) &= \lambda^2 e^{\lambda(s-1)}. \end{aligned}$$

On en déduit que

$$\mathbb{E}[N] = G'_N(1) = \lambda,$$

et

$$\mathbb{E}[N(N-1)] = G''_N(1) = \lambda^2.$$

Finalement,

$$\text{Var}[N] = \lambda^2 + \lambda - \lambda^2 = \lambda.$$

3. Soit $k \in \mathbb{N}$ fixé. Soit n assez grand pour avoir $\frac{\lambda}{n} < 1$ et $n \geq k$. Comme les lois considérées sont à valeurs discrètes⁶, il suffit, pour étudier la convergence en loi de montrer que pour tout $k \in \mathbb{N}$,

$$\mathbb{P}[X_n = k] \xrightarrow{n \rightarrow \infty} \mathbb{P}[N = k].$$

Il suffit de calculer :

$$\mathbb{P}[X_n = k] = \binom{n}{k} \left(\frac{\lambda}{n}\right)^k \left(1 - \frac{\lambda}{n}\right)^{n-k} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!} \frac{\lambda^k}{n^k} \left(1 - \frac{\lambda}{n}\right)^{n-k}.$$

On a les équivalents suivants :

$$\begin{aligned} n(n-1)(n-2)\cdots(n-k+1) &\sim_{n \rightarrow \infty} n^k, \\ \left(1 - \frac{\lambda}{n}\right)^{n-k} &\xrightarrow{n \rightarrow \infty} e^{-\lambda}. \end{aligned}$$

Finalement,

$$\mathbb{P}[X_n = k] \xrightarrow{n \rightarrow \infty} e^{-\lambda} \frac{\lambda^k}{k!} = \mathbb{P}[N = k].$$

Corrigé de l'exercice 3.8

1. Soit f une fonction mesurable positive. Par définition,

$$\mathbb{E}\left[f\left(\sum_{i=1}^n X_i\right)\right] = \int \cdots \int f\left(\sum_{i=1}^n x_i\right) \lambda^n e^{-\lambda \sum_{i=1}^n x_i} \prod_{i=1}^n \mathbf{1}_{\{x_i \geq 0\}} dx_i.$$

On fait le changement de variable linéaire bijectif, de jacobien 1 :

$$\begin{aligned} x_1 &= x'_1 \\ x_2 &= x'_2 - x'_1 \\ &\vdots \\ x_n &= x'_n - x'_{n-1}. \end{aligned}$$

6. ATTENTION! C'est totalement faux dans les autres cas!

Ainsi, l'intégrale devient

$$\mathbb{E} \left[f \left(\sum_{i=1}^n X_i \right) \right] = \int \cdots \int f(x'_n) \lambda^n e^{-\lambda x'_n} \mathbf{1}_{\{0 \leq x'_1 \leq x'_2 \leq \cdots \leq x'_n\}} dx'_1 \cdots dx'_n.$$

Une récurrence permet de calculer :

$$\int \cdots \int \mathbf{1}_{\{0 \leq x'_1 \leq \cdots \leq x'_{n-1} \leq x'_n\}} dx'_1 dx'_2 \cdots dx'_{n-1} = \frac{x'_n{}^{n-1}}{(n-1)!},$$

finalemt (en posant $x = x'_n$),

$$\mathbb{E} \left[f \left(\sum_{i=1}^n X_i \right) \right] = \int f(x) \frac{x^{n-1}}{(n-1)!} \lambda^n e^{-\lambda x} \mathbf{1}_{\{x \geq 0\}} dx$$

donc la loi de $\sum_{i=1}^n X_i$ est la loi $\Gamma(n, \lambda)$.

2. Comme, pour tout $i \geq 1$, on a $U_i \leq 1$,

$$\{N \geq n\} = \{U_1 \cdots U_n \geq e^{-\lambda}\} = \left\{ \sum_{i=1}^n \frac{-\log(U_i)}{\lambda} \leq 1 \right\}.$$

En posant $X_i = -\frac{\log(U_i)}{\lambda}$, on obtient des variables indépendantes de loi exponentielle de paramètre λ . Finalement, $\mathbb{P}(N \geq n) = \mathbb{P}(Z \leq 1)$, où Z suit la loi $\Gamma(n, \lambda)$.

3. Si $n \geq 2$, on obtient en intégrant par parties,

$$\begin{aligned} \mathbb{P}(N \geq n) &= \int_0^1 \frac{x^{n-1}}{(n-1)!} \lambda^n e^{-\lambda x} dx \\ &= -\frac{\lambda^{n-1}}{(n-1)!} e^{-\lambda} + \int_0^1 \frac{x^{n-2}}{(n-2)!} \lambda^{n-1} e^{-\lambda x} dx. \end{aligned}$$

On en déduit que

$$\mathbb{P}(N \geq n) - \mathbb{P}(N \geq n-1) = -\frac{\lambda^{n-1}}{(n-1)!} e^{-\lambda},$$

puis que

$$\mathbb{P}(N = n-1) = \frac{\lambda^{n-1}}{(n-1)!} e^{-\lambda},$$

donc N suit la loi de Poisson de paramètre λ .

4. Une mise en œuvre possible de cet algorithme est la suivante :

```
function Y = rpoisson(m, n, lambda)
    Y = zeros(m,n);
    barriere = exp(-lambda);
    for i = [1:m]
        for j = [1:n]
            // simulation
            nb_unifs = 0;
            produit = 1.;
            while %t
                u = rand();
                produit = produit * u;
                if (produit <= barriere)
```

```

                break;
            end
            nb_unifs = nb_unifs + 1;
        end
        // nb_unifs suit la loi poisson(lambda)
        // fin de la simulation
        Y (i,j) = nb_unifs;
    end
end
endfunction

```

Corrigé de l'exercice 4.1 Éléments de correction.

1. On montre assez facilement, en utilisant la croissance de f que, pour tout x dans l'intérieur de I ,

$$\lim_{\substack{y \rightarrow x \\ y < x}} f(y) = \sup \{f(y) \mid y < x\}.$$

De même pour la limite à droite. L'inégalité est évidente.

Au passage, on peut raisonner de la même manière lorsque x est une borne (incluse ou non) de l'intervalle I , mais les limites infinies peuvent survenir.

2. D'après ce qui précède, x est un point de discontinuité si et seulement si $f(x-) < f(x+)$. Dans ce cas, l'intervalle $]f(x-); f(x+)[$ contient des rationnels. Choisissons-en un, disons q_x . Par croissance de f , le rationnel q_x n'appartient à aucun autre intervalle du même type. Ainsi les points de discontinuité de f s'injectent dans l'ensemble des rationnels et sont donc en quantité dénombrable.
3. **Monotonie** Si x et x' sont deux réels tels que $x \leq x'$, alors $]-\infty; x] \subset]-\infty; x']$ donc $\mathbb{P}_X (]-\infty; x]) \leq \mathbb{P}_X (]-\infty; x'])$.

Limites Comme $\bigcap_{N=1}^{\infty}]-\infty; -N] = \emptyset$, et que cette intersection est décroissante, on a

$$\lim_{k \rightarrow -\infty} \mathbb{P}_X (]-\infty; k]) = \mathbb{P}_X (\emptyset) = 0.$$

On procède de même avec la réunion croissante

$$\bigcup_{N=1}^{\infty}]-\infty; N] = \mathbb{R}.$$

Continuité à droite Comme F est croissante, il suffit de montrer que pour tout réel x ,

$$F\left(x + \frac{1}{n}\right) \xrightarrow{n \rightarrow \infty} F(x),$$

ce qui découle (l'intersection étant décroissante) de

$$\lim_{n \rightarrow \infty} \mathbb{P}_X \left(]-\infty; x + \frac{1}{n}]\right) = \mathbb{P}_X \left(\bigcap_{n \geq 1}]-\infty; x + \frac{1}{n}]\right) = \mathbb{P}_X (]-\infty; x]).$$

Le fait que F admette des limites à gauche en tout point est une conséquence de sa monotonie.

4. Soit U un ouvert. On écrit

$$U = \bigsqcup_{i \in I} U_i,$$

où les U_i sont ses composantes connexes (donc disjointes). Étant ouvertes (et fermées) dans l'ouvert U , elles sont ouvertes dans \mathbb{R} et comme ce sont des connexes de \mathbb{R} , ce sont des intervalles ouverts. Chacune d'entre elles possède un rationnel. Comme elles sont disjointes deux à deux, on peut injecter l'ensemble d'indices I dans l'ensemble dénombrable \mathbb{Q} .

$$\begin{aligned}
 5. \quad \mathbb{P}(X \in]a; b]) &= \mathbb{P}(X \in]-\infty; b]) - \mathbb{P}(X \in]-\infty; a]) \\
 &= \mathbb{P}_X \left(\bigcup_{n=1}^{\infty}]-\infty; b - \frac{1}{n}] \right) - F(a) = F(b-) - F(a).
 \end{aligned}$$

6. La fonction F caractérise la loi de X sur les intervalles ouverts d'après le point précédent, donc sur les unions dénombrables disjointes de tels intervalles, c'est-à-dire tous les ouverts. Comme l'ensemble des ouverts est une famille de parties stable par intersections finies qui engendre la tribu borélienne, par le lemme de classe monotone, F caractérise la loi de X sur la tribu borélienne.

Corrigé de l'exercice 4.2

1. Si $0 < u < u' < 1$, alors, par croissance de F ,

$$\{x \in \mathbb{R} \mid F(x) \geq u'\} \subset \{x \in \mathbb{R} \mid F(x) \geq u\},$$

donc

$$\inf \{x \in \mathbb{R} \mid F(x) \geq u'\} \geq \inf \{x \in \mathbb{R} \mid F(x) \geq u\},$$

donc F^{-1} est croissante sur $]0; 1[$.

2. Soit $u \in]0; 1[$. Alors, pour tout entier $n \geq 1$,

$$F^{-1}(u) + \frac{1}{n} > F^{-1}(u),$$

donc par définition de F^{-1} ,

$$F\left(F^{-1}(u) + \frac{1}{n}\right) \geq u$$

Comme F est continue à droite, on en déduit

$$F\left(F^{-1}(u)\right) = \lim_{n \rightarrow \infty} F\left(F^{-1}(u) + \frac{1}{n}\right) \geq u.$$

3. Comme F^{-1} est croissante, il suffit de montrer que $F^{-1}(u-) = F^{-1}(u)$ pour tout $u \in]0; 1[$. On sait que $F^{-1}\left(u - \frac{1}{n}\right)$ tend *en croissant* vers $F^{-1}(u-)$. Comme F est continue à droite, on a alors

$$\lim_{n \rightarrow \infty} F\left(F^{-1}\left(u - \frac{1}{n}\right)\right) = F\left(F^{-1}(u-)\right).$$

Comme pour tout $n \geq 1$,

$$F^{-1}(u-) \geq F^{-1}\left(u - \frac{1}{n}\right),$$

On obtient, en appliquant la fonction croissante F , en utilisant l'inégalité de la question précédente puis en faisant tendre n vers $+\infty$,

$$F\left(F^{-1}(u-)\right) \geq u.$$

Donc par définition, $F^{-1}(u-) \geq F^{-1}(u)$ et par conséquent, $F^{-1}(u-) = F^{-1}(u)$.

4. L'implication directe découle directement de la définition de F^{-1} . Puis, si $F^{-1}(u) \leq x$, en appliquant la fonction croissante F et en utilisant l'inégalité de la question 3,

$$F(x) \geq F\left(F^{-1}(u)\right) \geq u.$$

5. Si U suit la loi uniforme sur $[0; 1]$, alors, d'après la question précédente,

$$\mathbb{P}\left(F^{-1}(U) \leq x\right) = \mathbb{P}(U \leq F(x)) = F(x).$$

Ainsi, les variables aléatoires X et $F^{-1}(U)$ ayant la même fonction de répartition, ont la même loi.

Corrigé de l'exercice 4.3

1. Pour la loi de Poisson (on évite de calculer les puissances et les factorielles à chaque étape pour des raisons de performance), on peut procéder comme suit.

```
function Y = rpoisson2(m,n,lambda)
    Y = zeros(m,n);
    for i = [1:m]
        for j = [1:n]
            U = rand();
            s = 0;
            k = 0;
            p = exp(-lambda);
            while %t
                if U < p
                    break;
                end
                k = k + 1;
                p = p * lambda / k;
            end
            Y(i,j) = k;
        end
    end
endfunction
```

2. En utilisant les fonctions `cumsum` et `dsearch`, on peut vectoriser le code et obtenir un programme très efficace.

```
function Y = rfinite(m, n, P)
    C = cumsum(P);
    U = rand(m,n);
    Y = dsearch(U, C);
endfunction
```

Corrigé de l'exercice 4.4

1. Soit F la fonction de répartition de la loi exponentielle de paramètre $\lambda > 0$. Des calculs élémentaires montrent que la fonction de répartition de la loi exponentielle vaut

$$F(x) = 1 - e^{-\lambda x}.$$

En résolvant l'équation d'inconnue u , $F(x) = u$, on trouve que la fonction F est bijective sur $]0; 1[$, de « vraie » inverse

$$F^{-1}(u) = -\frac{\ln(1-u)}{\lambda}.$$

2. On sait donc que si U est uniforme sur $[0; 1]$, la variable aléatoire $-\frac{\ln(1-U)}{\lambda}$ suit la loi exponentielle de paramètre λ . Bien sûr, comme $1 - U$ suit également la loi uniforme sur $[0; 1]$, il en est de même de $\frac{-\ln(U)}{\lambda}$. Le programme fait une ligne.

```
function Y = rexp(m, n, lambda)
    Y = - log(rand(m,n)) / lambda;
endfunction
```

3. Soit X suivant la loi de Cauchy centrée réduite et F sa fonction de répartition. En calculant l'intégrale

$$\frac{1}{\pi} \int_{-\infty}^x \frac{1}{1+t^2} dt = \frac{1}{\pi} \left(\arctan(x) - \frac{\pi}{2} \right),$$

on trouve que la fonction de répartition de X est $F(x) = \frac{1}{\pi} \arctan(x) + \frac{1}{2}$. Elle est bijective sur \mathbb{R} et son inverse est

$$F^{-1}(u) = \tan\left(\pi u - \frac{\pi}{2}\right).$$

4. D'après le critère de Riemann, l'intégrale

$$\int_{-\infty}^{\infty} |t| \frac{1}{1+t^2} dt$$

diverge, donc la loi de Cauchy n'admet pas d'espérance.

Corrigé de l'exercice 5.1

- Il suffit de tirer des variables uniformes sur C sous la forme d'un couple (U, V) avec U et V indépendantes de lois uniforme sur $[0; 1]$ et d'accepter le tirage lorsque $U^2 + V^2 \leq 1$. D'après la partie précédente, la loi du temps d'attente d'un tirage accepté est la loi géométrique de paramètre $\pi/4$, donc d'espérance $4/\pi$.
- Il suffit d'appliquer la loi des grands nombres avec les variables aléatoires indépendantes de Bernoulli $\mathbf{1}_{\{U_k \in \mathcal{Q}\}}$ d'espérances $\pi/4$.
- Voici un exemple de script possible.

```
// Calcul de pi par la méthode de Monte-Carlo
num_tirages = 10000;
num_succes = 0;
for i = [1:num_tirages]
    U = rand(1, 2);
    if ( sum (U .^ 2) <= 1 )
        num_succes = num_succes + 1;
    end
end
approx_pi = 4 * num_succes / num_tirages;
printf("Approximation de pi pour %d tirages : %f", num_tirages←
, approx_pi);
```

En voici un autre, sans boucle (donc beaucoup plus rapide) :

```
// Calcul de pi par la méthode de Monte-Carlo
// Version vectorisée
num_tirages = 10000;
U = rand(1, num_tirages);
V = rand(1, num_tirages);
Z = U.^2 + V.^2;
num_succes = sum( Z <= 1);
approx_pi = 4 * num_succes / num_tirages;
printf("Approximation de pi pour %d tirages : %f", num_tirages←
, approx_pi);
```

4. Soit Y_1, Y_2, \dots , des variables i.i.d. de même loi de Bernoulli de paramètre $\frac{\pi}{4}$ et $S_n = \sum_{k=1}^n Y_k$. Pour alléger les notations et gagner en généralité, on pose $p = \frac{\pi}{4}$, $q = 1 - p$, $\varepsilon = 0,01$ et $\alpha = 0,05$. On fait donc l'hypothèse que n est assez grand pour assimiler $\frac{S_n}{n}$ à une loi normale $\mathcal{N}(p, \frac{pq}{n})$. On cherche $n \geq 1$ tel que

$$\mathbb{P}\left(\left|\frac{S_n}{n} - p\right| \leq \varepsilon\right) \geq 1 - \alpha.$$

On pose $\frac{S_n}{n} = p + \sqrt{\frac{pq}{n}}N$, où N suit la loi normale centrée réduite. L'inéquation ci-dessus est équivalente à

$$\mathbb{P}\left(|N| \geq \varepsilon \sqrt{\frac{n}{pq}}\right) \geq 1 - \alpha.$$

Ce qui, pour des raisons de symétrie, équivaut encore à

$$\mathbb{P}\left(N \geq \varepsilon \sqrt{\frac{n}{pq}}\right) \geq 1 - \frac{\alpha}{2}.$$

On applique la fonction quantile F_X^{-1} pour obtenir finalement

$$n \geq \frac{pq \left(F_N^{-1}\left(1 - \frac{\alpha}{2}\right)\right)^2}{\varepsilon^2}.$$

Ce qui est important à noter est la proportionnalité à ε^{-2} . L'application numérique donne ici environ $n \approx 4600$ avec le script suivant.

```
p = %pi / 4;
q = 1 - p;
eps = 0.01;
alpha = 0.05;
Fn_m1 = cdfnor("X", 0, 1, 1-alpha, alpha);
n_min = p * q * Fn_m1^2 / eps^2;
disp(n_min);
```

Corrigé de l'exercice 5.2

1. Il est évident que f est positive et d'intégrale 1. Sa primitive nulle en $-\infty$, c'est-à-dire sa fonction de répartition est $F(X) = x^3$ et l'inverse de sa fonction de répartition est primitive nulle en $-\infty$, c'est-à-dire sa fonction de répartition est $F(X) = x^3$ et l'inverse de sa fonction de répartition est $F^{-1}(X) = \sqrt[3]{x}$. Le programme suivant simule la loi de densité f .

```
function Y = inv_rep(m,n)
    Y = rand(m,n) .^ (1. / 3.);
endfunction
```

2. On doit trouver une fonction de densité g d'une loi connue et facilement simulable, et une constante c la moins grande possible telle que pour tout $x \in [0; 1]$, $f(x) \leq cg(x)$. On va utiliser ici la loi uniforme sur $[0; 1]$ (donc ici g est constante et égale à 1) et $c = 3$. Pour simuler X , on commence par tirer V suivant la loi uniforme sur $[0; 1]$ avec `rand`, puis U une autre variable uniforme sur $[0; 1]$ indépendante de V . On accepte V comme un tirage de X si

$$U \leq \frac{f(V)}{cg(V)} = \frac{3V^2}{3 \times 1} = V^2.$$

```

function Y = rejet(m, n)
    Y = zeros(m,n);
    for i = 1:m
        for j = 1:n
            while %t
                essai = rand();
                valeur_test = rand();
                if valeur_test <= essai^2
                    break; // on accepte
                end
            end
            Y(i,j) = essai;
        end
    end
endfunction

```

3. Le nombre d'itération dans la boucle `while` est aléatoire, de loi géométrique de paramètre $\frac{1}{c}$ donc son espérance est $c = 3$. Si l'on demande un nombre N important de valeurs, d'après la loi des grands nombres, le nombre moyen d'itérations dans la boucle sera proche de 3. Comme chaque boucle `while` fait appel deux fois à la fonction `rand`, cela fait en moyenne 6 appels à `rand` par valeur renvoyée.

Corrigé de l'exercice 6.7

```

C = [3 1 ; 1 2];
A = chol(C);
mu = [-2 ; -3];
n = 1000;
N = rand(2, n, "normal");
X = A*N;
X(1,:) = X(1,:) + mu(1);
X(2,:) = X(2,:) + mu(2);
clf();
plot(X(1,:), X(2,:), "r.");
[Q,D] = spec(C);
u = Q(:,1) * sqrt(D(1,1));
xarrows([mu(1), u(1)], [mu(2), u(2)]);
v = Q(:,2) * sqrt(D(2,2));
xarrows([mu(1), v(1)], [mu(2), v(2)]);

```

Corrigé de l'exercice 7.3

```

// marche aléatoire en dimension 2
t_max = 100000;
// déplacements : haut bas gauche droite equiprobables
D = [1 0 ; 0 1 ; -1 0 ; 0 -1];
// on choisit des nombres entre 1 et 4
// pour connaître les déplacements
Alea = grand(1, t_max, "uin", 1, 4);
// déplacements aleatoires

```

```
Z = zeros(2, t_max);
for i = [1:4]
    Z(1, Alea==i) = D(i,1);
    Z(2, Alea==i) = D(i,2);
end
Positions = cumsum(Z, "c");
plot(Positions(1,:), Positions(2,:));
```

Corrigé de l'exercice 7.6 Voici un exemple de ce que l'on pouvait faire. J'ai rangé les deux fonctions principales dans un fichier de fonctions `galtonwatson.sci` que voici.

```
// galtonwatson.sci
// fichier de fonctions pour des experiences
// sur les processus de galton-watson

function Y = support_fini(m,n,eta)
    // simule la loi definie par le vecteur
    // stochastique eta sur l'ensemble fini
    // 0, 1, ..., length(eta) - 1
    Y = dsearch(rand(m,n), cumsum(eta));
endfunction

function Z = galton_watson(n, xi)
    // renvoie une simulation de
    // (Z_0, Z_1, ..., Z_n)
    // xi(N) doit être un vecteur aléatoire
    // de taille N compose d'entiers naturels
    Z = zeros(1,n+1); Z(1) = 1; // Z(1) = Z_0
    for j = 1:n
        Z(j+1) = sum( xi(Z(j)) );
    end
endfunction
```

Pour la suite, on peut séparer en plusieurs scripts. En voici pour le tracé d'une dizaine de courbes, le fichier de script s'appelle `gw_trace.sce`. Au début du script, on appelle le fichier de fonction pour en avoir toujours la dernière version (et être sûr qu'il soit chargé).

```
// gw_trace.sce
// Script pour experimenter sur les
// processus de Galton-Watson
exec('galtonwatson.sci');
// choix de la loi
eta = [.2 .4 .4];
function Y = xi(n)
    Y = support_fini(1,n,eta);
endfunction
// un premier echantillon
n = 15
disp(galton_watson(n, xi))
```

```

// Plusieurs experiences
num_exp = 20;
Z = zeros(num_exp, n+1);
for i = 1:num_exp
    Z(i,:) = galton_watson(n, xi);
end
// Trace des num_exp trajectoires
scf(1); clf();
plot(Z'); // trace chaque colonne de Z'

```

La figure 1 montre un exemple de ce que l'on pouvait obtenir. Pour la suite de l'exercice, on

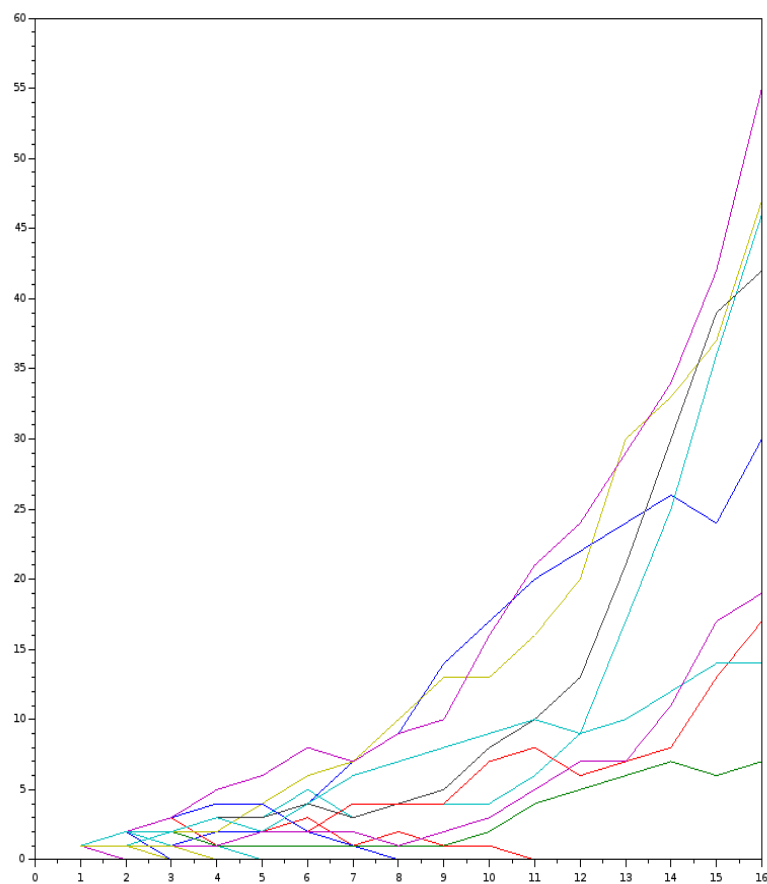


FIGURE 1 – 20 trajectoires de $(Z_0, Z_1, \dots, Z_{15})$ dans le cas où $\eta_0 = 1/5$ et $\eta_1 = \eta_2 = 2/5$.

pouvait poursuivre avec le script suivant.

```

// gw_extinction.sce
exec('galtonwatson.sci');
// estimation de la probabilité d'extinction
// au bout de n generations

```

```

n = 15
// choix de la loi
eta = [.2 .4 .4];
function Y = xi(n)
    Y = support_fini(1,n,eta);
endfunction
// estimation de P( Z_{15} = 0)
num_exp = 2000;
cpt_vivant = 0;
GW = zeros(1, n+1);
for i = 1:num_exp
    GW = galton_watson(n, xi);
    if (GW(16) ~= 0)
        cpt_vivant = cpt_vivant + 1;
    end
end
disp("fréquence empirique pour "); disp(num_exp); disp(" lancers←
: ");
disp(cpt_vivant / num_exp);
// Comparaison avec la valeur theorique
function y = f(x)
    y = (1. + 2. * x + 2. * x.^2) / 5.;
endfunction
qn = 0;
for i = 1:n
    qn = f(qn);
end
disp("valeur theorique : "); disp(1 - qn);

```

On devrait trouver des valeurs « assez proches » l'une de l'autre mais pour avoir une idée plus théorique de l'ordre de grandeur de l'écart, il faudrait davantage de renseignements sur la loi de Z_{15} (par exemple une majoration de son écart-type).

Il n'est pas très difficile de changer de loi de reproduction, il suffit simplement de changer la fonction `xi`. Voici, par exemple comment on pouvait choisir η comme étant la loi de Poisson de paramètre 3 en utilisant la fonction prédéfinie `grand`. On a dû réduire le nombre de générations à 12 pour éviter un dépassement de la pile.

```

// gw_trace_poisson.sce
exec('galtonwatson.sci');
// loi de reproduction
function Y = xi(n)
    Y = grand(1, n, "poi", 3)
endfunction
// Nombre de generations
n = 12;
num_exp = 20;
Z = zeros(num_exp, n+1);
for i = 1:num_exp
    Z(i,:) = galton_watson(n, xi);
end
// Trace des num_exp trajectoires

```



```
scf(1); clf();
plot(Z'); // trace chaque colonne de Z'
```

On peut observer dans la figure 2 que les ordres de grandeurs ne sont pas les mêmes que dans l'exemple précédent.

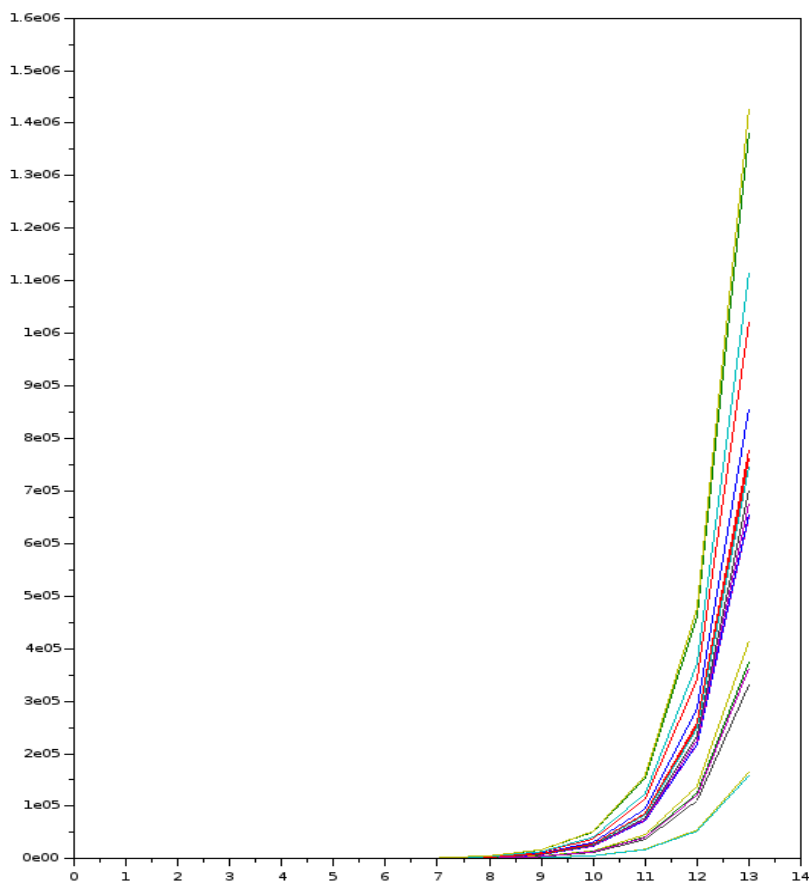


FIGURE 2 – 20 trajectoires de $(Z_0, Z_1, \dots, Z_{12})$ dans le cas où la loi de reproduction est la loi de Poisson de paramètre 3.

Références

- [BC07] Bernard Bercu and Djali Chafai. *Modélisation stochastique et simulation*. Dunod, 2007.
- [BP12] Marc Briane and Gilles Pagès. *Analyse, Théorie de l'intégration, Convolution et Transformation de Fourier, 5ème éd.* Vuibert, Paris, 365p, 2012.
- [Can13] Bernard Candelpergher. *Théorie des probabilités*. Calvage & Mounet, 2013.
- [Dur10] Rick Durrett. *Probability : theory and examples*. Cambridge university press, 2010.

- [LG06] Jean-François Le Gall. Intégration, probabilités et processus aléatoires. <http://www.math.u-psud.fr/~jfllegall/IPPA2.pdf>, 2006. Polycopié de cours.
- [LG12] Jean-François Le Gall. *Mouvement brownien, martingales et calcul stochastique*, volume 71. Springer Science & Business Media, 2012.
- [MT⁺00] George Marsaglia, Wai Wan Tsang, et al. The ziggurat method for generating random variables. *Journal of statistical software*, 5(8) :1–7, 2000.
- [RS12] Vincent Rivoirard and Gilles Stoltz. *Statistique mathématique en action*. Vuibert, 2012.
- [Rud87] Walter Rudin. *Real and complex analysis*. McGraw-Hill Higher Education, 1987.
- [Tou99] Paul S. Toulouse. *Thèmes de probabilités et statistiques*. Dunod, Paris, 1999.