



Université des Sciences et Technologies de Lille
U.F.R. de Mathématiques Pures et Appliquées
Bât. M2, F-59655 Villeneuve d'Ascq Cedex

Simulation

Charles SUQUET

Table des matières

1	Introduction	5
2	Méthode théorique pour simuler une v.a.r.	6
3	Méthodes particulières pour lois usuelles	9
3.1	Lois discrètes à support fini	9
3.2	Lois binomiales et multinomiales	11
3.3	Lois de Poisson	13
3.4	Lois géométriques	15
3.5	Lois gaussiennes	17
4	Algorithmes de rejet	17
4.1	Simulation de lois uniformes par rejet	18
4.2	Simulation de lois à densité par rejet	21
4.3	Simulation d'une loi discrète par rejet	26
5	Simulation de vecteurs aléatoires par transformation	28
5.1	Loi uniforme par transformation affine	29
5.1.1	Principe	29
5.1.2	Loi uniforme sur un parallélogramme	29
5.1.3	Loi uniforme sur un triangle ou un polygone	30
5.1.4	Loi uniforme sur un simplexe de \mathbb{R}^d	31
5.1.5	Loi uniforme sur un ellipsoïde	33
5.2	Vecteur gaussien de covariance donnée	34
6	Méthode polaire	36

Simulation de variables et vecteurs aléatoires

1 Introduction

La simulation informatique du hasard a de multiples applications : simulation de phénomènes physiques, méthodes de Monte-Carlo pour le calcul d'intégrales, étude de tests statistiques ou d'estimateurs, simulation de fonctionnements de réseaux ou de systèmes complexes, cryptographie, imagerie, algorithmes probabilistes, . . .

Théoriquement, la génération de nombres aléatoires suivant une loi donnée se ramène à la génération de suites de variables aléatoires indépendantes de loi uniforme sur $[0, 1]$. Si les X_i sont des variables de Bernoulli indépendantes de même paramètre $p = 1/2$, la v.a. $U := \sum_{k=1}^{+\infty} X_k 2^{-k}$ suit la loi uniforme sur $[0, 1]$. Le problème se ramène donc à la génération d'une suite de « bits » aléatoires indépendants pouvant prendre chacun la valeur 0 ou la valeur 1 avec même probabilité $1/2$. En d'autres termes, il suffirait de réaliser un jeu de pile ou face infini avec une pièce parfaitement équilibrée¹. Cette méthode n'est évidemment pas réaliste et en pratique on a recours à l'informatique pour « simuler » une telle suite. Pourquoi employer ici le mot « simuler » ? Parce qu'une suite de nombres générée par un algorithme n'est pas vraiment aléatoire. Si on connaît les valeurs d'initialisation et l'algorithme, on peut calculer (et donc prévoir) les termes de la suite. Néanmoins on considèrera que l'on a un bon générateur de nombres aléatoires si on ne parvient pas à distinguer la suite de nombres pseudo aléatoires produite d'une suite véritablement aléatoire. La signification précise de cette phrase demanderait tout un développement amenant à s'interroger sur la notion même de hasard. On pourra utilement consulter à ce sujet [2]. Pour l'utilisation en statistique, nous nous contenterons de dire qu'un générateur est acceptable s'il passe avec succès une batterie de tests statistiques courants.

Les fonctions `random` des principaux langages de programmation ou logiciels sont bâties sur des algorithmes arithmétiques dont le plus simple correspond au générateur congruentiel linéaire. Il s'agit de générer une suite de nombres $(X_n)_{n \geq 1}$ vérifiant une relation de récurrence

$$X_{n+1} = aX_n + c \pmod{M} \quad (1)$$

1. Si $p \neq 1/2$, la loi de U est une loi singulière à fonction de répartition continue mais n'ayant pas de densité par rapport à la mesure de Lebesgue.

et d'en déduire une suite $(U_n)_{n \geq 1}$ à valeurs dans $[0, 1[$ en prenant $U_n = X_n/M$. Par exemple la fonction `rand` de Scilab utilise (1) avec $M = 2^{31}$, $a = 843\,314\,861$ et $c = 453\,816\,693$. La suite (U_n) ainsi construite est complètement déterministe, car périodique. Cependant sa période est tellement grande qu'on peut en pratique la considérer comme une suite aléatoire, du moins pour des besoins statistique courants (son usage est déconseillé en cryptographie). Remarquons d'ailleurs que même si U_n était ici aléatoire, ses valeurs seraient de la forme $k2^{-31}$ et on obtiendrait la loi uniforme discrète sur $D_{31} = \{k2^{-31}; 0 \leq k < 2^{31}\}$ au lieu de la loi uniforme sur $[0, 1[$. Ceci n'est pas trop gênant pour les deux raisons suivantes. D'une part la loi uniforme μ_n sur $D_n = \{k2^{-n}; 0 \leq k < 2^n\}$ converge étroitement vers la loi uniforme sur $[0, 1[$ quand n tend vers l'infini² et d'autre part les nombres réels sont représentés en machine par des rationnels dyadiques de la forme $k2^{-j}$, de sorte que tous les réels de $[k2^{-j}, (k+1)2^{-j}[$ sont confondus.

Dans ce document, nous nous situons en aval du problème de la construction d'un générateur de nombres aléatoires. On suppose que l'on sait générer une suite i.i.d. (U_n) de variables aléatoires de loi uniforme sur $[0, 1]$. On se propose de construire et de justifier mathématiquement des algorithmes permettant à partir de là de simuler une variable aléatoire ou un vecteur aléatoire de loi donnée. On donnera une traduction de certains de ces algorithmes en Scilab à titre d'illustration.

2 Méthode théorique pour simuler une v.a.r.

Soit X une variable aléatoire réelle de fonction de répartition F , définie par

$$F(x) := \mathbf{P}(X \leq x). \quad (2)$$

Rappelons que F est croissante, continue à droite et limitée à gauche en tout point de \mathbb{R} , que l'ensemble de ses discontinuités est au plus dénombrable et que F a pour limite 0 en $-\infty$ et 1 en $+\infty$. Dans le cas particulier où F est continue et strictement croissante sur tout \mathbb{R} , elle réalise une bijection de \mathbb{R} sur $]0, 1[$ et admet donc un inverse $F^{-1} :]0, 1[\rightarrow \mathbb{R}$ au sens classique. Si U est une variable aléatoire de loi uniforme sur $]0, 1[$, alors $Y := F^{-1}(U)$ a même loi que X . On le vérifie facilement en calculant la fonction de répartition de Y :

$$\forall x \in \mathbb{R}, \quad \mathbf{P}(Y \leq x) = \mathbf{P}(F^{-1}(U) \leq x) = \mathbf{P}(U \leq F(x)) = F(x). \quad (3)$$

Dans cette suite d'égalités, la deuxième repose sur la croissance de F et de F^{-1} qui légitiment l'équivalence $F^{-1}(U) \leq x \Leftrightarrow U \leq F(x)$. La dernière égalité est due au calcul de la fonction de répartition de la loi uniforme sur $]0, 1[$ (qui coïncide sur $[0, 1]$ avec l'identité) et au fait que $F(x) \in [0, 1]$.

2. L'erreur commise sur la f.d.r. de μ_n en la remplaçant par la f.d.r. de la loi uniforme sur $[0, 1]$ est majorée par 2^{-n} et $2^{-31} \simeq 4,7 \times 10^{-10}$ est suffisamment petit pour un usage courant de cette approximation.

Cette propriété permet donc, à partir d'un générateur aléatoire fournissant des réalisations d'une v.a. uniforme, de simuler une v.a. de même loi que X , en admettant que l'on sache calculer F^{-1} . Dans un souci de généralisation, ceci nous conduit à poser la définition suivante.

Définition 1. Si F est la fonction de répartition d'une variable aléatoire, son inverse généralisée (ou fonction quantile) est définie par

$$\forall u \in]0, 1[, \quad F^{-1}(u) := \inf\{x \in \mathbb{R}; F(x) \geq u\}. \quad (4)$$

L'équation $F(x) = u$ peut avoir soit une solution unique, soit aucune solution, soit une infinité de solutions. La détermination graphique de $F^{-1}(u)$ dans chacune de ces configurations est illustrée figure 1.1.

Théorème 2. Soient X une variable aléatoire réelle de fonction de répartition F et U une variable aléatoire de loi uniforme sur $]0, 1[$. Alors X et $F^{-1}(U)$ ont même loi.

Démonstration. On voit facilement que la seule adaptation à apporter à la preuve du cas particulier où F est continue et strictement croissante est la justification de l'équivalence

$$\forall u \in]0, 1[, \forall x \in \mathbb{R}, \quad F^{-1}(u) \leq x \Leftrightarrow u \leq F(x), \quad (5)$$

avec F^{-1} définie par (4). Par commodité, nous noterons

$$A_u := \{t \in \mathbb{R}; F(t) \geq u\},$$

d'où $F^{-1}(u) = \inf A_u$.

Pour établir l'implication directe dans (5), supposons que $F^{-1}(u) \leq x$. Alors pour tout $n \geq 1$, $F^{-1}(u) < x + 1/n$ et comme $F^{-1}(u) = \inf A_u$, il existe un $t_n \in A_u$ tel que $F^{-1}(u) \leq t_n < x + 1/n$. Par la définition de A_u , on a $F(t_n) \geq u$ et par croissance de F on en déduit

$$u \leq F(t_n) \leq F(x + 1/n).$$

Grâce à la continuité à droite de F au point x , on en déduit³ en faisant tendre n vers l'infini que $u \leq F(x)$. Comme $x \in \mathbb{R}$ et $u \in]0, 1[$ étaient quelconques, l'implication directe de (5) est ainsi démontrée.

L'implication réciproque est immédiate. En effet si $u \leq F(x)$, cela signifie que $x \in A_u$. Or $F^{-1}(u) = \inf A_u$, donc $F^{-1}(u) \leq x$. \square

Remarque 3. Il était commode dans la démonstration de considérer la loi uniforme sur l'intervalle ouvert $]0, 1[$. Comme $\{0\}$ et $\{1\}$ sont de mesure de Lebesgue nulle, cette loi est la même que la loi uniforme sur $[0, 1[$ ou $]0, 1]$ ou $[0, 1]$. Du point de vue informatique, il y a cependant une nuance car la loi uniforme simulée par la machine est la loi uniforme discrète sur $\{k2^{-d}; 0 \leq k < 2^d\}$ ou sur $\{k2^{-d}; 0 < k \leq 2^d\}$.

Voici des exemples simples d'utilisation de F^{-1} pour la simulation.

3. En laissant tomber $F(t_n)$ dont la seule mission était d'établir l'inégalité $u \leq F(x + 1/n)$.

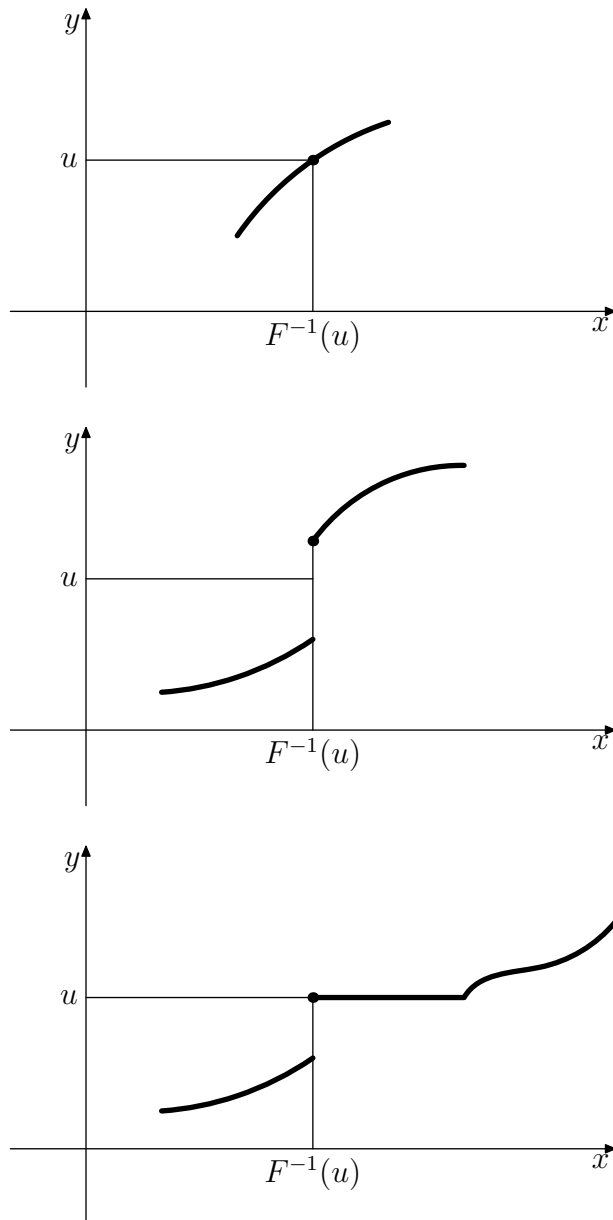


FIGURE 1.1 – Détermination graphique de l'inverse généralisé

Exemple 1 (Loi de Cauchy). Si U suit la loi uniforme sur $]0, 1[$, $Y := \tan(\pi(U - 1/2))$ suit la loi de Cauchy de densité $t \mapsto \frac{1}{\pi(1+t^2)}$.

Exemple 2 (Lois exponentielles). Si U suit la loi uniforme sur $]0, 1[$, $Y := -\frac{\ln U}{a}$ suit la loi exponentielle de paramètre a . En fait ici $F(x) = 1 - \exp(-ax)$ s'inverse en $F^{-1}(u) = -\frac{\ln(1-u)}{a}$, mais on exploite le fait que $1 - U$ a même loi que U .

Exemple 3 (Lois de Weibull). Les lois de Weibull sont très utilisées en fiabilité. La loi $\text{Weib}(a, b, c)$ de paramètres $a > 0$, $b \geq 0$ et $c > 0$ est caractérisée par sa fonction de

survie $G(x) = 1 - F(x)$ donnée par

$$G_{a,b,c}(x) = \exp\left(-\left(\frac{x-b}{c}\right)^a\right), \text{ pour } x \geq b.$$

Clairement b est un paramètre de localisation et c un paramètre d'échelle de sorte que $\text{Weib}(a, b, c)$ se déduit par translation et changement d'échelle de la loi $\text{Weib}(a) = \text{Weib}(a, 0, 1)$ de fonction de survie

$$G_a(x) = \exp(-x^a), \text{ pour } x \geq 0.$$

La simulation de la loi $\text{Weib}(a, b, c)$ se ramène ainsi à celle de $\text{Weib}(a)$. En exploitant à nouveau le fait que U et $1-U$ ont même loi, on voit immédiatement que $Y := (-\ln U)^{1/a}$ suit la loi $\text{Weib}(a)$.

3 Méthodes particulières pour lois usuelles

3.1 Lois discrètes à support fini

Soit X une variable aléatoire discrète dont l'ensemble des valeurs possibles est fini :

$$X(\Omega) = \{x_1, \dots, x_d\}.$$

Notons

$$p_k := \mathbf{P}(X = x_k), \quad s_0 := 0, \quad s_k := \sum_{i \leq k} p_i, \quad 1 \leq k \leq d.$$

Les points s_0, s_1, \dots, s_d induisent une partition de $[0, 1]$ et si U est une variable de loi uniforme sur $[0, 1]$, $\mathbf{P}(U \in [s_{k-1}, s_k]) = s_k - s_{k-1} = p_k$. On en déduit que

$$Y := \sum_{k=1}^d x_k \mathbf{1}_{[s_{k-1}, s_k]}(U) \tag{6}$$

a même loi que X . L'écriture (6) est commode pour un mathématicien, mais il serait maladroit de la programmer telle quelle, car les d multiplications $x_k \mathbf{1}_{[s_{k-1}, s_k]}(U)$ et la somme de leurs d résultats sont inutiles. En pratique, il suffit de trouver pour $U(\omega)$ donné, l'unique indice $k = k(\omega)$ tel que $s_{k-1} \leq U(\omega) < s_k$ et de décider alors que $Y(\omega) = x_k$. C'est exactement ce que fait la fonction Scilab `discr1.sci` dont voici le code.

```
function [y]=discr1(x,p)
//
// simule une variable aléatoire discrète
// d'ensemble de valeurs possibles x_1,...,x_d
// avec probabilités respectives p_1,...,p_d
// x=( x_1,...,x_d), p=(p_1,...,p_d)
```

```

//
if sum(p) ~= 1 then
    error('La somme des probabilités doit valoir 1');
end
rand('uniform');
d=length(p);
pp=[0 p(1:(d-1))];
cpp=cumsum(pp); cp=cumsum(p);
U=rand(1,1);
k=find((cpp<= U)&(U<cp));
y=x(k)
endfunction

```

Si $X(\Omega)$ est infini dénombrable, la formule (6) reste valable avec une série au lieu d'une somme finie. On ne peut évidemment pas la programmer sous cette forme! Une possibilité serait de réindexer les p_k (et donc aussi les x_k) de façon à obtenir une suite décroissante. Ensuite on testerait l'appartenance de $U(\omega)$ à $[s_{k-1}, s_k[$ en *s'arrêtant* dès que l'on obtient une réponse positive. Ainsi l'algorithme fournira une valeur $x_{k(\omega)}$ pour $Y(\omega)$ en un *temps fini*. Cette idée de commencer par les plus grosses valeurs des p_k peut d'ailleurs être utilisée pour optimiser la simulation de X dans le cas fini, particulièrement lorsque d est grand. Par exemple si $d = 1001$ et $p_{1001} = 1/2$, les autres p_k valant $1/2000$, il serait maladroit d'utiliser la fonction `discr1` telle qu'elle est programmée. Voici un deuxième code où l'on a essayé de minimiser le nombre de tests utilisés pour produire $Y(\omega)$.

```

function [y]=discr2(x,p)
//
// simule une variable aléatoire discrète
// d'ensemble de valeurs possibles x_1,...,x_d
// avec probabilités respectives p_1,...,p_d
// x=( x_1,...,x_d), p=(p_1,...,p_d)
// on optimise le nombre de tests en réarrangeant p
// par ordre décroissant et en quittant dès que la bonne valeur de
// k est trouvée
//
if sum(p) ~= 1 then
    error('La somme des probabilités doit valoir 1');
end
rand('uniform');
d=length(p);
[pr,i]=sort(p); //réarrangement de p
xr=x(i(:)); // réindexation correspondante pour x
cpr=cumsum(pr);
U=rand(1,1);

```

```

k=1;
while U>=cpr(k), k=k+1; end
y=xr(k)
endfunction

```

3.2 Lois binomiales et multinomiales

Pour simuler une variable aléatoire X de loi binomiale $\text{Bin}(n, p)$, plutôt que d'utiliser l'algorithme précédent, il est préférable de remarquer que la somme S_n de n variables de Bernoulli indépendantes et de même paramètre p suit la loi $\text{Bin}(n, p)$. Pour générer ces variables de Bernoulli, il suffit de les prendre égales à $\mathbf{1}_{\{U_i \leq p\}}$ où les U_i sont i.i.d. de loi uniforme sur $[0, 1]$. Ainsi X a même loi que

$$S_n := \sum_{k=1}^n \mathbf{1}_{\{U_k \leq p\}}.$$

On évite ainsi l'inconvénient du calcul des valeurs de la f.d.r. de X qui font intervenir des coefficients binomiaux et des puissances de p et $1 - p$.

En Scilab, un moyen commode de programmer une indicatrice est d'utiliser la fonction `bool2s` (dont le nom signifie *Boolean to string*). Elle prend en argument un booléen (vrai `%T` ou faux `%F`) et retourne la valeur 1 pour vrai et 0 pour faux. D'où le code très simple suivant :

```

function [Y] = simbin1(n,p)
//
// simule une v.a. de loi Bin(n,p)
//
rand('uniform');U=rand(1,n);
Y=sum(bool2s(U<=p))
endfunction

```

Pour les débutants en Scilab, noter que dans ce code, U est un vecteur de longueur n , il représente une réalisation $(U_1(\omega), \dots, U_n(\omega))$ du vecteur aléatoire (U_1, \dots, U_n) . De même $U \leq p$ est un vecteur booléen contenant les résultats des tests $U_1 \leq p, \dots, U_n \leq p$.

Cette méthode pour simuler une variable aléatoire de loi binomiale se généralise à la simulation d'un *vecteur aléatoire* de loi multinomiale. Rappelons que la loi multinomiale sert à modéliser le total des résultats observés pour chaque type dans une suite d'épreuves répétées indépendantes ayant chacune d types de résultats possibles. Par exemple si on lance 200 fois un dé, on obtient un vecteur de dimension 6 dont la i -ème composante est le nombre total d'apparitions de la face numéro i au cours des 200 lancers. Ce vecteur suit la loi multinomiale de paramètres 200 et $(p_1, p_2, p_3, p_4, p_5, p_6)$, où les p_i valent tous $1/6$ si le dé est équilibré. Plus formellement, le vecteur aléatoire N suit la loi multinomiale de

paramètres n et (p_1, \dots, p_d) où $n \in \mathbb{N}^*$ et les p_i sont strictement positifs et de somme 1, si pour tout d -uplet (j_1, j_2, \dots, j_d) d'entiers tels que $j_1 + j_2 + \dots + j_d = n$,

$$\mathbf{P}\{N = (j_1, j_2, \dots, j_d)\} = \frac{n!}{j_1! j_2! \dots j_d!} p_1^{j_1} p_2^{j_2} \dots p_d^{j_d}.$$

Un coup d'oeil sur cette formule devrait vous convaincre de l'intérêt d'éviter le calcul de ces probabilités. On remarque alors opportunément que le vecteur aléatoire N a *même loi* que $\sum_{k=1}^n X_k$, où les X_k sont des vecteurs aléatoires discrets de \mathbb{R}^d , indépendants et de même loi donnée par

$$\mathbf{P}\{X_k = v_i\} = p_i, \quad 1 \leq i \leq d,$$

où l'on a posé

$$v_i := (0, \dots, 0, \underset{\substack{\uparrow \\ i}}{1}, 0, \dots, 0).$$

Autrement dit, on se ramène à un modèle d'épreuves répétées indépendantes et le vecteur aléatoire X_k est un codage binaire du résultat de la k -ième épreuve. Pour simuler les X_k , on adapte de manière évidente (6) au cas d'un vecteur aléatoire discret. Notons donc $s_0 := 0$ et $s_i = p_1 + \dots + p_i$ pour $i = 1, \dots, d$. Les U_k désignant toujours des variables i.i.d. de loi uniforme sur $[0, 1]$, on voit finalement que N a même loi que

$$S_n := \sum_{k=1}^n \sum_{i=1}^d v_i \mathbf{1}_{\{s_{i-1} \leq U_k < s_i\}}.$$

Voici un code Scilab inspiré de cette formule (notez la permutation des sommations qui permet de construire le vecteur aléatoire $Y = S_n$ composante par composante, évitant l'utilisation de v_i).

```

function [Y]=simultin1(n,p)
//
// simulation d'un vecteur aléatoire de loi multinomiale de paramètres
// n (nombre d'épreuves) et p vecteur des probabilités de résultats
// élémentaires pour une épreuve.
// Retourne un vecteur colonne.
//
d=length(p);
s=[0 cumsum(p)]; // graduation de [0,1] en d intervalles de longueur p(i)
rand("uniform"); U=rand(1:n);
Y=zeros(d,1); // initialisation
for i=1:d,
Y(i,1)=sum(bool2s((s(i)<=U)&(U<s(i+1)))); // attention s(1)=0, s(d+1)=1
end
endfunction

```

3.3 Lois de Poisson

La variable aléatoire discrète X suit la loi de Poisson de paramètre α ($\alpha \in \mathbb{R}_+^*$) si

$$X(\Omega) = \mathbb{N} \quad \text{et} \quad \forall k \in \mathbb{N}, \quad \mathbf{P}(X = k) = \frac{e^{-\alpha} \alpha^k}{k!}.$$

L'algorithme de simulation que nous allons proposer pour cette loi repose sur le lemme suivant.

Lemme 4. *Soit $(E_i)_{i \geq 1}$ une suite de variables aléatoires indépendantes et de même loi exponentielle de paramètre α . Notons $S_1 := E_1$ et pour $n \geq 2$, $S_n := E_1 + \dots + E_n$. On a alors*

$$\forall n \geq 1, \quad \mathbf{P}(S_n \leq 1 < S_{n+1}) = \frac{e^{-\alpha} \alpha^n}{n!}. \quad (7)$$

Démonstration. Le vecteur aléatoire $V_{n+1} := (E_1, \dots, E_n, E_{n+1})$ a pour densité par rapport à la mesure de Lebesgue λ_{n+1} de \mathbb{R}^{n+1} :

$$f_{n+1} : (x_1, \dots, x_{n+1}) \mapsto \alpha^{n+1} \exp(-\alpha(x_1 + \dots + x_{n+1})) \mathbf{1}_{\mathbb{R}_+^{n+1}}(x_1, \dots, x_{n+1}),$$

parce que ses composantes sont indépendantes et de même densité f_1 par rapport à λ_1 . Pour exprimer $\mathbf{P}(S_n \leq 1 < S_{n+1})$ à l'aide de la loi de V_{n+1} , on introduit le borélien

$$A_{n+1} := \left\{ x \in \mathbb{R}^{n+1}; x_1 + \dots + x_n \leq 1 < x_1 + \dots + x_n + x_{n+1} \right\}$$

et on remarque que

$$\mathbf{P}(S_n \leq 1 < S_{n+1}) = \mathbf{P}(V_{n+1} \in A_{n+1}) = \int_{A_{n+1} \cap \mathbb{R}_+^{n+1}} f_{n+1} d\lambda_{n+1}.$$

Pour calculer cette intégrale, il est commode d'utiliser le changement de variable linéaire bijectif

$$\varphi : (x_1, \dots, x_{n+1}) \mapsto (s_1, \dots, s_{n+1}), \quad \text{où} \quad s_k := \sum_{i \leq k} x_i, \quad k = 1, \dots, n+1.$$

On voit immédiatement que l'application φ^{-1} est donnée par $x_1 = s_1$ et $x_k = s_k - s_{k-1}$ pour $k \geq 2$ et que son déterminant vaut 1. En notant pour alléger $A_{n+1}^+ := A_{n+1} \cap \mathbb{R}_+^{n+1}$, la formule de changement de variable pour les bijections linéaires s'écrit donc

$$\mathbf{P}(S_n \leq 1 < S_{n+1}) = \int_{\varphi(A_{n+1}^+)} \alpha^{n+1} \exp(-\alpha s_{n+1}) d\lambda_{n+1}(s_1, \dots, s_{n+1}).$$

Pour déterminer $\varphi(A_{n+1}^+)$, on note que A_{n+1}^+ est caractérisé par les inéquations :

$$\forall i = 1, \dots, n+1, \quad x_i \geq 0 \quad \text{et} \quad x_1 + \dots + x_n \leq 1 < x_1 + \dots + x_n + x_{n+1}.$$

En remplaçant x par $\varphi^{-1}(s)$ dans ces inéquations, on obtient la caractérisation de $\varphi(A_{n+1}^+)$, à savoir : $s_1 \geq 0$, $\forall i = 2, \dots, n+1$, $s_i - s_{i-1} \geq 0$ et $s_n \leq 1 < s_{n+1}$. On voit ainsi que

$$\varphi(A_{n+1}^+) = \{s \in \mathbb{R}^{n+1}; 0 \leq s_1 \leq s_2 \leq \dots \leq s_n \leq 1 < s_{n+1}\} = B_n \times]1, +\infty[,$$

où l'on a noté B_n le simplexe

$$B_n := \{s \in \mathbb{R}^n; 0 \leq s_1 \leq s_2 \leq \dots \leq s_n \leq 1\}.$$

Cette écriture de $\varphi(A_{n+1}^+)$ en produit cartésien nous permet d'appliquer le théorème de Fubini Tonelli pour obtenir

$$\begin{aligned} \mathbf{P}(S_n \leq 1 < S_{n+1}) &= \alpha^n \left\{ \int_{B_n} d\lambda_n(s_1, \dots, s_n) \right\} \left\{ \int_{]1, +\infty[} \alpha \exp(-\alpha s_{n+1}) d\lambda_1(s_{n+1}) \right\} \\ &= \alpha^n \lambda_n(B_n) \int_1^{+\infty} \alpha \exp(-\alpha t) dt \\ &= \alpha^n \lambda_n(B_n) e^{-\alpha}. \end{aligned}$$

Il ne reste plus qu'à vérifier que $\lambda_n(B_n) = 1/n!$ pour achever la preuve de (7). On peut le voir de manière géométrique en notant $C_n := [0, 1]^n$ et C'_n le sous-ensemble de C obtenu en supprimant dans C tous les points ayant au moins deux coordonnées égales. Comme $C_n \setminus C'_n$ est inclus dans une réunion finie d'hyperplans (d'équation $s_i = s_j$) tous de λ_n -mesure nulle, $\lambda_n(C_n) = \lambda_n(C'_n)$. On partitionne alors C'_n en $n!$ simplexes se déduisant de $B'_n := \{s \in \mathbb{R}^n; 0 \leq s_1 < s_2 < \dots < s_n \leq 1\}$ par une permutation de coordonnées. La mesure de Lebesgue λ_n étant invariante par permutation de coordonnées, on en déduit $\lambda_n(C'_n) = n! \lambda_n(B'_n)$, puis $1 = \lambda_n(C_n) = n! \lambda_n(B_n)$. \square

Pour compléter le lemme 4, remarquons que l'on peut étendre (7) au cas particulier $n = 0$. En posant $S_0 := 0$, (7) résulte dans ce cas de l'égalité

$$\mathbf{P}(1 < S_1) = \mathbf{P}(E_1 \in]1, +\infty[) = \int_1^{+\infty} \alpha \exp(-\alpha t) dt = e^{-\alpha}.$$

Il résulte immédiatement du lemme 4 que la variable aléatoire

$$Y := \sum_{k=1}^{+\infty} k \mathbf{1}_{\{S_k \leq 1 < S_{k+1}\}}$$

suit la loi de Poisson de paramètre α . Pour déduire de cette formule un algorithme effectif, on simule les E_i par la méthode de l'exemple 2, ce qui revient à poser $E_i = -\alpha^{-1} \ln U_i$, les U_i étant i.i.d. de loi uniforme sur $[0, 1]$. On dispose alors des équivalences

$$\begin{aligned} S_k \leq 1 < S_{k+1} &\Leftrightarrow \frac{-1}{\alpha} \sum_{i=1}^k \ln U_i \leq 1 < \frac{-1}{\alpha} \sum_{i=1}^{k+1} \ln U_i \\ &\Leftrightarrow \prod_{i=1}^{k+1} U_i < e^{-\alpha} \leq \prod_{i=1}^k U_i. \end{aligned} \quad (8)$$

L'algorithme est donc le suivant. On génère l'une après l'autre les variables U_i et on compare leur produit à $e^{-\alpha}$, en s'arrêtant lorsque (8) est vérifiée. On attribue alors à Y la valeur correspondante de k . Voici une traduction en Scilab de cet algorithme.

```
function [Y]=simpois1(alpha)
//
// simule une variable aléatoire suivant
// la loi de Poisson de paramètre alpha
//
rand('uniform');
a=exp(-alpha);
k=0; // initialisation
M=rand(1,1);
while (M>=a) do
    M=M*rand(1,1); k=k+1;
end
Y=k;
endfunction
```

Comme la suite de terme général $M_n := U_1 \dots U_n$ converge presque sûrement vers 0 (exercice!), on est sûr d'aboutir en un nombre fini de pas. De façon plus quantitative, on peut dire aussi que le nombre « moyen » de variables U_i à générer pour que l'algorithme aboutisse est égal à $1 + \alpha$ puisque $\mathbf{E}Y = \alpha$ (si l'algorithme répond $Y(\omega) = k$, c'est qu'il a calculé $k + 1$ variables U_i pour arriver à (8)). Notons N_n le nombre total de variables aléatoires U_i à générer pour simuler un échantillon de taille n de variables aléatoires indépendantes et de même loi que Y . En remarquant que $\text{Var}(1 + Y) = \text{Var} Y = \alpha$, on a par l'inégalité de Tchebycheff

$$\mathbf{P}(|N_n - n(1 + \alpha)| \geq t\sqrt{n\alpha}) \leq \frac{1}{t^2}.$$

Par exemple, la simulation de 400 v.a. de Poisson de paramètre $\alpha = 4$ « consomme » un nombre de variables uniformes compris avec une probabilité d'au moins 99% entre 1 600 et 2 400 .

3.4 Lois géométriques

Rappelons que X suit la loi géométrique de paramètre p ($p \in]0, 1[$) si

$$X(\Omega) = \mathbb{N}^* \quad \text{et} \quad \forall k \in \mathbb{N}^*, \mathbf{P}(X = k) = (1 - p)^{k-1}p.$$

C'est la loi du temps d'attente d'un premier succès dans une suite d'épreuve répétées indépendantes ayant chacune même probabilité de succès p . Par conséquent si $(U_i)_{i \geq 1}$ est une suite i.i.d. de variables suivant la loi uniforme sur $[0, 1]$, la variable aléatoire

$$Y := \min\{k \in \mathbb{N}^*; U_k < p\} \tag{9}$$

suit la loi géométrique⁴ de paramètre p . L'algorithme correspondant à (9) consiste donc à générer une v.a. U_k de loi uniforme tant que la valeur obtenue est supérieure ou égale à p et à s'arrêter à la première valeur de U_k strictement inférieure à p en retournant son indice.

```

fonction [Y]=simgeom1(p)
//
// simule une v.a. suivant la loi géométrique de paramètre p
// comme temps d'attente du premier succès dans un schéma de Bernoulli
//
rand('uniform');
k=1; // initialisation
U=rand(1,1);
while (U>=p) do
    U=rand(1,1); k=k+1;
end
Y=k;
endfunction

```

Cet algorithme s'arrête en un temps fini car $p > 0$ et $\min_{i \leq n} U_i$ converge p.s. vers 0 (exercice). Le nombre moyen de variables aléatoires uniformes utilisées est $\mathbf{E}Y = 1/p$, ce qui peut être coûteux pour les petites valeurs de p . De ce point de vue, le deuxième algorithme proposé ci-dessous, quoique moins naturel, est plus économique puisqu'il n'utilise qu'une seule variable aléatoire uniforme. Il repose sur le lemme suivant dont la vérification est immédiate.

Lemme 5. *Si E est une v.a. de loi exponentielle de paramètre a , alors*

$$\forall n \geq 1, \quad \mathbf{P}(n-1 < E \leq n) = (e^{-a})^{n-1} (1 - e^{-a}). \quad (10)$$

Comme $0 < a < +\infty$, e^{-a} et $1 - e^{-a}$ sont dans $]0, 1[$. Choissant alors a dans (10) tel que $1 - e^{-a} = p$, i.e. $a := -\ln(1 - p)$, on voit que la partie entière supérieure de E suit la loi géométrique de paramètre p .

En simulant E par la méthode de l'exemple 2, on aboutit au code Scilab suivant (ceil est la fonction partie entière supérieure) :

```

fonction [Y]=simgeom2(p)
//
// simule une v. a. suivant la loi géométrique de paramètre p
// comme partie entière supérieure d'une v.a. de loi
// exponentielle de paramètre a=-ln(1-p)
//
Y=ceil(log(rand(1,1,'uniform'))./log(1-p));
endfunction

```

4. En toute rigueur, il faudrait adopter la convention $\min \emptyset = +\infty$ et considérer Y comme variable aléatoire discrète $(\Omega, \mathcal{F}) \rightarrow \mathbb{N}^* \cup \{+\infty\}$. Pour en faire une variable discrète classique, il faut la définir sur $\Omega' := \Omega \setminus \{Y = +\infty\}$ muni de la tribu trace \mathcal{F}' de \mathcal{F} . Comme $\mathbf{P}(Y = +\infty) = \mathbf{P}(\forall k \in \mathbb{N}^*, U_k \geq p) = 0$ (exercice!), cela ne change pas la loi de Y . Voir à ce sujet la preuve de la proposition 8 a).

3.5 Lois gaussiennes

La plupart des logiciels utilisant un générateur de nombres aléatoires ont une option « gaussiennes » permettant de simuler des variables gaussiennes. Par exemple en Scilab l'instruction `rand('normal')` est un commutateur qui fait passer le générateur en mode gaussien. Après cette instruction, `rand(m,n)` génère une matrice $m \times n$ dont les termes sont considérés comme des $X_{i,j}(\omega)$, les $X_{i,j}$ étant des variables aléatoires de même loi $\mathfrak{N}(0,1)$ et indépendantes⁵. Cette fonctionnalité du générateur ne devrait pas nous dispenser de réfléchir un instant à la façon dont on peut programmer la simulation d'une variable aléatoire gaussienne à partir de la génération de variables uniformes.

En notant Φ la fonction de répartition de $\mathfrak{N}(0,1)$, qui est continue et strictement croissante sur \mathbb{R} , la méthode générale exposée à la section 2 nous propose de prendre $\Phi^{-1}(U)$. L'ennui c'est qu'on ne connaît pas d'expression analytique pour Φ ni pour Φ^{-1} et que l'inversion numérique de Φ serait assez coûteuse. Le lemme suivant fournit immédiatement un algorithme bien plus simple, connu sous le nom de méthode de Box Muller.

Lemme 6. *Si U_1 et U_2 sont indépendantes et de loi uniforme sur $[0,1]$, les variables aléatoires*

$$X := (-2 \ln U_1)^{1/2} \cos(2\pi U_2) \quad \text{et} \quad Y := (-2 \ln U_1)^{1/2} \sin(2\pi U_2)$$

sont indépendantes et de même loi $\mathfrak{N}(0,1)$.

La vérification est un simple exercice de Licence, laissé au lecteur.

4 Algorithmes de rejet

La méthode du rejet (appelée aussi d'acceptation-rejet) peut être décrite abstraitement comme suit. On suppose que l'on sait générer un vecteur aléatoire M_1 de \mathbb{R}^d suivant une certaine loi μ . On génère alors l'un après l'autre les vecteurs de la suite i.i.d. M_1, \dots, M_n, \dots en s'arrêtant au premier d'entre eux qui vérifie une certaine condition (\mathcal{H}_0). Soit T l'indice (aléatoire) correspondant. On a ainsi fabriqué un vecteur (doublement) aléatoire M_T . Comme T est aléatoire, la loi de ce vecteur n'est pas celle de M_1 , c'est une nouvelle loi ν . Si la simulation de M_1 et le test de (\mathcal{H}_0) sont facilement programmables, on dispose ainsi d'une méthode pour générer un vecteur aléatoire de loi ν . Nous allons voir comment fonctionne ce principe général d'abord pour la simulation d'un vecteur aléatoire de loi uniforme sur un borélien de \mathbb{R}^d , puis pour celle d'un vecteur aléatoire de densité connue et enfin pour simuler certaines lois discrètes.

5. Au risque d'enfoncer une porte ouverte, rappelons à ce propos que si X a pour loi $\mathfrak{N}(0,1)$, $\sigma X + m$ a pour loi $\mathfrak{N}(m,\sigma)$ (pour tous $\sigma \in \mathbb{R}_+$ et $m \in \mathbb{R}$). Ainsi pour simuler n'importe quelle loi gaussienne (en dimension 1), il suffit de savoir simuler $\mathfrak{N}(0,1)$.

4.1 Simulation de lois uniformes par rejet

Commençons par rappeler la définition de la loi uniforme sur un borélien.

Définition 7. Soient $(\Omega, \mathcal{F}, \mathbf{P})$ un espace probabilisé et B un borélien de \mathbb{R}^d tel que $0 < \lambda_d(B) < +\infty$, λ_d désignant la mesure de Lebesgue sur \mathbb{R}^d . Le vecteur aléatoire $M : \Omega \rightarrow \mathbb{R}^d$ suit la loi uniforme sur B si sa loi P_M a une densité de la forme $c\mathbf{1}_B$ par rapport à λ_d , c étant une constante (notation $X \sim \text{Unif}(B)$).

Il est clair que la seule valeur possible pour c est $1/\lambda_d(B)$. La caractérisation suivante est immédiate et bien plus utile en pratique que la définition 7.

$$M \sim \text{Unif}(B) \quad \Leftrightarrow \quad \forall A \in \text{Bor}(\mathbb{R}^d), \quad \mathbf{P}(M \in A) = \frac{\lambda_d(A \cap B)}{\lambda_d(B)}. \quad (11)$$

À partir d'une variable U de loi uniforme sur $[0, 1]$, on fabrique facilement V de loi uniforme sur $[a, b]$ en prenant $V := a + (b - a)U$. On en déduit la construction d'un vecteur aléatoire $M = (V_1, \dots, V_d)$ de loi uniforme sur le pavé $B = [a_1, b_1] \times \dots \times [a_d, b_d]$, en prenant $V_i := a_i + (b_i - a_i)U_i$, les U_i étant i.i.d. de loi uniforme sur $[0, 1]$. En effet la loi P_{V_i} de V_i a pour densité $c_i \mathbf{1}_{[a_i, b_i]}$ par rapport à λ_1 , avec $c_i = (b_i - a_i)^{-1}$. Les V_i héritent de l'indépendance des U_i (puisque $V_i = h_i(U_i)$ avec h_i mesurable), d'où $P_M = P_{V_1} \otimes \dots \otimes P_{V_d}$. Comme $\lambda_1 \otimes \dots \otimes \lambda_1 = \lambda_d$, on en déduit que P_M a pour densité par rapport à λ_d la fonction $c_1 \mathbf{1}_{[a_1, b_1]} \otimes \dots \otimes c_d \mathbf{1}_{[a_d, b_d]} = (c_1 \dots c_d) \mathbf{1}_B = \lambda_d(B)^{-1} \mathbf{1}_B$, autrement dit que P_M est la loi uniforme sur B .

En dehors de ce cas particulier où B est un pavé et de ceux qui s'y ramènent par transformation affine (par exemple loi uniforme sur un parallélogramme), la simulation d'un vecteur aléatoire de loi uniforme sur un borélien demande un peu plus de travail. L'algorithme du rejet est souvent une bonne solution à ce problème. On suppose que

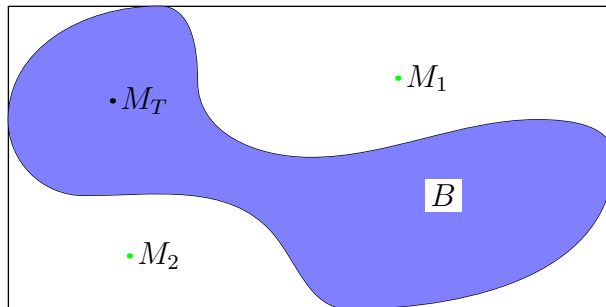


FIGURE 1.2 – Simulation par rejet de la loi uniforme sur B , ici $T(\omega) = 3$

l'on sait générer un vecteur aléatoire M_1 de loi uniforme sur un borélien C contenant B (c'est le cas notamment lorsque B est borné en prenant pour C un pavé assez grand). On génère alors séquentiellement les vecteurs i.i.d. M_1, \dots, M_n, \dots en s'arrêtant au premier d'entre eux qui vérifie la condition $M_i \in B$. Soit T l'indice (aléatoire) correspondant (T

est donc le numéro du premier point « tombé » dans B). Le vecteur aléatoire M_T suit la loi uniforme sur B .

La justification de cet algorithme repose sur le résultat suivant, qu'il est commode d'énoncer dans un cadre un peu plus général.

Proposition 8. *Soit $(M_n)_{n \in \mathbb{N}^*}$ une suite de vecteurs aléatoires $\Omega \rightarrow \mathbb{R}^d$, indépendants et de même loi μ . Soit B un borélien de \mathbb{R}^d tel que $\mu(B) > 0$. Pour tout $\omega \in \Omega$, on pose*

$$T(\omega) := \inf\{i \in \mathbb{N}^*; M_i(\omega) \in B\},$$

avec la convention $\inf \emptyset = +\infty$. On définit $M_T : \Omega \rightarrow \mathbb{R}^d$ par

$$M_T(\omega) := \begin{cases} M_{T(\omega)}(\omega) & \text{si } T(\omega) < +\infty, \\ 0 & \text{si } T(\omega) = +\infty. \end{cases}$$

Dans ces conditions,

- a) T est une variable aléatoire suivant la loi géométrique de paramètre $p = \mu(B)$.
- b) M_T est un vecteur aléatoire de loi ν donnée par

$$\forall A \in \text{Bor}(\mathbb{R}^d), \quad \nu(A) = \mathbf{P}(M_T \in A) = \frac{\mu(A \cap B)}{\mu(B)}.$$

Autrement dit, ν est la probabilité conditionnelle $\mu(\cdot \mid B)$.

En particulier lorsque μ est la loi uniforme sur un borélien C contenant B et tel que $0 < \lambda_d(B) < \lambda_d(C)$, ν est la loi uniforme sur B .

Preuve du a). Commençons par justifier la mesurabilité⁶ de T . *A priori* T est une application de Ω dans $\overline{\mathbb{N}^*} = \mathbb{N}^* \cup \{+\infty\}$. Les tribus concernées par cette mesurabilité sont \mathcal{F} et $\mathcal{P}(\overline{\mathbb{N}^*})$. En raison de la dénombrabilité de $\overline{\mathbb{N}^*}$, il suffit de vérifier que

$$\forall k \in \overline{\mathbb{N}^*}, \quad T^{-1}(\{k\}) \in \mathcal{F}. \quad (12)$$

Si $k \in \mathbb{N}^*$, on a

$$\{T = k\} = \{\forall i < k, M_i \notin B \text{ et } M_k \in B\} = \left(\bigcap_{1 \leq i < k} M_i^{-1}(B^c) \right) \cap M_k^{-1}(B), \quad (13)$$

tandis que dans le cas particulier $k = +\infty$,

$$\{T = +\infty\} = \{\forall i \in \mathbb{N}^*, M_i \notin B\} = \bigcap_{i \in \mathbb{N}^*} M_i^{-1}(B^c). \quad (14)$$

Les M_i étant des vecteurs aléatoires dans \mathbb{R}^d , donc mesurables \mathcal{F} - $\text{Bor}(\mathbb{R}^d)$, l'image réciproque par M_i d'un borélien quelconque de \mathbb{R}^d est un élément de \mathcal{F} . Il résulte alors

6. Dans un développement à l'oral de l'agrégation, il ne paraît pas opportun d'explicitier en détail les considérations de mesurabilité discutées ici. Cela ne vous exonère pas d'y avoir réfléchi au moins une fois pour pouvoir répondre à une éventuelle question, voire la susciter.

de (13) et de (14) que $T^{-1}(\{k\})$ est intersection finie ou dénombrable d'éléments de \mathcal{F} , ce qui établit (12). Ainsi T est bien une variable aléatoire discrète à valeurs dans $\overline{\mathbb{N}^*}$.

Sa loi est caractérisée par les $\mathbf{P}(T = k)$. Posons $p = \mu(B) = \mathbf{P}(M_i \in B)$. Si $k \in \mathbb{N}^*$, la décomposition (13) et l'indépendance des M_i nous donnent

$$\forall k \in \mathbb{N}^*, \quad \mathbf{P}(T = k) = (1 - p)^{k-1}p. \quad (15)$$

Calculons $\mathbf{P}(T = +\infty)$. Par hypothèse $\mu(B) > 0$, donc $1 - p = \mu(B^c) = \mathbf{P}(M_i^{-1}(B^c))$ est strictement inférieur à 1. En remarquant que pour tout $n \in \mathbb{N}^*$ on a l'inclusion $\cap_{i \in \mathbb{N}^*} M_i^{-1}(B^c) \subset \cap_{i \leq n} M_i^{-1}(B^c)$, on a par indépendance $\mathbf{P}(T = +\infty) \leq (1 - p)^n$, d'où en faisant tendre n vers l'infini,

$$\mathbf{P}(T = +\infty) = 0. \quad (16)$$

Ceci nous incite à transformer T en variable aléatoire discrète à valeurs dans \mathbb{N}^* . Posons $\Omega' := \{T < \infty\} = \{T = +\infty\}^c$. Par mesurabilité de T , Ω' appartient à \mathcal{F} et comme $\mathbf{P}(T = +\infty) = 0$, $\mathbf{P}(\Omega') = 1$. La restriction de l'ensemble d'arrivée de T à \mathbb{N}^* nous amène à restreindre son ensemble de départ à Ω' . Munissons alors Ω' de la tribu \mathcal{F}' , trace de \mathcal{F} sur Ω' , i.e. $\mathcal{F}' := \{A \cap \Omega'; A \in \mathcal{F}\}$. Comme \mathcal{F}' est incluse⁷ dans \mathcal{F} , on munit (Ω', \mathcal{F}') de la mesure restriction de \mathbf{P} qui reste une probabilité. En utilisant (13) et la définition de \mathcal{F}' , on vérifie facilement que $T : \Omega' \rightarrow \mathbb{N}^*$ est mesurable \mathcal{F}' - $\mathcal{P}(\mathbb{N}^*)$. On a donc bien transformé T en variable aléatoire discrète à valeurs dans \mathbb{N}^* et définie sur l'espace probabilisé $(\Omega', \mathcal{F}', \mathbf{P})$. Pour être complet, on peut remarquer que la loi de T n'est pas vraiment affectée par cette transformation. En effet dans la version originelle de $T : \Omega \rightarrow \overline{\mathbb{N}^*}$, la loi de T s'écrit

$$P_T = \sum_{k \in \overline{\mathbb{N}^*}} \mathbf{P}(T = k) \delta_k = \sum_{k \in \mathbb{N}^*} \mathbf{P}(T = k) \delta_k, \quad (17)$$

puisque $\mathbf{P}(T = +\infty) \delta_{+\infty}$ est la mesure nulle. Dans la version modifiée notée provisoirement $\tilde{T} : \Omega' \rightarrow \mathbb{N}^*$, la loi de \tilde{T} s'écrit aussi

$$P_{\tilde{T}} = \sum_{k \in \mathbb{N}^*} \mathbf{P}(T = k) \delta_k. \quad (18)$$

La différence entre P_T et $P_{\tilde{T}}$ est subtile. La première est une mesure sur la tribu $\mathcal{P}(\overline{\mathbb{N}^*})$, la seconde sur $\mathcal{P}(\mathbb{N}^*)$. Il résulte de (17) et (18) que $P_{\tilde{T}}$ est simplement la restriction de P_T de $\mathcal{P}(\overline{\mathbb{N}^*})$ à $\mathcal{P}(\mathbb{N}^*)$.

D'après (15), \tilde{T} suit la loi géométrique de paramètre $p = \mu(B)$. □

Preuve du b). Commençons par vérifier que M_T est un vecteur aléatoire, c'est-à-dire une application \mathcal{F} -Bor(\mathbb{R}^d) mesurable, en montrant que $M_T^{-1}(A) \in \mathcal{F}$ pour tout $A \in \text{Bor}(\mathbb{R}^d)$. En partitionnant Ω suivant les événements $\{T = k\}$, on a la décomposition :

$$M_T^{-1}(A) = \{\omega \in \Omega; M_{T(\omega)}(\omega) \in A\} = \bigcup_{k \in \overline{\mathbb{N}^*}} (\{M_T \in A\} \cap \{T = k\}).$$

7. Attention, ce n'est pas une *sous-tribu* de \mathcal{F} car \mathcal{F}' n'est pas une tribu sur Ω .

Cette union étant dénombrable, il suffit de vérifier que chacun de ses termes est un élément de \mathcal{F} . Si $k \in \mathbb{N}^*$, ceci résulte de la mesurabilité des M_i via la décomposition

$$\begin{aligned} \{M_T \in A\} \cap \{T = k\} &= \{\omega \in \Omega; T(\omega) = k \text{ et } M_k(\omega) \in A\} \\ &= \bigcap_{i < k} \{M_i \notin B\} \cap \{M_k \in B\} \cap \{M_k \in A\} \\ &= \bigcap_{i < k} M_i^{-1}(B^c) \cap M_k^{-1}(A \cap B). \end{aligned} \quad (19)$$

Dans le cas particulier $k = +\infty$,

$$\{M_T \in A\} \cap \{T = +\infty\} = \bigcap_{i \in \mathbb{N}^*} M_i^{-1}(B^c) \cap \{M_\infty \in A\}.$$

M_∞ étant le vecteur aléatoire constant 0, $\{M_\infty \in A\}$ vaut Ω ou \emptyset selon que 0 appartient ou non à A . On a donc une intersection dénombrable d'éléments de \mathcal{F} .

Maintenant que nous voilà rassurés sur la mesurabilité de M_T , on peut s'intéresser à sa loi ν que l'on détermine en calculant $\nu(A) = \mathbf{P}(M_T \in A)$ pour $A \in \text{Bor}(\mathbb{R}^d)$. En partitionnant par les $\{T = k\}$ et compte-tenu de (16), on obtient :

$$\nu(A) = \sum_{k \in \mathbb{N}^*} \mathbf{P}(M_T \in A \text{ et } T = k). \quad (20)$$

La décomposition (19) et l'indépendance des M_i nous donnent

$$\mathbf{P}(M_T \in A \text{ et } T = k) = (1 - p)^{k-1} \mu(A \cap B),$$

d'où en reportant dans (20) et en notant que $0 < 1 - p < 1$,

$$\nu(A) = \mu(A \cap B) \sum_{k \in \mathbb{N}^*} (1 - p)^{k-1} = \mu(A \cap B) \frac{1}{1 - (1 - p)} = \frac{\mu(A \cap B)}{\mu(B)}.$$

On a donc bien $\nu = \mu(\cdot \mid B)$. Dans le cas particulier où μ est la loi uniforme sur un borélien C contenant B et tel que $0 < \lambda_d(B) < \lambda_d(C)$, on obtient grâce à (11),

$$\forall A \in \text{Bor}(\mathbb{R}^d), \quad \nu(A) = \frac{\frac{\lambda_d(A \cap B)}{\lambda_d(C)}}{\frac{\lambda_d(B)}{\lambda_d(C)}} = \frac{\lambda_d(A \cap B)}{\lambda_d(B)},$$

ce qui montre que ν est la loi uniforme sur B . □

Exemple 4 (Processus de Poisson homogène sur B). Pour une édition ultérieure...

4.2 Simulation de lois à densité par rejet

La méthode du rejet permet aussi de simuler des variables ou des vecteurs aléatoires dont la loi est à densité par rapport à la mesure de Lebesgue, en s'aidant d'une autre densité dont on sait simuler la loi. Commençons par la description de l'algorithme.

On voudrait simuler un vecteur aléatoire Z dans \mathbb{R}^d , de densité f . On suppose que l'on sait simuler un vecteur aléatoire X de densité g et trouver une constante c telle que $f \leq cg$ (nécessairement $c \geq 1$, pourquoi?). Les X_i sont des vecteurs aléatoires indépendants de même loi ayant pour densité g , les U_i des variables aléatoires réelles indépendantes de même loi uniforme sur $[0, 1]$ et les suites $(X_i)_{i \geq 1}$ et $(U_i)_{i \geq 1}$ sont indépendantes. On génère la suite des $M_i := (X_i, cg(X_i)U_i)$ en s'arrêtant au premier indice i_0 (aléatoire) tel que $cg(X_{i_0})U_{i_0} \leq f(X_{i_0})$. On pose alors $Z = X_{i_0}$ et ce vecteur aléatoire Z sur \mathbb{R}^d a la loi de densité f .

La justification de cet algorithme repose sur la proposition 8 combinée avec les deux propositions suivantes dont nous différons légèrement la preuve.

Proposition 9. *Soit f une densité de probabilité sur \mathbb{R}^d et G son hypographe :*

$$G := \{(x, y) \in \mathbb{R}^d \times \mathbb{R}; 0 \leq y \leq f(x)\}.$$

Soit $M = (Z, Y)$ un vecteur aléatoire de $\mathbb{R}^d \times \mathbb{R}$ de loi uniforme sur G . Alors la loi du vecteur aléatoire Z de \mathbb{R}^d a pour densité f par rapport à λ_d .

Proposition 10. *Soit $X : \Omega \rightarrow \mathbb{R}^d$ un vecteur aléatoire dont la loi a pour densité g par rapport à λ_d . Posons*

$$M := (X, cg(X)U),$$

où U est une variable aléatoire de loi uniforme sur $[0, 1]$, indépendante de X et $c > 0$ une constante. On note H l'hypographe de cg :

$$H := \{(x, y) \in \mathbb{R}^d \times \mathbb{R}; 0 \leq y \leq cg(x)\}.$$

Alors M suit la loi uniforme sur H .

Avant de les démontrer, voyons d'abord comment on utilise les propositions 9 et 10.

Justification de l'algorithme. On conserve les notations des propositions 9 et 10. Comme $f \leq cg$, G est inclus dans H . De plus en utilisant l'identification $\lambda_{d+1} = \lambda_d \otimes \lambda_1$ et la définition d'une mesure produit, on vérifie facilement que $\lambda_{d+1}(G) = \int_{\mathbb{R}^d} f d\lambda_d = 1$, $\lambda_{d+1}(H) = \int_{\mathbb{R}^d} cg d\lambda_d = c$ et que $\lambda_{d+1}(H \setminus G) = \int_{\mathbb{R}^d} (cg - f) d\lambda_d$. La fonction $cg - f$ étant positive, cette dernière égalité montre que $\lambda_{d+1}(H \setminus G) = 0$ si et seulement si $f = cg$ λ_d presque partout. Mais dans ce cas $c = 1$ (puisque f et g sont des densités) donc $f = g$ λ_d presque partout et on sait déjà simuler un vecteur de densité g , ce qui rend la méthode du rejet inutile. On ne perd donc pas de généralité en supposant désormais que $\lambda_{d+1}(H \setminus G) > 0$. On a donc

$$0 < \lambda_{d+1}(G) < \lambda_{d+1}(H) < +\infty.$$

Par hypothèse, on sait simuler les suites indépendantes $(X_i)_{i \geq 1}$ et $(U_i)_{i \geq 1}$, la première étant i.i.d. avec les X_i de densité g et la seconde i.i.d. avec les U_i de loi uniforme sur $[0, 1]$. En raison de toutes ces indépendances, la suite des vecteurs $M_i := (X_i, cg(X_i)U_i)$ est i.i.d. et par la proposition 10, les M_i suivent la loi uniforme sur H .

Posons maintenant

$$T(\omega) := \inf\{i \in \mathbb{N}^*; M_i(\omega) \in G\}$$

et

$$M_T(\omega) := \begin{cases} M_{T(\omega)}(\omega) & \text{si } T(\omega) < +\infty, \\ 0 & \text{si } T(\omega) = +\infty. \end{cases}$$

Par la proposition 8, M_T est un vecteur aléatoire de \mathbb{R}^{d+1} , de loi uniforme sur G . La proposition 9 nous permet alors de conclure que sa projection $Z = X_T$ sur \mathbb{R}^d est un vecteur aléatoire de densité f . \square

Preuve de la proposition 9. Pour identifier la loi de Z lorsque $M = (Z, Y)$ suit la loi uniforme sur G , nous allons calculer $\mathbf{P}(Z \in B)$ pour B borélien quelconque de \mathbb{R}^d . On commence par remarquer que

$$\mathbf{P}(Z \in B) = \mathbf{P}((Z, Y) \in B \times \mathbb{R}) = \frac{\lambda_{d+1}(G \cap (B \times \mathbb{R}))}{\lambda_{d+1}(G)} = \lambda_{d+1}(G \cap (B \times \mathbb{R})). \quad (21)$$

Posons $G_B := G \cap (B \times \mathbb{R})$ et notons $(G_B)_x$ sa section à x fixé ($x \in \mathbb{R}^d$) :

$$(G_B)_x := \{y \in \mathbb{R}; (x, y) \in G_B\} = \begin{cases} [0, f(x)] & \text{si } x \in B \\ \emptyset & \text{si } x \notin B. \end{cases} \quad (22)$$

En identifiant λ_{d+1} avec $\lambda_d \otimes \lambda_1$ et en appliquant la définition d'une mesure produit, on obtient

$$\lambda_{d+1}(G_B) = (\lambda_d \otimes \lambda_1)(G_B) = \int_{\mathbb{R}^d} \lambda_1((G_B)_x) d\lambda_d(x).$$

Au vu de (22), $\lambda_1((G_B)_x) = f(x)\mathbf{1}_B(x)$, d'où $\lambda_{d+1}(G_B) = \int_B f d\lambda_d$, ce qui reporté dans (21) nous donne

$$\mathbf{P}(Z \in B) = \int_B f d\lambda_d.$$

Cette égalité étant vérifiée par tout borélien B de \mathbb{R}^d , la loi de Z a pour densité f par rapport à λ_d . \square

Preuve de la proposition 10. Pour identifier la loi de $M := (X, cg(X)U)$, nous allons calculer $\mathbf{P}(M \in A \times B)$ pour $A \in \text{Bor}(\mathbb{R}^d)$ et $B \in \text{Bor}(\mathbb{R})$ quelconques. M étant une fonction mesurable du couple (X, U) , il est commode d'exprimer cette probabilité à l'aide de la densité h de ce couple qui par indépendance de X et U , s'écrit $h(x, u) = g(x)\mathbf{1}_{[0,1]}(u)$. Ainsi

$$\begin{aligned} \mathbf{P}(M \in A \times B) &= \int_{\mathbb{R}^d \times \mathbb{R}} \mathbf{1}_{A \times B}(x, cg(x)u) g(x)\mathbf{1}_{[0,1]}(u) d(\lambda_d \otimes \lambda_1)(x, u) \\ &= \int_{\mathbb{R}^d} \mathbf{1}_A(x) \left\{ \int_{\mathbb{R}} \mathbf{1}_B(cg(x)u) g(x)\mathbf{1}_{[0,1]}(u) d\lambda_1(u) \right\} d\lambda_d(x). \end{aligned} \quad (23)$$

Notons $I(x)$ l'intégrale entre accolades. Si $g(x) = 0$, il est clair que $I(x) = 0$. Sinon⁸ $g(x) > 0$ et le changement de variable $v = cg(x)u$ à x fixé (effet d'une homothétie sur la mesure de Lebesgue) nous donne

$$\begin{aligned} I(x) &= \int_{\mathbb{R}} \mathbf{1}_B(v)g(x)\mathbf{1}_{[0,1]} \left(\frac{v}{cg(x)} \right) \frac{1}{cg(x)} d\lambda_1(v) = \frac{1}{c} \int_{\mathbb{R}} \mathbf{1}_B(v)\mathbf{1}_{[0,cg(x)]}(v) d\lambda_1(v) \\ &= \frac{1}{c} \int_{\mathbb{R}} \mathbf{1}_{B \cap [0,cg(x)]}(v) d\lambda_1(v) \\ &= \frac{1}{c} \lambda_1(B \cap [0, cg(x)]). \end{aligned} \quad (24)$$

Remarquons que si $g(x) = 0$, $B \cap [0, cg(x)]$ est soit \emptyset , soit $\{0\}$. Dans les deux cas sa mesure de Lebesgue est nulle. On peut donc retenir la formule (24) comme l'expression générale de $I(x)$, y compris lorsque $g(x) = 0$. Par report de cette expression dans (23),

$$\mathbf{P}(M \in A \times B) = \frac{1}{c} \int_{\mathbb{R}^d} \mathbf{1}_A(x) \lambda_1(B \cap [0, cg(x)]) d\lambda_d(x). \quad (25)$$

Pour comparer cette probabilité avec $\mu(A \times B)$, où μ désigne la loi uniforme sur H , faisons une partie du chemin en sens inverse en écrivant grâce à (11), à l'identification $\lambda_{d+1} = \lambda_d \otimes \lambda_1$ et à la définition d'une mesure produit :

$$\begin{aligned} \mu(A \times B) &= \frac{\lambda_{d+1}((A \times B) \cap H)}{\lambda_{d+1}(H)} = \frac{1}{c} (\lambda_d \otimes \lambda_1)((A \times B) \cap H) \\ &= \frac{1}{c} \int_{\mathbb{R}^d} \lambda_1(((A \times B) \cap H)_x) d\lambda_d(x), \end{aligned} \quad (26)$$

où $((A \times B) \cap H)_x$ est la section à x fixé de $(A \times B) \cap H$. Explicitons cette section :

$$\begin{aligned} ((A \times B) \cap H)_x &= \{y \in \mathbb{R}; (x, y) \in (A \times B) \cap H\} \\ &= \{y \in \mathbb{R}; x \in A, y \in B \cap [0, cg(x)]\} \\ &= \begin{cases} \emptyset & \text{si } x \notin A, \\ B \cap [0, cg(x)] & \text{si } x \in A. \end{cases} \end{aligned}$$

Par conséquent,

$$\lambda_1(((A \times B) \cap H)_x) = \lambda_1(B \cap [0, cg(x)]) \mathbf{1}_A(x),$$

ce qui par report dans (26) et comparaison avec (25) établit l'égalité :

$$\forall A \in \text{Bor}(\mathbb{R}^d), \forall B \in \text{Bor}(\mathbb{R}), \quad P_M(A \times B) := \mathbf{P}(M \in A \times B) = \mu(A \times B).$$

Nous avons ainsi montré que les mesures *finies* P_M et μ coïncident sur la π -classe \mathcal{R} des « rectangles mesurables » $A \times B$, donc par le théorème d'unicité des mesures elles coïncident sur la tribu engendrée $\sigma(\mathcal{R}) = \text{Bor}(\mathbb{R}^d) \otimes \text{Bor}(\mathbb{R})$, que l'on identifie avec $\text{Bor}(\mathbb{R}^{d+1})$. Ainsi $P_M = \mu$, *i.e.* la loi de M est bien la loi uniforme sur H . \square

8. On suppose implicitement que l'on a choisi une version de g telle que pour tout $x \in \mathbb{R}^d$, $g(x) \geq 0$.

Exemple 5 (Lois gamma). La loi Gamma de paramètre $a > 0$ est définie par sa densité

$$f(t) = \frac{1}{\Gamma(a)} t^{a-1} e^{-t} \mathbf{1}_{]0, +\infty[}(t).$$

Nous examinons le cas $0 < a < 1$. Le cas $a = 1$ est celui de la loi exponentielle et pour $a \geq 1$, nous renvoyons à [3] où plusieurs algorithmes sont proposés. L'intérêt de ce cas est de fournir un exemple où les densités f et g ne sont pas bornées. Pour contrôler l'explosion de f en zéro, il nous faut utiliser une densité g tendant vers $+\infty$ au moins aussi vite. Un bon candidat est la densité g de la loi de Weibull de paramètre a , qu'il est facile de simuler, cf. exemple 3. En effet en dérivant la f.d.r. $F = 1 - G_a$, on trouve

$$g(t) = at^{a-1} \exp(-t^a) \mathbf{1}_{]0, +\infty[}(t).$$

On vérifie par un simple calcul de maximum que pour tout $t > 0$,

$$\frac{f(t)}{g(t)} = \frac{\exp(t^a - t)}{a\Gamma(a)} \leq c := \frac{\exp(b(1-a))}{\Gamma(a+1)}, \quad \text{où } b = a^{a/(1-a)}.$$

Voici une implémentation en Scilab de l'algorithme du rejet pour simuler une loi Gamma de paramètre $a \in]0, 1[$. Pour des raisons de lisibilité, le code a été scindé en trois fonctions regroupées dans le même fichier.

```

function [y]=gamdns(a,t)
//
// densité de la loi Gamma(a)
//
y=exp(-t).*t.^(a-1)./gamma(a);
endfunction

function [y]=weibdns(a,t)
//
// densité de la loi de Weibull de paramètre a
//
y=a.*t.^(a-1).*exp(-t.^a);
endfunction

function [Z]=simgamm(a)
//
// simule une variable de loi Gamma(a) pour 0<a<1
// par la méthode du rejet en utilisant la densité de Weib(a)
//
if a>1 then error("erreur, le paramètre doit être <1"); end
// Calcul de la constante de rejet
b=a.^(a./(1-a)); c=exp(b.*(1-a))./gamma(1+a);
test = %T;

```

```

rand('uniform');
while test do
    U=rand();V= rand();
    X = (-log(V)).^(1/a) ; // simulation d'une v.a. de loi Weib(a)
    test =( c.*U.*weibdns(a,X) > gamdns(a,X) );
end
Z = X;
endfunction

```

4.3 Simulation d'une loi discrète par rejet

La portée de la méthode du rejet ne se limite pas aux lois à densité. On peut aussi appliquer les idées exposées ci-dessus à la simulation de lois discrètes. Pour ne pas alourdir les notations, nous nous limiterons au cas de lois discrètes où l'ensemble des valeurs possibles est inclus dans \mathbb{N} .

On se propose de simuler une variable aléatoire discrète Z dont la loi est donnée par l'ensemble $Z(\Omega) \subset \mathbb{N}$ et la fonction $f : \mathbb{N} \rightarrow [0, 1]$, $k \mapsto f(k) = p_k = \mathbf{P}(Z = k)$. On suppose pour cela que l'on sait simuler une variable aléatoire discrète X de loi donnée par $X(\Omega) \subset \mathbb{N}$ et $g : \mathbb{N} \rightarrow [0, 1]$, $k \mapsto g(k) = q_k = \mathbf{P}(X = k)$ et que l'on connaît une constante $c > 1$ telle que $f \leq cg$ (ceci suppose que $Z(\Omega) \subset X(\Omega)$).

Proposition 11. *Avec les notations ci-dessus, soit $(X_i)_{i \geq 1}$ une suite de variables aléatoires discrètes indépendantes de même loi que X et $(U_i)_{i \geq 1}$ une suite i.i.d. de variables aléatoires uniformes sur $[0, 1]$, les deux suites étant indépendantes. On note $(M_i)_{i \geq 1}$ la suite de vecteurs aléatoires*

$$M_i := (X_i, cg(X_i)U_i), \quad i \in \mathbb{N}^*$$

et on définit T par

$$T(\omega) := \inf\{i \in \mathbb{N}^*; cg(X_i)U_i \leq f(X_i)\},$$

avec la convention $\inf \emptyset = +\infty$. On définit $X_T : \Omega \rightarrow \mathbb{R}^d$ par

$$X_T(\omega) := \begin{cases} X_{T(\omega)}(\omega) & \text{si } T(\omega) < +\infty, \\ 0 & \text{si } T(\omega) = +\infty. \end{cases}$$

Alors X_T est une v.a. discrète de même loi que Z ($\mathbf{P}(X_T = k) = f(k)$ pour tout $k \in \mathbb{N}$).

Preuve. Notons $M := (X, cg(X)U)$ avec U uniforme sur $[0, 1]$ et indépendante de X et définissons M_T sur le même modèle que X_T ci-dessus. Avant de chercher la loi de M_T , intéressons nous à la loi μ de M . Pour la caractériser, il suffit de connaître $\mu(\{k\} \times [0, y])$ pour tout $k \in \mathbb{N}$ et tout $y \in \mathbb{R}_+$. Cette connaissance découle du calcul suivant où l'on suppose $g(k) > 0$, le cas $g(k) = 0$ étant trivial puisque $\mu(\{k\} \times [0, y]) \leq \mu(\{k\} \times \mathbb{R}_+) =$

$$\mathbf{P}(X = k) = g(k).$$

$$\begin{aligned} \mathbf{P}(M \in \{k\} \times [0, y]) &= \mathbf{P}(X = k \text{ et } cg(X)U \leq y) \\ &= \mathbf{P}\left(X = k \text{ et } U \leq \frac{y}{cg(k)}\right) \\ &= \mathbf{P}(X = k)\mathbf{P}\left(U \leq \frac{y}{cg(k)}\right) \quad (X \text{ et } U \text{ indépendantes}) \\ &= g(k) \min\left(\frac{y}{cg(k)}, 1\right). \end{aligned} \quad (27)$$

Considérons le borélien $B := \{(k, y) \in \mathbb{N} \times \mathbb{R}_+; 0 \leq y \leq f(k)\}$. Il est facile de voir que $\mu(B)$ n'est pas nul. En effet grâce à (27) et en rappelant que $f \leq cg$ et $c > 1$, on obtient

$$\mu(B) = \mathbf{P}(M \in B) = \sum_{k \in \mathbb{N}} g(k) \min\left(\frac{f(k)}{cg(k)}, 1\right) = \sum_{k \in \mathbb{N}} \frac{f(k)}{c} = \frac{1}{c}. \quad (28)$$

Ainsi les conditions d'application de la proposition 8 sont satisfaites⁹. On sait alors que M_T a pour loi $\mu(\cdot \mid B)$. Ceci nous permet de calculer la loi de X_T .

$$\begin{aligned} \mathbf{P}(X_T = k) &= \mathbf{P}(M_T \in \{k\} \times \mathbb{R}_+) = \mathbf{P}(M \in \{k\} \times \mathbb{R}_+ \mid M \in B) \\ &= \frac{\mathbf{P}(M \in (\{k\} \times \mathbb{R}_+) \cap B)}{\mu(B)} \\ &= c\mathbf{P}(M \in \{k\} \times [0, f(k)]) \\ &= cg(k) \min\left(\frac{f(k)}{cg(k)}, 1\right) \\ &= f(k). \end{aligned}$$

La variable aléatoire discrète X_T a donc même loi que Z . □

Exemple 6 (Lois de Zipf). Ces lois utilisées en linguistique et en sciences sociales sont construites à partir de la fonction ζ de Riemann $\zeta(a) = \sum_{k \geq 1} k^{-a}$ en posant

$$p_k = \frac{1}{\zeta(a)k^a}, \quad k \in \mathbb{N}^*.$$

Examinons d'abord le cas où $a = 2$. La valeur de $\zeta(2)$ étant bien connue, nous avons donc ici

$$p_k = f(k) = \frac{6}{\pi^2 k^2}, \quad k \in \mathbb{N}^*$$

et nous proposons de prendre

$$q_k = g(k) = \frac{1}{k(k+1)}, \quad k \in \mathbb{N}^*.$$

9. Même si cette proposition n'a été appliquée jusqu'ici qu'avec μ loi uniforme sur un borélien, elle est valide pour n'importe quelle loi de probabilité μ , revoyez la preuve si vous en doutez.

La raison de ce choix est que la loi correspondante est facile à simuler puisque c'est celle de $X := [U^{-1}]$ où U est uniforme sur $[0, 1]$: en effet

$$\mathbf{P}(X = k) = \mathbf{P}(k \leq U^{-1} < k + 1) = \mathbf{P}\left(\frac{1}{k+1} < U \leq \frac{1}{k}\right) = \frac{1}{k} - \frac{1}{k+1} = q_k.$$

La meilleure constante c possible est

$$c = \sup_{k \geq 1} \frac{p_k}{q_k} = \frac{6}{\pi^2} \sup_{k \geq 1} \frac{k+1}{k} = \frac{12}{\pi^2}.$$

Dans le cas général où le paramètre a est un réel quelconque de $]1, +\infty[$, on s'inspire de la même idée en prenant $X = [U^{-1/(a-1)}]$, voir [3, pp. 550–552] pour les détails.

Exemple 7 (Loi du nombre de coïncidences ou *matching distribution*). Soit Z le nombre de points fixes d'une permutation aléatoire sur $\{1, \dots, n\}$, choisie selon la loi uniforme sur l'ensemble de toutes les permutations de $\{1, \dots, n\}$ (voir par exemple l'exercice 1.12 *Le problème des appariements* dans [6]). On peut vérifier que la loi μ_n de Z est donnée par

$$\mathbf{P}(Z = k) = \frac{1}{k!} \sum_{j=0}^{n-k} \frac{(-1)^j}{j!}, \quad 0 \leq k \leq n.$$

On voit immédiatement que $\mathbf{P}(Z = k) \leq 1/k!$, on peut alors prendre pour loi de X la loi de Poisson de paramètre 1 et pour constante $c = e$. Le nombre d'itérations T suit la loi géométrique de paramètre $\mu_n(B_n)$ donc d'espérance $1/\mu_n(B_n)$ et d'après (28), $\mu_n(B_n) = 1/c = e^{-1}$. En moyenne on génère e variables de loi Pois(1) pour simuler Z et ce quelle que soit la valeur de n . Le coût global de l'algorithme du rejet est donc ici en $O(n)$ puisque pour chacune des variables de Poisson générée, on doit calculer $f(k)$ (addition de $n - k + 1$ termes¹⁰).

Ceci explique que l'algorithme du rejet soit concurrentiel avec l'algorithme plus simple qui consiste à générer un n -échantillon de la loi uniforme sur $[0, 1]$, à le trier par ordre croissant et à compter le nombre de valeurs dont l'indice après le tri est resté le même. Comme le coût des tris les plus performants est en $O(n \ln n)$, on voit que la méthode du rejet est plus économique, au moins pour les grandes valeurs de n .

5 Simulation de vecteurs aléatoires par transformation

Supposons que l'on sache simuler un vecteur aléatoire M de \mathbb{R}^d ayant une certaine loi $\mu = P_M$. Si h est une transformation mesurable, on a *ispro facto* un moyen de simuler un vecteur de loi $\nu = \mu \circ h^{-1}$, il suffit de prendre le vecteur $h(M)$. Bien sûr, en pratique, c'est le problème inverse qui se pose : étant donnée ν , trouver μ simulable et h telles que $\nu = \mu \circ h^{-1}$. Nous allons examiner plusieurs situations où cette méthode est pertinente, en commençant par le cas important des lois uniformes.

10. En programmant de manière économique le calcul des factorielles, *i.e.* en utilisant une seule multiplication pour passer de $j!$ à $(j+1)!$

5.1 Loi uniforme par transformation affine

5.1.1 Principe

La simulation de lois uniformes par transformation affine repose sur le résultat suivant.

Lemme 12. *Soient h une application affine bijective $\mathbb{R}^d \rightarrow \mathbb{R}^d$ et M un vecteur aléatoire de loi uniforme sur un borélien B de \mathbb{R}^d . Alors $h(M)$ suit la loi uniforme sur $h(B)$.*

Preuve. Commençons par chercher la mesure image de λ_d par h . On peut écrire $h = \varphi \circ \tau$, où τ est une translation et φ une bijection linéaire de $\mathbb{R}^d \rightarrow \mathbb{R}^d$. On sait que $\lambda_d \circ \varphi^{-1} = |\det(\varphi^{-1})|\lambda_d$. La mesure de Lebesgue étant invariante par translations, on en déduit que

$$\lambda_d \circ h^{-1} = (\lambda_d \circ \tau^{-1}) \circ \varphi^{-1} = \lambda_d \circ \varphi^{-1} = |\det(\varphi^{-1})|\lambda_d = c\lambda_d,$$

la constante c ne dépendant que de h .

Soit A' un borélien quelconque de \mathbb{R}^d . Puisque M suit la loi uniforme sur B , on a :

$$\mathbf{P}(h(M) \in A') = \mathbf{P}(M \in h^{-1}(A')) = \frac{\lambda_d(h^{-1}(A') \cap B)}{\lambda_d(B)} \quad (29)$$

Posons $B' := h(B)$. Comme h est bijective, on a $B = h^{-1}(B')$ (vérifiez!). On peut alors écrire $h^{-1}(A') \cap B = h^{-1}(A') \cap h^{-1}(B') = h^{-1}(A' \cap B')$. En reportant dans (29), on obtient

$$\mathbf{P}(h(M) \in A') = \frac{(\lambda_d \circ h^{-1})(A' \cap B')}{(\lambda_d \circ h^{-1})(B')} = \frac{c\lambda_d(A' \cap B')}{c\lambda_d(B')} = \frac{\lambda_d(A' \cap B')}{\lambda_d(B')},$$

ce qui montre que $h(M)$ suit la loi uniforme sur B' , puisque A' était quelconque. \square

5.1.2 Loi uniforme sur un parallélogramme

Pour simuler une loi uniforme sur un parallélogramme $ABCD$ ($\overrightarrow{AC} = \overrightarrow{AB} + \overrightarrow{AD}$), on part d'un vecteur $M = \overrightarrow{(U_1, U_2)}$ de loi uniforme sur $[0, 1]^2$. On détermine l'application linéaire φ par $\varphi(1, 0) = \overrightarrow{AB}$ et $\varphi(0, 1) = \overrightarrow{AD}$. On prend pour τ la translation de vecteur \overrightarrow{OA} . L'image du carré $[0, 1]^2$ par la transformation affine $h = \tau \circ \varphi$ est le parallélogramme $ABCD$ et le vecteur aléatoire $M' = h(M)$ suit la loi uniforme sur $ABCD$. On détermine les coordonnées V_1, V_2 de M' matriciellement par :

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} x_B - x_A & x_D - x_A \\ y_B - y_A & y_D - y_A \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} + \begin{bmatrix} x_A \\ y_A \end{bmatrix}.$$

5.1.3 Loi uniforme sur un triangle ou un polygone

Pour simuler un vecteur aléatoire de loi uniforme sur un triangle donné $T = ABC$, on part d'un vecteur de loi uniforme sur un triangle T_0 construit sur les vecteurs unitaires de la base canonique de \mathbb{R}^2 et on détermine une transformation affine h qui transforme T_0 en T . Notons $O = (0, 0)$, $I = (1, 0)$, $J = (0, 1)$ et $K = (1, 1)$. Soit U_1 et U_2 deux variables aléatoires indépendantes et de même loi uniforme sur $[0, 1]$. Définissons les statistiques d'ordre du vecteur (U_1, U_2) par

$$U_{2:1} = \min(U_1, U_2), \quad U_{2:2} = \max(U_1, U_2).$$

Il est bien connu (exercice de Licence) que $(U_{2:1}, U_{2:2})$ suit la loi uniforme sur le triangle OKJ et $(U_{2:2}, U_{2:1})$ la loi uniforme sur OIK . On peut donc choisir l'un de ces deux triangles pour T_0 . Si on prend OKJ , on détermine alors h en écrivant que $h(O) = A$, $h(J) = B$ et $h(K) = C$ (ceci détermine l'une des 6 applications affines transformant OKJ en ABC).

En fait pour simplifier la détermination de h (et même pour l'éviter), il est préférable de choisir pour T_0 le triangle OIJ . L'astuce est d'écrire T_0 comme l'enveloppe convexe de ses sommets :

$$T_0 = OIJ = \{M = p_1O + p_2I + p_3J; p_1, p_2, p_3 \geq 0, p_1 + p_2 + p_3 = 1\}.$$

On voit immédiatement que si $M \in T_0$ a pour coordonnées barycentriques (p_1, p_2, p_3) relativement à OIJ et si h est l'unique transformation affine telle que $h(O) = A$, $h(I) = B$ et $h(J) = C$, le point $h(M)$ a mêmes coordonnées barycentriques (p_1, p_2, p_3) , mais relativement à ABC . Il suffit donc de savoir choisir (p_1, p_2, p_3) aléatoires de telle sorte que M suive la loi uniforme sur T_0 . Pour cela on prend les *espacements* associés au vecteur (U_1, U_2) , c'est à dire les longueurs des trois segments de la subdivision de $[0, 1]$ générée par U_1 et U_2 :

$$p_1(\omega) = U_{2:1}(\omega), \quad p_2(\omega) = U_{2:2}(\omega) - U_{2:1}(\omega), \quad p_3(\omega) = 1 - U_{2:2}(\omega).$$

On laisse en exercice¹¹ la vérification du fait que la loi de

$$M = p_1O + p_2I + p_3J = (U_{2:2} - U_{2:1}, 1 - U_{2:2})$$

est bien la loi uniforme sur OIJ . Nous pouvons maintenant proposer l'algorithme suivant pour la simulation d'un vecteur aléatoire $h(M)$ de loi uniforme sur ABC .

1. Génération de U_1 et U_2 indépendants de loi uniforme sur $[0, 1]$.
2. Tri de (U_1, U_2) et calcul des espacements $U_{2:1}$, $U_{2:2} - U_{2:1}$ et $1 - U_{2:2}$.
3. Calcul de $h(M) = U_{2:1}A + (U_{2:2} - U_{2:1})B + (1 - U_{2:2})C$.

On peut immédiatement réinvestir cette technique pour la simulation d'un vecteur aléatoire M de loi uniforme sur un polygone Q du plan (pas forcément convexe). On

11. La proposition 13 ci-après donne une solution dans un cadre plus général.

commence par *triangler* Q , autrement dit à le découper en un nombre fini de triangles T_1, \dots, T_n (on a donc $Q = \cup_{1 \leq i \leq n} T_i$ et si $i \neq j$, $\lambda_2(T_i \cap T_j) = 0$). Une fois cette triangulation faite¹², on utilise l'algorithme suivant.

1. Calculer pour $i = 1, \dots, n$ les $a_i := \lambda_2(T_i)$ et $a := \lambda_2(Q) = a_1 + \dots + a_n$.
2. Générer une variable discrète X de loi sur $\{1, \dots, n\}$ donnée par $\mathbf{P}(X = k) = a_k/a$.
3. Pour $k = X(\omega)$ ainsi obtenu, générer indépendamment de X un vecteur aléatoire M de loi uniforme sur le triangle T_k .

La justification de cet algorithme est laissée en exercice et sa traduction en Scilab est un bon sujet de T.P. Un rappel peut-être pas superflu : l'aire d'un triangle ABC peut se calculer par la formule

$$\lambda_2(ABC) = \frac{1}{2} |\det(\overrightarrow{AB}, \overrightarrow{AC})|.$$

5.1.4 Loi uniforme sur un simplexe de \mathbb{R}^d

La simulation par transformation affine de la loi uniforme sur un triangle se généralise à la dimension d au cas du *simplexe*. Un simplexe de \mathbb{R}^d est l'enveloppe convexe de $d + 1$ points A_0, A_1, \dots, A_d « en position générale », *i.e.* si $d = 2$ les trois points ne sont pas alignés, si $d = 3$ les quatre points ne sont pas coplanaires et pour d quelconque A_1, \dots, A_{d+1} n'appartiennent pas à un même hyperplan de \mathbb{R}^d . Un simplexe de \mathbb{R}^3 est un *tétraèdre*. Soit donc T un simplexe de sommets A_0, \dots, A_d et T_0 le simplexe de sommets I_0, I_1, \dots, I_d , où $I_0 = O$ et pour $j = 1, \dots, d$, I_j est le point défini par $\overrightarrow{OI_j} = e_j$, j -ème vecteur de la base canonique de \mathbb{R}^d . On voit immédiatement que si (p_0, p_1, \dots, p_d) sont les coordonnées barycentriques de M dans le simplexe T_0 , ses coordonnées cartésiennes usuelles sont (p_1, \dots, p_d) .

$$T_0 = \left\{ M = \sum_{j=0}^d p_j I_j; \forall j, p_j \geq 0, \sum_{j=0}^d p_j = 1 \right\} = \left\{ (p_1, \dots, p_d) \in \mathbb{R}_+^d; \sum_{j=1}^d p_j \leq 1 \right\}.$$

D'autre part, si h est l'unique application affine telle que $H(I_j) = A_j$ pour $j = 0, 1, \dots, d$, alors $h(T_0) = T$ et les coordonnées barycentriques de $h(M)$ dans T sont les mêmes que celles de M dans T_0 . Il suffit donc de savoir choisir p_1, \dots, p_d aléatoires de telle sorte que $M = (p_1, \dots, p_d)$ suive la loi uniforme sur T_0 pour que

$$h(M) = p_0 A_0 + p_1 A_1 + \dots + p_d A_d$$

suive la loi uniforme sur $T = h(T_0)$. Bien sûr dans cette formule, p_0 est donné par $p_0 + (p_1 + \dots + p_d) = 1$.

Comme en dimension 2, nous aurons recours aux *espacements* d'un d -échantillon uniforme sur $[0, 1]$ pour choisir les p_j . Soit (U_1, \dots, U_d) un échantillon de variables aléatoires indépendantes de même loi uniforme sur $[0, 1]$. Notons $(U_{d:1}, \dots, U_{d:d})$ le vecteur des statistiques d'ordre obtenu par réarrangement croissant de l'échantillon :

$$\{U_1, \dots, U_d\} = \{U_{d:1}, \dots, U_{d:d}\} \quad \text{et} \quad U_{d:1} \leq U_{d:2} \leq \dots \leq U_{d:d}.$$

12. Il semble plus facile de la faire « à la main » que de la programmer dans le cas le plus général.

Posons $U_{d:0} = 0$ et $U_{d:d+1} = 1$. Définissons les espacements S_j associés par

$$S_j = U_{d:j+1} - U_{d:j}, \quad j = 0, \dots, d.$$

Dit autrement et sans formules, les espacements S_0, S_1, \dots, S_d sont les longueurs des $d+1$ intervalles découpés dans $[0, 1]$ par les nombres U_1, \dots, U_d , ces segments étant numérotés de gauche à droite.

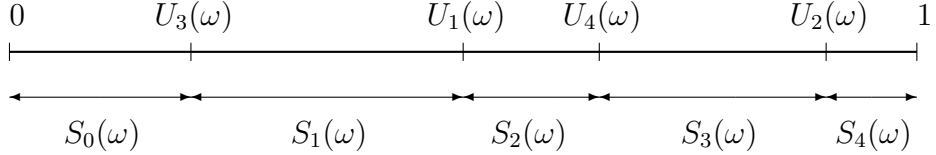


FIGURE 1.3 – Espacements d’un 4-échantillon

Proposition 13. Soit T un simplexe de \mathbb{R}^d de sommets A_0, \dots, A_d . Soit (U_1, \dots, U_d) un échantillon de variables aléatoires indépendantes de même loi uniforme sur $[0, 1]$ et $S_j, j = 0, 1, \dots, d$ les espacements associés. Alors le point aléatoire

$$\sum_{j=0}^d S_j A_j$$

suit la loi uniforme sur T .

Preuve. Compte tenu des explications données ci-dessus, la seule chose à vérifier est que le vecteur aléatoire $S = (S_1, \dots, S_d)$ suit la loi uniforme sur T_0 . Pour cela notons g une application borélienne $\mathbb{R}^d \rightarrow \mathbb{R}_+$ arbitraire et calculons de deux façons $\mathbf{E}g(S)$.

Calcul par transfert $\Omega \xrightarrow{S} \mathbb{R}^d$:

$$\mathbf{E}g(S) = \int_{\mathbb{R}^d} g(s_1, \dots, s_d) dP_S(s_1, \dots, s_d). \quad (30)$$

Calcul par transfert $\Omega \xrightarrow{(U_1, \dots, U_d)} \mathbb{R}^d$: avec des notations similaires à celles des statistiques d’ordre,

$$\mathbf{E}g(S) = \int_{[0,1]^d} g(u_{d:2} - u_{d:1}, \dots, u_{d:d+1} - u_{d:d}) d\lambda_d(u_1, \dots, u_d).$$

Découpons $[0, 1]^d$ en les $d!$ simplexes qui se déduisent par permutation de coordonnées de

$$\Delta := \{(u_1, \dots, u_d) \in [0, 1]^d; 0 \leq u_1 \leq u_2 \leq \dots \leq u_d \leq 1\},$$

Les frontières de ces simplexes sont incluses dans une réunion finie d'hyperplans d'équation $x_i = x_j$, donc de mesure nulle. En utilisant l'invariance de la mesure de Lebesgue par permutation sur les coordonnées, on obtient ainsi

$$\mathbf{E}g(S) = d! \int_{\Delta} g(u_2 - u_1, u_3 - u_2, \dots, u_d - u_{d-1}, 1 - u_d) d\lambda_d(u_1, \dots, u_d).$$

Effectuons maintenant le changement de variable $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^d, (u_1, \dots, u_d) \mapsto (t_1, \dots, t_d)$ donné par

$$t_1 = u_2 - u_1, t_2 = u_3 - u_2, \dots, t_{d-1} = u_d - u_{d-1}, t_d = 1 - u_d.$$

On vérifie facilement que φ est une bijection affine d'inverse donné par les égalités

$$u_j = 1 - \sum_{i=j}^d t_i, \quad j = 1, \dots, d.$$

Pour déterminer $\varphi(\Delta)$, on remarque en remplaçant les u_j par leur expression en fonction des t_i dans les inégalités définissant Δ , que $(t_1, \dots, t_d) \in \varphi(\Delta)$ si et seulement si

$$0 \leq 1 - (t_1 + \dots + t_d) \leq 1 - (t_2 + \dots + t_d) \leq \dots \leq 1 - t_d \leq 1.$$

Ceci équivaut à

$$t_1 + \dots + t_d \leq 1 \quad \text{et} \quad \forall j = 1, \dots, d, \quad t_j \geq 0.$$

Ainsi $\varphi(\Delta) = T_0$. D'autre part il est clair que le déterminant de l'application linéaire associée à la transformation affine φ vaut 1 en valeur absolue. Le changement de variable dans l'intégrale \int_{Δ} s'écrit donc finalement

$$\mathbf{E}g(S) = d! \int_{T_0} g(t_1, \dots, t_d) d\lambda_d(t_1, \dots, t_d) = \int_{\mathbb{R}^d} gd! \mathbf{1}_{T_0} d\lambda_d \quad (31)$$

Comme g est une fonction borélienne positive arbitraire, la comparaison avec (30) montre que la loi de S est la mesure de densité $d! \mathbf{1}_{T_0}$ par rapport à λ_d , c'est bien la loi uniforme sur T_0 . \square

5.1.5 Loi uniforme sur un ellipsoïde

Un ellipsoïde E de \mathbb{R}^d peut toujours être vu comme image par une transformation affine bijective h de la boule unité euclidienne

$$B_d := \{(x_1, \dots, x_d) \in \mathbb{R}^d; x_1^2 + \dots + x_d^2 \leq 1\}.$$

Supposons résolu le problème de la simulation de la loi uniforme sur B_d (on sait déjà le faire au moins par rejet à partir de la loi uniforme sur l'hypercube $[-1, 1]^d$, même si cette méthode n'est pas très performante en dimension élevée, cf. la section 6). La

simulation de la loi uniforme sur E se réduit alors à la détermination de h affine telle que $h(B_d) = E$.

Supposons d'abord que l'on connaisse le centre C de E et ses demi-axes $\overrightarrow{CA_1}, \dots, \overrightarrow{CA_d}$ (vecteurs orthogonaux). Il suffit de générer un vecteur colonne X de loi uniforme sur B_d et de calculer matriciellement $Y = C + HX$, où H est la matrice ayant pour colonnes les vecteurs $\overrightarrow{CA_i}$ écrits dans la base canonique de \mathbb{R}^d . Le vecteur colonne Y ainsi obtenu suit la loi uniforme sur E .

L'ellipsoïde E peut aussi être donné par son inéquation cartésienne qui après réduction peut s'écrire sous la forme matricielle

$$E = \{Y \in \mathbb{R}^d; (Y - C)'Q(Y - C) \leq 1\},$$

où Y et C sont des vecteurs colonnes, C désignant les coordonnées du centre de E et Q est une matrice symétrique définie positive. On cherche alors une matrice régulière H ($\det H \neq 0$) telle que l'application $X \mapsto C + HX$ transforme B_d en E . Ceci se produit si l'inéquation $(Y - C)'Q(Y - C) \leq 1$ est équivalente à $X'X \leq 1$ (inéquation de B_d). Or

$$(Y - C)'Q(Y - C) = X'H'QH X,$$

il suffit donc de trouver H telle que $H'QH = I$, matrice identité. Ceci revient à trouver H telle que $Q = H'^{-1}H^{-1}$ ou encore $Q = R'R$, en posant $R = H^{-1}$. L'algorithme de Cholesky [5, p. 95] permet de trouver une matrice triangulaire supérieure R ayant cette propriété. Il est implémenté en Scilab (fonction `chol`).

5.2 Vecteur gaussien de covariance donnée

Nous savons déjà simuler une suite de variables aléatoires gaussiennes indépendantes (cf. lemme 6). Ceci résout le problème de la simulation d'un vecteur aléatoire gaussien à composantes indépendantes (donc à matrice de covariance diagonale). Il nous reste à étudier le cas général du vecteur gaussien ayant une matrice de covariance Q quelconque, c'est-à-dire dans ce contexte, symétrique semi-définie positive. La méthode utilisée repose sur le résultat suivant (voir par exemple [4, Th. 16.2, p. 135]).

Proposition 14. *Soit X un vecteur (colonne) gaussien de \mathbb{R}^d , de vecteur moyenne μ . Il existe des variables aléatoires réelles indépendantes Y_1, \dots, Y_d , chaque Y_j ayant une loi $\mathfrak{N}(0, \sigma_j)$ avec $\sigma_j \geq 0$ et une matrice $d \times d$ orthogonale A telles que $X = \mu + AY$, en notant $Y = (Y_1, \dots, Y_d)'$.*

Notons que certains des σ_j peuvent être nuls et rappelons que si $\sigma = 0$, la loi gaussienne $\mathfrak{N}(m, \sigma)$ est la masse de Dirac au point m , autrement dit, la loi de la v.a. constante m . La démonstration de la proposition 14 va nous indiquer l'algorithme pour simuler X .

Preuve. La matrice de covariance Q étant symétrique semi-définie positive, on sait qu'on peut l'écrire

$$Q = ASA', \tag{32}$$

où S est une matrice diagonale à termes diagonaux σ_j positifs et A une matrice orthogonale (donc vérifiant $A' = A^{-1}$). On pose alors $Y = A'(X - \mu)$. Toute combinaison linéaire des composantes de Y peut s'écrire comme une combinaison linéaire des composantes de X plus une constante, donc est une v.a. gaussienne, puisque X est un vecteur gaussien par hypothèse. Ainsi Y est gaussien. De plus comme $A' = A^{-1}$, $X = \mu + AY$. Il nous reste à vérifier que la matrice de covariance de Y est S (ce qui entraîne l'indépendance des Y_j). Ceci résulte du calcul suivant :

$$\mathbf{E}(YY') = \mathbf{E}(A'(X - \mu)(X - \mu)'A) = A'\mathbf{E}((X - \mu)(X - \mu)')A \quad (33)$$

$$\begin{aligned} &= A'QA \\ &= A^{-1}(ASA^{-1})A \\ &= S. \end{aligned} \quad (34)$$

L'égalité (33) repose sur la propriété suivante. Soit Z une matrice aléatoire $d \times d$ dont toutes les composantes ont une espérance et B une matrice $d \times d$ non aléatoire. Alors $\mathbf{E}(BZ) = B\mathbf{E}Z$ et $\mathbf{E}(ZB) = (\mathbf{E}Z)B$. La vérification est immédiate en écrivant le produit des matrices et en utilisant la linéarité de l'espérance. L'égalité (34) exploite (32) et l'égalité entre la transposée de A et son inverse, due à l'orthogonalité de A . \square

Ce qui précède suggère l'algorithme suivant pour simuler le vecteur gaussien X de vecteur moyenne μ et de matrice de covariance Q .

1. Calculer la décomposition (32) pour Q .
2. Générer le vecteur colonne Y en simulant des gaussiennes $\mathfrak{N}(0, \sigma_j)$ pour les indices j tels que $\sigma_j > 0$ et compléter les autres composantes de Y par des zéros.
3. Calculer $X = \mu + AY$.

Pour réaliser pratiquement la première étape, Scilab nous fournit la fonction `svd` (*singular value decomposition*). Voici un extrait de la documentation en ligne (les noms des paramètres ont été modifiés pour éviter la confusion avec les notations ci-dessus) :

```
[A,S,B]=svd(Q)
DESCRIPTION
produces a diagonal matrix S , of the same dimension as Q and with
nonnegative diagonal elements in decreasing order, and unitary matrices
A and B so that X = A*S*B'.
```

En appliquant `svd` à une matrice Q symétrique semi-définie positive, on obtient $A = B$ matrice orthogonale et la diagonale de S est constituée de réels positifs rangés par ordre décroissant. Si d_0 est l'indice du dernier d'entre eux non nul, on génère les Y_1, \dots, Y_{d_0} indépendantes de lois $\mathfrak{N}(0, \sigma_j)$ et on prend $Y = (Y_1, \dots, Y_{d_0}, 0, \dots, 0)'$ (si $d_0 < d$).

6 Méthode polaire

Dans toute cette section, $\|x\|$ désigne la norme euclidienne :

$$\|x\| = (x_1^2 + \dots + x_d^2)^{1/2}$$

et on note respectivement C_d et B_d la sphère et la boule unité euclidiennes¹³ de \mathbb{R}^d :

$$C_d := \{x \in \mathbb{R}^d; \|x\| = 1\}, \quad B_d := \{x \in \mathbb{R}^d; \|x\| \leq 1\}.$$

Nous nous proposons d'obtenir des algorithmes de simulation de la loi uniforme sur C_d (cf. def. 17) et sur B_d . On sait déjà simuler la loi uniforme sur B_d en appliquant l'algorithme du rejet à partir de l'hypercube $[-1, 1]^d$. Cette méthode est acceptable pour les petites valeurs de d . Malheureusement, son coût explose plus qu'exponentiellement avec la dimension d . En effet le volume de B_d est (cf. proposition 21)

$$V_d := \lambda_d(B_d) = \frac{\pi^{d/2}}{\Gamma(d/2 + 1)}. \quad (35)$$

Or le coût de l'algorithme du rejet est proportionnel à $\lambda_d([-1, 1]^d)/V_d$ qui est d'après la formule de Stirling, équivalent à $\sqrt{2\pi} \left(\frac{2}{\pi e}\right)^{d/2} d^{(d+1)/2}$. Pour se faire une idée plus précise, regardons l'espérance du nombre d'itérations utilisées pour générer un vecteur de loi uniforme sur B_d par la méthode du rejet pour les « petites » valeurs de d . On déduit facilement de (35) les formules

$$\forall k \in \mathbb{N}^*, \quad V_{2k} = \frac{\pi^k}{k!}, \quad V_{2k+1} = \frac{2(4\pi)^k k!}{(2k+1)!}.$$

Ceci nous permet d'obtenir le tableau suivant (cf. [3, p. 232]).

d	Espérance nombre d'itérations	d	Espérance nombre d'itérations
2	$\frac{4}{\pi} \simeq 1,27$	7	$\frac{840}{\pi^3} \simeq 27,1$
3	$\frac{6}{\pi} \simeq 1,91$	8	$\frac{6\,144}{\pi^4} \simeq 63,1$
4	$\frac{32}{\pi^2} \simeq 3,24$	9	$\frac{15\,120}{\pi^4} \simeq 155,2$
5	$\frac{60}{\pi^2} \simeq 6,06$	10	$\frac{122\,880}{\pi^5} \simeq 401,5$
6	$\frac{384}{\pi^3} \simeq 12,3$	11	$\frac{332\,640}{\pi^5} \simeq 1\,087$

La méthode polaire que nous allons étudier a un coût qui croît linéairement avec d . En pratique elle l'emporte largement sur la méthode du rejet dès que $d > 5$.

13. Un géomètre noterait plutôt S^{d-1} au lieu de C_d . Notre choix de notation suit celui de Devroye [3]. Notons que dans cet ouvrage, C_d désigne aussi bien la sphère que la boule : la loi uniforme sur la sphère C_d y est appelée *uniform distribution on C_d* , tandis que la loi uniforme sur B_d est désignée par *uniform distribution in C_d* .

Pour atteindre notre objectif, on élargit le débat en discutant de la simulation de lois invariantes par les isométries euclidiennes (linéaires) de \mathbb{R}^d . Un vecteur aléatoire X suit une loi de ce type si AX a même loi que X pour toute matrice orthogonale A . Une référence pour ce sujet est Devroye [3, pp. 225–245], où ces lois sont appelées *radially symmetric*. Par commodité, nous adopterons le même vocabulaire en parlant de lois à symétrie radiale ou radialement symétriques. On dira que la loi de X est strictement à symétrie radiale si de plus, $\mathbf{P}(X = 0) = 0$.

Voici un exemple important de loi radialement symétrique, autre que la loi uniforme sur C_d que nous verrons un peu plus tard.

Proposition 15. *Si N_1, \dots, N_d sont i.i.d. $\mathfrak{N}(0, 1)$, la loi du vecteur aléatoire (N_1, \dots, N_d) est strictement à symétrie radiale.*

Preuve. Notons $N = (N_1, \dots, N_d)'$ (vecteur colonne). Clairement N est gaussien centré et sa matrice de covariance $N'N$ est la matrice identité. Soit A une matrice orthogonale quelconque. Alors AN est encore gaussien centré. Pour identifier sa loi, il suffit donc de calculer sa covariance. En notant que $A' = A^{-1}$, on obtient

$$(AN)'(AN) = N'A'AN = N'N,$$

ce qui montre que les deux vecteurs aléatoires gaussiens centrés N et AN ont même covariance, donc même loi. Comme A était arbitraire, N est bien à symétrie radiale (stricte puisque $\mathbf{P}(N = 0) = 0$, N ayant une densité par rapport à λ_d). \square

Notons $\mathbb{R}_*^d := \mathbb{R}^d \setminus \{0\}$. Le changement de variable

$$p : \mathbb{R}_*^d \longrightarrow \mathbb{R}_*^+ \times C_d, \quad x \longmapsto \left(\|x\|, \frac{x}{\|x\|} \right)$$

s'impose naturellement comme outil dans l'étude des lois strictement à symétrie radiale. Nous noterons q la deuxième application partielle de p :

$$q : \mathbb{R}_*^d \longrightarrow C_d, \quad x \longmapsto \frac{x}{\|x\|}.$$

Il est clair que p est une bijection continue¹⁴ de \mathbb{R}_*^d sur $\mathbb{R}_*^+ \times C_d$ et que son inverse s'écrit :

$$p^{-1} : \mathbb{R}_*^+ \times C_d \longrightarrow \mathbb{R}_*^d, \quad (r, s) \longmapsto rs.$$

L'application p fait penser au passage en coordonnées polaires $x \mapsto (r, \theta)$ en dimension 2. Remarquons cependant qu'avec cette notation, $p(x) = (r, (\cos \theta, \sin \theta))$. Le maniement de p est plus commode que celui des coordonnées polaires en dimension 2. En effet il n'est pas besoin de faire une coupure dans \mathbb{R}_*^2 pour que p soit continue. Notre premier travail sera de déterminer la mesure image de λ_d par p de façon à pouvoir faire le changement de variable p dans des intégrales.

14. On munit C_d de la topologie trace de \mathbb{R}^d et $\mathbb{R}_*^+ \times C_d$ de la topologie produit.

Proposition 16. Avec les notations ci-dessus, soit μ_d la mesure image de λ_d par p , définie sur la tribu produit $\text{Bor}(\mathbb{R}_*^+) \otimes \text{Bor}(C_d)$. Alors

$$\mu_d = \lambda_d \circ p^{-1} = \rho_d \otimes \nu_d,$$

où ρ_d est la mesure à densité $r \mapsto dV_d r^{d-1} \mathbf{1}_{\mathbb{R}_*^+}(r)$ par rapport à λ_1 et ν_d est la mesure de probabilité définie sur $\text{Bor}(C_d)$ par

$$\forall E \in \text{Bor}(C_d), \quad \nu_d(E) = \frac{1}{V_d} \lambda_d(p^{-1}([0, 1] \times E)) = \frac{1}{V_d} \lambda_d(\{x \in \mathbb{R}_*^d; \|x\| \leq 1, q(x) \in E\}).$$

La mesure ν_d est strictement à symétrie radiale.

Pour l'instant nous n'avons pas besoin de connaître la valeur explicite de $V_d = \lambda_d(B_d)$ donnée par (35). Nous la calculerons ultérieurement.

Définition 17. On appelle loi uniforme sur C_d la mesure ν_d définie ci-dessus.

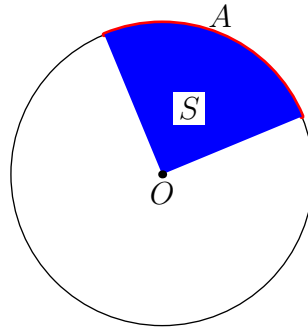


FIGURE 1.4 – Mesure de l'arc A par $\nu_2(A) = \lambda_2(S)/V_2 = \lambda_2(S)/\pi$

Pour donner une description imagée de la mesure ν_d , on peut dire qu'elle mesure le borélien E de C_d par le volume (normalisé par V_d) du « cône » de sommet O et de base E découpé dans la boule unité B_d . Par exemple en dimension 2, si on prend pour E un arc de cercle A , il sera mesuré par l'aire normalisée du secteur circulaire S associé (fig. 1.4).

Avec un léger abus, on peut considérer ν_d comme la loi d'un vecteur aléatoire X de \mathbb{R}^d tel que $\mathbf{P}(X \in C_d) = 1$. En toute rigueur il faudrait alors remplacer ν_d par $(\delta_1 \otimes \nu_d) \circ p$.

Preuve de la proposition 16. La mesure μ_d étant σ -finie, elle sera déterminée de manière unique par ses valeurs sur la famille des boréliens de la forme $]0, r] \times E$, $r \in \mathbb{R}_*^+$, $E \in \text{Bor}(C_d)$, laquelle est un π -système engendrant $\text{Bor}(\mathbb{R}_*^+) \otimes \text{Bor}(C_d)$. En appliquant la définition de μ_d , on a

$$\mu_d(]0, r] \times E) = \lambda_d(p^{-1}(]0, r] \times E)).$$

En remarquant que $p^{-1}(]0, r] \times E)$ est l'image de $p^{-1}(]0, 1] \times E)$ par l'homothétie de centre 0 et de rapport r , on en déduit

$$\mu_d(]0, r] \times E) = r^d \lambda_d(p^{-1}(]0, 1] \times E)) = V_d r^d \nu_d(E) = \rho_d(]0, r]) \nu_d(E) = (\rho_d \otimes \nu_d)(]0, r] \times E).$$

Comme $r > 0$ et $E \in \text{Bor}(C_d)$ étaient arbitraires, l'égalité de mesures $\mu_d = \rho_d \otimes \nu_d$ est établie.

La mesure ν_d , définie sur $\text{Bor}(C_d)$ est bien une probabilité puisque

$$\nu_d(C_d) = \frac{1}{V_d} \lambda_d(p^{-1}(]0, 1] \times C_d)) = \frac{\lambda_d(B_d)}{V_d} = 1.$$

La symétrie radiale de ν_d signifie son invariance par (la restriction à C_d de) toute transformation orthogonale φ de \mathbb{R}^d . Pour vérifier cette propriété, calculons $\nu_d \circ \varphi^{-1}(E)$ pour E borélien quelconque de C_d . Pour cela on remarque que

$$p^{-1}(]0, 1] \times \varphi^{-1}(E)) = \{rt; r \in]0, 1], t = \varphi^{-1}(s), s \in E\} \quad (36)$$

$$= \{r\varphi^{-1}(s); r \in]0, 1], s \in E\}$$

$$= \{\varphi^{-1}(rs); r \in]0, 1], s \in E\} \quad (37)$$

$$= \varphi^{-1}(\{rs; r \in]0, 1], s \in E\}) \quad (38)$$

$$= \varphi^{-1}(p^{-1}(]0, 1] \times E)) \quad (39)$$

Justifications.

(36) : comme φ est bijective, l'inverse ensembliste $\varphi^{-1}(E)$ est l'image directe de E par l'inverse ponctuel de φ , autrement dit $\varphi^{-1}(E) = \{\varphi^{-1}(s); s \in E\}$

(37) : l'application ponctuelle φ^{-1} est linéaire (ici r est un scalaire, s un vecteur).

(38) : même justification qu'en (36).

Pour finir, on rappelle que $\lambda_d \circ \varphi^{-1} = \lambda_d$ (invariance de la mesure de Lebesgue par toute isométrie euclidienne). Cette propriété combinée avec (39) nous donne

$$\begin{aligned} \nu_d \circ \varphi^{-1}(E) &= \lambda_d(p^{-1}(]0, 1] \times \varphi^{-1}(E))) \\ &= \lambda_d(\varphi^{-1}(p^{-1}(]0, 1] \times E))) \\ &= \lambda_d(p^{-1}(]0, 1] \times E)) = \nu_d(E). \end{aligned}$$

Comme le borélien E de C_d était quelconque, on a bien $\nu_d \circ \varphi^{-1} = \nu_d$. Cette égalité étant valide pour toute transformation orthogonale φ , la loi ν_d est à symétrie radiale. \square

Théorème 18. *Soit X un vecteur aléatoire de \mathbb{R}^d , strictement à symétrie radiale et ayant une densité f par rapport à λ_d . Alors*

a) *f est de la forme $f(x) = g(\|x\|)$, où g est une fonction mesurable positive telle que $\int_{\mathbb{R}_+^*} g(r) dV_d r^{d-1} d\lambda_1(r) = 1$;*

b) *la variable aléatoire $R := \|X\|$ et le vecteur aléatoire $S := \frac{X}{\|X\|}$ sont indépendants, R a pour densité $r \mapsto g(r) dV_d r^{d-1} \mathbf{1}_{\mathbb{R}_+^*}(r)$ par rapport à λ_1 et S suit la loi uniforme ν_d sur C_d .*

Preuve du a). On suppose d'abord que f est *continue*. Soit φ une transformation orthogonale. Le vecteur aléatoire $\varphi(X)$ admet alors pour densité $f \circ \varphi^{-1} |\det(\varphi^{-1})| = f \circ \varphi^{-1}$ (φ étant orthogonale, $\det(\varphi) = 1$). Comme X et $\varphi(X)$ ont même loi, on en déduit que $f = f \circ \varphi^{-1} \lambda_d$ presque-partout sur \mathbb{R}^d . Or ces deux applications sont continues (φ^{-1} est linéaire), donc elles sont égales *partout*. Ainsi $f = f \circ \varphi^{-1}$ partout, ce qui équivaut à $f \circ \varphi = f$ partout. Notons $e_1 = (1, 0, \dots, 0)$ le premier vecteur de la base canonique de \mathbb{R}^d . Soit x fixé dans \mathbb{R}_*^d et φ une transformation orthogonale telle que $\varphi\left(\frac{x}{\|x\|}\right) = e_1$. Alors on peut écrire

$$f(x) = f(\varphi(x)) = f\left(\|x\| \varphi\left(\frac{x}{\|x\|}\right)\right) = f(\|x\| e_1).$$

Ceci étant valable pour tout $x \neq 0$ (en utilisant à chaque fois une application φ dépendante de x), on vient d'établir que :

$$\forall x \in \mathbb{R}_*^d, \quad f(x) = f(\|x\| e_1) =: g(\|x\|). \quad (40)$$

Sans l'hypothèse additionnelle de continuité de f , la preuve est plus délicate. En effet on dispose alors seulement de l'égalité $f \circ \varphi = f \lambda_d$ presque-partout, cette égalité pouvant être fautive sur un ensemble de mesure nulle *dépendant de φ* . Le choix de φ dépendant de x comme ci-dessus devient alors illégitime. Une façon de résoudre le problème est de régulariser f par convolution.

Notons $Y_n := n^{-1/2}(N_1, \dots, N_d)$, où les N_i sont i.i.d. $\mathfrak{N}(0, 1)$ et indépendantes de X . Alors Y_n est à symétrie radiale par la proposition 15. Notons γ_n la densité du vecteur gaussien Y_n . Par indépendance, le vecteur $X + Y_n$ a pour densité $f * \gamma_n$ et cette densité hérite de la continuité de γ_n (c'est la régularisation par convolution). D'autre part $X + Y_n$ est aussi à symétrie radiale car si φ est une transformation orthogonale quelconque, $\varphi(X + Y_n) = \varphi(X) + \varphi(Y_n)$ a même loi que $X + Y_n$; en effet par indépendance de X et Y_n et symétrie radiale de leur loi respective,

$$P_{(\varphi(X), \varphi(Y_n))} = P_{\varphi(X)} \otimes P_{\varphi(Y_n)} = P_X \otimes P_{Y_n} = P_{(X, Y_n)}.$$

En utilisant le cas déjà traité où la densité d'une loi radialement symétrique admet une version continue, on dispose d'une fonction mesurable positive g_n telle que pour tout $x \in \mathbb{R}^d$, $f * \gamma_n(x) = g_n(\|x\|)$. Enfin, $f * \gamma_n$ converge dans $L^1(\lambda_d)$ vers f (c'est un exercice classique!). Par conséquent il existe une sous-suite $(f * \gamma_{n_k})_{k \geq 1}$ qui converge λ_d presque-partout sur \mathbb{R}^d vers f . On a donc

$$f(x) = \lim_{k \rightarrow +\infty} g_{n_k}(\|x\|), \quad \lambda_d - \text{p.p.},$$

ce qui entraîne l'existence d'une fonction mesurable positive g telle que $f(x) = g(\|x\|)$ λ_d presque-partout sur \mathbb{R}^d .

Il nous reste à vérifier que $\int_{\mathbb{R}_+^d} g(r) dV_d r^{d-1} d\lambda_1(r) = 1$. Pour cela on part de l'égalité $\int_{\mathbb{R}^d} f d\lambda_d = 1$ et on utilise le transfert $(\mathbb{R}^d, \lambda_d) \longrightarrow (\mathbb{R}_*^+ \times C_d, \rho_d \otimes \nu_d)$ et le théorème de

Fubini-Tonelli pour obtenir :

$$\begin{aligned}
1 &= \int_{\mathbb{R}^d} g(\|x\|) d\lambda_d(x) = \int_{\mathbb{R}_*^+ \times C_d} g(r) d(\rho_d \otimes \nu_d)(r, s) \\
&= \int_{\mathbb{R}_*^+} g(r) \left\{ \int_{C_d} d\nu_d(s) \right\} d\rho_d(r) \\
&= \int_{\mathbb{R}_*^+} g(r) d\rho_d(r) = \int_{\mathbb{R}_*^+} g(r) dV_d r^{d-1} d\lambda_1(r).
\end{aligned}$$

□

Preuve du b). Calculons de deux façons $\mathbf{E}h(R, S)$ où h est une fonction borélienne positive quelconque $h : \mathbb{R}_*^+ \times C_d \rightarrow \mathbb{R}^+$.

Calcul par transfert $\Omega \xrightarrow{(R,S)} \mathbb{R}_*^+ \times C_d$:

$$\mathbf{E}h(R, S) = \int_{\mathbb{R}_*^+ \times C_d} h(r, s) dP_{(R,S)}(r, s). \quad (41)$$

Calcul par transfert $\Omega \xrightarrow{X} \mathbb{R}^d$:

$$\mathbf{E}h(R, S) = \mathbf{E}h(p(X)) = \int_{\mathbb{R}^d} h\left(\|x\|, \frac{x}{\|x\|}\right) g(\|x\|) d\lambda_d(x).$$

En effectuant dans cette intégrale le transfert $\mathbb{R}_*^d \xrightarrow{p} \mathbb{R}_*^+ \times C_d$ on obtient :

$$\mathbf{E}h(R, S) = \int_{\mathbb{R}_*^+ \times C_d} h(r, s) g(r) d(\rho_d \otimes \nu_d)(r, s).$$

En notant τ la mesure de densité g par rapport à ρ_d , ceci peut se réécrire

$$\mathbf{E}h(R, S) = \int_{\mathbb{R}_*^+ \times C_d} h(r, s) d(\tau \otimes \nu_d)(r, s). \quad (42)$$

Comme la fonction borélienne positive h est arbitraire, la comparaison de (41) et (42) nous montre que $P_{(R,S)} = \tau \otimes \nu_d$. Les mesures τ et ν_d étant des probabilités sont les lois de R et de S respectivement (lois marginales du couple (R, S)). De plus R et S sont indépendantes puisque $P_{(R,S)} = P_R \otimes P_S$. □

Corollaire 19. Soient N_1, \dots, N_d des variables aléatoires i.i.d. $\mathfrak{N}(0, 1)$. Posons

$$R := (N_1^2 + \dots + N_d^2)^{1/2}.$$

Alors le vecteur aléatoire $(\frac{N_1}{R}, \dots, \frac{N_d}{R})$ suit la loi uniforme sur C_d .

Ce corollaire immédiat de la proposition 15 et du théorème 18 b) fournit un algorithme simple et performant pour simuler la loi uniforme sur C_d . Son coût est linéaire par rapport à la dimension d .

Corollaire 20. Soit X un vecteur aléatoire radialement symétrique de densité f avec fonction associée g , i.e. $f(x) = g(\|x\|)$. Soient T une variable aléatoire réelle de densité $r \mapsto g(r)dr^{d-1}V_d\mathbf{1}_{\mathbb{R}^+}(r)$ et W un vecteur aléatoire de loi uniforme sur C_d , indépendant de T . Alors $Y := TW$ a même loi que X .

C'est encore un corollaire du théorème 18 puisque

$$P_{p(X)} = P_{(\|X\|, q(X))} = (P_{\|X\|} \otimes P_{q(X)}) = (P_T \otimes P_W) = P_{(T,W)}.$$

Or $P_X = P_{p(X)} \circ p^{-1}$ et $P_Y = P_{(T,W)} \circ p^{-1}$, d'où $P_X = P_Y$.

L'intérêt pratique de ce corollaire est qu'il fournit un algorithme pour simuler la loi de X radialement symétrique et à densité.

1. Simuler W de loi uniforme sur C_d par la méthode du corollaire 19.
2. Simuler T de densité $r \mapsto g(r)dr^{d-1}V_d\mathbf{1}_{\mathbb{R}^+}(r)$
3. Calculer TW .

Pour le point 2, on peut soit utiliser l'inverse généralisé de la f.d.r. de T s'il est commode à calculer, soit une autre méthode (rejet, transformation, ...).

Un exemple important d'application de l'algorithme ci-dessus est la simulation de la loi uniforme sur B_d . Dans ce cas $f = V_d^{-1}\mathbf{1}_{B_d}$, d'où $g(r) = V_d^{-1}\mathbf{1}_{[0,1]}(r)$. La variable aléatoire positive T a donc une fonction de répartition donnée par

$$F_T(t) = \int_{]-\infty, t]} dr^{d-1}\mathbf{1}_{[0,1]}(r) d\lambda_1(r) = \begin{cases} 0 & \text{si } t < 0, \\ t^d & \text{si } 0 \leq t \leq 1, \\ 1 & \text{si } t > 1. \end{cases}$$

La méthode d'inversion de la f.d.r. nous dit alors que T a même loi que $U^{1/d}$ où U suit la loi uniforme sur $]0, 1[$.

Voici donc l'algorithme complet de simulation de la loi uniforme sur B_d :

1. Simuler N_1, \dots, N_d , i.i.d. $\mathfrak{N}(0, 1)$.
2. Calculer $R := (N_1^2 + \dots + N_d^2)^{1/2}$ et $W := (\frac{N_1}{R}, \dots, \frac{N_d}{R})$.
3. Simuler U uniforme sur $]0, 1[$ indépendante de (N_1, \dots, N_d) et calculer $T := U^{1/d}$.
4. Retourner $Y := TW = U^{1/d}(N_1^2 + \dots + N_d^2)^{-1/2}(N_1, \dots, N_d)$.

Là encore cet algorithme a un coût linéaire par rapport à d , ce qui est bien plus performant que la méthode du rejet.

Remarquons que nous n'avons pas eu besoin de connaître la valeur de V_d donnée par (35) pour simuler la loi uniforme sur B_d . La connaissance de V_d peut néanmoins être utile pour simuler d'autres lois que la loi uniforme sur B_d . Pour son calcul nous aurons recours une fois de plus au vecteur gaussien (N_1, \dots, N_d) .

Proposition 21. Le volume V_d de la boule unité euclidienne de \mathbb{R}^d est donné par

$$V_d := \lambda_d(B_d) = \frac{\pi^{d/2}}{\Gamma(d/2 + 1)},$$

où la fonction Γ est définie sur $]0, +\infty[$ par $\Gamma(a) = \int_0^{+\infty} t^{a-1}e^{-t} dt$.

Le vecteur gaussien (N_1, \dots, N_d) a pour densité $f(x) = (2\pi)^{-d/2} \exp(-\|x\|^2/2)$. En transformant l'intégrale $\int_{\mathbb{R}^d} f \, d\lambda_d$ par le transfert $(\mathbb{R}^d, \lambda_d) \longrightarrow (\mathbb{R}_*^+ \times C_d, \rho_d \otimes \nu_d)$, on obtient

$$\begin{aligned}
 1 = \int_{\mathbb{R}^d} f(x) \, d\lambda_d(x) &= \int_{\mathbb{R}_*^+} (2\pi)^{-d/2} e^{-r^2/2} \left\{ \int_{C_d} d\nu_d(s) \right\} dV_d r^{d-1} \, d\lambda_1(r) \\
 &= V_d (2\pi)^{-d/2} d \int_0^{+\infty} e^{-r^2/2} r^{d-1} \, dr \\
 &= V_d \pi^{-d/2} \frac{d}{2} \int_0^{+\infty} e^{-t} t^{d/2-1} \, dt \\
 &= V_d \pi^{-d/2} (d/2) \Gamma(d/2),
 \end{aligned}$$

d'où le résultat annoncé puisque $a\Gamma(a) = \Gamma(a+1)$.

Bibliographie

- [1] N. BOULEAU. *Probabilités de l'Ingénieur, variables aléatoires et simulation*, Hermann, 2002.
- [2] J.-P. DELAHAYE. Aléas du hasard informatique. Revue *Pour la Science*, n° 245, mars 1998, pp. 92–97.
- [3] L. DEVROYE. *Non-Uniform Random Variate Generation*. Springer, 1986.
- [4] J. JACOD et P. PROTTER. *L'essentiel en théorie des probabilités*. Cassini, 2003.
- [5] D. SERRE. *Les matrices, théorie et pratique*. Dunod, 2001.
- [6] Ch. SUQUET. *Introduction au Calcul des Probabilités*, polycopié de Deug MIAS et MASS, Université des Sciences et Technologies de Lille.
<http://math.univ-lille1.fr/~suquet/ens/ICP/Cmd130903.pdf>