TP3 : SIMULATION DE VARIABLES ALÉATOIRES

Résumé. Ce TP présente les fondements de la simulation de variables aléatoires de loi donnée à partir d'une suite dénombrable de variables uniformes sur le segment unité [0,1]. Remarquez que c'est ce dernier point qui pose problème... D'une manière générale, les outils du TP2 vous aideront à apprécier la qualité des simulations obtenues (*id est*, la conformité à la loi attendue). En particulier, nous verrons plusieurs méthodes de simulations pour la loi normale, et il faudra alors déterminer quels sont les avantages et inconvénients de chaque méthode.

Comme toujours, avant de commencer, prenez connaissance de l'appendice. Chargez également, si ce n'est pas encore fait, la boîte à outils StixBox.

1. Variables discrètes

Exercice 1. Déterminer la loi de la variable aléatoire `X` en sortie des algorithmes :

(1) `X=floor(3*rand)`,
(2) `X=floor(3*rand)*floor(2*rand)`,
(3) `X=round(4*rand)`.

Vérifiez vos réponses numériquement, par simulation de n-échantillons et comparaison des fréquences empiriques aux fréquences que vous proposez (pour ce faire, calculez au préalable un nombre n convenable, avec les techniques du TP précédent).

Application : écrivez une fonction qui simule élégamment un n-échantillon de loi uniforme sur $\{1,\dots,N\}$, où n et N seront les paramètres transmis en entrée.

Nous avons vu au premier exercice du TP2 comment simuler une variable aléatoire discrète. Dans certains cas particuliers, notamment le cas de la simulation de lois binômiales, la manière de procéder peut être particulièrement simple.

Exercice 2. Ecrire une fonction `Bin(k1,k2,n,p)` qui prend en argument 3 entiers `k1`, `k2`, `n` et un réel `p` de [0,1], et qui renvoie un `k1`×`k2`-échantillon de variables aléatoires de loi binômiale de paramètres `n` et `p`. Contrainte : il est interdit d'utiliser une quelconque instruction `if` ou une boucle `for`. Et la solution doit tenir sur une ligne...

Exercice 3. Soit $(X_n)_{n\geq 1}$ un échantillon de variables aléatoires de loi de Bernoulli de paramètre $p \in]0,1[$. Posons $Y = \inf\{k \in \mathbb{N}^* \mid X_k = 1\}$. (Remarquez que vu les hypothèses sur p, l'infimum précédent est atteint presque sûrement.)

(1) Quelle est la loi de Y ?
(2) Ecrire une fonction qui permette de simuler un n-échantillon de loi celle de Y. (Peut-on se passer d'instruction de boucle cette fois ?)
(3) Imaginons que vous disposiez d'un n-échantilllon de variables aléatoires de loi exponentielle de paramètre p (vous aurez à en construire un ci-dessous de vos propres mains, utilisez pour l'instant la fonction `rexpweib`). Comment en déduire un n-échantillon de loi celle de Y ? Ecrivez la fonction correspondante.

(4) Avec l'une ou l'autre de ces fonctions, tirez un n-échantillon, avec n petit puis grand, et déterminez graphiquement si votre n-échantillon est de qualité ou non. Remarquez que cela revient peu ou prou à tester le générateur de nombres uniformes de Matlab...

Exercice 4 [Loi de Poisson]. Ecrivez une fonction simulant un n-échantillon de loi de Poisson de paramètre λ en utilisant la méthode proposée en appendice (à partir de variables aléatoires uniformes). Quels avantages voyez-vous par rapport à la procédure qui aurait imité ce que nous avons fait dans le cas discret, à savoir, partitionner [0,1] en un nombre dénombrable d'intervalles, de longueurs respectives $\exp(-\lambda)\lambda^k/k!$, $k \in \mathbb{N}$, et déterminer le numéro d'ordre dans lequel chaque élément du n-échantillon uniforme eût été ?

2. Variables à densité

2.1. Pour s'échauffer...

Exercice 5. Ecrire des fonctions pour simuler un n-échantillon :

(1) de loi uniforme sur $[a,b]$, pour $a < b$ quelconque ;

(2) de loi de Laplace, de densité sur $\mathbb{R}$ donnée par $x \mapsto e^{-|x|}/2$.

Attention, ces fonctions ne doivent pas comporter de boucles `for`.

Exercice 6 [Loi de Cauchy]. Cette loi admet pour densité sur $\mathbb{R}$

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

Ecrire une fonction qui simule un n-échantillon de loi de Cauchy. Nous allons maintenant faire plus ample connaissance avec cette loi. Admet-elle un moment d'ordre 1 ? Pourtant elle est centrée... Essayer d'estimer une hypothétique « moyenne » en faisant comme si la loi des grands nombres s'appliquait. Or, la loi de Cauchy est parfois notée $C(0,1)$: à quoi correspond le 0 ? Et le 1 ? Vérifiez numériquement vos hypothèses sur un n-échantillon.

Exercice 7 [Loi exponentielle]. Dans l'exercice 3, nous avions eu besoin d'un n-échantillon de loi exponentielle. Ecrivez une fonction qui réalise ceci, à partir d'un n-échantillon de loi uniforme... sans boucle `for` bien sûr.

Comparez maintenant les temps d'exécution des deux fonctions obtenues à l'exercice 3 (en remplaçant l'appel à `rexpweib` par un appel à la fonction que vous venez d'écrire). Il faut aussi comparer la qualité des n-échantillons fournis par chacune d'entre elles, ... à moins que vous ne pensiez qu'ils soient de qualité rigoureusement égale ? Quel est l'argument qui me fait penser que pour de petites valeurs de p (et de grandes valeurs de n), la procédure de simulation par plafond d'exponentielles sera bien meilleure ? Simulez, simulez et dites-moi si vous pouvez mettre en évidence une différence dans la qualité des échantillons obtenus, ... Bref, dites-moi si je me fais du souci pour rien !

2.2. La foire aux méthodes de simulation de la loi normale.

Exercice 8. Ecrire des fonctions permettant de simuler un n-échantillon de loi normale standard,

(1) à l'aide de la méthode de rejet ;

(2) par l'algorithme de Box-Muller.

Aucune de ces fonctions ne devra comporter de boucle `for`. Comparez ensuite la vitesse d'éxécution de ces fonctions à celle obtenue par `randn`. Question subsidiaire : quelle est la méthode utilisée par `randn` ? Peut-on mettre en évidence, graphiquement [1], la qualité des échantillons obtenus par chaque méthode ?

Exercice 9. Simuler un n-échantillon de vecteurs gaussiens de dimension 2, de moyenne $m = (7{,}9)$ et de matrice de covariance $\Gamma = [4\ 3; 3\ 4]$. Et (pour une dernière fois) sans faire appel à `rnorm`... N'oubliez pas de calculer la covariance empirique de votre échantillon, juste histoire de voir si vous ne vous seriez pas trompé dans l'utilisation des commandes d'algèbre linéaire, comme `chol` !

3. Exercices supplémentaires

Exercice 10 [Statistiques d'ordres]. Soit $X = (X_1, \ldots, X_N)$ un N-échantillon de loi commune (diffuse [2]) μ, de fonction de répartition F (et d'inverse généralisée F^{-1} connue). On appelle statistiques d'ordre de X le vecteur $Z = (X_{(1)}, \ldots, X_{(N)})$ obtenu par permutation des éléments de X, et tels que les $X_{(i)}$ soient classés par ordre croissant. $X_{(N)}$ est ainsi en particulier le maximum des X_j, $j = 1, \ldots, N$.

On cherche à simuler Y de loi celle de $X_{(N)}$. Donnez une méthode pour le faire, qui ne requiert qu'une seule variable uniforme. Comment simuler un vecteur de loi celle de Z à partir des statistiques d'ordre d'un N-échantillon de loi uniforme sur [0,1] ?

Exercice 11 [Tirage uniforme de probabilités]. Soit χ le simplexe d'ordre N, c'est-à-dire l'ensemble des distributions de probabilités sur $\{1, \ldots, N\}$. C'est un sous-ensemble convexe de $\mathbb{R}^N$, de dimension $N-1$. Ecrivez une fonction qui tire un n-échantillon distribué uniformément sur χ. Ceci est facile pour $N = 2$, n'est-ce pas ?

Voici une solution qui peut sembler donner le bon résultat dans le cas général : on tire $U_1, \ldots, U_N$ indépendamment et uniformément sur le segment unité, et on lui associe la probabilité P, qui met un poids P_i sur l'élément i, où

$$P_i = \frac{U_i}{\sum_{k=1}^{N} U_k} \ .$$

Cette procédure semble raisonnable parce que les marginales P_i ont toutes la même loi, et sont donc toutes d'espérance $1/N$. Prouvez cependant qu'en faisant ainsi, on ne répond pas à la question, tout d'abord, graphiquement, puis par le calcul.

Sauriez-vous trouver la bonne méthode de simulation ?

4. Appendice : Méthodes théoriques pour la simulation de lois

4.1. **Génération de nombres pseudo-aléatoires.** Un générateur de nombres aléatoires dans l'intervalle [0,1] est une fonction `rand` qui vérifie les deux propriétés suivantes :

(i) un appel à la fonction `rand` donne une réalisation d'une variable aléatoire de loi uniforme sur [0,1],

(ii) les appels successifs à la fonction `rand` fournissent une réalisation d'une suite de variables aléatoires indépendantes.

1. Attendez patiemment le TP sur les tests pour avoir des méthodes numériques qui ne nécessiteront pas votre œil d'aigle, ni votre agilité à la représentation graphique...

2. ou tout autre hypothèse assurant que la probabilité qu'il existe i,j tels que $X_i = X_j$ soit nulle

En pratique, les ordinateurs utilisent une suite de nombres dit « pseudo-aléatoires » pour simuler le tirage de nombres suivant la loi uniforme sur [0,1]. Une méthode pour générer une telle suite $(x_n)_{n\in\mathbb{N}}$ est de définir une suite d'entiers (y_n), par une relation de récurrence de la forme :

$$y_{n+1} = ay_n + b \bmod m,$$

où a, b et m sont des entiers bien choisis, et de poser $x_n = y_n/m$.

Matlab dispose de deux générateurs de nombres aléatoires :

- `rand` pour la loi uniforme sur [0,1],
- `randn` pour la loi normale $\mathcal{N}(0,1)$.

On supposera que la fonction `rand` de Matlab a les deux propriétés (i) et (ii) et que `randn` a les propriétés analogues pour la loi normale $\mathcal{N}(0,1)$.

NB : Au début d'un programme, on peut initialiser le générateur, en lui fournissant la première valeur x_0 appelée la graine du générateur. Sinon, à chaque fois que l'on démarre Matlab, c'est la même suite de valeurs qui sera donnée. (Voir `help rand`.) Avec la commande `rand('state',sum(100*clock))`, le générateur `rand` est initialisé avec une graine dont la valeur dépend de l'heure.

4.2. **Simulation par inversion.** La méthode de simulation par inversion repose sur le résultat suivant.

Proposition 1. *Soit X une variable aléatoire réelle de fonction de répartition $F_X(t) = \mathbb{P}\{X \leq t\}$.*
On définit l'inverse généralisée F_X^{-1} de F_X sur $]0,1[$ par

$$F_X^{-1}(x) = \inf\{t \in \mathbb{R} \,|\, F_X(t) \geq x\}.$$

Alors, si U est une variable aléatoire de loi uniforme sur $]0,1[$, $F_X^{-1}(U)$ a même loi que X.

Ainsi, si on tire n nombres au hasard uniformément répartis entre 0 et 1, $(u_1,u_2,\ldots,u_n)$, l'échantillon recherché, $(x_1,x_2,\ldots,x_n)$, de loi celle de X, sera déterminé par $x_i = F_X^{-1}(u_i)$.

Exercice 12. Prouver la proposition 1, en comparant la fonction de répartition de $F_X^{-1}(U)$ à celle de X.

4.2.1. *Cas d'une variable continue.* En général, dans le cas d'une variable aléatoire continue, et plus particulièrement pour les variables à densité, il faut d'abord calculer F_X^{-1} et ensuite simuler X à partir d'une variable de loi uniforme sur [0,1].

4.2.2. *Cas d'une variable discrète.* Dans le cas d'une variable aléatoire discrète, la méthode est canonique. Soit X une variable aléatoire à valeurs dans $\{x_1,\ldots,x_r\}$ (les x_i étant ordonnés) de loi $(p_1,\ldots,p_r)$, avec $\mathbb{P}\{X = x_j\} = p_j$. On calcule alors que, si $\mathbb{I}_{[a,b]}$ désigne la fonction indicatrice de l'intervalle $[a,b]$ $(a < b)$ et si l'on note $p_0 = 0$,

$$F_X^{-1} = \sum_{j=1}^{r} x_j \, \mathbb{I}_{[p_0+\ldots+p_{j-1},\, p_1+\ldots+p_j[}.$$

Donc, pour simuler un échantillon $(y_1,\dots,y_k)$ d'une telle variable X, on simule un échantillon $(u_1,\dots,u_k)$ de variables de loi uniforme sur [0,1] et on pose $y_i = j$ si $u_i \in [p_0+\dots+p_{j-1}, p_1+\dots+p_j[$.

4.3. **Simulation par méthode de rejet.** La méthode d'inversion nécessitait la connaissance explicite de F_X^{-1}, ce qui n'est pas toujours le cas, ... notamment pour les lois normales !

4.3.1. *Cas particulier d'une variable à densité bornée à support compact.* Dans le cas particulier où f est une densité continue à support compact inclus dans un intervalle $[a,b]$, majorée par un réel $K > 0$, on peut utiliser la procédure suivante. On tire un point P selon une loi uniforme sur le rectangle $[a,b] \times [0,K]$. Si le point P se trouve dans la région située sous le graphe de f, on l'accepte et on pose $X = P_1$, sinon on le rejette et on tire un nouveau point uniformément, et ainsi de suite jusqu'à obtenir un point P qui se trouve dans la région située sous le graphe de f.

On peut montrer les deux résultats suivants :

- presque sûrement, il n'y a qu'un nombre fini de tirages avant que le point P soit accepté,
- la variable X ainsi définie a pour densité f.

4.3.2. *Variable à densité, cas général.* La méthode précédente peut être généralisée au cas où il existe une densité de probabilité g telle que

- on sait simuler une variable de densité g,
- il existe une constante K ($K \geq 1$!) telle que $f \leq Kg$.

La procédure suivante termine presque sûrement et simule une variable de densité f :

(1) On simule une variable U de densité g et une variable W indépendante de U de loi uniforme sur [0,1]. On pose $V = K\,W\,g(U)$.
(2) Si $V < f(U)$ on pose $X = U$, sinon on retourne en (1).

Exemple 1 [Loi normale]. f est la densité de la loi normale et g, celle de la loi de Laplace :

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) \leq Kg(x)$$

avec $K = \sqrt{\frac{2e}{\pi}}$ et $g(x) = \exp(-|x|)/2$.

4.4. **Procédés particuliers et méthodes *ad hoc*.**

4.4.1. *Loi de Poisson.* Si $(E_i)_{i\in\mathbb{N}^*}$ est un échantillon de variables aléatoires de loi exponentielle de paramètre 1, alors

$$X = \inf\,\{n \geq 0 \,|\, E_1 + \dots + E_{n+1} > \lambda\}$$

suit une loi de Poisson de paramètre λ.
Donc[3], si $(U_i)_{i\in\mathbb{N}^*}$ est un échantillon de variables aléatoires de loi uniforme sur [0,1], la variable

$$X = \inf\{n \geq 0 \,|\, U_1 \times \dots \times U_{n+1} < e^{-\lambda}\}$$

suit une loi de Poisson de paramètre λ.

3. Preuve par méthode d'inversion ! Voir exercice 7.

Remarque : Dans Matlab, la boîte à outils `stats` contient la fonction `poissrnd(λ,n,m)` permettant de simuler directement une matrice de taille `n`×`m` de variables aléatoires indépendantes de loi de Poisson de paramètre λ. On peut aussi utiliser `rpoiss` de Stixbox.

4.4.2. *Simulation d'une variable aléatoire de loi* $\mathcal{N}(0,1)$. F_X^{-1} n'est pas connue dans le cas de la loi gaussienne centrée réduite, mais Matlab possède déjà un générateur, utilisant la méthode de Box-Muller. Soient U et V deux variables aléatoires indépendantes de loi uniforme sur [0,1]. Les variables X et Y définies par

$$X = \sqrt{W}\cos(2\pi V)\ ,$$
$$Y = \sqrt{W}\sin(2\pi V)$$

sont indépendantes et de loi $\mathcal{N}(0,1)$. (Cela se prouve par changement de variables.) On en déduit la simulation d'un vecteur aléatoire gaussien de matrice de covariances l'identité et de moyennes le vecteur nul.

4.4.3. *Simulation de la loi* $\mathcal{N}(m,\Gamma)$. Ici, $m \in \mathbb{R}^d$ et Γ est une matrice de $\mathcal{M}_d(\mathbb{R})$, symétrique et postive. Soit C une matrice telle que $CC^T = \Gamma$ et Z un vecteur aléatoire gaussien de loi $\mathcal{N}(0,I_d)$. Alors $X = CZ + m$ a pour loi $\mathcal{N}(m,\Gamma)$.

NB : En pratique pour déterminer C, on utilise la décomposition de Cholesky lorsque Γ est définie positive (commande `chol`). Cette décomposition calcule une matrice triangulaire supérieure A telle que $A^TA = \Gamma$. Il suffit alors de prendre $C = A^T$.

Signalons enfin que tout ceci peut être évité en utilisant la commande `pnorm` de Stixbox...

4.5. **Rappel : la boîte à outils Stixbox.** Stixbox est une boîte à outils, soit un ensemble de fonctions, qui couvre l'essentiel des besoins en probabilités et statistiques. On y trouve en particulier les densités, les fonctions de répartition, fonctions quantiles et générateurs aléatoires relatifs aux lois de probabilités les plus classiques. Regardez le contenu de cette boîte à outils en tapant `help stixbox`. C'est elle que nous utiliserons dans les TPs suivants !

Références

[1] B. Ycart. *Simulation de variables aléatoires.* Université René Descartes, Paris.
[2] N. Bouleau. *Probabilités de l'ingénieur.* Hermann, Paris.
[3] S. Lemaire. Documents pour la préparation à l'agrégation d'Orsay.
[4] B. Laurent. Documents pour les TPs de Matlab de la maîtrise de mathématiques ingénierie