BE Probabilités avec MATLAB

1 Générateur de nombres pseudo-aléatoires par congruence linéaire

L'une des méthodes très utilisées actuellement, pour la génération de nombres aléatoires, est la méthode des congruences linéaires. Cette méthode permet d'obtenir une suite $\{x_n\}_{n=1...N}$ de nombres aléatoires de la façon suivante :

$$x_{n+1} = (a * x_n + b)(\text{mod } m)$$

La première valeur, notée x_0, est appelé la graine (*seed* en anglais). Les nombres a et b sont les coefficients entiers du générateur. Une suite de N valeurs pseudo-aléatoires sur l'intervalle $]0, 1[$ est obtenue en divisant par m les éléments de la suite $\{x_n\}_{n=1...N}$ (eux-mêmes compris entre 0 et $m-1$, notez en particulier que la graine x_0 devra être choisie entre 0 et $m-1$).

1. Ecrire un programme qui affiche les éléments de la série $\{x_n\}_{n=1...N}$ pour $N = 15$, générés selon cette méthode. On fixera dans un premier temps les valeurs numériques suivantes pour les paramètres du générateur : $a = 3$, $b = 3$, $m = 5$ et $x_0 = 0$.

Le choix de cette faible valeur de m ($m = 5$) permet d'observer plus facilement certaines propriétés :

— La suite générée présente un caractère périodique par répétition d'une séquence donnée.
— Chaque nombre n'est présent qu'une seule fois par séquence.
— Un nombre est absent de cette séquence : le générateur ne produit donc pas de manière équiprobable des nombres sur $\{0, 1, 2, 3, 4\}$ mais sur $\{0, 2, 3, 4\}$. La longueur de la séquence est donc de 4.

Plus généralement, la périodicité de la suite générée est observée, éventuellement après un transitoire, quelles que soient les valeurs de a, b et m. La longueur de la séquence répétée est appelée période. Le générateur n'est donc pas aléatoire mais pseudo-aléatoire. En pratique, un choix de m très grand conduit à une période très grande. On considère que plus cette période est grande, meilleur est le générateur. La période maximale est égale à m. Pour obtenir cette période maximale, il suffit que les paramètres vérifient certaines propriétés. On montre que, quelle que soit la graine, la période est maximale si et seulement si :

— PGCD$(a, m) = 1$ et PGCD$(b, m) = 1$, $a > 0$;
— si p est un nombre premier qui divise m, alors p divise $a - 1$;
— si 4 divise m, alors 4 divise $a - 1$.

1. Vérifier que les valeurs précédentes de a, b et m ne satisfont pas les 3 conditions ci-dessus. Pour ces valeurs, choisir comme nouvelle valeur de la graine $x_0 = 1$. Qu'observe-t-on ?
2. Vérifier que les valeurs suivantes $a = 5$, $b = 3$, $m = 8$ satisfont les 3 conditions ci-dessus. Lancer le programme précédent avec ces valeurs et avec $x_0 = 0$ puis avec $x_0 = 1$.
3. Proposer un choix aléatoire de la graine en utilisant la seconde actuelle, fournie par la commande `datetime('now').Second` de matlab. Vérifier, pour différentes valeurs de la graine, que la période reste maximale lorsque les 3 conditions sur a, b et m sont vérifiées.

2 Validation d'un générateur de nombre pseudo-aléatoire

Nous allons valider le générateur de nombres pseudo-aléatoires correspondant, sous MATLAB, à la fonction `randi`. La commande `randi(m,M,N)` fournit une matrice de taille $M \times N$ dont les éléments sont censés être indépendants et uniformes sur $\{1, ..., m\}$.

1. Utiliser, de manière appropriée, l'outil que constitue l'histogramme (fonction `hist` sous MATLAB) pour mettre en évidence le caractère (pseudo-)uniforme des réalisations. On prendra $m = 10$, $M = 1$ et on fera varier N.
2. Comparer les fréquences relatives des différentes valeurs possibles calculées à partir d'une normalisation appropriée de l'histogramme.

3 Exemples d'application

3.1 Génération de variables de Bernoulli

La fonction `rand` sous MATALB permet de générer des variables aléatoires uniformes sur l'intervalle $]0, 1[$. En utilisant un conditionnement sur ces variables, il est possible de générer des variables de Bernoulli. Une variable de Bernoulli est une variable aléatoire discrète X qui prend les valeurs 0 et 1 avec les probabilités

$$\begin{aligned} \mathbb{P}[X = 0] &= 1 - p, \\ \mathbb{P}[X = 1] &= p \end{aligned}$$

(avec $0 < p < 1$). On a alors $\mathbb{E}[X] = p$ et $\mathrm{Var}(X) = p(1 - p)$.

1. Générer une séquence de variables de Bernoulli en introduisant une condition (comparaison à un seuil) sur les valeurs obtenues en sortie de la fonction `rand`.
2. Vérifier que la moyenne et la variance empiriques de X obtenues grâce aux commandes `mean` et `var` sont proches des valeurs théoriques attendues. Vérifier que les fréquences relatives sont proches des probabilités théoriques.

3.2 Variables binômiales

La somme de n variables de Bernoulli de paramètre p est appelée *variable binômiale* $\mathcal{B}(n, p)$. On a alors $\mathbb{E}[X] = np$ et $\mathrm{Var}(X) = np(1 - p)$. Cette loi permet de modéliser le tirage *avec remise* de n boules dans une urne contenant des boules blanches et des boules noires en proportion p et $1 - p$. La probabilité d'avoir k boules blanches est alors donnée par :

$$\mathbb{P}[X = k] = \binom{n}{k} p^k (1 - p)^{n-k}, \qquad k \in \{0, ..., n\}.$$

1. En utilisant la réponse à la question précédente, générer des variables binômiales.
2. Comparer les valeurs empiriques de la moyenne et de la variance et les fréquences relatives aux valeurs théoriques de la moyenne, de la variance et des probabilités.

3.3 Génération de variables de Poisson

On montre que si $(U_n)_{n \in \mathbb{N}^*}$ est une suite de variables aléatoires indépendantes, de loi uniforme sur $[0, 1]$, alors la variable définie par

$$X = \inf \left\{ n \geq 0 \,|\, U_1 \times U_2 \times \ldots \times U_{n+1} < e^{-\lambda} \right\}$$

suit une loi (discrète) de Poisson de paramètre λ. L'objectif de cet exercice est de générer des échantillons de loi de Poisson à partir de ce résultat. L'algorithme de génération d'**une seule** variable aléatoire est donc :

1. `Initialisation :` $n = 0$, $Y_n = 1$;
2. `Générer un échantillon noté` U `suivant une loi uniforme sur` $[0;1]$; $Y_n \leftarrow Y_n \times U$
3. `Si` $Y_n < e^{-\lambda}$, `on pose` $X = n$. `Arrêt.`
   `Sinon, on incrémente` n, `et on recommence au point` **2.**

Il s'agit d'écrire un programme qui permet de générer des variables aléatoires de Poisson et de vérifier leurs propriétés. Les commandes seront sauvées dans un script nommé `Poisson.m`.

1. En utilisant l'algorithme présenté précédemment, générer N échantillons de loi Poisson de paramètre λ. (On veillera à respecter exactement l'ordre des commandes tel que décrit par le pseudo-code ci-dessus.) Le tester avec $N = 10000$ et $\lambda = 2$.
2. Calculer à partir des échantillons la moyenne et la variance empiriques et les comparer aux valeurs théoriques ($\mathbb{E}[X] = \mathrm{Var}(X) = \lambda$).
3. A l'aide de la fonction `hist`, déterminer les fréquences relatives et les comparer aux valeurs théoriques $\mathbb{P}[X = k]$ pour $k = 0, \ldots, K$, données par la fonction MATLAB `poisspdf`. Afficher sur une même figure les deux séries de valeurs. On choisira une valeur appropriée pour K selon les données (réalisations) générées par votre algorithme.

   Indication : La fonction `bar` permet d'afficher deux séries de valeurs "en parallèle," sous forme de barres de couleurs différentes. Il suffit pour cela d'utiliser `bar(C, [X, Y])` où `C` est un vecteur colonne contenant les labels des classes (des valeurs en abscisse), et `X, Y` sont des vecteurs colonne (de même taille que `C`) contenant les séries de données à afficher.

   Remarque : au cas où vous n'avez pas réussi à faire la Question 1, vous pouvez répondre à la Question 3 en générant des échantillons de loi de Poisson directement à l'aide de la fonction `poissrnd`.

3.4 Loi exponentielle

On rappelle que si X est une variable aléatoire continue de fonction de répartition $F(x)$ strictement croissante, alors la variable aléatoire $Y = F(X)$ est uniformément distribuée sur $[0, 1]$. Par conséquent, pour générer la variable aléatoire X, il suffit de générer la variable uniforme Y, et de calculer $X = F^{-1}(Y)$ (dans le cas où F^{-1} a une forme simple).

1. Appliquer cette propriété pour générer une variable de loi exponentielle de paramètre $\lambda > 0$.
2. Vérifier le résultat en calculant moyenne et variance, et en visualisant l'histogramme.
   Rappel : la densité de probabilité d'une variable exponentielle X est donnée par :

$$f(x) = \begin{cases} \lambda \exp(-\lambda x), & \text{si } x \geq 0, \\ 0, & \text{sinon.} \end{cases}$$

   On a alors $\mathbb{E}[X] = \frac{1}{\lambda}$ et $\mathrm{Var}(X) = \frac{1}{\lambda^2}$.

3.5 Méthode de Box-Muller

Soit U_1 et U_2 deux variables aléatoires indépendantes, de loi uniforme sur $]0, 1[$. On pose

$$X_1 = \sqrt{-2 \log U_1} \cos(2\pi U_2) \text{ et } X_2 = \sqrt{-2 \log U_1} \sin(2\pi U_2). \tag{1}$$

On montre que X_1 et X_2 sont des variables aléatoires gaussiennes centrées réduites, c'est-à-dire $X_1 \sim \mathcal{N}(0, 1)$ et $X_2 \sim \mathcal{N}(0, 1)$. On souhaite dans cet exercice vérifier par simulations ces densités

de probabilité théoriques, en utilisant l'histogramme. Sauver les commandes dans un script `box-muller.m` qui effectue les opérations suivantes :

1. Générer $N = 1000$ réalisations de U_1 et U_2 (stockées dans une matrice `U` de taille $2 \times N$).
2. Effectuer le changement de variable (1) pour obtenir les variables X_1 et X_2 (stockées dans une matrice `X` de taille $2 \times N$).
3. Tracer dans la même figure un histogramme de X_1 et la densité de la loi $\mathcal{N}(0,1)$, à l'aide de la fonction `normpdf`. On utilisera une discretisation assez fine de l'axe des abscisses pour l'histogramme, que l'on normalisera de façon appropriée.
   Rappel : La densité d'une loi au point x_0 peut s'écrire comme $\lim_{\delta \to 0} \frac{\mathbb{P}(x_0 \leq X \leq x_0 + \delta)}{\delta}$.

3.6 Illustration du théorème central-limite

Soient X_k, $k = 1, \cdots, K$, des variables aléatoires indépendantes et de même loi, de moyenne m et de variance σ^2. On pose

$$S_K = \sum_{k=1}^{K} X_k,$$

alors

$$\lim_{K \to +\infty} \frac{S_K - Km}{\sqrt{K}\sigma} \overset{\text{en loi}}{=} \mathcal{N}(0,1),$$

c'est-à-dire que l'on sait construire une loi gaussienne centrée réduite à partir de variables aléatoires de n'importe quelle loi, indépendantes et identiquement distribuées. Ce théorème va permettre de traiter de nombreux problèmes en faisant l'hypothèse de loi gaussienne pour le phénomène observé. On propose de vérifier ce théorème à partir de la somme de K variables aléatoires uniformes sur $]0,1[$.

1. Calculer les paramètres (théoriques) intervenant dans l'expression $\frac{S_K - Km}{\sqrt{K}\sigma}$ lorsque $K = 12$. Enoncer le principe associé.
2. En utilisant la fonction `rand` de MATLAB, générer une matrice contenant N réalisations de chacune des K variables X_k, indépendantes uniformes sur $]0,1[$. On prendra $N = 1000$ et, dans un premier temps, $K = 12$.
3. En appliquant la fonction `sum` à cette matrice, calculer un vecteur contenant N variables aléatoires correspondant chacune à la somme de K variables aléatoires indépendantes uniformes sur $]0,1[$.
4. Calculer une version centrée réduite de ce vecteur.
5. Tracer l'histogramme et, en superposition, la densité de la loi gaussienne $\mathcal{N}(0,1)$, comme dans la Question 3 de l'Exercice 3.5.
6. Répéter les opérations précédentes en faisant varier K de 1 à 12 (pour cela il est possible d'extraire le nombre de lignes nécessaires dans la matrice précédemment générée) et observer la convergence vers la loi gaussienne. On pourra utiliser la fonction `subplot(m,n,p)` afin d'afficher une grille $m \times n$ des 12 plots demandés.