Séance n°4 : Simulation d'une loi de probabilité

Module de Génie Informatique en Licence 3 d'électronique

`Laboratoire LTSI - UMR INSERM 642 - Université de Rennes1`

Nous allons lors de ce TP voir comment générer des réalisations d'une variable aléatoire de loi donnée à partir de réalisations d'une loi uniforme sur [0,1]. En effet, les calculateurs actuels permettent de générer des réalisations x_n d'une variable X de loi uniforme sur [0,1] (noté $X \sim \mathcal{U}([0,1])$ en abrégé). Ceci se fait à l'aide de la fonction *rand* sous Matlab. La solution proposée dans ce TP, plus connue sous le nom de *méthode d'inversion*, permet à partir de la suite (x_n) de générer des réalisations y_n d'une variable aléatoire Y de fonction de répartition F_Y strictement croissante. Cette méthode se résume à la ligne suivante :

$$\boxed{\forall\ n \in \mathbb{N}^*,\ y_n = F_Y^{-1}(x_n) \text{ où } F_Y^{-1} \text{ est l'application réciproque de } F_Y.}$$

Démo. : Montrons que la variable aléatoire Y, définie par $Y = F_Y^{-1}(X)$ où $X \sim \mathcal{U}([0,1])$, admet pour fonction de répartition la fonction F_Y. La fonction de répartition de Y est de manière générale définie par la mesure de probabilité $P(Y < y)$ qui pour toute valeur réelle y vérifie :

$$\begin{aligned}
P(Y < y) &= P(F_Y^{-1}(X) < y) \quad \text{par définition de } Y \\
P(Y < y) &= P(F_Y(F_Y^{-1}(X)) < F_Y(y)) \quad \text{puisque } F_Y \text{ est strictement croissante} \\
P(Y < y) &= P(X < F_Y(y)) \\
P(Y < y) &= F_X\left(F_Y(y)\right) \quad \text{par définition de la fonction de répartition, } F_X\text{, de } X \\
P(Y < y) &= F_Y(y)\mathbb{1}_{[0,1]}(F_Y(y)) + \mathbb{1}_{]1,+\infty[}(F_Y(y)) \quad \text{puisque } X \sim \mathcal{U}([0,1]) \\
P(Y < y) &= F_Y(y) \quad \text{puisque la mesure de probabilité } F_Y(y) \text{ appartient à } [0,1]
\end{aligned}$$

où $\mathbb{1}_{[a,b]}(.)$ désigne la fonction indicatrice, égale à 1 sur l'intervalle $[a,b]$, et à 0 ailleurs.

1) A titre d'exemple, générer $N_e = 1000$ réalisations d'une variable aléatoire de loi exponentielle[1] de paramètre $a = 0.2$ en utilisant la méthode d'inversion décrite ci-dessus. On stockera ces N_e réalisations dans le vecteur y. On rappelle par ailleurs qu'une variable aléatoire de loi exponentielle est à valeurs réelles positives, de loi continue avec une densité de probabilité de la forme :

$$p_Y(y) = ae^{-ay}\mathbf{1}_{[0,\infty[}(y), y \in \mathbb{R}$$

où a est un paramètre réel positif. La fonction de répartition correspondante est :

$$F_Y(y) = (1 - e^{-ay})\mathbf{1}_{[0,\infty[}(y), y \in \mathbb{R}$$

[1] la loi exponentielle est fréquemment rencontrée en pratique pour simuler des instants d'arrivée ou des durées aléatoires.

En observant que la restriction sur les réels positifs de cette fonction, $F_Y : \mathbb{R}^{*+} \rightarrow]0,1[$ est continue strictement croissante, cette fonction de répartition admet une application réciproque $F_Y^{-1} : [0,1[\rightarrow \mathbb{R}^{*+}$ définie par

$$\forall y \in [0,1[, \quad F_Y^{-1}(x) = -\frac{1}{a} log(1-x)$$

2) Vérifier à l'aide de la fonction *hist* de Matlab et des connaissances acquises lors de la séance n°1 de TP que les réalisations construites à la question précédente sont bien celle d'une variable aléatoire de loi exponentielle de paramètre $a = 0.2$.

3) Effectuer la vérification précédente mais cette fois en estimant la fonction de répartition de la variable aléatoire dont N_e réalisations ont été générées à la question 1). Pour cela, il faut se donner un estimateur $\widehat{F}_X$ de la fonction de répartition F_X d'une variable aléatoire X à partir des N_e tirages indépendants x_n de X. Ce dernier est donné par la formule suivante :

$$\widehat{F}_X(x) = \frac{1}{N_e} \sum_{1}^{N_e} H(x - x_i), x \in \mathbb{R}$$

où H est la fonction échelon unité continue à gauche. Cet estimateur a la forme d'une fonction en escalier avec des marches de hauteur proportionnelle à $1/N_e$ en supposant, ce que l'on fera dans la suite, que les tirages x_n sont tous distincts. Une implémentation optimisée de cet estimateur est proposée ci-dessous :

$N_a = 1000;$ % nombre de points où est calculée la fonction de répartition expérimentale
$z = sort(y);$ % ordonnancement des composantes du vecteur y dans l'ordre croissant
$\alpha = z(1) : (z(N_e) - z(1))/(N_a - 1) : z(N_e);$ % abscisses de la fct de répartition expérimentale

```
Pour n_a allant de 1 à N_a
    Pour ne allant de 1 à N_e
        Si α(n_a) > z(n_e) et α(n_a) ≤ z(n_e + 1)
            F̂_Y(n_a) = n_e/N_e;
        Fin
    Fin
    Si α(n_a) ≤ z_1
        F̂_Y(n_a) = 0
    Autrement si α(n_a) > z(N_e)
        F̂_Y(n_a) = 1;
    Fin
Fin
```