feuilles de TD Scilab online

I– Rappels sur rand() et grand()

rand() et grand()

- La commande `rand()` renvoie un nombre pseudo-aléatoire entre 0 et 1.
- La commande `grand(m,n,'uin',a,b)` renvoie une matrice de taille $m \times n$ dont les éléments sont des **entiers** suivant une loi uniforme sur $[\![a;b]\!]$.
- La commande `grand(m,n,'unf',a,b)` renvoie une matrice de taille $m \times n$ dont les éléments sont des réels suivant une loi uniforme sur $[a;b[$.
- La commande `grand(m,n,'bin',N,p)` renvoie une matrice de taille $m \times n$ dont les éléments suivent une loi binômiale de paramètres (N,p).
- La commande `grand(m,n,'geom',p)` renvoie une matrice de taille $m \times n$ dont les éléments suivent une loi géométrique de paramètre p.

II– Extraits de sujets de concours

Exercice 1 *extrait Ecricome 2017*
Une urne $\mathcal{U}$ contient 1 boule noire et 3 boules blanches indiscernables au toucher.
Une urne $\mathcal{V}$ contient 2 boules noires et 2 boules blanches indiscernables au toucher.
On lance une pièce équilibrée. Si elle retombe du côté Pile, on tire deux boules successivement et avec remise dans $\mathcal{U}$, et si on obtient Face, on tire deux boules successivement et avec remise dans $\mathcal{V}$.
On note T la variable aléatoire égale au nombre de fois où l'on a pioché une boule noire.
On rappelle qu'en langage `Scilab` l'instruction `grand(1,1,'uin',n1,n1)` renvoie un entier au hasard et uniformément compris entre `n1` et `n2`. Compléter le programme `Scilab` suivant afin qu'il affiche une simulation de la variable aléatoire T.

```
T = ...
if grand(1,1,"uin",1,2) == 1 then
    for k = 1:2
        if grand(1,1,"uin",1,4)<2 then
            T = T + 1
        end
    end
else
    ...
    ...
    ...
    ...
    ...
end
disp(T,"une simulation de T donne :")
```

Question rajoutée : déterminer de manière mathématique l'espérance de la variable T. Retrouver ce résultat par une simulation.

Exercice 2 *non extrait d'annales*
Un gardien de nuit doit ouvrir une porte dans le noir et choisit aléatoirement une clé parmi les 10 clés de son trousseau. Comme il ne peut pas les distinguer, il peut essayer plusieurs fois les mêmes clés sans s'en rendre compte. Soit X la variable aléatoire qui compte le nombre d'essais jusqu'à ce que la porte s'ouvre.

1. Que vaut $E(X)$?
2. Écrire un programme `Scilab` qui simule 100000 ouvertures de porte et retrouver la valeur théorique de $E(X)$.

Exercice 3 *extrait Ecricome 2018*

On considère une urne U contenant deux boules blanches et une boule noire indiscernables au toucher, ainsi qu'une urne V contenant une boule blanche et trois boules noires, elles aussi indiscernables au toucher. On effectue une suite de tirages d'une boule dans ces urnes en procédant comme suit :

- le premier tirage a lieu dans l'urne U ;
- tous les tirages s'effectuent avec remise de la boule piochée dans l'urne dont elle provient ;
- si l'on pioche une boule blanche lors d'un tirage, le tirage suivant a lieu dans l'autre urne ;
- si l'on pioche une boule noire lors d'un tirage, le tirage suivant a lieu dans la même urne.

Pour tout entier naturel non nul n, on note X_n la variable aléatoire égale au nombre de boules blanches piochées au cours des n premiers tirages.

On rappelle qu'en `Scilab`, l'instruction `grand(1,1,'uin',1,k)` renvoie un entier aléatoire compris entre 1 et k. Recopier et compléter les lignes à pointillés du script `Scilab` ci-dessous afin qu'il simule la variable aléatoire X_2 :

```
function X2 = simulation ()
    tirage1 = grand(1, 1, 'uin', 1, 3)
    if tirage1 < 3 then
        res1 = 1
        tirage2 = grand(1, 1, 'uin', 1, 4)
        if tirage2 == 1 then res2 = 1
        else res2 = 0
        end
    else
        res1 = 0
        tirage2 = ...
        if tirage2 < 3 then res2 = ...
        else res2 = ...
        end
    end
    X2 = res1 + res2
endfunction
```

Question rajoutée : déterminer de manière mathématique l'espérance de la variable X_2. Retrouver ce résultat par une simulation.

Exercice 4 *extrait Ecricome 2019*

Soit g la fonction numérique réelle définie sur l'intervalle $]0; +\infty[$ par :

$$g(x) = 2x - 1 + \ln\left(\frac{x}{x+1}\right)$$

On note $(\mathcal{C})$ sa courbe représentative dans un repère orthonormé.

1. **a.** Calculer la limite de g en 0 et interpréter graphiquement le résultat.

b. Montrer que $\lim\limits_{x\mapsto+\infty} \ln\left(\frac{x}{x+1}\right) = 0$. En déduire la limite de g en $+\infty$.

2. Étudier le sens de variation de g sur $]0; +\infty[$ et dresser son tableau de variation.

3. **a.** Démontrer que la courbe $(\mathcal{C})$ admet une asymptote oblique (D) dont on précisera une équation.

b. Étudier la position de $(\mathcal{C})$ par rapport à (D).

4. **a.** Démontrer que l'équation $g(x) = 0$ admet une unique solution α sur l'intervalle $]0; +\infty[$

b. Compléter l'algorithme de dichotomie ci-dessous écrit à l'aide de `Scilab` afin qu'il affiche un encadrement de α à 10^{-2} près lorsque a et b sont choisis tels que $\alpha \in [a, b]$.

```
function y = g(x)
    y = ...
endfunction
a = input('Entrer la valeur de a : ')
b = input('Entrer la valeur de b : ')
while b - a ...
    m = ...
    if g(a)*g(m) <= 0 then
        b = ...
    else
        ...
    end
end
disp(...)
```

c. Le programme `Scilab` affiche `0.88` comme résultat. Dans un même repère orthonormé, placer le réel α sur l'axe des abscisses, tracer la courbe $(\mathcal{C})$ et la droite (D).