Devoir maison $n^{o} 2$

A rendre le Mardi 1er Octobre 2019

Exercice 1.

On lance indéfiniment une pièce donnant "Pile" avec la probabilité p et "Face" avec la probabilité q=1-p. On suppose que $p \in]0,1[$ et on admet que les lancers sont mutuellement indépendants.

Pour tout entier naturel k, supérieur ou égal à 2, on dit que le $k^{i\grave{e}me}$ lancer est un changement s'il amène un résultat différent de celui du $(k-1)^{i\grave{e}me}$ lancer.

On note P_k (resp. F_k) l'événement : "on obtient Pile (resp. Face) au $k^{i\grave{e}me}$ lancer".

Pour ne pas surcharger l'écriture on écrira, par exemple, P_1F_2 à la place de $P_1 \cap F_2$.

Pour tout entier naturel n supérieur ou égal à 2, on note X_n la variable aléatoire égale au nombre de changements survenus durant les n premiers lancers.

Partie 1 : étude de quelques exemples.

- 1. Donner la loi de X_2 .
- 2. (a) Donner la loi de X_3 .
 - (b) Vérifier que $E(X_3) = 4pq$ et que $V(X_3) = 2pq(3 8pq)$.
- 3. (a) Trouver la loi de X_4 .
 - (b) Calculer $E(X_4)$.

Partie 2 : étude du cas $p \neq q$. Dans cette partie, n désigne un entier naturel supérieur ou égal à 2.

- 1. Exprimer $P(X_n = 0)$ en fonction de p, q et n.
- 2. En décomposant l'événement $(X_n = 1)$ en une réunion d'événements incompatibles, montrer que $P(X_n = 1) = \frac{2pq}{q-p} \left(q^{n-1} p^{n-1}\right).$
- 3. En distinguant les cas n pair et n impair, exprimer $P(X_n = n 1)$ en fonction de p et q.
- 4. Pour tout entier naturel k, supérieur ou égal à 2, on note Z_k la variable aléatoire qui vaut 1 si le $k^{i\hat{e}me}$ lancer est un changement et 0 sinon (Z_k est donc une variable de Bernouilli).

Écrire X_n à l'aide de certaines des variables Z_k et en déduire $E(X_n)$.

Partie 3: simulation

- On rappelle qu'en Scilab on code pile par 1 et face par 0. Un lancer consiste donc à simuler une variable de Bernoulli de paramètre p (i.e. une binomiale de taille 1 et de paramètre p).
- On rappelle également que l'instruction L=grand(1,n,'bin',N,p) crée une matrice à 1 ligne et n colonnes dont chaque coefficient est la simulation d'une variable binomiale de taille N et de paramètre p. Ainsi, l'instruction L=grand(1,n,'bin',1,p) simule n variables de Bernoulli de paramètre p (n lancers de pièce).
- \bullet On rappelle enfin que l'opérateur mathématique " \neq " est codé par l'opérateur " $\!\!\!<\!\!\!>$ " en Scilab.
 - 1. Compléter les instructions suivantes pour créer une fonction qui, étant donnés n et p, effectue n lancers de pièces (avec proba p d'obtenir pile) et renvoit en sortie le nombre X de changements survenus au cours des n lancers :

```
function X=nbr_changements(n,p)
    X=0
    L=grand(1,n,'bin',1,p)
    for k=2:n do
        if L(k) <> L(k-1) then
            X=********
        end
    end
endfunction
```

2. On rajoute ensuite les instructions suivantes. Expliquer ce que font ces instructions en précisant ce qui est calculé dans la variable f.

```
n=10
p=1/4
q=1-p
S=0
for k=1:10000 do
    X=nbr_changements(n,p)
    if X==1 then
        S=S+1
    end
end
f=S/10000
```

3. On rajoute à la fin du programme :

```
disp(f)
disp(2*p*q/(q-p)*(q^{n-1}-p^{n-1}))
```

Après exécution du programme, on obtient les résultats suivants :

- 0.0564
- 0.0563107

Pourquoi les 2 valeurs affichées sont-elles proches?

Exercice 2.

Partie 1 : Préliminaire Soit $x \in \mathbb{R} \setminus \{1\}$. Donner la valeur de $\sum_{k=0}^{n} x^k$ puis en dérivant des deux côtés l'égalité obtenue, montrer que :

$$\sum_{k=1}^{n} kx^{k-1} = \frac{nx^{n+1} - (n+1)x^n + 1}{(1-x)^2}$$

Partie 2 : On désigne par n un entier naturel supérieur ou égal à 2. On note p un réel de]0;1[et on pose q=1-p.

On dispose d'une pièce donnant "Pile" avec la probabilité p et "Face" avec la probabilité q.

On lance cette pièce et on arrête les lancers dans l'une des deux situations suivantes :

- Soit si l'on a obtenu "Pile".
- \bullet Soit si l'on a obtenu n fois "Face".

Pour tout entier naturel k non nul, on note P_k (respectivement F_k l'événement « on obtient "Pile" (respectivement "Face") au k^e lancer ».

On note T_n le nombre de lancers effectués, X_n le nombre de "Pile" obtenus et enfin Y_n le nombre de "Face" obtenus.

On admet que T_n , X_n et Y_n sont des variables aléatoires toutes les trois définies sur un espace probabilisé $(\Omega; \mathcal{A}; P)$ que l'on ne cherchera pas à préciser.

- 1. Loi de T_n .
 - (a) Pour tout k de [1; n-1], déterminer, en distinguant le cas k=1, la probabilité $P(T_n=k)$.
 - (b) Déterminer $P(T_n = n)$.
 - (c) Vérifier que $\sum_{k=1}^{n} P(T_n = k) = 1$.
 - (d) Établir que T_n possède une espérance et vérifier que $E\left(T_n\right) = \frac{1-q^n}{1-q}$.
- 2. Loi de X_n .
 - (a) Donner la loi de X_n .
 - (b) Vérifier que $E(X_n) = 1 q^n$.

- 3. Loi de Y_n .
 - (a) Déterminer, pour tout k de $\llbracket 0; n-1 \rrbracket$, la probabilité $P(Y_n=k)$.
 - (b) Déterminer $P(Y_n = n)$.
 - (c) Écrire une égalité liant les variables aléatoires T_n , X_n et Y_n , puis en déduire $E(Y_n)$.

4. Simulation informatique:

(a) Compléter les trois instructions manquantes pour que le programme suivant simule l'expérience aléatoire décrite ci-dessus et pour qu'il affiche, dans cet ordre, les valeurs prises par les variables aléatoires T_n , X_n et Y_n , à l'exécution de l'instruction $\mathtt{disp(t,x,y)}$

```
function [t,x,y]=simul(n,p)
    t=0,x=0,y=0
    while (x==0)&(t<n) do
        t=******
        if rand() >p then
            y=******
        else
            x=******
        end
    end
endfunction
```

(b) On considère la fonction suivante :

```
function L=loi_theorique(n,p)
   L=zeros(1,n)
   L(1)=p
   for k=2:n do
        L(k)=q*L(k-1)
   end
   L(n)=q/p*L(n-1)
endfunction
```

Pour tout $k \in [\![1,n-1]\!]$ donner la valeur p_k mémorisée dans la case $\mathtt{L}(\mathtt{k})$ du vecteur ligne \mathtt{L} , puis donner la valeur p_n mémorisée dans $\mathtt{L}(\mathtt{n})$.

Cette fonction renvoit en sortie la loi d'une variable aléatoire. Laquelle?

(c) On rajoute à la suite de ces fonctions le programme suivant :

```
n=10
p=1/3
T= zeros(1,1000)
for k=1:1000 do
     [t,x,y]=simul(n,p)
     T(k)=t
end
M=tabul(T,'i')
disp(M, "M=")
```

Après exécution, on obtient le résultat suivant :

M=

```
1.
        320.
2.
        222.
3.
        156.
4.
        81.
5.
        73.
6.
        55.
7.
        26.
8.
        23.
9.
        19.
```

25.

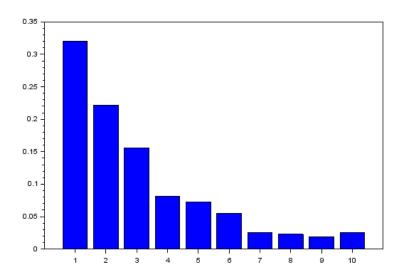
10.

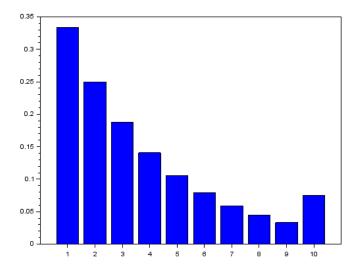
Que fait ce programme? Donner la valeur de la fréquence de l'évènement $[T_n = 5]$ obtenue pour ces 1000 simulations de T_n .

(d) On rajoute à la suite des instruction suivantes :

```
x=M(:,1)
f=M(:,2)/1000
scf(0), clf()
bar(x,f)
scf(1), clf()
bar(1:n,loi_de_T(n,p))
```

On obtient les 2 graphiques suivants :





Commenter les graphiques obtenus.

Exercice 3.

Dans cet exercice, la lettre n désigne un entier naturel.

On dispose d'une urne contenant au départ n boules blanches et (n+2) boules noires. On dispose également d'une réserve infinie de boules blanches et de boules noires.

Pour tout entier naturel j, on dit que l'urne est dans l'état j lorsqu'elle contient j boules blanches et (j+2) boules noires. Au départ, l'urne est donc dans l'état n.

On réalise une succession d'épreuves, chaque épreuve se déroulant selon le protocole suivant : Pour tout entier naturel j non nul, si l'urne est dans l'état j, on extrait une boule au hasard de l'urne.

- Si l'on obtient une boule blanche, alors cette boule n'est pas remise dans l'urne et on enlève de plus une boule noire de l'urne, l'urne est alors dans l'état (j-1).
- Si l'on obtient une boule noire, alors cette boule est remise dans l'urne et on remet en plus une boule blanche et une boule noire dans l'urne, l'urne est alors dans l'état (j + 1).
- 1. Dans cette question, on suppose que n=1 (l'urne contient donc une boule blanche et 3 boules noires) et on note X_1 la variable aléatoire égale au nombre de boules blanches encore présentes dans l'urne après la première épreuve et X_2 la variable aléatoire égale au nombre de boules blanches encore présentes dans l'urne après la deuxième épreuve.

On admet que X_1 et X_2 sont définies sur un certain espace probabilisé (Ω, \mathcal{A}, P) que l'on ne cherchera pas à déterminer.

- (a) Donner la loi de X_1 .
- (b) Déterminer la loi de X_2 .
- (c) Simulation informatique de l'expérience aléatoire décrite ci-dessus.

On rappelle que floor(rand()*n renvoie au hasard un entier compris entre 0 et n-1. Compléter le programme suivant pour qu'il simule l'expérience aléatoire décrite dans cet exercice et pour qu'il affiche les valeurs des variables aléatoires X_1 et X_2 .

```
tirage=floor(rand()*4)
    if tirage==0 then
        X1 = \dots
   else
        X1=.....
   end
   if (X1==0) then
       X2=.....
    else
         tirage=floor(rand()*6)
                   if tirage<=1 then
                        X2=.....
                    else X2=.....
                   end
      end
disp(X2,X1)
```

On revient au cas général (n est donc un entier naturel quelconque supérieur ou égal à 1) et on décide que les tirages s'arrêtent dès que l'urne ne contient plus de boules blanches.

Pour tout j de \mathbb{N} , on note alors E_j l'évènement : « l'urne est dans l'état j initialement et les tirages s'arrêtent au bout d'un temps fini ». On pose $e_j = P(E_j)$ et l'on a bien sûr $e_0 = 1$.

2. Montrer, en considérant les deux résultats possibles du premier tirage (c'est-à-dire au début du jeu lorsque l'urne est dans l'état n) que :

$$\forall n \in \mathbb{N}^*, \quad e_n = \frac{n}{2n+2}e_{n-1} + \frac{n+2}{2n+2}e_{n+1}.$$

- 3. (a) Montrer par récurrence que : $\forall n \in \mathbb{N}, e_n \geq e_{n+1}$.
 - (b) En déduire que la suite (e_n) est convergente.

On admet pour la suite que $\lim_{n\to+\infty} e_n = 0$.

- 4. Pour tout entier naturel n, on pose $u_n = (n+1)e_n$.
 - (a) Pour tout entier naturel n de \mathbb{N}^* , écrire u_{n+1} en fonction de u_n et u_{n-1} .
 - (b) En déduire l'expression de u_n en fonction de n et e_1 .
 - (c) Montrer enfin que l'on a : $\forall n \in \mathbb{N}, \ e_n = (2e_1 1) \frac{n}{n+1} + \frac{1}{n+1}$.

Déterminer la valeur de e_1 , puis en déduire, pour tout entier naturel n, l'expression de e_n en fonction de n