David Michel
david.michel@ens-rennes.fr
perso.eleves.ens-rennes.fr/people/David.Michel

ECE 1
Lycée Saint-Vincent
2016-2017

TP9 : Introduction aux probabilités – Simulation – Correction

$\boxed{\text{Q.1}}$ La commande **rand()** simule une variable aléatoire suivant la loi uniforme sur $[0,1]$ donc $\mathbb{P}(\textbf{rand()} \in [a,b]) = b - a$ pour tous $a, b \in [0,1]$ (et on peut modifier les bornes de l'intervalle et les changer en bornes fermées sans changer la probabilité).

Ici, la condition est réalisée lorsque **rand()**$<$**p**, soit avec une probabilité

$$\mathbb{P}(\textbf{rand()} < \mathbf{p}) = \mathbb{P}(\textbf{rand()} \in [0, \mathbf{p}]) = \mathbf{p}.$$

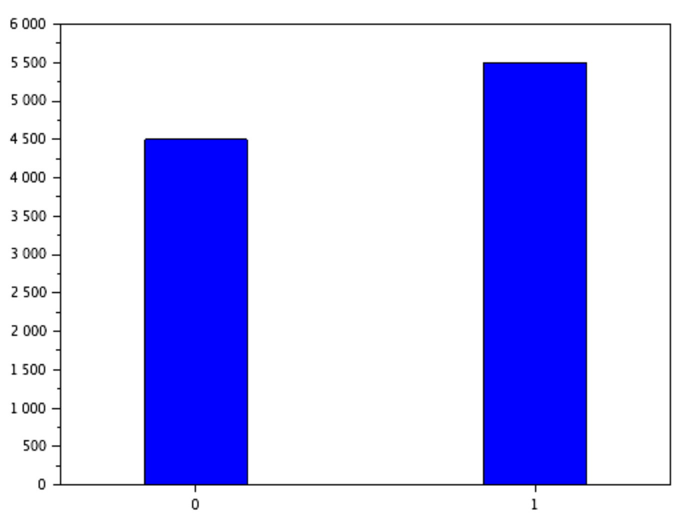
Ainsi, ce programme affiche 0 avec probabilité **p** et 1 avec probabilité $1 - \mathbf{p}$. On a donc simulé une variable aléatoire suivant la loi de Bernoulli de paramètre **p**.

Pour le vérifier expérimentalement, on répète un grand nombre de fois l'expérience et on calcule les fréquences d'apparition de 0 et de 1. En effet, ces fréquences doivent tendre, lorsque le nombre de répétition de l'expérience tend vers l'infini, vers les probabilités correspondantes. Plutôt que d'afficher 0 et 1, on va plutôt incrémenter le compteur correspondant.

```
p=0.45;
compteur0=0;
compteur1=0;
for k=1:10^4
   if rand()<p then
      compteur0=compteur0+1;
   else
      compteur1=compteur1+1;
   end
end
disp("La fréquence d apparition de 0 est "+string(compteur0/10^4)+
        " et celle de 1 est "+string(compteur1/10^4))
```

On peut obtenir : Ces fréquences confirment donc le calcul de probabilité effectué plus haut.

```
La fréquence d apparition de 0 est 0.4499 et celle de 1
     est 0.5501
```



On peut aussi raisonner graphiquement. On peut en effet représenter dans un diagramme en bâton les résultats obtenus. Après le code précédent, on ajoute, en lieu et place de **disp(...)** :

bar([0,1],[compteur0,compteur1],0.3)

On obtient la figure ci-contre. On peut constater que les proportions sont bien celles attendues.

$\boxed{\text{Q.2}}$ La variable de sortie, $\mathbf{Y}$ vaut initialement 1 puis ne peut qu'augmenter. Ainsi, l'événement $\{\mathbf{Y} = 0\}$ est vide. Ainsi, $\mathbb{P}(\mathbf{Y} = 0) = \mathbb{P}(\emptyset) = 0$.

On obtient $\mathbf{Y} = 1$ si et seulement si on ne passe pas dans la boucle, c'est-à-dire que la condition **rand()**$<$**q** est fausse. Ainsi $\mathbb{P}(\mathbf{Y{=}1}) = \mathbb{P}(\mathbf{rand()} \geq \mathbf{q})$. Or **rand()** suit la loi uniforme sur [0,1] donc

$$\mathbb{P}(\mathbf{Y{=}1}) = \mathbb{P}(\mathbf{rand()} > \mathbf{q}) = 1 - \mathbf{q} = \mathbf{p}$$

De même, on obtient $\mathbf{Y} = 2$ si et seulement si on passe une première fois dans la boucle, puis on n'y passe pas lors du deuxième test. Ici, il est important de supposer que les appels successifs à la commande **rand()** de Scilab produisent des réalisations de variables indépendantes. Ainsi,

$$\mathbb{P}(\mathbf{Y} = 2) = \mathbb{P}(\underbrace{\mathbf{rand()} < \mathbf{q}}_{\text{passer dans la boucle}}) \times \mathbb{P}(\underbrace{\mathbf{rand()} > \mathbf{q}}_{\text{ne pas passer dans la boucle}}) = \mathbf{q}(1 - \mathbf{q}) = (1 - \mathbf{p})\mathbf{p}$$

Avec le même raisonnement, on obtient $\mathbf{Y} = n$ si et seulement si on passe $n - 1$ fois dans la boucle et qu'au n'y passe pas lors du n-ième test. Grâce à l'indépendance des appels à **rand()**, on obtient :

$$\mathbb{P}(\mathbf{Y} = n) = \mathbf{q}^{n-1}(1 - \mathbf{q}) = (1 - \mathbf{p})^{n-1}\mathbf{p}.$$

On dit que $\mathbf{Y}$ suit la loi géométrique de paramètre $\mathbf{p}$. Cette loi est aussi appelée loi du premier succès. Le programme étudié peut être vu comme la modélisation de l'expérience suivante : on lance une pièce truquée (probabilité $\mathbf{p}$ d'avoir pile) jusqu'à obtenir un pile, et $\mathbf{Y}$ compte le nombre de lancers nécessaires.

Enfin, il est possible que le test de la boucle **while** soit positif à chaque fois, indéfiniment, et donc que la fonction ne s'arrête jamais, ce qui est problématique. . .Mais quelle est la probabilité que cela arrive ? Si c'est le cas, $\mathbf{Y}$ prend toutes les valeurs de $\mathbb{N}^*$. En fait,

$$\mathbb{P}(\text{"boucle infinie"}) = \mathbb{P}(\forall n \in \mathbb{N}^*,\ \mathbf{Y} \geq n) = \mathbb{P}\left(\bigcap_{n \in \mathbb{N}^*} \underbrace{\{\mathbf{Y} \geq n\}}_{=A_n}\right).$$

La suite des événements $(A_n)_{n \in \mathbb{N}^*}$ est décroissante (*ie.* $A_{n+1} \subset A_n$) donc

$$\mathbb{P}(\text{"boucle infinie"}) = \mathbb{P}\left(\bigcap_{n \in \mathbb{N}^*} A_n\right) = \lim_{n \to +\infty} \mathbb{P}(A_n)$$

et le même raisonnement que précédemment montre que $\mathbb{P}(A_n) = \mathbf{q}^n$. Ainsi,

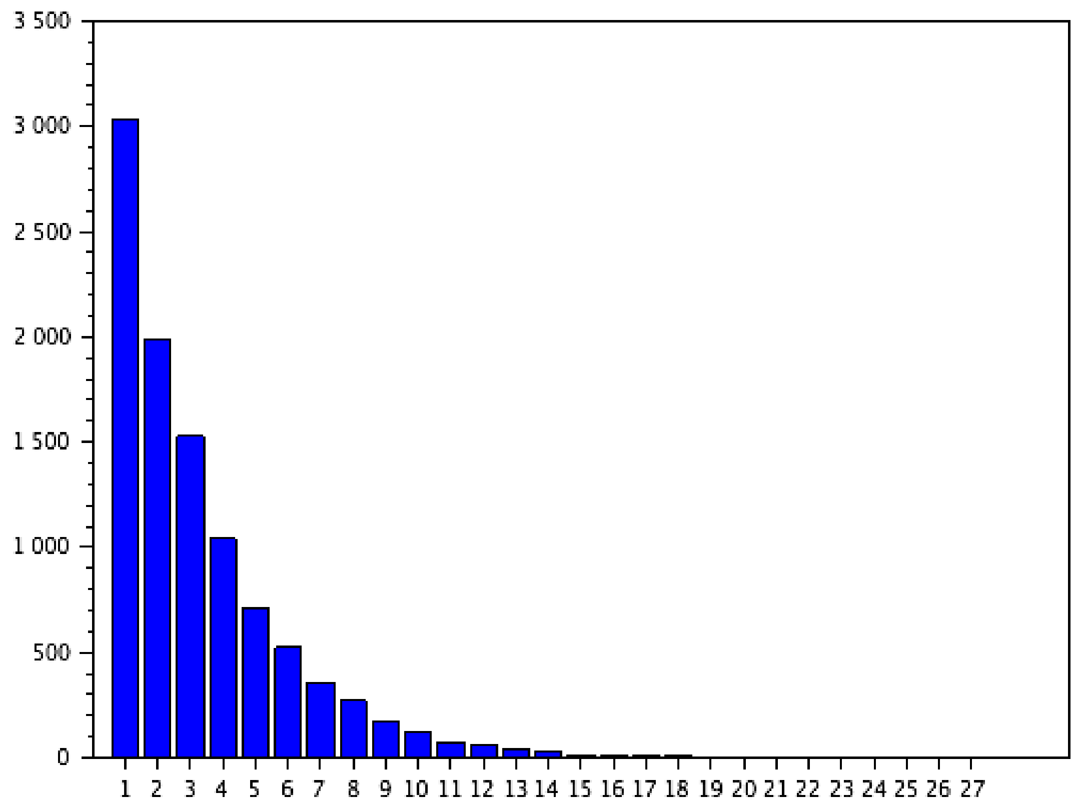
$$\mathbb{P}(\text{"boucle infinie"}) = 0$$

puisque $0 < \mathbf{q} < 1$.

Cherchons à visualiser la loi de probabilité. Ici, l'ensemble des issues possibles pour $\mathbf{Y}$ est infini (c'est $\mathbb{N}^*$) donc on ne pourra pas tout représenter. On ne va donc pouvoir visualiser qu'une partie de la loi. Pour ceci, on va répéter un grand nombre de fois l'expérience aléatoire et stocker la valeur de $\mathbf{Y}$ obtenue dans un vecteur ligne. Ensuite, on va compter le nombre d'occurence de chacune des réalisations, que l'on représentera ensuite dans un diagramme en bâtons. Pour le comptage, on crée un vecteur ligne qui va correspondre à chaque issue de l'expérience : on va donc considérer tous les entiers de 1 à **max(Y)**. Ensuite, on parcourt $\mathbf{Y}$ puis on ajoute 1 dans le vecteur **Occurrences** à la case correspondante au nombre $\mathbf{y} \in \mathbf{Y}$ que l'on est en train de considérer.

```
Y=[];
p=0.3;
for k=1:10^4
   Y=[Y,Attends(p)];
end
Occurrences=zeros(1,max(Y));
for y=Y
   Occurrences(y)=Occurrences(y)+1;
end
bar(1:max(Y),Occurrences)
```

On obtient, avec **p=0.3** :



Les valeurs obtenues et la vitesse de décroissance (géométrique) tendent à confirmer les résultats théoriques obtenus. Bien sûr, on aurait pu adopter le cheminement inverse et commencer par effectuer ce test numérique pour conjecturer certaines propriétés sur la loi.

$\boxed{\text{Q.3}}$ a. Pour simuler des tirages jusqu'à obtenir deux boules identiques, il faut commencer par calculer la probabilité d'obtenir un tel tirage. Pour cela, on peut déterminer le nombre de tirage avec deux boules identiques, ainsi que le nombre de tirages total.

Numérotons les boules d'un lot de 1 à n. Ainsi, dans l'urne se trouvent deux boules numérotées 1, deux boules numérotées 2,..., deux boules numérotées n.

– Il y a donc n tirages avec deux boules identiques : $(1,1),(2,2),\ldots,(n,n)$.

– Déterminons le nombre total de tirages. Il y a d'abord les tirages identiques :

$(1,1)$	$(2,2)$	$\cdots$	$\cdots$	$\cdots$	(n,n)

Il y a ensuite les tirages contenant une boule numérotée 1 (et qu'on n'a pas encore comptés) :

$(1,1)$	$(2,2)$	$\cdots$	$\cdots$	$\cdots$	(n,n)
$(1,2)$	$(1,3)$	$\cdots$	$\cdots$	$(1,n)$	

On continue avec les tirages contenant une boule numérotée 2 (et qu'on n'a pas encore comptés) :

$(1,1)$	$(2,2)$	$\cdots$	$\cdots$	$\cdots$	(n,n)
$(1,2)$	$(1,3)$	$\cdots$	$\cdots$	$(1,n)$	
$(2,3)$	$(2,4)$	$\cdots$	$(2,n)$		

et ainsi de suite jusqu'à obtenir tous les cas :

$(1,1)$	$(2,2)$	$\cdots$	$\cdots$	$\cdots$	(n,n)
$(1,2)$	$(1,3)$	$\cdots$	$\cdots$	$(1,n)$	
$(2,3)$	$(2,4)$	$\cdots$	$(2,n)$		
$\vdots$	$\vdots$	$\vdots$			
$(n-2,n-1)$	$(n-2,n)$				
$(n-1,n)$					

En comptant le nombre de cases ligne par ligne (de bas en haut), on obtient qu'il y a $\displaystyle\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

Ainsi,

$$\mathbb{P}(\text{"tirages identiques"}) = \frac{n}{\frac{n(n+1)}{2}} = \frac{2}{n+1}.$$

Passons à la simulation. Nous devons simuler des tirages tant que cet événement n'est pas réalisé. Ainsi, après un premier tirage, il faut en effectuer un autre avec probabilité $1-\frac{2}{n+1}$. On peut donc simuler l'expérience ainsi :

```
function Y=T(m)
   Y=1
   while rand()<1-2/(n+1)
      Y=Y+1
   end
endfunction
```

La variable **Y** est donc bien un compteur qui s'incrémente avec la même probabilité que celle du tirage de deux boules distinctes.

b. On calcule ici la somme des T_k :

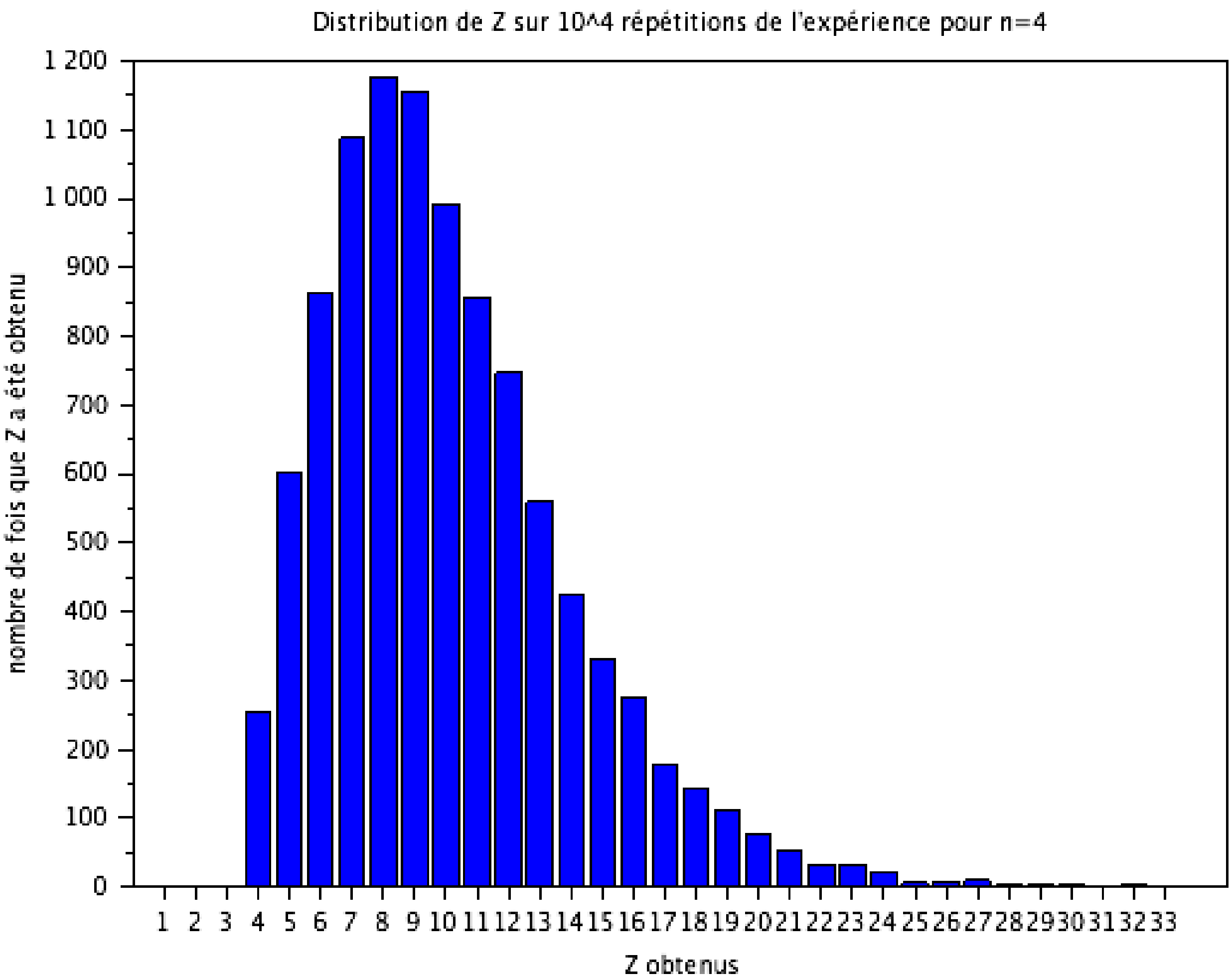
```
n=input("entrez n ")
   Z=0;
   for k=1:n
      Z=Z+T(k);
   end
disp(Z)
```

On peut maintenant, comme aux questions précédentes, représenter graphiquement les résultats obtenus.

On fixe $n = 4$. On veut visualiser la distribution de probabilités régissant le nombre Z de tirages qu'il faut effectuer pour vider l'urne.

```
n=4;
Issues=[];
for i=1:10^4
   Z=0;
   for k=1:n
      Z=Z+T(k);
   end
   Issues=[Issues,Z];
end
Compteur=zeros(1,max(Issues));
for z=Issues
   Compteur(z)=Compteur(z)+1;
end
bar(1:max(Issues),Compteur)
xtitle("Distribution de Z sur 10^4 répétitions de l"'expérience pour n=4",
"Z obtenus","nombre de fois que Z a été obtenu")
```

On peut obtenir :


Distribution de Z sur 10^4 répétitions de l'expérience pour n=4
nombre de fois que Z a été obtenu
Z obtenus
1 200
1 100
1 000
900
800
700
600
500
400
300
200
100
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33