TP 2

On rappelle que la commande `help` X permet d'obtenir une aide détaillée sur la commande X.

Exercice 1 Processus de branchement

On s'intéresse à la simulation d'un processus de branchement. On rappelle qu'on peut le définir par récurrence à partir du nombre initial d'individus Z_0 comme

$$Z_{n+1} = \sum_{i=1}^{Z_n} \xi_{n,i}$$

à partir de v.a. i.i.d. $(\xi_{n,i})$ à valeurs dans $\mathbf{N}$, dont on note μ la loi, appelée loi de branchement.

1. (a) Comprendre et exécuter le code suivant qui simule 1000 générations d'un processus, pour la loi de branchement $\mu(0) = 1/4$, $\mu(1) = 1/2$, $\mu(2) = 1/4$.

```
mu=[0.25,0.5,0.25];
T=1000; // nombre de générations simulées
Z=zeros(1,T+1);
Z(1)=30; // population initiale
for t=1:T
    u=rand(1,Z(t));
    M=zeros(1,Z(t));
    for k=1:size(mu,2)
        M(1:Z(t))=M(1:Z(t)) + (u>=sum(mu(1:k)));
    end
Z(t+1)=sum(M);
end
```

Correction : Le code produit récursivement, pour chaque génération, pour chacun des $Z(t)$ individus vivants à la génération t, un nombre d'enfants distribué suivant la loi μ, qu'il range dans le vecteur M. Pour cela, il tire un vecteur `u` *contenant $Z(t)$ v.a. indépendantes de loi uniforme sur $[0,1]$, $U_1, \ldots, U_{Z(t)}$. L'individu z aura alors k enfants si $\sum_{i=1}^{k} \mu(i-1) \leqslant U_z \leqslant \sum_{i=1}^{k+1} \mu(i-1)$ (la première somme est nulle si $k = 0$). Soit M_z le nombre d'enfants de l'individu z obtenu ainsi. On a bien $\mathbf{P}(M_z = k) = \mathbf{P}(U_z \in [\sum_{i=1}^{k} \mu(i-1), \sum_{i=1}^{k+1} \mu(i-1)]) = \mu(k)$. Dans la boucle*

```
for k=1:size(mu,2)
        M(1:Z(t))=M(1:Z(t)) + (u>=sum(mu(1:k)));
end
```

le terme `u>=sum(mu(1:k))` permet d'ajouter à chaque coordonnée du vecteur `M` un k-ième enfant tant que la coordonnée correspondante de `u` n'a pas dépassé $\sum_{i=1}^{k} \mu(i-1)$.

(b) Tracer le graphe du nombre d'individus en fonction du temps. Vous pouvez effectuer plusieurs simulations pour voir si le comportement observé est le même.

Correction : `plot2d((1:T+1),Z)`.

(c) Que prévoit la théorie ? Est-ce cohérent avec les observations de la question précédente ?

Correction : μ est critique (son espérance vaut 1). La théorie prévoit qu'il y a extinction presque sûre. Si on fait beaucoup d'essais, on finit par voir des cas où elle n'a pas lieu avant la millième génération. C'est lié au fait que dans le cas critique l'espérance du nombre total d'enfants (qui est fini p.s.) est, elle, infinie. On peut en fait montrer que pour un processus de branchement critique de variance finie σ^2 partant d'un individu, $\mathbf{P}(Z_n > 0) \sim \frac{2}{\sigma^2 n}$.

2. Comparer graphiquement les comportements pour les lois initiales suivantes et vérifier si cela est conforme à ce que prédit la théorie (attention, commencer par simuler un petit nombre de générations car la taille de la population peut parfois croître exponentiellement vite)

(a) $\mu(0) = 0.25$, $\mu(1) = 0.55$, $\mu(2) = 0.2$

(b) $\mu(0) = 0.25$, $\mu(1) = 0.45$, $\mu(2) = 0.3$

(c) $\mu(0) = 0.9$, $\mu(10) = 0.1$

(d) $\mu(0) = 0.9$, $\mu(11) = 0.1$

Correction : Pour le code, il faut juste changer la première ligne (définition de μ).

(a) `mu=[0.25,0.55,0.2]` *La loi devient sous-critique, le processus s'éteint presque sûrement exponentiellement vite.*

(b) `mu=[0.25,0.45,0.3]` *La loi devient surcritique, le processus a une probabilité strictement positive de survivre, et dans ce cas il explose exponentiellement vite.*

(c) `mu=[0.9,zeros(1,9),.1]` *Le processus est à nouveau critique, on se retrouve dans le premier cas de figure.*

(d) `mu=[0.9,zeros(1,10),.1]` *Le processus est surcritique, cf (b).*

3. On reprend la loi de branchement de la question 2b en supposant que $Z_0 = 1$.

   (a) Calculer la probabilité d'extinction théorique pour ce processus.

   Correction : On cherche le plus petit point fixe de la fonction génératrice $\mu(0)+\mu(1)s+\mu(2)s^2$. Il s'agit de $\mu(0)/\mu(2) \simeq .833$.

   (b) Vérifier expérimentalement que la probabilité d'extinction observée est la même (simuler un grand nombre de populations, et regarder quelle proportion s'éteint).

   Correction :

```
mu=[0.25,0.45,0.3];
n=1000; //nombre de populations simulées
T=50; //nombre de générations simulées
S=0;
for i=(1:n)
Z=1;
for t=1:T
if Z==0
    else
u=rand(1,Z);
     M=zeros(1,Z);
    for k=1:size(mu,2)
        M(1:Z)=M(1:Z) + (u>=sum(mu(1:k)));
     end
Z=sum(M);
end
end
S=S+(Z==0);
end
S=S/n
```