1 Objectifs

Dans ce T.D., nous commencerons par étudier la notion de **simulation** d'une expérience aléatoire, et plus généralement d'une variable aléatoire, à travers l'exemple du schéma de Bernoulli et de la loi géométrique.

Nous verrons ensuite quelques illustrations de l'absence d'espérance en simulant une expérience aléatoire.

2 Simulation de la loi géométrique et estimation de son espérance

2.1 Simulation d'un schéma de Bernoulli

On rappelle que :

- ☞ chaque exécution de la fonction `rand()` fournit un réel de l'intervalle $[0;1[$, suivant une loi uniforme ;
- ☞ les appels successifs fournissent des valeurs indépendantes les unes des autres ;
- ☞ pour tout réel p de $[0;1]$, le test `rand()<=p` a une probabilité p d'être vrai (et $1-p$ d'être faux ...).

Simuler une loi donnée, c'est écrire une fonction dont le résultat est un nombre suivant la loi donnée.

1. Que simule la fonction suivante ?

```
function X=ber(p)
   if rand()<=p then
   X=1
   else X=0
   end
endfunction
```

2. Compléter la fonction suivante afin qu'elle simule la loi géométrique.

```
function X=geo_1(p)
    X=1
    while ....................
        X=X+1
    end
endfunction
```

3. Quel est le nombre moyen de boucles réalisées par cette algorithme lorsque $p = 1/2$? Et lorsque $p = 1/100$?

2.2 Simulation directe

Voici une méthode permettant de simuler la loi géométrique sans recourir à une boucle.

4. Soit p un réel de $[0;1[$ et U une variable de loi uniforme sur $[0;1[$. Soit $X = \left\lfloor \dfrac{\ln(U)}{\ln(1-p)} \right\rfloor + 1$, où $\lfloor \alpha \rfloor$ désigne la partie entière du réel α.

a) Déterminer $X(\Omega)$.

b) Montrer que X suit une loi géométrique de paramètre p.

5. Sachant que la partie entière s'écrit `floor(.)`, proposer une fonction `geo_2` simulant la loi géométrique sans boucle.

2.3 Utilisation de la fonction `grand`

La fonction `grand` est une fonction généralisant la fonction `rand()`.
`rand()` simule une variable aléatoire suivant la loi uniforme sur $[0;1[$, tandis que `grand` peut simuler toutes les lois usuelles à notre programme [(1)].
En voici la syntaxe générale :

```
grand(n_ligne,n_colonne,"nom_loi",paramètre(s))
```

fournit une matrice à `n_ligne` lignes et `m_colonne` colonnes dont chaque coefficient est le résultat d'une simulation, indépendante des autres.
Et pour les lois usuelles :

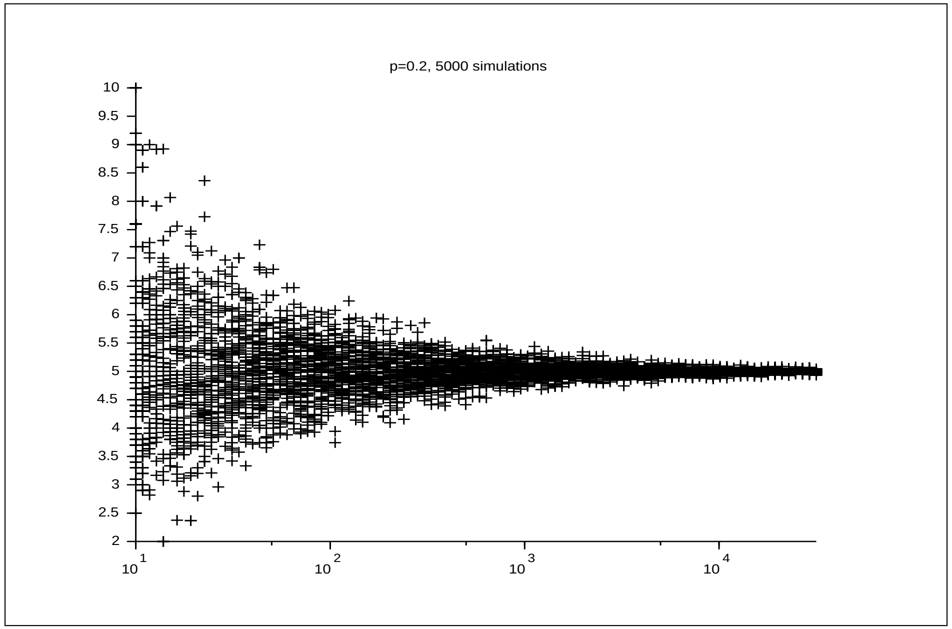
`grand(m,n,"uin",a,b)`	$\mathcal{U}([\![a;b]\!])$
`grand(m,n,"bin",N,p)`	$\mathcal{B}(N,p)$
`grand(m,n,"geom",p)`	$\mathcal{G}(p)$
`grand(m,n,"poi",lambda)`	$\mathcal{P}(\lambda)$
`grand(m,n,"unf",a,b)`	$\mathcal{U}([a;b[)$
`grand(m,n,"exp",1/lambda)`	$\mathcal{E}(\lambda)$
`grand(m,n,"gam",b,nu)`	$\Gamma(b,\nu)$
`grand(m,n,"nor",m,s)`	$\mathcal{N}(m,s^2)$

(1). On peut par ailleurs montrer que toute loi discrète ou à densité peut être simulée à l'aide de la loi uniforme sur $[0;1[$, ce qui explique que la plupart des langages informatiques ne proposent qu'une fonction `rand`. Mais Scilab est bien plus riche

Outre le fait d'éviter d'écrire un algorithme de simulation, cette fonction présente l'intérêt d'être plus rapide que celles écrites par l'utilisateur, ce qui permet de simuler instantanément des dizaines de milliers de valeurs.

3 Illustration de l'existence de l'espérance

6. Interprêter le graphique suivant, obtenu à l'aide de l'algorithme ci-dessous en exécutant `convergence_g(.2,50)` :




```
function convergence_g(p,n)
    x=logspace(1,4.5,100)
    for i=1:n
    y=[]
       for k=x
       y=[y mean(grand(1,k,"geom",p))]
    end
    plot2d("ln",x,y,-1)
    xtitle("p="+string(p)+", "+string(100*i)+" simulations")
    end
endfunction
```

On consultera l'aide en ligne pour comprendre le rôle de `logspace` et de `''ln''`.
On appelle **méthodes de Monte-Carlo** les méthodes consistant à simuler une expérience aléatoire pour estimer certaines grandeurs liées à l'expérience.
Ici, la simulation de la loi géométrique de paramètre $0,2$ permet que conjecturer que son espérance est 5.

4 Illustration de l'absence d'espérance

Une urne contient initialement une boule rouge et une boule bleue.
On procède à une succession de tirages suivant le processus :
• si la boule tirée est bleue, on la remet accompagnée d'une autre boule bleue dans l'urne ;
• si la boule est rouge, on arrête les tirages.
On note X le nombre de tirages effectués.

7. Déterminer la loi de X.

8. Montrer que, si U suit une loi uniforme sur $[\,0\,;\,1\,[$, alors $\mathrm{V} = \left\lfloor \dfrac{1}{\mathrm{U}} \right\rfloor$ suit la loi de X.

9. Compléter

`floor(1 ./grand(1,n,........))`

afin de simuler une série de n variables X.

10. Transformer la fonction `convergence_g` de 2.5 en une nouvelle fonction `convergence_t` réalisant le même type d'illustration.

11. Exécuter `convergence_t(100)`.

12. Afin d'éviter l'écrasement vertical des données, remplacer le paramètre `''ln''` de la fonction `plot2d` pour avoir aussi une échelle logarithmique en ordonnées, puis exécuter à nouveau `convergence_t(100)`.

13. Quelle interprétation donner à cette illustration ?