# SIMULATION DES LOIS USUELLES

Le but de ce TP est de simuler des lois usuelles en n'utilisant que `rand()` et les procédés de transfert. Chaque partie de programmation est précédée de calculs qui on déjà été traités en cours mais que vous devez refaire.

- La fonction `rand()` retourne un nombre réel au hasard choisi en suivant la loi $\mathcal{U}([0;\,1[)$.
- La fonction `rand(1,n)` rempli une matrice ligne de longueur $n$ en suivant la règle précédente.

**Consignes** On n'utilisera qu'un seul fichier, dans lequel vous écrirez toutes les fonctions les unes à la suites des autres. Vous testerez chaque fonction avec la commande `histplot` (voir l'exemple).

## I Simulation des lois uniformes sur un intervalle réel

**Approche mathématique** Soit $\alpha$ un réel strictement positif et $\beta$ un réel Soit $X \hookrightarrow \mathcal{U}([0;\,1[)$ on pose $Y = \alpha X + \beta$
**Question :** Montrer que $Y \hookrightarrow \mathcal{U}([\beta;\,\beta+\alpha[)$.

**Implémentation** Compléter le programme suivant pour qu'il simule $n$ fois une variable aléatoire suivant la loi $\mathcal{U}([a;\,b[)$

```
function T=uniforme(n,a,b)
        T= ........ *rand(1,n)+......
endfunction

histplot(10,uniforme(10000,1,3))
```

## II Simulation des lois uniformes sur un intervalle d'entiers

**Approche mathématique** Soit $p$ un entier naturel non nul et $X \hookrightarrow \mathcal{U}([1;\,p+1[)$, on pose $Y = \lfloor X \rfloor$.
**Question :** Montrer que $Y \hookrightarrow \mathcal{U}([\![1,\,p]\!])$.

**Implémentation** En utilisant la fonction `uniforme(n,a,b)`,écrire une fonction `unifentier(n,p)` qui renvoie une matrice ligne à $n$ colonnes donc chaque terme est choisi selon loi $\mathcal{U}([\![1,\,p]\!])$.

## III Simulation de la loi exponentielle

**Approche mathématique** Soit $X \hookrightarrow \mathcal{U}([0;\,1[)$ et soit $\lambda > 0$. On pose $Y = -\dfrac{1}{\lambda}\ln(1-X)$.
**Question :** Montrer que $Y \hookrightarrow \mathcal{E}(\lambda)$.

**Implémentation** En utilisant la fonction `rand()` écrire une fonction `exponentielle(n,L)` qui renvoie une matrice ligne à $n$ colonnes dont chaque terme est choisi selon loi $\mathcal{E}(L)$.

## IV Simulations de la loi géométrique

### IV.1 Par transfert

**Approche mathématique** Pour tout nombre réel $x$, on note $\lfloor x \rfloor$ la partie entière de $x$.
Soit $X$ la variable aléatoire suivant la loi exponentielle de paramètre $\lambda$ ($\lambda > 0$).
On pose $Y = \lfloor X \rfloor$, $Y$ est donc la partie entière de $X$.

1. Montrer que $Y$ prend ses valeurs dans $\mathbb{N}$.
2. Pour tout $k$ de $\mathbb{N}^*$, calculer $P(Y = k-1)$.

3. En déduire que la variable aléatoire $Y+1$ suit une loi géométrique dont on donnera le paramètre.

**Implémentation** En utilisant les questions précédentes et en utilisant `rand()` écrire une écrire une fonction `geometrique(n,p)` qui renvoie une matrice ligne à $n$ colonnes dont chaque terme est choisi selon loi $\mathcal{G}(p)$.

### IV.2 D'après la définition

On rappelle que l'on obtient une loi géométrique quand on étude le rang de premier succès lors de la répétition d'expériences de Bernouilli indépendantes. Compléter le programme suivant pour simuler la loi géométrique.

```
function T=geometrique2(n,p)
T=zeros(n)
//i cases du tableau
for i=1:n
        k= .....//numero du lancer

        //
        while rand()......
        .......
        end

        T(i)=........
        end
endfunction
```

## V Simulations de la loi binomiale

On rappelle que l'on obtient une loi binomiale quand en comptant le nombre de succès quand on réalise d'expérience de Bernouilli indépendante.

Compléter le programme suivant pour simuler la loi binomiale.

```
function T=binomiale(n,L,p)
//n nombre d'expérience
//p reel entre 0 1
// L nombre de lancers dans chaque expérience

Lancers=rand(n,L) // une matrice

T=sum(.....,'c')' //cf help sum

endfunction
```