TP 6 - Simulations de tirages

Exercice 1 (*Représentation graphiques de fonctions*)

On s'intéresse aux fonctions définies pour $n \in \mathbb{N}$, pour $x \geqslant 0$, par : $f_n(x) = \frac{e^{-x}}{1+x^n}$.

1. Définir une fonction Scilab `function y=f(n,x)` pour modéliser ces fonctions.
2. Représenter graphiquement les premières fonctions. Que constate-t-on à la limite $n \to \infty$?

Pour $n \in \mathbb{N}$, et $x \geqslant 0$, on définit : $g_n(x) = \frac{x^n}{n!} \cdot e^{-x}$.

3. Représenter les premières fonctions g_n sur $[0;+\infty[$.
4. Tracer la courbe d'équation $y = \frac{1}{\sqrt{2\pi x}}$.
5. Vérifier empiriquement que pour n entier grand, on a : $n! \sim \frac{n^n}{e^n} \cdot \sqrt{2\pi n}$. *(Formule de Stirling)*

Exercice 2 (*Tirage avec remise*)

On a une urne remplie de N boules :

- 1 bleue
- $N-1$ vertes

On effectue des tirages **sans remise** dans l'urne, jusqu'à l'obtention de la boule bleue.
On note X la variable aléatoire qui prend pour valeur le nombre de tirages nécessaires à l'obtention de la boule bleue.

1. On nous demande de compléter le script ci-dessous pour simuler un échantillon de X.
2. Définir une fonction `x=simulerX(n)` qui simule une instance de X.
3. Adapter le script pour utiliser la commande `histplot`.
4. Adapter le script pour simuler avec un nombre différent de boules bleues.

aCompleter.sci

```
N = input('Donner un entier naturel non nul')
S = zeros(1,N) // permettra de calculer les fréquences empiriques
for k = 1:10000
    i = 1       // i modélise le rang du tirage
    M = N       // M modélise le nombre de boules restant dans l'urne
    while (_________________)
        i = i+1
        M = _______________
    end
    S(i) = S(i)+1
end
// bar : représentation graphique en histogramme
bar(S/10000)
```

Exercice 3 (*Obtention d'une boule de chaque couleur*)

On fait une succession illimitée de tirages « pile ou face » avec pour probabilités $p,q \in]0;1[$.
On tire jusqu'à avoir obtenu au moins un « pile » et un « face ».
On admet que, presque-sûrement, on finit par y arriver.
On note T et U les *v.a.* modélisant :

- le nombre T de tirages nécessaires,
- le nombre U de « pile » obtenu.
- le nombre V de « face » obtenu.

(On utilisera le fichier `pileFaceTrous.sci`*)*

1. Comment s'écrit T en fonction de U et V ?
2. Compléter la définition de la fonction `[u,v]=simulerUV(p)`.
3. Pour quelle valeur de $p \in]0;1[$, le temps T est-il minimisé ?
4. Les variables aléatoires U et V sont-elles indépendantes ?
5. Comment évoluent les variables U et V quand le paramètre p varie ?

pileFaceTrous.sci

```
function [u,v]=simulerUV(p)
  u=0 // nombre de Pile obtenu
  v=0 // nombre de Face obtenu
  while(___)
    if rand()<p // si on tombe sur pile
      u=___
    else
      v=___
    end
  end
end

p=.5
N=10^1 // taille de l'échantillon
echantillonU=[]
echantillonV=[]
for k=1:N
  [u,v]=simulerUV(p)
  echantillonU=[echantillonU,u]
  echantillonV=[echantillonV,v]
end
echantillonT=____
```