Python : Simulations de variables aléatoires

Dans la suite on suppose avec importé la bibliothèque numpy.random à l'aide de :
import numpy.random as rd

1) Simuler les lois usuelles

Modèle général des commandes :
rd.binomial(n,p) : *Renvoie une simulation d'une binomiale de paramètres n et p.*
rd.binomial(n,n,100) : *Renvoie 100 simulations indépendantes dans une matrice ligne de binomiales de paramètres n et p*
rd.binomial(15,0.3,[N,M]) : *Renvoie une matrice de taille NxM remplie de simulations indépendantes de binomiales de paramètres n et p.*

rd.randint(a,b+1) : *Renvoie un nombre entier aléatoire entre a et b*
rd.poisson(lambda) : *Renvoie une simulation d'une Poisson de paramètre* lambda
rd.geometric(p) : *Renvoie une simulation d'une Géométrique de paramètre* p

2) Simuler les lois usuelles « à la main »

Voici des programmes permettant de simuler les lois usuelles sans appeler directement les commandes associées. Ces programmes sont à comprendre car ils permettent, en les adaptant un peu, de simuler des lois non usuelles mais similaires aux lois usuelles.

a) Loi géométrique de paramètre p :

```
X=1
while rd.random()>p :            #tant qu'on des échecs
    X=X+1
print(X)
```

b) Loi binomiale de paramètres n et p

```
X=0
for k in range(1,n+1):           # on répète n expériences
    If rd.random()<p:            #à chaque succès, on augmente X de 1
        X=X+1
print(X)
```

3) Obtenir des valeurs approchées d'espérance (Monte Carlo)

L'espérance d'une variable aléatoire correspond au résultat moyen espéré. En d'autres termes, cela signifie que, si l'on simule un grand nombre de fois (disons 100 ou 1000) une variable et qu'on calcule la moyenne des simulations, on va obtenir une valeur proche de l'espérance. Ce procédé se nomme la méthode de Monte Carlo.
Nous verrons plus tard qu'il est sous tendu théoriquement par le résultat fondamental des probabilités : la loi des grands nombres.

Méthode :

On suppose disposer d'une fonction simulX() renvoyant une simulation d'une variable aléatoire X.
Il suffit d'écrire, par exemple, ces lignes :

```
E=0
N=1000
for k in range(0,N) :
    E=E+simulX()
E=E/1000
```

Notamment, le programme :

```
N=1000
X=rd.randint(1,11,N)
print(np .mean(X))          # np.mean calcule la moyenne des valeurs de X
```

Affiche une valeur proche de 5.5, qui correspond bien à l'espérance d'une loi uniforme discrète sur {1,2...,10}.