Feuille de TP nº 4

Convergences et approximations de variables aléatoires (correction)

Exercice 1. Calculer des valeurs approchées des intégrales suivantes à 10^{-4} près (avec une niveau de confiance $1-\alpha$) avec un niveau de confiance de 95% et les comparer à les approximation de leurs valeurs exactes [2] fournies par Python :

$$\int_0^1 \frac{dt}{1+t}, \qquad \int_0^2 \frac{dt}{1+t}, \qquad \int_0^1 \frac{dt}{1+t^2}, \qquad \int_{-1}^1 \sqrt{1-t^2}\,dt, \qquad \int_0^1 \ln(t)\,dt.$$

Correction :

- Ici $\varepsilon = 10^{-4}$, $\alpha = 0.05 = \dfrac{1}{20}$, $a=0$, $b=1$, $U \hookrightarrow \mathcal{U}([0\,;1])$ et $f : t \longmapsto \dfrac{1}{1+t}$ continue sur $[0\,;1]$. On a

$$\mathbb{E}(f(U)^2) = \int_0^1 \frac{dt}{(1+t)^2} = \left[-\frac{1}{1+t}\right]_0^1 = -\frac{1}{2}+1 = \frac{1}{2}$$

$$\mathbb{E}(f(U)) = \int_0^1 \frac{dt}{1+t} = [\ln(t)]_0^1 = \ln(2)$$

donc $\mathbb{V}(f(U)) = \mathbb{E}(f(U)^2) - \mathbb{E}(f(U))^2 = \dfrac{1}{2} - \ln(2)^2$. On sait que $\ln(2) \geqslant \dfrac{2}{3}$ donc $\mathbb{V}(f(U)) \leqslant \dfrac{1}{18}$. Prenons donc $M = \dfrac{1}{18}$. Il suffit donc que $n \geqslant \dfrac{(b-a)^2 M}{\alpha\varepsilon^2} = \dfrac{1}{9} \times 10^9$.

```
n=int(10**9/9)+1
import numpy.random as rd
S=0
for k in range(n):
    S=S+1/(1+rd.random())
print(S/n)
```

On a trouvé (on ne gardant que les trois premiers chiffres après les virgules)`0.693`et il s'agit bien des quatre premiers chiffres de l'écriture décimale de $\ln(2)$.

- Ici $\varepsilon = 10^{-4}$, $\alpha = 0.05 = \dfrac{1}{20}$, $a=0$, $b=2$, $U \hookrightarrow \mathcal{U}([0\,;2])$ et $f : t \longmapsto \dfrac{1}{1+t}$ continue sur $[0\,;2]$. On a

$$\mathbb{E}(f(U)^2) = \int_0^2 \frac{dt}{2(1+t)^2} = \frac{1}{2}\left[-\frac{1}{1+t}\right]_0^2 = \frac{1}{2}\left(-\frac{1}{3}+1\right) = \frac{1}{3}$$

$$\mathbb{E}(f(U)) = \int_0^2 \frac{dt}{2(1+t)} = \frac{1}{2}[\ln(1+t)]_0^2 = \frac{\ln(3)}{2}$$

donc $\mathbb{V}(f(U)) = \mathbb{E}(f(U)^2) - \mathbb{E}(f(U))^2 = \dfrac{1}{3} - \dfrac{(\ln(3))^2}{4}$. On sait que $\ln(3) \geqslant 1$ donc $\mathbb{V}(f(U)) \leqslant \dfrac{1}{3} - \dfrac{1}{4} = \dfrac{1}{12}$. Prenons donc $M = \dfrac{1}{12}$. Il suffit donc que $n \geqslant \dfrac{(b-a)^2 M}{\alpha\varepsilon^2} = \dfrac{1}{3} \times 10^9$.

```
n=int(10**9/3)+1
import numpy.random as rd
```

2. Il faut savoir les calculer bien sûr. Elles font respectivement $\ln(2)$, $\ln(3)$, $\dfrac{\pi}{4}$, $\dfrac{\pi}{2}$, -1.

```
S=0
for k in range(n):
    S=S+1/(1+2*rd.random())
print(2*S/n)
```

On a trouvé (on ne gardant que les trois premiers chiffres après les virgules)`1.098`et il s'agit bien des quatre premiers chiffres de l'écriture décimale de $\ln(3)$.

- Ici $\varepsilon = 10^{-4}$, $\alpha = 0.05 = \dfrac{1}{20}$, $a = 0$, $b = 1$, $U \hookrightarrow \mathcal{U}([0\,;1])$ et $f : t \longmapsto \dfrac{1}{1+t^2}$ continue sur $[0\,;1]$. On a

$$\mathbb{E}(f(U)^2) = \int_0^1 \frac{dt}{(1+t^2)^2} = \int_0^1 \frac{1+t^2-t^2}{(1+t^2)^2}\,dt = \int_0^1 \frac{1}{1+t^2}\,dt - \frac{1}{2}\int_0^1 \frac{t\times 2t}{(1+t^2)^2}\,dt.$$

Faisons une IPP avec $u : t \longmapsto t$ et $v : t \longmapsto -\dfrac{1}{1+t^2}$ de classe C^1 sur $[0\,;1]$. On a $u' \longmapsto 1$ et $v' : t \longmapsto \dfrac{2t}{(1+t^2)^2}$ donc

$$\int_0^1 \frac{t\times 2t}{(1+t^2)^2}\,dt = \left[-\frac{t}{1+t^2}\right]_0^1 - \int_0^1 -\frac{1}{1+t^2}\,dt = -\frac{1}{2}+\frac{\pi}{4}.$$

donc $\mathbb{E}(f(U)^2) = \dfrac{\pi}{4} - \dfrac{1}{2}\left(-\dfrac{1}{2}+\dfrac{\pi}{4}\right) = \dfrac{\pi+2}{8}$ donc $\mathbb{V}(f(U)) = \dfrac{\pi+2}{8} - \dfrac{\pi^2}{16} = \dfrac{2\pi+4-\pi^2}{8}$. On utilisant le fait que $3,1 \leqslant \pi \leqslant 3,2$, on peut montrer que c'est inférieur à $\dfrac{1}{10}$ donc prenons $M = \dfrac{1}{10}$. Il suffit donc que $n \geqslant \dfrac{(b-a)^2 M}{\alpha\varepsilon^2} = 10^9$.

```
n=10**9
import numpy.random as rd
S=0
for k in range(n):
    S=S+1/(1+rd.random()**2)
print(S/n)
```

On a trouvé (on ne gardant que les trois premiers chiffres après les virgules)`0.785`et il s'agit bien des quatre premiers chiffres de l'écriture décimale de $\dfrac{\pi}{4}$.

- Ici $\varepsilon = 10^{-4}$, $\alpha = 0.05 = \dfrac{1}{20}$, $a = -1$, $b = 1$, $U \hookrightarrow \mathcal{U}([-1\,;1])$ et $f : t \longmapsto \sqrt{1-t^2}$ continue sur $[-1\,;1]$. On a

$$\mathbb{E}(f(U)^2) = \int_{-1}^1 (1-t^2)\frac{dt}{2} = \frac{1}{2}\left[t-\frac{t^3}{3}\right]_{-1}^1 = \frac{4}{3},$$

$$\mathbb{E}(f(U)) = \int_{-1}^1 \sqrt{1-t^2}\,\frac{dt}{2} = \int_0^1 \sqrt{1-t^2}\,dt = \frac{\pi}{4}.$$

(calcul classique via le changement e variable $t = \sin(x)$). Ainsi

$$\mathbb{V}(f(U)) = \mathbb{E}(f(U)^2) - \mathbb{E}(f(U))^2 = \frac{4}{3} - \frac{\pi^2}{16} = \frac{64-3\pi^2}{48} \leqslant \frac{36}{48} = \frac{3}{4}.$$

Prenons $M = \dfrac{3}{4}$. Il suffit donc que $n \geqslant \dfrac{(b-a)^2 M}{\alpha\varepsilon^2} = 6\times 10^9$.

```
n=6*10**9
import numpy.random as rd
S=0
for k in range(n):
    S=S+np.sqrt(1-(2*rd.random()-1)**2)
print(2*S/n)
```

On a trouvé (on ne gardant que les trois premiers chiffres après les virgules)`1.570`et il s'agit bien des quatre premiers chiffres de l'écriture décimale de $\dfrac{\pi}{2}$.

- Ici $\varepsilon = 10^{-4}$, $\alpha = 0.05 = \dfrac{1}{20}$, $a = 0$, $b = 1$, $U \hookrightarrow \mathcal{U}([0\,;1])$ et $f : t \longmapsto \ln(t)$ continue sur $]0\,;1]$. On a

$$\int_\varepsilon^1 (\ln(t))^2\,dt = [\ln(t)(t\ln(t)-t)]_\varepsilon^1 - \int_\varepsilon^1 \frac{1}{t}(t\ln(t)-t)\,dt$$

via une IPP avec $t \longmapsto \ln(t)$ et $t \longmapsto t\ln(t) - t$ de classe C^1 sur $[\varepsilon\,;1]$. Ainsi

$$\int_\varepsilon^1 (\ln(t))^2\,dt = -\ln(\varepsilon)(\varepsilon\ln(\varepsilon)-t) - \int_\varepsilon^1 (\ln(t)-1)\,dt = -\ln(\varepsilon)(\varepsilon\ln(\varepsilon)-\varepsilon) - [t\ln(t)-t-t]_\varepsilon^1$$

On fait tend ε vers 0 et, par croissances comparées, on obtient que cette intégrale tend vers 2. Ainsi $\displaystyle\int_0^1 (\ln(t))^2\,dt$ converge et vaut 2. Et donc $\mathbb{E}((f(U))^2) = 2$. Aussi

$$\int_\varepsilon^1 \ln(t)\,dt = [t\ln(t)-t]_\varepsilon^1 \xrightarrow[\varepsilon\to 0]{} 1,$$

par croissances comparées. Ainsi $\displaystyle\int_0^1 \ln(t)\,dt$ converge et vaut -1. Et donc $\mathbb{E}(f(U)) = 1$ si bien que $\mathbb{V}(f(U)) = 2 - (-1)^2 = 1$. On prend donc $M = 1$ ici. Il suffit donc que $n \geqslant \dfrac{(b-a)^2 M}{\alpha\varepsilon^2} = 2\times 10^9$.

```
n=2*10**9
import numpy.random as rd
S=0
for k in range(n):
    S=S+np.log(rd.random())
print(S/n)
```

On a trouvé (on ne gardant que les trois premiers chiffres après les virgules) `-1.000`et il s'agit bien des quatre premiers chiffres de l'écriture décimale de -1.

Exercice 2 (Le cas d'une intégrale sur $\mathbb{R}_+$ ou $\mathbb{R}$). Soit f une fonction définie sur $\mathbb{R}_+$ et telle que $\displaystyle\int_0^{+\infty} f(x)\,dx$ converge absolument.

1) a) Justifier que

$$\int_0^{+\infty} f(x)\,dx = \int_0^1 f(-\ln(t))\,\frac{dt}{t}.$$

b) A l'aide de la méthode de Monte-Carlo, déterminer un algorithme permettant de calculer une valeur approchée de $\displaystyle\int_0^{+\infty} f(x)\,dx$.

2) En remarquant que $\displaystyle\int_0^{+\infty} f(x)\,dx = \int_0^{+\infty} f(x)e^x \times e^{-x}\,dx$, proposer une autre méthode permettant de calculer une valeur approchée de $\displaystyle\int_0^{+\infty} f(x)\,dx$.

3) Tester ces deux méthodes avec Python pour calculer $\displaystyle\int_0^{+\infty} e^{-x^2/2}\,dx = \sqrt{\frac{\pi}{2}}$. Comparer avec la vraie valeur.

4) Comment faire pour approcher $\displaystyle\int_{-\infty}^{+\infty} f(x)\,dx$ avec Python lorsque celle-ci converge absolument ?
On pourra s'aider d'une bijection de $]0\,;1[$ *dans* $\mathbb{R}$ *(comme par exemple* $t \longmapsto \ln(1/t-1)$, $t \longmapsto \ln(-\ln(t))$ *ou* $t \longmapsto \tan(\pi(1/2-t))$*), ou encore couper l'intégrale en deux, ou utiliser une loi Normale.*

Correction :

1) a) Puisque $\displaystyle\int_0^{+\infty} f(x)\,dx$ converge et que $\varphi : t \longmapsto -\ln(t)$ est une fonction de classe C^1 et strictement décroissante sur $]0\,;1[$ vérifiant $\lim_{t\to 0} -\ln(t) = +\infty$ et $\lim_{t\to 1} -\ln(t) = 1$, on peut faire le changement de variable donné par φ :

$$\int_0^{+\infty} f(x)\,dx = \int_1^0 f(\varphi(t))\varphi'(t)\,dt = \int_1^0 f(-\ln(t))\varphi'(t)\frac{-1}{t}\,dt = \int_0^1 f(-\ln(t))\,\frac{dt}{t}.$$

b) Le théorème de transfert entraîne donc que, si $U \hookrightarrow \mathcal{U}(]0\,;1[)$, alors

$$\mathbb{E}\left(\frac{f(-\ln(U))}{U}\right) = \int_0^1 f(-\ln(t))\,\frac{dt}{t} = \int_0^{+\infty} f(x)\,dx.$$

Si on suppose de plus que $\displaystyle\int_0^1 f^2(-\ln(t))\,\frac{dt}{t^2} = \int_0^{+\infty} f^2(x)e^x\,dx$ converge *(mais en fait pas besoin, la loi faible des grands nombres étant vraie sans cette hypothèse, mais le programme l'exige)*, alors $\mathbb{E}\left(\left(\dfrac{f(-\ln(U))}{U}\right)^2\right)$ existe donc $\dfrac{f(-\ln(U))}{U}$ admet une variance et la loi des grands nombres assure que

$$\frac{1}{n}\sum_{k=1}^{n}\frac{f(-\ln(U_i))}{U_i} \xrightarrow[n\to+\infty]{\mathbb{P}} \mathbb{E}\left(\frac{f(-\ln(U))}{U}\right) = \int_0^{+\infty} f(x)\,dx,$$

où $(U_i)_{i\geqslant 1}$ est une suite de variables aléatoires indépendantes de loi $\mathcal{U}(]0\,;1[)$. Si on implémente f en Python par `t`, alors le programme suivant renvoie une approximation de cette intégrale :

```
import numpy.random as rd
n=1000#n assez grand
S=0
for k in range(n):
    U=rd.random()
    S=S+f(-np.log(U))/U
print(S/n)
```

2) Si $X \hookrightarrow \mathcal{E}(1)$, alors la formule de transfert entraîne que

$$\int_0^{+\infty} f(x)\,dx = \int_0^{+\infty} f(x)e^x \times e^{-x}\,dx = \mathbb{E}(f(X)e^X).$$

Si on suppose de plus que $\int_0^{+\infty} f(x)e^x\,dx = \int_0^{+\infty}(f(x)e^x)^2 \times e^{-x}\,dx$ converge *(mais en fait pas besoin, la loi faible des grands nombres étant vraie sans cette hypothèse, mais le programme l'exige)*, alors $\mathbb{E}\left(\left(f(X)e^X\right)^2\right)$ existe donc $f(X)e^X$ admet une variance et la loi des grands nombres assure que

$$\frac{1}{n}\sum_{k=1}^{n} f(X_i)e^{X_i} \xrightarrow[n\to+\infty]{\mathbb{P}} \mathbb{E}\left(f(X)e^X\right) = \int_0^{+\infty} f(x)\,dx,$$

où $(X_i)_{i\geqslant 1}$ est une suite de variables aléatoires indépendantes de loi $\mathcal{E}(1)$. Si on implémente f en Python par `t`, alors le programme suivant renvoie une approximation de cette intégrale :

```
import numpy.random as rd
n=1000#n assez grand
S=0
for k in range(n):
    x=rd.exponential()
    S=S+f(X)*np.exp(X)
print(S/n)
```

En fait c'est la même méthode puisqu que $-\ln(U)$ suit une loi exponentielle de paramètre 1 lorsque U suit une loi uniforme sur $]0\,;1[$ (grand classique).

3) Méthode de la question 1 :

```python
import numpy.random as rd
n=1000#n assez grand
S=0
for k in range(n):
    U=rd.random()
    S=S+np.exp(-(-np.log(U))**2/2)/U
print(S/n)
```

Méthode de la question 2 :

```python
import numpy.random as rd
n=1000#n assez grand
S=0
for k in range(n):
    X=rd.exponential()
    S=S+np.exp(-X**2/2)*np.exp(X)
print(S/n)
```

4) • On peut montrer que $t \longmapsto \ln(1/t-1)$, $t \longmapsto \ln(-\ln(t))$ ou $t \longmapsto \tan(\pi(1/2-t))$ sont des bijections de $]0\,;1[$ dans $\mathbb{R}$ et donc on peut faire, comme dans la question 1, le changement de variable donné par cette fonction. Cela donne respectivement :

$$\int_{-\infty}^{+\infty} f(x)\,dx = \int_0^1 f\left(\ln(1/t-1)\right)\frac{dt}{t(t-1)} = \mathbb{E}\left(\frac{f(\ln(1/U-1))}{U(1-U)}\right),$$

$$\int_{-\infty}^{+\infty} f(x)\,dx = -\int_0^1 f\left(\ln(-\ln(t))\right)\frac{dt}{t\ln(t)} = \mathbb{E}\left(-\frac{f(\ln(-\ln(U)))}{U\ln(U)}\right),$$

$$\int_{-\infty}^{+\infty} f(x)\,dx = \int_0^1 \pi f\left(\tan(\pi(1/2-t))\right)\frac{dt}{\cos^2(\pi(1/2-t))} = \mathbb{E}\left(\frac{\pi f(\tan(\pi(1/2-U)))}{\cos^2(\pi(1/2-t))}\right).$$

On utilise ensuite la méthode de Monte-Carlo. Par exemple avec l'intégrale de l'exemple et la première fonction, cela donne :

```python
import numpy.random as rd
n=1000#n assez grand
S=0
for k in range(n):
    U=rd.random()
    S=S+f(np.log(1/U-1))/(U*(1-U))
print(S/n)
```

• Autre méthode : on a

$$\int_{-\infty}^{+\infty} f(x)\,dx = \int_{-\infty}^{0} f(x)\,dx + \int_0^{+\infty} f(x)\,dx = \int_0^{+\infty} f(-t)\,dt + \int_0^{+\infty} f(x)\,dx,$$

via changement de variable $t \longmapsto -t$. Ainsi, par linéarité :

$$\int_{-\infty}^{+\infty} f(x)\,dx = \int_0^{+\infty} (f(x)+f(-x))\,dx = \int_0^{+\infty} (f(x)+f(-x))\,dx.$$

On peut procéder alors comme dans la question 1 :

```python
import numpy.random as rd
n=1000#n assez grand
S=0
for k in range(n):
    U=rd.random()
    S=S+(f(-np.log(U))+f(np.log(U)))/U
print(S/n)
```

ou comme dans la question 2 :

```python
import numpy.random as rd
n=1000#n assez grand
S=0
for k in range(n):
    X=rd.exponential()
    S=S+(f(X)+f(-X))*np.exp(X)
print(S/n)
```

- Autre méthode : on a

$$\int_{-\infty}^{+\infty} f(x)\,dx = \int_{-\infty}^{+\infty} f(x)\sqrt{2\pi}e^{x^2/2} \times \frac{e^{-x^2/2}}{\sqrt{2\pi}}\,dx = \mathbb{E}\left(f(Z)\sqrt{2\pi}e^{Z^2/2}\right),$$

avec $Z \hookrightarrow \mathcal{N}(0,1)$. Et donc on propose le programme suivant :

```python
import numpy.random as rd
n=1000#n assez grand
S=0
for k in range(n):
    Z=rd.normal()
    S=S+f(Z)*np.sqrt(2*np.pi)*np.exp(Z**2/2)
print(S/n)
```

Exercice 3 (Estimation de la fonction de répartition).

1) Soit X une variable aléatoire réelle ne prenant que des valeurs positives. En utilisant la méthode de Monte-Carlo, proposer une fonction qui prend en entrée x et qui calcule une valeur approchée de $F_X(x)$, la fonction de répartition de X en x.

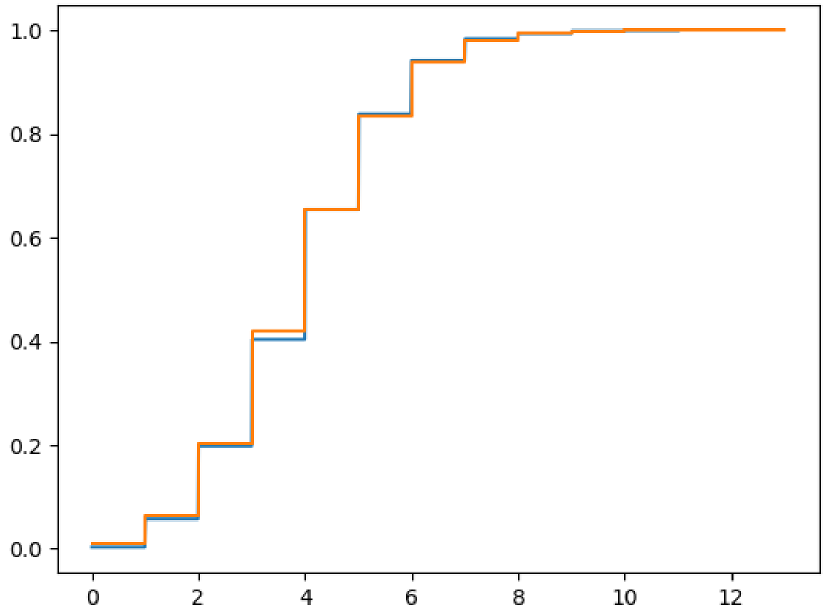
2) Utiliser cette fonction pour illustrer (en superposant la fonction de répartition estimée avec la fonction de répartition théorique) le fait que :
   - $X_1+X_2 \hookrightarrow \mathcal{B}(n_1+n_2,p)$ lorsque X_1 et X_2 sont indépendantes de lois $\mathcal{B}(n_1,p)$ et $\mathcal{B}(n_2,p)$ respectivement.
   - $X_1+X_2 \hookrightarrow \mathcal{P}(a_1+a_2)$ lorsque X_1 et X_2 sont indépendantes de lois $\mathcal{P}(a_1)$ et $\mathcal{P}(a_2)$ respectivement.
   - $X_1+X_2 \hookrightarrow \mathcal{N}(m_1+m_2,\sigma_1^2+\sigma_2^2)$ lorsque X_1 et X_2 sont indépendantes de lois $\mathcal{N}(m_1,\sigma_1^2)$ et $\mathcal{N}(m_2,\sigma_2^2)$ respectivement.
   - $X_1+X_2+\cdots+X_n \hookrightarrow \gamma(n)$ lorsque $X_1,\dots,X_n$ sont indépendantes de loi $\mathcal{E}(1)$.
     Pour la fonction de répartition en un réel x de la loi $\gamma(n)$, on utilisera `sp.gdtr(1,n,x)` après `import scipy.special as sp`.

Correction :

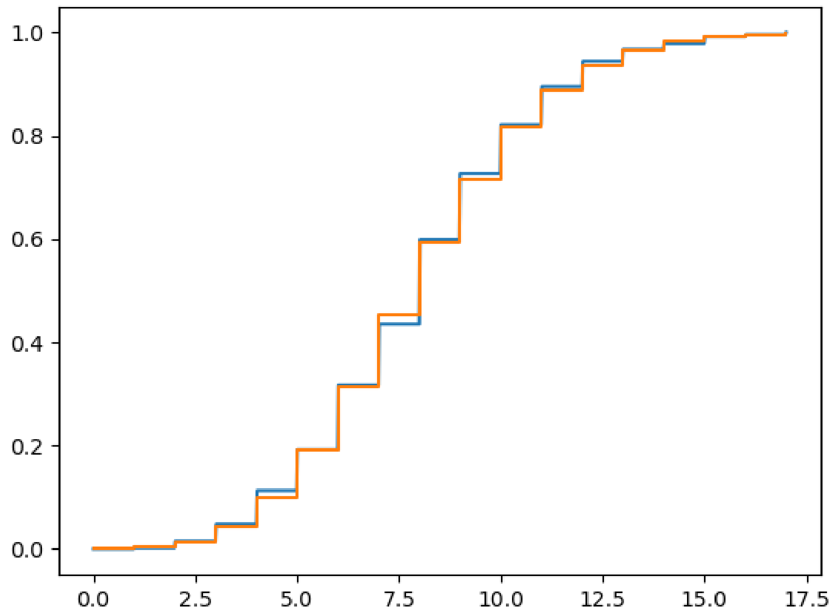
1) Pour tout $x \in \mathbb{R}$,
$$F_X(x) = \mathbb{P}(X \leqslant x) = \mathbb{E}(\mathbb{1}_{X\leqslant x}).$$
Puisque $\mathbb{1}_{X\leqslant x}$ est une variable aléatoire finie, elle admet bien sûr une variance et une espérance. Si $(X_i)_{i\geqslant 1}$ est une suite de variables aléatoires indépendantes de même loi que X, alors $(\mathbb{1}_{X_i\leqslant x})_{i\geqslant 1}$ est une suite de variables aléatoires indépendantes (par théorème des coalitions) de même loi que $\mathbb{1}_{X\leqslant x}$ et donc la loi faible des grands nombres entraîne que
$$\frac{1}{n}\sum_{k=1}^{n} \mathbb{1}_{X_i\leqslant x} \xrightarrow[n\to+\infty]{\mathbb{P}} \mathbb{E}(\mathbb{1}_{X\leqslant x}) = F_X(x).$$
Si `X` est un vecteur en Python qui contient n (avec grand) réalisations de X et si `x` implémente le réel x, alors `X<=x` est un vecteur avec des `True` là où les coordonnées de `X` sont inférieures ou égales à x et `False` sinon. Ainsi `X<=x` contient n réalisations de $\mathbb{1}_{X\leqslant x}$ (où les `True` valent 1 et les `False` valent 0). Ainsi la commande `np.mean(X<=x)` renvoie une approximation de $F_X(x)$.

2) •
```python
n1=9
n2=4
p=0.3
X1=rd.binomial(n1,p,1000)
X2=rd.binomial(n2,p,1000)
X=X1+X2
```

```
#On trace la fonction de répartition empirique
#(obtenue avec la méthode de Monte-Carlo)
Abs=np.linspace(0,np.max(X)+1,1000)
Ord=[np.mean(X<=x) for x in Abs]
plt.plot(Abs,Ord)
#On trace la fonction de répartition théorique (cf. 1ère année)
n=n1+n2
prob=np.zeros(n+1)
frep=np.zeros(n+1)
prob[0]=(1-p)**n
frep[0]=(1-p)**n
for k in range(1,n+1):
    c=(n-k+1)*p/(k*(1-p))
    prob[k]=c*prob[k-1]
    frep[k]=frep[k-1]+prob[k]
plt.step(range(0,n+1),frep,where='post')
plt.show()
```

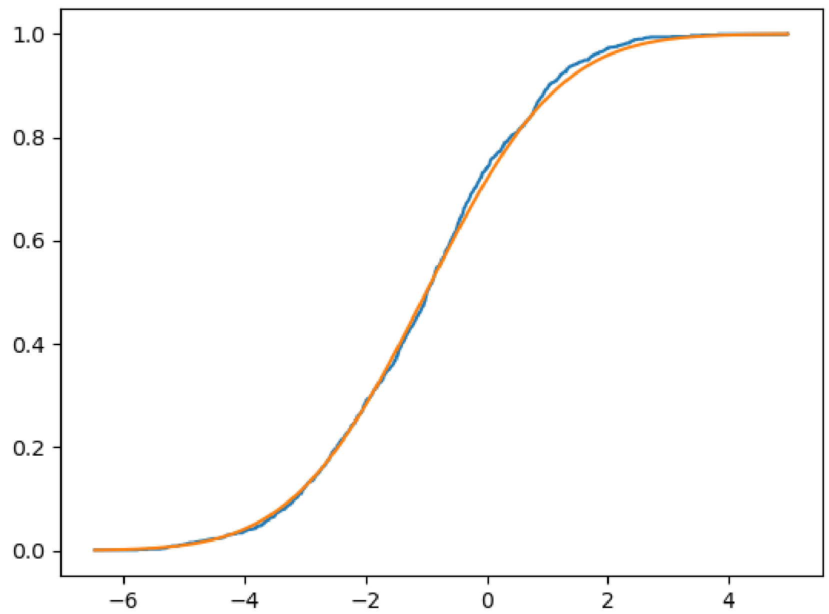



- ```
  a1=5
  a2=3
  X1=rd.poisson(a1,1000)
  X2=rd.poisson(a2,1000)
  X=X1+X2
  #On trace la fonction de répartition empirique
  #(obtenue avec la méthode de Monte-Carlo)
  Abs=np.linspace(0,np.max(X),1000)
  Ord_emp=[np.mean(X<=x) for x in Abs]
  plt.plot(Abs,Ord_emp)
  #On trace la fonction de répartition théorique (cf. 1ère année)
  a=a1+a2
  prob=np.exp(-a)
  frep=prob
  F=[frep]
  k=0
  for k in range(1,np.max(X)+1):
      prob=prob*a/k
      frep=frep+prob
      F.append(frep)
  plt.step(range(np.max(X)+1),F,where='post')
  plt.show()
  ```
  ```



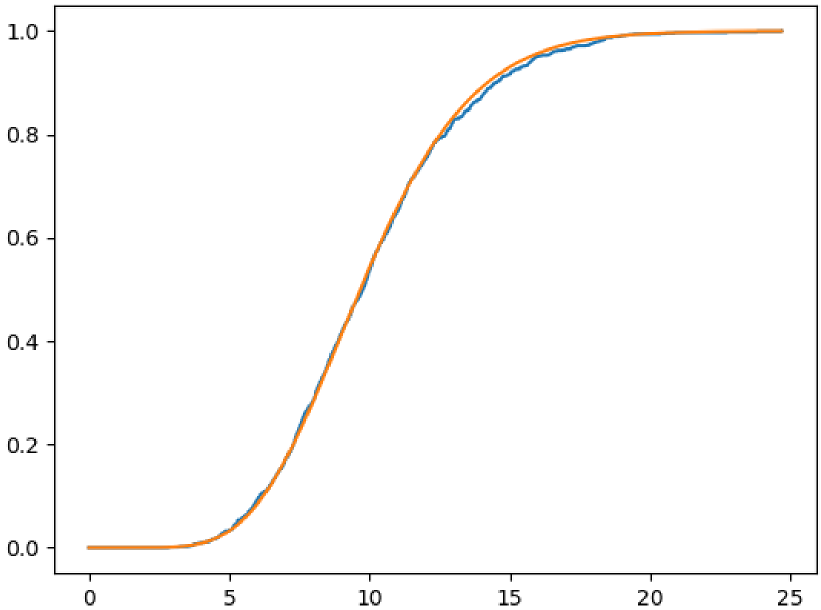
- On rappelle (grand classique à démontrer) que $x \longmapsto \Phi\left(\dfrac{x-(m_1+m_2)}{\sqrt{\sigma_1^2+\sigma_2^2}}\right)$ est la fonction de répartition d'une variable aléatoire de loi $\mathcal{N}(m_1+m_2, \sigma_1^2+\sigma_2^2)$.

```
import scipy.special as sp
m1=5; var1=1
m2=-6; var2=2
X1=rd.normal(m1,np.sqrt(var1),1000)
X2=rd.normal(m2,np.sqrt(var2),1000)
X=X1+X2
#On trace la fonction de répartition empirique
#(obtenue avec la méthode de Monte-Carlo)
Abs=np.linspace(np.min(X),np.max(X),1000)
Ord_emp=[np.mean(X<=x) for x in Abs]
plt.plot(Abs,Ord_emp)
#On trace la fonction de répartition théorique
Ord_theo=[sp.ndtr((x-(m1+m2))/np.sqrt(var1+var2)) for x in Abs]
plt.plot(Abs,Ord_theo)
plt.show()
```



•

```python
import scipy.special as sp
n=10
X=np.zeros(1000)
for k in range(n):
    X=X+rd.exponential(1,1000)
#On trace la fonction de répartition empirique
#(obtenue avec la méthode de Monte-Carlo)
Abs=np.linspace(0,np.max(X),1000)
Ord_emp=[np.mean(X<=x) for x in Abs]
plt.plot(Abs,Ord_emp)
#On trace la fonction de répartition théorique
Ord_theo=[sp.gdtr(1,n,x) for x in Abs]
plt.plot(Abs,Ord_theo)
plt.show()
```



Exercice 4 (Produit de deux variables aléatoires de loi uniforme). Soient X et Y deux variables aléatoires indépendantes de même loi uniforme sur $]0\,;1[$. On note $Z = XY$.

1) Écrire une fonction Python qui prend en entrée t et qui calcule une valeur approchée de $F_Z(t)$ à l'aide de la méthode de Monte-Carlo.
2) Avec Python, représenter graphiquement la courbe de F_Z sur $[-1/2\,;3/2]$. Lui superposer la courbe de la fonction $t \longmapsto t(1-\ln(t))\mathbb{1}_{]0\,;1[}(t) + \mathbb{1}_{[1\,;+\infty[}(t)$. Que conjecturer ?
3) Montrer la conjecture.
   On commencera par calculer la fonction de répartition de $-\ln(Z)$.

Correction :

1) On l'a vu dans l'exercice précédente : si $t \in \mathbb{R}$ et $(Z_i)_{i\geqslant 1}$ est une suite de variables aléatoires indépendantes de même loi que Z, alors la loi faible des grands nombres entraîne que

$$\frac{1}{n}\sum_{k=1}^{n} \mathbb{1}_{Z_i\leqslant t} \xrightarrow[n\to+\infty]{\mathbb{P}} \mathbb{E}(\mathbb{1}_{Z\leqslant t}) = F_Z(t).$$

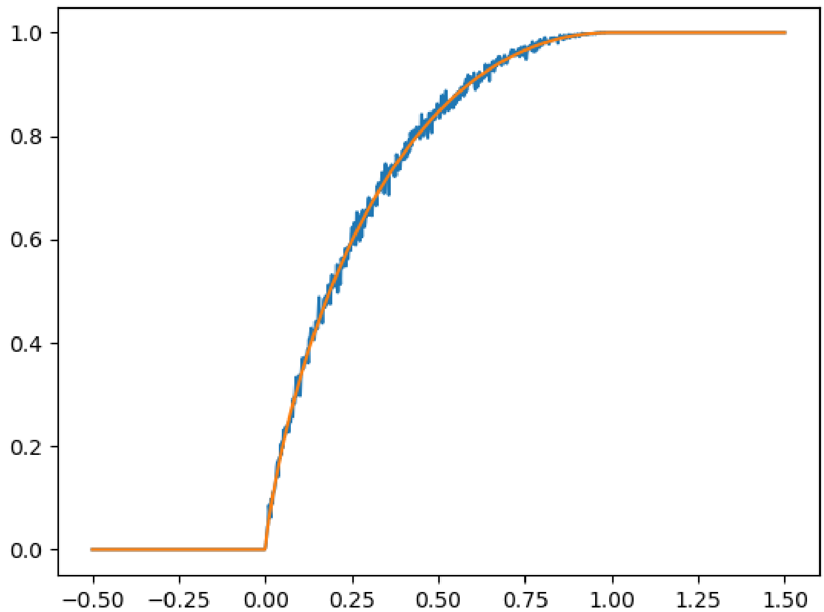
```python
import numpy.random as rd
def Frep_Z(t):
    n=1000
    Z=np.mean([rd.random()*rd.random()<=t for k in range(n)])
    return Z
```

Autre version :

```
import numpy.random as rd
def Frep_Z(t):
    n=1000
    S=0
    for k in range(n):
        X=rd.random()
        Y=rd.random()
        if X*Y<=t:
            S=S+1
    return S/n
```

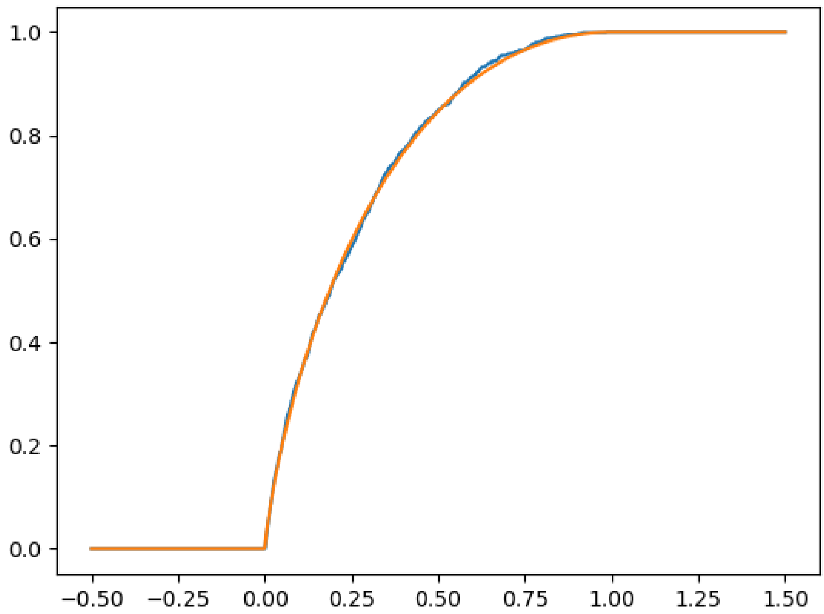
2)

```
import numpy as np
import matplotlib.pyplot as plt

Abs=np.linspace(-1/2,3/2,1000)
Ord1=[Frep_Z(t) for t in Abs]
plt.plot(Abs,Ord1)

def F(t):

    if t<=0:
        return 0
    elif t>=1:
        return 1
    else:
        return t*(1-np.log(t))

Ord2=[F(t) for t in Abs]
plt.plot(Abs,Ord2)
plt.show()
```



On peut faire mieux en simulant une seul fois 1000 simulations de Z au lieu de, pour chaque valeur de t, refaire 1000 simulations :

```
Abs=np.linspace(-1/2,3/2,1000)
Z=[rd.random()*rd.random() for k in range(1000)]
Ord1=[Z<=t for t in Abs]
plt.plot(Abs,Ord1)
Ord2=[F(t) for t in Abs]
plt.plot(Abs,Ord2)
plt.show()
```



On conjecture que Z est une variable aléatoire dont la loi admet $t \longmapsto t(1-\ln(t))\mathbb{1}_{]0\,;1[}(t) + \mathbb{1}_{[1\,;+\infty[}(t)$ pour fonction de répartition.

3) On a $-\ln(Z) = -\ln(X) - \ln(Y)$. On a $(-\ln(X))(\Omega) = \mathbb{R}_+^*$. Si $x > 0$, alors

$$\mathbb{P}(-\ln(X) \leqslant x) = \mathbb{P}(X \geqslant e^{-x}) = 1 - \mathbb{P}(X < e^{-x}) = 1 - e^{-x}$$

puisque $e^{-x} \in]0\,;1[$ et X est uniforme sur $]0\,;1[$. On a $\mathbb{P}(-\ln(X) \leqslant x) = 0$ si $x \leqslant 0$. On en déduit que $-\ln(X)$ suit une loi exponentielle de paramètre 1 puisque la fonction de répartition caractérise la loi. Il en est de même pour $-\ln(Y)$. Par théorème des coalitions, $-\ln(X)$ et $-\ln(Y)$ sont indépendantes et donc leur somme, $-\ln(Z)$ suit une loi $\gamma(2)$. Ainsi :

- Si $x \leqslant 0$, $\mathbb{P}(-\ln(Z) \leqslant x) = 0$.
- Si $x > 0$,
$$\mathbb{P}(-\ln(Z) \leqslant x) = \int_0^x ue^{-u}\,du.$$

On a $Z(\Omega) =]0\,;1[$ donc :

- Si $t \leqslant 0$, $\mathbb{P}(Z \leqslant t) = 0$.
- Si $t \geqslant 0$, $\mathbb{P}(Z \leqslant t) = 1$.
- Si $t \in]0\,;1[$,
$$\mathbb{P}(Z \leqslant t) = \mathbb{P}(-\ln(Z) \geqslant -\ln(t)) = 1 - \int_0^{-\ln(t)} ue^{-u}\,du.$$
Une IPP avec $u \longmapsto -e^{-u}$ et $u \longmapsto u$ de classe C^1 sur $[0\,;-\ln(t)]$ donne :
$$\mathbb{P}(Z \leqslant t) = 1 - \left(\left[-ue^{-u}\right]_0^{-\ln(t)} - \int_0^{-\ln(t)} -e^{-u}\,du\right) = 1 - \left(\left[-ue^{-u}\right]_0^{-\ln(t)} - \left[e^{-u}\right]_0^{-\ln(t)}\right)$$
et donc $\mathbb{P}(Z \leqslant t) = t - t\ln(t)$.

D'où la conjecture.

Exercice 5 (D'après EDHEC 2017 et les oraux ESCP 2003). Soit $(X_n)_{n\geqslant 1}$ une suite de variables aléatoires définies sur un même espace probabilisé $(\Omega, \mathcal{A}, \mathscr{P})$ qui sont indépendantes et de même loi $\mathcal{U}([0\,;1])$. Pour tout $n \in \mathbb{N}^*$, on note $M_n = \max\{X_1, \ldots, X_n\}$. Plus rigoureusement

$$\forall \omega \in \Omega, \qquad M_n(\omega) = \max\{X_1(\omega), \ldots, X_n(\omega)\}.$$

On admet que M_n est que variable aléatoire. Enfin, on pose $Y_n = n(1 - M_n)$.

1) a) Soit $n \in \mathbb{N}^*$. Calculer F_{M_n} puis montrer que M_n est une variable aléatoire à densité.

b) Établir l'existence de $\mathbb{E}(M_n)$ et $\mathbb{E}(M_n^2)$. Les calculer.

c) Soit $\varepsilon > 0$. Donner un majorant (ne dépendant que de n et de ε) de $\mathbb{P}\big((M_n - 1)^2 \geqslant \varepsilon^2\big)$.

d) Conclure que $\lim\limits_{n \to +\infty} \mathbb{P}(|M_n - 1| \geqslant \varepsilon) = 0$. Que signifie ce résultat ?

2) a) Écrire une fonction en Python appelée `SimulM` qui prend en entrée n et qui renvoie une simulation de M_n.

b) Pour $n = 1000$, construire l'histogramme renormalisé de 10000 réalisations de Y_n avec 15 classes. Lui superposer la densité d'une variable aléatoire de loi $\mathcal{E}(1)$. Que peut-on conjecturer ?

c) Pour tout $n \in \mathbb{N}^*$, calculer F_{Y_n}.

d) Pour tout $x \in \mathbb{R}_+^*$, calculer $\lim\limits_{n \to +\infty} F_{Y_n}(x)$.

e) Montrer la conjecture de la question 2b.

3) On considère les instruction suivantes en Python :

```
import numpy as np
import numpy.random as rd

def SimulZ(n):
    U=n*rd.random(n)
    return np.min(U)

m=5000; n=100; T=np.zeros(m)
for k in range(m):
    T[k]=SimulZ(n)
print(np.sum(T<=1)/m)
```

a) Montrer que la fonction `SimulZ`, de paramètre $n \in \mathbb{N}^*$ simule une variable aléatoire de même loi que Y_n.

b) En exécutant 10 fois le programme précédent, l'écran affiche les résultats suivants :

`0.6418 0.6334 0.6452 0.629 0.6212 0.626 0.641 0.6282 0.63 0.6308`

La valeur affichée à l'écran est une valeur approchée d'un nombre p. Préciser, à l'aide des questions précédentes, la valeur exacte de p.

Correction :

1) a) Soit $x \in \mathbb{R}$. On a, par indépendance,

$$F_{M_n}(x) = \mathbb{P}\left(\bigcap_{i=1}^{n} [X_i \leqslant x]\right) = \prod_{i=1}^{n} \mathbb{P}(X_i \leqslant x) = \mathbb{P}(X_1 \leqslant x)^n = \begin{cases} 0 & \text{si} \quad x < 0 \\ x^n & \text{si} \quad x \in [0\,;1] \\ 1 & \text{si} \quad x > 1 \end{cases}$$

Il s'agit d'une fonction de classe C^1 sur $\mathbb{R}$ sauf peut-être en 0 et en 1. Elle est continue sur $\mathbb{R}$ donc M_n est une variable à densité. Une densité est :

$$f_n : x \longmapsto \begin{cases} nx^{n-1} & \text{si} \quad x \in [0\,;1] \\ 0 & \text{sinon} \end{cases}$$

b) M_n^2 admet une espérance, par théorème de transfert puisque $x \longmapsto x^2 f_n(x)$ est continue sur $[0\,;1]$ et nulle en dehors de ce segment. Ainsi $\mathbb{E}(M_n^2)$ existe et donc $\mathbb{E}(M_n)$ aussi.

- $\mathbb{E}(M_n) = \displaystyle\int_{-\infty}^{+\infty} x f_n(x)\,dx = \int_0^1 nx^n\,dx = \frac{n}{n+1}$.
- Par transfert, $\mathbb{E}(M_n^2) = \displaystyle\int_{-\infty}^{+\infty} x^2 f_n(x)\,dx = \int_0^1 nx^{n+1}\,dx = \frac{n}{n+2}$.

c) Comme $(M_n-1)^2 \geqslant 0$, l'inégalité de Markov entraîne que

$$\mathbb{P}\big((M_n-1)^2 \geqslant \varepsilon^2\big) \leqslant \frac{\mathbb{E}((M_n-1)^2)}{\varepsilon^2} = \frac{\mathbb{E}(M_n^2) - 2\mathbb{E}(M_n) + 1}{\varepsilon^2},$$

par linéarité. On a

$$\mathbb{E}(M_n^2) - 2\mathbb{E}(M_n) + 1 = \frac{n}{n+2} - \frac{2n}{n+1} + 1 = \frac{n(n+1) - 2n(n+2) + (n+1)(n+2)}{(n+1)(n+2)} = \frac{2}{(n+1)(n+2)}$$

donc $\mathbb{P}\big((M_n-1)^2 \geqslant \varepsilon^2\big) \leqslant \dfrac{2}{\varepsilon^2(n+1)(n+2)}$.

d) Puisque $t \longmapsto t^2$ est strictement croissante sur $\mathbb{R}_+$, on a

$$\mathbb{P}\big(|M_n - 1| \geqslant \varepsilon\big) = \mathbb{P}\big((M_n-1)^2 \geqslant \varepsilon^2\big) \leqslant \frac{2}{\varepsilon^2(n+1)(n+2)}.$$

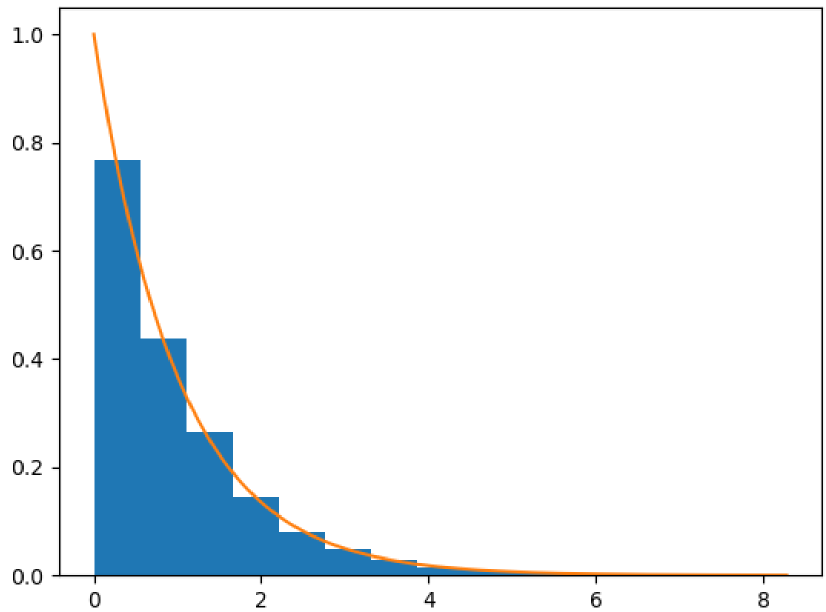
Par théorème d'encadrement $\lim\limits_{n\to+\infty} \mathbb{P}(|M_n-1| \geqslant \varepsilon) = 0$. Cela étant valable pour tout $\varepsilon > 0$, on en déduit que $(M_n)_{n\geqslant 1}$ converge en probabilité vers 1 (vers une variable aléatoire constante égale à 1 pour être plus rigoureux).

2) a)

```python
import numpy as np
import numpy.random as rd

def SimulM(n):
    return np.max(rd.random(n))
```

b)

```python
import matplotlib.pyplot as plt

n=1000
Y=[n*(1-SimulM(n)) for k in range(10000)]
plt.hist(Y,15,density=True)

Abs=np.linspace(0,np.max(Y),1000)
Ord=[np.exp(-x) for x in Abs]
plt.plot(Abs,Ord)
plt.show()
```



On conjecture que $(Y_n)_{n\geqslant 1}$ converge en loi vers une variable aléatoire de loi $\mathcal{E}(1)$.

c) Soit $n \in \mathbb{N}^*$. Soit $x \in \mathbb{R}$. On a

$$F_{Y_n}(x) = \mathbb{P}\left(M_n \leqslant 1 - \frac{x}{n}\right) = 1 - F_{M_n}\left(1 - \frac{x}{n}\right)$$

donc, d'après la question 1,

$$F_{Y_n}(x) = 1 - \begin{cases} 0 & \text{si} \quad 1 - \frac{x}{n} < 0 \\ \left(1 - \dfrac{x}{n}\right)^n & \text{si} \quad 1 - \frac{x}{n} \in [0\,;1] \\ 1 & \text{si} \quad 1 - \frac{x}{n} > 1 \end{cases}$$

donc

$$F_{Y_n}(x) = \begin{cases} 1 & \text{si} \quad x > n \\ 1 - \left(1 - \dfrac{x}{n}\right)^n & \text{si} \quad x \in [0\,;n] \\ 0 & \text{si} \quad x < 0 \end{cases}$$

d) Pour tout $x \in \mathbb{R}_+^*$, pour n assez grand, $x \in [0\,;n]$ donc

$$F_{Y_n}(x) = 1 - \left(1 - \frac{x}{n}\right)^n \xrightarrow[n \to +\infty \to 1]{} -e^{-x},$$

puisque (classique) $\dfrac{x}{n} \xrightarrow[n \to +\infty]{} 0$ donc $n \ln\left(1 - \dfrac{x}{n}\right) \sim n\left((-\dfrac{x}{n}\right) \xrightarrow[n \to +\infty]{} -x$ et par continuité de $\exp$ en $-x$.

e) On reconnaît que $(F_n)_{n \geqslant 1}$ converge simplement vers la fonction de répartition d'une variable aléatoire de loi exponentielle de paramètre 1. Ainsi $(Y_n)_{n \geqslant 1}$ converge en loi vers une variable aléatoire de loi $\mathcal{E}(1)$.

3) On considère les instruction suivantes en Python :

```
import numpy as np
import numpy.random as rd

def SimulZ(n):
    U=n*rd.random(n)
    return np.min(U)

m=5000; n=100; T=np.zeros(m)
for k in range(m):
    T[k]=SimulZ(n)
print(np.sum(T<=1)/m)
```

a) Soit $n \in \mathbb{N}^*$. La commande `rd.random(n)` renvoie un vecteur contenant n réalisations de variables aléatoires de loi $\mathcal{U}(]0\,;1[)$ indépendantes. Donc la commande `n*rd.random(n)` renvoie un vecteur contenant n réalisations de variables aléatoires de loi $\mathcal{U}(]0\,;n[)$ indépendantes. Ainsi `SimulZ(n)` simule la variable aléatoire

$$Z_n = \min\{T_1, \ldots, T_n\},$$

où $T_1, \ldots, T_n$ sont des variables aléatoires indépendantes de loi $\mathcal{U}(]0\,;n[)$.

Si $x \in]0\,;n[$, par indépendance de $T_1, \ldots, T_n$,

$$\mathbb{P}(Z_n \leqslant x) = 1 - \mathbb{P}(Z_n > x) = 1 - \mathbb{P}(X_1 > x)^n = 1 - (1 - \mathbb{P}(X_1 \leqslant x))^n = 1 - \left(1 - \frac{x}{n}\right)^n.$$

Si $x < 0$, $\mathbb{P}(Z_n \leqslant x) = 0$ et, si $x > n$, $\mathbb{P}(Z_n \leqslant x) = 1$. On reconnaît que Z_n et Y_n ont la même fonction de répartition donc suivent la même loi, puisque la fonction de répartition caractérise la loi.

b) La commande `T` contient, après la boucle, 5000 réalisations de Z_n... ou de Y_n qui a la même loi. La commande `np.sum(T<=1)/m` contient la proportion de ces réalisations qui sont inférieures ou égales à 1. La loi faible des grands nombres assure que c'est une approximation de $\mathbb{P}(Z_n \leqslant 1)$, lui même approximativement égal à $\mathbb{P}(Z \leqslant 1) = 1 - e^{-1}$ où $Z \hookrightarrow \mathcal{E}(1)$.

Ces valeurs sont donc des valeurs approchées de $p = 1 - 1/e$.

Exercice 6 (Convergence vers la loi de Gumbel).

1) Montrer que $f : x \longmapsto \exp\left(-(t + e^{-t})\right)$ est une densité de probabilité. On dit qu'une variable aléatoire qui

admet pour densité f suit une loi de Gumbel.

2) Soit U une variable aléatoire de loi $\mathcal{U}(]0\,;1[)$. On admet que $X = \ln\left(\frac{U}{1-U}\right)$ est une variable aléatoire. Calculer sa fonction de répartition et montrer que c'est une variable à densité.

3) Soient $(X_n)_{n\geqslant 1}$ une suite de variables aléatoires définies sur un même espace $(\Omega, \mathcal{A}, \mathscr{P})$ qui sont indépendantes et de même loi que X. Pour tout $n \in \mathbb{N}^*$, on note $Y_n = \max\{X_1, \ldots, X_n\}$. Plus rigoureusement

$$\forall \omega \in \Omega, \qquad Y_n(\omega) = \max\{X_1(\omega), \ldots, X_n(\omega)\}.$$

On pose $Z_n = Y_n - \ln(n)$.

a) Écrire une fonction en Python qui prend en entrée n et qui simule Y_n.

b) Pour $n = 1000$, tracer l'histogramme renormalisé de 10000 réalisations de Y_n avec 20 classes. Lui superposer le graphe de f. Que peut-on conjecturer ?

4) Montrer que la conjecture de la question précédente.

Correction :

1) La fonction f est positive (c'est une exponentielle) et continue sur $\mathbb{R}$ par composition de fonctions qui le sont. Soient A et B des réels tels que $A < B$. On a :

$$\int_A^B \exp\left(-(t+e^{-t})\right)\, dt = \int_A^B e^{-t}e^{-e^{-t}}\, dt = \left[e^{-e^{-t}}\right]_A^B = e^{-e^{-B}} - e^{-e^{-A}}.$$

Puisque $e^{-e^{-B}} \xrightarrow[B\to+\infty]{} 1$ et $e^{-e^{-A}} \xrightarrow[A\to+\infty]{} 0$, on en déduit que $\int_{-\infty}^{+\infty} \exp\left(-(t+e^{-t})\right)\, dt$ converge et vaut 1. Ainsi f est bien une densité de probabilité.

2) Soit $x \in \mathbb{R}$. Comme $\exp$ est strictement croissante sur $\mathbb{R}$, on a

$$\begin{aligned}\mathbb{P}(X \leqslant x) = \mathbb{P}\left(\frac{U}{1-U} \leqslant e^x\right) = \mathbb{P}\left(U \leqslant e^x(1-U)\right) = \mathbb{P}\left(U(1+e^x) \leqslant e^x\right) &= \mathbb{P}\left(U \leqslant \frac{e^x}{1+e^x}\right)\\ &= \mathbb{P}\left(U \leqslant \frac{1}{e^{-x}+1}\right).\end{aligned}$$

Puisque $\frac{1}{e^{-x}+1} \in]0\,;1[$ et que U suit une loi uniforme sur $]0\,;1[$, on a

$$F_X(x) = \mathbb{P}(X \leqslant x) = \frac{1}{e^{-x}+1}.$$

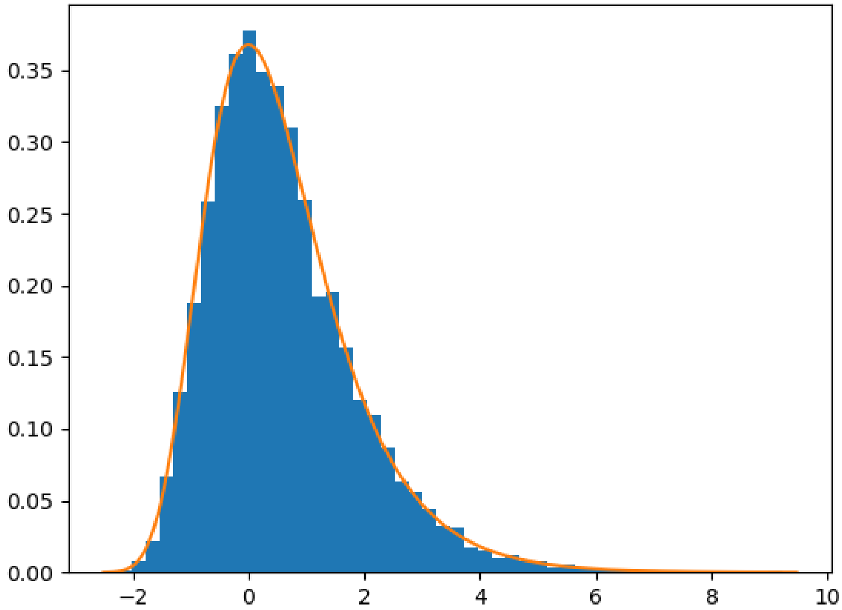
La fonction F_X est de classe C^1 sur $\mathbb{R}$ donc X est à densité.

3) a)

```python
import numpy as np
import numpy.random as rd

def SimulY(n):
    T=np.zeros(n)
    for k in range(n):
        U=rd.random()
        T[k]=np.log(U/(1-U))
    return np.max(T)
```

b)

```python
import matplotlib.pyplot as plt
N=10000
L=[SimulY(1000)-np.log(1000) for k in range(N)]
plt.hist(L,50,density=True)

Abs=np.linspace(np.min(L),np.max(L),1000)
Ord=[np.exp(-(x+np.exp(-x))) for x in Abs]
plt.plot(Abs,Ord)
plt.show()
```



On conjecture que$(Z_n)_{n\geqslant 1}$ converge en loi vers une variable aléatoire de loi de Gumbel.
Écrire une fonction en Python qui prend en entrée n et qui simule Y_n.

c) Pour $n = 1000$, tracer l'histogramme renormalisé de 10000 réalisations de Y_n avec 20 classes. Lui superposer le graphe de f. Que peut-on conjecturer ?

4) Soient $n \in \mathbb{N}^*$ et $x \in \mathbb{R}$. On a

$$F_{Z_n}(x) = \mathbb{P}(Y_n \leqslant x + \ln(n)) = \mathbb{P}\left(\bigcap_{i=1}^{n}[X_i \leqslant x + \ln(n))\right) = \prod_{i=1}^{n}\mathbb{P}(X_i \leqslant x + \ln(n)) = \mathbb{P}(X \leqslant x + \ln(n))^n,$$

puisque les X_i, $i \leqslant 1$, sont indépendantes et de même loi. Ainsi

$$F_{Z_n}(x) = \left(\frac{1}{e^{-x-\ln(n)}+1}\right)^n = \left(\frac{1}{e^{-x}/n+1}\right)^n = \exp\left(-n\ln\left(1+\frac{e^{-x}}{n}\right)\right).$$

On a $\ln\left(1+\dfrac{e^{-x}}{n}\right) \underset{n\to+\infty}{\sim} \dfrac{e^{-x}}{n}$ car $\dfrac{e^{-x}}{n} \xrightarrow[n\to+\infty]{} 0$ donc $-n\ln\left(1+\dfrac{e^{-x}}{n}\right) \underset{n\to+\infty}{\sim} -e^{-x} \xrightarrow[n\to+\infty]{} e^{-x}$. Par continuité de la fonction exponentielle en $-e^{-x}$, on obtient $F_{Z_n}(x) \xrightarrow[n\to+\infty]{} e^{-e^{-x}} = \displaystyle\int_{-\infty}^{x} \exp\left(-(t+e^{-t})\right) dt$.
Ainsi $(Z_n)_{n\geqslant 1}$ converge en loi vers une variable aléatoire de loi de Gumbel, ce qui démontre la conjecture.

Exercice 7 (Simulation d'une loi de Poisson).

1) À l'aide de l'approximation Binomiale/Poisson, écrire une fonction Python, appelée `Simul_1`, qui prend en entrée $a > 0$ et qui simule une variable aléatoire de loi de Poisson de paramètre a à l'aide d'une variable aléatoire de loi binomiale de premier paramètre 1000 en utilisant `rd.binomial()`.

2) On a vu dans l'exercice 11 du TP n° 1 que, si $a > 0$ et si $(U_i)_{i\geqslant 1}$ est une suite de variables aléatoires indépendantes et de même loi uniforme sur $]0\,;1[$, alors

$$Y = \min\{n \in \mathbb{N} \,|\, U_1U_2\ldots U_{n+1} \leqslant e^{-a}\}$$

suit une loi de Poisson de paramètre a. Écrire une fonction Python, appelée `Simul_2`, qui prend en entrée $a > 0$ et qui simule une variable aléatoire loi de Poisson de paramètre a à l'aide de ce dernier résultat.

3) Soit $a > 0$. Soit U une variable aléatoire de loi uniforme sur $]0\,;1[$. Pour tout $k \in \mathbb{N}$, notons $f_k = \displaystyle\sum_{k=0}^{n} e^{-a}\frac{a^k}{k!}$.
   a) Calculer $\mathbb{P}(f_{k-1} < U \leqslant f_k)$ pour tout $k \in \mathbb{N}^*$.
   b) En déduire que $Y = \min\{k \in \mathbb{N} \,|\, U \leqslant f_k\}$ suit une loi de Poisson de paramètre a.

c) Compléter alors le programme suivant pour qu'il prenne en entrée $a > 0$ et simule une variable aléatoire de loi de Poisson de paramètre a à l'aide de ce dernier résultat.

```
import ......................
import ......................
def Simul_3(a):
    p=np.exp(-a)
    f=p
    k=0
    U = ......................
    ........................
        k=k+1
        p = ...................
        f=f+p
    return ..................
```

4) Si on importe la librairie `time`, alors la commande `time.time()` renvoie un flottant contenant l'heure en secondes (avec une certaine précision). Si on exécute cette commande avant et après un script, l'écart entre les deux valeurs donne alors le temps d'exécution du script. Recopier et exécuter le script suivant :

```
import time
N=100000
a=4
s1=time.time(); X1=[Simul_1(a) for k in range(N)]; e1=time.time()
s2=time.time(); X2=[Simul_2(a) for k in range(N)]; e2=time.time()
s3=time.time(); X3=[Simul_3(a) for k in range(N)]; e3=time.time()
s4=time.time(); X4=[rd.poisson(a) for k in range(N)]; e4=time.time()
print(e1-s1,e2-s2,e3-s3,e4-s4)
M=np.max([np.max(X1),np.max(X2),np.max(X3),np.max(X4)])
valeurs=[i-0.5 for i in range(M+1)]
plt.hist([X1,X2,X3,X4],bins=valeurs,density=True)
plt.show()
```

Commenter.

Correction :

1)

```
import numpy as np
import numpy.random as rd

def Simul_1(a):
    return rd.binomial(1000,a/1000)
```

2)

```
def Simul_2(a):
    n=0
    prod=rd.random()
    p=np.exp(-a)
    while prod>p:
        prod=prod*rd.random()
        n=n+1
    return n
```

a) Pour tout $k \in \mathbb{N}$, f_k est la fonction de répartition en $n \in \mathbb{N}^*$ d'une variable aléatoire de loi de Poisson donc $f_k \in [0\,;1]$. Ainsi

$$\forall k \in \mathbb{N}^*, \qquad \mathbb{P}(f_{k-1} < U \leqslant f_k) = F_U(f_k) - F_U(f_{k-1}) = f_k - f_{k-1} = e^{-a}\frac{a^k}{k!}$$

(puisque $F_U(t) = t$ lorsque $t \in [0\,;1]$).

b) *Rigoureusement Y est définie ainsi :*

$$\forall \omega \in \Omega, \qquad Y(\omega) = \{k \in \mathbb{N} \,|\, U(\omega) \leqslant f_k\}.$$

- *Pour tout $\omega \in \Omega$, la suite $(f_j)_{j\in\mathbb{N}}$ est croissante et de limite 1 donc, comme $U(\omega) \in]0\,;1[$, il existe un rang à partir duquel f_k dépasse $U(\omega)$. Ainsi la partie $\{k \in \mathbb{N} \,|\, U(\omega) \leqslant f_k\}$ est non vide dans $\mathbb{N}$ est donc minorée. Ainsi $Y(\omega)$ existe.*
- *Ensuite (on va le montrer dans un instant), pour tout $k \in \mathbb{N}$, $[U = k] = [f_{k-1} < U \leqslant f_k] \in \mathcal{A}$ (en prenant la convention que $f_{k-1} = 0$). Ainsi Y est bien une variable aléatoire.*

Soit $k \in \mathbb{N}$. La variable aléatoire Y prend la valeur k si et seulement si f_k est le premier des f_j qui dépasse la valeur prise par U si et seulement si $U \leqslant f_k$ mais $U > f_{k-1}$ (en prenant la convention que $f_{k-1} = 0$). On a donc $[U = k] = [f_{k-1} < U \leqslant f_k]$ donc $\mathbb{P}(U = k) = \mathbb{P}(f_{k-1} < U \leqslant f_k) = F_U(f_k) - F_U(f_{k-1}) = f_k - f_{k-1} = e^{-a}\dfrac{a^k}{k!}$ et donc U suit une loi de Poisson de paramètre a.

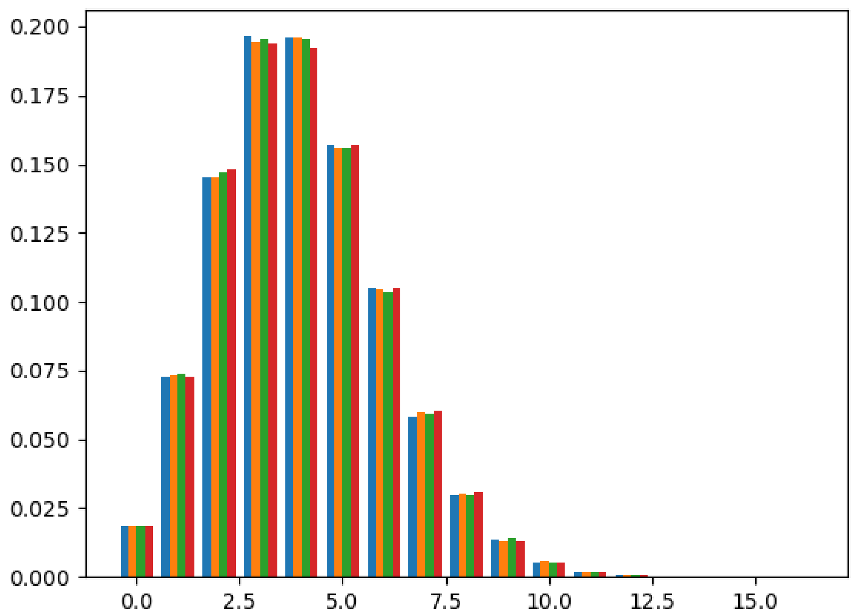
c) On remarque (on l'a déjà fait plusieurs fois que)

$$\forall k \in \mathbb{N}, \qquad \mathbb{P}(Y = k) = \frac{a}{k}\mathbb{P}(Y = k-1).$$

Ainsi, on initialise des variables `p` et `f` les valeurs de $\mathbb{P}(Y = 0)$ et f_0 qui valent e^{-a}. On stocke dans `U` une réalisation de U et, tant que `U>f`, on calcule le f_j suivant grâce à la formule ci-dessus. Voici :

```
def Simul_3(a):
    p=np.exp(-a)
    f=p
    k=0
    U=rd.random()
    while U>f:#Tant que f ne dépasse pas U
        k=k+1#On passe au rang suivant
        p=p*a/k#On calcule la proba suivante
        f=f+p#On calcule le fj suivant
    return k
```

3) On obtient



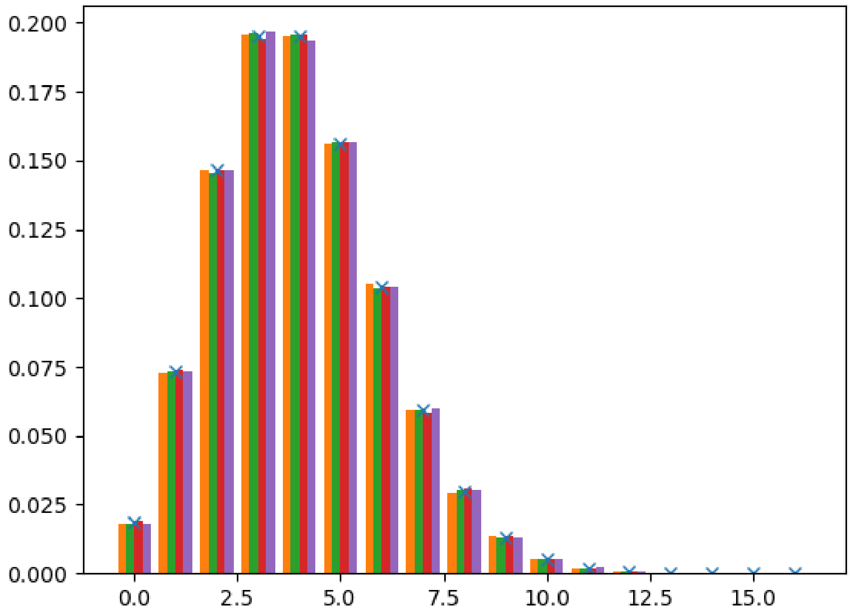
`0.3740048408508301 0.4815382957458496 0.7002227306365967 0.13211798667907715.`

On constate que, de ces quatre méthodes de simulation, c'est la quatrième (celle fournir par Python avec `rd.random()`) qui est la plus rapide. C'est ensuite la première qui est la plus rapide, puis la deuxième, puis la troisième.

Pour y voir plus clair sur la bonne approximation de la loi de Poisson, superposons à cet histogramme une petite croix aux points de coordonnées $\left(k, e^{-a}\dfrac{a^k}{k!}\right)$, $0 \leqslant k \leqslant M$:

```
T=np.zeros(M+1)
T[0]=np.exp(-a)
for k in range(M):
    T[k+1]=T[k]*a/(k+1)
plt.plot(range(M+1),T,'x')
```

On obtient :



On constate que les 4 méthodes semblent bien fournir une bonne simulation d'une variable aléatoire de loi de Poisson.

Exercice 8.

1) Soit $p \in]0\,;1[$. Soit $(X_n)_{n\geqslant 1}$ une suite de variables aléatoires indépendantes de loi $\mathcal{B}(p)$. Notons m et σ^2 l'espérance et la variance d'une loi $\mathcal{B}(p)$.
   a) Justifier que, pour tous $n \in \mathbb{N}^*$ et $x \in \mathbb{R}$,
   $$\mathbb{P}\left(\sqrt{n}\,\frac{\overline{X}_n - m}{\sigma} \leqslant x\right) = F_{n,p}\left(x\sqrt{np(1-p)} + np\right),$$
   où $F_{n,p}$ désigne la fonction de répartition d'une variable aléatoire de loi $\mathcal{B}(n,p)$.
   b) Écrire une fonction Python qui prend en entrée n, p et un réel t et qui renvoie $F_{n,p}(t)$.
   c) Avec Python, tracer la courbe représentative de $x \longmapsto F_{n,p}\left(x\sqrt{np(1-p)} + np\right)$ sur l'intervalle $[-4\,;4]$ avec $n = 1000$ et $p = 0,3$.
   d) Lui superposer la courbe de Φ, la fonction de répartition d'une variable aléatoire de loi $\mathcal{N}(0,1)$. Commenter.
   On rappelle que, si on réalise l'importation suivante : `import scipy.special as sp`, *alors la commande* `sp.ndtr` *est une fonction Python qui implémente* Φ.
2) Soit $a \in \mathbb{R}_+^*$. Soit $(X_n)_{n\geqslant 1}$ une suite de variables aléatoires indépendantes de loi $\mathcal{P}(a)$. Notons m et σ^2 l'espérance et la variance d'une loi $\mathcal{P}(a)$.
   a) Justifier que, pour tous $n \in \mathbb{N}^*$ et $x \in \mathbb{R}$,
   $$\mathbb{P}\left(\sqrt{n}\,\frac{\overline{X}_n - m}{\sigma} \leqslant x\right) = F_{na}\left(x\sqrt{na} + na\right),$$
   où, pour tout $b \in \mathbb{R}_+^*$, F_b désigne la fonction de répartition d'une variable aléatoire de loi $\mathcal{P}(b)$.
   b) Écrire une fonction Python qui prend en entrée b et un réel t et qui renvoie $F_b(t)$.
   c) Avec Python, tracer la courbe représentative de $x \longmapsto F_{na}\left(x\sqrt{na} + na\right)$ sur l'intervalle $[-4\,;4]$ avec $n = 100$ et $a = 3$.
   d) Lui superposer la courbe de Φ, la fonction de répartition d'une variable aléatoire de loi $\mathcal{N}(0,1)$. Commenter.

Correction :

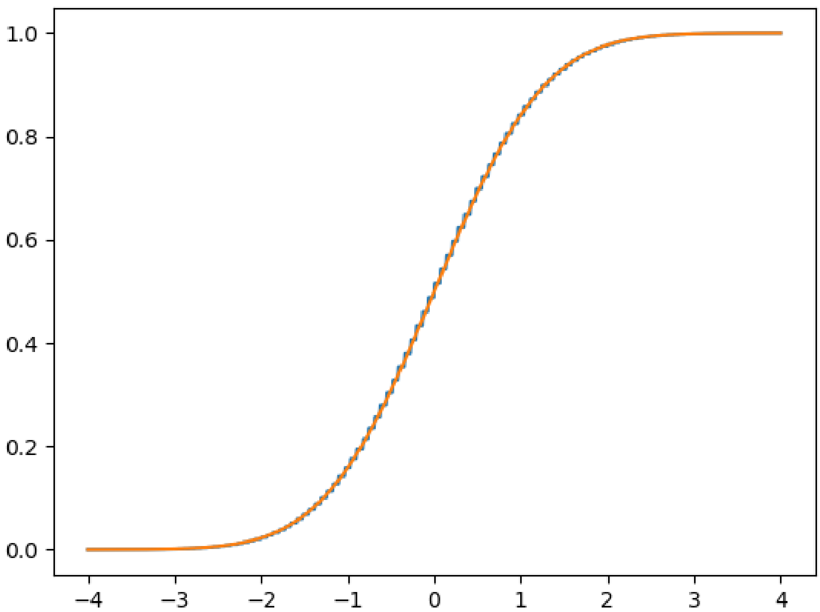
1) a) Soient $n \in \mathbb{N}^*$ et $x \in \mathbb{R}$. La variable aléatoire $n\overline{X}_n = X_1 + \cdots = X_n$ suit une loi $\mathcal{B}(n,p)$ donc

$$\mathbb{P}\left(\sqrt{n}\,\frac{\overline{X}_n - m}{\sigma} \leqslant x\right) = \mathbb{P}\left(\overline{X}_n \leqslant \frac{x\sigma}{\sqrt{n}} + m\right) \mathbb{P}\left(n\overline{X}_n \leqslant x\sqrt{np(1-p)} + np\right) = F_{n,p}\left(x\sqrt{np(1-p)} + np\right).$$

b)
```
import numpy as np

def FRepBin(n,p,t):
    if t<0 or t>n:
        return 0
    else:
        prob=(1-p)**n
        F=prob
        for k in range(1,int(t)+1):
            prob=prob*(n-k+1)*p/(k*(1-p))
            F=F+prob
        return F
```

c)
```
import matplotlib.pyplot as plt

def ApproxPhi(n,p,x):
    y=x*np.sqrt(n*p*(1-p))+n*p
    return FRepBin(n,p,y)

abs=np.linspace(-4,4,1000)
ord=[ApproxPhi(1000,0.3,x) for x in abs]
plt.plot(abs,ord)
```

d)
```
import scipy.special as sp
ord2=[sp.ndtr(x) for x in abs]
plt.plot(abs,ord2)
plt.show()
```



2) a) Soient $n \in \mathbb{N}^*$ et $x \in \mathbb{R}$. La variable aléatoire $n\overline{X}_n = X_1 + \cdots = X_n$ suit une loi $\mathcal{P}(na)$ donc

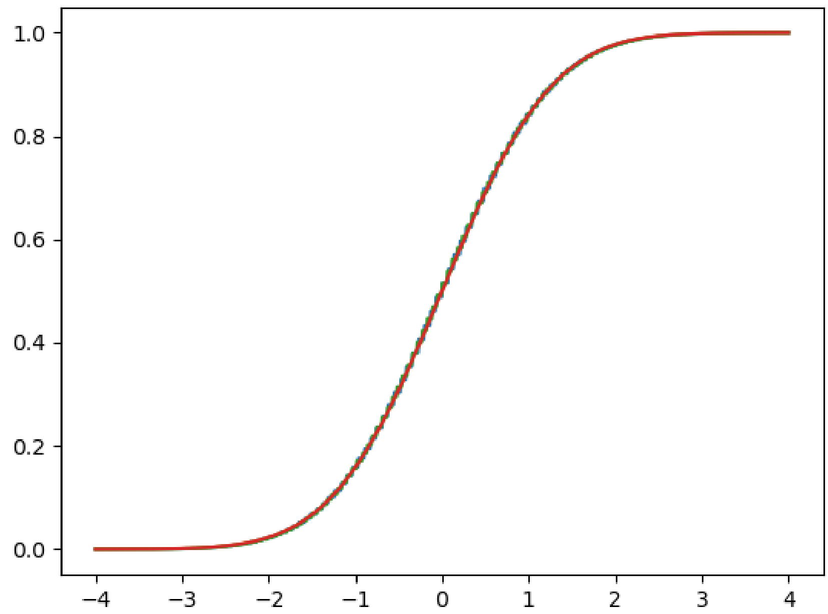
$$\mathbb{P}\left(\sqrt{n}\,\frac{\overline{X}_n - m}{\sigma} \leqslant x\right) = \mathbb{P}\left(\overline{X}_n \leqslant \frac{x\sigma}{\sqrt{n}} + m\right) \mathbb{P}\left(n\overline{X}_n \leqslant x\sqrt{na} + na\right) = F_{na}\left(x\sqrt{na} + na\right).$$

b)
```
import numpy as np

def FRepPoi(b,t):
```

```
    if t<0:
        return 0
    else:
        prob=np.exp(-b)
        F=prob
        for k in range(1,int(t)+1):
            prob=prob*b/k
            F=F+prob
        return F
```

c)
```
import matplotlib.pyplot as plt

def ApproxPhi(n,a,x):
    y=x*np.sqrt(n*a)+n*a
    return FRepPoi(n*a,y)

abs=np.linspace(-4,4,1000)
ord=[ApproxPhi(100,3,x) for x in abs]
plt.plot(abs,ord)
```

d)
```
import scipy.special as sp
ord2=[sp.ndtr(x) for x in abs]
plt.plot(abs,ord2)
plt.show()
```



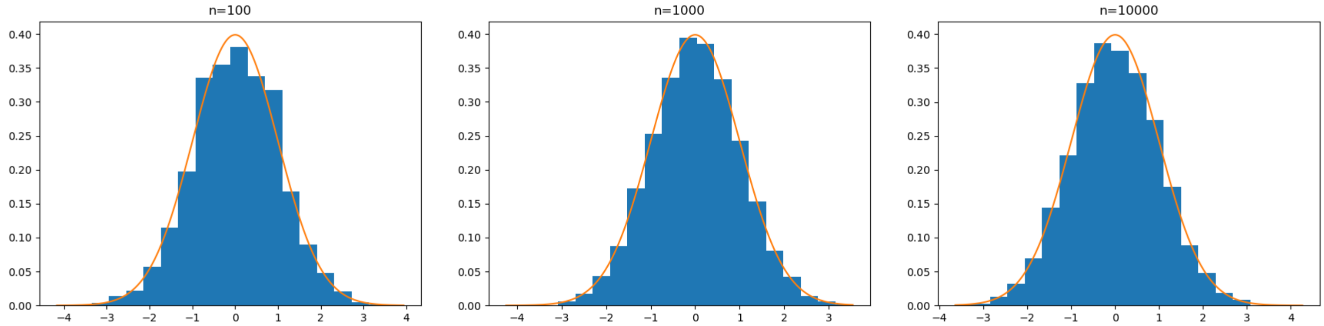
Exercice 9.

1) Écrire une fonction en Python qui prend en argument $n \in \mathbb{N}^*$ et le(s) paramètre(s) d'une certaine loi de probabilité et qui :
   - Stocke $N = 10000$ réalisations de $Y = \sqrt{n}\,\dfrac{\overline{X}_n - m}{\sigma}$, avec $\overline{X}_n = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} X_i$ où $X_1, \ldots, X_n$ sont des variables aléatoires indépendantes de la loi en question, dans un vecteur `Y`.
   - Trace l'histogramme renormalisé des valeurs de `Y` avec 20 classes via la commande : `plt.hist(Y,bins=20,density=True)`
   - Lui superpose la courbe de la densité de la loi Normale.
2) Tester avec $n = 100$, $n = 1000$, $n = 10000$ et les lois usuelles du programme (uniforme, Bernoulli, binomiale, géométrique, Poisson, exponentielle, gamma, Normale). Commenter.

Correction :

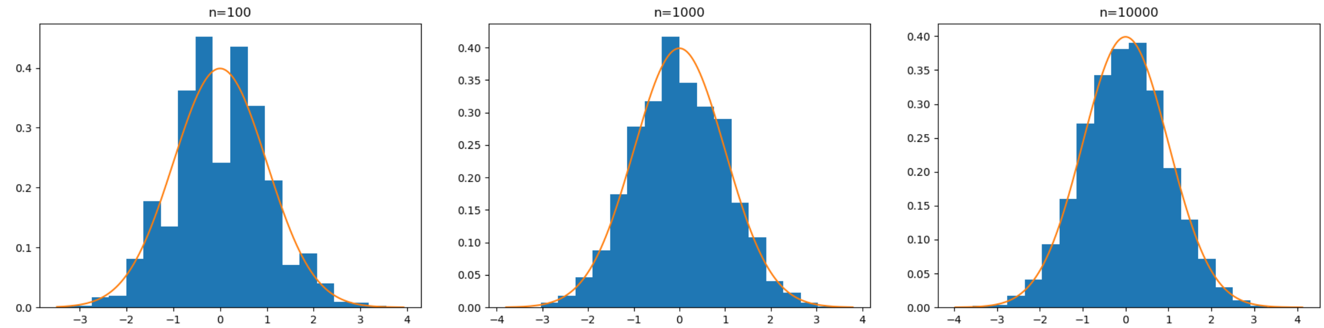
- Pour la loi uniforme discrète :

```
def TraceHistoUnid(n,a,b):
    m=(a+b)/2;  sig=np.sqrt((b-a)*(b-a+2)/12)
    N=10000
    Y=[]
    for k in range(N):
        X=rd.randint(a,b+1,n)
        Y.append(np.sqrt(n)*(np.mean(X)-m)/sig)
    plt.hist(Y,bins=20,density=True)
    Abs=np.linspace(np.min(Y),np.max(Y),1000)
    Ord=[np.exp(-x**2/2)/np.sqrt(2*np.pi) for x in Abs]
    plt.plot(Abs,Ord)
    plt.title('n='+str(n))
    plt.show()

plt.figure();  TraceHistoUnid(100,-5,9)
plt.figure();  TraceHistoUnid(1000,-5,9)
plt.figure();  TraceHistoUnid(10000,-5,9)
```



- Pour la loi de Bernoulli :

```
def TraceHistoBer(n,p):
    m=p;  sig=np.sqrt(p*(1-p))
    N=10000
    Y=[]
    for k in range(N):
        X=rd.binomial(1,p,n)
        Y.append(np.sqrt(n)*(np.mean(X)-m)/sig)
    plt.hist(Y,bins=20,density=True)
    Abs=np.linspace(np.min(Y),np.max(Y),1000)
    Ord=[np.exp(-x**2/2)/np.sqrt(2*np.pi) for x in Abs]
    plt.plot(Abs,Ord)
    plt.title('n='+str(n))
    plt.show()

plt.figure();  TraceHistoBer(100,0.3)
plt.figure();  TraceHistoBer(1000,0.3)
plt.figure();  TraceHistoBer(10000,0.3)
```



- etc.

Je vous laisse essayer pour chaque loi (l'occasion de réviser les commandes au programme). On obtient des histogrammes similaires à chaque fois.

Exercice 10.

1) Montrer que Φ est une bijection de $\mathbb{R}$ dans $]0\,;1[$.
2) Soit $U \hookrightarrow \mathcal{U}(]0\,;1[)$. Montrer que $Y = \Phi^{-1}(U)$ suit une loi $\mathcal{N}(0,1)$.
3) a) A l'aide d'un algorithme de dichotomie, écrire une fonction qui prend en entrée t et qui calcule $\Phi^{-1}(t)$. *On cherchera l'unique antécédent de t entre -5 et 5.*
   b) En déduire une fonction Python, intitulé `NormalInvRep`, sans argument renvoie une simulation d'une variable aléatoire de loi $\mathcal{N}(0,1)$ avec cette méthode.

Correction :

1) La fonction Φ est continue (c'est la fonction de répartition d'une variable à densité) strictement croissante (sa dérivée $x \longmapsto \dfrac{1}{\sqrt{2\pi}}e^{-x^2/2}$ est une fonction strictement positive) sur $\mathbb{R}$, $\lim\limits_{x\to-\infty} \Phi = 0$ et $\lim\limits_{x\to+\infty} \Phi = 1$ (car c'est une fonction de répartition). Ainsi le théorème de la bijection entraîne que Φ est une bijection de $\mathbb{R}$ dans $]0\,;1[$.
2) Soit $x \in \mathbb{R}$. Puisque Φ est strictement croissante, on a

$$\mathbb{P}(Y \leqslant x) = \mathbb{P}(U \leqslant \Phi(x)) = \Phi(x)$$

   puisque $\Phi(x) \in\,]0\,;1[$ et que $U \hookrightarrow \mathcal{U}(]0\,;1[)$. Ainsi Y admet Φ pour fonction de répartition. Comme la fonction de répartition caractérise la loi, on a bien $Y \hookrightarrow \mathcal{N}(0,1)$.
3) a)

```python
import scipy.special as sp

def InvPhi(t):
    a=-5
    b=5
    while b-a>0.0001:
        c=(a+b)/2
        if sp.ndtr(c)>t:
            b=c
        else:
            a=c
    return a
```

   b)

```python
import numpy.random as rd

def NormalInvRep():
    U=rd.random()
    return InvPhi(U)
```

Exercice 11 (Méthode de Box-Muller). On considère deux variables aléatoires U et V définies sur un même espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$, indépendantes et de même loi uniforme sur $]0\,;1]$.

1) a) Montrer que $\sin$ réalise une bijection de $\left[-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right]$ sur $[-1\,;1]$. On note Arcsin sa bijection réciproque.
   b) Montrer que Arcsin est dérivable sur $]-1\,;1[$ et que

$$\forall x \in\,]-1\,;1[, \qquad \mathrm{Arcsin}'(x) = \frac{1}{\sqrt{1-x^2}}.$$

   c) Montrer que Arcsin est impaire.
2) On pose $X = \ln(\sqrt{-2\ln(U)})$. On admet que c'est une variable aléatoire sur $(\Omega, \mathcal{A}, \mathbb{P})$.
   a) Calculer la fonction de $-2\ln(U)$
   b) Montrer que X est une variable aléatoire à densité donc une densité est $g : x \longmapsto \exp\left(2x - \dfrac{e^{2x}}{2}\right)$.
3) On pose $Y = \sin(2\pi V)$ et $Z = \ln(|Y|)$. On admet que Y et Z sont des variables aléatoires sur $(\Omega, \mathcal{A}, \mathbb{P})$.
   a) Montrer que, pour tout $x \in\,]0\,;1]$, $[Y \geqslant x] = [\mathrm{Arcsin}(x) \leqslant 2\pi V \leqslant \pi - \mathrm{Arcsin}(x)]$.

b) Montrer que, pour tout $x \in [-1\,;0]$, $[Y \leqslant x] = [\pi - \mathrm{Arcsin}(x) \leqslant 2\pi V \leqslant 2\pi + \mathrm{Arcsin}(x)]$.

c) En déduire la fonction de répartition de Y.

d) Montrer que Z admet pour densité la fonction

$$h : x \longmapsto \begin{cases} \dfrac{2}{\pi}\dfrac{e^x}{\sqrt{1-e^{2x}}} & \text{si} \quad x < 0 \\ 0 & \text{si} \quad x \geqslant 0 \end{cases}$$

4) Justifier que la variable aléatoire $S = X + Z$ admet une densité et donner en une densité sous forme d'une intégrale.

5) On pose $T = \sqrt{-2\ln(U)}\sin(2\pi V)$. On admet que c'est une variable aléatoire sur $(\Omega, \mathcal{A}, \mathbb{P})$. Remarquons que $\ln(|T|) = S$.

a) Montrer qu'une densité de $|T|$ sur R est

$$f : x \longmapsto \begin{cases} \dfrac{2}{\pi}\displaystyle\int_{\ln(x)}^{+\infty} \dfrac{\exp\left(-e^{2t}/2\right)}{\sqrt{e^{2t}-x^2}} e^{2t}\,dt & \text{si} \quad x > 0 \\ 0 & \text{si} \quad x \leqslant 0 \end{cases}$$

b) Soit $x > 0$. A l'aide du changement de variable $t = \dfrac{1}{2}\ln(x^2+u^2)$, établir que $f(x) = \dfrac{2}{\sqrt{2\pi}}e^{-x^2/2}$.

6) Notons F et F_T les fonctions de répartition respectives de $|T|$ et T.

a) Montrer que $-T = \sqrt{-2\ln(U)}\sin(2\pi(1-V))$. En déduire que T et $-T$ ont la même loi.

b) En déduire que, pour tout $x \in \mathbb{R}$, $\mathbb{P}(T = x) = 0$.

c) Montrer que, pour tout $x \in \mathbb{R}_+$, $F_T(x) = \dfrac{1+F(x)}{2}$.

d) Montrer que, pour tout $x \in \mathbb{R}_-^*$, $F_T(x) = \dfrac{1-F(-x)}{2}$.

e) Conclure que T est bien une variable à densité.

f) En déduire que T suit une loi $\mathcal{N}(0,1)$.

7) Proposer une méthode de simulation d'une variable aléatoire de loi $\mathcal{N}(0,1)$.

Correction :

1) a) Ultra classique : $\sin$ est continue et strictement croissante sur $\left[-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right]$ donc le théorème de la bijection assure qu'elle réalise une bijection de $\left[-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right]$ sur $\sin\left(\left[-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right]\right) = \left[\sin\left(-\dfrac{\pi}{2}\right)\,;\sin\left(\dfrac{\pi}{2}\right)\right] = [-1\,;1]$.

b) La fonction $\sin$ est dérivable sur $\left]-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right[$ et $\sin' = \cos$ ne s'annule pas sur $\left]-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right[$ donc $\mathrm{Arcsin} = \sin^{-1}$ est dérivable sur $\sin\left(\left]-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right[\right) =]-1\,;1[$. De plus, pour tout $x \in]-1\,;1[$,

$$\mathrm{Arcsin}'(x) = \frac{1}{\sin'(\mathrm{Arcsin}(x))} = \frac{1}{\cos(\mathrm{Arcsin}(x))}.$$

On a $\cos^2(\mathrm{Arcsin}(x)) = 1 - \sin^2(\mathrm{Arcsin}(x)) = 1 - x^2$ donc $\cos(\mathrm{Arcsin}(x)) = \sqrt{1-x^2}$, puisque $\cos$ est positive sur $\left]-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right[$ et $\mathrm{Arcsin}(x) \in \left]-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right[$. Ainsi

$$\forall x \in]-1\,;1[\,, \qquad \mathrm{Arcsin}'(x) = \frac{1}{\sqrt{1-x^2}}.$$

c) Pour tout $x \in [-1\,;1]$, $-x \in [-1\,;1]$ et

$$\sin(-\mathrm{Arcsin}(x)) = -\sin(\mathrm{Arcsin}(x)) = -x = \sin(\mathrm{Arcsin}(-x)).$$

Comme $\sin$ est injective sur $\left[-\dfrac{\pi}{2}\,;\dfrac{\pi}{2}\right]$, on en déduit que $-\mathrm{Arcsin}(x) = \mathrm{Arcsin}(-x)$. Ainsi Arcsin est impaire.

2) a) Soit $x \in \mathbb{R}$. On a

$$\mathbb{P}(-2\ln(U) \leqslant x) = \mathbb{P}(U \geqslant e^{-x/2}) = 1 - F_U(e^{-x/2}) = 1 - \begin{cases} 0 & \text{si} \quad e^{-x/2} < 0 \\ e^{-x/2} & \text{si} \quad 0 \leqslant e^{-x/2} \leqslant 1 \\ 1 & \text{si} \quad e^{-x/2} > 0 \end{cases}$$

Donc

$$\mathbb{P}(-2\ln(U) \leqslant x) = \begin{cases} 1 - e^{-x/2} & \text{si} \quad x \geqslant 0 \\ 0 & \text{si} \quad x < 0 \end{cases}$$

On reconnaît que $-2\ln(U)$ suit une variable de loi $\mathcal{E}(1/2)$.

b) Soit $x \in \mathbb{R}$. On a

$$F_Z(x) = \mathbb{P}(\sqrt{-2\ln(U)} \leqslant e^x) = \mathbb{P}(-2\ln(U) \leqslant e^{x/2}) = 1 - e^{-e^{2x}/2}.$$

La fonction F_Z est de classe C^1 sur $\mathbb{R}$ donc Z est une variable aléatoire à densité. Une densité est :

$$x \longmapsto F_Z'(x) = -\left(-\frac{1}{2} \times 2e^{2x}\right))e^{-e^{2x}/2} = \exp\left(2x - \frac{e^{2x}}{2}\right).$$

3) On pose $Y = \sin(2\pi V)$ et $Z = \ln(|Y|)$. On admet que Y et Z sont des variables aléatoires sur $(\Omega, \mathcal{A}, \mathbb{P})$.

a) ⚠ *Attention à une subtilité (qui est très classique pour la fonction* Arctan*) :* $\text{Arcsin}(\sin(x)) \neq x$ *a priori. Ce n'est le cas que lorsque* $x \in \left[-\frac{\pi}{2}\,;\frac{\pi}{2}\right]$ *donc il faut « s'y ramener » lorsque ce n'est pas le cas.*

Soit $x \in]0\,;1]$. On a $V(\Omega) =]0\,;1]$ donc $(2\pi V)(\Omega) =]0\,;2\pi]$. Soit $\omega \in \Omega$. Notons $v = V(\omega)$ et $y = Y(\omega)$.

- Supposons que $y \geqslant x$. On a alors $y > 0$ et donc $2\pi v \in]0\,;\pi]$. Comme Arcsin est croissante sur $[-1\,;1]$, on a alors $\text{Arcsin}(y) \geqslant \text{Arcsin}(x)$.
  - Si $2\pi v \in \left]0\,;\frac{\pi}{2}\right]$, alors $\text{Arcsin}(y) = \text{Arcsin}(\sin(2\pi v)) = 2\pi v$. Ainsi $\text{Arcsin}(x) \leqslant 2\pi v$.
  - Si $2\pi v \in \left[\frac{\pi}{2}\,;\pi\right[$, alors $\pi - 2\pi v \in \left[0\,;\frac{\pi}{2}\right]$ donc

  $$\text{Arcsin}(y) = \text{Arcsin}(\sin(2\pi v)) = \text{Arcsin}(\sin(\pi - 2\pi v)) = \pi - 2\pi v$$

  et donc $\pi - 2\pi v \geqslant \text{Arcsin}(x)$ et donc $\pi - \text{Arcsin}(x) \geqslant 2\pi v$.

  Ainsi $\text{Arcsin}(x) \leqslant 2\pi v \leqslant \pi - \text{Arcsin}(x)$.
- Réciproquement, supposons que $\text{Arcsin}(x) \leqslant 2\pi v \leqslant \pi - \text{Arcsin}(x)$. On a alors $0 \leqslant 2\pi v \leqslant \pi$.
  - Si $2\pi v \leqslant \frac{\pi}{2}$ alors, par croissance de $\sin$ sur $\left]0\,;\frac{\pi}{2}\right]$, $x \leqslant \sin(2\pi v) = y$.
  - Si $2\pi v \geqslant \frac{\pi}{2}$ alors, par décroissance de $\sin$ sur $\left[\frac{\pi}{2}\,;\pi\right]$,

  $$y = \sin(2\pi v) \geqslant \sin(\pi - \text{Arcsin}(x)) = \sin(\text{Arcsin}(x)) = x.$$

On a bien $y \geqslant x$ si et seulement si $\text{Arcsin}(x) \leqslant 2\pi v \leqslant \pi - \text{Arcsin}(x)$.
Ainsi $[Y \geqslant x] = [\text{Arcsin}(x) \leqslant 2\pi V \leqslant \pi - \text{Arcsin}(x)]$.
On peut aussi dessiner un cercle trigonométrique pour y voir plus clair.

b) Soit $x \in [-1\,;0]$. On a $V(\Omega) =]0\,;1]$ donc $(2\pi V)(\Omega) =]0\,;2\pi]$. Soit $\omega \in \Omega$. Notons $v = V(\omega)$ et $y = Y(\omega)$.

- Supposons que $y \leqslant x$. On a alors $y \leqslant 0$ et donc $2\pi v \in [\pi\,;2\pi]$ donc $2\pi - 2\pi v \in [0\,;\pi]$.
  - Si $x \neq 0$, en appliquant le raisonnement de la question précédente à $2\pi - 2\pi v$ au lieu de v et à $-x$, on obtient $\text{Arcsin}(-x) \leqslant 2\pi - 2\pi v \leqslant \pi - \text{Arcsin}(-x)$ donc $-\text{Arcsin}(x) \leqslant 2\pi - 2\pi v \leqslant \pi + \text{Arcsin}(x)$ donc $-2\pi - \text{Arcsin}(x) \leqslant -2\pi v \leqslant -\pi + \text{Arcsin}(x)$ et donc $2\pi + \text{Arcsin}(x) \geqslant 2\pi v \leqslant \pi - \text{Arcsin}(x)$.
  - Si $x = 0$, $\text{Arcsin}(x) = 0$ et on a bien $\pi - 0 \leqslant 2\pi v \leqslant 2\pi + 0$.
- Réciproquement, supposons que $\pi - \text{Arcsin}(x) \leqslant 2\pi v \leqslant 2\pi + \text{Arcsin}(x)$. On a alors $\pi \leqslant 2\pi v \leqslant 2\pi$.

— Si $2\pi v \leqslant \dfrac{3\pi}{2}$ alors, par décroissance de $\sin$ sur $\left]\pi\,;\dfrac{3\pi}{2}\right]$,

$$x = \sin(\mathrm{Arcsin}(x)) = \sin(\pi - \mathrm{Arcsin}(x)) \geqslant \sin(2\pi v) = y.$$

— Si $2\pi v \geqslant \dfrac{3\pi}{2}$ alors, par croissance de $\sin$ sur $\left[\dfrac{3\pi}{2}\,;2\pi\right]$,

$$y = \sin(2\pi v) \leqslant \sin(2\pi + \mathrm{Arcsin}(x)) = \sin(\mathrm{Arcsin}(x)) = x.$$

On a bien $y \leqslant x$ si et seulement si $\pi - \mathrm{Arcsin}(x) \leqslant 2\pi v \leqslant 2\pi + \mathrm{Arcsin}(x)$.
Ainsi $[Y \leqslant x] = [\pi - \mathrm{Arcsin}(x) \leqslant 2\pi V \leqslant 2\pi + \mathrm{Arcsin}(x)]$.

c) • Si $x < -1$, alors $\mathbb{P}(Y \leqslant x) = 0$.

• Si $x \in [-1\,;1]$,

$$\mathbb{P}(Y \leqslant x) = \mathbb{P}(\pi - \mathrm{Arcsin}(x) \leqslant 2\pi V \leqslant 2\pi + \mathrm{Arcsin}(x)) = \mathbb{P}\left(\frac{1}{2} - \frac{\mathrm{Arcsin}(x)}{2\pi} \leqslant V \leqslant 1 + \frac{\mathrm{Arcsin}(x)}{2\pi}\right).$$

On a $-1 \leqslant x \leqslant 0$ donc $-\dfrac{\pi}{2} \leqslant \mathrm{Arcsin}(x) \leqslant 0$ donc $-\dfrac{1}{4} < \dfrac{\mathrm{Arcsin}(x)}{2\pi} \leqslant 0$. Ainsi $\dfrac{1}{2} \leqslant \dfrac{1}{2} - \dfrac{\mathrm{Arcsin}(x)}{2\pi} \leqslant \dfrac{3}{4}$ et $\dfrac{3}{4} < \dfrac{\mathrm{Arcsin}(x)}{2\pi} \leqslant 1$. Par conséquent

$$\mathbb{P}(Y \leqslant x) = 1 + \frac{\mathrm{Arcsin}(x)}{2\pi} - \left(\frac{1}{2} - \frac{\mathrm{Arcsin}(x)}{2\pi}\right) = \frac{1}{2} + \frac{\mathrm{Arcsin}(x)}{\pi}.$$

• Si $x \in \,]0\,;1[$,

$$\begin{aligned}\mathbb{P}(Y \leqslant x) = 1 - \mathbb{P}(Y > x) &= 1 - \mathbb{P}(\mathrm{Arcsin}(x) \leqslant 2\pi V \leqslant \pi - \mathrm{Arcsin}(x)) \\ &= 1 - \mathbb{P}\left(\frac{\mathrm{Arcsin}(x)}{2\pi} \leqslant V \leqslant \frac{1}{2} - \frac{\mathrm{Arcsin}(x)}{2\pi}\right).\end{aligned}$$

On a $0 < x \leqslant 1$ donc $0 < \mathrm{Arcsin}(x) \leqslant \dfrac{\pi}{2}$ donc $0 < \dfrac{\mathrm{Arcsin}(x)}{2\pi} \leqslant \dfrac{1}{4}$ et $\dfrac{1}{4} \leqslant \dfrac{1}{2} - \dfrac{\mathrm{Arcsin}(x)}{2\pi} < \dfrac{1}{2}$. On a donc

$$\mathbb{P}(Y \leqslant x) = 1 - \left(\frac{1}{2} - \frac{\mathrm{Arcsin}(x)}{2\pi} - \frac{\mathrm{Arcsin}(x)}{2\pi}\right) = \frac{1}{2} + \frac{\mathrm{Arcsin}(x)}{\pi}.$$

• Si $x > 1$, alors $\mathbb{P}(Y \leqslant x) = 1$.

On résume :

$$\forall x \in \mathbb{R}, \qquad F_Y(x) = \begin{cases} 0 & \text{si} \quad x < -1 \\ \dfrac{1}{2} + \dfrac{\mathrm{Arcsin}(x)}{\pi} & \text{si} \quad -1 \leqslant x \leqslant 1 \\ 1 & \text{si} \quad x > 1 \end{cases}$$

d) Soit $x \in \mathbb{R}$. On a

$$F_Z(x) = \mathbb{P}(|Y| \leqslant e^x) = \mathbb{P}(-e^x \leqslant Y \leqslant e^x) = F_Y(e^x) - F_Y(-e^{-x})$$

• Si $x \geqslant 0$, alors $e^x \geqslant 1$ et $-e^x \leqslant -1$ donc $F_Z(x) = 1 - 0 = 1$.

• Si $x \leqslant 0$, alors $0 < e^x < 1$ et $-1 < -e^x < 0$ donc

$$F_Z(x) = \frac{1}{2} + \frac{\mathrm{Arcsin}(e^x)}{\pi} - \frac{1}{2} - \frac{\mathrm{Arcsin}(-e^x)}{\pi} = \frac{2}{\pi}\,\mathrm{Arcsin}(e^x).$$

La fonction F_Z est C^1 sur $\mathbb{R}$ sauf peut-être en 0. De plus $\dfrac{2}{\pi}\,\mathrm{Arcsin}(e^x) \underset{x\to 0}{\longrightarrow} \dfrac{2}{\pi}\,\mathrm{Arcsin}(1) = 1$, par continuité de $\exp$ en 0 et de Arcsin en 1. Ainsi F_Z est continue sur $\mathbb{R}$ tout entier. On en déduit que Z est à densité. Une densité est alors

$$x \longmapsto \begin{cases} F_Z'(x) & \text{si} \quad x < 0 \\ 0 & \text{si} \quad x \geqslant 0 \end{cases} = \begin{cases} \dfrac{2}{\pi} e^x \,\mathrm{Arcsin}'(e^x) & \text{si} \quad x < 0 \\ 0 & \text{si} \quad x \geqslant 0 \end{cases} = \begin{cases} \dfrac{2}{\pi}\dfrac{e^x}{\sqrt{1-e^{2x}}} & \text{si} \quad x < 0 \\ 0 & \text{si} \quad x \geqslant 0 \end{cases}$$

4) On a $2x - \dfrac{e^{2x}}{2} = e^{2x}\left(2xe^{-2x} - \dfrac{1}{2}\right) \xrightarrow[x\to+\infty]{} -\infty$ par croissances comparées. On a $2x - \dfrac{e^{2x}}{2} \xrightarrow[x\to-\infty]{} -\infty$. Ainsi g tend vers 0 en $+\infty$ et $-\infty$. Comme g est continue sur $\mathbb{R}$, on en déduit que g est bornée. Comme X et Z sont indépendantes (par théorème des coalitions puisque V et U sont indépendantes). Par conséquent $S = X + Z$ admet une densité donnée par la fonction

$$\varphi : x \longmapsto \int_{-\infty}^{+\infty} g(u)h(x-u)\,du$$

Pour tout $u \in \mathbb{R}$, on a $h(x-u) \neq 0$ si et seulement si $x - u \leqslant 0$ si et seulement si $u \geqslant x$. Ainsi

$$\varphi : x \longmapsto \int_{x}^{+\infty} e^{2u-e^{2u}/2}\frac{e^{x-u}}{\sqrt{1-e^{2(x-u)}}}\,du.$$

5) a) Pour tout $x \in \mathbb{R}_-$, $F_{|T|}(x) = 0$. Si $x \in \mathbb{R}_+^*$, alors

$$F_{|T|}(x) = \mathbb{P}(|T| \leqslant x) = \mathbb{P}(S \leqslant \ln(x)) = F_S(\ln(x)).$$

Puisque $\ln$ est de classe C^1 sur $\mathbb{R}_+^*$ et F_Z de classe C^1 sur $\mathbb{R}$ sauf peut-être en un nombre fini de points, $F_Z \circ \ln$ est de classe C^1 sur $\mathbb{R}_+^*$ sauf peut-être en un nombre fini de points. Ainsi $F_{|T|}$ est de classe C^1 sur $\mathbb{R}$ sauf peut-être en un nombre fini de points. Puisque F_Z est continue sur $\mathbb{R}$, $F_{|T|}$ est continue sur $\mathbb{R}$ sauf peut-être en 0. Mais on a $F_S(y) \xrightarrow[y\to-\infty]{} 0$ et $\ln(x) \xrightarrow[x\to0^+]{} -\infty$ si bien que $F_{|T|}(x) \xrightarrow[x\to0^+]{} 0$. Ainsi $F_{|T|}$ est continue sur $\mathbb{R}$. On en déduit que $|T|$ est à densité. Une densité de $|T|$ est

$$f : x \longmapsto \begin{cases} \dfrac{\varphi(\ln(x))}{x} & \text{si} \quad x > 0 \\ 0 & \text{si} \quad x \leqslant 0 \end{cases}$$

Si $x > 0$, alors

$$\frac{\varphi(\ln(x))}{x} = \frac{1}{x}\int_{\ln(x)}^{+\infty} e^{2u-e^{2u}/2}\frac{e^{\ln(x)-u}}{\sqrt{1-e^{2(\ln(x)-u)}}}\,du = \frac{1}{x}\int_{\ln(x)}^{+\infty} e^{2u-e^{2u}/2}\frac{xe^{-u}}{\sqrt{1-x^2e^{-2u}}}\,du.$$

On simplifie les x et on factorise par e^{-2u} dans la racine :

$$\frac{\varphi(\ln(x))}{x} = \int_{\ln(x)}^{+\infty} e^{2u-e^{2u}/2}\frac{e^{-u}}{e^{-u}\sqrt{e^{2u}-x^2}}\,du.$$

On obtient bien :

$$f : x \longmapsto \begin{cases} \dfrac{2}{\pi}\displaystyle\int_{\ln(x)}^{+\infty} \frac{\exp\left(-e^{2t}/2\right)}{\sqrt{e^{2t}-x^2}}e^{2t}\,dt & \text{si} \quad x > 0 \\ 0 & \text{si} \quad x \leqslant 0 \end{cases}$$

b) Soit $x > 0$. Faisons le changement de variable $t = \dfrac{1}{2}\ln(u^2+x^2)$ (i.e. $u^2 = e^{2t} - x^2$) donné par la fonction $x \longmapsto \dfrac{1}{2}\ln(u^2+x^2)$ qui est de classe C^1 et strictement croissante sur $]0\,;+\infty[$ car $e^{2t} - x^2 \xrightarrow[t\to\ln(x)]{} 0$ et $e^{2t} - x^2 \xrightarrow[t\to+\infty]{} +\infty$. On a « $dt = \dfrac{u}{u^2+x^2}$ »

$$f(x) = \frac{2}{\pi}\int_0^{+\infty} \frac{\exp\left(-e^{\ln(u^2+x^2)}/2\right)}{\sqrt{u^2}}e^{\ln(x^2+u^2)}\,\frac{u}{u^2+x^2}\,du = \frac{2}{\pi}\int_0^{+\infty} \exp\left(-(u^2+x^2)/2\right)\,du$$

donc

$$f(x) = \frac{2}{\pi}e^{-x^2/2}\int_0^{+\infty} e^{-u^2/2}\,du.$$

Par parité :

$$\int_0^{+\infty} e^{-u^2/2}\,du = \frac{1}{2}\int_{-\infty}^{+\infty} e^{-u^2/2}\,du = \frac{1}{2}\sqrt{2\pi}.$$

Et donc $f(x) = \dfrac{2}{\pi}e^{-x^2/2}\dfrac{1}{2}\sqrt{2\pi} = \dfrac{2}{\sqrt{2\pi}}e^{-x^2/2}$.

6) a) On a $\sin(2\pi(1-V)) = \sin(-2\pi V)$ puisque $\sin$ est 2π-périodique. On a donc $\sin(2\pi(1-V)) = -\sin(2\pi V)$ puisque $\sin$ est impaire. On en déduit que $-T = \sqrt{-2\ln(U)}\sin(2\pi(1-V))$. Puisque $1-V$ et V ont même loi, on en déduit que T et $-T$ ont la même loi.

b) Puisque $|T|$ est à densité, on a $\mathbb{P}(|T| = x) = 0$. Pour tout $x \in \mathbb{R}$,

- Si $x \geqslant 0$, $[T = x] \subset [|T| = x]$ donc $\mathbb{P}(T = x) \leqslant \mathbb{P}(|T| = x) = 0$ et donc $\mathbb{P}(T = x) = 0$.
- Si $x < 0$, $[T = x] \subset [|T| = -x]$ donc $\mathbb{P}(T = x) \leqslant \mathbb{P}(|T| = -x) = 0$ et donc $\mathbb{P}(T = x) = 0$.

c) Pour tout $x \in \mathbb{R}_+$,

$$\begin{aligned} F(x) = \mathbb{P}(|T| \leqslant x) = \mathbb{P}(-x \leqslant T \leqslant x) = \mathbb{P}(T \leqslant x) - \mathbb{P}(T \leqslant -x) &= \mathbb{P}(T \leqslant x) - \mathbb{P}(-T \geqslant x) \\ &= \mathbb{P}(T \leqslant x) - 1 + \mathbb{P}(-T < x). \end{aligned}$$

Puisque $-T$ et T ont la même loi et que $\mathbb{P}(-T = x) = 0$, on a $\mathbb{P}(-T < x) = \mathbb{P}(T \leqslant x) = F_T(x)$. Ainsi $F(x) = 2F_T(x) - 1$ et donc $F_T(x) = \dfrac{1 + F(x)}{2}$.

d) Pour tout $x \in \mathbb{R}_-^*$,

$$\begin{aligned} F(-x) = \mathbb{P}(|T| \leqslant -x) = \mathbb{P}(x \leqslant T \leqslant -x) = \mathbb{P}(T \leqslant -x) - \mathbb{P}(T \leqslant x) &= \mathbb{P}(-T \geqslant x) - \mathbb{P}(T \leqslant x) \\ &= 1 - \mathbb{P}(-T < x) - \mathbb{P}(T \leqslant x). \end{aligned}$$

Puisque $-T$ et T ont la même loi et que $\mathbb{P}(-T = -x) = 0$, on a $\mathbb{P}(-T < x) = \mathbb{P}(T \leqslant x) = F_T(x)$. Ainsi $F(-x) = 1 - 2F_T(x)$ et donc $F_T(x) = \dfrac{1 - F(-x)}{2}$.

e) On sait que F est continue sur $\mathbb{R}$ et C^1 sur $\mathbb{R}$ sauf en un nombre fini de points. Les questions précédentes entraînent que F_T est continue sur $\mathbb{R}$ sauf peut-être en 0 et qu'elle est C^1 sur $\mathbb{R}$ sauf en un nombre fini de points.

On a $F(0) = \displaystyle\int_{-\infty}^{0} f(t)\,dt = 0$.

- Si $x > 0$, $F_T(x) = \dfrac{1 + F(x)}{2} \xrightarrow[x \to 0]{} \dfrac{1 + F(0)}{2} = \dfrac{1}{2}$, par continuité de F en 0.
- Si $x < 0$, $F_T(x) = \dfrac{1 - F(-x)}{2} \xrightarrow[x \to 0]{} \dfrac{1 - F(0)}{2} = \dfrac{1}{2}$, par continuité de F en 0.

On en déduit que F est continue sur $\mathbb{R}$. On en déduit que T est bien une variable à densité.

f) En dérivant, on obtient qu'une densité de T est

$$x \longmapsto \begin{cases} \dfrac{f(x)}{2} & \text{si} \quad x > 0 \\ -\dfrac{-f(-x)}{2} & \text{si} \quad x \leqslant 0 \end{cases} = \frac{1}{\sqrt{2\pi}} e^{-x^2} 2$$

Ainsi T suit une loi $\mathcal{N}(0,1)$.

7) Il suffit de simuler la variable T en simulant U et V :

```
import numpy as np
import numpy.random as rd
def BoxMuller():
    U=rd.random()
    V=rd.random()
    return np.sqrt(-2*np.log(U))*np.sin(2*np.pi*V)
```

Exercice 12 (Comparaison des méthodes de simulation de variables aléatoires de loi $\mathcal{N}(0,1)$).
En s'inspirant du script de la dernière question de l'exercice 3, créer des vecteurs contenant 100000 réalisations de variables aléatoires indépendantes de loi $\mathcal{N}(0,1)$ obtenus chacun avec une des méthodes de simulations vues précédemment. Afficher un histogramme renormalisé et le temps d'exécution de chaque algorithme. Commenter.

Correction : A VENIR