Exercices Centrale 2 - PSI

Exercice 1

2018 - Centrale 2 - PSI

Soit $f : x \mapsto \sum_{n\geq 1} \frac{x^n}{n^2}$ et $g : x \mapsto \int_0^x \frac{\ln(1-t)}{t}\,\mathrm{d}t$.

1 Déterminer le domaine de définition de f puis tracer la courbe représentative de f à l'aide de Python à la précision 10^{-5}.

2 Montrer que g est définie sur $[-1\,;1]$, de classe $\mathcal{C}^1$ sur $]-1\,;1[$ puis tracer g à l'aide de Python. Que peut-on conjecturer ?

3 Calculer f' sur $]-1\,;1[$. En déduire que $f+g=0$ sur $]-1\,;1[$.

4 A l'aide d'une intégration par parties et d'un changement de variables, relier $g(x), g(1-x)$ et $g(1)$.

5 En déduire la valeur de $\sum_{n\geq 1} \frac{1}{2^n n^2}$.

Solution

1 f est la somme d'une série entière. Puisque $\frac{\frac{1}{(n+1)^2}}{\frac{1}{n^2}} = \frac{n^2}{(n+1)^2} \to 1$, la règle de D'Alembert assure que la rayon de convergence de cette série entière est 1.

De plus on observe que la série est convergente pour $x=1$ (série de Riemann), ainsi que pour $x=-1$ (par convergence absolue, par exemple).

Donc f est définie sur $[-1\,;1]$.

On va bien entendu approximer $f(x)$ par une somme partielle de la série entière. Pour assurer une précision d'au moins 10^{-5}, il nous faut contrôler le reste de la série entière et s'arranger pour que ce reste soit inférieur à 10^{-5}.

D'où l'idée d'essayer de majorer ce reste en valeur absolue. Si en plus on arrive à le majorer uniformément en x, ça sera d'autant plus simple à implémenter.

Pour tout $x \in [-1\,;1]$ et tout $n \in \mathbb{N}^*$:

$$\begin{aligned} |R_n(x)| &= \left| \sum_{k=n+1}^{+\infty} \frac{x^k}{k^2} \right| \\ &\leq \sum_{k=n+1}^{+\infty} \frac{|x|^k}{k^2} \\ &\leq \sum_{k=n+1}^{+\infty} \frac{1}{k^2} \end{aligned}$$

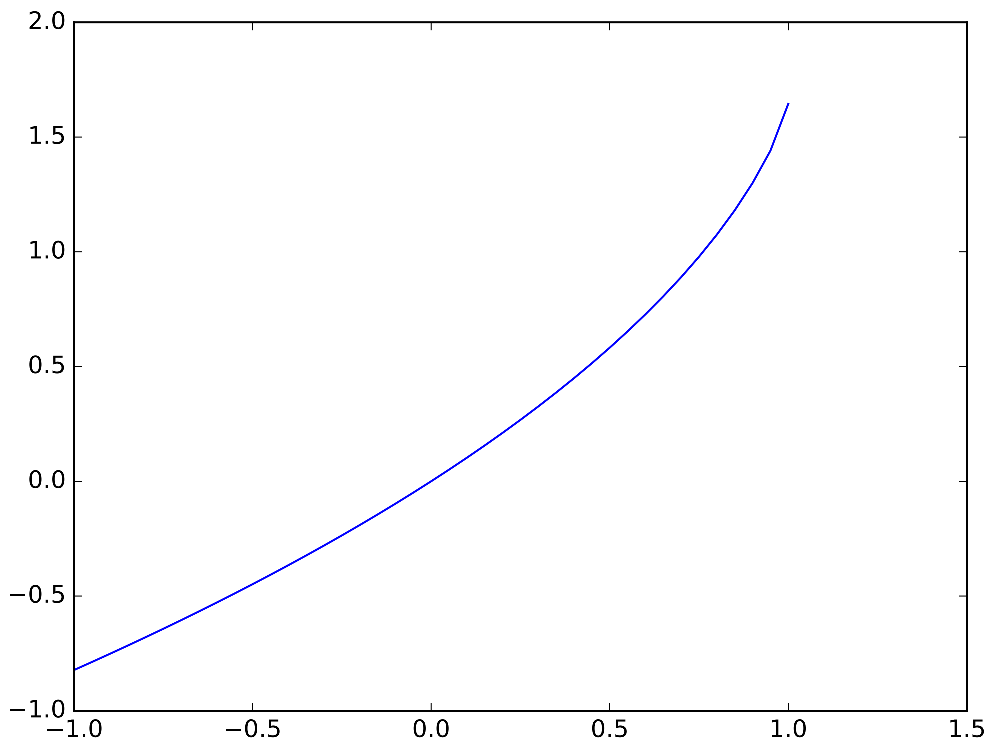
Une classique comparaison avec une intégrale s'impose !

$$\begin{aligned} |R_n(x)| &\leq \sum_{k=n+1}^{+\infty} \int_{k-1}^{k} \frac{\mathrm{d}t}{t^2} \\ &\leq \int_n^{+\infty} \frac{\mathrm{d}t}{t^2} \\ &\leq \frac{1}{n} \end{aligned}$$

Il suffit donc de prendre $n \geq 10^5$ pour obtenir la précision requise :

```
import matplotlib.pyplot as plt

import numpy as np

def f(x):

    s = 0

    puiss = 1

    for n in range(1,10**5+1):

        puiss *= x

        s += puiss / n**2

    return s


les_x =  np.arange(-1,1.05,0.05)

les_y = [f(x) for x in les_x]

plt.plot(les_x,les_y)

plt.show()
```

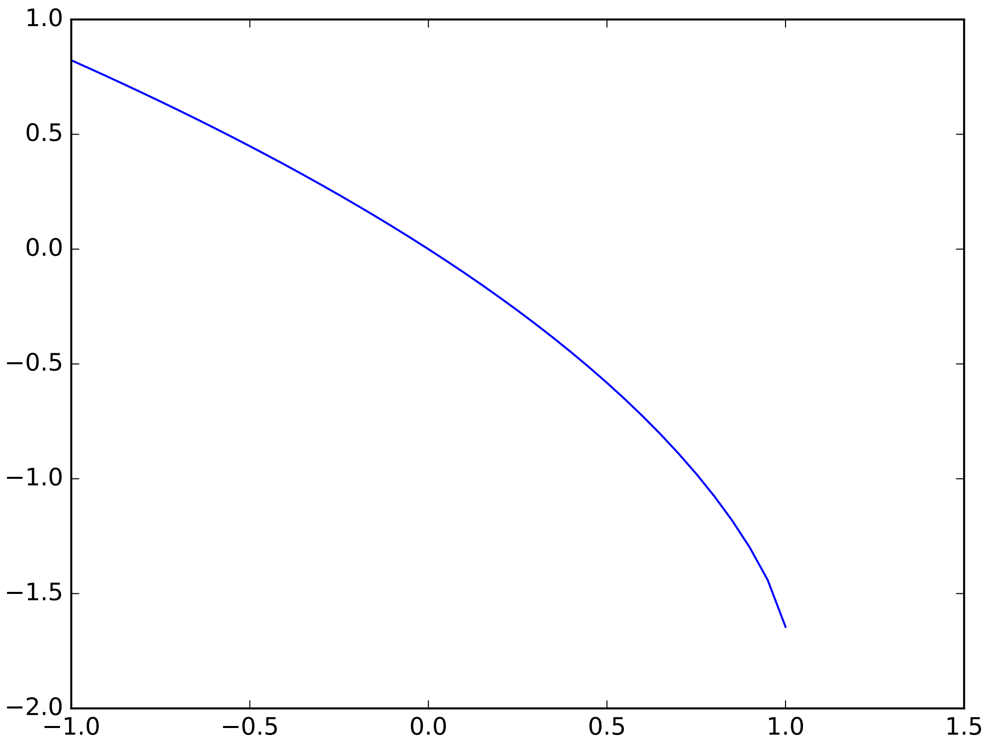
On obtient une sympathique (quoiqu'un peu banale) courbe :



2 La fonction $\varphi : t \mapsto \dfrac{\ln(1-t)}{t}$ est continue sur l'intervalle $[-1\,;1[$ (elle se prolonge par continuité en 0 selon l'équivalent classique $\ln(1-t) \sim -t$), donc la fonction g (qui n'est autre que la primitive de φ qui s'annule en 0) est définie et de classe $\mathcal{C}^1$ sur $[-1\,;1[$.
φ n'étant pas définie en 1, $g(1)$ est une intégrale impropre.
Par le changement de variable $u = 1-t$, on se ramène à l'étude de la convergence en 0^+ de l'intégrale de $u \mapsto \dfrac{\ln(u)}{1-u}$.
Or cette fonction est de signe constant au voisinage de 0^+ et elle est équivalente en 0^+ à $u \mapsto \ln(u)$, dont l'intégrale en convergente (sa primitive $u \mapsto u\ln(u) - u$ admet une limite finie en 0^+).
Donc g est aussi définie en 1, et donc g est définie sur $[-1\,;1]$.

```
from scipy.integrate import quad

from math import log

def phi(t) :

    if t == 0 :

        return -1

    else :

        return log(1-t)/t

def g(x):

    return quad(phi,0,x)[0]

les_z = [g(x) for x in les_x]

plt.plot(les_x,les_z)

plt.show()
```

On obtient une non moins sympathique (quoique non moins banale) courbe :



Aurait-on par extraordinaire hasard que $g = -f$? ?

3 Une série entière est dérivable terme à terme dans l'intérieur de son disque de convergence, donc :

$$\forall x \in]-1\,;1[, f'(x) = \sum_{n=1}^{+\infty} \frac{x^{n-1}}{n} = \frac{1}{x}\sum_{n=1}^{+\infty} \frac{x^n}{n} = -\frac{\ln(1-x)}{x}$$

(on a reconnu un développement en série entière classique).
Comme déjà signalé plus haut, g est une primitive de la fonction φ. Donc :

$$\forall x \in]-1\,;1[, g'(x) = \varphi(x) = \frac{\ln(1-x)}{x}$$

Et donc on observe que $f' + g' = 0$ sur $]-1\,;1[$.
Donc la fonction $f + g$ est constante sur $]-1\,;1[$.
Comme d'autre part elle vaut clairement 0 en 0, il vient que :

$$\forall x \in]-1\,;1[, f + g = 0$$

4

$$\begin{aligned}
g(x) &\underset{\text{IPP}}{=} \Big[\ln(1-t)\ln t\Big]_0^x + \int_0^x \frac{\ln t}{1-t}\,\mathrm{d}t \\
&= \ln(1-x)\ln x + \int_0^x \frac{\ln t}{1-t}\,\mathrm{d}t \\
&\underset{u=1-t}{=} \ln(1-x)\ln x - \int_1^{1-x} \frac{\ln(1-u)}{u}\,\mathrm{d}u \\
&= \ln(1-x)\ln x + \int_{1-x}^1 \frac{\ln(1-u)}{u}\,\mathrm{d}u \\
&= \ln(1-x)\ln x + \int_0^1 \frac{\ln(1-u)}{u}\,\mathrm{d}u - \int_0^{1-x} \frac{\ln(1-u)}{u}\,\mathrm{d}u \\
&= \ln(1-x)\ln x + g(1) - g(1-x)
\end{aligned}$$

5 Appliquant la relation précédente à $x = \frac{1}{2}$ il vient :

$$g(\frac{1}{2}) = \frac{1}{2}(g(1) + \ln^2(1/2))$$

Ou encore (puisque $f = -g$) :

$$\sum_{n \geq 1} \frac{1}{2^n n^2} = f(1/2) = \frac{1}{2}(f(1) - \ln^2 2) = \frac{1}{2}\Big(\sum_{n=1}^{+\infty} \frac{1}{n^2} - \ln^2 2\Big)$$

EXERCICE 2

2016 - Centrale 2 - PSI

Pour $t \neq -1$ on note $\varphi(t) = (f(t), g(t))$ avec $f(t) = \dfrac{t}{1+t^3}$ et $g(t) = \dfrac{t^2}{1+t^3}$.

1 Tracer avec Python le support de cet arc. Étudier les symétries, réduire l'intervalle d'étude, déterminer le comportement lorsque t tend vers -1.

2 On remarque la présence d'une boucle. Estimer numériquement la longueur de cette boucle.

3 Trouver une équation cartésienne de la courbe de la forme $F(x, y) = 0$.

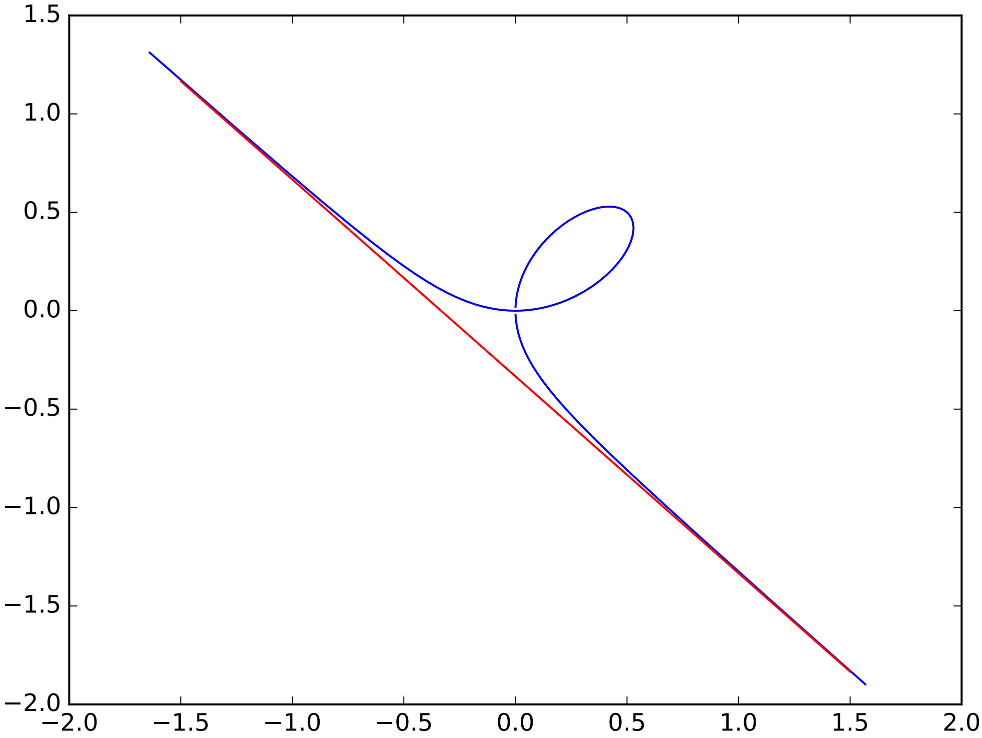
4 Montrer que $\varphi(t_1), \varphi(t_2)$ et $\varphi(t_3)$ sont alignés si et seulement si $t_1 t_2 t_3 = -1$.

Solution

1 Après avoir un peu tâtonné pour trouver quelles valeurs du paramètre représenter pour bien voir la courbe :

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

T1 = np.arange(-50,-1.2,0.01)
X1 = [ t/(1+t**3) for t in T1]
Y1 =  [ t**2/(1+t**3) for t in T1]
plt.plot(X1,Y1,color='blue')
T2 = np.arange(-0.8,50,0.01)
X2 = [ t/(1+t**3) for t in T2]
Y2 =  [ t**2/(1+t**3) for t in T2]
plt.plot(X2,Y2,color='blue')
plt.show()
```

On obtient :



Cette courbe est connue sous le nom de « Folium de Descartes ».

2 On dirait sur le graphe que la courbe présente une symétrie par rapport à la première bissectrice.On soupçonne donc l'existence d'une fonction $u(t)$ telle que $(f(u(t)), g(u(t))) = (g(t), f(t))$, c'est à dire :

$$\left\{\begin{array}{rcl} \frac{u(t)}{1+u^3(t)} & = & \frac{t^2}{1+t^3} \\ \frac{u^2(t)}{1+u^3(t)} & = & \frac{t}{1+t^3} \end{array}\right.$$

Par quotient il vient que nécessairement $u(t) = 1/t$.

Et on vérifie facilement qu'on a bien alors pour tout $t \neq 0$ que $(f(u(t)), g(u(t))) = (g(t), f(t))$.

On a donc bien une symétrie par rapport à la première bissectrice, et on peut réduire l'intervalle d'étude à $]-1\,;1]$.

Toujours en observant le graphe de la courbe on soupçonne une asymptote au voisinage de $t = -1$.

Commençons par étudier une éventuelle direction asymptotique :

$$\lim_{-1} \frac{g(t)}{f(t)} = \lim_{-1} t = -1$$

Voyons à présent s'il y a bel et bien une asymptote correspondant à cette direction :

$$\lim_{-1} g(t) + f(t) = \lim_{-1} \frac{t+t^2}{1+t^3} = \lim_{-1} \frac{t}{1-t+t^2} = -1/3$$

On a donc une asymptote d'équation $y = -x - 1/3$.

Lorsque t tend vers -1^+, $f(t)$ tend vers $-\infty$ donc le point de paramètre t tend vers $(-\infty, +\infty)$.

Et lorsque t tend vers -1^-, le point de paramètre t tend vers $(+\infty, -\infty)$.

C'est à ce moment que j'ai rajouté l'asymptote en question sur le graphe :

```
X = [-1.5 , 1.5]

Y = [1.5-1/3,-1.5-1/3]

plt.plot(X,Y,color='red')


plt.show()
```

3 La boucle correspond aux valeurs du paramètre entre 0 et $+\infty$. Il nous faut donc estimer l'intégrale :

$$\int_0^{+\infty} \sqrt{f'^2 + g'^2} = \int_0^{+\infty} \frac{\sqrt{(1-2t^3)^2 + (2t-t^4)^2}}{(1+t^3)^2} \, \mathrm{d}t$$

```
from scipy.integrate import quad

from math import sqrt

print( quad(lambda  t : sqrt((1-2*t**3)**2+(2*t-t**4)**2)
/(1+t**3)**2,0,np.inf) )

>>>  (1.6391629072271778, 4.925895123985061e-10)
```

4 La stratégie est (si possible) d'exprimer le paramètre t en fonction de $x = f(t)$ et $y = g(t)$, ce qui en l'occurrence est un jeu d'enfant puisque $t = y/x$.
Il vient alors $x = \frac{y/x}{1+(y/x)^3}$, c'est à dire :

$$x^3 + y^3 - xy = 0$$

5 Les trois points en question sont alignés si et seulement si il existe trois réels a, b, c tels que les trois points soient sur la droite d'équation $ax + by + c = 0$.
C'est à dire si et seulement si il existe trois réels a, b, c tels que pour tout $i \in \{1, 2, 3\}$ on ait :

$$a\frac{t_i}{1+t_i^3} + b\frac{t_i^2}{1+t_i^3} + c = 0$$

Ou encore :

$$ct_i^3 + bt_i^2 + at_i + c = 0$$

Les seuls polynômes de degré 3 admettant chacun des t_i pour racine sont les polynômes de la forme :

$$\lambda(X-t_1)(X-t_2)(X-t_3) = \lambda(X^3 - (t_1+t_2+t_3)X^2 + (t_1t_2+t_2t_3+t_3t_1)X - t_1t_2t_3)$$

Pour que les trois réels a, b, c existent, il faut et il suffit donc que $-t_1t_2t_3 = 1$ (pour que les coefficients de degré 3 et 0 soient égaux).

Exercice 3

2015 -Centrale 2 - PSI

Remarque : ce n'était pas évoqué dans le sujet originel, mais il me semble qu'il faut admettre le théorème (hors-programme) de Fubini sur les sommes doubles :
Si $\sum_{i\in\mathbb{N}}\sum_{j\in\mathbb{N}}|a_{i,j}|$ est convergente, alors on a convergence et égalité des deux sommes doubles :

$$\sum_{i\in\mathbb{N}}\sum_{j\in\mathbb{N}} a_{i,j} = \sum_{j\in\mathbb{N}}\sum_{i\in\mathbb{N}} a_{i,j}$$

Soit $(X_k)_{k\in\mathbb{N}^*}$ une suite de variables aléatoires réelles à valeurs dans $\mathbb{N}$ suivant la même loi et mutuellement indépendantes. Soit également N une variable aléatoire à valeurs dans $\mathbb{N}$ indépendantes des X_k. On pose :

$$Y = \sum_{k=1}^{N} X_k$$

1 Etude numérique d'un cas particulier.

- **a.** Définir une fonction Python permettant de calculer Y pour $X_1 \sim \mathcal{B}(50, 1/50)$ et $N \sim \mathcal{P}(1/12)$.
- **b.** Réaliser dix fois de suite une série de 1000 expériences et donner la moyenne et l'écart-type de Y.

2 On se replace dans le cas général.

- **a.** Exprimer G_Y en fonction de G_{X_1} et G_N.
- **b.** En déduire $E(Y)$.
- **c.** Déterminer $E(Y)$ pour $N \sim \mathcal{P}(m)$ et $X_1 \sim \mathcal{B}(n, p)$.

3 Dans une entreprise de 50 employés, soit Y la variable aléatoire indiquant le nombre de blessés dans une période τ, N indiquant le nombre d'accidents dans cette période et X_1 le nombre de blessés par accident. On suppose que $X_1 \sim \mathcal{B}(50, 1/50)$ et $N \sim \mathcal{P}(1/12)$.

- **a.** Quel est le nombre moyen de blessés durant la période τ ?
- **b.** quelle est la probabilité qu'il y ait au moins un blessé ?
- **c.** Déterminer la variance de Y.

Solution :

1 **a.**

```
def Y():

    import numpy.random as rd

    N = rd.poisson(1/12,1)

    Y = sum( rd.binomial(50,1/50,N) )

    return Y
```

b. On réalise 1000 fois l'expérience et on estime l'espérance et l'écart-type :

```
In [1]: Ys = [Y() for _ in range(1000)]

In [2]: m = sum(Ys)/1000
```

```
In [3]: s = sqrt( sum([y**2 for y in Ys])/1000 - m**2 )

In [4]: m , s
Out[4]: (0.084, 0.41102797958289894)
```

2 **a.** Pour tout $t \in [-1,1]$:

$$\begin{aligned}
G_Y(t) &= \sum_{k=0}^{+\infty} P(Y=k)t^k \\
&= \sum_{k=0}^{+\infty}\Big(\sum_{n=0}^{+\infty} P(Y=k;N=n)\Big)t^k \text{ (par les probabilités totales)} \\
&= \sum_{n=0}^{+\infty}\Big(\sum_{k=0}^{+\infty} P(X_1+\cdots+X_n=k;N=n)t^k\Big) \text{ (par Fubini)} \\
&= \sum_{n=0}^{+\infty}\Big(\sum_{k=0}^{+\infty} P(X_1+\cdots+X_n=k)P(N=n)t^k\Big) \text{ (par indépendance)} \\
&= \sum_{n=0}^{+\infty}\Big(G_{X_1+\cdots+X_n}(t)\Big)P(N=n) \\
&= \sum_{n=0}^{+\infty} G_{X_1}^n(t)P(N=n) \text{ (par indépendance et identique distribution des } X_i\text{)} \\
&= G_N(G_{X_1}(t))
\end{aligned}$$

Donc $G_Y = G_N \circ G_{X_1}$.

b. Si N et les X_i admettent des espérances, leurs fonction génératrices sont dérivables en 1 et on a $E(N)=G'_N(1)$ et $E(X_1)=G'_{X_1}(1)$.
Alors la composition $G_N \circ G_{X_1}$ est également dérivable en 1 et donc Y admet une espérance :

$$\begin{aligned}
E(Y) &= G'_Y(1) \\
&= G'_N(G_{X_1}(1)).G'_{X_1}(1) \\
&= G'_N(1).G'_{X_1}(1) \\
&= E(N).E(X_1)
\end{aligned}$$

Formule au demeurant assez naturelle : on a en moyenne $E(N)$ termes dans la somme, et chacun de ces termes vaut en moyenne $E(X_1)$.
(remarque au passage : en proba, on fait le malin en mettant son intuition en avant APRES avoir fait le calcul soigneusement)

c. A-t-on déjà vu une question aussi facile ? ?

$$E(Y) = E(N).E(X_1) = mnp$$

On s'empresse alors de vérifier la cohérence de ce qui précède en confrontant cette valeur théorique à l'estimation obtenue avec Python pour $m = 1/12, n = 50$ et $p = 1/50$:

```
In [5]: 1/12
Out[5]: 0.08333333333333333
```

3 **a.** Ben y'a rien à faire, on dirait : $E(Y) = E(N)E(X_1) = (1/12) \times 50 \times (1/50) = 1/12$.

b. C'est $1 - P(Y = 0)$, qu'on va calculer grâce à la fonction génératrice (pour se la jouer un peu, car en fait la formule des probabilités totales ferait tout aussi bien l'affaire ici). Notons $\lambda = 1/12, n = 50$ et $p = 1/50$.

$$\begin{array}{rcl} P(Y=0) & = & G_Y(0) \\ & = & G_N(G_{X_1}(0)) \\ & = & G_N(P(X_1=0)) \\ & = & \displaystyle\sum_{k=0}^{+\infty} e^{-\lambda}\frac{\lambda^k}{k!}P(X_1=0)^k \\ & = & \displaystyle\sum_{k=0}^{+\infty} e^{-\lambda}\frac{\lambda^k}{k!}(1-p)^{nk} \\ & = & e^{-\lambda}\displaystyle\sum_{k=0}^{+\infty} \frac{[(1-p)^n\lambda]^k}{k!} \\ & = & e^{-\lambda}e^{(1-p)^n\lambda} \\ & = & e^{-(1-(1-p)^n)\lambda} \end{array}$$

Et donc la probabilité cherchée est :

$$1 - e^{-(1-(49/50)^{50})/12} = 0.05160657732263041$$

c. Puisque N et X_1 admettent des variances, leurs fonctions génératrices sont deux fois dérivables en 1, et donc celle de Y aussi, ce qui permet de calculer $E(Y(Y-1))$:

$$\begin{array}{rcl} G''_Y(t) & = & \Big(G'_N(G_{X_1}(t)).G'_{X_1}(t)\Big)' \\ & = & G''_N(G_{X_1}(t))[G'_{X_1}(t)]^2 + G'_N(G_{X_1}(t)).G''_{X_1}(t) \end{array}$$

Et donc :

$$\begin{array}{rcl} E(Y(Y-1)) & = & G''_Y(1) \\ & = & G''_N(1)[[G'_{X_1}(1)]^2 + G'_N(G_{X_1}(1)).G''_{X_1}(1) \\ & = & E(N(N-1))[E(X_1)]^2 + E(N).E(X_1(X_1-1)) \\ & = & \lambda^2(np)^2 + \lambda(np(1-p) + (np)^2 - np) \end{array}$$

Et donc :

$$\begin{array}{rcl} V(Y) & = & E(Y(Y-1)) + E(Y) - E(Y)^2 \\ & = & \lambda^2(np)^2 + \lambda(np(1-p) + (np)^2 - np) + \lambda np - (\lambda np)^2 \\ & = & \lambda np\Big(1 + (n-1)p\Big) \end{array}$$

Vérifions ce résultat en le confrontant à l'écart-type empirique obtenu avec Python :

```
In [6]: sqrt((1+49/50)/12)
Out[6]: 0.406201920231798
```

Exercice 4 *2018 - Centrale 2 - PSI*

On pose, pour tout $n \in \mathbb{N}^*$, $A_n : x \mapsto \sum_{k=1}^{n} \frac{x^k}{k}$.

1 Montrer que pour tout $y \in \mathbb{R}^+$, il existe un unique $x \in \mathbb{R}^+$ tel que $A_n(x) = y$. On note $f_n(y)$ cette unique solution.

2 Écrire une fonction Python qui prend en argument n et x et qui renvoie $A_n(x)$.

3 Utiliser la fonction *fsolve* pour obtenir $f_n(y)$.

4 Tracer, pour différentes valeurs de y, les valeurs $(f_n(y))_{n\leq 100}$.

5 Montrer que la suite $(f_n(y))_n$ converge.

Solution :

1 La fonction A_n est un polynôme. Elle est donc continue sur $\mathbb{R}^+$, et dérivable sur $\mathbb{R}^+$:

$$\forall x \geq 0,\ A_n'(x) = \sum_{k=1}^{n} x^{k-1} = \sum_{k=0}^{n-1} x^k \geq 0$$

Donc A_n est strictement croissante sur $\mathbb{R}^+$.
Comme de plus $A_n(0) = 0$ et $A_n \underset{+\infty}{\longrightarrow} +\infty$, il vient que A_n réalise une bijection de $\mathbb{R}^+$ dans $\mathbb{R}^+$.
Et donc pour tout $y \in \mathbb{R}^+$ il existe un unique $x \in \mathbb{R}^+$ tel que $A_n(x) = y$.

2

```python
def A(n,x):
    s = 0
    puiss = 1.
    for k in range(1,n+1):
        puiss *= x
        s += puiss / k
    return s
```

3

```python
from scipy.optimize import fsolve
def f(n,y):
    return fsolve( lambda x : A(n,x)-y , 1)[0]
```

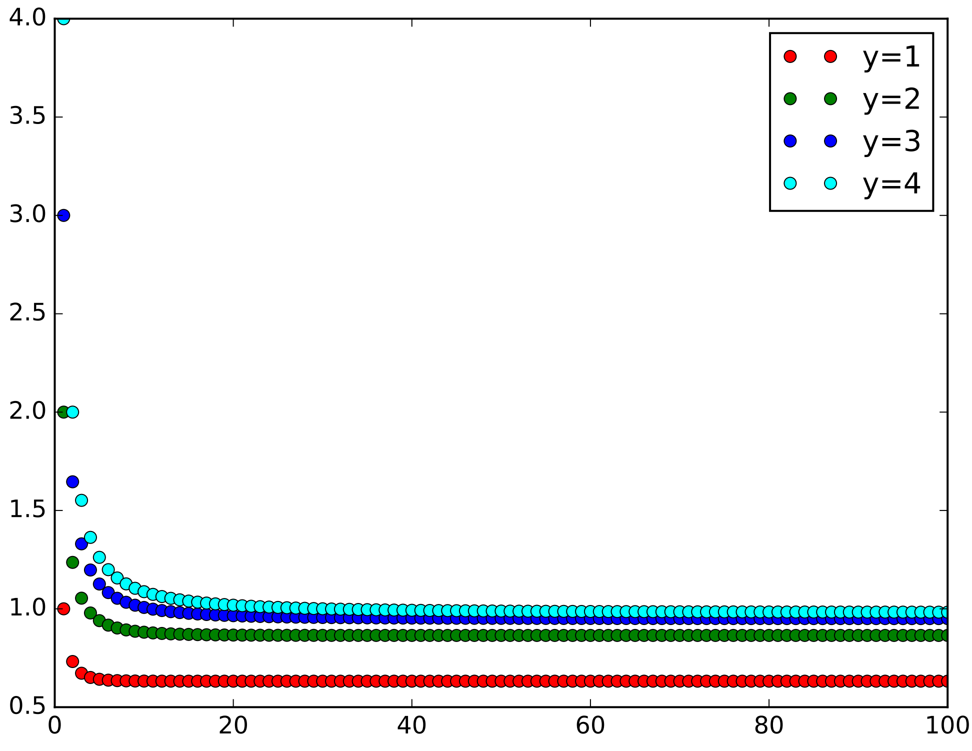
Remarque : si vous vous demandez pourquoi j'ai pris 1 comme second paramètre de *fsolve* au lieu du naturel 0, la réponse vous attend à la question suivante.

4

```python
import matplotlib.pyplot as plt
Y = [1,2,3,4]
```

```
couleurs = ['red','green','blue','cyan']
les_n = [n for n in range(1,101)]
for k in range(len(Y)):
    y = Y[k]
    coul = couleurs[k]
    les_f = [f(n,y) for n in les_n]
    plt.plot(les_n,les_f,'o',color=coul,label='y='+str(y))
plt.legend()
plt.show()
```

On obtient :



On dirait bien que la suite $(f_n(y))_n$ est décroissante et convergente vers une limite $\ell(y)$, et que $y \mapsto \ell(y)$ est croissante.

Remarque : j'avais dans un premier temps pris 0 comme second paramètre pour l'utilisation de *fsolve* et j'avais constaté (probablement comme vous si vous avez fait la même chose) qu'a partir de $y = 2$ un curieux « décrochage »se produisait : les valeurs de $f_n(y)$ commençaient à se comporter comme sur le graphe ci-dessus pour les premières valeurs de n, mais brutalement $f_n(y)$ passait à 0. J'ai soupçonné un problème de précision de calcul avec *fsolve* et j'ai changé le point de départ de l'algorithme de résolution en 1, avec un résultat probant.

5 Fixons un $y > 0$ et notons pour alléger la rédaction $u_n = f_n(y)$. Très classiquement :

$$\begin{aligned} A_n(u_{n+1}) &= A_{n+1}(u_{n+1}) - \frac{u_{n+1}^{n+1}}{n+1} \\ &\leq A_{n+1}(u_{n+1}) \\ &\leq y \\ &\leq A_n(u_n) \end{aligned}$$

Et puisque la fonction A_n est croissante, il vient que $u_{n+1} \leq u_n$.
Comme d'autre part on a $u_n \geq 0$, la suite (u_n) est décroissante et minorée. Elle est donc convergente.

6 C'est un peu étonnant qu'on ne nous demande pas de déterminer la limite de $(f_n(y))$. Je soupçonne que la question venait « en direct »lors de l'oral.
Commençons par travailler un peu au corps la fonction $A_n(x)$:

$$\begin{aligned} A_n(x) &= \int_0^x \sum_{k=0}^{n-1} t^k \,\mathrm{d}t \\ &= \int_0^x \frac{1-t^n}{1-t} \,\mathrm{d}t \\ &= -\ln(1-x) - \int_0^x \frac{t^n}{1-t} \,\mathrm{d}t \end{aligned}$$

En particulier :

$$y = A_n(u_n) = -\ln(1-u_n) - \int_0^{u_n} \frac{t^n}{1-t} \,\mathrm{d}t$$

Notons ℓ la limite de la suite (u_n) (avec les mêmes notations qu'à la question précédente).
Supposons que $\ell \geq 1$. Alors on a :

$$\forall n \in \mathbb{N}, y = \sum_{k=1}^{n} \frac{u_n^k}{k} \geq \sum_{k=1}^{n} \frac{l^k}{k} \geq \sum_{k=1}^{n} \frac{1}{k}$$

Ce qui est impossible du fait de la divergence de la série harmonique.
Donc $\ell < 1$.
On peut alors majorer l'intégrale qui nous pourrit les calculs :

$$0 \leq \int_0^{u_n} \frac{t^n}{1-t} \,\mathrm{d}t \leq \int_0^{u_n} \frac{u_n^n}{1-u_n} \,\mathrm{d}t = \frac{u_n^{n+1}}{1-u_n} \longrightarrow 0$$

(car à partir d'un certain rang on a $u_n \leq \frac{\ell+1}{2} < 1$. On voit ici pourquoi j'ai voulu montrer que $\ell < 1$)
On a donc :

$$-\ln(1-u_n) \longrightarrow y$$

Ou encore :

$$u_n \longrightarrow 1 - e^{-y}$$