TD Math & Python (2). Corrigé

Exercice A

1)

```
def vaY() :
     y = 0
     N = rd.poisson(1/10)
     for k in range(N) :
          y = y + rd.binomial(50,1/50)
     return y

def momentY() :
     e = 0 ; v = 0
     for k in range(1000) :
          y = vaY() ; e = e + y ; v = v + y**2
     return e/N , (v/N-(e/N)**2)**(0.5)
```

On obtient environ 0.1 comme espérance (en effet, $E(X) = 1$ et $E(N) = 0.1$

2) a) Par la formule de Wald, on a $G_Y(z) = G_N(G_X(z))$ et $E(Y) = G'_Y(1) = G'_N(1)G'_X(1) = E(N)E(X)$.

b) On a $E(Y) = E(N)E(X) = \lambda np$ et $V(Y) = G''_Y(1) + G'_Y(1) - G'_Y(1)^2$.

Or, $G''_Y(z) = G''_N(G_X(z))G'_X(z)^2 + G'_N(G_X(z))G''_X(z)$, donc $G''_Y(1) = G''_N(1)G'_X(1)^2 + G'_N(1)G''_X(1)$.

On a $G_N(z) = e^{\lambda(z-1)}$ et $G_X(z) = (q + pz)^n$, donc $G''_N(1) = \lambda^2$ et $G''_X(1) = p^2n(n-1)$.

On en conclut $V(Y) = \lambda^2p^2n^2 + \lambda p^2n(n-1) + \lambda np - (\lambda pn)^2 = \lambda np + \lambda p^2n(n-1)$.

Exercice B

1)

```
def e(a,b) :
     M = np.array([[3*a-2*b,-6*a+6*b+3],[a-b,-2*a+3*b+1]])
     V = alg.eigvals(M)
     return abs(V[1]-V[0])
```

2) a)

```
def hasard(p) :
     N = 500 ; c = 0
     for k in range(N) :
          a = rd.geometric(p) ; b = rd.geometric(p)
          if ecart(a,b) > 1/10 :  c = c + 1
     return c
```

b)

```
P = [k/100 for k in range(1,100)] ; M = [ hasard(p)/500 for p in P]
```

```
plt.plot(P,M)
F = [ (2-2*p+p**2)/(3-p) /500 for p in P] ; plot(P,F)
plt.show()
```

c) $N(a,b)$ est diagonalisable ssi elle admet deux valeurs propres distinctes, c'est-à-dire $e(a,b) > 0$.

3) a) En résolvant $MX = (a+1)X$, on obtient le vecteur propre $(3,1)$.

On cherche ensuite $P = \begin{pmatrix} 2 & \alpha \\ 3 & \beta \end{pmatrix}$ inversible telle que $MP = PN$.

b) On a $P(a+1=b) = \sum_{k=1}^{+\infty} P(a+1=k, b=k) = \sum_{k=1}^{+\infty} p^2 q^{2k-1} = \dfrac{p^2 q}{(1-q^2)} = \dfrac{pq}{(1+q)}$.

Donc $P(a+1 \neq b) = 1 - \dfrac{pq}{1+q} = \dfrac{2-2p+p^2}{2-p}$.

Remarque : Ici, on a $\{\lambda, \mu\} = \{a+1, b\}$, donc $e(a,b)$ est un entier naturel.

Autrement dit, $e(a,b) \geq \frac{1}{2}$ ssi λ et μ sont distincts. Donc la probabilité que $e(a,b) \geq \frac{1}{2}$ vaut $P(a+1 \neq b)$.

On prend $\frac{1}{2}$ et non pas 1 compte tenu des erreurs d'évaluations numériques (de même si on testait $\lambda = \mu$).

Ecercice C

1)

```
def essai(n) :
     A = np.zeros((n,n))
     for i in range(n) :
          for j in range(n) :
             A[i,j] = 2*rd.randint(2)-1
     return A

def moments(n) :
     N = 1000 ; s = 0 ; v = 0
     for i in range(N) :
          d = alg.det(essai(n)) ; s = s + d ; v = v + d**2
     return(["n=",n,";",s/N,v/N])

for n in [1,2,3,4,5] :  print(moments(n))
```

2) On a $D_n = \sum_{i=1}^{n} X_{i1} C_{i1}$, où les C_{i1} sont les cofacteurs.

Pour tout i, la variable X_{i1} est indépendante de C_{i1} (qui est une fonction des X_{kj}, avec $k \neq i$ et $j \geq 2$).

Donc $E(X_{i1}C_{i1}) = E(X_{i1})E(C_{i1}) = 0$, car $E(X_{i1}) = 0$. Donc par linéarité, $E(D_n) = 0$.

Donc $V(D_n) = E(D_n^2) = \sum_{i,j} X_{i1}C_{i1}X_{j1}C_{j1}$.

Les variables $X_{i1}X_{j1}$ et $C_{i1}C_{j1}$ sont indépendantes, donc on obtient $E(D_n^2) = \sum_{i,j} E(X_{i1}X_{j1})E(C_{i1}C_{j1})$.

Pour $i \neq j$, $E(X_{i1}X_{j1}) = 0$. Donc $E(D_n^2) = \sum_{i=1}^{n} E(X_{i1}^2)E(C_{i1}^2) = \sum_{i=1}^{n} E(C_{i1}^2)$.

Or, la loi de C_{i1} est la loi de D_{n-1}, donc $E(D_n^2) = nE(D_{n-1}^2)$.

Par récurrence immédiate, on en déduit $E(D_n^2) = n!$

Remarque mathématique : Les variables aléatoires $X_{i1}C_{i1}$ ne sont pas indépendantes (en effet, certaines v.a. X_{kj} interviennent à la fois dans C_{i1} et dans C_{j1}).

Exercice D

1)

```
def test(i) :
     L = [rd.randint(3)]
     for k in range(2,31) :
          x = rd.randint(3)
          if not(x in L) : L.append(x)
          if len(L) == i :  return k
```

2)

```
def repartition(i,n) :
     Y = np.zeros(31)
     for j in range(100) :
          k = test(i) ; Y[k] += 1
     return(Y)
```

Par exemple, `repartition(2,100)` renvoie $[0, 0, 58, 31, 7, 4, 0, 0, 0, ..., 0]$.

Et `repartition(3,100)` renvoie $[0, 0, 0, 12, 23, 25, 14, 9, 2, 5, 3, 2, 2, 1, 0, 1, 0, 0, 0, 1, 0, ..., 0]$.

3) $Y_3 - Y_2$ suit la loi géométrique (à valeurs dans $\mathbb{N}^*$) associée à la loi de Bernoulli $\mathcal{B}(\frac{1}{3})$.

On a notamment $P(Y_3 = n + k, Y_2 = n) = \frac{1}{3}\left(\frac{2}{3}\right)^{k-1}$ et $E(Y_3 - Y_2) = 3$.

Exercice E

1) On a $|X_n| \leq \sum_{k=1}^{+\infty} 2^{-k} = 1$.

Remarque utile pour la suite : La définition de X_n est liée à l'écriture en base 2 des nombres réels. Si les ε_k valaient 0 ou 1, la loi de X_n serait la loi uniforme sur l'ensemble des nombres réels de $[0, 1[$ s'écrivant en base 2 avec n bits (ε_k représentant alors le k-ième chiffre du développement en base 2).

Ici, par un simple changement affine $[0, 1] \to [-1, 1]$, on en déduit que X_n suit une loi uniforme sur un ensemble équiréparti de $[-1, 1]$ dont la limite (la réunion) quand $n \to +\infty$ est dense dans $[-1, 1]$.

D'où, pour toute fonction $f \in C^0([-1, 1])$, $\lim_{n\to+\infty} E(f(X_n)) = \frac{1}{2}\int_{-1}^{+1} f(x)\, dx$ valeur moyenne de f sur $[-1, 1]$.

2) Posons $S_{n,N} = \dfrac{1}{N}\sum_{k=1}^{N} \cos(tX_{n,k})$. On a $E(S_{n,N}) = E(X_n)$ et $V(S_N) = \dfrac{N\ V(X_n)}{N^2} = \dfrac{V(X_n)}{N}$.

Par Bienaymé-Tchebychev, $P\left(|S_{n,N} - E(S_{n,N})| \geq \varepsilon\right) \leq \dfrac{V(S_N)}{\varepsilon^2} \to 0$ lorsque $N \to +\infty$.

3)

```
def Rademacher() :  return 2*rd.binomial(1,1/2)-1
def X(n) :
     s = 0
     for k in range(1,n):
         s += Rademacher() / 2**k
```

```
    return s

def moyenne(n,t) :
    A = np.array([ cos(t*X(n)) for k in range(1000) ])
    return np.mean(A)     ### on peut aussi utiliser une boucle for

def graphe(n) :
    X = np.arange(-10,10.1,0.1) ; Y = [ moyenne(n,t) for t in X ]
    plt.plot(X,Y)
graphe(10) ; graphe(20)  # trop de graphes superposés → problèmes possibles ....
plt.show()
```

On conjecture que la suite des fonctions $t \longmapsto E(\cos(tX_n)$ converge.

Compte tenu de la remarque du 1), la limite est $L(t) = \frac{1}{2}\int_{-1}^{+1} \cos(tx)\, dx = \frac{\sin t}{t}$ fonction sinus cardinal.

4) Par indépendance des $e^{it\varepsilon_k}$, On a $\phi_{X_n(t)} = \prod_{k=1}^{n} E(e^{it\varepsilon_k}) = \prod_{k=1}^{n} \cos\left(\frac{t}{2^k}\right)$.

En utilisant $\cos\theta = \frac{1}{2}\frac{\sin(2\theta)}{\sin\theta}$, on a $\phi_{X_n(t)} = \prod_{k=1}^{n} \cos\left(\frac{t}{2^k}\right) = \frac{1}{2^n}\frac{\sin(t)}{\sin(t/2^n)}$.

Donc $\lim_{n\to+\infty} \phi_{X_n(t)} = \frac{\sin t}{t}$, car $\sin\left(\frac{t}{2^n}\right) \sim \frac{t}{2^n}$ lorsque $n \to +\infty$.

Exercice E

1) `print((5+1j)**4*(239-1j))` renvoie `(114244+114244j)`

En sachant que pour tout $x > 0$, $\arg(x+iy) = \arctan\left(\frac{y}{x}\right)$, alors on obtient :

$\theta = 4\arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right) = \frac{\pi}{4}\ [2\pi]$. Comme $\arctan\left(\frac{1}{5}\right) \in \left[0, \frac{\pi}{4}\right]$, alors $\theta \in [-\pi, \pi]$. Donc $\theta = \frac{\pi}{4}$.

2) On sait que $\arctan(x) = \sum_{k=0}^{+\infty} \frac{(-1)^k}{2k+1} x^{2k+1}$ qui vérifie le CSSA pour tout $|x| \leq 1$.

On utilise alors la majoration du reste d'une série alternée.

Donc $\forall x \in [0,1]$, $\left|\arctan(x) - \sum_{k=0}^{+\infty} \frac{(-1)^k}{2k+1} x^{2k+1}\right| \leq \frac{(2n+2)!}{(2n+3)!}\,|x|^{2n+3}$.

On prend par exemple n tel que $\left(\frac{1}{5}\right)^n < 10^{-15}$ c'est-à-dire $n > 15\frac{\log 10}{\log 5}$, c'est-à-dire $n \geq 22$.

On a alors : $\frac{16}{2n+1}\left(\frac{1}{5}\right)^n + \frac{4}{2n+1}\left(\frac{1}{239}\right)^n < 10^{-15}$. D'où :

```
def somme(x)
    s = 0
    for k in range(23) :  s += (-1)**k/(2*k+1)*x**(2*k+1)
    return s
print( 16 * somme(1/5) - 4 * somme(1/239))
```