Oraux Centrale avec Python

Arithmétique et algèbre générale

Exercice 1

a)

```python
def prime(n):
    if n < 2:
        return False
    d = 2
    while d * d <= n:
        if n % d == 0:
            return False
        d += 1
    return True
```

b)

```python
def pi(n):
    p = 2
    s = 0
    while p <= n:
        if prime(p):
            s += 1
        p += 1
    return s
```

c) On réalise le script suivant :

```python
for n in (10, 1e2, 1e3, 1e4, 1e5, 1e6):
    print(n / pi(n))
```

qui fournit les valeurs :

2,5 4,0 5,952 380 952 380 952 6 8,136 696 501 220 504 10,425 354 462 051 71 12,739 178 068 231 038

ce qui laisse conjecturer que $\lim \frac{n}{\pi(n)} = +\infty$, voire que $\frac{n}{\pi(n)} = \Theta(\ln n)$.

d) $\frac{\ln n}{2\ln 2} \leqslant \frac{n}{\pi(n)}$ donc $\lim \frac{n}{\pi(n)} = +\infty$.

e) D'après l'inégalité admise, $\pi(n) = \frac{n}{k}$ implique $\ln n \leqslant 2k\ln 2$, soit $n \leqslant 2^{2k}$.

f) On réalise le script suivant (qui utilise le fait qu'une solution doit être multiple de 11) :

```python
sol = []
for n in range(11, 2**11, 11):
    if 2 * n == 11 * pi(n):
        sol.append(n)
print(sol)
```

qui fournit les valeurs : 561, 583, 594, 605, 616, 627, 649, 660.

g) F_k est une partie de $\mathbb{N}$ majorée par 2^{2k} donc est finie, et non vide car elle contient 2. Soit n son plus grand élément. On a $\frac{n}{\pi(n)} \leqslant k < \frac{n+1}{\pi(n+1)} \leqslant \frac{n+1}{\pi(n)}$ donc $n \leqslant k\pi(n) < n+1$, et s'agissant d'entiers, $n = k\pi(n)$, donc n est solution de (E_k).

Exercice 2

a) $\varphi(1) = 1$, $\varphi(10) = 4$, $\varphi(p) = p - 1$ lorsque p est premier.

b) On suppose $n \geqslant 2$ et on considère la propriété $\mathscr{P}(i)$ suivante :

$\mathscr{P}(i)$: *pour tout* $j \in [\![1, n-1]\!]$, `table[j] == 0` *si et seulement si* $\text{pgcd}(j, n) \in [\![2, i-1]\!]$.

Nous allons prouver par récurrence qu'elle est vraie à l'entrée de la boucle indexée par $i \in [\![2, \lfloor n/2 \rfloor]\!]$.

- si $i = 2$, c'est clair puisqu'à l'entrée de la boucle indexée par i toutes les valeurs de `table[j]` pour $j \in [\![1, n-1]\!]$ sont égales à 1.
- Si $i \geqslant 2$, on suppose $\mathscr{P}(i)$ vraie à l'entrée de la boucle indexée par i. Considérons un entier j tel que $\text{pgcd}(j, n) = i$. S'il en existe, c'est que i divise n et qu'il existe $k \in [\![1, \lfloor (n-1)/i \rfloor]\!]$ tel que $j = ki$.
  De plus, si i divise n alors $\text{pgcd}(i, n) = i$ et on a donc `table[i] == 1` d'après $\mathscr{P}(i)$.
  Les deux conditions pour que s'exécute la boucle secondaire sont donc réunies, et à l'issue de celle-ci on aura bien `table[j] == 0`.

Sachant que pour tout $j \in [\![1, n-1]\!]$, $\text{pgcd}(j, n) \in [\![1, \lfloor n/2 \rfloor]\!]$ on en déduit que cette fonction renvoie la liste des entiers de $[\![1, n-1]\!]$ qui sont premiers avec n.

c) On en déduit la fonction :

```python
def phi(n):
    return len(premAvec(n))
```

d) On calcule $\varphi(1\,024 \times 81) = 27\,648 = \varphi(1\,024)\varphi(81)$, ce qui laisse conjecturer le résultat suivant : si m et n sont premiers entre eux alors $\varphi(mn) = \varphi(m)\varphi(n)$.

e) On observe que $\varphi(n)$ est le nombre d'éléments inversibles dans $\mathbb{Z}/n\mathbb{Z}$.

Par ailleurs on peut définir une fonction $f : \mathbb{Z}/mn\mathbb{Z} \to \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ en posant : $f(\overline{x}) = (\overline{x}, \overline{x})$. En effet, si $x \equiv y \pmod{mn}$ alors $x \equiv y \pmod{m}$ et $x \equiv y \pmod{n}$.

Il s'agit de plus d'un morphisme d'anneau (évident), et un isomorphisme lorsque m et n sont premiers entre eux : en effet, si $f(\overline{x}) = (\overline{0}, \overline{0})$ alors m et n divisent x donc mn divise x, et $\overline{x} = 0$. f est donc une injection entre deux ensembles de même cardinal.

Les éléments inversibles de $\mathbb{Z}/mn\mathbb{Z}$ correspondent donc par l'intermédiaire de f aux couples formés d'un élément inversible de $\mathbb{Z}/m\mathbb{Z}$ et d'un élément inversible de $\mathbb{Z}/n\mathbb{Z}$ et ainsi $\varphi(mn) = \varphi(m)\varphi(n)$.

Exercice 3

a) D'après le principe des tiroirs, il existe $0 \leqslant i < j \leqslant n$ tel que $\sigma^i(x) = \sigma^j(x)$. Alors $\sigma^{j-i}(x) = x$, donc $\text{Per}(\sigma, x)$ existe, et $\text{Per}(\sigma, x) \leqslant j - i \leqslant n$.

L'ordre de σ est alors le ppcm des périodes des $x \in \mathrm{E}_n$.

b)

```python
def periode(sigma, x):
    y = sigma[x]
    p = 1
    while y != x:
        y = sigma[y]
        p += 1
    return p
```

c)

```python
def listeDesPériodes(sigma):
    return [periode(sigma, x) for x in range(len(sigma))]
```

On obtient la liste des périodes suivantes : $[2, 7, 7, 2, 7, 7, 7, 7, 7, 1]$. σ est d'ordre 14.

d) $\mathscr{R}_\sigma$ est *réflexive* car $x = \sigma^0(x)$, *symétrique* car $y = \sigma^k(x) \iff x = \sigma^{-k}(y)$, et *transitive* car $y = \sigma^k(x)$ et $z = \sigma^l(y)$ entraine $z = \sigma^{k+l}(x)$.

e) Soit $y \in \Omega_\sigma(x)$, et $k \in \mathbb{Z}$ tel que $y = \sigma^k(x)$. Écrivons la division euclidienne de k par $p = \text{Per}(\sigma, x)$: $k = np + r$ avec $n \in \mathbb{Z}$ et $r \in [\![0, p-1]\!]$. Alors $y = \sigma^r(x)$. L'inclusion réciproque est évidente.

f) On rédige d'abord une fonction qui calcule l'orbite d'un élément $x \in E_n$, en marquant chaque élément de cette orbite :

```
def orbite(sigma, x, dejavu):
    dejavu[x] = True
    lst = [x]
    y = sigma[x]
    while y != x:
        dejavu[y] = True
        lst.append(y)
        y = sigma[y]
    return lst
```

On calcule ensuite séquentiellement les orbites des éléments non marqués, pour éviter les doublons :

```
def listeDesOrbites(sigma):
    n = len(sigma)
    dejavu = [False] * n
    lst = []
    for x in range(n):
        if not dejavu[x]:
            lst.append(orbite(sigma, x, dejavu))
    return lst
```

La permutation donnée en exemple fournit les orbites $\big[[0,3],[1,6,8,4,2,7,5],[9]\big]$.

Exercice 4

a) Soit $P \neq Q$ dans $\mathscr{A}$. On pose $P = \sum_{i=0}^{+\infty} a_i X^i$ et $Q = \sum_{j=0}^{+\infty} b_j X^j$ et on note k le plus petit entier vérifiant $a_k \neq b_k$. Sans perte de généralité, supposons $a_k = 0$ et $b_k = 1$. Alors $Q(-2) - P(-2) \equiv 2^k \mod (2^{k+1})$, ce qui prouve que $P(-2) \neq Q(-2)$.

b) Il s'agit de justifier l'existence de la décomposition de n dans la base $b = -2$. On raisonne par récurrence sur $|n|$.

- Si $n = 0$, le polynôme nul convient.
- Si $|n| > 0$, deux cas de figure sont possibles :
  - si n est pair, on applique l'hypothèse de récurrence à $-n/2$: il existe $Q \in \mathscr{A}$ tel que $-n/2 = Q(-2)$ et on pose $P = XQ$;
  - si n est impair, on applique l'hypothèse de récurrence à $(1-n)/2$: il existe $Q \in \mathscr{A}$ tel que $(1-n)/2 = Q(-2)$ et on pose $P = 1 + XQ$.

c) La question précédente fournit la démarche à suivre :

```
def decomposition(n):
    p = []
    while n != 0:
        if n % 2 == 0:
            p.append(0)
            n = - n // 2
        else:
            p.append(1)
            n = (1 - n) // 2
    return p
```

(le polynôme est représenté par la liste d'indice croissant de ses coefficients).

d) Pour $n = 2015$ on obtient la liste $[1,1,0,0,0,1,0,0,0,0,0,1,1]$, représentant le polynôme $P = 1 + X + X^5 + X^{11} + X^{12}$.

Algèbre linéaire

Exercice 5

a)

```python
def ecart(a, b):
    M = np.array([[3*a-2*b, -6*a+6*b+3], [a-b, -2*a+3*b+1]], dtype=float)
    v1, v2 = alg.eigvals(M)
    e = abs(v1 - v2)
    return round(e, 2)
```

b)

```python
def hasard(p):
    s = 0
    for _ in range(500):
        a, b = rd.geometric(p, 2)
        if ecart(a, b) >= 1e-1:
            s += 1
    return s
```

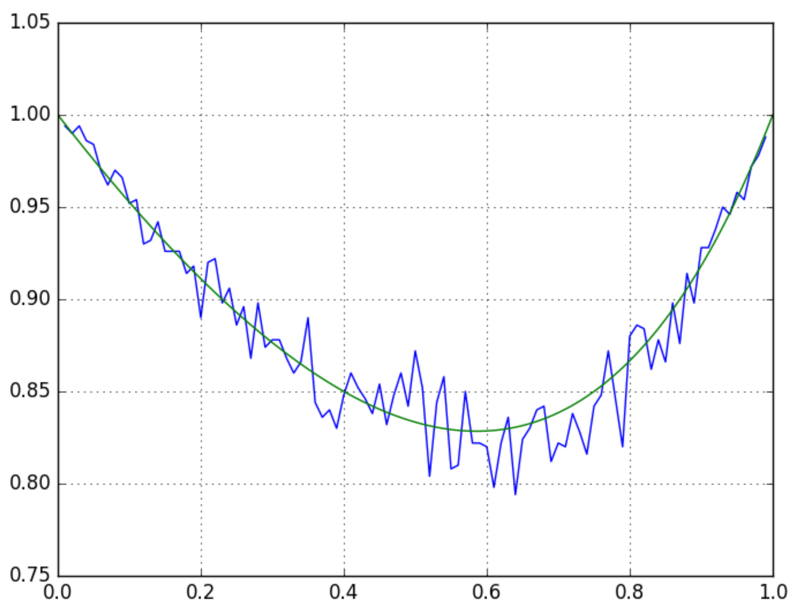
c) On réalise le script suivant :

```python
X, Y = [], []
for k in range(1, 100):
    p = k / 100
    X.append(p)
    Y.append(hasard(p)/500)
plt.plot(X, Y)
```

d) On ajoute au script précédent les lignes :

```python
P = np.linspace(0, 1, 256)
F = [(2-2*p+p**2)/(2-p) for p in P]
plt.plot(P, F)
```

On obtient le graphe :



e) Posons $X = \begin{pmatrix} x \\ y \end{pmatrix}$. La résolution de l'équation $M(a,b)X = (a+1)X$ se ramène à $x = 3y$, donc $X_1 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ est un vecteur propre associé à la valeur propre $(a+1)$. On cherche ensuite un vecteur X_2 linéairement indépendant avec X_1 et vérifiant : $M(a,b)X_2 = X_1 + bX_2$; la résolution se ramène à l'équation $(a-b)x + (1-2(a-b))y = 1$ qui fournit (entre autre) la solution $X_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$. Posons $P = \begin{pmatrix} 3 & 2 \\ 1 & 1 \end{pmatrix}$. Alors $M(a,b) = P\begin{pmatrix} a+1 & 1 \\ 0 & b \end{pmatrix}P^{-1}$.

La valeur propre $a+1$ est d'ordre 1, donc $M(a,b)$ est diagonalisable si et seulement si $b \neq a+1$.

f) On a $\mathrm{P}(a+1=b)=\sum_{k=1}^{+\infty}\mathrm{P}(a=k-1 \text{ et } b=k)=\sum_{k=1}^{+\infty}p(1-p)^{k-1}\times p(1-p)^k=\sum_{k=1}^{+\infty}p^2(1-p)^{2k-1}=\dfrac{p^2(1-p)}{1-(1-p)^2}$.

Ainsi, $\mathrm{P}(a+1\neq b)=1-\dfrac{p^2(1-p)}{1-(1-p)^2}=\dfrac{2-2p+p^2}{2-p}$, résultat conforme à la simulation numérique.

Exercice 6

a)

```python
def A(n, a):
    M = np.zeros((n, n), dtype=float)
    for i in range(n-1):
        M[i, i+1] = 1/a
        M[i+1, i] = a
    return M
```

b) On réalise le script suivant :

```python
for n in range(3, 9):
    for a in (-2, -1, 1, 2, 3):
        print(alg.eigvals(A(n, a)).round(2))
```

qui suggère que $\mathrm{A}_{n,a}$ possède n valeurs propres distinctes indépendantes de a.

c) On définit les polynômes P_n pour $n<9$ à l'aide du script suivant :

```python
p = [None] * 9
p[1] = Polynomial([0, 1])
p[2] = Polynomial([-1, 0, 1])
for n in range(3, 9):
    p[n] = p[1] * p[n-1] - p[n-2]
```

puis on calcule les raines des polynômes $\mathrm{P}_3,\dots,\mathrm{P}_8$:

```python
for n in range(3, 9):
    print(p[n].roots().round(2))
```

d) On conjecture que P_n est égal au polynôme caractéristique $\mathrm{C}_{n,a}$ de $\mathrm{A}_{n,a}$, ce que l'on prouve par récurrence.

- C'est vrai pour $n=1$ et $n=2$.
- Si $n\geqslant 3$, on suppose le résultat acquis aux rangs $n-1$ et $n-2$. Le calcul bien connu du déterminant d'une matrice tridiagonale fournit la relation de récurrence $\mathrm{C}_{n,a}=\mathrm{XC}_{n-1,a}-a\times\dfrac{1}{a}\mathrm{C}_{n-2,a}=\mathrm{XP}_{n-1}-\mathrm{P}_{n-2}=\mathrm{P}_n$ qui montre que la récurrence se propage.

e) La suite $d_n=\mathrm{P}_n(0)$ vérifie : $d_1=0$, $d_2=-1$ et $d_{n+2}=-d_n$, ce qui prouve que $\det \mathrm{A}_{n,a}=\begin{cases}0 & \text{si } n \text{ est impair}\\ (-1)^p & \text{si } n=2p\end{cases}$. Ainsi, $\mathrm{A}_{n,a}$ est inversible si et seulement si n est pair.

Posons $\mathrm{D}=\operatorname{diag}(1,a,\dots,a^{n-1})$. Alors $\mathrm{DA}_{n,a}\mathrm{D}^{-1}=\mathrm{A}_{n,1}$, qui est une matrice symétrique réelle donc diagonalisable. La matrice $\mathrm{A}_{n,a}$ est donc elle aussi diagonalisable.

f) Munissons $\mathbb{R}^n$ de la norme $\|\cdot\|_\infty$, et $\mathcal{M}_n(\mathbb{R})$ da la norme subordonnée $\|\mathrm{A}\|=\max_{1\leqslant i\leqslant n}\sum_{j=1}^{n}|a_{ij}|$. Si X est un vecteur propre associé à la valeur propre λ, on a $\mathrm{A}_{n,a}\mathrm{X}=\lambda\mathrm{X}$ donc $|\lambda|.\|\mathrm{X}\|_\infty\leqslant\|\mathrm{A}_{n,a}\|.\|\mathrm{X}\|_\infty$, soit $|\lambda|\leqslant\|\mathrm{A}_{n,a}\|=|a|+\dfrac{1}{|a|}$.

Analyse

Exercice 7

a)

```
def L(n):
    return 2**int(np.log2(n))

def S(n):
    s = 0
    while n > 0:
        s += n % 2
        n //= 2
    return s
```

b)

```
def sommePartielle(N, alpha):
    s = 0
    for n in range(1, N+1):
        s += 1 / L(n)**alpha / S(n)
    return s
```

c) On a $\frac{b_{n+1}}{2} = 2^n a_{2^{n+1}} \leqslant \sum_{k=2^n}^{2^{n+1}-1} a_k \leqslant 2^n a_n = b_n$ donc $\frac{1}{2}\sum_{n=0}^{N} b_{n+1} \leqslant \sum_{k=1}^{2^{N+1}-1} a_k \leqslant \sum_{n=0}^{N} b_n$. Les séries positives $\sum a_n$ et $\sum b_n$ ont même nature.

d) On a $\frac{n}{2} \leqslant \mathrm{L}(n) \leqslant n$ et $1 \leqslant \mathrm{S}(n) \leqslant \log n + 1$ donc $\frac{1}{n^\alpha(\log n+1)} \leqslant \frac{1}{\mathrm{L}(n)^\alpha \mathrm{S}(n)} \leqslant \frac{2^\alpha}{n^\alpha}$. La série $\sum \frac{1}{\mathrm{L}(n)^\alpha \mathrm{S}(n)}$ est donc convergente pour $\alpha > 1$, et divergente pour $\alpha < 1$.

Pour $\alpha = 1$, posons $b_k = \sum_{n=2^k}^{2^{k+1}-1} \frac{1}{\mathrm{L}(n)\mathrm{S}(n)} = \frac{1}{2^k}\sum_{n=2^k}^{2^{k+1}-1} \frac{1}{\mathrm{S}(n)} = \frac{a_k}{2^k}$.

On a $\mathrm{S}(n) \in [\![1, k+1]\!]$, et en regroupant par paquets, $a_k = \sum_{p=1}^{k+1} \frac{1}{p}\binom{k}{p-1} = \frac{1}{k+1}\sum_{p=1}^{k+1}\binom{k+1}{p} = \frac{1}{k+1}(2^{k+1}-1)$. Ainsi, $b_k = \frac{1}{k+1}\left(2 - \frac{1}{2^k}\right) \sim \frac{2}{k+1}$. On en déduit que $\sum b_k$ diverge, et donc aussi $\sum \frac{1}{\mathrm{L}(n)\mathrm{S}(n)}$.

Exercice 8

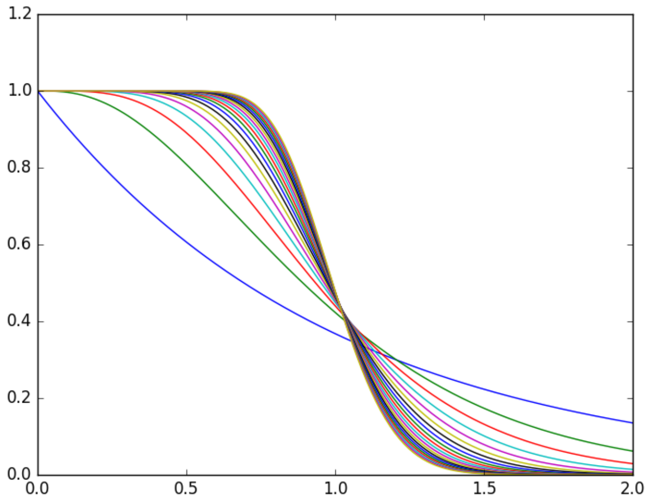
a)

```
def P(n, x):
    s = 1
    f = 1
    for k in range(1, n):
        f *= k
        s += (n * x)**k / f
    return s
```

b) On réalise le script suivant :

```
X = np.linspace(0, 2, 256)
for n in range(1, 41, 2):
    Y = [np.exp(-n * x) * P(n, x) for x in X]
    plt.plot(X, Y)
```

qui laisse penser que la suite (f_n) converge simplement sur $[0,1[$ et sur $]1,+\infty[$, respectivement vers 1 et vers 0..



c) $k! = n! \times (n+1)\cdots k \geqslant n! \times n^{k-n}$, donc $n^{k-n} \leqslant \dfrac{k!}{n!}$.

On a $1 = \mathrm{e}^{-nx}\displaystyle\sum_{k=0}^{+\infty}\frac{(nx)^k}{k!}$ donc $1 - f_n(x) = \mathrm{e}^{-nx}\displaystyle\sum_{k=n}^{+\infty}\frac{(nx)^k}{k!}$ et $0 \leqslant 1 - f_n(x) \leqslant \mathrm{e}^{-nx}\displaystyle\sum_{k=n}^{+\infty}\frac{(nx)^k}{n!n^{k-n}} = \frac{e^{-nx}n^n}{n!}\sum_{k=n}^{+\infty}x^k = \frac{\mathrm{e}^{-nx}(nx)^n}{n!(1-x)}$.

Posons $u_n = \dfrac{e^{-nx}(nx)^n}{n!}$. On calcule $\dfrac{u_{n+1}}{u_n} = x\,\mathrm{e}^{-x}\Big(1+\dfrac{1}{n}\Big)^n$ donc $\lim \dfrac{u_{n+1}}{u_n} = x\,\mathrm{e}^{1-x}$. Une étude de la fonction $x \mapsto x\,\mathrm{e}^{1-x}$ montre que pour tout $x \in [0,1[$, $xe^{1-x} < 1$, donc $\lim u_n = 0$ et (f_n) converge simplement vers 1 sur $[0,1[$.

d) Posons $\alpha_k = \dfrac{(nx)^k}{k!}$. Pour $0 \leqslant k \leqslant n-1$, $\dfrac{\alpha_{k+1}}{\alpha_k} = \dfrac{nx}{k+1} \geqslant x > 1$ donc $\alpha_k < \alpha_{k+1}$. D'où : $f_n(x) \leqslant \mathrm{e}^{-nx}\displaystyle\sum_{k=0}^{n-1}\frac{(nx)^n}{n!} = n\,\mathrm{e}^{-nx}\frac{(nx)^n}{n!}$.

On a $0 \leqslant f_n(x) \leqslant nu_n$ (avec les notations de la question précédente) et $\lim \dfrac{(n+1)u_{n+1}}{nu_n} = x\,\mathrm{e}^{1-x}$. Une étude de la fonction $x \mapsto x\,\mathrm{e}^{1-x}$ montre que pour tout $x > 1$, $xe^{1-x} < 1$, donc $\lim nu_n = 0$ et (f_n) converge simplement vers 0 sur $]1,+\infty[$.

e) $1 = \mathrm{e}^{-n}\displaystyle\sum_{k=0}^{+\infty}\frac{n^k}{k!}$ donc $u_n = 1 - \mathrm{e}^{-n}\displaystyle\sum_{k=n}^{+\infty}\frac{n^k}{k!}$.

On a $a_n + b_n = \dfrac{n!\,\mathrm{e}^n}{n^n} - c_n$ donc $\dfrac{e^{-n}n^n}{n!}(a_n + b_n - c_n) = 1 - 2\dfrac{\mathrm{e}^{-n}\,n^n}{n!}c_n = 1 - 2u_n$.

f) Si $k \geqslant 2n$, $\dfrac{\lambda_{n,k}}{\lambda_{n,k-1}} = \dfrac{n}{k} \leqslant \dfrac{1}{2}$ donc $\lambda_{n,k} \leqslant \Big(\dfrac{1}{2}\Big)^{k-2n+1}\lambda_{n,2n} = \Big(\dfrac{1}{2}\Big)^{k-2n+1}\dfrac{n^n n!}{(2n)!}$. Or $\dfrac{(2n)!}{n!} = (n+1)(n+2)\cdots(n+n) \geqslant n^n$, donc $\lambda_{n,k} \leqslant \Big(\dfrac{1}{2}\Big)^{k-2n+1}$. De ceci il résulte que $0 \leqslant a_n \leqslant 1$.

Par ailleurs, $b_n - c_n = \displaystyle\sum_{k=0}^{n-1}(\lambda_{n,k+n} - \lambda_{n,k})$.

Il reste à prouver que $0 \leqslant b_n - c_n \leqslant 1$ pour conclure : $1 - 2u_n = \mathrm{O}\Big(\dfrac{1}{\sqrt{n}}\Big)$ (formule de Stirling) donc $\lim u_n = \dfrac{1}{2}$.

Probabilités

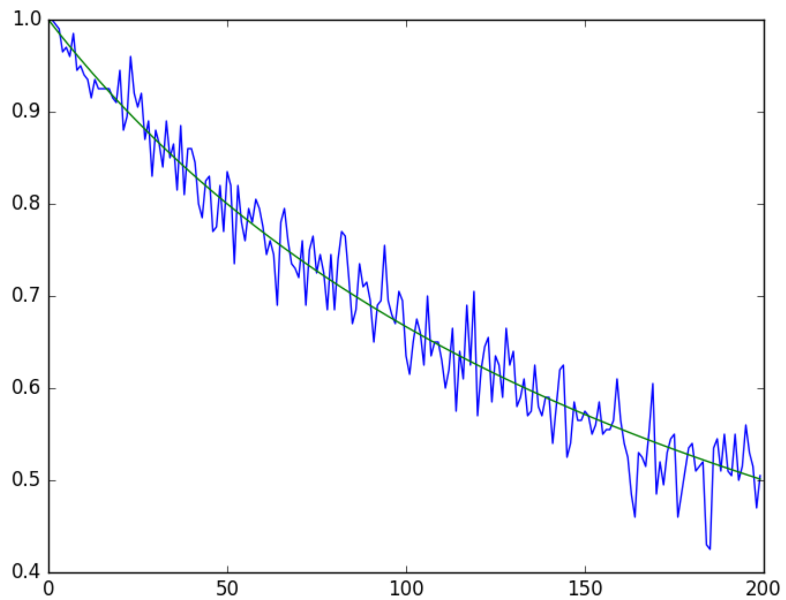
Exercice 9

a)

```python
def simulation(p, k):
    s = 0
    while s < k:
        s += 1 + rd.binomial(1, p)
    if s == k:
        return 1
    return 0
```

b) Le script ci-dessous superpose la proportion de réussite en fonction de p avec $1/\mathrm{E}(\mathrm{Y}_1) = 1/(1+p)$. Pour chaque valeur de p on réalise 200 expériences avec $k = 1000$.

```python
x = []
y = []
k = 1000
for p in np.arange(0, 1, 1/200):
    x.append(0)
    for _ in range(200):
        x[-1] += simulation(p, k)
    x[-1] /= 200
    y.append(1/(1+p))
plt.plot(x)
plt.plot(y)
```



c) $\mathrm{P}(\mathrm{E}_k \cap (\mathrm{Y}_1 = j)) = \mathrm{P}(\mathrm{Y}_1 = j) \times \mathrm{P}(\mathrm{E}_k \mid \mathrm{Y}_1 = j) = \mathrm{P}(\mathrm{Y}_1 = j) \times \mathrm{P}(\mathrm{E}_{k-j}) = f_j u_{k-j}$.

d) $\displaystyle\sum_{j=1}^{+\infty} \mathrm{P}(\mathrm{Y}_1 = j) = 1$ donc $\mathrm{P}(\mathrm{E}_k) = \displaystyle\sum_{j=1}^{+\infty} \mathrm{P}(\mathrm{E}_k \cap (\mathrm{Y}_1 = j))$. Mais si $j > k$, $\mathrm{P}(\mathrm{E}_k \cap (\mathrm{Y}_1 = j)) = 0$ donc d'après la question précédente, $u_k = \displaystyle\sum_{j=1}^{k} f_j u_{k-j}$.

e) $0 \leqslant u_k \leqslant 1$ donc si $t \in [0,1[$, $u_k t^k = \mathrm{O}(t^k)$ et $\sum t^k$ converge donc $\sum u_k t^k$ aussi. Ainsi, le rayon de convergence R vérifie $\mathrm{R} \geqslant 1$.

Pour tout $t \in [-1,1[$, $u(t)f(t) = \displaystyle\sum_{k=0}^{+\infty} \sum_{j=0}^{k} f_j u_{k-j} t^k = \sum_{k=1}^{+\infty} u_k t^k$ (car $f_0 = 0$) donc $u(t)f(t) = u(t) - 1$ et $u(t) = \dfrac{1}{1-f(t)}$.

f) Dans le cas d'une loi géométrique, $f_k = (1-p)^{k-1}p$ pour $k \geqslant 1$ et $f(t) = \sum_{k=1}^{+\infty} p(1-p)^{k-1}t^k = \dfrac{pt}{1-(1-p)t}$. Ainsi, $u(t) = \dfrac{1-(1-p)t}{1-t}$. On calcule $u(t) = 1 + \sum_{k=1}^{+\infty} pt^k$ donc $u_k = p$. Par ailleurs, $\mathrm{E}(\mathrm{Y}_1) = \dfrac{1}{p}$.

Dans le cas d'une loi de Bernoulli, $f_1 = 1-p$, $f_2 = p$ et $f_k = 0$ sinon donc $f(t) = (1-p)t + pt^2$ et $u(t) = \dfrac{1}{1-(1-p)t-pt^2} = \dfrac{1}{(1-t)(1+pt)}$. On calcule $u(t) = \dfrac{1}{1+p}\sum_{k=0}^{+\infty} t^k + \dfrac{p}{1+p}\sum_{k=0}^{+\infty}(-1)^k(pt)^k$. Ainsi, $u_k = \dfrac{1+(-1)^k p^{k+1}}{1+p}$ et $\lim u_k = \dfrac{1}{1+p}$. Par ailleurs, $\mathrm{E}(\mathrm{Y}_1) = 1+p$.

Dans les deux cas on observe que $\lim u_k = \dfrac{1}{\mathrm{E}(\mathrm{Y}_1)}$.

Exercice 10

a) On peut simuler le jeu à p joueurs de la façon suivante (chacun d'eux est représenté par un entier de $\mathbb{Z}/p\mathbb{Z}$) :

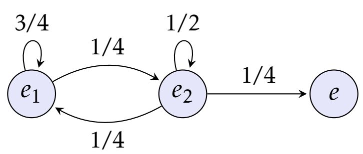
```
def experience(p):
    x, y = 0, 1
    n = 0
    while x != y:
        x = (x + 2 * rd.randint(0, 2) - 1) % p
        y = (y + 2 * rd.randint(0, 2) - 1) % p
        n += 1
    return n
```

On réalise 100 000 expériences pour estimer l'espérance de la variable T :

```
nb_exp = 100000
n = 0
for _ in range(nb_exp):
    n += experience(5)
print(n / nb_exp)
```

ce qui laisse penser que $\mathrm{E}(\mathrm{T}) = 12$.

b) Notons e_1 l'événement « les deux discoplanes sont entre les mains de deux voisins immédiats », e_2 l'événement « les deux discoplanes sont entre les mains de joueurs non voisins immédiats » et e l'événement « les deux discoplanes sont entre les mains d'un même joueur ». On peut représenter le jeu par un diagramme de Markov :

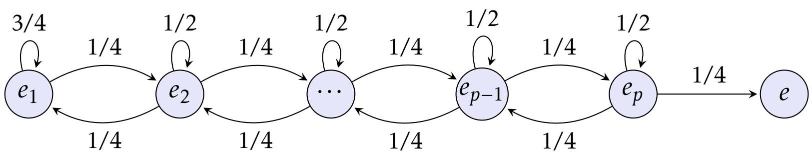


On dispose des relations $\begin{cases} a_{n+1} = \dfrac{3}{4}a_n + \dfrac{1}{4}b_n \\ b_{n+1} = \dfrac{1}{4}a_n + \dfrac{1}{2}b_n \end{cases}$ et $\mathrm{P}(t=n) = \dfrac{1}{4}b_{n-1}$. Ainsi, $\sum_{n=1}^{+\infty}\mathrm{P}(t=n)z^n = \dfrac{z}{4}\sum_{n=1}^{+\infty} b_n z^n = \dfrac{z}{4}\mathrm{B}(z)$.

Les relations de récurrence traduisent les égalités : $\begin{cases} \mathrm{A}(z) - 1 = \dfrac{3z}{4}\mathrm{A}(z) + \dfrac{z}{4}\mathrm{B}(z) \\ \mathrm{B}(z) = \dfrac{z}{4}\mathrm{A}(z) + \dfrac{z}{2}\mathrm{B}(z) \end{cases}$; la résolution fournit $\mathrm{B}(z) = \dfrac{4z}{5z^2-20z+16}$.

On a donc $\sum_{n=1}^{+\infty}\mathrm{P}(t=n)z^n = \dfrac{z^2}{5z^2-20z+16} = f(z)$ et $\mathrm{E}(t) = \sum_{n=1}^{+\infty} n\mathrm{P}(t=n) = f'(1) = 12$.

c) Pour un polygone à $2p+1$ côtés, le jeu se modélise par une chaîne de markov de la forme suivante :



Notons T_i la variable aléatoire qui est égale à la durée du jeu lorsqu'on part de l'état e_i. On dispose des relations :

$$\begin{cases} E(T_1) = 1 + \dfrac{3}{4}E(T_1) + \dfrac{1}{4}E(T_2) \\ E(T_i) = 1 + \dfrac{1}{4}E(T_{i-1}) + \dfrac{1}{2}E(T_i) + \dfrac{1}{4}E(T_{i+1}) \qquad \text{si } 2 \leqslant i \leqslant p-1 \\ E(T_p) = 1 + \dfrac{1}{2}E(T_p) + \dfrac{1}{4}E(T_{p-1}) \end{cases}$$

Ce système s'écrit matriciellement :

$$\begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \begin{pmatrix} E(T_1) \\ E(T_2) \\ \vdots \\ \vdots \\ E(T_p) \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ \vdots \\ \vdots \\ 4 \end{pmatrix}$$

En réalisant les opérations élémentaires $L_2 \leftarrow L_2 + L_1, \dots, L_p \leftarrow L_p + L_{p-1}$ le système devient triangulaire supérieur, en réalisant ensuite les opérations élémentaires $L_{p-1} \leftarrow L_{p-1} + L_p, \dots, L_1 \leftarrow L_1 + L_2$ on obtient le système trivial, qui fournit $E(T_1) = 4(1 + 2 + 3 + \cdots + p) = 2p(p+1)$.