Exercice. Etude préalable

1. Soit $P \in \mathbb{C}[X]$ de degré 2 de racines α et β. Exprimer la racine de P' en fonction de α et β.
2. Soit $P \in \mathbb{C}[X]$ de degré n. Que peut-on dire de la racine de $P^{(n-1)}$?

Corrigé.

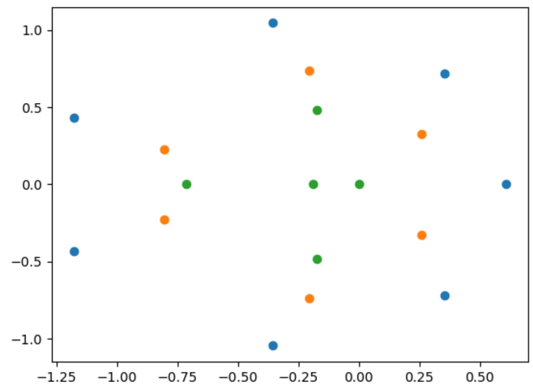
1. Nous avons
$$P = c(X-\alpha)(X-\beta) = cX^2 - c(\alpha+\beta)X + c\alpha\beta.$$
Donc
$$P' = 2cX - c(\alpha+\beta).$$
Ainsi la racine de P' est donnée par $\frac{\alpha+\beta}{2}$.
2. On écrit
$$P = c\prod_{k=1}^{n}(X-\alpha_k).$$
Alors
$$P = cX^n - c\left(\sum_{k=1}^{n}\alpha_k\right)X^{n-1} + Q,$$
avec $Q \in \mathbb{C}[X]$ de degré inférieur ou égal à $n-2$. Ainsi
$$P^{(n-1)} = cn!X - c(n-1)!\sum_{k=1}^{n}\alpha_k.$$
Donc la racine de $P^{(n-1)}$ est donnée par
$$\frac{1}{n}\sum_{k=1}^{n}\alpha_k,$$
il s'agit donc de l'isobarycentre des racines de P comptées avec multiplicité.

Exercice. Partie Python

1. Ecrire une fonction Python qui prend en variable un polynôme et renvoie la représentation de ses racines dans le plan complexe.
2. Ecrire une fonction Python qui prend en variable un polynôme et renvoie la représentation des racines de P et de ses dérivées. Pour n'afficher les racines qu'à la demande de l'utilisateur, on pourra utiliser la fonction plt.waitforbuttonpress().
3. Que peut-on conjecturer sur l'enveloppe complexe des racines du polynômes et de ses dérivées ?

Corrigé.

```python
import math
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
from numpy.polynomial import Polynomial

P = Polynomial([-3, 2, 0, 1,4, 6, 7, 4])

#Question 1

def racine(P):
    x = [P.roots()[i].real for i in range(len(P.roots()))]
    y = [P.roots()[i].imag for i in range(len(P.roots()))]
    plt.scatter(x,y)
    plt.show()

#Question 2

def racineiterees(P):
    n = P.degree()
    x = [P.roots()[i].real for i in range(len(P.roots()))]
    y = [P.roots()[i].imag for i in range(len(P.roots()))]
    plt.scatter(x,y)
    plt.waitforbuttonpress()
    Q = P
    for i in range(n):
        Q = Q.deriv()
        x = [Q.roots()[i].real for i in range(len(Q.roots()))]
        y = [Q.roots()[i].imag for i in range(len(Q.roots()))]
        plt.scatter(x,y)
        plt.waitforbuttonpress()

```



Nous pouvons conjecturer que les enveloppes convexes sont décroissantes pour l'inclusion.

Exercice. Partie théorique

1. On considère $(z_i)_{1\leq i\leq n} \in (\mathbb{C}^*)^n$. Montrer que 0 est barycentre à coefficients positifs des $(z_i)_{1\leq i\leq n}$ si et seulement s'il l'est des $\left(\frac{1}{z_i}\right)_{1\leq i\leq n}$.
2. On considère $\varphi : \mathbb{C}_n[X]\backslash\{0\} \longrightarrow \mathbb{C}_p[X]$ définie par
$$\forall P \in \mathbb{C}_p[X]\backslash\{0\}, \varphi(P) = \frac{P'}{P}.$$
Montrer que l'application φ transforme les produits en sommes.
3. En déduire la décomposition en éléments simples de $\frac{P'}{P}$.
4. Démontrer le théorème de Gauss-Lucas : toute racine de P' est combinaison convexe des racines de P.

Corrigé.

1. On suppose que 0 est barycentre à coefficients positifs des $(z_i)_{1\leq i\leq n}$: il existe $(\lambda_i)_{1\leq i\leq n} \in (\mathbb{R}_+)^n$ tels que
$$0 = \sum_{i=1}^{n} \lambda_i z_i.$$
Alors, comme les λ_i sont réels et les z_i non nuls,
$$0 = \overline{0} = \sum_{i=1}^{n} \lambda_i \overline{z_i} \frac{z_i}{z_i} = \sum_{i=1}^{n} \underbrace{\lambda_i |z_i|^2}_{=:\mu_i \in \mathbb{R}_+} \frac{1}{z_i} = \sum_{i=1}^{n} \mu_i \frac{1}{z_i}.$$

Ce qui montre que 0 est barycentre à coefficients positifs des $\left(\frac{1}{z_i}\right)_{1\leq i\leq n}$. Réciproquement on applique ce qui précède avec la famille $\left(\frac{1}{z_i}\right)_{1\leq i\leq n}$ pour obtenir que 0 est barycentre à coefficients positifs des $(z_i)_{1\leq i\leq n} = \left((z_i^{-1})^{-1}\right)_{1\leq i\leq n}$.

2. Soit $P_1, P_2 \in \mathbb{C}_n[X]\backslash\{0\}$. Alors

$$\varphi(P_1P_2) = \frac{(P_1P_2)'}{P_1P_2} = \frac{P_1'P_2 + P_1P_2'}{P_1P_2} = \frac{P_1'}{P_1} + \frac{P_2'}{P_2} = \varphi(P_1) + \varphi(P_2).$$

3. On écrit la décomposition en produits d'irréductibles du polynôme $P \in \mathbb{C}_n[X]$:

$$P = c\prod_{k=1}^{m}(X-\alpha_k)^{\nu_k}.$$

Alors la décomposition en éléments simples de $\frac{P'}{P}$ est donnée par

$$\frac{P'}{P} = \varphi(P) = \varphi(c) + \sum_{k=1}^{m}\nu_k\varphi(X-\alpha_k) = 0 + \sum_{k=1}^{m}\frac{\nu_k}{X-\alpha_k}.$$

4. Soit $z \in \mathbb{C}$ racine du polynôme dérivée P'. Si z est racine du polynôme P alors z est combinaison convexe triviale des racines P. Sinon, d'après ce qui précède,

$$0 = \frac{P'(z)}{P(z)} = \sum_{k=1}^{m}\frac{\nu_k}{z-\alpha_k} = \sum_{k=1}^{m}\frac{\nu_k}{|z-\alpha_k|^2}(\overline{z}-\overline{\alpha_k}),$$

i.e.

$$0 = \overline{0} = \sum_{k=1}^{m}\frac{\nu_k}{|z-\alpha_k|^2}(z-\alpha_k),$$

i.e.

$$z = \frac{\sum_{k=1}^{m}\frac{\nu_k}{|z-\alpha_k|^2}\alpha_k}{\sum_{k=1}^{m}\frac{\nu_k}{|z-\alpha_k|^2}}.$$

Donc z est bien combinaison convexe de racines de P.

Exercice. Partie théorique
Soit $\alpha \in \mathbb{R}_+^*$. On considère deux variables aléatoires X et Y sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ à valeurs dans $[\![2, +\infty[\![$ dont la loi conjointe est donnée par

$$\forall (p,q) \in [\![2, +\infty[\![^2, \mathbb{P}(X = p, Y = q) = \frac{\alpha}{p^q}.$$

1. Donner la loi marginale de la variable aléatoire X.
2. Calculer, pour $n \in [\![2, +\infty[\![$,
$$r_n := \sum_{p=2}^{n} \mathbb{P}(X = p).$$
3. En déduire la valeur de α.
4. La variable aléatoire X admet-elle une espérance finie.
5. Donner une approximation décimale correcte de $\mathbb{P}(X = Y)$.

Corrigé.

1. Pour tout $p \in [\![2, +\infty[\![$,
$$\mathbb{P}(X = p) = \sum_{q=1}^{+\infty} \mathbb{P}(X = p, Y = q) = \alpha \sum_{q=2}^{+\infty} \left(\frac{1}{p}\right)^q = \alpha \frac{\left(\frac{1}{p}\right)^2}{1 - \frac{1}{p}} = \frac{\alpha}{p(p-1)}.$$
Donc X est à valeurs dans $[\![2, +\infty[\![$ de loi
$$\delta_X = \alpha \sum_{p=2}^{+\infty} \frac{1}{p(p-1)} \delta_p.$$

2. Soit $n \in [\![2, +\infty[\![$. Alors, d'après la question précédente,
$$r_n = \alpha \sum_{p=2}^{n} \frac{1}{p(p-1)} = \alpha \left(\sum_{p=2}^{n} \frac{1}{p-1} - \sum_{p=2}^{n} \frac{1}{p} \right) = \alpha \left(1 - \frac{1}{n}\right).$$

3. Comme X est une variable aléatoire, nous devons avoir
$$1 = \mathbb{P}(X \in [\![2, +\infty[\![) = \sum_{p=2}^{+\infty} \mathbb{P}(X = p) = \lim_{n \to +\infty} r_n = \alpha.$$

4. La série
$$\sum_{p=2}^{+\infty} p\mathbb{P}(X = p) = \sum_{p=2}^{+\infty} \frac{1}{p-1}$$
est divergente, donc la variable aléatoire X n'admet pas d'espérance.

5. Nous avons

$$\mathbb{P}(X=Y)=\sum_{p=2}^{+\infty}\mathbb{P}(X=p,Y=p)=\sum_{p=2}^{+\infty}\frac{1}{p^p}=\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\frac{1}{25}+\frac{1}{36}+...\simeq 0,4.$$

Exercice. Partie Python

1. Ecrire une fonction f pour x dans $[0,1[$ qui renvoie l'entier naturel p non nul tel que

$$r_{p-1}\leq x<r_p.$$

2. On considère

$$S_{p,q}=\sum_{k=2}^{q}\frac{1}{p^k}.$$

Ecrire une fonction g pour $p\in\mathbb{N}^*$ et $y\in[r_{p-1},r_p[$ qui renvoie l'entier naturel q non nul tel que

$$S_{p,q-1}\leq y-r_{p-1}<S_{p,q}.$$

3. Proposer une procédure informatique qui permet de simuler le résultat des deux variables aléatoires X et Y. On donne la fonction random() du module numpy.random qui renvoie x dans $[0,1[$ avec $\mathbb{P}(a<x<b)=b-a$. Puis simuler 25 expériences.

Corrigé.

1. Nous avons

$$r_1=0,\quad r_p=1-\frac{1}{p}\underset{p\to+\infty}{\nearrow}1.$$

Donc, pour tout $x\in[0,1[$, il existe un unique $p\in[\![2,+\infty[\![$ tel que

$$r_{p-1}\leq x<r_p.$$

2. Nous avons

$$S_{p,1}=0,\quad S_{p,q}=\sum_{k=2}^{q}\left(\frac{1}{p}\right)^k=\frac{\frac{1}{p^2}-\frac{1}{p^{q+1}}}{1-\frac{1}{p}}=\frac{1-\frac{1}{p^{q-1}}}{p(p-1)}\underset{q\to+\infty}{\nearrow}\frac{1}{p(p-1)},$$

et

$$y-r_{p-1}\in[0,r_p-r_{p-1}[=\left[0,\frac{1}{p(p-1)}\right[.$$

Donc il existe un unique $q\in[\![2,+\infty[\![$ tel que

$$S_{p,q-1}\leq y-r_{p-1}<S_{p,q}.$$

3. On considère U une variable aléatoire réelle de loi uniforme sur $[0,1]$. Alors, pour tout $p\in[\![2,+\infty[\![$,

$$\mathbb{P}(f(U)=p)=\mathbb{P}(r_{p-1}\leq U<r_p)=r_p-r_{p-1}=\mathbb{P}(X=p).$$

Donc

$$f(U) \overset{loi}{=} X.$$

Puis pour Y nous avons, pour tout $q \in [\![2, +\infty[\![$,

$$\begin{aligned}\mathbb{P}(g(U, f(U)) = q) &= \mathbb{P}(S_{f(U),q-1} \leq U - r_{f(U)-1} < S_{f(U),q}) \\ &= \mathbb{P}(S_{f(U),q-1} + r_{f(U)-1} \leq U < S_{f(U),q} + r_{f(U)-1}).\end{aligned}$$

Donc, par formule des probabilités totales, nous avons

$$\mathbb{P}(g(U, f(U)) = q) = \sum_{p=2}^{+\infty} \mathbb{P}(S_{p,q-1} + r_{p-1} \leq U < S_{p,q} + r_{p-1}, f(U) = p)$$

$$\begin{aligned}&= \sum_{p=2}^{+\infty} \mathbb{P}(S_{p,q-1} + r_{p-1} \leq U < S_{p,q} + r_{p-1}, r_{p-1} \leq U < r_p) \\ &= \sum_{p=2}^{+\infty} \mathbb{P}\left(S_{p,q-1} + r_{p-1} \leq U < \min(S_{p,q} + r_{p-1}, r_p)\right),\end{aligned}$$

avec, comme U est de loi uniforme,

$$\mathbb{P}\left(S_{p,q-1} + r_{p-1} \leq U < \min(S_{p,q} + r_{p-1}, r_p)\right) = \min(S_{p,q} + r_{p-1}, r_p) - S_{p,q-1} - r_{p-1}.$$

Or

$$\begin{aligned}S_{p,q} + r_{p-1} &= \frac{1 - \frac{1}{p^{q-1}}}{p(p-1)} + 1 - \frac{1}{p-1} = \frac{1 - \frac{1}{p^{q-1}} + p(p-1) - p}{p(p-1)} = \frac{p^{q-1} - 1 + p^{q+1} - 2p^q}{p^q(p-1)} \\ &= \frac{p^{q-1} - 1 - p^q}{p^q(p-1)} + \frac{p^{q+1} - p^q}{p^q(p-1)} \leq \frac{1-p}{p(p-1)} + 1 = 1 - \frac{1}{p} = r_p.\end{aligned}$$

Donc

$$\min(S_{p,q} + r_{p-1}, r_p) - S_{p,q-1} - r_{p-1} = S_{p,q} - S_{p,q} = \frac{1}{p^q}.$$

Par conséquent

$$\mathbb{P}(g(U, f(U)) = q) = \sum_{p=2}^{+\infty} \mathbb{P}\left(S_{p,q-1} + r_{p-1} \leq U < \min(S_{p,q} + r_{p-1}, r_p)\right) = \sum_{p=2}^{+\infty} \frac{1}{p^q} = \mathbb{P}(Y = q).$$

Donc

$$g(U, f(U)) \overset{loi}{=} Y.$$

```python
import numpy.random as rd

#question 1

def r(p):
    return 1 - 1/p

def f(x):
    n = 2
    while r(n) <= x:
        n=n+1
    return n

#question 2

def S(p,q):
    s = 0
    for k in range(2,q+1):
        s = s + 1/(p**k)
    return s

def g(x,p): #y dans [r(p-1), r(p)[
    q = 2
    while S(p,q) <= x - r(p-1):
        q=q+1
    return q

#question 3

N = 25

for i in range(N):
    u = rd.random()
    p = f(u)
    q = g(u,p)
    print(p,q)

```