Préparation des oraux – informatique

1. *(Centrale)* On note χ_n le polynôme caractéristique de $A_n = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1/n \\ 1 & \ddots & \ddots & \vdots & \vdots \\ 0 & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 1/n \end{pmatrix} \in \mathcal{M}_n(\mathbb{R})$.

a) Écrire une fonction Python prenant pour paramètre n et renvoyant les coefficients de χ_n sous forme d'un tableau `numpy` (cf. `numpy.poly`).

b) Afficher les coefficients de χ_n pour $n \in [\![2,8]\!]$ et conjecturer la valeur de χ_n. Démontrer ce résultat.

c) Afficher les modules des racines de χ_n pour $n \in [\![2,8]\!]$. Que peut-on conjecturer ? Démontrer ce résultat.

2. *(Centrale)* Programmer en Python le calcul de $u_n = \displaystyle\int_0^1 \left(3x^2 - 2x^3\right)^n \mathrm{d}x$.

Écrire u_n comme une somme et en déduire une autre implémentation du calcul de u_n.

Calculer u_n pour $n \in [\![0,50]\!]$ par les deux méthodes et commenter. Montrer que (u_n) converge vers 0.

Tracer les valeurs de $\sqrt{n}u_n$ pour $n = 100k,\ k \in [\![0,100]\!]$. Conjecturer.

Montrer que $u_n \sim \dfrac{A}{\sqrt{n}}$ où $A = \displaystyle\int_0^{+\infty} e^{-3t^2}\mathrm{d}t$ *(on pourra utiliser le changement de variable $u = 1 - x$).*

3. *(Centrale)* Donner un code Python permettant de calculer

$$u_n = \frac{1}{n}\sum_{k=1}^{n} \cos\left(\frac{k\pi}{n}\right)\arctan\left(\frac{k}{n}\right).$$

a) Tracer les u_n pour $n \in [\![1,100]\!]$. Que remarque-t-on ? Le démontrer.

b) On note ℓ la limite de (u_n) et $v_n = \dfrac{n}{\pi}(\ell - u_n)$. Tracer les v_n pour $n \in [\![1,100]\!]$. Conjecture ?

c) Soit g une fonction de classe $\mathcal{C}^2$ sur $[0,1]$ et $M = \max\limits_{[0,1]} |g''|$. Montrer que, pour $n \geq 2$:

$$\forall k \in [\![1,n]\!] \quad \forall t \in \left[\frac{k-1}{n}, \frac{k}{n}\right] \qquad \left|g(t) - g\left(\frac{k}{n}\right) - g'\left(\frac{k}{n}\right)\left(t - \frac{k}{n}\right)\right| \leq \frac{M}{2n^2}.$$

En déduire :

$$\int_0^1 g(t)\,\mathrm{d}t - \frac{1}{n}\sum_{k=1}^{n} g\left(\frac{k}{n}\right) = \frac{g(0) - g(1)}{2n} + o\left(\frac{1}{n}\right).$$

4. *(Centrale)* Pour $n \geq 3$, montrer que l'équation (E_n) $e^x = x^n$ admet une unique solution u_n dans $[1,2]$.

Afficher avec Python les u_n pour $n = 10k,\ k \in [\![1,100]\!]$. Conjecture ? Démontrer le résultat.

On note ℓ la limite de (u_n). Avec Python, conjecturer les valeurs a, b telles que

$$u_n - \ell = \frac{a}{n} + \frac{b}{n^2} + o\left(\frac{1}{n^2}\right).$$

Démontrer le résultat.

5. *(Centrale)* Justifier que $\langle P, Q\rangle = \displaystyle\int_{-1}^{1} PQ$ définit un produit scalaire et $N_\infty(P) = \max\limits_{[-1,1]} |P|$ une norme sur $E = \mathbb{R}[X]$.

Soit $F = \mathbb{R}_5[X]$; déterminer la base orthonormale $(E_0, \ldots, E_5)$ déduite de la base canonique par l'algorithme de Gram-Schmidt.

Tracer les graphes des E_i et déterminer les valeurs de t où $N_\infty(E_i)$ est atteinte.

Montrer que, si $P \in F$ vérifie $\|P\| = 1$, alors $N_\infty(P) \leq 3\sqrt{2}$. Quand a-t-on égalité ?

Trouver a et b tels que, pour tout P de F, $a\|P\| \leq N_\infty(P) \leq b\|P\|$. Donner des exemples de P pour lesquels il y a égalité à droite ou à gauche.

6. *(Centrale)* Intégrer $xy'' - y' + 4x^3 y = 0$ à l'aide du changement de variable $t = x^2$. On pose $a = \sqrt{\pi/2}$.

On trouve que la solution f vérifiant $f(a) = f'(a) = 1$ est $x \mapsto \sin\left(x^2\right) - \dfrac{\cos\left(x^2\right)}{2a}$.

Comparer les graphes sur $[a, 4]$ de cette solution exacte, de la solution approchée fournie par la méthode d'Euler (que l'on programmera) et de celle fournie par la fonction `scipy.integrate.odeint`.

7. *(Centrale)* Une particule se déplace sur une surface comportant 4 positions possibles : A_0 et A_3 qui sont des puits, A_1 et A_2 qui sont des positions intermédiaires. À chaque étape :

- si la particule est dans un puits, elle y reste avec une probabilité de 1
- si elle est en A_1, elle va en A_0 avec une probabilité p, en A_2 avec une probabilité $1 - p$
- si elle est en A_2, elle va en A_1 avec une probabilité p, en A_3 avec une probabilité $1 - p$

On note x_n la variable aléatoire donnant la position de la particule à l'étape n : $x_n(\Omega) = [\![0, 3]\!]$.

a) Écrire une fonction Python qui donne x_{n+1} en fonction de x_n et de p, puis une fonction renvoyant x_n en fonction de n, x_0 et p.

b) Tracer l'histogramme des x_n obtenues durant N essais (cf. `matplotlib.pyplot.bar`).

c) Soit $X_n = \begin{pmatrix} P(x_n = 0) \\ P(x_n = 1) \\ P(x_n = 2) \\ P(x_n = 3) \end{pmatrix}$. Trouver $A \in \mathcal{M}_4(\mathbb{R})$, indépendante de n, telle que $X_{n+1} = AX_n$ pour tout n.

d) Montrer que A est diagonalisable si et seulement si $p \in]0, 1[$.

e) Pour $p = 1/2$, diagonaliser A avec Python et calculer $\lim\limits_{n\to\infty} X_n$. Comparer avec les résultats précédents.

N.B. Prendre garde que, si `A` est un tableau `numpy` carré et `n` un entier naturel, `A**n` se contente d'élever chaque coefficient de `A` à la puissance `n` !! Cf. `numpy.linalg.matrix_power` pour un calcul efficace des puissances d'une matrice carrée, ou programmer soi-même une exponentiation rapide…

8. *(Centrale)* On donne $n + 1$ entiers naturels $a_0 < a_1 < \cdots < a_n$ et, pour $P \in \mathbb{R}_n[X]$, on pose

$$\|P\|_A = \max_{i\in[\![0,n]\!]} |P(a_i)| \quad \text{où} \quad A = (a_0, \ldots, a_n).$$

Écrire une fonction Python `N(A,P)` qui renvoie $\|P\|_A$.

Montrer que $\|\cdot\|_A$ est une norme sur $\mathbb{R}_n[X]$. On veut évaluer d_n, distance de X^n à $\mathbb{R}_{n-1}[X]$.

Pour $A = (0, 1)$, calculer le minimum des $\|X + a\|_A$ pour a décrivant $\mathbb{R}$.

Pour $A = (0, 1, 2)$, calculer le minimum des $\left\|X^2 + aX + b\right\|_A$ pour (a, b) décrivant $\mathbb{R}^2$ *(on pourra utiliser la fonction* `fmin` *de* `scipy.optimize` *qui prend comme argument une fonction f et un point initial et renvoie le minimum de f).*

Dans le cas général, pour $P \in \mathbb{R}_{n-1}[X]$, montrer qu'il existe une unique famille de réels $(b_0, \ldots, b_n)$ telle que

$$X^n + P(X) = \sum_{k=0}^{n} b_k \prod_{j\in[\![0,n]\!]\setminus\{k\}} (X - a_j).$$

Calculer $\sum\limits_{k=0}^{n} b_k$ et montrer que

$$\forall k \in [\![0, n]\!] \qquad \prod_{j\in[\![0,n]\!]\setminus\{k\}} |a_k - a_j| \geq \frac{n!}{\binom{n}{k}}.$$

En déduire que $\|X^n + P(X)\|_A \geq \dfrac{n!}{2^n}$.

Calculer d_n lorsque $A = (0, 1, \ldots, n)$.

9. *(Centrale)* Pour p dans $\mathbb{N}^*$, on définit le polynôme

$$Q_p(X) = (2p+2)\,X^{2p+1} - 2pX^{2p-1} - (2p-1)\,X^{2p-2} - \cdots - 2X - 1.$$

En utilisant le module `Polynomial`, écrire une fonction Python renvoyant Q_p.

Calculer les racines de Q_p pour $p \in \{3, 4, 5, 10\}$; quelle conjecture peut-on faire sur les racines réelles ?

Tracer les racines des Q_p pour $p < 30$ ainsi que le cercle unité ; quelle conjecture peut-on faire sur les racines complexes ?

Écrire une fonction Python renvoyant le minimum des modules des racines de Q_p.

Montrer que $R_p(X) = -X^{2p+1}Q_p(1/X)$ admet une unique racine réelle positive. En déduire qu'il en est de même de Q_p.

Montrer que $Q_p(X) = (2p+2)\,X^{2p+1} - \dfrac{1 - X^{2p+1}}{(1-X)^2} + \dfrac{(2p+1)\,X^{2p}}{1-X}$.

En déduire qu'à partir d'un certain rang $3/2 < \alpha_p < 2$, où α_p est la racine réelle positive de Q_p.

10. *(Centrale)* Dans $\mathbb{R}^n$ on désigne par $\|\cdot\|$ et $\langle \cdot, \cdot \rangle$ la norme euclidienne et le produit scalaire canoniques. Pour A et B dans $\mathcal{M}_n(\mathbb{R})$, on note (N) la propriété

$$\forall X \in \mathbb{R}^n \quad \|AX\| = \|BX\|.$$

Montrer que A et B vérifient (N) si et seulement si

$$\forall (i,j) \in [\![1, n]\!]^2 \quad \langle C_i(A), C_j(A)\rangle = \langle C_i(B), C_j(B)\rangle$$

où $C_j(M)$ désigne le j-ième vecteur colonne de la matrice M.

Coder en Python $d(A,B) = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} \left(\langle C_i(A), C_j(A)\rangle - \langle C_i(B), C_j(B)\rangle\right)^2$, puis une fonction booléenne renvoyant `True` si et seulement si A et B vérifient (N).

Montrer que, si A est inversible, A et B vérifient (N) si et seulement s'il existe une matrice orthogonale Ω telle que $B = \Omega A$.

11. *(Centrale)* On fixe un entier $N \geq 2$, un réel p dans $]0, 1[$ et l'on pose $q = 1 - p$. Une infinité de joueurs $(J_k)_{k\in\mathbb{N}}$ participent à un tournoi : le match 1 oppose J_0 à J_1 et J_0 l'emporte avec la probabilité p. Puis, pour $k \geq 2$, J_k affronte le vainqueur du match $k-1$ avec la probabilité p de gagner. Chaque perdant est définitivement éliminé.

Le tournoi s'achève lorsqu'un même joueur parvient à enchaîner N victoires consécutives, ledit joueur étant alors déclaré vainqueur.

On note $T_{N,p}$ le nombre de matchs nécessaires pour qu'un joueur soit vainqueur.

Pour $n \in \mathbb{N}^*$ on note A_n l'événement "il n'y a pas de vainqueur au match n".

Écrire une fonction Python `T(N,p)` qui renvoie $T_{N,p}$ suite à une simulation d'un tournoi.

Écrire une fonction `moyenne(N,p)` qui renvoie la moyenne des $T_{N,p}$ obtenus lors de 10^4 tournois.

Que donnent `moyenne(3,0.3)` et `moyenne(6,0.5)` ?

Montrer que

$$\forall n \in [\![1, N-1]\!] \quad P(A_n) = 1 \quad \text{et} \quad P(A_N) = 1 - q^{N-1}.$$

Que représente l'événement $A_{n-1} \backslash A_n$? Établir

$$\forall n \geq N+1 \quad P(A_{n-1}) - P(A_n) = pq^{N-1}P(A_{n-N}).$$

Montrer l'existence et calculer les valeurs de $\lim\limits_{n\to\infty} P(A_n)$ et $\sum\limits_{n=1}^{\infty} P(A_n)$.

En déduire l'espérance de $T_{N,p}$ et comparer avec les résultats des simulations.