

□ **Exercice 1** (RMS 19-971)

1. Python
2. Python
3. Par le lemme de relèvement, on écrit $f(t) = \rho(t)e^{iu(t)}$. Il suffit alors de faire le calcul

□ **Exercice 2** (RMS 17-953)

1. Python
2. Python
3. Ordre $n - 2$ car la matrice A_n est de rang 2.
4. On sait que

$$\alpha_n = \min_{X \neq 0} \frac{X^\top A_n X^\top}{\|X\|^2} \text{ et } \beta_n = \max_{X \neq 0} \frac{X^\top A_n X^\top}{\|X\|^2}$$

On voit que pour $X_n = (1, \dots, 1)^\top$, $X_n^\top A_n X_n \geq 0$ et pour $Y_n = (-1, 1, 0, \dots, 0)^\top$, $Y_n^\top A_n Y_n < 0$. On en déduit que $\alpha_n < 0 < \beta_n$.

5. En utilisant que $F_n \sim C\varphi^n$, on montre que $\frac{X_n^\top A_n X_n^\top}{\|X_n\|^2} \rightarrow \infty$ cela implique que $\beta_n \rightarrow +\infty$.

On voit que α_n est décroissante car un élément $Y_n \in \mathcal{M}_{n,1}(\mathbf{R})$ peut être prolongé en un élément de $\mathcal{M}_{n+1,1}(\mathbf{R})$ en ajoutant un 0. De plus, en utilisant que $F_n = C\varphi^n + C'\psi^n$ où $\psi = -1/\varphi$ on a

$$\frac{X^\top A X}{\|X\|^2} = Cq_\varphi(X) + C'q_\psi(X)$$

où q_φ est la forme quadratique associée à la matrice (φ^{i+j}) . On a donc

$$q_\varphi(X) = \left(\sum_{i=1}^n \varphi^i x_i \right)^2 \geq 0$$

Comme $C \geq 0$ et $C' \leq 0$, pour minorer la forme quadratique étudiée on doit majorer $q_\psi(X)$. On voit facilement par Cauchy Schwarz que

$$q_\psi(X) \leq \sum_{i=0}^{\infty} \psi^{2i}$$

On peut en déduire que α_n est minorée. Elle converge.

□ **Exercice 3** (RMS 19-993)

1. On peut faire le calcul explicitement.
2. Il y a une chance sur 3 que le rat sorte immédiatement et dans ce cas $T = t_1$, une chance sur 3 qu'il commence par la porte 2. A partir de là le nombre de coup pour sortir est donné par une loi géométrique de paramètre 1/2. On calcule facilement le temps total associé car le temps dans le tunnel est $t_1 - t_2 - t_1 - \dots$. De même dans le dernier cas.
3. Le temps d'attente suit une loi géométrique de raison 1/3. Une fois le temps connu, le nombre de t_2 et t_3 est donné par la loi binomiale.

□ **Exercice 4** (RMS 19-949)

1. OK
2. Python
3. Python. On conjecture que $d_n = \frac{1}{(n+1)^2}$.
4. F_n définie sur $\mathbf{R} \setminus \{-1, \dots, -(n+1)\}$
5. On multiplie la première ligne par $\prod_{k=1}^{n+1} (X+k)$. En développant selon cette ligne $G_n \in \mathbf{R}_n[X]$.

6. On voit que F_n et donc G_n s'annule en $1, 2, \dots, n$.

7. On sait que $G_n = \alpha(X-1) \cdots (X-n)$. Il n'est pas nécessaire de calculer α .

On voit que $a_n = F_n(0) = \frac{1}{(n+1)!} G_n(0) = (-1)^n \frac{\alpha}{n+1}$

On calcule $G_n(-1)$. En développant selon la première ligne, $G_n(-1) = n!b_n$. On en déduit que $b_n = (-1)^n \alpha(n+1)$ et donc $d_n = \frac{1}{(n+1)^2}$.

La propriété admise est le déterminant de Gramm.

□ **Exercice 5** (RMS 17-960)

1. On voit que $c_n = 2^n$. Cela se prouve en développant v_{n+1} en fonction de v_n où $v_n = u_n + 1$.

2. Pareil, cela fait toujours 5 sauf au premier rang.

□ **Exercice 6**

□ **Exercice 7**

Dans tout l'exercice, (a_n) est une suite à termes positifs.

On note $R_n = \sum_{k=1}^n \tilde{a}_k$ et $S_n = \sum_{k=1}^n a_k$ où $\tilde{a}_n = (a_1 \dots a_n)^{1/n}$.

1. Python

2. (a) OK

(b) On applique ce qui précède à kx_k

3. On voit que

$$\sum_{k=0}^{\infty} ka_k \sum_{n=k}^{\infty} \frac{1}{n(n+1)} = \sum_{k=0}^{\infty} a_k < +\infty$$

Par Stirling on montre que $n! \geq n^n e^{-n}$ et donc $\frac{1}{(n!)^{1/n}} \leq \frac{e}{n}$. Donc

$$\tilde{a}_n \leq \frac{1}{n(n!)^{1/n}} \sum_{k=1}^n ka_k \leq \frac{e}{n^2} \sum_{k=1}^n ka_k$$

□ **Exercice 8**

1. Python

2. Python

3. A x fixé dans $[0, a]$. On étudie la suite récurrente définie par $u_0 = 0$ et $u_{n+1} = u_n + \frac{1}{2\sqrt{a}}(x - u_n^2)$. On voit que la suite est croissante, à valeurs dans $[0, \sqrt{x}]$ tend vers \sqrt{x}

4. Pour la CU. Pour ε fixé on cherche N tel que $n \geq N$ implique $\|\sqrt{x} - P_n\|_{\infty} < \varepsilon$. Le résultat est vrai sur $[0, \varepsilon^2]$

De plus

$$\sqrt{x} - P_{n+1} = (\sqrt{x} - P_n) \left[1 - \frac{\sqrt{x} + P_n}{2\sqrt{a}} \right]$$

donc sur $[\varepsilon^2, a]$,

$$\sqrt{x} - P_{n+1} \leq (\sqrt{x} - P_n) \left[1 - \frac{\varepsilon}{2\sqrt{a}} \right]$$

□ **Exercice 9**

1. Théorème de Cauchy

2. Python

3. Python

4. Python

5. Il suffit de faire le calcul.

□ **Exercice 10**

3. Soit $\phi : t \mapsto G(\gamma_1(t), \dots, \gamma_p(t))$. On a $\phi'(t) = \sum_{k=1}^p G(\gamma_1(t), \dots, \gamma'_k(t), \dots, \gamma_p(t))$.

4. On a $\chi_A(x) = \det(xE_1 - C_1 | \dots | xE_n - C_n)$. En dérivant

$$\chi'_A(x) = \sum_{k=1}^n \det(xE_1 - C_1 | \dots | E_k | \dots | xE_n - C_n) = \text{tr}(H(x))$$

5. On se donne n matrices carrées R_0, \dots, R_{n-1} telles que $H(x) = \sum_{k=0}^{n-1} R_{n-1-k} x^k$.

(a) OK

(b) Comme $(xI_n - A)H(x) = \chi_A(x)$ on a

$$\chi_A(x)I_n = (xI_n - A) \sum_{k=0}^{n-1} R_{n-1-k} x^k = \sum_{k=1}^{n-1} (R_{n-k} - R_{n-1-k}A) x^k + R_0 x^n - R_{n-1}A$$

En posant $\chi_A(x) = a_0 + a_1 x + \dots + a_n x^n$, on obtient que $R_0 = a_n I_n = I_n$, $R_{n-k} = R_{n-1-k}I_n + a_k I_n$ et $-R_{n-1}A = a_0 I_n$.

(c) On voit que $R_0 = x^n I_n$ car dans $H(x)$ les coefficients diagonaux sont des polynômes unitaire de degré $n-1$ alors qu'en dehors de la diagonale ils sont de degré au plus $n-2$. On peut alors procéder par récurrence, en utilisant d'après 4) que $(k+1)a_{k+1} = \text{tr}(R_{n-1-k})$.

6. Conclure finalement que $P = \chi_A$ et $C = A^{-1}$.