

---

## MERCREDI 10 JUIN

---

Certains énoncés ont été transmis par des candidats aux concours en 2025 - merci à eux.

Autant que possible, j'ai veillé à établir des énoncés mathématiquement corrects, faites-moi signe si vous rencontrez des points litigieux.

---

<b>Thèmes abordés</b>	
135-py-50	Méthode d'Euler
135-py-51	Fonctions de matrice
135-py-65	Matrices de Jacobi
135-py-66	Matrice de Vandermonde
136-py-08	Polynômes
136-py-10	Nombres de Salem (nombres algébriques)
136-py-11	Matrices circulantes
136-py-15	Matrices de Householder
136-py-25	Critères d'équirépartition
136-py-28	Résultant de deux polynômes

---

[ 1. ] Pour  $x \in \mathbb{R}$ , déterminer

$$\lim_{n \rightarrow +\infty} \left(1 + \frac{x}{n}\right)^n.$$

[ 2. ] Pour  $n \in \mathbb{N}^*$ , on pose

$$A_n = \begin{pmatrix} 1 & 1/n & 1/n \\ 1/n & 1 & 1/n \\ 1/n & 1/n & 1 \end{pmatrix}.$$

[ 2.a. ] Écrire une fonction Python d'argument  $n \in \mathbb{N}^*$  qui renvoie la matrice  $A_n$ .

[ 2.b. ] Écrire un code Python renvoyant les valeurs propres de  $A_n^n$  pour  $n \in \llbracket 5, 10 \rrbracket$ .

[ 2.c. ] Diagonaliser  $A_n$  et déterminer la limite de  $A_n^n$  lorsque  $n$  tend vers  $+\infty$ . Vérifier le résultat avec Python.

[ 3. ] On note  $A_{n,m} \in \mathfrak{M}_m(\mathbb{R})$  la matrice dont les coefficients diagonaux sont égaux à 1 et les autres coefficients sont égaux à  $1/n$ .

[ 3.a. ] Exprimer la matrice  $A_{n,m}$  en fonction de  $n \in \mathbb{N}^*$ , de  $I_n$  et de la matrice  $J_m \in \mathfrak{M}_m(\mathbb{R})$  dont tous les coefficients sont égaux à 1.

[ 3.b. ] Déterminer la limite de  $A_{n,m}^n$  lorsque  $n$  tend vers  $+\infty$  comme une combinaison linéaire des matrices  $I_m$  et  $J_m$ .

[ 1. ] Lorsque  $n$  est assez grand pour que  $|x/n| < 1$ , on peut composer par la fonction  $\ln$ .

$$\left(1 + \frac{x}{n}\right)^n = \exp\left[n \ln\left(1 + \frac{x}{n}\right)\right] = \exp\left[x + o(1/n)\right] \xrightarrow{n \rightarrow +\infty} e^x.$$

[ 2.a. ] Il suffit de ne pas calculer six fois le même quotient pour que le code soit élégant.

```
def A(n):
    q = 1/n
    return np.array([[1,q,q],[q,1,q],[q,q,1]])
```

[ 2.b. ] Il faut savoir calculer les puissances d'une matrice (avec la fonction `matrix_power` du module `numpy.linalg`), ainsi que son spectre (la fonction `eigvals` du même module donne une valeur approchée des valeurs propres).

```
for n in range(5, 11):
    M = alg.matrix_power(A(n), n)
    print(alg.eigvals(M))
```

Il semblerait qu'une valeur propre double soit proche de 0,35 et qu'une valeur propre simple soit proche de 6,19.

[ 2.c. ] Comme  $J_3^2 = 3J_3$ , la matrice  $\Pi = (1/3) \cdot J_3$  est une matrice de projection. La matrice  $I_3 - \Pi$  est donc la projection associée à  $\Pi$  et la décomposition

$$J_3 = 3 \cdot \Pi + 0 \cdot (I_3 - \Pi)$$

nous donne la forme diagonalisée de chaque polynôme en  $J_3$ . En particulier, comme

$$A_n = \frac{1}{n} \cdot J_3 + \left(1 - \frac{1}{n}\right) \cdot I_3 = \frac{3}{n} \cdot \Pi + \left(1 - \frac{1}{n}\right) \cdot (\Pi + I_3 - \Pi) = \left(1 + \frac{2}{n}\right) \cdot \Pi + \left(1 - \frac{1}{n}\right) \cdot (I_3 - \Pi)$$

alors

$$\forall k \in \mathbb{N}, \quad A_n^k = \left(1 + \frac{2}{n}\right)^k \cdot \Pi + \left(1 - \frac{1}{n}\right)^k \cdot (I_3 - \Pi).$$

En particulier,

$$A_n^n = \left(1 + \frac{2}{n}\right)^n \cdot \Pi + \left(1 - \frac{1}{n}\right)^n \cdot (I_3 - \Pi) \xrightarrow{n \rightarrow +\infty} e^2 \cdot \Pi + e^{-1} \cdot (I_3 - \Pi).$$

#### • Variante

Sachant que  $J_3^2 = 3J_3$ , on sait que les seules valeurs propres possibles pour  $J_3$  sont 0 et 3. En étudiant les noyaux de  $J_3$  et de  $J_3 - 3I_3$ , on forme la matrice

$$P = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix}$$

qui est inversible et qui vérifie

$$P^{-1}J_3P = \text{Diag}(3, 0, 0).$$

On en déduit que

$$P^{-1}A_nP = \frac{1}{n} \cdot P^{-1}J_3P + \left(1 - \frac{1}{n}\right) \cdot I_3$$

et donc que

$$P^{-1}A_n^nP = (P^{-1}A_nP)^n = \text{Diag}\left(\left(1 + \frac{2}{n}\right)^n, \left(1 - \frac{1}{n}\right)^n, \left(1 - \frac{1}{n}\right)^n\right) \xrightarrow{n \rightarrow +\infty} \text{Diag}(e^2, e^{-1}, e^{-1}).$$

✎ On peut confirmer numériquement cette étude si on sait calculer l'inverse d'une matrice (fonction `inv` du module `numpy.linalg`) ainsi que le produit de trois matrices (fonction `dot`).

```
P = np.array([[1,1,1],[1,-1,0],[1,0,-1]])
```

```
def forme_diag(n):
    D = alg.inv(P).dot(A(n)).dot(P)
    return alg.matrix_power(D, n)
```

✎ Bien évidemment, les coefficients de l'ordre de  $10^{-14}$  ou de  $10^{-29}$  doivent être tenus pour nuls !  
Les coefficients diagonaux de la matrice calculée pour  $n = 100$  sont respectivement proches de 0,724 (une fois) et de 0,366 (deux fois) et on calcule par ailleurs  $e^2 \approx 7,389056$  et  $e^{-1} \approx 0,367879$ .

[ 3.a. ] Comme plus haut, on remarque facilement que  $J_m^2 = mJ_m$  et on en déduit que

$$A_{n,m} = \frac{1}{n} \cdot J_m + \left(1 - \frac{1}{n}\right) \cdot I_m = \left(1 + \frac{m-1}{n}\right) \cdot \Pi_m + \left(1 - \frac{1}{n}\right) \cdot (I_m - \Pi_m)$$

où  $\Pi_m = (1/m) \cdot J_m$  est une projection.

[ 3.b. ] Comme plus haut,  $\Pi_m$  et  $(I_m - \Pi_m)$  sont deux projections qui vérifient

$$\Pi_m(I_m - \Pi_m) = (I_m - \Pi_m)\Pi_m = O_m.$$

On en déduit que

$$\forall \alpha, \beta \in \mathbb{C}, \forall k \in \mathbb{N}^*, \quad (\alpha \cdot \Pi_m + \beta \cdot [I_m - \Pi_m])^k = \alpha^k \cdot \Pi_m + \beta^k \cdot (I_m - \Pi_m)$$

et en particulier que

$$\forall n \in \mathbb{N}^*, \quad A_{n,m}^n = \left(1 + \frac{m-1}{n}\right)^n \cdot \Pi_m + \left(1 - \frac{1}{n}\right)^n \cdot (I_m - \Pi_m)$$

d'où finalement

$$\begin{aligned} \lim_{n \rightarrow +\infty} A_{n,m}^n &= e^{m-1} \cdot \Pi_m + e^{-1} \cdot (I_m - \Pi_m) = \frac{e^{m-1}}{m} \cdot J_m + e^{-1} \cdot \left(I_m - \frac{1}{m} \cdot J_m\right) \\ &= e^{-1} \cdot I_m + \frac{e^{m-1} - e^{-1}}{m} \cdot J_m \end{aligned}$$

✎ On peut vérifier numériquement.

```
def Am(m, n):
    q = 1/n
    return (1-q)*np.eye(m)+q*np.ones((m,m))
```

```
m = 4
```

```
for k in range(1, 5):
    n = 10**k
    M = Am(m,n)
    print(alg.matrix_power(M, n))
```

[ 1. ] Écrire une fonction `appliquer(f, M)` dont le premier argument est une application  $f : \mathbb{C} \rightarrow \mathbb{C}$  et le second argument une matrice diagonalisable  $M \in \mathfrak{M}_n(\mathbb{C})$ . Si la matrice  $M$  est égale à

$$P \operatorname{Diag}(a_1, \dots, a_n) P^{-1},$$

alors la fonction doit renvoyer la matrice

$$P \operatorname{Diag}(f(a_1), \dots, f(a_n)) P^{-1}.$$

[ 2. ] Dans cette question, on suppose que

$$f = [x \mapsto x^2] \quad \text{et que} \quad M = \frac{1}{10} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}.$$

Comparer le résultat de `appliquer(f, M)` et la matrice  $M^2$ .

[ 3.a. ] Écrire une fonction `S(N, M)` dont le premier argument est un entier  $N \in \mathbb{N}$  et le second une matrice carrée  $M$ , qui renvoie la matrice

$$S_N(M) = \sum_{k=0}^N \frac{(-1)^k}{(2k+1)!} M^{2k+1}.$$

[ 3.b. ] Tester avec la matrice  $M$  définie à la question précédente pour  $N = 9$ .

[ 3.c. ] Comparer avec le résultat de `appliquer(f, M)` pour une fonction  $f$  bien choisie.

[ 4. ] Soit  $M \in \mathfrak{M}_n(\mathbb{C})$ , une matrice diagonalisable.

Soient  $\varphi$  et  $\omega$ , les endomorphismes canoniquement associés aux matrices

$$M = P \operatorname{Diag}(a_1, \dots, a_n) P^{-1} \quad \text{et} \quad P \operatorname{Diag}(f(a_1), \dots, f(a_n)) P^{-1}.$$

[ 4.a. ] Démontrer que, pour toute valeur propre  $\lambda$  de  $M$ ,

$$\omega|_{E_\lambda(M)} = f(\lambda) \operatorname{Id}_{E_\lambda(M)}$$

où  $E_\lambda(M)$  désigne le sous-espace propre de  $\varphi$  associé à la valeur propre  $\lambda$ .

[ 4.b. ] On suppose que

$$M = Q \operatorname{Diag}(b_1, \dots, b_n) Q^{-1}.$$

Démontrer que

$$P \operatorname{Diag}(f(a_1), \dots, f(a_n)) P^{-1} = Q \operatorname{Diag}(f(b_1), \dots, f(b_n)) Q^{-1}.$$

[ 5.a. ] Démontrer que les matrices

$$M_k = \begin{pmatrix} 1/k & 1 \\ 0 & 0 \end{pmatrix} \quad \text{et} \quad N_k = \begin{pmatrix} -1/k & 1 \\ 0 & 0 \end{pmatrix}$$

sont diagonalisables.

[ 5.b. ] Soit  $f = [x \mapsto |x|]$ . Comparer  $f(M_k)$  et  $f(N_k)$ .

[ 1. ] La fonction `eig` du module `numpy.linalg` nous donne d'une part la liste des valeurs propres et d'autre part une matrice de vecteurs propres si bien que, si la matrice  $M$  est diagonalisable, on peut la diagonaliser sans peine.

```
M = 0.1*np.array([[1,2,3],[4,5,6],[7,8,9]])
spectre, P = alg.eig(M)
diag = np.diag(spectre)
```

☛ Pour vérifier ce calcul, il ne faut pas être naïf! L'exécution de la commande

```
print(P.dot(diag).dot(alg.inv(P))==M)
```

renvoie une matrice dont tous les coefficients sont égaux à False!

Les erreurs d'arrondi sont en cause, bien sûr. On contourne le problème en calculant la norme de la différence. On rappelle que la norme euclidienne canonique (ou **norme de Frobenius**) pour une matrice à coefficients réels est définie par  $\sqrt{\text{tr } M^T \cdot M}$ .

```
erreur = P.dot(diag).dot(alg.inv(P)) - M
print(np.sqrt(np.trace(erreur.T.dot(erreur))))
```

Il n'est pas mauvais de savoir que `alg.norm(erreur)` donne le même résultat avec moins d'efforts.

↳ Pour une matrice à coefficients complexes, il vaut mieux calculer la norme produit avec `abs(M).max()`.

• En supposant que la fonction `f` prise comme second argument s'applique à un tableau numpy, on peut donner un code très simple.

```
def appliquer(f, M):
    spectre, P = alg.eig(M)
    diag = np.diag(f(spectre))
    return P.dot(diag).dot(alg.inv(P))
```

[ 2. ]

```
fM = appliquer(lambda x: x**2, M)
M2 = M.dot(M) # plus rapide que alg.matrix_power(M, 2)
```

Comme on s'y attend, la matrice obtenue est égale à la matrice  $M^2$  aux erreurs d'arrondi près (la norme de Frobenius de la différence est de l'ordre de  $10^{-15}$ ).

[ 3.a. ] Pour limiter au minimum les calculs matriciels, il faut remarquer que le premier terme de la somme ( $k = 0$ ) est égal à  $M$  et que

$$\forall k \in \mathbb{N}, \quad \frac{(-1)^{k+1}}{(2[k+1]+1)!} M^{2(k+1)+1} = \frac{-1}{(2k+2)(2k+3)} M^2 \cdot \frac{(-1)^k}{(2k+1)!} M^{2k+1}.$$

Il s'agit donc de calculer la somme

$$\sum_{k=0}^N u_k = u_0 + \sum_{k=0}^{N-1} u_{k+1} \quad \text{où} \quad u_{k+1} = \frac{-1}{(2k+2)(2k+3)} M^2 \cdot u_k.$$

• Le code Python s'en déduit facilement, même s'il faut faire attention à bien détacher la variable locale `U` de l'argument `M`.

Le produit matriciel  $I_n \cdot M$  crée une nouvelle matrice, on aurait pu aussi bien procéder à une copie avec l'instruction `U = copy.deepcopy(M)`.

```
def S(N, M):
    U, M2 = np.eye(len(M)).dot(M), M.dot(M) # copy.deepcopy(M), M.dot(M)
    somme = U
    for k in range(N):
        U = -U.dot(M2)/((2*k+2)*(2*k+3))
        somme += U
    return somme
```

[ 3.b. ] On obtient une matrice... Il est intéressant de constater qu'on obtient la même matrice (aux erreurs d'arrondi près) pour  $N = 10$ ,  $N = 11$ ,  $N = 12$ ...

[ 3.c. ] Normalement, on a reconnu le développement limité de la fonction `sin`.

En "appliquant" la fonction `sin` à la matrice `M`, on trouve en effet une matrice très proche de la matrice calculée à la question précédente.

```
A = S(9, M)
B = appliquer(np.sin, M)
print(alg.norm(A-B))
```

La norme calculée est de l'ordre de  $10^{-16}$ , on peut donc considérer que les deux matrices sont bien égales.

[ 4. a. ] Par définition, un vecteur  $x$  appartient au sous-espace propre  $E_\lambda(M)$  si, et seulement si,  $\varphi(x) = \lambda x$ . Il s'agit de vérifier que, pour un tel vecteur  $x$ , on a  $u(x) = f(\lambda) \cdot x$ .

☞ Si  $F$  est un sous-espace stable par un endomorphisme  $u \in L(E)$ , la notation  $u_F$  désigne l'endomorphisme de  $F$  induit par restriction de  $u$ , c'est-à-dire l'application linéaire de  $F$  dans  $F$  définie par

$$\forall x \in F, \quad u_F(x) = u(x).$$

Si le sous-espace  $F$  n'est pas stable par  $u$  (si par exemple, comme ici, on n'a pas encore cherché à vérifier si  $F$  était stable par  $u$ ), alors la notation  $u|_F$  désigne l'application linéaire de  $F$  dans  $E$  définie par

$$\forall x \in F, \quad u|_F(x) = u(x).$$

La seule différence entre ces deux applications est donc l'espace d'arrivée; la première application linéaire est un endomorphisme :  $u_F \in L(F)$  (endomorphisme de  $F$ ) mais pas l'autre :  $u|_F \in L(E, F)$ .

☛ Comme la matrice  $P$  est inversible, c'est une matrice de passage : on notera  $\mathcal{B} = (\varepsilon_1, \dots, \varepsilon_n)$  la base de  $E$  telle que  $P$  soit la matrice de passage de la base canonique  $\mathcal{B}_{\text{can}}$  à la base  $\mathcal{B}$ .

$$P = \mathfrak{Mat}(\mathcal{B}_{\text{can}} \rightarrow \mathcal{B}) \quad P^{-1} = \mathfrak{Mat}(\mathcal{B} \rightarrow \mathcal{B}_{\text{can}})$$

Par hypothèse, la matrice  $M$  représente l'endomorphisme  $\varphi$  dans la base canonique de  $\mathbb{R}^n$  et, d'après la formule du changement de base, la matrice  $P^{-1}MP$  représente l'endomorphisme  $\varphi$  dans la base  $\mathcal{B}$ .

$$M = \mathfrak{Mat}_{\mathcal{B}_{\text{can}}}(\varphi) \quad P^{-1}MP = \mathfrak{Mat}_{\mathcal{B}}(\varphi)$$

☛ Par hypothèse, la matrice  $P^{-1}MP = \text{Diag}(a_1, \dots, a_n)$  est diagonale, donc les vecteurs  $\varepsilon_1, \dots, \varepsilon_n$  de la base  $\mathcal{B}$  (= les vecteurs représentés dans la base canonique par les colonnes de la matrice de passage  $P$ ) sont des vecteurs propres de  $\varphi$  et, plus précisément,

$$\forall 1 \leq k \leq n, \quad \varphi(\varepsilon_k) = a_k \cdot \varepsilon_k.$$

Le sous-espace propre  $E_\lambda(M)$  est donc engendré par les vecteurs  $\varepsilon_k$  tels que  $a_k = \lambda$  : si  $x \in E_\lambda(M)$ , alors il existe des scalaires  $(x_k)_{1 \leq k \leq n}$  tels que

$$x = \sum_{k=1}^n x_k \cdot \varepsilon_k = \sum_{\substack{1 \leq k \leq n : \\ a_k = \lambda}} x_k \cdot \varepsilon_k.$$

☛ Par définition de  $u$ , la base  $\mathcal{B}$  est aussi une base de vecteurs propres pour  $u$  :

$$\forall 1 \leq k \leq n, \quad u(\varepsilon_k) = f(a_k) \cdot \varepsilon_k$$

et par conséquent

$$u(x) = \sum_{\substack{1 \leq k \leq n : \\ a_k = \lambda}} x_k \cdot u(\varepsilon_k) = \sum_{\substack{1 \leq k \leq n : \\ a_k = \lambda}} x_k f(a_k) \cdot \varepsilon_k = f(\lambda) \sum_{\substack{1 \leq k \leq n : \\ a_k = \lambda}} x_k \cdot \varepsilon_k = f(\lambda) \cdot x.$$

[ 4. b. ] On suppose cette fois qu'il existe une autre base  $\mathcal{C} = (\eta_k)_{1 \leq k \leq n}$  constituée de vecteurs propres pour  $\varphi$  :

$$Q = \mathfrak{Mat}(\mathcal{B}_{\text{can}} \rightarrow \mathcal{C}) \quad \text{Diag}(b_1, \dots, b_n) = Q^{-1}MQ = \mathfrak{Mat}_{\mathcal{C}}(\varphi).$$

On a donc

$$\text{Diag}(b_1, \dots, b_n) = Q^{-1}(P \times \mathfrak{Mat}_{\mathcal{B}}(\varphi) \times P^{-1})Q = (P^{-1}Q)^{-1} \text{Diag}(a_1, \dots, a_n)(P^{-1}Q). \quad (*)$$

☛ Les deux familles  $(a_k)_{1 \leq k \leq n}$  et  $(b_k)_{1 \leq k \leq n}$  représentent les valeurs propres de  $\varphi$  comptées avec multiplicité, elles se déduisent donc l'une de l'autre par une permutation : il existe une permutation  $\sigma \in \mathfrak{S}_n$  telle que

$$\forall 1 \leq k \leq n, \quad b_k = a_{\sigma(k)}.$$

• Comme le spectre de  $\varphi$  est un ensemble fini de cardinal inférieur à  $n$ , il existe un polynôme  $L \in \mathbb{C}[X]$  tel que

$$\forall 1 \leq k \leq n, \quad L(a_k) = f(a_k). \quad (\dagger)$$

• En général, la théorie des polynômes interpolateurs de Lagrange impose de choisir des abscisses deux à deux distinctes.

Ici, on sait que : si  $a_k = a_\ell$ , alors  $f(a_k) = f(a_\ell)$  et cette contrainte fait qu'on n'a pas besoin de supposer que les  $a_k$  soient deux à deux distincts.

Par conséquent,

$$\forall 1 \leq k \leq n, \quad f(b_k) = f(a_{\sigma(k)}) = L(a_{\sigma(k)}) = L(b_k). \quad (\ddagger)$$

On déduit alors de (\*) que

$$L(\text{Diag}(b_1, \dots, b_n)) = (P^{-1}Q)^{-1} \times L(\text{Diag}(a_1, \dots, a_n)) \times (P^{-1}Q).$$

• La formule de changement de base  $[A \mapsto R^{-1}AR]$  est un morphisme d'algèbres, donc

$$L(R^{-1}AR) = R^{-1}L(A)R,$$

quel que soit le polynôme  $L$ .

• Comme  $L$  est un polynôme, d'après (\dagger),

$$L(\text{Diag}(a_1, \dots, a_n)) = \text{Diag}(L(a_1), \dots, L(a_n)) = \text{Diag}(f(a_1), \dots, f(a_n))$$

et, de même, d'après (\ddagger),

$$L(\text{Diag}(b_1, \dots, b_n)) = \text{Diag}(f(b_1), \dots, f(b_n)).$$

On a donc démontré que

$$Q \text{Diag}(f(b_1), \dots, f(b_n)) Q^{-1} = P \text{Diag}(f(a_1), \dots, f(a_n)) P^{-1}.$$

• **Variante** (plus courte, pas nécessairement plus simple)

Notons  $v$ , l'endomorphisme canoniquement associé à la matrice  $Q \text{Diag}(f(b_1), \dots, f(b_n)) Q^{-1}$ . D'après la question précédente,

$$\forall \lambda \in \text{Sp}(M), \quad v|_{E_\lambda(M)} = f(\lambda) \text{Id}_{E_\lambda(M)} = u|_{E_\lambda(M)}.$$

Comme  $M$  est supposée diagonalisable,

$$E = \bigoplus_{\lambda \in \text{Sp}(M)} E_\lambda(M)$$

et on vient de constater que les endomorphismes  $u$  et  $v$  coïncident sur chacun des sous-espaces propres. Par conséquent,  $u = v$  et

$$Q \text{Diag}(f(b_1), \dots, f(b_n)) Q^{-1} = \mathfrak{Mat}_{\mathcal{B}_{\text{can}}}(v) = \mathfrak{Mat}_{\mathcal{B}_{\text{can}}}(u) = P \text{Diag}(f(a_1), \dots, f(a_n)) P^{-1}.$$

• Cette propriété rend légitime la notation  $f(M)$  pour désigner la matrice  $P \text{Diag}(f(a_1), \dots, f(a_n)) P^{-1}$  puisqu'on obtient la même matrice, quelle que soit la manière de diagonaliser la matrice  $M$ .

• Le cours se borne à étudier les matrices  $f(M)$  quand  $f$  est une application polynomiale. On n'est donc pas obligé de se restreindre aux applications polynomiales.

Cela dit, comme le spectre d'une matrice est un ensemble fini et que toute application d'une partie finie de  $\mathbb{C}$  à valeurs dans  $\mathbb{C}$  est en fait une application polynomiale (Théorie de l'interpolation de Lagrange), ce n'est pas une si grande surprise.

[ 5.a. ] Les matrices  $M_k$  et  $N_k$  sont triangulaires, donc leurs valeurs propres sont leurs coefficients diagonaux. Ce sont donc des matrices de  $\mathfrak{M}_2(\mathbb{R})$  qui ont deux valeurs propres distinctes, elles sont diagonalisables.

[ 5.b. ] Si  $P^{-1}M_kP$  et  $P^{-1}N_kP$  sont diagonales (les matrices  $M_k$  et  $N_k$  sont alors des **matrices co-diagonalisables**), alors elles commutent et on en déduit facilement que  $M_k$  et  $N_k$  commutent.

↳ *La réciproque est vraie, mais moins évidente.*

Mais

$$\forall k \in \mathbb{N}^*, \quad M_k N_k = \frac{1}{k^2} \begin{pmatrix} -1 & k \\ 0 & 0 \end{pmatrix} \neq \frac{1}{k^2} \begin{pmatrix} -1 & -k \\ 0 & 0 \end{pmatrix} = N_k M_k$$

donc il n'existe pas de matrice inversible  $P$  telle que  $P^{-1}M_kP$  et  $P^{-1}N_kP$  soient toutes les deux diagonales.

• Les sous-espaces propres de  $M_k$  et de  $N_k$  sont à peu près aussi évidents que leurs spectres. Avec

$$P = \begin{pmatrix} 1 & -k \\ 0 & 1 \end{pmatrix} \quad \text{et} \quad Q = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix},$$

on a

$$P^{-1}M_kP = \text{Diag}(1/k, 0) \quad \text{et} \quad Q^{-1}N_kQ = \text{Diag}(-1/k, 0).$$

↳ *Ce n'est pas prudent d'omettre l'indice  $k$  sur les matrices de passage, mais c'est bien pratique.*

• Par conséquent,

$$P^{-1}f(M_k)P = \text{Diag}(1/k, 0) = P^{-1}M_kP \quad \text{et} \quad Q^{-1}f(N_k)Q = \text{Diag}(1/k, 0) = -Q^{-1}N_kQ.$$

On en déduit que

$$\forall k \geq 2, \quad f(M_k) = M_k \quad \text{et que} \quad f(N_k) = -N_k \neq M_k.$$

↳ *On voit sur cet exemple que deux matrices diagonalisables peuvent avoir des images par  $f$  différentes alors que leurs spectres ont même image par  $f$ .*

↳ *On peut également vérifier que : si  $M$  est semblable à  $\text{Diag}(a, b)$  avec  $a < 0 < b$ , alors  $f(M) \neq M$  et  $f(M) \neq -M$ .*

On considère les matrices

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & & & & \\ 0 & & & & \\ \vdots & & & & \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix} \in \mathfrak{M}_n(\mathbb{R}) \quad \text{et} \quad \forall 1 \leq \ell \leq n, \quad X_\ell = \begin{pmatrix} \sin \frac{\ell\pi}{n+1} \\ \vdots \\ \sin \frac{\ell n\pi}{n+1} \end{pmatrix} \in \mathfrak{M}_{n,1}(\mathbb{R}).$$

On note  $P \in \mathfrak{M}_n(\mathbb{R})$ , la matrice dont les colonnes sont  $X_1, \dots, X_n$ . Enfin, pour  $p$  et  $q$  dans  $\mathbb{Z}$ , on pose

$$T_p = \sum_{k=1}^n \cos \frac{pk\pi}{n+1} \quad \text{et} \quad S_{p,q} = \sum_{k=1}^n \sin \frac{pk\pi}{n+1} \sin \frac{qk\pi}{n+1}.$$

- [ 1. ] Calculer  $T_1$ .
- [ 2. ] La matrice  $A$  est-elle diagonalisable? Que dire de ses sous-espaces propres?
- [ 3. ] Écrire une fonction  $A(n)$  qui renvoie la matrice  $A$  de taille  $n$ .
- [ 4. ] La fonction suivante renvoie la matrice  $P$  de taille  $n$ .

```
def P(n):
    M = np.pi*np.ones((n, n), dtype=int)
    for i in range(n):
        M[i,:] *= (i+1)
    for j in range(n):
        M[:,j] *= (j+1)
    return np.sin(M/(n+1))
```

Donner un script qui calcule  $P^{-1}AP$ .

- [ 5. ] Proposer une conjecture sur le cardinal du spectre de  $A$  ainsi qu'une conjecture sur la matrice  $P$ .
- [ 6. ] Calculer la somme  $T_p$  pour tout  $p \in \mathbb{Z}$ .
- [ 7. ] Calculer la somme  $S_{p,q}$  pour tout  $1 \leq p, q \leq n$ .
- [ 8. ] Démontrer les conjectures faites sur le spectre de  $A$  et sur la matrice  $P$ .

- [ 1. ] Voir plus loin le calcul de  $T_p$ .
- [ 2. ] La matrice  $A$  est symétrique réelle, donc elle est diagonalisable et ses sous-espaces propres sont deux à deux orthogonaux.
- [ 3. ] Code sans difficulté.

```
def A(n):
    M = np.zeros((n, n), dtype=int)
    for i in range(1, n):
        M[i,i-1] = 1
        M[i-1,i] = 1
    return M
```

Les pythoniens fanatiques auront vu que la matrice  $A$  est la somme de deux matrices qui ressemblent à l'identité ( $\text{np.eye}(n)$  – oui, c'est un calembour).

```
def A_var(n):
    J = np.eye(n,k=1,dtype=int)+np.eye(n,k=-1,dtype=int)
    return J
```

- [ 4. ] Le produit matriciel s'appelle `dot` et c'est aussi bien une fonction du module `numpy` qu'une méthode de l'objet matrice.

```
np.dot(A, B)
A.dot(B)
```

En revanche, le calcul de l'inverse s'effectue grâce à une fonction du sous-module `numpy.linalg`.

**def** conjuguer(n) :

Q = P(n)

**return** np.round(alg.inv(Q).dot(A(n)).dot(Q)), 2)

• On a choisi d'arrondir le résultat au centième pour qu'il soit plus lisible.

[ 5. ] Après quelques essais, on constate que  $P^{-1}AP$  est diagonale, que les valeurs propres de  $A$  varient entre  $-2$  et  $2$  et que  $A$  possède  $n$  valeurs propres distinctes.

• On a constaté sur les exemples que les colonnes de  $P$  étaient des vecteurs propres de  $A$ . S'il y a effectivement  $n$  valeurs propres deux à deux distinctes, alors les sous-espaces propres sont des droites vectorielles et ces droites sont deux à deux orthogonales. Les colonnes de  $P$  sont donc deux à deux orthogonales.

• Il est alors tentant de vérifier si la matrice  $P$  est orthogonale en calculant  $P.P^T$ . Après quelques essais, on devine plutôt que

$$\forall n \in \mathbb{N}^*, \quad P_n.P_n^T = \frac{n+1}{2} \cdot I_n.$$

[ 6. ] Pour alléger l'écriture des calculs, nous allons poser

$$\theta = \frac{\pi}{2(n+1)} \quad \text{et} \quad \omega = \exp \frac{i\pi}{n+1} = \exp(2i\theta)$$

si bien que  $\omega^{2(n+1)} = 1$ .

• Puisqu'il s'agit d'alléger les calculs (et que l'entier  $n$  est fixé), on s'autorise à écrire  $\theta$  et  $\omega$  au lieu de  $\theta_n$  et  $\omega_n$ .

• Pour tout  $p \in \mathbb{Z}$ ,

$$T_p = \sum_{k=1}^n \cos \frac{kp\pi}{n+1} = \Re \sum_{k=1}^n (\omega^p)^k.$$

On reconnaît une somme géométrique de raison  $\omega^p$ .

• Si  $p$  est un multiple de  $2(n+1)$ , alors  $\omega^p = 1$  et  $T_p = n$ .

• Si  $p$  n'est pas un multiple de  $2(n+1)$ , alors  $\omega^p \neq 1$  et

$$\begin{aligned} \sum_{k=1}^n (\omega^p)^k &= \omega^p \frac{\omega^{p(n+1)} - 1}{\omega^p - 1} = e^{2ip\theta} \frac{e^{i(n+1)p\theta} - 1}{e^{ip\theta} - 1} = e^{i(n+1)p\theta} \frac{\sin np\theta}{\sin p\theta} \\ &= i^p \frac{\sin np\theta}{\sin p\theta} \end{aligned}$$

puisque  $(n+1)\theta = \pi/2$ .

Si  $p$  est impair, alors la somme est imaginaire pure, donc  $T_p = 0$ .

Si  $p = 2k$  est pair, alors

$$T_{2k} = (-1)^k \frac{\sin 2nk\theta}{\sin 2k\theta}.$$

Mais  $2nk\theta + 2k\theta = 2(n+1)k\theta = k\pi$ , donc

$$\sin 2nk\theta = \sin(k\pi - 2k\theta) = (-1)^{k+1} \sin 2k\theta$$

et finalement  $T_{2k} = -1$ .

• On distingue donc trois cas :

— si  $p$  est impair, alors  $T_p = 0$ ;

— si  $p$  est un multiple de  $2(n+1)$ , alors  $T_p = n$ ;

— si  $p$  est pair sans être un multiple de  $2(n+1)$ , alors  $T_p = -1$ .

[ 7. ] Quels que soient  $p$  et  $q$  dans  $\mathbb{Z}$ ,

$$\begin{aligned} S_{p,q} &= \sum_{k=1}^n \sin 2kp\theta \sin 2kq\theta = \frac{1}{2} \sum_{k=1}^n (\cos 2k(p-q)\theta - \cos 2k(p+q)\theta) \\ &= \frac{T_{p-q} - T_{p+q}}{2}. \end{aligned}$$

• Comme  $1 \leq p, q \leq n$ , alors  $2 \leq p+q \leq 2n$  et  $-n < p-q < n$ . Par conséquent, l'entier  $p+q$  n'est jamais un multiple de  $2(n+1)$  et l'entier  $p-q$  est un multiple de  $2(n+1)$  si, et seulement si,  $p = q$ .

☞ Les entiers  $p - q$  et  $p + q$  ont même parité.

☛ Pour  $p \neq q$ , alors  $S_{p,q} = (0-0)/2 = 0$  si  $(p-q)$  et  $(p+q)$  sont impairs et  $S_{p,q} = (-1+1)/2 = 0$  si  $(p-q)$  et  $(p+q)$  sont pairs.

☛ Pour  $p = q$ , alors  $p + q$  est pair et  $S_{p,p} = [n - (-1)]/2 = (n + 1)/2$ .

☛ Finalement,

$$\forall 1 \leq p \leq n, \quad S_{p,p} = \frac{n+1}{2} \quad \text{et} \quad \forall 1 \leq p < q \leq n, \quad S_{p,q} = S_{q,p} = 0.$$

[ 8. ] Les égalités précédentes signifient que, pour le produit scalaire canonique sur  $\mathfrak{M}_{n,1}(\mathbb{R})$ , les colonnes  $X_p$  de la matrice  $P$  sont deux à deux orthogonales :

$$\forall 1 \leq p < q \leq n, \quad \langle X_p | X_q \rangle = X_p^T \cdot X_q = S_{p,q} = 0$$

et que leurs normes sont toutes égales :

$$\forall 1 \leq p \leq n, \quad \|X_p\| = \sqrt{X_p^T \cdot X_p} = \sqrt{\frac{n+1}{2}}.$$

Par conséquent,

$$P^T \cdot P = \frac{n+1}{2} \quad \text{ou, si on préfère,} \quad \sqrt{\frac{2}{n+1}} P \in O_n(\mathbb{R}).$$

☛ En linéarisant à nouveau, pour tout  $1 \leq \ell \leq n$ ,

$$AX_\ell = \begin{pmatrix} \sin 4\ell\theta \\ \vdots \\ \sin 2(k-1)\ell\theta + \sin 2(k+1)\ell\theta \\ \vdots \\ \sin 2n\ell\theta = \sin 2(n-1)\ell\theta + \sin 2(n+1)\ell\theta \end{pmatrix} = 2 \cos 2\ell\theta \cdot X_\ell$$

car  $\sin 2(n+1)\ell\theta = 0$  pour tout entier  $\ell$ .

On a déjà démontré que  $X_\ell \neq 0$  (puisque sa norme n'est pas nulle), donc on a démontré que la colonne  $X_\ell$  est bien un vecteur propre de  $A$  associé à la valeur propre  $2 \cos 2\ell\theta$ .

☛ Ainsi, on a trouvé une base orthogonale  $(X_\ell)_{1 \leq \ell \leq n}$  de vecteurs propres et les valeurs propres associées

$$2 \cos \frac{\pi \ell}{n+1} \quad \text{pour } 1 \leq \ell \leq n$$

sont deux à deux distinctes car  $0 < 2\ell\theta < \pi$  et  $\cos$  est injective sur  $[0, \pi]$ .

Pour tout entier  $n \geq 1$ , la matrice  $A$  est donc diagonalisable et

$$P^{-1}AP = \text{Diag}\left(2 \cos \frac{\pi}{n+1}, 2 \cos \frac{2\pi}{n+1}, \dots, 2 \cos \frac{n\pi}{n+1}\right).$$

☞ Ici, les vecteurs propres sont parachutés par l'énoncé, mais on peut les trouver avec méthode (et de la culture).

☛ Tout d'abord, le lemme d'Hadamard sur les matrices à diagonale fortement dominante montre que  $A - \lambda I_n$  est inversible pour tout nombre complexe  $\lambda$  tel que  $|\lambda| > 2$ .

☛ Ensuite, comme la matrice  $A$  est symétrique réelle, on sait que ses valeurs propres appartiennent au segment  $[-2, 2]$ , donc chaque valeur propre de  $A$  peut s'écrire sous la forme  $\lambda = 2 \cos \alpha$  avec  $0 \leq \alpha \leq \pi$ .

☛ Enfin, on peut remarquer que l'égalité vectoriel  $AX = \lambda X$  avec  $X = (x_1, \dots, x_n) \neq 0$  revient à trouver une famille

$$Y = (y_0, y_1, \dots, y_n, y_{n+1}) = (0, x_1, \dots, x_n, 0) \in \mathbb{R}^{n+2} \setminus \{0\}$$

telle que

$$\forall 1 \leq k \leq n, \quad y_{k-1} + y_{k+1} = \lambda y_k = 2 \cos \alpha y_k.$$

On sait exprimer les familles qui vérifient une telle relation de récurrence linéaire d'ordre deux et on en déduit l'expression des vecteurs propres de  $A$ .

Pour  $n \in \mathbb{N}^*$ , on pose

$$\omega_n = \exp \frac{2i\pi}{n}, \quad W_n = (\omega_n^{k\ell})_{0 \leq k, \ell \leq n-1} \in \mathfrak{M}_n(\mathbb{C}) \quad \text{et} \quad V_n = \frac{1}{\sqrt{n}} W_n.$$

- [ 1. ] Calculer  $V_n^2$ ,  $V_n \overline{V_n}$  et  $V_n^4$ .  
 [ 2. ] Calculer  $\det W_n$  puis  $\det V_n$ .  
 [ 3. ] Écrire une fonction  $V(n)$  qui renvoie la matrice  $V_n$ . Tester pour  $n = 4$ .  
 [ 4. ] Écrire une fonction  $\text{Spectre}(n)$  qui renvoie une valeur approchée du spectre de  $V_n$ . Conjecturer le nombre de valeurs propres de  $V_n$ .  
 [ 5. ] Démontrer cette conjecture.

[ 1. ] Notons  $c_{k,\ell}$ , les coefficients de  $W_n^2$  (pour  $0 \leq k, \ell < n$ ). D'après la formule du produit matriciel,

$$c_{k,\ell} = \sum_{0 \leq m < n} \omega_n^{km} \omega_n^{m\ell} = \sum_{m=0}^{n-1} (\omega_n^{k+\ell})^m.$$

Si  $k+\ell$  est un multiple de  $n$ , alors  $\omega_n^{k+\ell} = 1$  et  $c_{k,\ell} = n$ . Mais si  $k+\ell$  n'est pas un multiple de  $n$ , alors  $\omega_n^{k+\ell}$  est une racine  $n$ -ième de l'unité distincte de 1 et  $c_{k,\ell} = 0$ .

• Pour  $0 \leq k, \ell < n$ , on a  $0 \leq k+\ell < 2n$ , donc la somme  $k+\ell$  est un multiple de  $n$  si, et seulement si, elle est nulle (pour  $k = \ell = 0$  seulement) ou si elle est égale à  $n$  (pour  $1 \leq k < n$  et  $\ell = n - k$ ). On en déduit que

$$W_n^2 = n \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{pmatrix} \quad \text{et donc que} \quad W_n^4 = n^2 \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{pmatrix} = n^2 I_n.$$

• Notons  $d_{k,\ell}$ , les coefficients de  $W_n \overline{W_n}$  (pour  $0 \leq k, \ell < n$ ). De même,

$$d_{k,\ell} = \sum_{m=0}^{n-1} (\omega_n^{k-\ell})^m$$

donc  $d_{k,\ell} = n$  si  $k-\ell$  est un multiple de  $n$  et  $d_{k,\ell} = 0$  sinon.

Pour  $0 \leq k, \ell < n$ , on a  $-n < k-\ell < n$ , donc  $k-\ell$  est un multiple de  $n$  si, et seulement si,  $k = \ell$ . On en déduit que  $W_n \overline{W_n} = n I_n$ .

• Finalement,

$$V_n^2 = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{pmatrix}, \quad V_n \overline{V_n} = V_n^4 = I_n.$$

[ 2. ] Comme  $\det \overline{W_n} = \overline{\det W_n}$  et que  $W_n \overline{W_n} = n I_n$ , alors  $|\det W_n|^2 = n^n$  et donc  $|\det W_n| = n^{n/2}$ .

• La matrice  $W_n$  est aussi la matrice de Vandermonde associée aux nombres complexes  $a^{2k}$  pour  $0 \leq k < n$  où  $a = e^{i\pi/n}$ . Par conséquent,

$$\det W_n = \prod_{0 \leq k < \ell < n} (a^{2\ell} - a^{2k}) = \prod_{0 \leq k < \ell < n} a^{k+\ell} (a^{(\ell-k)} - a^{-(\ell-k)}) = \prod_{0 \leq k < \ell < n} 2i a^{k+\ell} \sin \frac{(\ell-k)\pi}{n}.$$

Pour  $0 \leq k < \ell < n$ , on a  $0 < \ell - k < n$ , donc chaque sinus est strictement positif et le produit compte  $\binom{n}{2}$  facteurs. Par conséquent,

$$\det W_n = (2i)^{\binom{n}{2}} \prod_{0 \leq k < \ell < n} \sin \frac{(\ell-k)\pi}{n} \times \prod_{\ell=1}^{n-1} \prod_{k=0}^{\ell-1} a^{k+\ell}$$

avec

$$\prod_{\ell=1}^{n-1} \prod_{k=0}^{\ell-1} a^{k+\ell} = \prod_{\ell=1}^{n-1} a^{\ell^2} \cdot a^{(\ell-1)\ell/2} = a^N \quad \text{où} \quad N = \sum_{\ell=1}^{n-1} \frac{3\ell^2 - \ell}{2} = \frac{n(n-1)^2}{2}.$$

↳ On doit savoir que

$$\sum_{\ell=1}^{n-1} \ell = \frac{(n-1)n}{2} \quad \text{et que} \quad \sum_{\ell=1}^{n-1} \ell^2 = \frac{(n-1)(2n-1)n}{6}.$$

• Finalement, comme  $a = e^{i\pi/n}$ ,

$$\det W_n = 2^{\binom{n}{2}} \prod_{0 \leq k < \ell < n} \sin \frac{(\ell - k)\pi}{n} \times i^{\binom{n}{2}} \times i^{(n-1)^2} = |\det W_n| \cdot i^{(n-1)(3n-2)/2}.$$

↳ En dépit des dénominateurs, tous les exposants qui précèdent sont des entiers — heureusement !

Et comme  $|\det W_n|$  est connu, on en déduit que

$$\det W_n = n^{n/2} \cdot i^{(n-1)(3n-2)/2} \quad \text{puis que} \quad \det V_n = \frac{\det W_n}{(\sqrt{n})^n} = i^{(n-1)(3n-2)/2}.$$

[ 3. ] On peut définir la matrice  $W_n$  en fatiguant plus la machine qu'on ne se fatigue les neurones.

```
def W(n):
    w = np.exp(2j*np.pi/n)
    M = np.zeros((n, n), dtype= complex)
    for k in range(n):
        for ell in range(n):
            M[k, ell] = w**(k*ell)
    return M
```

• On peut préférer une version plus économique, au prix d'un code un peu plus élaboré.

```
def W(n):
    w = np.exp(2j*np.pi/n)
    M = np.zeros((n, n), dtype= complex)
    wk = 1
    for k in range(n):
        wk1 = 1
        for ell in range(n):
            M[k, ell] = wk1
            wk1 *= wk
        wk *= w
    return M
```

• La matrice  $W_n$  étant connue, il est facile d'en déduire  $V_n$ .

```
def det_V(n):
    return np.round(alg.det(W(n)/np.sqrt(n)), 2)
```

On reconnaît alors

$$W_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad W_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & j & j^2 \\ 1 & j^2 & j \end{pmatrix}, \quad W_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}.$$

[ 4. ] Pour obtenir une valeur approchée des valeurs propres de  $V_n$ , il suffit de connaître la bonne fonction Python.

```
def Spectre(n):
    V = W(n)/np.sqrt(n)
    return np.round(alg.eigvals(V), 2)
```

• On sait que  $V_n^4 = I_n$ , donc les valeurs propres de  $V_n$  sont des racines du polynôme annulateur  $X^4 - 1$ .

Les essais numériques montrent que, pour  $n \geq 5$ , les quatre nombres complexes  $\pm 1$  et  $\pm i$  sont bien des valeurs propres de  $V_n$ .

↳ *En revanche, pour  $n \leq 3$ , la matrice  $V_n$  possède au plus  $n$  valeurs propres distinctes!*

[ 5. ] Notons  $e_k$ ,  $0 \leq k < n$ , les vecteurs de la base canonique de  $\mathbb{C}^n$ . On voit sur la matrice  $W_n$  qu'aucun vecteur de la base canonique n'est un vecteur propre de  $W_n$ . Par conséquent, chaque couple  $(e_k, W_n e_k)$  est une famille libre.

• On a démontré que

$$W_n^2 e_0 = n \cdot e_0 \quad \text{et que} \quad \forall 1 \leq k < n, \quad W_n^2 e_k = n \cdot e_{n-k} \quad \text{d'où} \quad W_n^2 e_{n-k} = n \cdot e_k. \quad (\dagger)$$

On en déduit que

$$\forall 1 \leq k < n \quad W_n^2 (e_k + e_{n-k}) = n \cdot (e_k + e_{n-k}) \quad \text{et que} \quad W_n^2 (e_k - e_{n-k}) = -n \cdot (e_k - e_{n-k}). \quad (\star)$$

• Pour tout entier  $0 \leq k < n$ , on pose  $X_k = W_n e_k$  (= la  $k$ -ième colonne de la matrice  $W_n$ ) et remarquons que la colonne  $X_{n-k} = W_n e_{n-k}$  est la conjuguée de la colonne  $X_k$ . On a alors

$$W_n (e_k + e_{n-k}) = X_k + \overline{X_k} = 2 \begin{pmatrix} 1 \\ \cos \frac{2k\pi}{n} \\ \vdots \\ \cos \frac{2(n-1)k\pi}{n} \end{pmatrix} \quad (\ddagger)$$

et

$$W_n (e_k - e_{n-k}) = X_k - \overline{X_k} = 2i \begin{pmatrix} 0 \\ \sin \frac{2k\pi}{n} \\ \vdots \\ \sin \frac{2(n-1)k\pi}{n} \end{pmatrix}. \quad (\ddagger)$$

• Les relations  $(\star)$  montrent que les deux sous-espaces vectoriels

$$P_k = \text{Vect}(e_k + e_{n-k}, X_k + X_{n-k}) \quad \text{et} \quad Q_k = \text{Vect}(e_k - e_{n-k}, X_k - X_{n-k})$$

sont stables par  $W_n$ , quel que soit l'entier  $1 \leq k < n/2$ .

↳ *C'est parce que  $e_k + e_{n-k} = e_{n-k} + e_{n-(n-k)}$  qu'on peut (ou qu'on doit) se limiter aux indices  $k$  tels que  $k < n - k$ , c'est-à-dire tels que  $k < n/2$ .*

De même, d'après  $(\ddagger)$ , le plan

$$P_0 = \text{Vect}(e_0, X_0)$$

est stable par  $W_n$  et, si la dimension  $n = 2q$  est paire, alors  $e_{n-q} = e_q$  et le plan

$$P_q = \text{Vect}(e_q, X_q)$$

est stable par  $W_n$ .

• Soit  $\varphi$ , l'endomorphisme de  $\mathbb{C}^n$  canoniquement associé à  $W_n$ .

Nous allons maintenant représenter matriciellement l'endomorphisme induit par restriction de  $\varphi$  aux différents plans stables que nous venons de rencontrer.

• D'après  $(\ddagger)$ , la matrice relative à  $(e_0, X_0)$  de l'endomorphisme induit par restriction à  $P_0$  et, si  $n = 2p$  est pair, la matrice relative à  $(e_p, X_p)$  de l'endomorphisme induit par restriction à  $Q_0$  sont toutes les deux égales à la matrice

$$\begin{pmatrix} 0 & n \\ 1 & 0 \end{pmatrix}.$$

Les valeurs propres de cette matrice sont  $\pm\sqrt{n}$ , donc les deux réels  $\pm\sqrt{n}$  sont des valeurs propres de  $W_n$ .

↳ *La matrice  $W_2$  admet  $\pm\sqrt{2}$  pour valeurs propres.*

• La matrice  $W_3$  admet  $\pm\sqrt{3}$  pour valeurs propres et comme  $\text{tr } W_3 = 1 + 2j = j - j^2 = i\sqrt{3}$ , elle admet  $i\sqrt{3}$  comme troisième et dernière valeur propre.

• Si les vecteurs  $e_k - e_{n-k}$  et  $X_k - X_{n-k}$  ne sont pas colinéaires, alors le sous-espace stable  $Q_k$  est un plan et le couple  $(e_k - e_{n-k}, X_k - X_{n-k})$  est une base de ce plan.

Dans cette base, l'endomorphisme induit par restriction de  $\varphi$  au plan  $Q_k$  est représenté par la matrice

$$\begin{pmatrix} 0 & -n \\ 1 & 0 \end{pmatrix}$$

d'après (\*). Les valeurs propres de cette matrice :  $\pm i\sqrt{n}$  sont alors des valeurs propres de  $W_n$ .

• Pour  $n = 2$ , le vecteur  $e_k - e_{n-k} = e_1 - e_1$  est nul et ce cas est sans objet.

• Pour  $n = 3$ , les vecteurs  $e_k - e_{n-k} = e_1 - e_2 = (0, 1, -1)$  et  $X_k - X_{n-k} = (j - j^2) \cdot (0, 1, -1)$  sont proportionnels et le sous-espace stable  $Q_1$  est une droite, ce qui prouve que le vecteur  $e_1 - e_2$  qui dirige cette droite est un vecteur propre. Le calcul qui précède montre que la valeur propre associée à ce vecteur est égale à  $(j - j^2) = i\sqrt{3}$ , ainsi qu'on l'a vu plus haut.

• Pour  $n = 4 = 2 \cdot 2$ , on a explicité quatre sous-espaces stables :

$$P_0 = \text{Vect}((1, 0, 0, 0), (1, 1, 1, 1)), \quad P_1 = \text{Vect}((0, 1, 0, 1), (2, 0, -2, 0))$$

$$P_2 = \text{Vect}((0, 0, 1, 0), (1, -1, 1, -1)), \quad Q_1 = \text{Vect}((0, 1, 0, -1), (0, 2i, 0, -2i)) = \text{Vect}((0, 1, 0, -1)).$$

Le vecteur  $(0, 1, 0, -1)$  qui dirige  $Q_1$  est un vecteur propre de  $W_4$  associé à la valeur propre  $2i$ . Dans les bases "naturelles" des trois plans  $P_0, P_1$  et  $P_2$ , la matrice de l'endomorphisme induit par restriction de  $\varphi$  est la matrice

$$\begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix},$$

dont les valeurs propres sont  $\pm 2$ .

Si  $W_4$  admettait une quatrième valeur propre, ce serait nécessairement  $-2i$  et les quatre valeurs propres seraient des valeurs propres simples. Dans ces conditions, la trace de  $W_4$  serait nulle, ce qui est faux.

La matrice  $W_4$  n'a donc que trois valeurs propres :  $2i$  et  $\pm 2$ . Comme  $\text{tr } W_4 = 2 + 2i$ , on en déduit que  $2$  est une valeur propre double et que  $2i$  et  $-2$  sont des valeurs propres simples de  $W_4$ .

• Considérons maintenant un entier  $n \geq 5$ .

D'après (§), les vecteurs  $e_1 + e_{n-1}$  et  $X_1 + X_{n-1}$  ne sont pas proportionnels (comparer les coordonnées d'indice 0) et dans cette base "naturelle" du plan stable  $P_1$ , la matrice de l'endomorphisme induit par restriction de  $\varphi$  est

$$\begin{pmatrix} 0 & n \\ 1 & 0 \end{pmatrix}$$

donc  $\pm\sqrt{n}$  sont des valeurs propres de  $W_n$ .

De même, les vecteurs

$$e_1 - e_{n-1} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix} \quad \text{et} \quad X_1 - X_{n-1} = 2i \begin{pmatrix} 0 \\ \sin \frac{2\pi}{n} \\ \sin \frac{4\pi}{n} \\ \vdots \\ -\sin \frac{4\pi}{n} \\ -\sin \frac{2\pi}{n} \end{pmatrix}$$

ne sont pas proportionnels ( $\sin \frac{4\pi}{n} > 0$  car  $n \geq 5$ ). Donc le sous-espace stable  $Q_1$  est un plan et, d'après (\*), la matrice de l'endomorphisme induit par restriction de  $\varphi$  au plan  $Q_1$  relative à la base "naturelle" de ce plan est

$$\begin{pmatrix} 0 & -n \\ 1 & 0 \end{pmatrix}$$

donc  $\pm i\sqrt{n}$  sont des valeurs propres de  $W_n$ .

• On a ainsi démontré que, pour tout entier  $n \geq 5$ , la matrice  $V_n$  admettait exactement quatre valeurs propres :  $\pm 1$  et  $\pm i$ .

[ 1. ] Soient  $P \in \mathbb{Z}[X]$  et  $k \in \mathbb{N}$ . Démontrer qu'il existe un, et un seul, polynôme  $Q \in \mathbb{Z}[X]$  tel que

$$P(X + 10^k) - P = 10^k Q.$$

Que dire de  $Q$  si les coefficients de  $P$  sont des entiers naturels ?

[ 2. ] On suppose que  $P$  n'est pas constant. Déterminer le degré et le coefficient dominant de  $Q$  en fonction de ceux de  $P$ .

[ 3. ] Pour  $n \in \mathbb{N}$ , on note  $s(n)$ , la somme des chiffres dans l'écriture décimale de  $n$ .

[ 3.a. ] Écrire une fonction  $s(n)$  qui renvoie la valeur de  $s(n)$ .

[ 3.b. ] Justifier la terminaison et la correction de l'algorithme utilisé.

[ 3.c. ] Sur des exemples, conjecturer si la suite de terme général  $s(P(n))$  peut être bornée.

Dans la suite du sujet, on suppose que les coefficients de  $P$  sont des entiers naturels.

[ 4. ] Soit  $k \in \mathbb{N}^*$ , le nombre de chiffres dans l'écriture décimale de  $P(n)$ . Démontrer que

$$s(P(n + 10^k)) > s(P(n)).$$

[ 5. ] Démontrer la conjecture faite plus haut [3.c.].

[ 6. ] La suite de terme général  $s(P(n))$  admet-elle une sous-suite bornée ?

[ 7. ] Existe-t-il un polynôme  $P \in \mathbb{Z}[X]$ , non constant, tel que la suite de terme général  $s(|P(n)|)$  soit bornée ?

[ 1. ] Soit  $m \in \mathbb{N}^*$ . D'après la formule du binôme,

$$\begin{aligned} (X + 10^k)^m - X^m &= \sum_{i=1}^m \binom{m}{i} (10^k)^i X^{m-i} = 10^k \sum_{i=0}^{m-1} \underbrace{\binom{m}{i+1}}_{\in \mathbb{N}} 10^{ki} X^{m-1-i} \\ &= 10^k m X^{m-1} + \dots \end{aligned} \quad (\dagger)$$

• Si  $P \in \mathbb{Z}[X]$  est un polynôme dont le degré est inférieur à  $d$ , alors il existe des entiers relatifs  $a_0, \dots, a_d$  tels que

$$P = \sum_{m=0}^d a_m X^m$$

et d'après la décomposition précédente

$$P(X + 10^k) - P = 10^k \sum_{m=0}^d \sum_{i=0}^{m-1} a_m \binom{m}{i+1} 10^{ki} X^{m-1-i}.$$

Cela prouve l'existence et l'unicité du polynôme  $Q \in \mathbb{Z}[X]$ .

• Si les  $a_m$  sont des entiers naturels, il est clair que les coefficients de  $Q$  sont aussi des entiers naturels.

[ 2. ] Si  $P$  est constant, alors  $Q = 0$ ...

• On a remarqué plus haut (†) que : si  $P = X^m$  avec  $m \geq 1$ , alors  $\deg Q = m - 1$  et le coefficient dominant de  $Q$  est égal à  $m$ .

Par linéarité, si  $d = \deg P \geq 1$  et si le coefficient dominant de  $P$  est égal à  $a_d$ , alors  $\deg Q = d - 1$  et le coefficient dominant de  $Q$  est égal à  $da_d$ .

[ 3.a. ] On a le choix entre une version itérative :

```
def s(n):
    somme = 0
    while n > 0:
        somme += (n%10) # ajout d'une nouvelle décimale
        n = n//10
    return somme
```

et une version récursive :

```
def s_rec(n):
    if n==0:
        return 0
    else:
        return (n%10) + s_rec(n//10)
```

[ 3.b. ] Notre algorithme est calqué sur l'algorithme de décomposition en base  $b$  (avec ici  $b = 10$ ). On effectue une succession de divisions euclidiennes, chaque quotient devenant le nouveau dividende. De la sorte, on obtient une famille strictement décroissante d'entiers naturels et une telle famille est finie (Axiome de bon ordre).

✎ Pour formaliser la démonstration, on rappelle que chaque entier naturel  $n$  est représenté d'une unique manière en base  $b = 10$  :

$$n = \sum_{k=0}^d d_k 10^k.$$

En posant

$$q_m = \sum_{k=m}^d d_k 10^{k-m} \quad \text{et} \quad r_m = \sum_{k=0}^{m-1} d_k 10^k,$$

on a (il ne s'agit que de l'astuce taupinale)

$$n = 10^m q_m + r_m \quad \text{et} \quad q_m = 10 \left( \sum_{k=m+1}^d d_k 10^{k-(m+1)} \right) + d_m = 10q_{m+1} + d_m.$$

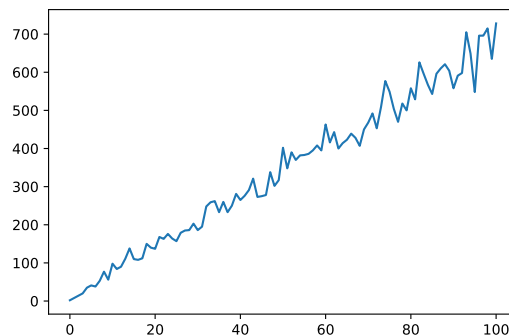
Comme  $0 \leq d_m < 10$ , le couple  $(q_{m+1}, d_m)$  est bien constitué du quotient et du reste de la division euclidienne de  $q_m$  par 10. Voilà pour la correction de l'algorithme.

Comme  $0 \leq 10q_{m+1} \leq q_m$ , la famille des  $q_m$  est strictement décroissante et, comme on l'a dit, l'Axiome de bon ordre prouve la terminaison de l'algorithme.

[ 3.c. ] Une chose est sûre : si la suite de terme général  $s(P(n))$  diverge, alors elle diverge lentement, très lentement...

```
def essai(P, N, x):
    X = list(range(0, N+1))
    S = [s(P(x**n)) for n in X]
    plt.plot(X, S)
```

On fait un essai avec le polynôme  $P = 1 - 2X + 3X^2$ , l'entier  $N = 100$  et  $x = 7$ .

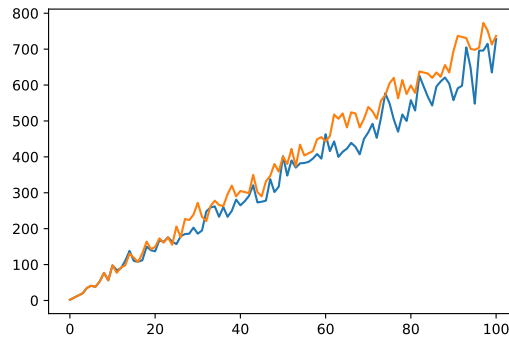


✎ Autre chose : cette question est absurde ! Le module `Polynomial` considère que les coefficients du polynôme sont des flottants (même si on a pris soin de définir un polynôme à coefficients entiers). Par conséquent, l'application de la fonction `s` est une blague.

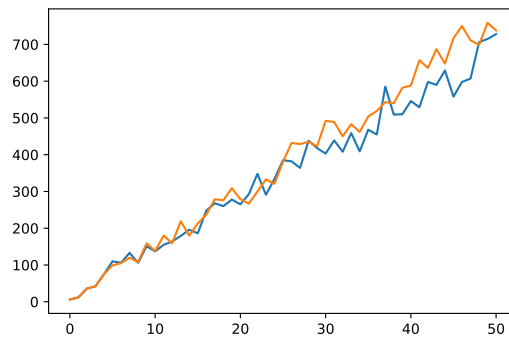
Pour bien faire, il faudrait prendre le temps de coder l'évaluation d'un polynôme.

```
def polynome(C):
    def evaluer(x):
        p, xi = 0, 1
        for i, c in enumerate(C):
            p += c*xi
            xi *= x
        return p
    return evaluer
```

La courbe orange donne les résultats (exact) obtenus en calculant exclusivement sur des entiers (avec les mêmes données). On vient bien que ce n'est pas la même chose, mais ce n'est quand même pas si différent.

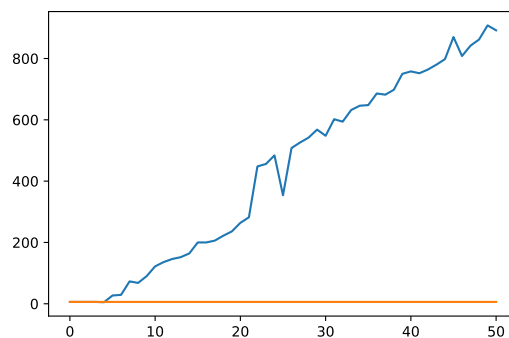


• On recommence avec le polynôme  $P = 1 + 2X^2 + 3X^4$ ,  $N = 50$  seulement (pour éviter les débordements de capacité) et  $x = 7$ .



Mêmes conclusions!

• On recommence à nouveau mais avec  $x = 10$  pour que la suite extraite de terme général  $s(P(10^n))$  soit constante : cette fois, on n'est plus dans le domaine de l'erreur d'arrondi!



C'est quand même agaçant : on demande de prouver la terminaison et la correction d'un algorithme vu en cours, donc sans aucun mystère, et on néglige de mettre l'accent sur une difficulté spécifique au calcul numérique.

[ 4. ] Par définition de  $k$ , l'entier  $P(n)$  s'écrit avec  $k$  chiffres décimaux, donc il est strictement inférieur à  $10^k$ . D'après la première question,

$$P(n + 10^k) = P(n) + 10^k Q(n)$$

et comme les coefficients de  $P$  sont des entiers naturels, alors  $Q(n)$  est aussi un entier naturel (non nul!). Le facteur  $10^k$  nous assure alors que

$$s(P(n + 10^k)) = s(P(n)) + s(Q(n)) > s(P(n)).$$

[ 5. ] D'après la question précédente, il existe une suite extraite de  $(s(P(n)))$  qui est strictement croissante. Comme les  $s(P(n))$  sont des entiers, cette suite extraite tend vers  $+\infty$ , donc la suite de terme général  $s(P(n))$  n'est pas bornée.

[ 6. ] Comme les coefficients de  $P$  sont des entiers naturels, les décimales non nulles de  $P(10^k)$  sont les coefficients de  $P$  :

$$P(10^k) = a_0 + a_1 10^k + a_2 10^{2k} + \dots + a_{d-1} 10^{(d-1)k} + a_d 10^{dk}.$$

Donc, comme on l'a vu plus, la suite extraite de terme général  $s(P(10^k))$  est constante.

[ 7. ] Probablement pas, le contraire m'étonnerait.

Quand  $n$  tend vers  $+\infty$ , l'entier  $|P(10^n n)|$  est approché par  $|a_d| 10^{n^d} n^d$  et j'imagine que  $s(|P(10^n n)|)$  ne peut pas être très inférieur à  $s(|a_d| 10^{n^d} n^d)$ . Affaire à suivre pour ceux que ça amuse!

Un réel  $\alpha$  est dit **algébrique** lorsqu'il existe un polynôme  $P \in \mathbb{Q}[X]$  non nul tel que  $P(\alpha) = 0$ . On pose alors

$$I(\alpha) = \{P \in \mathbb{Q}[X] : P(\alpha) = 0\}.$$

[ 1. ] Démontrer que  $I(\alpha)$  est un idéal de  $\mathbb{Q}[X]$ , puis qu'il existe un polynôme unitaire  $\pi_\alpha \in \mathbb{Q}[X]$  tel que

$$I(\alpha) = \{\pi_\alpha P, P \in \mathbb{Q}[X]\}.$$

Le polynôme  $\pi_\alpha$  est le **polynôme minimal** de  $\alpha$ .

[ 2. ] On note  $S$ , l'ensemble des nombres algébriques  $\alpha$  tels que  $\pi_\alpha$  appartienne à  $\mathbb{Z}[X]$ , que  $|\alpha| > 1$  et que les racines complexes de  $\pi_\alpha$  autres que  $\alpha$  soient de module strictement inférieur à 1.

☞ L'ensemble  $S$  est l'ensemble des **nombre de Salem**.

On note  $E$ , l'ensemble des polynômes minimaux des éléments de  $S$ .

[ 2.a. ] Démontrer que  $\mathbb{Q} \cap S = \mathbb{Z} \setminus \{-1; 0; 1\}$  et que le nombre d'or

$$\varphi = \frac{1 + \sqrt{5}}{2}$$

appartient à  $S$ .

[ 2.b. ] Démontrer que  $S \subset \mathbb{R}$ .

[ 3. ] On considère les polynômes

$$P_1 = X^3 - X - 1$$

$$P_2 = X^3 + X + 1$$

$$P_3 = X^4 - X^3 - 1$$

et on admet que ces polynômes sont les polynômes minimaux de chacune de leurs racines.

[ 3.a. ] Écrire un programme vérifiant si ces trois polynômes appartiennent à  $E$ .

[ 3.b. ] Pour ceux de ces polynômes qui appartiennent à  $E$ , évaluer à  $10^{-13}$  près la valeur de la racine  $\alpha \in S$ .

[ 3.c. ] Évaluer les 70 premiers termes des suites de termes généraux

$$a^n - \lfloor a^n \rfloor \quad \text{et} \quad \sum_{k=0}^n \sin^2(a^k \pi).$$

Que peut-on conjecturer ?

[ 4. ] Démontrer que  $\pi_\alpha$  est irréductible dans  $\mathbb{Q}[X]$ . En déduire que les racines de  $\pi_\alpha$  sont simples.

[ 5. ] Démontrer que la matrice compagnon associée à  $\pi_\alpha$  est diagonalisable.

[ 1. ] Question de cours.

[ 2.a. ] Le polynôme minimal d'un rationnel  $\alpha = p/q$  est  $X - p/q$ . Il est à coefficients entiers si, et seulement si,  $\alpha$  est un entier. Et, par définition, le module de tout élément de  $S$  est strictement supérieur à 1, donc  $\mathbb{Q} \cap S = \mathbb{Z} \setminus \{-1; 0; 1\}$ .

☛ Le nombre d'or  $\varphi$  est une racine de  $X^2 - X - 1 \in \mathbb{Z}[X]$  et l'autre racine de ce polynôme est égale à  $-1/\varphi = 1 - \varphi$  (relations entre coefficients et racines).

[ 2.b. ] Soit  $\alpha \in S$ .

Comme  $\pi_\alpha$  est un polynôme à coefficients réels, ses racines complexes sont deux à deux conjuguées.

Comme  $\alpha$  est une racine de  $\pi_\alpha$ , alors  $\bar{\alpha}$  est aussi une racine de  $\pi_\alpha$  et ces deux racines ont même module. Or, par hypothèse,  $\pi_\alpha$  n'a qu'une seule racine dont le module soit strictement supérieur à 1. Donc  $\alpha = \bar{\alpha}$ , c'est-à-dire  $\alpha \in \mathbb{R}$ .

[ 3.a. ] On commence par écrire efficacement la liste des trois polynômes.

```
X = Polynomial([0,1])
donnees = [X**3-X-1, X**3+X+1, X**4-X**3-1]
```

Pour chacun d'eux, on calcule la liste des racines et on vérifie que la liste des racines dont le module est strictement supérieur à 1 est un singleton.

```
def Salem(P):
    racines = P.roots()
    modules = np.abs(racines)
    return len([m for m in modules if m>1])==1
```

[ 3.b. ] On fait deux fois les mêmes calculs : une première fois pour se limiter aux polynômes concernés par la question, une seconde fois pour afficher une valeur approchée de la racine cherchée.

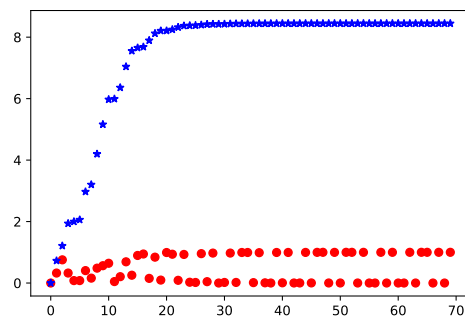
```
RacinesAlpha = []
for P in donnees:
    if Salem(P):
        print(P)
        for z in P.roots():
            if np.abs(z)>1:
                RacinesAlpha.append(z.real)
```

Pour  $P_1$ , on trouve  $\alpha \approx 1.324717957244747$  (supprimer les deux dernières décimales pour répondre à la question posée) et pour  $P_3$ , on trouve  $\alpha \approx 1.3802775690976143$  (supprimer les trois dernières décimales).

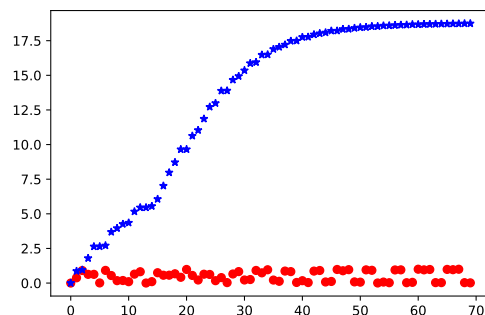
[ 3.c. ] Le code ne présente aucune difficulté.

```
N = 70
for a in RacinesAlpha:
    U, V = [], [0]
    for k in range(N):
        x = a**k
        U.append(x-np.floor(x))
        V.append(V[-1]+np.sin(x*np.pi)**2)
    plt.figure()
    plt.plot(U,'ro')
    plt.plot(V[1:], 'b*')
```

• Pour le polynôme  $P_1$ , il semble que la suite  $(a^n - \lfloor a^n \rfloor)_{n \in \mathbb{N}}$  soit divergente et que ses seules valeurs d'adhérence soient 0 et 1 et d'autre part que la série  $\sum \sin^2(\pi a^n)$  soit convergente.



• Même conjecture pour le polynôme  $P_3$ .



☞ Notons  $\alpha, b$  et  $c$ , les trois racines de  $P_1$ . La suite de terme général

$$u_n = \alpha^n + b^n + c^n$$

vérifie  $u_0 = 3, u_1 = 0$  (somme des racines) et  $u_2 = (\alpha + b + c)^2 - 2(\alpha b + bc + c\alpha) = 2$  et comme c'est une somme de trois suites géométriques, elle vérifie également la relation de récurrence linéaire d'ordre trois :

$$\forall n \in \mathbb{N}, \quad u_{n+3} = u_{n+1} + u_n.$$

Par conséquent,  $u_n \in \mathbb{N}$  pour tout  $n \in \mathbb{N}$  et comme  $|b| = |c| < 1$ ,

$$\alpha^n = \underbrace{u_n}_{\in \mathbb{N}} - \underbrace{(b^n + c^n)}_{\rightarrow 0}.$$

☛ Lorsque le réel  $(b^n + c^n)$  est proche de 0 et positif, la partie entière  $[\alpha^n]$  est égale à  $u_n$  et la partie fractionnaire  $\alpha^n - [\alpha^n]$  est proche de 1; lorsque  $(b^n + c^n)$  est proche de 0 et négatif, la partie entière  $[\alpha^n]$  est égale à  $u_n + 1$  et la partie fractionnaire est proche de 0.

☛ Plus simplement,

$$\sin^2(\pi\alpha^n) = \sin^2[\pi(u_n - b^n - c^n)] = \sin^2[\pi(b^n + c^n)] \underset{n \rightarrow +\infty}{\sim} [\pi(b^n + c^n)]^2$$

et on en déduit que la série  $\sum \sin^2(\pi\alpha^n)$  est absolument convergente par comparaison à une série géométrique.

☛ On peut adapter ce raisonnement aux quatre racines de  $P_3$  avec une relation de récurrence linéaire d'ordre quatre.

[ 4. ] Comme  $\mathbb{C}$  est un corps, si  $\pi_\alpha$  peut être factorisé sous la forme  $PQ$  dans  $\mathbb{Q}[X]$ , alors on peut supposer que  $P(\alpha) = 0$ . Par définition du polynôme minimal,  $\pi_\alpha$  divise  $P$  et, par construction de  $P$ , le polynôme  $P$  divise  $\pi_\alpha$ . Par conséquent,  $P$  et  $\pi_\alpha$  sont associés et  $\pi_\alpha$  est donc irréductible.

☛ Le polynôme dérivé  $\pi'_\alpha$  est lui aussi à coefficients dans  $\mathbb{Z}$ . Par conséquent, le pgcd de  $\pi_\alpha$  et de  $\pi'_\alpha$  est à coefficients dans  $\mathbb{Q}$  (algorithme d'Euclide).

Si  $\pi_\alpha$  admet une racine au moins double  $z_0$ , alors  $\pi_\alpha$  et son polynôme dérivé  $\pi'_\alpha$  sont divisibles par  $(X - z_0)$  dans  $\mathbb{C}[X]$  et de ce fait le pgcd de  $\pi_\alpha$  et de  $\pi'_\alpha$  est un diviseur non constant de  $\pi_\alpha$  qui n'est pas associé à  $\pi_\alpha$  (pour une raison de degré) dans  $\mathbb{Q}[X]$  : cela contredit le fait que  $\pi_\alpha$  soit irréductible dans  $\mathbb{Q}[X]$ .

[ 5. ] Classique (mais pas au programme) : le polynôme caractéristique d'une matrice compagnon est aussi son polynôme minimal.

Soient  $n \in \mathbb{N}^*$  et

$$F = \sum_{k=0}^{n-1} a_k X^k \in \mathbb{C}_{n-1}[X].$$

Pour  $P \in \mathbb{C}_{n-1}[X]$ , on note  $\Phi(P)$ , le reste de la division euclidienne du polynôme  $PF$  par  $X^n - 1$ .

[ 1. ] Démontrer que  $\Phi$  est un endomorphisme de  $\mathbb{C}_{n-1}[X]$ .

[ 2.a. ] Pour  $n = 4$ , écrire une fonction `MatCan(F)` qui renvoie la matrice de  $\Phi$  relative à la base canonique de  $\mathbb{C}_3[X]$ . Tester cette fonction avec

$$F = 1 + X + X^2 + X^3.$$

[ 2.b. ] Tester avec d'autres polynômes de degré 3 et calculer les vecteurs propres de la matrice. Quelle conjecture peut-on faire ?

[ 3. ] On revient au cas général.

Écrire la matrice de  $\Phi$  relative à la base canonique de  $\mathbb{C}_{n-1}[X]$ . L'endomorphisme  $\Phi$  est-il diagonalisable ?

[ 1. ] Pour tout  $P \in \mathbb{C}_{n-1}[X]$ , le reste de la division euclidienne de  $PF$  par  $(X^n - 1)$  est un polynôme dont le degré est strictement inférieur à  $n$  (= le degré du diviseur), donc  $\Phi(P) \in \mathbb{C}_{n-1}[X]$ .

• Par définition, quels que soient  $P_1$  et  $P_2$  dans  $\mathbb{C}_{n-1}[X]$ , quel que soit  $\lambda \in \mathbb{C}$ ,

$$P_1 F = (X^n - 1)Q_1 + \Phi(P_1)$$

$$P_2 F = (X^n - 1)Q_2 + \Phi(P_2)$$

$$(\lambda P_1 + P_2)F = (X^n - 1)Q_0 + \Phi(\lambda P_1 + P_2)$$

et, en combinant les deux premières relations, on obtient également :

$$(\lambda P_1 + P_2)F = (X^n - 1)Q_0 + [\lambda \Phi(P_1) + \Phi(P_2)]$$

où, selon la propriété bien connue,

$$\deg[\lambda \Phi(P_1) + \Phi(P_2)] \leq \max\{\deg P_1, \deg P_2\} < n.$$

Par unicité de la division euclidienne, on peut identifier les restes :

$$\forall P_1, P_2 \in \mathbb{C}_{n-1}[X], \forall \lambda \in \mathbb{C}, \quad \Phi(\lambda P_1 + P_2) = \lambda \Phi(P_1) + \Phi(P_2)$$

donc  $\Phi$  est bien linéaire.

[ 2.a. ] Pour chaque vecteur  $E = X^0, X^1, X^2, X^3$  de la base canonique de  $\mathbb{C}_3[X]$ , on calcule le reste de la division euclidienne du produit  $EF$  par  $Q = X^4 - 1$ .

Les coordonnées relatives à la base canonique du polynôme  $\Phi(E)$  sont en fait ses coefficients, il reste donc à écrire les coefficients de  $\Phi(X^i)$  sur la colonne  $n^{\circ}i$  de la matrice.

```
X = Polynomial([0,1])
n = 4
Q = X**n-1

def matrice_Phi(F):
    M = np.zeros((n,n))
    for j in range(n):
        E = X**j
        R = (E*F)%Q
        for i, x in enumerate(R.coef):
            M[i,j] = x
    return M
```

• La matrice associée au polynôme  $1 + X + X^2 + X^3$  est la matrice dont les seize coefficients sont égaux à 1.

[ 2.b. ] Après quelques essais, il semblerait que la matrice de l'endomorphisme  $\Phi$  associée au polynôme  $F = a_3 X^3 + a_2 X^2 + a_1 X + a_0$  soit la matrice

$$\begin{pmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{pmatrix}$$

[ 3. ] Soit  $n \geq 1$ . Pour  $0 \leq i < n$ , il est clair que

$$\begin{aligned} X^i \sum_{k=0}^{n-1} a_k X^k &= \sum_{k=0}^{n-1} a_k X^{k+i} \\ &= \sum_{k=0}^{n-i-1} a_k X^{k+i} + \sum_{k=n-i}^n a_k (X^{k+i} - X^{k+i-n}) + \sum_{k=n-i}^n a_k X^{k+i-n} \\ &= (X^n - 1) \sum_{k=n-i}^n a_k X^{k+i-n} + \left( \sum_{k=i}^{n-1} a_{k-i} X^k + \sum_{k=0}^{i-1} a_{n-i+k} X^k \right). \end{aligned}$$

La colonne  $n^{\circ}i$  de la matrice est donc

$$\begin{pmatrix} a_{n-i} \\ \vdots \\ a_{n-1} \\ a_0 \\ \vdots \\ a_{n-i-1} \end{pmatrix}.$$

• On a ainsi démontré dans le cas général la conjecture émise au [2.b].

↳ Pour ce qui suit, il faut savoir que la matrice

$$J_n = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ & \ddots & & \\ & & 1 & 0 \\ & & & \ddots \\ & & & & 0 \\ & & & & & \ddots \\ & & & & & & 0 \\ & & & & & & & \ddots \\ & & & & & & & & 0 \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 0 \\ & & & & & & & & & & & \ddots \\ & & & & & & & & & & & & 0 \\ & & & & & & & & & & & & & \ddots \\ & & & & & & & & & & & & & & 0 \end{pmatrix} \in \mathfrak{M}_n(\mathbb{R})$$

est une matrice de permutation, associée à un cycle d'ordre  $n$ .

Les puissances de  $J_n$  s'obtiennent donc par décalage et  $J_n^n = I_n$ .

On vérifie que la matrice de l'endomorphisme  $\Phi$  associé au polynôme

$$F = a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}$$

est égale à  $F(J_n)$ . Comme  $X^n - 1$  est un polynôme annulateur de  $J_n$  et que ce polynôme est scindé (sur  $\mathbb{C}$ ) à racines simples, la matrice  $J_n$  est diagonalisable. Par conséquent, tout polynôme en  $J_n$  est diagonalisable et en particulier, l'endomorphisme  $\Phi$  est diagonalisable.

Pour tout vecteur  $u \in \mathbb{R}^n$  non nul, on pose

$$H_u = I_n - 2 \frac{u \cdot u^T}{u^T \cdot u}$$

et on convient que  $H_u = I_n$  si  $u$  est le vecteur nul.

L'espace  $\mathbb{R}^n$  est muni de sa structure euclidienne canonique et la base canonique est notée  $(e_k)_{1 \leq k \leq n}$ .

- [ 1. ] Écrire une fonction d'argument  $u$  qui renvoie la matrice  $H_u$ .
- [ 2. ] Soit  $u \neq 0$ . Démontrer que  $H_u$  est la matrice de la réflexion par rapport à l'hyperplan  $(\mathbb{R} \cdot u)^\perp$ .
- [ 3. ] En particulier, trouver l'image du vecteur  $(1, 2, 3)$  par la réflexion d'hyperplan  $[x + y + z = 0]$ .
- [ 4. ] Soit  $r$ , la réflexion par rapport à un hyperplan  $H$ . Démontrer qu'il existe un vecteur  $u$  tel que

$$\mathfrak{Mat}_{\mathcal{B}_{can}}(r) = H_u.$$

- [ 5. ] Soient  $v \in \mathbb{R}^n$ , non colinéaire au vecteur (unitaire)  $e_1$ , et  $u = v - \|v\| \cdot e_1$ . Démontrer que

$$H_u(v) = \|v\| \cdot e_1.$$

- [ 6. ] Soit  $A \in \mathfrak{M}_n(\mathbb{R})$ . Démontrer qu'il existe un réel  $\alpha$ , un vecteur  $u \in \mathbb{R}^n$  et une matrice  $B \in \mathfrak{M}_{n-1}(\mathbb{R})$  tels que

$$H_u A = \begin{pmatrix} \alpha & * \\ 0 & B \end{pmatrix}.$$

- [ 7. ] Soit  $M = (m_{i,j})_{1 \leq i,j \leq n} \in \mathfrak{M}_n(\mathbb{R})$ .

- [ 7.a. ] Démontrer qu'il existe une matrice orthogonale  $Q$  et une matrice triangulaire supérieure  $R$  telles que

$$M = QR.$$

- [ 7.b. ] Démontrer l'inégalité d'Hadamard :

$$|\det M|^2 \leq \prod_{j=1}^n \sum_{i=1}^n m_{i,j}^2.$$

🔗 Les **matrices de Householder** sont très utiles en analyse numériques, de même que la factorisation QR. Un tour sur Wikipédia sera instructif.

- [ 1. ] L'argument  $u$  est un objet qui contient trois nombres réels, son type exact importe peu : il suffit que  $u$  puisse être transformé en un vecteur colonne  $U \in \mathfrak{M}_{3,1}(\mathbb{R})$ .

```
def H(u):
    U = np.array(u).reshape((3,1))
    matrice = np.eye(3) - 2*U.dot(U.T)/U.T.dot(U)
    return matrice
```

Le dénominateur est une matrice qui appartient à  $\mathfrak{M}_1(\mathbb{R})$ . Avec la commande `np.vdot(U, U)`, on aurait eu un flottant. Aucune incidence sur les calculs tant qu'on ne divise pas par zéro!

- [ 3. ] Le vecteur cherché est  $(-3, -2, -1)$ .

```
u = np.array([1,1,1])
x = np.array([1,2,3])
y = H(u).dot(x)
```

🔗 Il est paradoxal de procéder ainsi car l'un des principaux intérêts des matrices de Householder consiste précisément à éviter le calcul matriciel! En effet, la complexité du produit matriciel  $(H_u, X) \mapsto H_u X$  est en  $\mathcal{O}(n^2)$  alors qu'on peut calculer le vecteur  $H_u X$  en  $\mathcal{O}(n)$  de la manière suivante.

```
y = x-2*np.dot(u, x)/np.dot(u, u)*u
```

[ 4. ] Quel que soit le vecteur  $u$  non nul et orthogonal à l'hyperplan  $H$ , la projection orthogonale sur  $H$  s'exprime

$$\forall x \in \mathbb{R}^n, \quad \pi_H(x) = x - \frac{\langle u | x \rangle}{\|u\|^2} \cdot u$$

donc la symétrie orthogonale par rapport à  $H$  (= la réflexion d'hyperplan  $H$ ) s'exprime

$$\forall x \in \mathbb{R}^n, \quad r(x) = 2\pi_H(x) - x = x - 2 \frac{\langle u | x \rangle}{\|u\|^2} \cdot u = \left( I_{\mathbb{R}^n} - \frac{2}{\|u\|^2} \cdot u \cdot \langle u | \right)(x).$$

Comme la base canonique de  $\mathbb{R}^n$  est ici supposée orthonormée, on en déduit que la matrice de  $r$  relative à la base canonique est la matrice  $H_u$ .

[ 5. ] Comme  $v$  et  $e_1$  ne sont pas colinéaires, le vecteur  $u$  est non nul.

Comme  $e_1$  est unitaire,

$$\langle u | v \rangle = \|v\|^2 - \|v\| \langle v | e_1 \rangle, \quad \|u\|^2 = 2\|v\|(\|v\| - \langle v | e_1 \rangle).$$

Par conséquent,

$$H_u(v) = v - 2 \frac{\langle u | v \rangle}{\|u\|^2} \cdot u = v - u = \|v\| \cdot e_1$$

(en assimilant  $H_u$  à la réflexion d'hyperplan  $(\mathbb{R} \cdot u)^\perp$ ).

[ 6. ] Si la matrice  $A$  est de la forme

$$\begin{pmatrix} \alpha & \star \\ 0 & B \end{pmatrix},$$

alors il suffit de choisir  $H_u = I_n$ , c'est-à-dire  $u = 0$ .

Si la matrice  $A$  n'est pas de cette forme, cela signifie que  $v = Ae_1$  n'est pas colinéaire à  $e_1$  (et en particulier non nul).

↳ *L'hypothèse faite porte seulement sur la première colonne de  $A$ , c'est-à-dire sur le vecteur  $Ae_1$ .*

En posant  $u = v - \|v\| \cdot e_1$ , on a  $\|v\| \cdot e_1 = H_u v = (H_u A)e_1$ . Par conséquent, en posant  $v = Ae_1$  et  $\alpha = \|v\|$ , on a trouvé une matrice  $H_u$  telle que

$$H_u A = \begin{pmatrix} \alpha & \star \\ 0 & B \end{pmatrix}$$

pour une matrice  $B \in \mathfrak{M}_{n-1}(\mathbb{R})$  convenable.

[ 7.a. ] On raisonne par récurrence sur la taille de la matrice (= on recommence en appliquant au bloc  $B \in \mathfrak{M}_{n-1}(\mathbb{R})$  une transformation analogue à celle qu'on a appliquée à la matrice  $A \in \mathfrak{M}_n(\mathbb{R})$ ).

Il existe donc une famille  $(u_1, \dots, u_n)$  de vecteurs de  $\mathbb{R}^n$  (éventuellement nuls) tels que le produit  $R = H_{u_n} \cdots H_{u_1} A$  soit une matrice triangulaire supérieure.

La matrice  $Q = H_{u_1}^\top \cdots H_{u_n}^\top$  est une matrice orthogonale (en tant que produit de matrices orthogonales) et  $A = QR$ .

[ 7.b. ] Notons  $R_1, \dots, R_n$  et  $M_1, \dots, M_n$  les colonnes des matrices  $R$  et  $M$ . Ces colonnes sont liées par la relation

$$\forall 1 \leq j \leq n, \quad M_j = QR_j$$

et par conséquent

$$\forall 1 \leq j \leq n, \quad \sum_{i=1}^n m_{i,j}^2 = \|M_j\|^2 = M_j^\top \cdot M_j \stackrel{(\dagger)}{=} R_j^\top \cdot R_j \stackrel{(\ddagger)}{=} \sum_{i=1}^j r_{i,j}^2 \geq r_{j,j}^2$$

puisque la matrice  $Q$  est orthogonale  $(\dagger)$  et que la matrice  $R$  est triangulaire supérieure  $(\ddagger)$ .

Comme la matrice  $Q$  est orthogonale, son déterminant est égal à  $\pm 1$  et comme la matrice  $R$  est triangulaire, son déterminant est vite calculé :

$$|\det M|^2 = |\det R|^2 = \prod_{j=1}^n r_{j,j}^2.$$

On en déduit que

$$|\det M|^2 \leq \prod_{j=1}^n \sum_{i=1}^n m_{i,j}^2.$$

↳ L'inégalité d'Hadamard est un grand classique et il est utile d'en connaître une démonstration plus simple, qui repose sur le même principe : décomposer la matrice  $M$  en produit d'une matrice orthogonale et d'une matrice triangulaire.

**Variante** — On commence en remarquant que l'inégalité est évidente lorsque la matrice  $M$  n'est pas inversible.

On suppose donc que la matrice  $M$  est inversible : c'est donc la matrice de passage de la base canonique  $\mathcal{B}_{\text{can}} = (E_1, \dots, E_n)$  de l'espace  $\mathfrak{M}_{n,1}(\mathbb{R})$  des matrices colonnes à la base  $\mathcal{B}_M$  constituée par les colonnes  $(M_1, \dots, M_n)$  de  $M$ .

$$M = \text{Mat}(\mathcal{B}_{\text{can}} \rightarrow \mathcal{B}_M) = \text{Mat}_{\mathcal{B}_M, \mathcal{B}_{\text{can}}}(\mathbf{I})$$

En appliquant l'algorithme de Gram-Schmidt à cette base, on définit une base orthonormée  $\mathcal{B}_U = (U_1, \dots, U_n)$  et la matrice de passage  $P$  de la base  $\mathcal{B}_M$  à la base  $\mathcal{B}_U$  est triangulaire supérieure (propriété caractéristique de l'algorithme de Gram-Schmidt).

$$P = \text{Mat}(\mathcal{B}_M \rightarrow \mathcal{B}_U) = \text{Mat}_{\mathcal{B}_U, \mathcal{B}_M}(\mathbf{I})$$

La matrice  $U$  dont les colonnes sont  $U_1, \dots, U_n$  est une matrice orthogonale (puisque ses colonnes forment une base orthonormée de  $\mathfrak{M}_{n,1}(\mathbb{R})$ ) et, par définition,

$$U = \text{Mat}_{\mathcal{B}_U, \mathcal{B}_{\text{can}}}(\mathbf{I}) = \text{Mat}_{\mathcal{B}_M, \mathcal{B}_{\text{can}}}(\mathbf{I}) \times \text{Mat}_{\mathcal{B}_U, \mathcal{B}_M}(\mathbf{I}) = MP.$$

On a donc  $M = UP^{-1}$  puisque  $P$  est inversible. La matrice  $U$  est orthogonale et la matrice  $P^{-1}$  est, comme son inverse  $P$ , triangulaire supérieure.

Soit  $(u_n)_{n \in \mathbb{N}}$ , une suite réelle. Quels que soient les réels  $a < b$ , on pose

$$S_n(a, b) = \#\{k \in \llbracket 1, n \rrbracket : a \leq u_k \leq b\}.$$

La suite  $(u_n)_{n \in \mathbb{N}}$  est dite **équirépartie** sur  $[0, 1]$  lorsque

$$\forall n \in \mathbb{N}, \quad u_n \in [0, 1] \quad \text{et} \quad \forall 0 \leq a < b \leq 1, \quad \lim_{n \rightarrow +\infty} \frac{S_n(a, b)}{n} = (b - a). \quad (\text{A})$$

La **partie fractionnaire**  $\{x\}$  d'un réel  $x$  est définie par  $\{x\} = x - \lfloor x \rfloor$ .

[ 1. ] Soit  $(u_n)_{n \in \mathbb{N}}$ , une suite équirépartie sur  $[0, 1]$ . Démontrer que l'ensemble  $\{u_n, n \in \mathbb{N}\}$  est dense dans  $[0, 1]$ . La réciproque est-elle vraie ?

[ 2. ] On suppose qu'une suite  $(u_n)_{n \in \mathbb{N}}$  est représentée par une fonction python  $u$ .

[ 2.a. ] Écrire une fonction  $S(u, n, a, b)$  qui renvoie la valeur de  $S_n(a, b)$ .

[ 2.b. ] Appliquer cette fonction aux suites de termes généraux

$$u_n = \{\sqrt{2n}\}, \quad v_n = |\cos n|, \quad w = \{n/2\}.$$

Ces suites semblent-elles équiréparties ?

[ 3. ] Pour  $n \in \mathbb{N}^*$ , on pose

$$D_n = \sup_{0 \leq a < b \leq 1} \left| \frac{S_n(a, b)}{n} - (b - a) \right| \quad \text{et} \quad D_n^* = \sup_{0 \leq \alpha \leq 1} \left| \frac{S_n(0, \alpha)}{n} - \alpha \right|.$$

[ 3.a. ] Écrire une fonction  $D(u, n)$  qui renvoie la valeur de  $D_n$  pour une suite représentée par la fonction  $u$ .

[ 3.b. ] Démontrer que

$$\forall n \in \mathbb{N}^*, \quad D_n^* \leq D_n \leq 2D_n^*.$$

[ 3.c. ] Démontrer qu'une suite  $(u_n)_{n \in \mathbb{N}}$  est équirépartie sur  $[0, 1]$  si, et seulement si, la suite  $(D_n)_{n \geq 1}$  tend vers 0.

[ 4.a. ] Démontrer qu'une suite  $(u_n)_{n \in \mathbb{N}}$  équirépartie sur  $[0, 1]$  vérifie la propriété suivante : pour toute fonction continue  $f : [0, 1] \rightarrow \mathbb{C}$ ,

$$\frac{1}{n} \sum_{k=1}^n f(u_k) \xrightarrow{n \rightarrow +\infty} \int_0^1 f(x) dx. \quad (\text{B})$$

[ 4.b. ] Soient  $0 \leq a < b \leq 1$  et  $\varepsilon > 0$ . Démontrer l'existence de deux fonctions continues  $\psi_\varepsilon$  et  $\varphi_\varepsilon$  telle que

$$\forall x \in [0, 1], \quad \psi_\varepsilon(x) \leq \mathbb{1}_{[a, b]}(x) \leq \varphi_\varepsilon(x) \quad \text{et que} \quad \int_0^1 (\varphi_\varepsilon - \psi_\varepsilon)(x) dx \leq \varepsilon.$$

En déduire qu'une suite qui vérifie la propriété (B) est équirépartie sur  $[0, 1]$ .

[ 5. ] Pour tout  $k \in \mathbb{Z}$ , on pose  $e_k = [t \mapsto \exp(2i\pi kt)]$  et on note  $\mathbb{T}$ , l'espace vectoriels des **polynômes trigonométriques** :

$$\mathbb{T} = \text{Vect}_{\mathbb{C}}(e_k, k \in \mathbb{Z}).$$

On admet la variante suivante du Théorème d'approximation de Weierstrass : Toute fonction continue et 1-périodique de  $\mathbb{R}$  dans  $\mathbb{C}$  est limite pour la convergence uniforme sur  $\mathbb{R}$  d'une suite de polynômes trigonométriques.

[ 5.a. ] Démontrer qu'une suite  $(u_n)_{n \in \mathbb{N}}$  équirépartie sur  $[0, 1]$  vérifie la propriété suivante :

$$\forall p \in \mathbb{N}^*, \quad \frac{1}{n} \sum_{k=1}^n \exp(2i\pi p u_k) \xrightarrow{n \rightarrow +\infty} 0. \quad (\text{C})$$

[ 5.b. ] Démontrer réciproquement que toute suite  $(u_n)_{n \in \mathbb{N}}$  qui vérifie la propriété (C) vérifie aussi la propriété (B). On pourra commencer par considérer  $f \in \mathcal{C}^0([0, 1], \mathbb{C})$  telle que  $f(0) = f(1)$ .

[ 6. ] Soit  $\theta > 0$ . Démontrer que la suite  $(\{n\theta\})_{n \in \mathbb{N}^*}$  est équirépartie si, et seulement si,  $\theta \notin \mathbb{Q}$ .

Les deux critères d'équirépartition de Riemann (B) et de Weyl (C) étudiés à la fin de cet exercice sont présentés dans le sujet Mines 1999 MP 1.

[ 1. ] Si  $0 \leq a < b \leq 1$ , alors  $S_n(a, b)/n \sim (b - a) > 0$ , donc  $S_n(a, b)$  tend vers  $+\infty$  : il existe donc une infinité d'indices  $k \in \mathbb{N}$  tels que  $u_k \in [a, b]$ .

L'ensemble  $\{u_n, n \in \mathbb{N}\}$  est donc dense dans  $[0, 1]$ .

La réciproque est fautive, bien entendu !

L'ensemble  $\mathcal{D}$  des rationnels dyadiques, c'est-à-dire de la forme  $\frac{2m+1}{2^k}$  avec  $m \in \llbracket 0, 2^k \rrbracket$  et  $k \in \mathbb{N}$ , est dense dans  $\mathbb{R}$ .

Cet ensemble peut facilement être muni d'une relation de bon ordre, inspirée de l'ordre lexicographique : avec

$$x = \frac{2m+1}{2^k} \quad \text{et} \quad y = \frac{2n+1}{2^\ell},$$

on convient que  $x < y$  si  $k < \ell$  (indépendamment de  $m$  et  $n$ ) ou si  $k = \ell$  et  $m \leq n$ .

On peut alors considérer une suite  $(u_n)_{n \in \mathbb{N}}$  qui parcourt l'ensemble  $\mathcal{D}$  en croissant (pour la relation  $<$ ), en prenant deux fois chaque valeur inférieure à  $1/2$  et une seule fois chaque valeur strictement supérieure à  $1/2$ . On devine que

$$\lim_{n \rightarrow +\infty} \frac{S_n(0, 1/2)}{n} = \frac{2}{3} \quad \text{et que} \quad \lim_{n \rightarrow +\infty} \frac{S_n(1/2, 1)}{n} = \frac{1}{3}.$$

(C'est un peu fastidieux à rédiger.)

[ 2.a. ] Le code est sans mystère.

```
def S(u, n, a, b):
    compteur = 0
    for k in range(n):
        valeur = u(k+1)
        compteur += (a <= valeur) and (valeur <= b)
    return compteur
```

Cependant, on peut produire un code analogue mais beaucoup plus rapide en tirant parti des capacités de numpy.

```
def S_var(u, n, a, b):
    U = np.array([u(k+1) for k in range(n)])
    filtrage = (U >= a) * (U <= b)
    return np.sum(filtrage)
```

[ 2.b. ] On choisit quelques intervalles  $[a, b]$  et on compare la longueur du segment à la proportion de termes qui se trouvent dans ce segment.

```
A = [0, 0.2, 0.4, 0.6, 0.8]
B = [0.2, 0.4, 0.6, 0.8, 1]
for a, b in zip(A, B):
    for k in range(2, 6):
        n = 10**k
        prop = S_var(u, n, a, b)/n
        err = np.abs(prop - b + a) / (b - a)
        print("{:5.2%}".format(err))
    print()
```

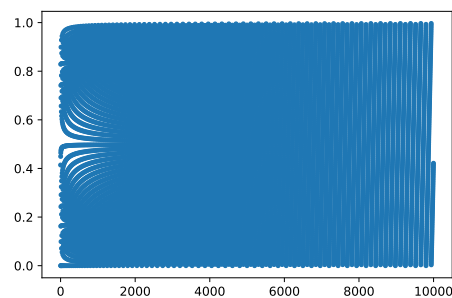
Il semble alors que la suite  $(u_n)_{n \in \mathbb{N}}$  soit équiréparties. En revanche, il est assez clair que  $(v_n)_{n \in \mathbb{N}}$  et  $(w_n)_{n \in \mathbb{N}}$  ne sont pas équiréparties : les pourcentages d'erreur calculés ne sont pas négligeables et ne diminuent pas quand l'entier  $n$  augmente !

☞ Une impression visuelle serait sans doute plus convaincante.

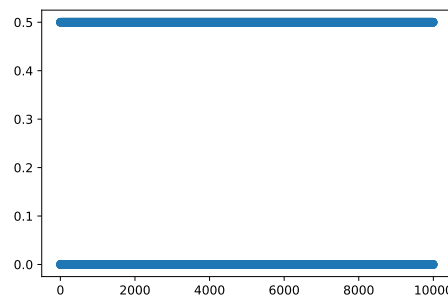
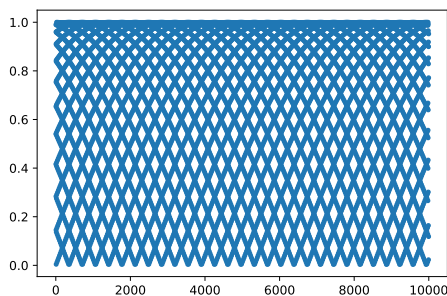
```
N = 10000
U = [u(k+1) for k in range(N)]
V = [v(k+1) for k in range(N)]
W = [w(k+1) for k in range(N)]

plt.figure()
plt.plot(U, '.')
plt.figure()
plt.plot(V, '.')
plt.figure()
plt.plot(W, 'o')
```

Les points semblent assez uniformément répartis pour la suite  $u$ .



En revanche, ils sont beaucoup moins bien répartis pour la suite  $v$  (à gauche) et pas du tout répartis pour la suite  $w$  (à droite).



[ 3.a. ] À première vue, ça paraît impossible, puisqu'il y a une infinité de segments  $[a, b]$  possibles...

Mais quand on y réfléchit sérieusement, on se rappelle qu'il n'y a que  $n$  termes de la suite considérés et, de ce fait, il n'y a qu'un nombre *fini* d'intervalles à prendre en compte.

Supposons donc que les termes  $u_1, \dots, u_n$  soient rangés dans l'ordre croissant :

$$0 \leq u_{\sigma(1)} \leq u_{\sigma(2)} \leq \dots \leq u_{\sigma(n)} \leq 1.$$

Il faut prendre en compte :

- les segments  $[u_{\sigma(k)}, u_{\sigma(k)}]$  de longueur nulle qui contiennent un terme ;
- les segments  $[u_{\sigma(k)}, u_{\sigma(k+1)}]$  de longueur  $u_{\sigma(k+1)} - u_{\sigma(k)}$  (qui peut être nulle...) et qui contiennent deux termes ;
- etc...
- le segment  $[u_{\sigma(1)}, u_{\sigma(n)}]$  de longueur  $u_{\sigma(n)} - u_{\sigma(1)}$  et qui contient  $n$  termes ;
- et enfin les intervalles ouverts

$$]0, u_{\sigma(1)}[, ]u_{\sigma(1)}, u_{\sigma(2)}[, \dots, ]u_{\sigma(k)}, u_{\sigma(k+1)}[, \dots, ]u_{\sigma(n-1)}, u_{\sigma(n)}[, ]u_{\sigma(n)}, 1[$$

qui ne contiennent aucun point.

La borne supérieure cherchée est en fait un maximum et ce maximum est la valeur calculée pour l'un de ces intervalles.

```
def D(u, n):
    # On trie les n termes de la suite
    U = sorted([u(k+1) for k in range(n)])
    max_ref = 1/n # cas des singletons
    for d in range(1, n):
        for i in range(n-d):
            valeur = np.abs((d+1)/n - U[i+d] + U[i])
            if max_ref < valeur:
                max_ref = valeur
    if U[0] > max_ref:
        max_ref = U[0]
    if 1-U[-1] > max_ref:
        max_ref = 1-U[-1]
    for i in range(1, n):
        valeur = U[i]-U[i-1]
        if max_ref < valeur:
            max_ref = valeur
    return max_ref
```

Ce code est **quadratique** en  $n$  et met plusieurs minutes à tourner pour  $n = 10^5$ . Cela dit, avec de la patience, le résultat obtenu est de l'ordre de  $10^{-2}$  pour  $u$ ; de l'ordre de 0,2 pour  $v$  et égal à 0,5 pour  $w$ .

- ☞ *Étrange question!*
- ☞ *Fournir une réponse exacte est délicat et le code met du temps à tourner.*
- ☞ *On pourrait fournir une réponse approchée en calculant les termes  $u_k$ ,  $1 \leq k \leq n$ , et en sélectionnant aléatoirement  $N$  segments  $[a, b] \subset [0, 1]$  et en déterminant la plus grande valeur de*

$$\varphi(a, b) = \left| \frac{S_n(a, b)}{n} - (b - a) \right|$$

pour cette famille de  $N$  segments ("méthode de Monte-Carlo"). Mais pour avoir une précision correcte, il faudrait prendre une "grande" valeur de  $N$ , ce qui conduit également à des calculs assez longs.

- ☞ *En tout cas, il ne faut surtout pas utiliser la fonction  $S(u, n, a, b)$  définie précédemment!*
- Mieux vaut alors passer une telle question et consacrer du temps à ce qui suit.*

Étant donnés deux polynômes

$$P = a_0 + a_1X + \dots + a_nX^n \quad \text{et} \quad Q = b_0 + b_1X + \dots + b_mX^m$$

à coefficients complexes, le **résultant** de  $P$  et  $Q$  est le déterminant  $\text{Res}(P, Q)$  de la matrice

$$S(P, Q) = \begin{pmatrix} a_0 & 0 & 0 & \dots & b_0 & 0 & \dots & 0 \\ a_1 & a_0 & \ddots & \ddots & b_1 & b_0 & \ddots & \vdots \\ a_2 & a_1 & \dots & 0 & b_2 & b_1 & \dots & \vdots \\ \vdots & \vdots & \vdots & a_0 & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & a_n & \dots & \dots & \dots & b_m \end{pmatrix} \in \mathfrak{M}_{n+m}(\mathbb{C}).$$

[ 1. ] Coder  $\text{Res}(P, Q)$  pour  $P$  et  $Q$  de degré 2.

[ 2. ] Calculer  $\text{Res}(P, Q)$  pour

[ 2.a. ]  $P = (X-1)(X-2)$  et  $Q = (X-1)(X-3)$ ,

[ 2.b. ]  $P = (X-1)(X-2)$  et  $Q = (X-3)(X-4)$ ,

[ 2.c. ]  $P = X^2 - 2X + 1$  et  $Q = X^2 - 2$ .

[ 3. ] Que peut-on conjecturer ?

[ 4. ] On considère l'application  $\varphi$  définie par

$$\forall (U, V) \in \mathbb{C}_{m-1}[X] \times \mathbb{C}_{n-1}[X], \quad \varphi(U, V) = PU + QV.$$

[ 4.a. ] Démontrer que  $\varphi$  est linéaire et prend ses valeurs dans  $\mathbb{C}_{n+m-1}[X]$ .

[ 4.b. ] Démontrer que  $\varphi$  est injective si, et seulement si,  $P$  et  $Q$  n'ont pas de racine complexe commune.

[ 4.c. ] On pose

$$\mathcal{B} = ((1, 0), \dots, (X^{m-1}, 0), (0, 1), \dots, (0, X^{n-1}))$$

$$\mathcal{B}' = (1, X, \dots, X^{n+m-1}).$$

Démontrer que

$$S(P, Q) = \mathfrak{Mat}_{\mathcal{B}, \mathcal{B}'}(\varphi).$$

[ 5. ] Pour  $P \in \mathbb{C}[X]$ , le **discriminant**  $\Delta$  de  $P$  est défini par

$$\Delta = \text{Res}(P, P').$$

[ 5.a. ] Démontrer que les racines de  $P$  sont simples si, et seulement si,  $\Delta \neq 0$ .

[ 5.b. ] Soit  $P = aX^2 + bX + c$ , un polynôme de degré 2. Dédurre de ce qui précède que  $P$  admet deux racines complexes distinctes si, et seulement si,  $b^2 - 4ac \neq 0$ .

[ 1. ] La définition de la matrice  $S(P, Q)$  est tout sauf simple. Il faut prendre le temps de bien la comprendre :

— Dans un premier temps, on recopie en décalant d'un cran à chaque fois les coefficients de  $P$ ;

— dans un second temps, on recopie en décalant d'un cran à chaque fois les coefficients de  $Q$ ;

— on obtient au bout du compte une matrice carrée de taille  $n + m$ .

Sur la colonne de rang 0, le coefficient  $a_0$  est sur la ligne de rang 0 et le coefficient  $a_n$  est sur la ligne de rang  $n$ .

À cause du décalage, sur la colonne de rang  $j$ , le coefficient  $a_0$  est sur la ligne de rang  $0 + j = j$  et le coefficient  $a_n$  est sur la ligne de rang  $n + j$ .

Au terme de la première partie, le coefficient  $a_n$  est sur la dernière ligne, c'est-à-dire sur la ligne de rang  $n + (m - 1)$ . Cette colonne est donc la colonne de rang  $j = (m - 1)$ .

Dans la seconde partie de la matrice, c'est-à-dire pour  $m \leq j < m + n$ , le coefficient  $b_0$  est sur la ligne de rang  $(j - m)$  et le coefficient  $b_m$  est sur la ligne de rang  $(j - m) + m = j$ .

En particulier, sur la dernière colonne, c'est-à-dire pour  $j = m + n - 1$ , le coefficient  $b_m$  est sur la ligne de rang  $j = m + n - 1$ , c'est-à-dire sur la dernière ligne.

✎ Maintenant que tout est clair, on peut tirer parti des commodités offertes par numpy pour coder efficacement la matrice  $S(P, Q)$ .

```
def S(P, Q):
    m, n = P.degree(), Q.degree()
    A = np.zeros((m+n, m+n))
    for j in range(m):
        A[j:j+n+1, j] = P.coef
    for j in range(m, m+n):
        A[j-m:j+1, j] = Q.coef
    return A

def Res(P, Q):
    return alg.det(S(P, Q))
```

[ 2.a. ] Le résultant affiché est de l'ordre de  $10^{-16}$ . Comme il s'agit en fait d'un entier (= déterminant d'une matrice à coefficients entiers), on conclut que ce résultant est nul.

[ 2.b. ] De même, aux erreurs d'arrondi près, le résultant est égal à 12.

[ 2.c. ] Résultant égal à 1.

[ 3. ] Pas grand'chose si on n'a jamais entendu parler du résultant...

Si on a jeté un œil sur la suite de l'énoncé [5.a.] (ou si on a déjà rencontré le résultant au cours de ses aventures mathématiques), on peut conjecturer que le résultant de deux polynômes est nul si, et seulement si, ces deux polynômes ont une racine en commun.

↳ Pour apprécier ce théorème à sa juste valeur, il faut remarquer qu'on peut calculer le résultant de deux polynômes en ignorant tout de leurs racines respectives : il suffit de connaître les coefficients de ces polynômes.

[ 4.a. ] Comme  $\deg P = n$  et  $\deg Q = m$ , alors

$$\forall U \in \mathbb{C}_{m-1}[X], \quad \deg(PU) \leq n + m - 1 \quad \text{et} \quad \forall V \in \mathbb{C}_{n-1}[X], \quad \deg(QV) \leq m + n - 1.$$

Par conséquent,  $\deg \varphi(U, V) \leq n + m - 1$ .

La linéarité de  $\varphi$  est claire.

[ 4.b. ] Comme  $P$  et  $Q$  sont des polynômes non constants, ils sont scindés dans  $\mathbb{C}[X]$ . Donc :

- ou bien  $P$  et  $Q$  sont premiers entre eux;
- ou bien  $P$  et  $Q$  ont une racine commune.

✎ Si  $P$  et  $Q$  ont une racine commune  $\alpha$ , alors il existe deux polynômes  $P_0 \in \mathbb{C}_{m-1}[X]$  et  $Q_0 \in \mathbb{C}_{n-1}[X]$  tels que

$$P = (X - \alpha)P_0 \quad \text{et} \quad Q = (X - \alpha)Q_0.$$

Dans ces conditions,  $(Q_0, -P_0) \in \mathbb{C}_{n-1}[X] \times \mathbb{C}_{m-1}[X]$  est non nul et

$$\varphi(Q_0, -P_0) = (X - \alpha)P_0Q_0 - (X - \alpha)P_0Q_0 = 0,$$

donc l'application  $\varphi$  n'est pas injective.

✎ Si  $P$  et  $Q$  sont premiers entre eux et s'il existe  $(U, V) \in \mathbb{C}_{m-1}[X] \times \mathbb{C}_{n-1}[X]$  tel que  $\varphi(U, V) = 0$ , alors

$$PU = Q(-V).$$

On déduit du Théorème de Gauss que  $P$  divise  $V$  et que  $Q$  divise  $U$ .

Mais  $\deg V < n = \deg P$  et  $\deg U < m = \deg Q$ , donc en fait  $(U, V) = (0, 0)$  et cela prouve que l'application  $\varphi$  est injective.

✎ On a ainsi démontré que  $\varphi$  était injective si, et seulement si,  $P$  et  $Q$  n'avaient aucune racine commune.

↳ Si on considère des polynômes à coefficients dans un corps  $\mathbb{K}$ , il ne faut plus raisonner sur les racines mais sur les diviseurs irréductibles de  $P$  et de  $Q$  (les polynômes non constants ne sont pas nécessairement scindés).

Si  $P$  et  $Q$  n'ont aucun facteur irréductible commun (= s'ils sont premiers entre eux), la démonstration qu'on a donnée prouve que l'application  $\varphi$  est injective.

Si  $P$  et  $Q$  ont un facteur irréductible  $R_0$  (= s'ils ne sont pas premiers entre eux), il est facile d'adapter la démonstration qu'on a donnée pour prouver que  $\varphi$  n'est pas injective.

*Autrement dit : pour des polynômes à coefficients dans un corps  $\mathbb{K}$ , l'application linéaire  $\varphi$  est injective si, et seulement si, les deux polynômes  $P$  et  $Q$  sont premiers entre eux.*

[ 4.c. ] Il suffit de revenir à la définition de la matrice d'une application linéaire.

[ 5.a. ] Les racines complexes de  $P$  sont simples si, et seulement si, les polynômes  $P$  et  $P'$  n'ont pas de racine commune, c'est-à-dire si l'application linéaire  $\varphi$  est injective.

Or l'espace de départ  $\mathbb{C}_{n-2}[X] \times \mathbb{C}_{n-1}[X]$  et l'espace d'arrivée  $\mathbb{C}_{(n-1)+n-1}[X] = \mathbb{C}_{2n-2}[X]$  de  $\varphi$  ont même dimension :

$$(n-2+1) + (n-1+1) = 2n-1 = (2n-2) + 1$$

donc (Théorème du rang) l'application  $\varphi$  est injective si, et seulement si, le déterminant de sa matrice est non nul.

Donc les racines complexes de  $P$  sont simples si, et seulement si, le discriminant de  $P$  est non nul.

↳ *Le même raisonnement peut se tenir dans un corps  $\mathbb{K}$  quelconque : le discriminant de  $P \in \mathbb{K}[X]$  est nul si, et seulement si,  $P$  admet un facteur carré (= l'un des facteurs irréductibles apparaît avec une valuation supérieure à 2).*

[ 5.b. ] Avec  $P = aX^2 + bX + c$  et  $P' = 2aX + b$ , on a  $n = 2$  et  $m = 1$ , donc  $S(P, P') \in \mathfrak{M}_3(\mathbb{C})$  :

$$S(P, P') = \begin{pmatrix} c & b & 0 \\ b & 2a & b \\ a & 0 & 2a \end{pmatrix}$$

et  $\text{Res}(P, P') = 4a^2c + ab^2 - 2ab^2 = a(4ac - b^2)$  (avec la règle de Sarrus, bien entendu).

Comme  $a \neq 0$  (puisque  $\deg P = 2$ ), on en déduit que  $P$  admet deux racines complexes distinctes si, et seulement si,  $4ac - b^2 \neq 0$ .

↳ *Pour la culture, si  $P = X^3 + pX + q$  (forme canonique d'un polynôme de degré 3), alors  $P' = 3X^2 + p$  et  $m + n = 5$  :*

$$S(P, P') = \begin{pmatrix} q & 0 & p & 0 & 0 \\ p & q & 0 & p & 0 \\ 0 & p & 3 & 0 & p \\ 1 & 0 & 0 & 3 & 0 \\ 0 & 1 & 0 & 0 & 3 \end{pmatrix}.$$

On retrouve ainsi  $\text{Res}(P, P') = 27q^2 + 4p^3$ .