

## Exercice de l'oral de centrale 2.

**Exercice 1**

Soit  $P$  un polynôme de  $\mathbb{R}[X]$ . On pose  $S(P) = \sum_{k=0}^{+\infty} \frac{P(k)}{k!}$ .

1. a) Montrer que  $S(P)$  est bien défini.
- b) Montrer que  $S$  est une forme linéaire.
- c) Calculer avec Python  $\frac{1}{e} \sum_{k=0}^{50} \frac{P(k)}{k!}$  avec  $P = X^d$ ,  $d \in \{1, \dots, 10\}$ , puis avec  $P = X^9 + 36X^6 - X^3 + X^2 - 3$ .

Observations ?

2. Soit  $(H_n)$  la suite de polynômes définie par  $H_0 = 1$  et, pour  $n \in \mathbb{N}$ ,  $H_{n+1} = (X - n)H_n$ .
  - a) Montrer que  $(H_n)_{n \in \mathbb{N}}$  est une base de  $\mathbb{R}[X]$ .
  - b) Calculer  $S(H_n)$  pour tout  $n \in \mathbb{N}$ .
  - c) Comment calculer  $S(P)$  pour  $P$  quelconque ?

SOLUTION. —

1. (a) Si  $n$  est le degré de  $P$  et  $\alpha$  son coefficient dominant, on a  $P(x) \sim \alpha x^n$  donc  $\frac{P(k)}{k!} \sim \frac{\alpha k^n}{k!} = O\left(\frac{1}{k^2}\right)$ . Le critère de domination assure donc la convergence de la série  $S(P)$ .
- (b) Pour  $P, Q \in \mathbb{R}[X]$  et  $\lambda, \mu \in \mathbb{R}$ , on a pour tout  $k \in \mathbb{N}$  l'identité  $\frac{(\lambda P + \mu Q)(k)}{k!} = \lambda \frac{P(k)}{k!} + \mu \frac{Q(k)}{k!}$ . On peut sommer ces égalités (les séries convergent d'après la question précédente) pour obtenir

$$S(\lambda P + \mu Q) = \sum_{k=0}^{+\infty} \frac{(\lambda P + \mu Q)(k)}{k!} = \lambda \sum_{k=0}^{+\infty} \frac{P(k)}{k!} + \mu \sum_{k=0}^{+\infty} \frac{Q(k)}{k!} = \lambda S(P) + \mu S(Q).$$

(c)

```
>>> from math import factorial, exp
>>> def somme(P):
s = 0
for i in range(51):
s+ = P(i)/factorial(i)
return(s)
>>> for d in range(10):
P = lambda x: x**d
print(somme(P)/exp(1))
1.0000000000000002
1.0000000000000002
2.0
5.0000000000000001
15.000000000000002
51.99999999999998
203.0
876.9999999999999
4140.0
21147.000000000004
>>> P = lambda x: x**9 + 36*x**6 - x**3 + x**2 - 3
>>> somme(P)/exp(1)
28448.999999999985
```

Les résultats numériques permettent de formuler l'hypothèse que  $S(P)/e$  est un entier lorsque  $P$  est à coefficients entiers...

2. (a) La famille  $(H_n)_{n \in \mathbb{N}}$  est formée de polynômes de degrés étagés, c'est donc une base de  $\mathbb{R}[X]$ .

(b) Soit  $n \in \mathbb{N}^*$ . Pour tout  $k \in \mathbb{N}$ , on a  $H_n(k) = \prod_{i=0}^k (n-i) = 0$  si  $k \leq n-1$ , et  $H_n(k) = \frac{k!}{(k-n)!}$  si  $k \geq n$ . Il vient  $S(H_n) = \sum_{k=n}^{+\infty} \frac{k!}{k!(k-n)!} = \sum_{k=n}^{+\infty} \frac{1}{(k-n)!} = \sum_{i=0}^{+\infty} \frac{1}{i!} = e$ . Ce résultat reste valable pour  $n = 0$ .

(c) Il suffit de décomposer  $P$  dans la base  $(H_n)_{n \in \mathbb{N}}$  pour calculer  $S(P)$  par linéarité. Précisément, si  $P = \sum_{k=0}^n \alpha_k H_k$  alors  $S(P) = \sum_{k=0}^n \alpha_k S(H_k) = e \sum_{k=0}^n \alpha_k$ .

Pour ce faire, si  $n = \deg(P)$ , il suffit de faire la division euclidienne de  $P$  par  $H_n$  pour trouver  $\alpha_n$  (quotient) et le reste, qui vaut  $P - \alpha_n H_n = \sum_{k=0}^{n-1} \alpha_k H_k$ . On itère alors le procédé avec une division par  $H_{n-1}$ , etc.

**N.B.** Si  $P$  est à coefficients entiers,  $\alpha_n$  est le coefficient dominant de  $P$  donc un entier ( $H_n$  est unitaire). Et le reste de la division euclidienne vaut  $P - \alpha_n H_n$  donc reste à coefficients entiers. En itérant, tous les coefficients  $\alpha_k$  sont des entiers donc  $S(P)/e = \sum_{k=0}^n \alpha_k$  est un entier.

## Exercice 2

Soit  $A = (a_{i,j})_{1 \leq i,j \leq n} \in \mathcal{M}_n(\mathbb{C})$ . On pose, pour  $i \in \{1, \dots, n\}$ ,  $\rho_i(A) = |a_{i,i}| - \sum_{j \neq i} |a_{i,j}|$ . On dit que  $A$  est à diagonale strictement dominante si, pour tout  $i \in \{1, \dots, n\}$ ,  $\rho_i(A) > 0$ .

1. Soit  $A$  une matrice à diagonale strictement dominante. Soit  $X = (x_1, \dots, x_n)^\top \in \mathbb{C}^n$  tel que  $AX = 0$ . Soit  $i_0$  tel que  $|x_{i_0}| = \max_{1 \leq k \leq n} |x_k|$ .

Montrer que  $|a_{i_0,i_0}| \times |x_{i_0}| \leq |x_{i_0}| \sum_{j \neq i_0} |a_{i_0,j}|$ . En déduire que  $A$  est inversible.

2. Que dire de la réciproque ?
3. écrire une fonction qui à une matrice  $A$  associe  $\rho_1(A) \times \dots \times \rho_n(A)$ .
4. Comparer  $\rho_1(A) \times \dots \times \rho_n(A)$  et  $|\det A|$ . Conjecture ?
5. Démontrer la conjecture précédente.

SOLUTION. —

1. Comme  $AX = 0$ , la composante numéro  $i_0$  de ce vecteur est nulle, c'est-à-dire que  $\sum_{j=1}^n a_{i_0,j} x_j = 0$ , donc

$a_{i_0,i_0} x_{i_0} = - \sum_{j \neq i_0} a_{i_0,j} x_j$ . L'inégalité triangulaire et l'hypothèse de maximalité de  $|x_{i_0}|$  mènent alors à

$$|a_{i_0,i_0}| |x_{i_0}| \leq \sum_{j \neq i_0} |a_{i_0,j}| |x_j| \leq |x_{i_0}| \sum_{j \neq i_0} |a_{i_0,j}|,$$

donc  $\rho_{i_0}(A) |x_{i_0}| \leq 0$ . Comme  $\rho_{i_0}(A)$  est strictement positif, cela implique que  $x_{i_0} = 0$ . Par maximalité de  $|x_{i_0}|$ , toutes les composantes de  $X$  sont nulles, donc le noyau de  $A$  est réduit au vecteur nul, donc  $A$  est inversible.

2. La réciproque est fausse, comme le montre l'exemple de la matrice antidiagonale inversible (et qui n'est pas à diagonale strictement dominante) suivante :

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

On peut généraliser ce contre-exemple à une matrice carrée d'ordre quelconque.

3. On commence par écrire une fonction `rho(A, i)` qui calcule  $\rho_i(A)$ , puis une fonction, appelée `Rho(A)`, qui effectue le produit des  $\rho_i(A)$  :

```
import numpy as np
import numpy.linalg as alg

def rho(A, i):
```

```

n = len(A)
s = 0
for j in range(n):
    s = s + abs(A[i, j])
    s = 2*abs(A[i, i]) - s
return(s)

def Rho(A):
    n = len(A)
    p = 1
    for i in range(n):
        p = p*rho(A, i)
    return(p)

M = np.array([[3, 1, -1], [2, 5, -1], [-1, -4, -6]])
print(Rho(M))

```

Le résultat de la dernière commande est 2, comme on peut le vérifier par un calcul mental élémentaire. On note que  $\text{Rho}(A)$  est définie pour toutes les matrices carrées complexes : c'est à l'utilisateur de veiller à l'employer pour des matrices à diagonale strictement dominante.

4. On effectue la différence  $|\det(A)| - \prod_{i=1}^n \rho_i(A)$  pour plusieurs matrices à diagonale strictement dominante, dont une matrice diagonale :

```

import numpy.linalg as alg

N = np.diag([2, 3, -4])
O = np.array([[-3, 1, 0], [1, -2, 0], [5, 6, -12]])
print(abs(alg.det(M)) - Rho(M))
print(abs(alg.det(N)) - Rho(N))
print(abs(alg.det(O)) - Rho(O))

```

On obtient 84, 0 et 58. D'autres expérimentations suggèrent que  $\prod_{i=1}^n \rho_i(A)$  est majoré par  $|\det A|$  pour toute matrice à diagonale strictement dominante.

5. On pose

$$D = \text{normalfontdiag}\left(\frac{1}{\rho_1(A)}, \dots, \frac{1}{\rho_n(A)}\right) \quad \text{et} \quad B = DA.$$

Ainsi, la matrice  $B$  est obtenue en divisant la ligne  $i$  de  $A$  par le nombre  $\rho_i(A)$ , non nul car strictement positif, pour tout  $i \in \llbracket 1, n \rrbracket$ . On note  $b_{i,j}$  les coefficients de la matrice  $B$ , et on constate que

$$\forall i \in \llbracket 1, n \rrbracket, \quad \rho_i(B) = |b_{i,i}| - \sum_{j \neq i} |b_{i,j}| = \frac{1}{\rho_i(A)} \left( |a_{i,i}| - \sum_{j \neq i} |a_{i,j}| \right) = 1.$$

Soient  $\lambda$  une valeur propre complexe de  $B$ , et  $X = (x_1, \dots, x_n)^\top$  un vecteur propre associé. On note  $i$  un indice tel que  $|x_i| = \max_{1 \leq j \leq n} |x_j|$ . Comme  $X$  est un vecteur propre, on a  $x_i \neq 0$  et, quitte à diviser par  $x_i$ , on peut supposer que  $x_i = 1$  et que  $|x_j| \leq 1$  pour tout  $j \neq i$ . La  $i$ -ème composante de l'égalité  $BX = \lambda X$  s'écrit alors

$$\sum_{j=1}^n b_{i,j} x_j = \lambda x_i \quad \text{ou encore} \quad b_{i,i} + \sum_{j \neq i} b_{i,j} x_j = \lambda.$$

L'inégalité triangulaire et la majoration  $\forall j \neq i, |x_j| \leq 1$  entraînent alors que

$$|\lambda| \geq |b_{i,i}| - \left| \sum_{j \neq i} b_{i,j} x_j \right| \geq |b_{i,i}| - \sum_{j \neq i} |b_{i,j}| |x_j| \geq |b_{i,i}| - \sum_{j \neq i} |b_{i,j}| = \rho_i(B) = 1.$$

On a donc démontré que toutes les valeurs propres de  $B$  sont de module au moins 1, et on en déduit que

$$|\det B| = \left| \prod_{\lambda \in \text{Spec}(B)} \lambda^{m_\lambda} \right| \geq 1.$$

Par multilinéarité du déterminant par rapport aux lignes, on a  $\det B = \frac{\det A}{\rho_1(A) \cdots \rho_n(A)}$ , et l'inégalité ci-dessus fournit la majoration attendue :

$$\rho_1(A) \times \cdots \times \rho_n(A) \leq |\det A|.$$

REMARQUE. — On peut tenter une démonstration à partir des valeurs propres  $\mu_k$  de la matrice  $A$ . Le même raisonnement que ci-dessus montre alors que  $|\mu_k| \geq \rho_i(A)$ , où  $i$  est un indice correspondant au maximum du module des composantes d'un vecteur propre de  $A$  associé à  $\mu_k$ . Malheureusement, on ne dispose d'aucun renseignement sur l'application  $k \mapsto i$  ainsi définie : rien n'indique qu'elle soit une permutation de  $\llbracket 1, n \rrbracket$ , et on obtient que  $|\det A| = |\prod_{k=1}^n \mu_k|$  est minoré par un produit de certains  $\rho_i(A)$ , sans avoir la certitude que chacun d'eux intervient une fois et une seule.

La démonstration présentée évacue ce problème en considérant la matrice à diagonale strictement dominante  $B$  pour laquelle  $i \mapsto \rho_i(B)$  est constante, et dans laquelle on a caché une et une seule fois chaque  $\rho_i(A)$ .

### Exercice 3

1. Soit  $M = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1/4 & \frac{1}{3} & \frac{1}{6} \\ 1/4 & 1/4 & 1/4 \end{pmatrix}$ .

- Afficher les trente premières puissances de  $M$ .
  - Afficher les valeurs propres des trente premières puissances de  $M$ .
  - Que peut-on raisonnablement conjecturer ?
- Soient  $(a_1, \dots, a_n) \in \mathbb{C}^n$  et  $V = (a_i^{j-1})_{1 \leq i, j \leq n}$ . On suppose les  $a_i$  distincts. Montrer que  $V$  est inversible.
  - Soient  $A \in \mathcal{M}_p(\mathbb{C})$  et  $D = \{z \in \mathbb{C}, |z| < 1\}$ . On suppose que  $A^n \rightarrow 0$ . Montrer que le spectre de  $A$  est inclus dans  $D$ .
  - Soient  $a, b, c$  dans  $\mathbb{C}$  distincts. On suppose que  $a^n + b^n + c^n \rightarrow 0$ . Montrer que  $a, b$  et  $c$  sont dans  $D$ .
  - Montrer que si  $\text{tr}(A^n) \rightarrow 0$  alors le spectre de  $A$  est inclus dans  $D$ .

SOLUTION. —

- La première et la dernière ligne de  $M$  étant proportionnelles, on sait déjà que  $0 \in \text{Spec}(M)$ . On exécute les lignes suivantes :

```
import numpy as np
import numpy.linalg as alg

M = np.array([[1/3, 1/3, 1/3], [1/4, 1/3, 1/6], [1/4, 1/4, 1/4]])
for n in range(30):
    P = alg.matrix_power(M, n)
    print(P)
    print([n, alg.eigvals(P)])
```

Les valeurs propres de  $M$  sont données par  $[-5.55111512e-17, 8.33333333e-01, 8.33333333e-02]$ , et on interprète cela comme

$$\text{Spec}(M) = \left\{ 0, \frac{1}{10} \left( 8 + \frac{1}{3} \right), \frac{1}{100} \left( 8 + \frac{1}{3} \right) \right\} = \left\{ 0, \frac{5}{6}, \frac{1}{12} \right\},$$

ce que confirmerait un calcul exact du polynôme caractéristique :

$$\chi_M = X \left( X^2 - \frac{11}{12}X + \frac{5}{72} \right).$$

Ces valeurs propres, disons  $\lambda_1, \lambda_2, \lambda_3$ , sont de module strictement plus petit que 1, donc les valeurs propres de  $M^n$ , qui sont  $\lambda_1^n, \lambda_2^n, \lambda_3^n$ , tendent vers zéro quand  $n$  tend vers l'infini. On le constate plus ou moins avec Python, car les problèmes d'instabilité numérique faussent les résultats.

- C'est un résultat qui figure explicitement dans le cours puisque la valeur du déterminant de Vandermonde est

$$\det(V) = \det \left( (a_i^{j-1})_{1 \leq i, j \leq n} \right) = \prod_{1 \leq i < j \leq n} (a_j - a_i).$$

Si les  $a_i$  sont deux à deux distincts, ce déterminant est non nul, donc  $V$  est inversible.

3. On note  $\lambda_1, \dots, \lambda_p$  les valeurs propres complexes de  $A$  répétées avec leur ordre de multiplicité. Comme  $A$  est trigonalisable dans  $\mathbb{C}$ , il existe une matrice inversible  $P \in GL_p(\mathbb{C})$  et une matrice triangulaire supérieure  $T \in \mathcal{T}_p^{\text{sup}}(\mathbb{C})$  telles que  $T = P^{-1}AP$ . Les éléments diagonaux de  $T$  sont les valeurs propres  $\lambda_1, \dots, \lambda_p$  de  $A$ , et un calcul très classique montre que les termes diagonaux de  $T^n$  sont  $\lambda_1^n, \dots, \lambda_p^n$ .

Comme  $T = P^{-1}AP$ , on obtient

$$\forall n \in \mathbb{N}, \quad T^n = P^{-1}A^nP.$$

L'application  $\varphi: N \mapsto P^{-1}NP$  est un endomorphisme de l'espace vectoriel de dimension finie  $\mathcal{M}_p(\mathbb{C})$ , donc est continue. Comme la suite  $(A^n)_{n \in \mathbb{N}}$  converge vers zéro, il en est de même de la suite image  $(\varphi(A^n))_{n \in \mathbb{N}} = (T^n)_{n \in \mathbb{N}}$ . Pour traduire cette limite, le cours dit qu'on peut choisir n'importe quelle norme sur  $\mathcal{M}_p(\mathbb{C})$ , par exemple la norme infinie  $\|M\|_\infty = \max_{1 \leq i, j \leq p} |m_{i,j}|$ . La convergence de la suite  $(T^n)$  vers zéro entraîne que chaque suite d'éléments de  $T^n$  d'une position  $(i, j)$  fixée converge vers zéro, et c'est en particulier le cas pour ses termes diagonaux :

$$\forall k \in \llbracket 1, p \rrbracket, \quad \lim_{n \rightarrow +\infty} \lambda_k^n = 0.$$

Or la suite géométrique complexe de terme général  $\lambda^n$  converge vers zéro si et seulement si  $|\lambda| < 1$ . On en déduit que

$$\text{Spec}(A) \subset D.$$

4. On pose  $\forall n \in \mathbb{N}, u_n = a^n + b^n + c^n$ , et on note  $X_n$  et  $U_n$  les vecteurs colonnes suivants, et  $V$  la matrice de Vandermonde du triplet  $(a, b, c)$  :

$$\forall n \in \mathbb{N}, \quad X_n = \begin{pmatrix} a^n \\ b^n \\ c^n \end{pmatrix}, \quad U_n = \begin{pmatrix} u_n \\ u_{n+1} \\ u_{n+2} \end{pmatrix}, \quad V = \begin{pmatrix} 1 & 1 & 1 \\ a & b & c \\ a^2 & b^2 & c^2 \end{pmatrix}.$$

On peut alors écrire que

$$\forall n \in \mathbb{N}, \quad \begin{cases} a^n + b^n + c^n = u_n \\ a^{n+1} + b^{n+1} + c^{n+1} = u_{n+1} \\ a^{n+2} + b^{n+2} + c^{n+2} = u_{n+2} \end{cases} \quad \text{donc} \quad VX_n = U_n.$$

Comme  $V$  est inversible, on en déduit que  $\forall n \in \mathbb{N}, X_n = V^{-1}U_n$ . Comme  $(u_n)$  converge vers zéro par hypothèse, il en est de même de la suite de vecteurs  $(U_n)_{n \in \mathbb{N}}$ , et comme la matrice  $V^{-1}$  est indépendante de  $n$  et que l'application  $U \in \mathcal{M}_{1,3}(\mathbb{C}) \mapsto V^{-1}U$  est continue, on en déduit que la suite  $(V^{-1}U_n)_{n \in \mathbb{N}} = (X_n)_{n \in \mathbb{N}}$  converge vers zéro, donc que les trois suites de termes généraux  $a^n, b^n$  et  $c^n$  convergent vers zéro, donc que

$$(a, b, c) \in D^3.$$

5. Il faut adapter légèrement la question précédente, en tenant compte du fait que les valeurs propres peuvent être multiples. On modifie alors les notations de la question (c) : on désigne par  $\lambda_1, \dots, \lambda_r$  les valeurs propres complexes *deux à deux distinctes* de  $A$ , et par  $m_1, \dots, m_r$  leurs multiplicités respectives. L'hypothèse est donc que

$$\text{tr}(A^n) = m_1\lambda_1^n + \dots + m_r\lambda_r^n \xrightarrow{n \rightarrow +\infty} 0.$$

On note  $X_n$  (respectivement  $U_n$ ) le vecteur colonne de composantes  $m_k\lambda_k^n$  (respectivement  $\text{tr}(A^{n+k-1})$ ) pour  $k \in \llbracket 1, r \rrbracket$ . Si  $V$  est la matrice de Vandermonde associée au  $r$ -uplet  $(\lambda_1, \dots, \lambda_r)$ , on a encore

$$\forall n \in \mathbb{N}, \quad VX_n = U_n,$$

Comme  $V$  est inversible, on conclut comme précédemment que

$$\text{Spec}(A) \subset D.$$

#### Exercice 4

Soient, si  $(a_{n-1}, \dots, a_0) \in \mathbb{R}^n$ ,  $P = X^n + a_{n-1}X^{n-1} - \dots - a_1X - a_0$  et

$$M = \begin{pmatrix} 0 & \cdots & \cdots & 0 & a_0 \\ 1 & 0 & & \vdots & a_1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & a_{a+2} \\ 0 & \cdots & 0 & 1 & a_{n-1} \end{pmatrix}.$$

1. écrire une fonction `Matrice(L)` qui pour une liste  $[a_{n-1}, \dots, a_0]$  renvoie la matrice  $M$ .
2. Calculer le polynôme caractéristique de `Matrice(L)` pour des listes de 8 entiers aléatoires compris entre 0 et 9. Conjecture ?
3. Calculer  $\chi_M$ .
4. À quelle condition la matrice  $M$  est-elle diagonalisable ?

SOLUTION. —

1. On utilise la bibliothèque `numpy`, qui contient les commandes élémentaires relatives à la construction des matrices et aux opérations matricielles :

```
import numpy as np

def Matrice(L):
    n = len(L)
    M = np.zeros((n, n))
    print(M)
    for k in range(n - 1):
        M[k + 1, k] = 1
    for k in range(n):
        M[k, n - 1] = L[n - k - 1]
    return(M)
```

2. On commence par écrire une fonction calculant le polynôme caractéristique d'une matrice carrée, qui utilise la commande `eigvals` de la bibliothèque `numpy.linalg` et la commande `Polynomial` de la bibliothèque `numpy.polynomial` : cette fonction, appelée `charpoly` est une hérésie, puisqu'elle consiste à calculer les valeurs propres  $\lambda_i$  répétées avec multiplicité (opération hautement non linéaire) pour fabriquer ensuite le polynôme caractéristique comme le produit des  $X - \lambda_i$  (alors que le calcul direct de  $\chi_M$  est de nature multilinéaire). Si quelqu'un a une meilleure idée...

```
import numpy.linalg as alg
import numpy.polynomial as pol

def charpoly(M):
    Sp = alg.eigvals(M)
    P = pol.Polynomial([1])
    for alpha in Sp:
        P = P*pol.Polynomial([1, - alpha])
    return(P)
```

On calcule le polynôme caractéristique de `Matrice(L)` pour 5 listes de 8 entiers aléatoires compris entre 0 et 9 après avoir chargé la bibliothèque `numpy.random` de la façon suivante :

```
import numpy.random as rd

for n in range(5):
    L = rd.randint(0, 10, 8)
    M = Matrice(L)
    P = charpoly(M)
    print(L)
    print(P)
```

Par exemple, pour la liste

$[5, 9, 5, 3, 0, 1, 8, 9]$ ,

on obtient un résultat peu lisible qui, une fois convenu que les nombres de valeur absolue plus petite que  $10^{-15}$  sont nuls, donne le polynôme

$$X^8 - 5X^7 - 9X^6 - 5X^5 - 3X^4 - X^2 - 8X - 9.$$

On conjecture donc que  $\chi_M = X^a + \sum_{k=0}^{n-1} a_k X^k$  si  $L = [a_{n-1}, \dots, a_0]$ .

3. On pose

$$P = X^a + \sum_{k=0}^{n-1} a_k X^k = X^a + a_{n-1} X^{n-1} - \dots - a_1 X - a_0.$$

Dans le déterminant  $\chi_M = \det(XI_n - M)$ , on effectue la transformation suivante sur les lignes :  $L_1 \leftarrow L_1 + \sum_{k=2}^n X^{k-1} L_k$ . On obtient

$$\chi_M = \begin{vmatrix} X & \cdots & \cdots & 0 & -a_0 \\ -1 & \ddots & \vdots & 0 & -a_1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & X & -a_{a+2} \\ 0 & \cdots & 0 & -1 & X - a_{n-1} \end{vmatrix} = \begin{vmatrix} 0 & \cdots & \cdots & 0 & P \\ -1 & \ddots & \vdots & 0 & -a_1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & -a_{a+2} \\ 0 & \cdots & 0 & -1 & X - a_{n-1} \end{vmatrix}.$$

Le développement du dernier déterminant par rapport à la première ligne donne

$$\chi_M = (-1)^{n+1} P \begin{vmatrix} -1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 \end{vmatrix} = (-1)^{n+1} (-1)^{n-1} P = P.$$

4. Soient  $\lambda \in \mathbb{R}$  et  $(E_k)_{1 \leq k \leq n}$  la base canonique de l'espace  $\mathcal{M}_{n,1}(\mathbb{R})$  des colonnes. Les  $n-1$  premières colonnes de la matrice

$$M - \lambda I_n = \begin{pmatrix} -\lambda & \cdots & \cdots & 0 & a_0 \\ 1 & -\lambda & & \vdots & a_1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & -\lambda & a_{a+2} \\ 0 & \cdots & 0 & 1 & a_{n-1} - \lambda \end{pmatrix}$$

sont échelonnées sur la base  $\mathcal{M}_{n,1}(\mathbb{R})$  de l'espace des colonnes, donc forment une famille libre. Par suite,  $\text{rg}(M - \lambda I_n) \geq n-1$ , donc  $\dim \ker(M - \lambda I_n) \leq 1$ . En particulier, si  $\lambda$  est une valeur propre de  $M$ , la dimension du sous-espace propre associé vaut exactement 1 :

$$\forall \lambda \in \text{Spec}(M), \quad \dim E_\lambda(M) = 1.$$

En appliquant l'équivalence du cours «  $M$  est diagonalisable sur  $\mathbb{K}$  si et seulement si  $\chi_M$  est scindé sur  $\mathbb{K}$  et pour toute valeur propre de  $M$ , la dimension du sous-espace propre associé est égal à la multiplicité de la valeur propre dans  $\chi_M$  », on obtient

$$M \text{ est diagonalisable si et seulement si } P = X^a + \sum_{k=0}^{n-1} a_k X^k \text{ est scindé à racines simples.}$$

### Exercice 5

Si  $P, Q \in \mathbb{R}[X]$ , on pose  $\langle P, Q \rangle = \int_{-\infty}^{+\infty} P(t)Q(t)e^{-t^2} dt$ . Si  $n \in \mathbb{N}$ , on pose  $I_n = \int_{-\infty}^{+\infty} t^n e^{-t^2} dt$ .

1. Montrer que ces intégrales sont bien définies et que  $\langle \cdot, \cdot \rangle$  est un produit scalaire.
2. Exprimer  $I_n$  en fonction de  $I_{n+2}$ .
3. écrire un programme récursif permettant de calculer  $I_n$ .
4. écrire un programme permettant de calculer le produit scalaire de deux polynômes  $P$  et  $Q$ .

SOLUTION. —

1. Si  $P$  et  $Q$  sont deux polynômes, la fonction  $f: t \mapsto P(t)Q(t)e^{-t^2}$  est continue sur  $[0, +\infty[$ , et les résultats de croissance comparée montrent que  $f(t) = o(\frac{1}{t^2})$  au voisinage de l'infini, donc que  $f$  est intégrable sur  $\mathbb{R}$ , donc que  $\langle P, Q \rangle$  est une intégrale convergente.

La bilinéarité de  $\langle \cdot, \cdot \rangle$  résulte de la linéarité de l'intégrale, de la commutativité du produit dans  $\mathbb{R}$ , et son caractère défini positif vient de la positivité stricte de l'intégrale des fonctions continues.

2. On remarque que  $t \mapsto t^n e^{-t^2}$  possède la même parité que  $n$ , donc que  $I_n = 0$  pour tout  $n$  impair et que  $I_n = 2 \int_0^{+\infty} t^n e^{-t^2} dt$  pour tout  $n$  pair.

Une intégration par parties, pratiquée d'abord sur un segment  $[0, m]$  dont on fait ensuite tendre la borne  $m$  vers l'infini, prouve que, pour tout entier  $n \geq 2$  pair :

$$I_n = 2 \int_0^{+\infty} t^{n-1} \times t e^{-t^2} dt = \left[ -t^{n-1} e^{-t^2} \right]_0^{+\infty} + (n-1) \int_0^{+\infty} t^{n-2} e^{-t^2} dt = \frac{n-1}{2} I_{n-2}.$$

Cette relation reste vraie pour les entiers impairs, puisqu'elle s'écrit  $0 = 0$ .

3. On choisit d'écrire une fonction récursive. Elle ne nécessite l'usage d'aucun module particulier :

```
def Calcul(n):
    if n%2 != 0:
        return(0)
    else:
        return(CalculPairs(n))

def CalculPairs(n)
    if n == 0:
        return(sqrt(pi))
    else:
        return((n - 1)*CalculPairs(n - 2)/2)
```

4. On utilise ici le module `numpy.polynomial`. On suppose que les variables  $P$  et  $Q$  sont des objets de type `polynomial`, ce qui permet de les multiplier par l'opérateur `*` et d'obtenir le degré de  $P$  par `P.degree()` et son coefficient de degré  $k$  par `P.coef[k]`, puisque `P.coef` est le tableau de ses coefficients, classés par ordre croissant.

```
import numpy.polynomial as pol

def ps(P, Q):
    R = P*Q
    s = 0
    for k in range(1 + R.degree()):
        s += R.coef[k]*I(k)
    return(s)
```

### Exercice 6

Soient  $n \in \mathbb{N}$  et, pour  $t \in \mathbb{R}$  et  $k \in \llbracket 0, n \rrbracket$ ,  $B_{n,k}(t) = \binom{n}{k} t^k (1-t)^{n-k}$ . Soient  $\sigma = (P_k)_{0 \leq k \leq n}$  une suite d'éléments de  $\mathbb{R}^2$  et  $f$  la fonction vectorielle définie par  $f(t) = \sum_{k=0}^n B_{n,k}(t) P_k$ . On dit que la courbe paramétrée par  $f$  est la *courbe de Bézier* associée à  $\sigma$ .

- Cas  $n = 3$ . On donne  $P_0 = (-2, 2)$ ,  $P_1 = (1, 5)$ ,  $P_2 = (5, 3)$  et  $P_3 = (3, 1)$ .
  - Calculer  $f(t)$  pour  $t \in [0, 1]$ .
  - Tracer sur le même graphe la courbe paramétrée par  $f$  et la ligne polygonale joignant  $P_0, P_1, P_2, P_3$ .
- Cas général. On suppose  $n \geq 2$ .
  - Calculer  $f(0)$ ,  $f(1)$ ,  $f'(0)$  et  $f'(1)$  en fonction des points  $P_k$ .
  - Montrer que la famille  $(B_{n,k})_{k \in \llbracket 0, n \rrbracket}$  forme une base de  $\mathbb{R}_n[X]$ . En déduire que toute fonction vectorielle polynomiale est une courbe de Bézier.
  - Soit  $g(t) = (-t^3 + 2t + 1, t^3 + 5t^2 - 2)$ . À quelle suite  $\sigma$  la courbe paramétrée par  $g$  est-elle associée ?
- Tracer sur un même graphe les courbes représentatives sur  $[0, 1]$  des fonctions  $B_{7,k}$  pour  $k \in \llbracket 1, 6 \rrbracket$ . Pour quelles valeurs de  $k$  la norme  $\|B_{7,k}\|_\infty$  est-elle maximale ? Conjecturer le résultat pour  $n$  entier quelconque. Que vaut  $\alpha_n = \max_{1 \leq k \leq n-1} \|B_{n,k}\|_\infty$  ?

SOLUTION. —

1. On peut bien sûr programmer une classe polynôme, etc. Comme  $n = 3$  ici, le parti pris dans ce qui suit est de manipuler les polynômes comme des listes de taille 4 (leurs coefficients).

Cas  $n = 3$ . On donne  $P_0 = (-2, 2)$ ,  $P_1 = (1, 5)$ ,  $P_2 = (5, 3)$  et  $P_3 = (3, 1)$ .

a)

```
B0 = [1, - 3, 3, -1]
B1 = [0, 3, -6, 3]
B2 = [0, 0, 3, - 3]
B3 = [0, 0, 0, 1]
X = [-2, 1, 5, 3]
Y = [2, 5, 3, 1]
PX = [0, 0, 0, 0]
for k in range(4):
    PX[k] = X[0]*B0[k]+X[1]*B1[k]+X[2]*B2[k]+X[3]*B3[k]
PX
>>> [-2, 9, 3, -7]
PY = [0, 0, 0, 0]
for k in range(4):
    PY[k] = Y[0]*B0[k]+Y[1]*B1[k]+Y[2]*B2[k]+Y[3]*B3[k]
PY
>>> [2, 9, -15, 5]
```

$f$  est donc la fonction polynomiale  $t \mapsto (-2 + 9t + 3t^2 - 7t^3, 2 + 9t - 15t^2 + 5t^3)$ .

b)

```
import numpy as np
T = np.linspace(0, 1)
X1 = []
Y1 = []
for k in range(len(T)):
    X1.append(PX[0]+T[k]*PX[1]+T[k]**2*PX[2]+T[k]**3*PX[3])
    Y1.append(PY[0]+T[k]*PY[1]+T[k]**2*PY[2]+T[k]**3*PY[3])
import matplotlib.pyplot as plt
plt.plot(X, Y)
plt.plot(X1, Y1)
plt.show()
```

2. (a) Comme  $B_{n,k}(0) = 0$  sauf pour  $k = 0$ , il vient  $f(0) = B_{n,0}(0)P_0 = P_0$ . De même  $B'_{n,k}(0) = 0$  pour  $k \geq 2$ , et  $B'_{n,0}(0) = -n$  et  $B'_{n,1}(0) = n$ , donc  $f'(0) = -nP_0 + nP_1$ . (Donc  $\overrightarrow{P_0P_1}$  dirige la tangente en  $P_0 = f(0)$ .)

Symétriquement, on obtient  $f(1) = P_n$  et  $f'(1) = -nP_{n-1} + nP_n$ .

- (b) La famille  $(B_{n,k})_{k \in [0,n]}$  est étagée en valuations (seul  $B_{n,0}$  a un coefficient constant, seuls  $B_{n,0}$  et  $B_{n,1}$  ont un terme de degré 1, etc), ce qui fait que la matrice de la famille dans la base canonique est « anti-triangulaire », en particulier échelonnée donc inversible. C'est donc une matrice de passage et les  $B_{n,k}$  forment donc une base de  $\mathbb{R}_n[X]$ .

On en déduit que toute fonction polynomiale scalaire de degré  $n$  s'écrit  $t \mapsto \sum_{k=0}^n \alpha_k B_{n,k}(t)$ . Si  $f$  est une fonction vectorielle polynomiale, de la forme  $t \mapsto (f_1(t), f_2(t))$  avec  $f_1, f_2$  de degrés au plus  $n$ , alors il existe une (unique) famille de scalaires  $(\alpha_0, \dots, \alpha_n, \beta_0, \dots, \beta_n)$  telle que  $f$  s'écrive  $t \mapsto \left( \sum_{k=0}^n \alpha_k B_{n,k}(t), \sum_{k=0}^n \beta_k B_{n,k}(t) \right) = \sum_{k=0}^n B_{n,k}(t) P_k$  où  $P_k = (\alpha_k, \beta_k)$ , donc  $f$  est donc une courbe de Bézier.

- (c) On a  $B_{3,0} = 1 - 3X + 3X^2 - X^3$ ,  $B_{3,1} = 3X - 6X^2 + 3X^3$ ,  $B_{3,2} = 3X^2 - 3X^3$  et  $B_{3,3} = X^3$ .

Alors  $1 + 2X - X^3 = B_{3,0} + 5X - 3X^2 = B_{3,0} + \frac{5}{3}B_{3,1} + 7X^2 - 5X^3 = B_{3,0} + \frac{5}{3}B_{3,1} + \frac{7}{3}B_{3,2} + 2X^3 = B_{3,0} + \frac{5}{3}B_{3,1} + \frac{7}{3}B_{3,2} + 2B_{3,3}$ .

De même  $-2 + 5X^2 + X^3 = -2B_{3,0} - 6X + 11X^2 - X^3 = -2B_{3,0} - 2B_{3,1} - X^2 + 5X^3 = -2B_{3,0} - 2B_{3,1} - \frac{1}{3}B_{3,2} + 4X^3 = -2B_{3,0} - 2B_{3,1} - \frac{1}{3}B_{3,2} + 4B_{3,3}$ .

Il vient  $g(t) = B_{3,0}(t)(1, -2) + B_{3,1}(t)(\frac{5}{3}, -2) + B_{3,2}(t)(\frac{7}{3}, -\frac{1}{3}) + B_{3,3}(t)(2, 4)$ , d'où  $\sigma$ .

3. On peut se contenter des courbes des fonctions  $B_{7,k}$  pour  $k \in [1, 3]$ , les trois autres s'en déduisent par symétrie par rapport à la droite d'équation  $x = \frac{1}{2}$ .

```

f = lambda x: 7*x*(1 - x)**6
g = lambda x: 21*x**2*(1 - x)**5
h = lambda x: 35*x**3*(1 - x)**4
plt.plot(T, f(T))
plt.plot(T, g(T))
plt.plot(T, h(T))
plt.show()

```

Le tracé indique que  $\|B_{7,k}\|_\infty$  est maximale pour  $k = 1$  (donc pour  $k = 6$ ).

Pour  $n$  quelconque, on conjecture que  $\alpha_n = \max_{1 \leq k \leq n-1} \|B_{n,k}\|_\infty = \|B_{n,1}\|_\infty = \|B_{n,n-1}\|_\infty$ .

L'étude des variations de la fonction  $B_{n,1}$  sur  $[0, 1]$  montre qu'elle est positive, croissante sur  $[0, 1/n]$  et décroissante ensuite. Donc  $\|B_{n,1}\|_\infty = B_{n,1}(1/n) = (1 - 1/n)^{n-1} \xrightarrow{+\infty} 1/e$ .

Pour  $k$  quelconque, on a de même par l'étude des variations  $\|B_{n,k}\|_\infty = B_{n,k}(k/n) = \binom{n}{k} (k/n)^k (1 - k/n)^{n-k}$ .  
Donc  $\alpha_n = \max_{1 \leq k \leq n-1} \binom{n}{k} (k/n)^k (1 - k/n)^{n-k}$

Pour  $k \in \{1, \dots, n-2\}$ , on a après simplification

$$\frac{\|B_{n,k+1}\|_\infty}{\|B_{n,k}\|_\infty} = \left(\frac{k+1}{k}\right)^k \left(\frac{n-k-1}{n-k}\right)^{n-k-1} = e^{k \ln(1+1/k) - (n-k-1) \ln(1+1/(n-k-1))}.$$

Or la fonction  $g : x \mapsto \frac{1}{x} \ln(1+x)$  est décroissante sur  $]0, 1]$  donc, lorsque  $k \leq n-k-1$ , soit  $k \leq \frac{n-1}{2}$ , on a  $1/k \geq 1/(n-k-1)$  donc  $g(1/k) \leq g(1/(n-k-1))$  soit  $k \ln(1+1/k) - (n-k-1) \ln(1+1/(n-k-1)) \leq 0$  et finalement  $\frac{\|B_{n,k+1}\|_\infty}{\|B_{n,k}\|_\infty} \leq 1$ .

La suite des valeurs  $\|B_{n,k}\|_\infty$ , pour  $1 \leq k \leq \lfloor n/2 \rfloor$ , est décroissante, et par symétrie elle est croissante pour  $\lfloor n/2 \rfloor + 1 \leq k \leq n-1$ . La valeur maximale est donc bien  $\|B_{n,1}\|_\infty = \|B_{n,n-1}\|_\infty$  soit  $\alpha_n = (1 - 1/n)^{n-1}$ .

### Exercice 7

Soit  $S = \{(u_n)_{n \in \mathbb{N}}, \forall n \in \mathbb{N}^*, u_{n+1} = \frac{u_n^2}{n+1}\}$ . On désigne par  $u_n(x)$  le  $n$ -ième terme de la suite de  $S$  telle que  $u_0 = x$ .

1. écrire une fonction `Suite(n, x)` qui renvoie  $u_n(x)$ . Tester la fonction pour quelques valeurs. Tracer les premiers termes de la suite pour différentes valeurs de  $x$ . Commenter.
2. Tester pour  $n = 31$  et  $x = 1,6616$ , pour  $n = 17$  et  $x = 1,6617$ . Commenter.
3. Montrer l'équivalence des trois assertions suivantes :
  - (i)  $\exists n \in \mathbb{N}, u_{n+1} \leq u_n$
  - (ii)  $\exists n \in \mathbb{N}^*, u_n < 1$
  - (iii)  $(u_n)$  tend vers 0.
4. Montrer que  $\exists N \in \mathbb{N}, u_N \geq N+2 \Rightarrow \forall n \geq N, u_n \geq n+2$ .
5. étudier les cas  $x = 1$  et  $x = 2$ . On pose  $E_0 = \{x, u_n(x) \rightarrow 0\}$  et  $E_\infty = \{x, u_n(x) \rightarrow +\infty\}$ . Montrer que  $E_0$  et  $E_\infty$  sont deux intervalles tels que  $\mathbb{R}_+^* = E_0 \cup E_\infty$ .

SOLUTION. —

```

1 def Suite(n, x):
    y = x
    for i in range(1, n + 1):
        y = (y**2)/i
    return y
import matplotlib.pyplot as plt
plt.plot([Suite(n, 1.6616) for n in range(32)])
plt.show()

```

2. Python renvoie 0.0 dans le premier cas, 49.02934290985918 dans le second, ce qui montre la sensibilité à de petites perturbations...
3. Supposons (i), soit donc  $n$  tel que  $u_{n+1} \leq u_n$ . Nécessairement  $u_{n+1} \geq 0$  donc  $u_{n+1}^2 \leq u_n^2$ , donc  $u_{n+2} = \frac{u_{n+1}^2}{n+2} \leq \frac{u_n^2}{n+2} \leq \frac{u_n^2}{n+1} = u_{n+1}$ . Par récurrence, on montre que la suite  $(u_k)_{k \geq n}$  est alors décroissante. En particulier, elle est majorée par son premier terme  $u_n$ , et en considérant un entier  $k \geq n$  tel que  $\frac{u_n^2}{k+1} < 1$ , qui existe, on a alors  $u_{k+1} = \frac{u_k^2}{k+1} \leq \frac{u_n^2}{k+1} < 1$ , d'où (ii).

Supposons (ii) et soit  $n \in \mathbb{N}^*$  tel que  $u_n < 1$ . Alors une récurrence élémentaire montre que  $\forall k \geq n, 0 \leq u_k < 1$  donc  $0 \leq u_{k+1} < \frac{1}{k+1}$ . Par le théorème d'encadrement, on a donc (iii).

Supposons enfin (iii). La suite est à valeurs positives (à partir du rang 1) et de limite nulle, elle ne peut donc être strictement croissante, d'où (i).

4. Supposons trouvé  $N \in \mathbb{N}$  tel que  $u_N \geq N + 2$ . Soit, pour  $n \geq N$ ,  $\mathcal{P}(n)$  la propriété  $u_n \geq n + 2$ .  
 —  $\mathcal{P}(N)$  est l'hypothèse faite ici.  
 — Soit  $n \geq N$ , supposons  $\mathcal{P}(n)$ . Alors  $u_n^2 \geq (n+2)^2$  donc  $u_{n+1} \geq \frac{(n+2)^2}{n+1} \geq n+3$ , en effet  $(n+2)^2 = n^2 + 4n + 4 \geq n^2 + 4n + 3 = (n+1)(n+3)$ , soit  $\mathcal{P}(n+1)$ .

On conclut par récurrence que  $\forall n \geq N, u_n \geq n + 2$ .

5. Pour  $x = 1$ , on a  $u_0 = 1 = u_1$  et  $u_2 = \frac{1}{2} \leq u_1$  donc  $u$  est de limite nulle, soit  $1 \in E_0$ .  
 Pour  $x = 2$ , on a  $u_0 = 2, u_1 = 4 \geq 3$  donc  $\forall n \geq 1, u_n \geq n + 2$  donc  $u$  est de limite  $+\infty$ , soit  $2 \in E_\infty$ .  
 La question c montre que pour tout  $x$ , on a soit  $(u_n(x))$  strictement croissante (et positive), soit il existe  $n \in \mathbb{N}$  tel que  $u_{n+1}(x) \leq u_n(x)$  et alors  $(u_n(x))$  est de limite nulle. Dans tous les cas,  $(u_n(x))$  admet une limite dans  $\mathbb{R}_+ \cup \{+\infty\}$ .

Or si  $u_n(x) \xrightarrow{+\infty} \ell \in \mathbb{R}_+$ , alors  $\frac{u_n(x)^2}{n+1} \xrightarrow{+\infty} 0$  soit  $u_{n+1}(x) \xrightarrow{+\infty} 0$  donc  $\ell = 0$ . Donc pour tout  $x > 0$ ,  $(u_n(x))$  admet une limite égale à 0 ou à  $+\infty$ , c'est-à-dire que  $\mathbb{R}_+^* = E_0 \cup E_\infty$ .

Enfin pour  $0 < x \leq y$ , on a par une récurrence immédiate  $\forall n \in \mathbb{N}, u_n(x) \leq u_n(y)$  donc  $\lim_{n \rightarrow +\infty} u_n(x) \leq \lim_{n \rightarrow +\infty} u_n(y)$ . Donc si  $y \in E_0$  alors  $x \in E_0$ . Finalement  $\forall y \in E_0, [0, y] \subset E_0$ .  $E_0$  est donc convexe donc c'est un intervalle.

De même si  $x \in E_\infty$  alors  $y \in E_\infty$ . Finalement  $\forall x \in E_\infty, [x, +\infty[ \subset E_\infty$ .  $E_\infty$  est donc convexe : c'est un intervalle.

### Exercice 8

Soit  $(E_n)$  l'équation  $x^n = e^x$ , pour  $n \in \mathbb{N}$ .

1. a) Montrer que  $(E_n)$  possède une unique solution  $x_n$  dans  $[0, n]$ . **Correction : il faut supposer  $n \geq 3$ .**  
 b) Calculer quelques termes et conjecturer le comportement de  $(x_n)$ .  
 c) Conjecturer que  $x_n = 1 + \frac{a}{n} + \frac{b}{n^2} + o(\frac{1}{n^2})$ .  
 d) Donner un développement asymptotique à trois termes de  $x_n$ .
2. Montrer que  $(E_n)$  possède une unique solution  $y_n \in [n, +\infty[$ . Déterminer un développement asymptotique de  $y_n$ .

SOLUTION. — Pour  $n = 0$ , l'équation  $1 = e^x$  possède une seule solution. On suppose  $n \geq 1$ .

1. (a) L'équation  $(E_n)$  est équivalente à  $f(x) = \frac{1}{n}$ , où

$$f: x \in \mathbb{R}_+^* \mapsto \frac{\ln x}{x}.$$

Une étude rapide de  $f$  [avec  $f'(x) = \frac{1-\ln x}{x^2}$ ] donne le tableau de variation suivant :

|         |           |            |               |              |
|---------|-----------|------------|---------------|--------------|
| $x$     | 0         | 1          | $e$           | $+\infty$    |
| $f'(x)$ |           | +          | 0             | −            |
| $f(x)$  | $-\infty$ | $\nearrow$ | $\frac{1}{e}$ | $\searrow$ 0 |

Pour  $n = 1$  ou  $2$ , l'équation  $x^n = e^x$  n'a donc pas de solution. En revanche, si  $n \geq 3$ , de sorte que  $\frac{1}{n} < \frac{1}{e}$ , l'équation  $f(x) = \frac{1}{n}$  possède deux solutions  $x_n$  et  $y_n$  vérifiant l'encadrement :

$$1 < x_n < e < y_n.$$

On suppose  $n \geq 3$  jusqu'à la fin de l'exercice.

- (b) On programme une méthode de dichotomie avec précision fixée, qui calcule une racine de  $h$  sur  $[a, b]$  lorsqu'on sait que  $h$  est continue et vérifie  $h(a)h(b) < 0$ . Si  $h$  est, de plus, strictement monotone, cette méthode donne l'unique racine de  $h$  sur  $[a, b]$  (c'est le cas ici de  $f_n$  sur  $[1, e]$ ).

```
def dichotomie(h, a, b, precision):
    while b - a > precision:
        m = (a + b)/2
        if h(a)*h(m) <= 0:
            b = m
        else:
            a = m
    return a
```

On définit aussi  $f$  ainsi que la fonction `petiteracine(n, precision)` qui renverra  $x_n$  grâce à une définition locale de la fonction  $f - \frac{1}{n}$ .

```
def f(x):
    return(log(x)/x)
def petiteracine(n, precision):
    def h(x):
        return(f(x) - 1/n)
    return(dichotomie(h, 1, exp(1), precision))
```

Voici les résultats obtenus, à la précision choisie de  $10^{-3}$  (et par défaut) :

| $n$   | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 100   |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $x_n$ | 1.856 | 1.429 | 1.294 | 1.226 | 1.183 | 1.154 | 1.133 | 1.118 | 1.010 |

On conjecture que la suite  $(x_n)$  décroît et converge vers 1.

- (c) Le calcul de  $n(x_n - 1)$  pour  $n = 1000, 2000, \dots, 10000$ , grâce à

```
for k in range(1, 11):
    print(1000*k*(petiteracine(1000*k, 10**(-6)) - 1))
```

renvoie des nombres compris entre 1,001 et 0,995 : on ne se mouille pas trop en conjecturant que  $a = 1$ .  
Le calcul de  $n^2(x_n - 1 - \frac{1}{n})$  pour  $n = 1000, 2000, \dots, 10000$ , grâce à

```
for k in range(1, 11):
    print(((1000*k)**2*(petiteracine(1000*k, 10**(-10)) - 1 - 1/(1000*k))))
```

renvoie des nombres compris entre 1,502 et 1,496 : on ne se mouille pas trop en conjecturant que  $b = \frac{3}{2}$ .

- (d) On note  $g$  la restriction de  $f$  à  $]0, e[$ . La fonction  $g$  est de classe  $\mathcal{C}^\infty$  et est strictement monotone. De plus,  $g'(x) = f'(x) = \frac{1-\ln x}{x^2} \neq 0$  pour tout  $x \in ]0, e[$ , donc  $g$  réalise un  $\mathcal{C}^\infty$  difféomorphisme sur son image, qui est  $] -\infty, \frac{1}{e}[$ . Comme  $g^{-1}$  est de classe  $\mathcal{C}^\infty$  au voisinage de zéro, la suite de terme général  $x_n = g^{-1}(\frac{1}{n})$  admet un développement asymptotique à tout ordre en la variable  $\frac{1}{n}$ , donné par la formule de Taylor-Young pour  $g^{-1}$  en zéro. En particulier

$$x_n = g^{-1}(0) + \frac{(g^{-1})'(0)}{n} + \frac{(g^{-1})''(0)}{2n^2} + o\left(\frac{1}{n^2}\right) = 1 + \frac{a}{n} + \frac{b}{n^2} + o\left(\frac{1}{n^2}\right),$$

avec  $a = (g^{-1})'(0)$  et  $b = (g^{-1})''(0)/2$ . Or  $(g^{-1})'(x) = \frac{1}{g'(g^{-1}(x))} = \frac{1}{f'(g^{-1}(x))}$ , donc

$$a = \frac{1}{f'(1)} = 1.$$

On calcule ensuite  $f''(x) = -\frac{1}{x^3} - 2\frac{1-\ln x}{x^3} = \frac{2\ln x - 3}{x^3}$  et  $(g^{-1})''(x) = -\frac{f''(g^{-1}(x))}{(f'(g^{-1}(x)))^3}$ , donc

$$b = -\frac{f''(1)}{2(f'(1))^3} = \frac{3}{2}.$$

2. Voir la question (a)i pour l'existence et l'unicité de l'unique solution  $y_n \in [n, +\infty[$  de  $(E_n)$ . Comme  $f(n) = \frac{\ln n}{n} > \frac{1}{n}$  et  $f(n^2) = \frac{2\ln n}{n^2} < \frac{1}{n}$ , on peut affirmer que

$$n < y_n < n^2,$$

On écrit ensuite la relation  $f(y_n) = \frac{1}{n}$  sous la forme  $y_n = n \ln y_n$ , et on itère cette relation, ce qui donne  $y_n = n \ln(n \ln y_n) = n \ln n + n \ln \ln y_n$ . Comme  $y_n < n^2$ , on a  $0 < n \ln(\ln y_n) < n \ln(2 \ln n)$ , qui est négligeable devant  $n \ln n$ . On vient donc de trouver un équivalent de  $y_n$  :

$$y_n \sim n \ln n.$$

Pour aller plus loin, on itère la relation de définition de  $y_n$  sous la forme  $y_n = n \ln n + n \ln(\ln(n \ln y_n)) = n \ln n + n \ln(\ln n + \ln \ln y_n)$ . De  $y_n \sim n \ln n$ , on déduit que  $\ln \ln y_n \sim \ln \ln n$ , puis que, quand  $n$  tend vers  $+\infty$ ,

$$\begin{aligned} y_n &= n \ln n + n \ln \left( \ln n \left[ 1 + \frac{\ln(\ln y_n)}{\ln n} \right] \right) \\ &= n \ln n + n \ln(\ln n) + \ln \left[ 1 + \frac{\ln(\ln y_n)}{\ln n} \right] \\ &= n \ln n + n \ln(\ln n) + o(n \ln(\ln n)). \end{aligned}$$

On pourrait poursuivre les calculs en utilisant les mêmes idées.

### Exercice 9

Si  $x \in \mathbb{R}$ , on pose  $\langle x \rangle = x - \lfloor x \rfloor$ . Soit  $(u_n)$  une suite réelle. On dit que  $(u_n)$  est équirépartie modulo 1 si, pour tout  $(a, b) \in [0, 1]^2$  avec  $a < b$ , on a  $c_n(a, b) \xrightarrow{n \rightarrow +\infty} b - a$ , où  $c_n(a, b) = \frac{1}{n} \text{Card}\{k \in \{1, \dots, n\}, \langle u_k \rangle \in [a, b]\}$ .

- On pose, pour  $n \in \mathbb{N}$ ,  $u_n = \left(\frac{1+\sqrt{5}}{2}\right)^n$ .
  - Déterminer l'ensemble des suites  $(w_n)$  vérifiant  $\forall n \in \mathbb{N}, w_{n+2} = w_{n+1} + w_n$ .
  - En déduire l'existence d'une suite  $(v_n)$  de limite nulle telle que  $\forall n \in \mathbb{N}, u_n + v_n \in \mathbb{N}$ .
  - Montrer que  $(u_n)$  n'est pas équirépartie modulo 1.
- On pose, pour  $n \in \mathbb{N}$ ,  $u_n = \sqrt{n}$ .
  - écrire un programme permettant de calculer  $c_n(a, b)$ .
  - Afficher les 50 premiers termes pour  $a = 0$  et  $b = 0,5$ . Conjecture ?
- Faire le même type d'étude pour  $u_n = \ln(n)$ .
- Montrer que la suite de terme général  $u_n = \sqrt{n}$  est équirépartie modulo 1.

SOLUTION. —

- (a) L'équation caractéristique associée à la relation de récurrence linéaire d'ordre 2 à coefficients constants  $w_{n+2} - w_{n+1} - w_n = 0$  est  $z^2 - z - 1 = 0$ . Ses solutions complexes sont en fait réelles et valent  $\frac{1 \pm \sqrt{5}}{2}$ . Le cours affirme alors que les suites (réelles) cherchées sont celles qui vérifient

$$\exists (\lambda, \mu) \in \mathbb{R}^2, \quad \forall n \in \mathbb{N}, \quad w_n = \lambda \left( \frac{1 - \sqrt{5}}{2} \right)^n + \mu \left( \frac{1 + \sqrt{5}}{2} \right)^n.$$

- On pose  $\forall n \in \mathbb{N}, v_n = \left(\frac{1-\sqrt{5}}{2}\right)^n$  et  $w_n = u_n + v_n$ . La raison  $\frac{1-\sqrt{5}}{2}$  appartient à  $] -1, 0[$  car  $2 < \sqrt{5} < 3$ , donc  $(v_n)$  est une suite (géométrique) de limite nulle.

Par ailleurs,  $(w_n)$  est solution de l'équation étudiée à la question précédente, et il suffit manifestement que ses deux premiers termes soient des entiers naturels pour qu'elle soit à valeurs dans  $\mathbb{N}$  (récurrence immédiate à partir de l'égalité  $w_{n+2} = w_{n+1} + w_n$ ). Les deux calculs suivants achèvent la preuve :

$$w_0 = 1 + 1 = 2 \in \mathbb{N} \quad \text{et} \quad w_1 = \frac{1 + \sqrt{5}}{2} + \frac{1 - \sqrt{5}}{2} = 1 \in \mathbb{N}.$$

- On pose  $a = \frac{1}{4}$  et  $b = \frac{3}{4}$ . Comme  $\lim v_n = 0$ , il existe  $n_0 \in \mathbb{N}$  tel que  $\forall n \geq n_0, |v_n| < \frac{1}{4}$ . Alors, pour tout  $n \geq n_0$ , en utilisant la propriété  $\forall (k, x) \in \mathbb{Z} \times \mathbb{R}, \lfloor k + x \rfloor = k + \lfloor x \rfloor$  de la partie entière, on obtient

$$\langle u_n \rangle = u_n - \lfloor u_n + v_n - v_n \rfloor = -v_n - \lfloor -v_n \rfloor = \begin{cases} 1 - v_n \in ]\frac{3}{4}, 1] & \text{si } v_n \geq 0 \\ -v_n \in ]0, \frac{1}{4}[ & \text{si } v_n < 0. \end{cases}$$

Dans tous les cas,  $\langle u_n \rangle \notin [\frac{1}{4}, \frac{3}{4}]$  dès que  $n \geq n_0$ , et par conséquent,  $\forall n \geq n_0, 0 \leq c_n(\frac{1}{4}, \frac{3}{4}) \leq \frac{n_0}{n}$ , donc  $\lim c_n(\frac{1}{4}, \frac{3}{4}) = 0 \neq \frac{3}{4} - \frac{1}{4}$ . La suite  $(u_n)$  n'est pas équirépartie modulo 1.

- (a) On suppose que la suite  $(u_n)$  est donnée par une fonction  $f$  d'une variable réelle :  $\forall n \in \mathbb{R}, u_n = f(n)$ .

```
def c(a, b, n, f):
    compteur = 0
    for k in range(1, n + 1):
        uu = f(k) - floor(f(k))
        if a <= uu and uu <= b:
            compteur = compteur + 1
    return(compteur)
```

(b) L'interprétation des résultats de

```
for n in range(1,51):
    print(c(0, 0.5, n, sqrt)/n)
```

n'est pas évidente du tout : le quotient  $\frac{c_n(0,0.5)}{n}$  se rapproche de  $\frac{1}{2}$  par valeurs supérieures, mais oscille beaucoup. Si on remplace les 50 premiers quotients par les termes d'indice multiple de 1000 jusqu'à 100 000, c'est plus clair, et on conjecture que la suite  $(\sqrt{n})$  est équirépartie.

3. Cette fois, on observe une oscillation entre 0, 38 et 0, 62 quand on lance ce qui semble indiquer que la suite  $(\ln n)$  n'est pas équirépartie.
4. Soit  $(A, B) \in \mathbb{R}^2$  tel que  $A < B$ . Si l'on note  $p, p+1, \dots, p+j-1$  les entiers appartenant à  $[A, B]$ , on a  $A+j-1 \leq p+j-1 \leq B$ , donc  $j \leq B-A+1$ , et  $B < p+j < A+1+j$ , donc  $B-A-1 < j$ . En résumé, le nombre  $j$  d'entiers appartenant à  $[A, B]$  vérifie

$$B-A-1 < j \leq B-A+1.$$

Soient  $m \in \mathbb{N}$  et  $(a, b) \in [0, 1]^2$  tels que  $a < b$ . Comme  $m+a \leq \sqrt{k} \leq m+b$  équivaut à  $(m+a)^2 \leq k \leq (m+b)^2$ , le nombre  $j_m$  d'entiers  $k$  satisfaisant  $m+a \leq \sqrt{k} \leq m+b$  vérifie

$$2(b-a)m + b^2 - a^2 - 1 < j_m \leq 2(b-a)m + b^2 - a^2 + 1.$$

L'ensemble des  $k \in \{1, \dots, n\}$  tels que  $\langle \sqrt{k} \rangle \in [a, b]$  admet la partition en la réunion des blocs complets

$$U_m = \{k \in \mathbb{N}, m+a \leq \sqrt{k} \leq m+b\} \quad \text{pour} \quad 1 \leq m \leq \lfloor \sqrt{n} \rfloor - 1$$

et du bloc éventuellement incomplet  $\{k \in \{1, \dots, n\}, \lfloor \sqrt{n} \rfloor + a \leq \sqrt{k} \leq \lfloor \sqrt{n} \rfloor + b\}$  : on ne sait pas qui, de  $\sqrt{n}$  ou de  $\lfloor \sqrt{n} \rfloor + b$ , est le plus grand. On en déduit l'encadrement

$$j_1 + \dots + j_{\lfloor \sqrt{n} \rfloor - 1} \leq c_n(a, b) \leq j_1 + \dots + j_{\lfloor \sqrt{n} \rfloor}.$$

En posant  $\alpha = b^2 - a^2 - 1$  et en utilisant l'encadrement de  $j_m$  calculé plus haut, on trouve

$$(b-a)\lfloor \sqrt{n} \rfloor (\lfloor \sqrt{n} \rfloor - 1) + \alpha (\lfloor \sqrt{n} \rfloor - 1) \leq c_n(a, b) \leq (b-a)\lfloor \sqrt{n} \rfloor (\lfloor \sqrt{n} \rfloor + 1) + (\alpha + 2)\lfloor \sqrt{n} \rfloor.$$

Le majorant et le minorant de cet encadrement étant équivalents à  $(b-a)n$  quand  $n$  tend vers  $+\infty$ , on en déduit que  $\lim_{n \rightarrow \infty} \frac{c_n(a,b)}{n} = b-a$ , donc que la suite de terme général  $u_n = \sqrt{n}$  est équirépartie modulo 1.

REMARQUE. — La suite  $(\ln n)$  n'est pas équirépartie et en voici la preuve. On commence par évaluer le nombre  $j_m$  d'entiers  $k$  tels que  $m \leq \ln k \leq m + \frac{1}{2}$  pour un entier  $m \in \mathbb{N}$  donné. Comme ci-dessus, on a  $e^{m+1/2} - e^m - 1 < j_m \leq e^{m+1/2} - e^m + 1$  et, si l'on pose  $\alpha = e^{1/2} - 1$ , cet encadrement s'écrit

$$\alpha e^m - 1 < j_m \leq \alpha e^m + 1.$$

Comme ci-dessus, on a l'encadrement  $j_1 + \dots + j_{\lfloor \ln n \rfloor - 1} \leq c_N(0, \frac{1}{2}) \leq j_1 + \dots + j_{\lfloor \ln n \rfloor}$ , qui s'écrit aussi

$$m_n := \alpha \frac{e^{\lfloor \ln n \rfloor} - e}{e-1} - \lfloor \ln n \rfloor \leq c_n \left(0, \frac{1}{2}\right) \leq M_n := \alpha \frac{e^{\lfloor \ln n \rfloor + 1} - e}{e-1} + \lfloor \ln n \rfloor$$

Si  $n = n_p := \lfloor e^p \rfloor$  avec  $p \in \mathbb{N}^*$ , on a  $e^{p-1} < n < e^p$ , donc  $p-1 < \ln n < p$ , donc  $\lfloor \ln n \rfloor = p-1$ , et le majorant  $M_{n_p}$  vérifie  $M_{n_p} \sim \frac{\alpha e^p}{e-1} \sim \beta n_p$ , où l'on a posé  $\beta = \frac{\alpha}{e-1}$ . En supposant que la suite  $(\ln n)$  est équirépartie, on en déduirait que  $\frac{1}{2} \leq \beta = \frac{e^{1/2}-1}{e-1}$ . C'est faux : la commande `print((exp(1/2) - 1)/(exp(1) - 1))` renvoie 0.38.

### Exercice 10

Soit  $P_n : t \mapsto \sum_{k=0}^n \frac{t^k}{k!}$ .

1. Tracer le graphe de  $P_n$  pour  $n$  allant de 2 à 7 sur l'intervalle qui vous paraît le plus judicieux. Que dire des racines de  $P_n$  ?
2. Déterminer des valeurs approchées des racines complexes de  $P_n$  pour  $n$  allant de 2 à 7 en utilisant la méthode de Newton.
3. Représenter ces racines sur le plan complexe et commenter.
4. Montrer que  $P_n$  est scindé à racines simples sur  $\mathbb{C}$ .

SOLUTION. —

1. Une solution...

```
import matplotlib.pyplot as plt
import math as ma
f = []
for k in range(2,8):
    f.append(lambda x:sum(x**i/ma.factorial(i) for i in range(k + 1)))
T = np.linspace(-5, 1)
for k in range(len(f)):
    plt.plot(T, f[k](T))
plt.show()
```

Les racines de  $P_n$  sont bien sûr négatives, simples. Que dire d'autre ?

2. Une solution, bien peu économique...

```
import math as ma
import matplotlib.pyplot as plt

def derive(poly):
    return [i*poly[i] for i in range(1, len(poly))]

def horner(poly, x):
    R = 0
    for c in poly[::-1]:
        R* = x
        R+ = c
    return R

def newton(poly, x0, eps):
    pPrime = derive(poly)
    while abs(x0) < 20:
        if horner(pPrime, x0) == 0:
            return None
        else:
            x1 = x0 - horner(poly, x0) / horner(pPrime, x0)
            if abs(x1 - x0) < eps:
                return x1
    x0 = x1

l = []
for i in range(2,8):
    f = [1/ma.factorial(k) for k in range(i + 1)]
for a in range(-10, 10):
    for b in range(-10, 10):
        u = newton(f, complex(a, b), 0.1)
        if type(u) == complex:
            l.append(u)

X = []
for k in range(len(l)):
    X.append(l[k].real)

Y = []
for k in range(len(l)):
    Y.append(l[k].imag)

plt.plot(X, Y, 'o')
plt.show()
```

3. Chaque racine est bien sûr approchée de multiples fois dans le script précédent, par des approximations légèrement différentes. D'où dans le tracé un nuage de points quasi -confondus pour chaque racine. On observe (résultat non demandé ici) que les racines de  $P_6 = P'_7$  sont dans l'enveloppe convexe de celles de  $P_7$ , que les racines de  $P_5 = P'_6$  sont dans l'enveloppe convexe de celles de  $P_6$ , etc.

4. Soit  $n \in \mathbb{N}^*$ . Comme  $P'_n = P_{n-1}$  donc une racine double éventuelle de  $P_n$  est aussi racine de  $P_{n-1}$ , donc de  $P_n - P_{n-1} = \frac{X^n}{n!}$ . Autrement dit, la seule racine double possible de  $P_n$  est 0, or il est clair que 0 n'est pas racine de  $P_n$ . Donc  $P_n$  est scindé à racines simples sur  $\mathbb{C}$ .

### Exercice 11

Soit  $f: [0, 1] \rightarrow [0, 1]$ , telle que  $f(x) = x$  si  $x \in \{0, 1\}$  et  $f(x) = 0, a_2 a_1 a_3 \dots$  si  $x \in ]0, 1[$  a pour développement décimal illimité  $x = 0, a_1 a_2 a_3 \dots$

1. Tracer la courbe représentative de  $f$ .
2. La fonction  $f$  est-elle continue? Continue par morceaux?
3. Donner une valeur approchée de  $\int_0^1 f(x) dx$ .

SOLUTION. —

1.

```
def f(x):
    if x == 0 or x == 1:
        return(x)
    else:
        return(10*x - int(10*x))

import numpy as np
import matplotlib.pyplot as plt
T = np.linspace(0, 1, 200)
T = list(T)
F = [f(x) for x in T]
plt.plot(T, F)
plt.show()
```

2.  $f$  est continue par morceaux, car affine sur chaque intervalle  $[k/10(k+1)/10[$ , mais n'est clairement pas continue.
3. Question étrange. La valeur exacte se calcule sans peine :  $\int_0^1 f = \sum_{k=0}^9 \int_{k/10}^{(k+1)/10} (10x-k)dx = 10 \int_0^{1/10} 10x dx = \frac{1}{2}$ . Peut être l'examineur attend-il que le candidat programme une méthode des rectangles ou des trapèzes...

### Exercice 12

Soit  $f_n: x \mapsto -a_0 + \sum_{k=1}^n a_k x^k$  avec  $a_0 > 0$ ,  $a_1 > 0$  et  $a_k \geq 0$  pour  $k \in \{2, \dots, n\}$ .

1. Montrer qu'une telle fonction  $f_n$  s'annule une unique fois sur  $\mathbb{R}_+^*$ . On note  $u_n$  son zéro.
2. On pose  $g_n: x \mapsto -1 + \sum_{k=1}^n (k+1)x^k$ .
  - (a) Représenter les graphes sur  $[0, 1]$  des  $g_n$  pour  $n \in \{1, \dots, 7\}$ . Conjecture sur la suite  $(u_n)$ ?
  - (b) Donner une expression simple de  $g_n$  et en déduire le résultat.
3. Déterminer la limite de  $(u_n)$  pour  $a_k = k!$  pour  $k \in \mathbb{N}$ .

SOLUTION. —

1. La fonction  $f_n$  est polynomiale, donc dérivable, et  $f'_n: x \mapsto a_1 + \sum_{k=2}^n k a_k x^{k-1}$  est strictement positive sur  $\mathbb{R}_+$  d'après les hypothèses sur les signes des coefficients. Elle admet donc au plus une racine dans  $\mathbb{R}_+$ , donc dans  $\mathbb{R}_+^*$ .  
Par ailleurs,  $f_n(0) = -a_0 < 0$  et  $\lim_{+\infty} f_n(x) = +\infty$ , donc il existe  $b \in \mathbb{R}_+^*$  tel que  $f(b) > 0$ . En appliquant le théorème des valeurs intermédiaires à la fonction continue  $f_n$  sur  $[0, b]$ , on prouve l'existence d'au moins une racine de  $f_n$  dans  $\mathbb{R}_+^*$ .  
Finalement, une telle fonction  $f_n$  s'annule une unique fois sur  $\mathbb{R}_+^*$ , en  $u_n$ .

2. (a) Voici une manière de tracer des graphes sur un même dessin.

```
def g(n, x):
    resultat = -1
    for k in range(1, n + 1):
        resultat = resultat + (k + 1)*x**k
    return(resultat)
x = linspace(0, 1, 100)
L = []
couleurs = ["red", "blue", "green", "orange", "brown", "pink", "black"]
for k in range(1,8):
    def h(x):
        return(g(k, x))
    y = h(x)
    L.append(y)
for k in range(1,8):
    plot(x, L[k - 1], color = couleurs[k - 1])
    grid()
    title("Graphes des fonctions g_k sur [0, 1]")
    xlabel("x")
    ylabel("y = g_k(x)")
    axhline(color = "black")
    axvline(color = "black")
show()
```

Sur le graphe demandé par l'énoncé (à droite), on n'y voit rien. Sur le zoom (à droite), on conjecture que  $(u_n)$  décroît, et que sa limite  $\ell$  n'est pas nulle (proche de 0,3?).

- (b) On pose  $\forall x \in \mathbb{R}, G_n(x) = -x + \sum_{k=1}^n x^{k+1}$ . On constate que  $g_n = G'_n$  et que pour tout  $x \neq 1$ , on a

$$G_n(x) = -x + \frac{x^2 - x^{n+2}}{1-x}. \text{ Il en résulte que}$$

$$\forall x \neq 1, \quad g_n(x) = -1 + \frac{(2x - (n+2)x^{n+1})(1-x) + x^2 - x^{n+2}}{(1-x)^2} = \frac{(n+1)x^{n+2} - (n+2)x^{n+1} - 2x^2 + 4x - 1}{(1-x)^2}.$$

Le nombre  $u_n$  vérifie donc la relation

$$(n+1)u_n^{n+2} - (n+2)u_n^{n+1} - 2u_n^2 + 4u_n - 1 = 0.$$

Comme la suite de fonctions  $(g_n)$  est croissante sur  $[0, 1]$ , c'est-à-dire que pour tout  $x \in [0, 1]$ , la suite  $(g_n(x))$  est croissante, on a

$$0 = g_n(u_n) \leq g_{n+1}(u_n).$$

Comme  $g_{n+1}$  est une fonction croissante sur  $[0, 1]$ , on en déduit que la racine  $u_{n+1}$  de  $g_{n+1}$  est plus petite que  $u_n$ , ce qui montre que  $(u_n)$  décroît. Décroissante et minorée (par zéro), la suite  $(u_n)$  converge, et sa limite vérifie  $0 \leq \ell \leq u_1 = \frac{1}{2}$ .

Les résultats de croissances comparées montrent alors que les deux suites de termes généraux  $(n+1)u_n^{n+2}$  et  $(n+2)u_n^{n+1}$  convergent vers zéro. En passant à la limite dans l'équation satisfaite par  $u_n$  on trouve alors

$$2\ell^2 - 4\ell + 1 = 0.$$

La limite cherchée appartenant à  $[0, 1]$ , elle vaut (la valeur numérique est en accord avec le zoom fourni plus haut) :

$$\ell = \frac{2 - \sqrt{2}}{2} \approx 0,29.$$

3. Dans cette question, on a donc  $f_n(x) = -1 + \sum_{k=1}^n k!x^k$ . Les mêmes arguments que dans le cas de la suite

$(g_n)$  montrent que  $f_n$  possède une unique racine dans  $\mathbb{R}_+^*$ , notée encore  $u_n$ , que la suite  $(u_n)$  décroît, donc qu'elle converge vers une limite notée encore  $\ell$ , avec  $\ell \geq 0$ . Si jamais  $\ell$  était strictement positive, on aurait  $\forall n, 0 < \ell \leq v_n$  donc, par croissance de la fonction  $f_n$  sur  $\mathbb{R}_+$ ,

$$f_n(\ell) \leq f_n(v_n) = 0.$$

Or le membre de gauche de cette inégalité tend vers  $+\infty$  par croissances comparées (le rayon de convergence de la série entière de terme général  $k!z^k$  est nul), donc  $+\infty \leq 0$  : absurde. L'hypothèse  $\ell > 0$  est donc fausse, donc

$$\ell = 0.$$