
CONCOURS BLANC
Mathématiques - mardi 12 mai 2026
Durée : 3 h

L'usage d'abaques, de tables, de calculatrice et de tout instrument électronique susceptible de permettre au candidat d'accéder à des données et de les traiter par les moyens autres que ceux fournis dans le sujet est interdit.

Chaque candidate ou candidat est responsable de la vérification de son sujet d'épreuve : pagination et impression de chaque page. Ce contrôle doit être fait en début d'épreuve. En cas de doute, il convient d'alerter au plus tôt l'équipe de surveillance qui vérifiera et, éventuellement, remplacera le sujet.

Ce sujet comporte 6 pages numérotées de 1 à 6.

Si, au cours de l'épreuve, une candidate ou un candidat repère ce qui lui semble être une erreur d'énoncé, elle ou il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives prises.

Ce sujet comporte 3 exercices indépendants.

On pourra admettre le résultat d'une question ou d'une sous-question pour passer aux questions suivantes, à condition de le mentionner explicitement.

Une annexe dans laquelle certaines commandes Python sont rappelées est jointe à la fin du sujet. **Pour les questions d'informatique, on considérera que les importations de modules nécessaires ont été préalablement faites.**

Exercice 1. un montage électrique

On propose de modéliser un câble coaxial par un circuit électrique constitué d'une chaîne de n modules identiques comportant chacun trois conducteurs ohmiques en dérivation avec un quatrième conducteur ohmique. Tous les conducteurs ohmiques sont de résistance $r > 0$. Dans tout l'exercice, on fixe donc $r > 0$.

Pour simuler un câble de grande longueur on souhaite déterminer la valeur de la résistance équivalente de la chaîne lorsque n tend vers $+\infty$.

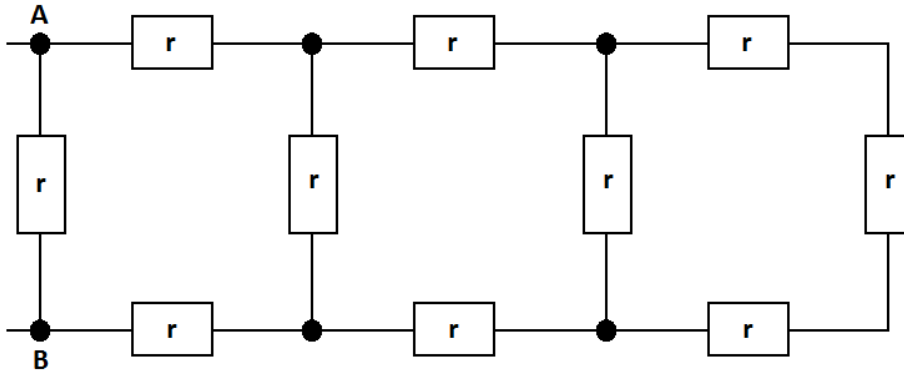


Schéma du circuit lorsque $n = 3$

Pour $n \in \mathbb{N}^*$ on note R_n la résistance équivalente de la chaîne à n modules ; et on pose par convention $R_0 = r$. Pour $n = 1$, la chaîne est par exemple équivalente à un circuit avec un seul conducteur ohmique de résistance R_1 (figure de gauche ci-dessous).

De manière générale, le lien entre R_n et R_{n+1} s'obtient en disant que les deux circuits présentés sur la figure de droite sont équivalents :

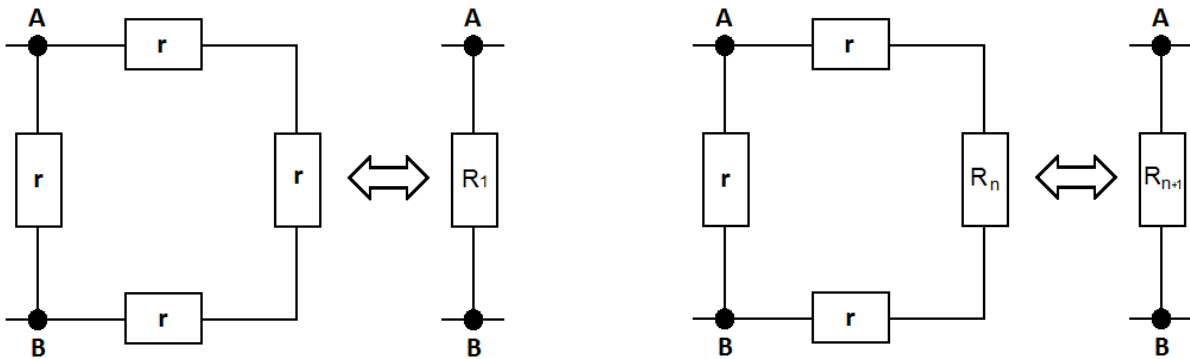


Schéma du circuit lorsque $n = 1$

Passage de R_n à R_{n+1}

On admet que les règles de calculs des résistances équivalentes en électricité donnent : $\frac{1}{R_1} = \frac{1}{r} + \frac{1}{3r}$, et, de manière générale :

$$\forall n \in \mathbb{N}, \frac{1}{R_{n+1}} = \frac{1}{r} + \frac{1}{2r + R_n}$$

1. (a) Montrer que $R_1 = \frac{3}{4}r$. Puis, de manière générale, montrer que pour tout $n \geq 0$, $R_{n+1} = f(R_n)$ où f est la fonction donnée par

$$f : x \mapsto r \frac{2r + x}{3r + x}.$$

- (b) Justifier alors que la suite (R_n) est correctement définie.
- (c) i. Écrire une fonction Python Termes(n, r) d'argument un entier naturel n et $r > 0$, et qui calcule et renvoie la valeur du terme R_n

- ii. En utilisant la fonction `Termes`, écrire alors une fonction `Liste_termes(N, r)` qui prend en arguments un entier N non nul et $r > 0$ et qui renvoie la **liste** des termes de R_0 à R_N .
 - iii. Afin d'optimiser la fonction `Liste_termes`, proposer une autre fonction `Liste_mieux(N, r)` qui renvoie la liste $[R_0, \dots, R_N]$ sans utiliser la fonction `Termes`.
 - iv. Écrire un programme Python permettant de tracer le graphe de la suite (R_n) pour $n \in \llbracket 0, N \rrbracket$. *On utilisera des variables globales pour r et N .*
2. Dans cette question, on étudie la convergence de la suite (R_n) .
- (a) Étudier les variations de f sur \mathbb{R}^+ .
 - (b) Montrer que (R_n) converge.
 - (c) Montrer que $\lim_{n \rightarrow +\infty} R_n = (\sqrt{3} - 1)r$. Interpréter ce résultat physiquement.
3. Dans cette question, on note $R^* = (\sqrt{3} - 1)r$ la limite de la suite (R_n) et on s'intéresse à quelle valeur de $n \in \mathbb{N}$ choisir pour que R_n soit suffisamment proche de R^* .
- (a) Montrer que pour tout réel $x > 0$,

$$f(x) - R^* = \frac{r^2}{(3r + x)(3r + R^*)} (x - R^*)$$

(b) Montrer que : $\frac{1}{3(2 + \sqrt{3})} \leq \frac{1}{9}$

- (c) En utilisant les deux questions précédentes, démontrer que : $\forall x > 0$

$$|f(x) - R^*| \leq \frac{1}{9}|x - R^*|$$

(d) En déduire que : $\forall n \in \mathbb{N}, |R_n - R^*| \leq \frac{(2 - \sqrt{3})r}{9^n}$.

- (e) Écrire une fonction Python `Premier(r, eps)`, prenant en arguments la valeur r et un seuil $eps > 0$ et renvoyant le plus petit entier $n \in \mathbb{N}$ tel que $|R_n - R^*| \leq eps$

Exercice 2. Autofécondation d'un individu diploïde

On s'intéresse à l'évolution du génotype d'un individu diploïde se reproduisant par autofécondation. On se limite à l'étude d'un seul gène présentant deux allèles différents a et A . L'individu possède deux versions du gène, il peut donc être hétérozygote aA , homozygote aa ou homozygote AA .

On suppose qu'il se reproduit à chaque génération par autofécondation. Le génotype du descendant dépend alors uniquement du génotype de l'ancêtre et de l'aléa, lié aux mécanismes de la reproduction sexuée (méiose avec séparation des chromosomes homologues et fécondation avec rencontre aléatoire des gamètes).

1. (a) Compléter le tableau suivant donnant les probabilités conditionnelles d'avoir un descendant aa , aA ou AA sachant qu'on a un ancêtre aa , aA ou AA . *On recopiera le tableau sur la copie et on justifiera ses réponses par des explications en Français.*

	ancêtre aa	ancêtre aA	ancêtre AA
descendant aa	1	$\frac{1}{4}$	0
descendant aA	?	?	?
descendant AA	?	?	?

- (b) Écrire une fonction Python `Tableau()` qui renvoie le tableau de la question précédente.

Dans la suite de l'exercice, on note aa_n (respectivement aA_n et AA_n) l'événement "l'individu de la n -ème génération est de génotype aa " (respectivement aA et AA). On note p_n, q_n et r_n les probabilités

suivantes : $p_n = \mathbb{P}(aa_n)$, $q_n = \mathbb{P}(aA_n)$, $r_n = \mathbb{P}(AA_n)$, et on note $X_n = \begin{pmatrix} p_n \\ q_n \\ r_n \end{pmatrix}$.

2. (a) Démontrer que pour tout $n \in \mathbb{N}$, on a $X_{n+1} = MX_n$ pour une matrice $M \in \mathcal{M}_3(\mathbb{R})$ à préciser.

- (b) En déduire l'expression de X_n en fonction de M^n et de X_0 .

- (c) Écrire une fonction python `puissances(n)` prenant en argument un entier n non nul et qui calcule et renvoie la n ème puissance de la matrice M
on pourra utiliser la fonction `Tableau()` de la question précédente

3. (a) Démontrer que pour tout $n \in \mathbb{N}^*$ on a $M^n = \begin{pmatrix} 1 & x_n & 0 \\ 0 & y_n & 0 \\ 0 & x_n & 1 \end{pmatrix}$ où (x_n) et (y_n) sont les suites définies par $x_1 = \frac{1}{4}$, $y_1 = \frac{1}{2}$ et : $\forall n \in \mathbb{N}^*$, $x_{n+1} = \frac{2x_n + 1}{4}$ et $y_{n+1} = \frac{y_n}{2}$.

- (b)
 - i. Écrire deux fonctions python `suite_x(n)` et `suite_y(n)` d'argument un entier n non nul et qui renvoient respectivement le terme x_n et y_n
 - ii. Écrire alors une fonction `puissances_bis(n)` qui renvoie la puissance n de M et qui utilise les deux fonctions `suite_x(n)` et `suite_y(n)`
 - iii. Écrire une fonction python `puissances_mieux(n)` qui renvoie la puissance n de M , mais qui n'utilise aucune des fonctions précédentes.

- (c) En déduire l'expression de M^n puis celles de p_n, q_n et r_n en fonction de n et de p_0, q_0 et r_0 .

4. (a) On suppose que $p_0, q_0, r_0 \in]0, 1[$. Quelles sont les limites des suites (p_n) , (q_n) et (r_n) ? Interpréter ce résultat en terme des génotypes des descendants.

(b) Pour $n \in \mathbb{N}$, on s'intéresse aux évènements H_n : "le descendant de la n -ème génération est homozygote" et T_n : "c'est à partir de la n -ème génération que les tous les descendants sont homozygotes". Montrer que $\mathbb{P}(H_n) = 1 - q_n$ puis que $\mathbb{P}(T_n) = \frac{q_0}{2^n}$.

(c) On s'intéresse à la quantité $S_n = \sum_{k=0}^n k \mathbb{P}(T_k)$. Montrer que :

$$\forall n \in \mathbb{N}, S_n = q_0 \times \left(2 - \frac{n+2}{2^n}\right).$$

En déduire la valeur de $\lim_{n \rightarrow +\infty} S_n$ et interpréter ce résultat.

Exercice 3. On se place dans l'espace vectoriel \mathbb{R}^4 et on considère les vecteurs suivants :

$$u_1 = (1, -1, 1, 2) ; u_2 = (0, 1, 2, 1) ; u_3 = (2, -1, 4, 5)$$

On définit , pour tout réel a :

$$F_a = \{(x, y, z, t) \in \mathbb{R}^4 : x + ay - z = 0 \text{ et } 2x + y + (a - 1)t = 0\} \quad \text{et} \quad G = \text{Vect}(u_1, u_2, u_3).$$

1. (a) Montrer que F_a est un sous-espace vectoriel de \mathbb{R}^4 .
(b) Déterminer une base et la dimension de F_a en fonction de a .
2. (a) La famille (u_1, u_2, u_3) est-elle libre ?
(b) Déterminer une base et la dimension de G .
(c) Déterminer la (les) équation(s) de G .
3. (a) Existe-t-il $a \in \mathbb{R}$ tel que $u_1 \in F_a$? Même question pour u_2 et u_3 .
(b) Montrer que : pour tout réel a , G n'est pas inclus dans F_a .
4. (a) Le vecteur $(0, 0, 0, 1)$ appartient-il à G ?
(b) En déduire qu'il existe $a \in \mathbb{R}$ tel que F_a n'est pas inclus dans G .
(c) Déterminer base et dimension de $F_a \cap G$.

Annexe Python

- Dans le module `matplotlib.pyplot` importé sous l'alias `plt` :
`plt.plot(X, Y)` prend en entrée deux vecteurs ou deux listes de même taille, et réalise le tracé des points d'abscisses prises dans `X` et d'ordonnées prises dans `Y`.
On utilise `plt.show()` pour afficher le tracé.
- Dans le module `numpy` importé sous l'alias `np` :
`np.dot` renvoie le produit matriciel (si A et B sont deux matrices de tailles compatibles, alors `np.dot(A, B)` renvoie le produit matriciel AB)
`np.zeros((n, p))` renvoie le tableau nul à n lignes et p colonnes.
`np.eye(n)` renvoie la matrice identité de $\mathcal{M}_n(\mathbb{R})$

FIN DU SUJET
