

DS n°9

le 13/03/2026

Durée : 4h

- L'usage de la calculatrice n'est pas autorisé.
- La clarté et la précision des raisonnements interviendront pour une grande part dans la notation.
- Le résultat d'une question peut être admis afin de traiter une question suivante.
- On encadrera le résultat de chaque question.

### Exercice 1 :

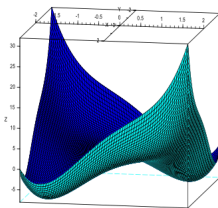
On considère la fonction  $f$  qui à tout couple  $(x, y)$  de  $\mathbb{R}^2$  associe le réel :

$$f(x, y) = x^4 + y^4 - 2(x - y)^2$$

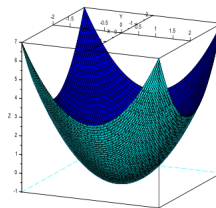
1. Justifier que  $f$  est de classe  $C^2$  sur  $\mathbb{R}^2$ .
2. (a) Calculer les dérivées partielles d'ordre 1 de  $f$ .
  - (b) Montrer que le gradient de  $f$  est nul si, et seulement si, on a : 
$$\begin{cases} x^3 - x + y = 0 \\ y^3 + x - y = 0 \end{cases}$$
  - (c) En déduire que  $f$  possède trois points critiques :  $(0, 0)$ ,  $(\sqrt{2}, -\sqrt{2})$ ,  $(-\sqrt{2}, \sqrt{2})$ .
3. (a) Calculer les dérivées partielles d'ordre 2 de  $f$ .
  - (b) Écrire la matrice hessienne de  $f$  en chaque point critique.
  - (c) Déterminer les valeurs propres de chacune de ces trois matrices puis montrer que  $f$  admet un minimum local en deux de ses points critiques. Donner la valeur de ce minimum.
  - (d) Déterminer les signes de  $f(x, x)$  et  $f(x, -x)$  au voisinage de  $x = 0$ . Conclure quant à l'existence d'un extremum en le troisième point critique de  $f$ .
4. (a) Pour tout  $(x, y)$  de  $\mathbb{R}^2$ , calculer  $f(x, y) - (x^2 - 2)^2 - (y^2 - 2)^2 - 2(x + y)^2$ .
  - (b) Que peut-on déduire de ce calcul quant au minimum de  $f$  ?
5. (a) Compléter la deuxième ligne du script suivant afin de définir la fonction  $f$ .

```
def f(x, y) :
    ...
```

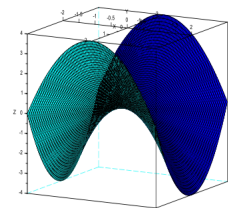
- (b) À partir de la fonction précédente, on trace la représentation en trois dimension du graphique de la fonction  $f$ . On obtient l'une des trois nappes ci-dessous. Laquelle ?



Nappe 1



Nappe 2



Nappe 3

## Exercice 2

Dans tout cet exercice,  $f$  désigne la fonction définie sur  $]0, +\infty[$  par :  $\forall x \in ]0, +\infty[, f(x) = x - \ln(x)$ .

### Partie I : Étude de la fonction $f$

1. Dresser le tableau de variations de  $f$  en précisant ses limites en 0 et en  $+\infty$ .
2. Montrer que l'équation  $f(x) = 2$ , d'inconnue  $x \in ]0, +\infty[$ , admet exactement deux solutions, que l'on note  $a$  et  $b$ , telles que  $0 < a < 1 < b$ .
3. Montrer :  $b \in [2; 4]$ . On pourra utiliser  $\ln(2) \approx 0,7$ .

### Partie II : Étude d'une suite

On pose :  $u_0 = 4$  et  $\forall n \in \mathbb{N}, u_{n+1} = \ln(u_n) + 2$ .

1. Montrer que la suite  $(u_n)_{n \in \mathbb{N}}$  est bien définie et que l'on a :  $\forall n \in \mathbb{N}, u_n \in [b, +\infty[$ .
2. Déterminer la monotonie de la suite  $(u_n)_{n \in \mathbb{N}}$ . En déduire qu'elle converge et préciser sa limite.
3. (a) Montrer :  $\forall n \in \mathbb{N}, u_{n+1} - b \leq \frac{1}{2}(u_n - b)$ .  
(b) En déduire :  $\forall n \in \mathbb{N}, 0 \leq u_n - b \leq \frac{1}{2^{n-1}}$ .
4. (a) Écrire une fonction Python d'en-tête `suite(n)` qui, prenant en argument un entier  $n$  de  $\mathbb{N}$ , renvoie la valeur de  $u_n$ .  
On suppose le module `numpy` importé par la commande `import numpy as np`.  
(b) Recopier et compléter la ligne 3 de la fonction Python suivante afin que, prenant en argument un réel `epsilon` strictement positif, elle renvoie une valeur approchée de  $b$  à `epsilon` près.

```

1. def valeur_approchee(epsilon):
2.     n = 0
3.     while .....
4.         n = n+1
5.         b = suite(n)
6.     return b

```

### Partie III : Étude d'une fonction de deux variables

On considère la fonction  $H$  de classe  $\mathcal{C}^2$  sur l'ouvert  $U = \mathbb{R}_+^* \times \mathbb{R}$  définie par :

$$\forall (x, y) \in \mathbb{R}_+^* \times \mathbb{R}, H(x, y) = \frac{x^2}{2} - xy - 2x + e^y.$$

1. (a) Calculer les dérivées partielles d'ordre 1 de  $H$  en tout  $(x, y)$  de  $U$ .  
(b) Montrer que la fonction  $H$  admet exactement deux points critiques :  $(a, \ln(a))$  et  $(b, \ln(b))$ , où les réels  $a$  et  $b$  sont ceux introduits dans la question 2.
2. (a) Écrire la matrice hessienne, notée  $M_a$ , de  $H$  au point  $(a, \ln(a))$ .  
(b) Montrer que  $M_a$  admet deux valeurs propres distinctes, notées  $\lambda_1$  et  $\lambda_2$ , vérifiant : 
$$\begin{cases} \lambda_1 + \lambda_2 = a + 1 \\ \lambda_1 \lambda_2 = a - 1. \end{cases}$$
  
(c) La fonction  $H$  présente-t-elle un extremum local au point  $(a, \ln(a))$  ?
3. La fonction  $H$  présente-t-elle un extremum local au point  $(b, \ln(b))$  ?

## Exercice 3 :

Soit  $p$  un réel appartenant à l'intervalle ouvert  $]0; 1[$ . On note  $q = 1 - p$ .

On dispose dans tout l'exercice d'une même pièce dont la probabilité d'obtenir PILE vaut  $p$ .

## Partie I : Étude d'une première expérience

On procède à l'expérience suivante  $\mathcal{E}$  : “ On effectue une succession illimitée de lancers de la pièce ”.

On note :

- pour tout entier naturel non nul  $n$ ,  $X_n$  la variable aléatoire égale au nombre de PILE obtenus lors des  $n$  premiers lancers de la pièce ;
- pour tout entier naturel non nul  $j$ ,  $F_j$  l'événement : “ La pièce donne FACE lors du  $j$ -ième lancer “ ;
- $Y$  la variable aléatoire égale au nombre de FACE obtenus avant l'apparition du second PILE.

Par exemple, si les lancers ont donné dans cet ordre :

” FACE, PILE, FACE, FACE, FACE, PILE“

alors  $Y = 4$ .

On admet que les variables aléatoires  $X_n$  ( $n \in \mathbb{N}^*$ ) et  $Y$  sont définies sur un même espace probabilisé  $(\Omega, P)$  modélisant l'expérience  $\mathcal{E}$ .

### 1. Simulation informatique.

- (a) Compléter la fonction Python suivante, qui prend un réel  $p \in ]0, 1[$  en entrée et qui renvoie 1 avec probabilité  $p$  et 0 sinon.

On suppose qu'on a importé le module `numpy.random` par la commande `import numpy.random as rd`. On rappelle qu'alors la commande `rd.random()` simule la réalisation d'une variable aléatoire suivant une loi uniforme sur  $[0, 1]$ .

```
def lancer(p):
    a = rd.random()
    if a ... :
        L = ...
    else :
        L = ...
    return L
```

- (b) Compléter la fonction Python suivante, qui prend un réel  $p \in ]0, 1[$  en entrée, qui simule autant de lancers de la pièce que nécessaire jusqu'à l'obtention du premier PILE et renvoie le nombre de lancers effectués.

```
def premierPile(p):
    Z = 1 # nombre de lancers effectués
    L = lancer(p) # r\'esultat du premier lancer
    while L != ... :
        Z = ...
        L = ...
    return Z
```

- (c) Écrire une fonction Python d'en-tête `def nbFaces(p)` : qui simule autant de lancers de la pièce que nécessaire jusqu'à l'obtention du second PILE, et renvoie le nombre de FACE obtenus en tout.
- (d) On exécute alors le programme suivant :

```
p = 0.4
N = 10000
S = 0
for k in range(N):
    S = S + nbFaces(p)

moy = S/N
print(moy)
```

Le résultat affiché est 2.9943.

Expliquer la signification de ce résultat.

2. Soit  $n$  un entier naturel non nul. Donner la loi de  $X_n$ . Préciser la valeur de son espérance  $E(X_n)$  et de sa variance  $V(X_n)$ .
3. Déterminer les valeurs prises par la variable aléatoire  $Y$ .
4. Donner les valeurs des probabilités :

$$P(Y = 0), \quad P(Y = 1) \quad \text{et} \quad P(Y = 2)$$

5. Soit  $n$  un entier naturel. Justifier que les événements :

$$(Y = n) \quad \text{et} \quad (X_{n+1} = 1) \cap \overline{F_{n+2}}$$

sont égaux.

6. Prouver que :

$$\forall n \in \mathbb{N}, \quad P(Y = n) = (n + 1)p^2q^n$$

7. Vérifier par le calcul que :

$$\sum_{n=0}^{+\infty} P(Y = n) = 1$$

8. Démontrer que la variable aléatoire  $Y$  possède une espérance  $E(Y)$  et donner sa valeur.
9. Le résultat de la question précédente est-il cohérent avec le résultat obtenu par simulation.
10. Soit  $k \in \mathbb{N}^*$ . On note  $Y_k$  la variable aléatoire égale au nombre de FACE obtenus avant l'apparition du  $k$ -ième PILE. En particulier, on a  $Y_2 = Y$ .  
En généralisant la méthode utilisée dans les questions précédentes, déterminer la loi de  $Y_k$ .

## Partie II : Étude d'une seconde expérience

On procède à l'expérience suivante :

$\mathcal{F}$  : " Deux joueurs se relaient pour effectuer des lancers successifs de la pièce pendant la pause déjeuner.

Le joueur 1 arrive à 12h (considéré comme l'instant 0) et joue jusqu'à l'arrivée du joueur 2.

Le joueur 2 arrive au hasard entre 12h et 13h puis joue jusqu'à 13h (considéré comme l'instant 1). "

On note :

- $R$  la variable aléatoire égale à la durée (en heure) du jeu pour le joueur 1 ;
- $S$  la variable aléatoire égale à la durée (en heure) du jeu pour le joueur 2 ;
- $T$  la variable aléatoire égale à la durée (en heure) de jeu effectuée par le joueur ayant joué le plus longtemps c'est-à-dire que :

$$T = \max(R, S)$$

Pour tout variable aléatoire  $X$ , on note  $F_X$  la fonction de répartition de  $X$ .

On admet que  $R$  et  $S$  sont deux variables aléatoires définies sur un même espace probabilisé muni d'une probabilité  $P$  modélisant l'expérience  $\mathcal{F}$ . En outre, on suppose que :

$$\boxed{R \text{ suit la loi uniforme sur } [0; 1] \text{ et que } S = 1 - R}$$

(cette dernière relation traduisant que le temps total consacré au jeu par le joueur 1 et le joueur 2 est exactement d'une heure).

1. Expliciter la fonction  $F_R$  puis la fonction  $F_S$ . Reconnaître alors la loi suivie par la variable aléatoire  $S$ .
2. Pour tout réel  $t$ , prouver que :

$$P(T \leq t) = P\left((R \leq t) \cap (R \geq 1 - t)\right)$$

3. Déterminer, pour tout  $t \in \left[\frac{1}{2}; 1\right]$ , l'expression de  $F_T$  en fonction de  $T$ .
4. Justifier que  $T$  suit la loi uniforme sur  $\left[\frac{1}{2}; 1\right]$ .
5. En déduire que  $T$  admet une espérance  $E(T)$  et une variance  $V(T)$  que l'on précisera.