



PRÉPASIMPLE

prepasimple.fr

EDHEC

Mathématiques — Option économique

2017

Mathématiques appliquées

Mathématiques appliquées — Prépa ECG

Annales de concours · Sujet

Exercice 1

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

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

1) Justifier que f est de classe $\mathcal{C}^2$ sur $\mathbb{R}^2$.

2) a) Calculer les dérivées partielles d'ordre 1 de f .

2) b) Montrer que le gradient de f est nul si, et seulement si, on a :

$$\begin{cases} x^3 - x + y = 0, \\ y^3 + x - y = 0. \end{cases}$$

2) c) En déduire que f possède trois points critiques : $(0, 0)$, $(\sqrt{2}, -\sqrt{2})$ et $(-\sqrt{2}, \sqrt{2})$.

3) a) Calculer les dérivées partielles d'ordre 2 de f .

3) b) Écrire la matrice hessienne de f en chaque point critique.

3) 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.

3) 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 $f(x, y) - (x^2 - 2)^2 - (y^2 - 2)^2 - 2(x + y)^2$.

- 4) b) Que peut-on déduire de ce calcul quant au minimum de f ?
- 5) a) Compléter la ligne return du script suivant afin de définir la fonction f .

Python

```
import numpy as np

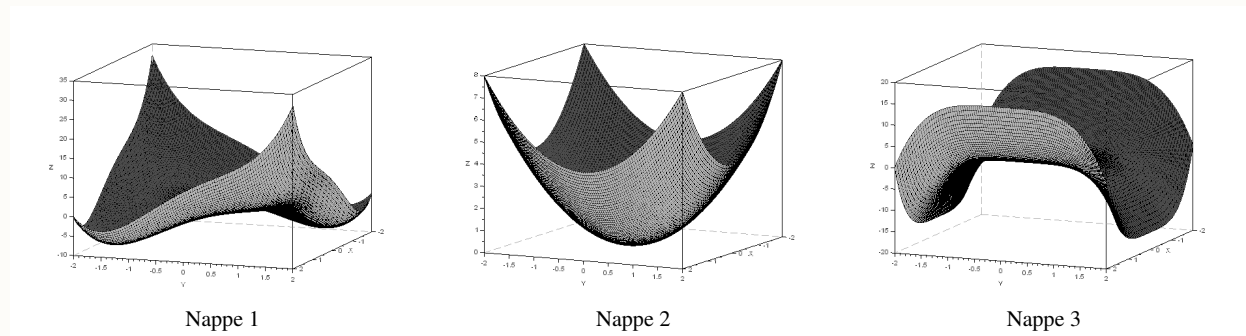
import matplotlib.pyplot as plt

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

x = np.linspace(-2, 2, 101)
y = x
X, Y = np.meshgrid(x, y)
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
ax.plot_surface(X, Y, f(X, Y), cmap="gray", edgecolor="black", linewidth=0.2)
plt.show()
```

`np.linspace(a, b, m)` renvoie m valeurs régulièrement espacées entre a et b , bornes incluses. `np.meshgrid(x, y)` construit deux tableaux de coordonnées permettant d'évaluer f sur toutes les paires de valeurs de x et y . Les opérations arithmétiques appliquées à ces tableaux agissent élément par élément. `plt.figure()` crée une figure; `fig.add_subplot(111, projection="3d")` y crée un repère tridimensionnel unique. `ax.plot_surface(X, Y, Z, ...)` représente la surface de hauteur Z aux coordonnées X, Y ; `cmap="gray"` utilise des niveaux de gris, `edgecolor="black"` colore les mailles en noir et `linewidth=0.2` en fixe l'épaisseur. `plt.show()` affiche la figure.

- 5) b) Le script précédent, une fois complété, renvoie l'une des trois nappes suivantes. Laquelle? Justifier la réponse.



Exercice 2

On note E l'espace vectoriel des fonctions polynomiales de degré inférieur ou égal à 2 et on rappelle que la famille (e_0, e_1, e_2) est une base de E , les fonctions e_0, e_1, e_2 étant définies par :

$$\forall t \in \mathbb{R}, \quad e_0(t) = 1, \quad e_1(t) = t, \quad e_2(t) = t^2.$$

On considère l'application φ qui, à toute fonction P de E , associe la fonction, notée

$\varphi(P)$, définie par :

$$\forall x \in \mathbb{R}, \quad (\varphi(P))(x) = \int_0^1 P(x+t) dt.$$

- 1) a) Montrer que φ est linéaire.
- 1) b) Déterminer $(\varphi(e_0))(x)$, $(\varphi(e_1))(x)$ et $(\varphi(e_2))(x)$ en fonction de x , puis écrire $\varphi(e_0)$, $\varphi(e_1)$ et $\varphi(e_2)$ comme combinaisons linéaires de e_0, e_1, e_2 .
- 1) c) Dédurre des questions précédentes que φ est un endomorphisme de E .
- 2) a) Écrire la matrice A de φ dans la base (e_0, e_1, e_2) . On vérifiera que la première ligne de A est $\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \end{pmatrix}$.
- 2) b) Justifier que φ est un automorphisme de E .
- 2) c) L'endomorphisme φ est-il diagonalisable ?
- 3) Compléter les commandes Python suivantes pour que soit affichée la matrice A^n pour une valeur de n entrée par l'utilisateur :

Python

```
import numpy as np

import numpy.linalg as al

n = int(input("Entrez une valeur entière naturelle pour n : "))

A = np.array(...)

print(...)
```

`input` lit une chaîne saisie au clavier et `int` la convertit en entier ; on suppose que l'utilisateur saisit un entier naturel. `np.array` crée un tableau à partir d'une liste de lignes. `al.matrix_power(M, n)` renvoie la puissance entière n de la matrice carrée M . `print` affiche son argument.

- 4) a) Montrer par récurrence que, pour tout entier naturel n , il existe un réel u_n tel que l'on ait :

$$A^n = \begin{pmatrix} 1 & n/2 & u_n \\ 0 & 1 & n \\ 0 & 0 & 1 \end{pmatrix}.$$

Donner u_0 et établir que :

$$\forall n \in \mathbb{N}, \quad u_{n+1} = u_n + \frac{1}{6}(3n + 2).$$

- 4) b) En déduire, par sommation, l'expression de u_n pour tout entier naturel n .
- 4) c) Écrire A^n sous forme de tableau matriciel.

Exercice 3

Soit V une variable aléatoire suivant la loi exponentielle de paramètre 1, dont la fonction de répartition est la fonction F_V définie par :

$$F_V(x) = \begin{cases} 0 & \text{si } x \leq 0, \\ 1 - e^{-x} & \text{si } x > 0. \end{cases}$$

On pose $W = -\ln V$ et on admet que W est aussi une variable aléatoire dont la fonction de répartition est notée F_W . On dit que W suit la loi de Gumbel.

- 1) a) Montrer que : $\forall x \in \mathbb{R}, \quad F_W(x) = e^{-e^{-x}}$.

- 1) b) En déduire que W est une variable à densité.

On désigne par n un entier naturel non nul et par $X_1, \dots, X_n$ des variables aléatoires définies sur le même espace probabilisé, indépendantes et suivant la même loi que V , c'est-à-dire la loi $\mathcal{E}(1)$.

On considère la variable aléatoire Y_n définie par $Y_n = \max(X_1, \dots, X_n)$, c'est-à-dire que, pour tout ω de Ω , on a $Y_n(\omega) = \max(X_1(\omega), \dots, X_n(\omega))$. On admet que Y_n est une variable aléatoire à densité.

2) a) Montrer que la fonction de répartition F_{Y_n} de Y_n est définie par :

$$F_{Y_n}(x) = \begin{cases} 0 & \text{si } x < 0, \\ (1 - e^{-x})^n & \text{si } x \geq 0. \end{cases}$$

2) b) En déduire une densité f_{Y_n} de Y_n .

3) a) Donner un équivalent de $1 - F_{Y_n}(t)$ lorsque t est au voisinage de $+\infty$, puis montrer que l'intégrale $\int_0^{+\infty} (1 - F_{Y_n}(t)) dt$ est convergente.

3) b) Établir l'égalité suivante :

$$\forall x \in \mathbb{R}_+, \quad \int_0^x (1 - F_{Y_n}(t)) dt = x(1 - F_{Y_n}(x)) + \int_0^x t f_{Y_n}(t) dt.$$

3) c) Montrer que $\lim_{x \rightarrow +\infty} x(1 - F_{Y_n}(x)) = 0$.

3) d) En déduire que Y_n possède une espérance et prouver l'égalité :

$$E(Y_n) = \int_0^{+\infty} (1 - F_{Y_n}(t)) dt.$$

4) a) Montrer, grâce au changement de variable $u = 1 - e^{-t}$, que l'on a :

$$\forall x \in \mathbb{R}_+, \quad \int_0^x (1 - F_{Y_n}(t)) dt = \int_0^{1-e^{-x}} \frac{1 - u^n}{1 - u} du.$$

4) b) En déduire que

$$\int_0^x (1 - F_{Y_n}(t)) dt = \sum_{k=1}^n \frac{(1 - e^{-x})^k}{k},$$

puis donner $E(Y_n)$ sous forme de somme.

5) On pose $Z_n = Y_n - \ln n$.

5) a) On rappelle que `rd.exponential(1, n)` simule n variables aléatoires indépendantes suivant toutes la loi exponentielle de paramètre 1.

Compléter la déclaration de fonction Python suivante afin qu'elle simule la variable aléatoire Z_n .

Python

```
import numpy as np

import numpy.random as rd

def simul_Z(n):
    x = rd.exponential(1, n)
    Z = ...
    return Z
```

`np.max(x)` renvoie le maximum des valeurs du tableau `x` et `np.log(n)` le logarithme népérien de n .

5) b) Voici deux scripts, le second utilisant la fonction `simul_Z` de la question précédente.

Script (1)

Python

```
import numpy as np

import numpy.random as rd

import matplotlib.pyplot as plt

V = rd.exponential(1, 10000)
W = -np.log(V)
s = np.linspace(0, 10, 11)
plt.hist(W, bins=s, weights=np.ones(10000)/10000,
         color="white", edgecolor="black")
plt.show()
```

Script (2)

Python

```
import numpy as np

import matplotlib.pyplot as plt

n = int(input("Entrez la valeur entière strictement positive de n : "))
Z = []

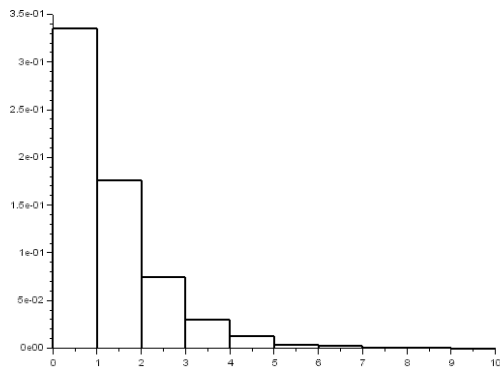
for k in range(10000):
    Z.append(simul_Z(n))

s = np.linspace(0, 10, 11)
plt.hist(Z, bins=s, weights=np.ones(10000)/10000,
         color="white", edgecolor="black")
plt.show()
```

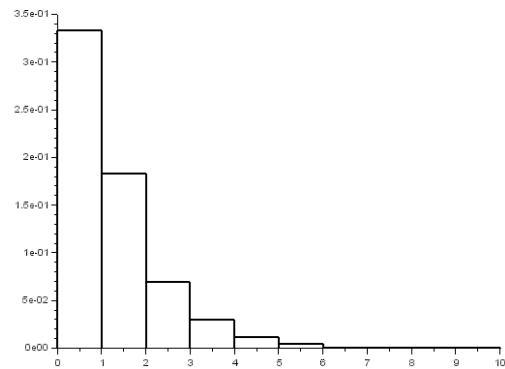
`np.ones(m)` crée un tableau de m coefficients égaux à 1. `plt.hist(T, bins=s, weights=p, ...)` trace un histogramme des observations T , délimité par les bornes du tableau s , chaque observation recevant le poids correspondant du tableau p . Ici, les classes ont une largeur de 1 et les hauteurs sont les effectifs de classe divisés par 10000; les observations hors de l'intervalle représenté ne sont pas renormalisées. Les options `color="white"` et `edgecolor="black"` donnent des rectangles blancs de contour noir. `Z.append(v)` ajoute la valeur v à la liste Z .

Chacun de ces scripts simule 10000 variables indépendantes, regroupe les valeurs renvoyées en 10 classes, qui sont les intervalles $[0, 1]$, $]1, 2]$, $]2, 3]$, ..., $]9, 10]$, et trace l'histogramme correspondant : la largeur de chaque rectangle est égale à 1 et sa hauteur est proportionnelle à l'effectif de la classe. La convention Python inclut la borne gauche plutôt que la borne droite, sauf dans la dernière classe ; cette différence aux bornes ne modifie pas les probabilités pour les variables à densité considérées.

Le script (1), dans lequel les variables aléatoires suivent la loi de Gumbel, loi suivie par W , renvoie l'histogramme (1) ci-dessous, alors que le script (2), dans lequel les variables aléatoires suivent la même loi que Z_n , renvoie l'histogramme (2) ci-dessous, pour lequel on a choisi $n = 1000$.



Histogramme (1)



Histogramme (2) pour n = 1000

Quelle conjecture peut-on émettre quant au comportement de la suite des variables aléatoires (Z_n) ?

6) On note F_{Z_n} la fonction de répartition de Z_n .

6) a) Justifier que, pour tout réel x , on a : $F_{Z_n}(x) = F_{Y_n}(x + \ln n)$.

6) b) Déterminer explicitement $F_{Z_n}(x)$.

6) c) Montrer que, pour tout réel x , on a :

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

6) d) Démontrer le résultat conjecturé à la question 5) b).

Problème

Partie 1 : étude d'une variable aléatoire.

Les sommets d'un carré sont numérotés 1, 2, 3 et 4 de telle façon que les côtés du carré relient le sommet 1 au sommet 2, le sommet 2 au sommet 3, le sommet 3 au sommet 4 et le sommet 4 au sommet 1.

Un mobile se déplace aléatoirement sur les sommets de ce carré selon le protocole suivant : au départ, c'est-à-dire à l'instant 0, le mobile est sur le sommet 1 ; lorsque le mobile est à un instant donné sur un sommet, il se déplace à l'instant suivant sur l'un quelconque des trois autres sommets, et ceci de façon équiprobable.

Pour tout n de $\mathbb{N}$, on note X_n la variable aléatoire égale au numéro du sommet sur lequel se situe le mobile à l'instant n . D'après le premier des deux points précédents, on a donc $X_0 = 1$.

1) Donner la loi de X_1 , ainsi que l'espérance $E(X_1)$ de la variable aléatoire X_1 .

On admet pour la suite que la loi de X_2 est donnée par :

$$P(X_2 = 1) = \frac{1}{3}, \quad P(X_2 = 2) = P(X_2 = 3) = P(X_2 = 4) = \frac{2}{9}.$$

2) Pour tout entier naturel n supérieur ou égal à 2, donner, en justifiant, l'ensemble des valeurs prises par X_n .

- 3) a) Utiliser la formule des probabilités totales pour établir que, pour tout entier naturel n supérieur ou égal à 2, on a :

$$P(X_{n+1} = 1) = \frac{1}{3}(P(X_n = 2) + P(X_n = 3) + P(X_n = 4)).$$

- 3) b) Vérifier que cette relation reste valable pour $n = 0$ et $n = 1$.

- 3) c) Justifier que, pour tout n de $\mathbb{N}$, on a

$$P(X_n = 1) + P(X_n = 2) + P(X_n = 3) + P(X_n = 4) = 1 \text{ et en déduire l'égalité :}$$

$$\forall n \in \mathbb{N}, \quad P(X_{n+1} = 1) = -\frac{1}{3}P(X_n = 1) + \frac{1}{3}.$$

- 3) d) Établir alors que : $\forall n \in \mathbb{N}, \quad P(X_n = 1) = \frac{1}{4} + \frac{3}{4} \left(-\frac{1}{3}\right)^n.$

- 4) a) En procédant de la même façon qu'à la question précédente, montrer que l'on a :

$$\forall n \in \mathbb{N}, \quad P(X_{n+1} = 2) = \frac{1}{3}(P(X_n = 1) + P(X_n = 3) + P(X_n = 4)).$$

- 4) b) En déduire une relation entre $P(X_{n+1} = 2)$ et $P(X_n = 2)$.

- 4) c) Montrer enfin que : $\forall n \in \mathbb{N}, \quad P(X_n = 2) = \frac{1}{4} - \frac{1}{4} \left(-\frac{1}{3}\right)^n.$

- 5) On admet que, pour tout entier naturel n , on a :

$$P(X_{n+1} = 3) = -\frac{1}{3}P(X_n = 3) + \frac{1}{3}, \quad P(X_{n+1} = 4) = -\frac{1}{3}P(X_n = 4) + \frac{1}{3}.$$

En déduire sans calcul que :

$$\forall n \in \mathbb{N}, \quad P(X_n = 3) = P(X_n = 4) = \frac{1}{4} - \frac{1}{4} \left(-\frac{1}{3}\right)^n.$$

- 6) Déterminer, pour tout entier naturel n , l'espérance $E(X_n)$ de la variable aléatoire X_n .

Partie 2 : calcul des puissances d'une matrice A .

Pour tout n de $\mathbb{N}$, on considère la matrice ligne de $\mathcal{M}_{1,4}(\mathbb{R})$:

$$U_n = \left(P(X_n = 1) \quad P(X_n = 2) \quad P(X_n = 3) \quad P(X_n = 4) \right).$$

- 7) a) Montrer, grâce à certains résultats de la partie 1, que, si l'on pose

$$A = \frac{1}{3} \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

on a :

$$\forall n \in \mathbb{N}, \quad U_{n+1} = U_n A.$$

- 7) b) Établir par récurrence que : $\forall n \in \mathbb{N}, \quad U_n = U_0 A^n$.

- 7) c) En déduire la première ligne de A^n .

- 8) Expliquer comment choisir la position du mobile au départ pour trouver les trois autres lignes de la matrice A^n , puis écrire ces trois lignes.

Partie 3 : une deuxième méthode de calcul des puissances de A .

On considère les matrices I et J suivantes :

$$I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad J = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

- 9) Déterminer les réels a et b tels que $A = aI + bJ$.
- 10) a) Calculer J^2 puis établir que, pour tout entier naturel k non nul, on a : $J^k = 4^{k-1}J$.
- 10) b) À l'aide de la formule du binôme de Newton, en déduire, pour tout entier naturel n non nul, l'expression de A^n comme combinaison linéaire de I et J .
- 10) c) Vérifier que l'expression trouvée reste valable pour $n = 0$.

Partie 4 : informatique.

- 11) a) Compléter le script Python suivant pour qu'il affiche les 100 premières positions, autres que celle d'origine, du mobile dont le voyage est étudié dans ce problème, ainsi que le nombre n de fois où il est revenu sur le sommet numéroté 1 au cours de ses 100 premiers déplacements. On pourra utiliser `np.sum`.

Python

```
import numpy as np

import numpy.random as rd

A = np.array(...)/3
```

```
x = np.zeros(100, dtype=int)
position = 1
for k in range(100):
    u = rd.random()
    j = 0
    cumul = A[position-1, j]
    while u >= cumul:
        j = j + 1
        cumul = cumul + A[position-1, j]
    position = j + 1
    x[k] = position
n = ...
print(x)
print(n)
```

`np.zeros(100, dtype=int)` crée un tableau de 100 entiers initialisés à zéro. `rd.random()` simule une variable uniforme sur $[0, 1[$. À chaque étape, les cumuls des probabilités de la ligne de transition de la position courante découpent cet intervalle; la boucle sélectionne le sommet suivant à l'aide du tirage u . Les indices de tableau commencent à 0, tandis que les sommets sont numérotés de 1 à 4. `np.sum` additionne les valeurs d'un tableau et peut compter les valeurs vraies d'une comparaison.

- 11) b) Après avoir exécuté cinq fois ce script, les réponses concernant le nombre de fois où le mobile est revenu sur le sommet numéroté 1 sont $n = 23$, $n = 28$, $n = 23$, $n = 25$ et $n = 26$. En quoi est-ce normal?