Devoir surveillé n° 5
22 février 2025
Durée de l'épreuve : 3h

Le devoir comporte deux problèmes indépendants.
La calculatrice est autorisée. Il sera tenu compte de la rigueur, du soin et de la rédaction dans la notation. Les résultats seront soulignés ou encadrés.

Problème 1

On suppose que toutes les variables aléatoires présentées dans cet exercice sont définies sur le même espace probabilisé.

Partie A

Soit f la fonction définie sur $\mathbb{R}$ par :

$$\forall t \in \mathbb{R} \qquad f(t) = \begin{cases} \dfrac{1}{t^3} & \text{si } t \geqslant 1 \\ 0 & \text{si } -1 < t < 1 \\ -\dfrac{1}{t^3} & \text{si } t \leqslant -1. \end{cases}$$

1. Démontrer que la fonction f est paire. Tracer rapidement le graphe de cette fonction.
2. Justifier que l'intégrale $\displaystyle\int_1^{+\infty} f(t)\, dt$ converge et calculer sa valeur.
3. a) À l'aide d'un changement de variable, **montrer que** pour tout réel A strictement supérieur à 1, on a : $\displaystyle\int_{-A}^{-1} f(t)\, dt = \int_1^A f(u)\, du$.
   En déduire que l'intégrale $\displaystyle\int_{-\infty}^{-1} f(t)\, dt$ converge et donner sa valeur.
   b) Montrer que la fonction f est une densité de probabilité.
4. On considère une variable aléatoire X admettant f pour densité. On note F_X la fonction de répartition de X.
   a) Montrer que, pour tout réel x, on a :
   $$F_X(x) = \begin{cases} \dfrac{1}{2x^2} & \text{si } x \leqslant -1 \\ \dfrac{1}{2} & \text{si } -1 < x < 1 \\ 1 - \dfrac{1}{2x^2} & \text{si } x \geqslant 1. \end{cases}$$
   b) Démontrer que X admet une espérance, puis que cette espérance est nulle.
   c) La variable aléatoire X admet-elle une variance ?
5. Soit Y la variable aléatoire définie par $Y = |X|$.
   a) Donner la fonction de répartition de Y, et montrer que Y est une variable aléatoire à densité.
   b) Montrer que Y admet pour densité la fonction f_Y définie par :
   $$f_Y(x) = \begin{cases} \dfrac{2}{x^3} & \text{si } x \geqslant 1 \\ 0 & \text{sinon.} \end{cases}$$
   c) Montrer que Y admet une espérance et la calculer.

Partie B

Soit D une variable aléatoire prenant les valeurs -1 et 1 avec équiprobabilité, indépendante de la variable aléatoire Y.
Soit T la variable aléatoire définie par $T = DY$.

1. a) Déterminer la loi de la variable $Z = \dfrac{D+1}{2}$. En déduire l'espérance et la variance de D.
   b) Justifier que T admet une espérance et préciser sa valeur.
   c) Montrer que pour tout réel x, on a :
   $$P(T \leqslant x) = \frac{1}{2}\ P(Y \leqslant x) + \frac{1}{2}\ P(Y \geqslant -x).$$
   d) En déduire la fonction de répartition de T.
2. Soit U une variable aléatoire suivant la loi uniforme sur $[0,\ 1[$ et V la variable aléatoire définie par : $\quad V = \dfrac{1}{\sqrt{1-U}}$.
   a) Rappeler la fonction de répartition de U.
   b) Déterminer la fonction de répartition de V et vérifier que les variables V et Y suivent la même loi.
3. a) Écrire une fonction `D(n)` en langage python, , qui prend un entier $n \geqslant 1$ en argument, et renvoie une liste contenant n réalisations de la variable aléatoire D.

b) On considère la fonction suivante :

```
def mystere(n):
    L=D(n)
    R=[]
    for x in L:
        R.append(x/np.sqrt(1-rd.random()))
    return np.mean(R)
```

À la fin de l'exécution de cette fonction, de quelle variable aléatoire les valeurs de la liste `R` sont-elles une simulation ? Pour `n` assez grand, quelle sera la valeur renvoyée ?

Problème 2

Un industriel cherche à optimiser son processus de cuisson des petits pois dans une cuve chauffée. Le cahier des charges demande à ce que les petits pois cuisent 10 minutes à une température comprise entre 90°C et 92°C. L'industriel utilise la puissance de chauffe pour commander les températures.

Partie 1 : Modélisation du processus

Dans cette partie, on suppose que la température des petits pois et la température de l'eau de la cuve sont uniformes et que la cuisson peut donc être modélisée par le système d'équation différentielles suivant :

$$\begin{cases} m_1C_1\dfrac{\mathrm{d}T_1}{\mathrm{d}t} &= h_1S_1(T_2-T_1) \\[2ex] m_2C_2\dfrac{\mathrm{d}T_2}{\mathrm{d}t} &= h_1S_1(T_1-T_2)+h_2S_2(T_\infty-T_2)+u \end{cases} \tag{1}$$

où m_1 et m_2 sont les masses respectives des petits pois et de l'eau dans la cuve (en kg), C_1 et C_2 sont les capacités thermiques massiques respectives du petit pois et de l'eau (en $J\cdot K^{-1}\cdot kg^{-1}$), h_1 et h_2 sont les coefficients de transfert thermique entre l'eau et les petits pois d'une part, et entre l'eau et l'extérieur d'autre part (en $W\cdot m^{-2}\cdot K^{-1}$), S_1 et S_2 sont respectivement la surface totale des petits pois et la surface d'échange entre l'eau et l'extérieur (en m^2), T_1, T_2 et T_∞ sont les températures respectives des petits pois, de l'eau et de l'extérieur (en K) et u est la puissance de la résistance chauffante utilisée pour chauffer la cuve (en W). Cette puissance de chauffe u constitue le moyen d'action de l'industriel sur les températures T_1 et T_2 et est donc appelé la **commande**.

On suppose que toutes ces quantités sont constantes au cours du temps, en dehors des températures T_1 et T_2 et de la commande u. En particulier, la température extérieur est constante et vaut $T_\infty = 20$°C. Le temps t sera exprimé en **minutes**.

Commande en boucle ouverte (cas u constant)

1. On note $x_1 = T_1 - T_\infty$ et $x_2 = T_2 - T_\infty$. Établir les équation différentielles vérifiées par x_1 et x_2. On les écrira sous la forme :

$$\begin{cases} \dfrac{\mathrm{d}x_1}{\mathrm{d}t} &= f_1(x_1,x_2,u) \\[2ex] \dfrac{\mathrm{d}x_2}{\mathrm{d}t} &= f_2(x_1,x_2,u) \end{cases}.$$

On admettra que l'on peut réécrire ces équations sous la forme :

$$\frac{\mathrm{d}x}{\mathrm{d}t} = Ax+ub, \tag{2}$$

avec $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $\dfrac{\mathrm{d}x}{\mathrm{d}t} = \begin{pmatrix} \dfrac{\mathrm{d}x_1}{\mathrm{d}t} \\ \dfrac{\mathrm{d}x_2}{\mathrm{d}t} \end{pmatrix}$, A une matrice de $\mathscr{M}_2(\mathbb{R})$, b une matrice de $\mathscr{M}_{2,1}(\mathbb{R})$ et $u\in\mathbb{R}$ constant.

Dans la suite du problème, on prendra

$$A = \begin{pmatrix} -5 & 5 \\ 4 & -6 \end{pmatrix} \qquad \text{et} \qquad b = \begin{pmatrix} 0 \\ 9 \end{pmatrix}.$$

2. On suppose qu'il existe $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ dans $\mathscr{M}_{2,1}(\mathbb{R})$ et u dans $\mathbb{R}$ tels que $AX+ub=0$.
   a) Montrer qu'alors $x_1 = x_2$.
   b) Interpréter
   c) Calculer la valeur de u si on considère que $x_1 = x_2 = 72$.
3. a) Calculer les valeurs propres de A.
   b) Montrer que A est diagonalisable.
   c) Donner une base de vecteurs propres de A. On choisira les premières coordonnées de ces vecteurs propres égales à 1.
   d) En déduire les matrices P et P^{-1} vérifiant $A = PDP^{-1}$ avec $D = \begin{pmatrix} -1 & 0 \\ 0 & -10 \end{pmatrix}$.
4. Soit x une solution de (2). On définit $z = P^{-1}x$, où $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$.
   a) Montrer que $\dfrac{\mathrm{d}z}{\mathrm{d}t} = P^{-1}\dfrac{\mathrm{d}x}{\mathrm{d}t}$.

b) En déduire que
$$\frac{\mathrm{d}z}{\mathrm{d}t} = Dz + u\beta, \tag{3}$$
où β est une matrice colonne à expliciter.

c) On suppose que $T_1(0) = 20$°C et $T_2(0) = 92$°C. Calculer $z(0)$.

d) Résoudre l'équation différentielle matricielle (3) en supposant que $u = 16$.
On écrira séparément les équations différentielles vérifiées par z_1 et z_2.

e) En utilisant la relation $x = Pz$, montrer que pour tout $t \geqslant 0$:
$$\begin{cases} x_1(t) &= 72 - 40\mathrm{e}^{-t} - 32\mathrm{e}^{-10t} \\ x_2(t) &= 72 - 32\mathrm{e}^{-t} + 32\mathrm{e}^{-10t} \end{cases}.$$

5. a) Étudier sur $\mathbb{R}^+$ la fonction x_2 définie à la question 4.e). Tracer sa courbe représentative.
On fera figurer sur le graphique tous les éléments importants.
b) Donner une interprétation du comportement de la fonction x_2.
c) On définit $t_R = \min\{t \geqslant 0 \ ; \ \forall t' \geqslant t, \ T_2(t') \geqslant 90\}$. Que représente ce temps t_R ? Le placer sur la courbe représentative de x_2. On rappelle que $x_2 = T_2 - T_\infty$.
d) Montrer que $\ln(16)$ minutes est une bonne approximation de t_R. Comment chaque valeur propre de A influence-t-elle t_R ?

Commande en boucle fermée (cas u variable)

Pour améliorer les performances, on décide de faire varier la puissance de chauffe en fonction de la température de l'eau en considérant, pour tout $t \geqslant 0$: $u(t) = 16 + k(72 - x_2(t))$, où $k \geqslant 0$ est une constante de réglage appelée **gain.**

6. Expliquer ce choix de fonction pour la puissance de chauffage. Quel est l'intérêt ?
7. Que vaut t_R pour $k = 0$?
8. On définit $w(t) = x(t) - \begin{pmatrix} 72 \\ 72 \end{pmatrix}$. Montrer que $\dfrac{\mathrm{d}w}{\mathrm{d}t} = (A - kbc^T)w$, où $c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ et c^T désigne sa transposée.
9. On note $\lambda_1(k)$ et $\lambda_2(k)$ les valeurs propres de $A - kbc^T$ telles que $\lambda_1(k) < \lambda_2(k)$.
a) Donner l'expression de $\lambda_1(k)$ et $\lambda_2(k)$ en fonction de k.
b) Montrer que λ_1 et λ_2 sont des fonctions décroissantes de k sur $\mathbb{R}^+$.
10. Quel est l'effet du gain k sur t_R ? Quels inconvénients à augmenter k peut-on envisager ?

Partie 2 : Étude informatique des commandes

Cette partie consiste à mettre en place des programme informatiques en lien avec la partie 1.

Les programmes sont à rédiger en langage Python. L'annexe comporte des rappels sur les commandes utiles. Avant chaque algorithme, on écrira brièvement le raisonnement suivi et la formule qu'il est censé calculer.

Recherche du minimum d'une fonction

Soit $f : t \mapsto 72 - 32\mathrm{e}^{-t} + 32\mathrm{e}^{-10t}$.

11. Écrire une fonction d'en-tête `f(t)` prenant en entrée t (un réel) et qui renvoie la valeur de $f(t)$.
12. On définit la fonction d'en-tête `minimum_f(N)` prenant en entrée N (un entier) par le code (incomplet) suivant :

```
1 def minimum_f(N):
2     Lt1=np.linspace(0,5,N)
3     Ly1=[]
4     for k in range(0,N):
5         Ly1.append(f(Lt1[k]))
6     ...
7     ...
8     return ...
```

Décrire en quelques mots ce que contiennent les variables `Lt1` et `Ly1` à l'issue des lignes 2 à 5.

13. Compléter sur la copie (avec autant de lignes que nécessaire) la fonction `minimum_f` pour qu'elle renvoie une liste composée des deux éléments suivants :
- une valeur approchée m du minimum de f sur $[0; 5]$;
- le temps t_m en lequel cette valeur approchée est atteinte (c'est-à-dire $m = f(t_m)$).

On veillera à n'écrire qu'une seule boucle après celle des lignes 4 et 5 et à ne pas utiliser la commande `min`.

14. On teste la fonction `minimum_f` sur différentes valeurs de N. Voici les résultats obtenus (tronqués à 4 chiffres après la virgule) :
- pour $N_1 = 20$, la fonction donne $m_1 = 49.7070$ et $t_{m_1} = 0.2631$;
- pour $N_2 = 20000$, la fonction donne $m_2 = 49.7012$ et $t_{m_2} = 0.2557$.

Il y a un différence entre t_{m_1} et t_{m_2}.

a) Cette différence était-elle prévisible et à quoi est-elle due ?
b) Est-ce t_{m_1} ou t_{m_2} qui devrait être le plus proche de la valeur exacte recherchée ?

15. On teste maintenant sur une troisième valeur située entre N_1 et N_2 : pour $N_3 = 50$, la fonction donne $m_3 = 49.9370$ et $t_{m_3} = 0.3061$. Commenter ce résultat.

Recherche du point d'annulation d'une fonction

Pour une fonction $F : I \to \mathbb{R}$ donnée, avec I un intervalle de $\mathbb{R}$, on s'intéresse à la résolution approchée de l'équation $F(x) = 0$. On suppose que F s'annule en un point appelé $x^* \in I$ et que F est suffisamment régulière pour que les calculs suivants soient bien définis. L'objectif est d'obtenir une valeur approchée de x^*.

On utilise la méthode de Newton, qui consiste à définir la suite $(u_n)_{n\in\mathbb{N}}$ par :

$$\begin{cases} u_0 \text{ à choisir} \\ u_{n+1} = u_n - \dfrac{F(u_n)}{F'(u_n)} \quad \text{pour } n \geqslant 0. \end{cases}$$

On admet que, sous des hypothèses non précisées ici, cette suite converge vers x^* et que, s'il y a deux solutions à l'équation $F(x) = 0$, alors selon le choix de u_0 la suite va converger vers une solution ou vers l'autre.

16. Soit $e > 0$. À l'aide d'une boucle `while`, écrire une fonction d'en-tête `Newt(F,G,u0,e)` qui calcule les termes de la suite en s'arrêtant au premier terme u_{n_0} vérifiant $|F(u_{n_0})| < e$ (en supposant qu'un tel n_0 existe). Les arguments d'entrée sont : F une fonction, G une fonction qui correspond à F', u_0 le premier terme de la suite et e. La fonction doit renvoyer u_{n_0}.
17. Par rapport à l'objectif de résoudre de façon approchée l'équation $F(x) = 0$, expliquer à quoi correspond la quantité e utilisée dans l'écriture de `Newt`. Parmi les choix suivants pour e, indiquer (avec justification succincte) celui qui semble le plus pertinent pour obtenir la meilleure précision sur la valeur approchée : $e_1 = 100$, $e_2 = 1$, $e_3 = 10^{-8}$.
18. Toujours par rapport à l'objectif de résoudre de façon approchée l'équation $F(x) = 0$, justifier s'il est pertinent ou pas d'effectuer la modification suivante sur la fonction `Newt` : renvoyer le premier terme vérifiant $F(u_n) = 0$.
19. À la question 16., on a supposé l'existence d'un n_0 vérifiant $|F(u_{n_0})| < e$. Que se passe-t-il dans la fonction `Newt` si un tel n_0 n'existe pas ?
20. On souhaite modifier la fonction `Newt` de manière à fixer un nombre maximal d'itérations pour la boucle `while`. Écrire une nouvelle fonction d'en-tête `NewtS(F,G,u0,e,nder)`, basée sur la fonction `Newt`, telle que :
    - s'il existe un terme u_{n_0} tel que $|F(u_{n_0})| < e$ avec $n_0 \leqslant n_{\text{der}}$, la fonction renvoie u_{n_0} ;
    - si le nombre d'itérations dépasse (strictement) la valeur n_{der}, la boucle est stoppée et la fonction renvoie le booléen `False`.
21. Expliquer comment la méthode de Newton peut être utilisée pour trouver une valeur approchée du temps t_R défini à la question 5.c) de la partie 1.
22. On définit les fonctions suivantes (où `f` a été définie à la question 11.) :

```
def F1(t):
    return f(t)-70
def G1(t):
    return 32*np.exp(-t)-320*np.exp(-10*t)
```

puis on exécute pour `e=10**-14` et `nder=9` les deux appels suivants :

- `NewtS(F1,G1,2,e,nder)` qui renvoie `2.7725887222252261` ;
- `NewtS(F1,G1,0.1,e,nder)` qui renvoie `0.0072246696343453579`.

Interpréter les résultats en lien avec la partie 1.

Annexes

On suppose que le module `numpy` est importé via `import numpy as np`. Dans le tableau les variables `a` et `b` sont des réels et `N` est un entier.

Python	Interprétation
`np.exp(a)`	Renvoie $\exp(a)$
`np.linspace(a,b,N)`	Renvoie un tableau à une dimension contenant N valeurs équiréparties dans $[a;b]$; ces valeurs sont les $t_k = a + \dfrac{b-a}{N-1}k$ pour $k \in [\![0; N-1]\!]$.
`np.mean(L)`	renvoie la moyenne d'une liste ou d'un tableau `L`.

On suppose que le module `random` est importé via `import random as rd`. Dans le tableau les variables `a` et `b` sont entiers et `N` est un entier.

Python	Interprétation
`rd.random()`	Simule une réalisation d'une variable $X \hookrightarrow \mathscr{U}([0,\ 1])$.
`rd.randint(a,b)`	Simule une réalisation d'une variable $X \hookrightarrow \mathscr{U}([\![a,\ b]\!])$.