Devoir surveillé n$^{\text{o}}$ 1

21 septembre 2024

Durée de l'épreuve : 2h

Le devoir comporte un exercice et un problème indépendants. La calculatrice n'est pas autorisée. Il sera tenu compte de la rigueur, du soin et de la rédaction dans la notation. Les résultats seront soulignés ou encadrés.

Exercice : Étude d'une fonction de deux variables

On considère la fonction f définie, pour tout couple (x,y) du pavé ouvert $]0,+\infty[\times]0,+\infty[$, par :

$$f(x,y) = (x+y)\left(\frac{1}{x}+\frac{1}{y}\right).$$

1. Montrer que, pour tout couple (x,y) de $]0,+\infty[\times]0,+\infty[$, on a :

$$f(x,y) = 2 + \frac{y}{x} + \frac{x}{y} \quad \text{et} \quad f(x,y) = \frac{(x+y)^2}{xy}.$$

2. Justifier rigoureusement que f est de classe $\mathscr{C}^2$ sur $]0,+\infty[\times]0,+\infty[$.
3. a) Déterminer les dérivées partielles d'ordre 1 de f.
   b) Montrer que f possède une infinité de points critiques et les déterminer.
4. a) Comparer les réels $(x+y)^2$ et $4xy$.
   b) En déduire que f admet sur $]0,+\infty[\times]0,+\infty[$ un minimum global en tous ses points critiques et donner sa valeur.
5. La fonction f admet-elle un maximum local ou global ?
6. Soit g la fonction définie pour tout (x,y) de $]0,+\infty[\times]0,+\infty[$, par :

$$g(x,y) = 2\ln(x+y) - \ln x - \ln y.$$

   Montrer que : $\forall (x,y) \in]0,+\infty[\times]0,+\infty[$, $g(x,y) \geqslant 2\ln 2$.

PROBLÈME : Étude d'un modèle d'évolution du nombre de feuilles d'une plante

Ce problème comporte deux parties indépendantes. Les questions 4, 7 et 9c. sont des questions d'informatique. Elles peuvent être traitées indépendamment des questions mathématiques qui les précèdent.

Dans le domaine de l'écophysiologie végétale on s'intéresse aux différentes phases du développement des plantes afin de mieux contrôler les rendements de culture. Plus précisément, on s'intéresse à l'évolution du nombre de feuilles d'une plante en fonction du "temps thermique". Ce dernier correspond à la somme cumulée des températures au cours des différents jours de l'expérience. De telles mesures permettent de mettre en évidence trois phases de développement appelées phases de rosette, d'élongation et de floraison. La modélisation la plus souvent utilisée consiste à supposer que l'évolution du nombre de feuilles en fonction du temps thermique se comporte comme une fonction affine au cours de chacune des trois phases précédentes (cf. figure 1).

L'objectif est alors de mettre en place des méthodes automatiques permettant de détecter les instants séparant d'une part les phases de rosette et d'élongation (T1) et d'autre part les phases d'élongation et de floraison (T2). Il est, en effet, intéressant de voir si ces instants ont tendance à changer en fonction de certaines conditions expérimentales auxquelles les plantes pourraient être soumises.

Partie I

Dans cette partie, on **se focalise sur l'une des trois phases précédentes** et on s'intéresse à l'ajustement affine que l'on obtient à partir du nuage de points $((x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n))$, n étant un entier naturel strictement plus grand que 1, en utilisant le critère des moindres carrés où l'on supposera que les x_i sont distincts. Dans la notation (x_i,y_i), x_i correspond au temps thermique de la $i$ème observation et y_i correspond au nombre de feuilles de la $i$ème observation (cf. figure 2).

Dans cette partie on note $\bar{x} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} x_i$ et $\bar{y} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} y_i$.

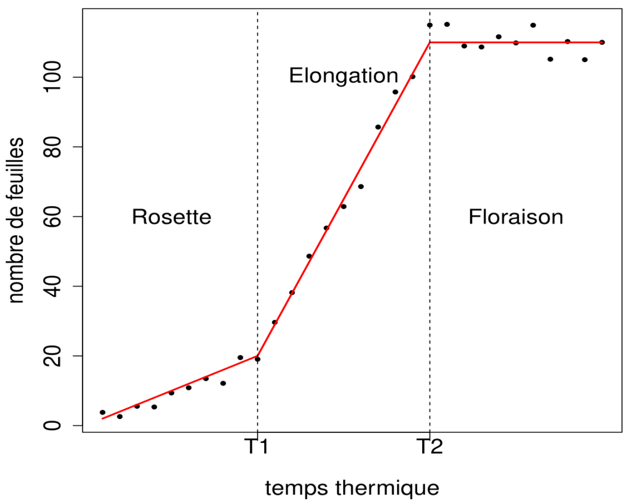

FIGURE 1 – Schéma illustrant l'évolution du nombre de feuilles d'une plante en fonction du temps thermique ; y sont représentés les valeurs expérimentales (' • ') et les ajustements affines dans chacune des trois phases.

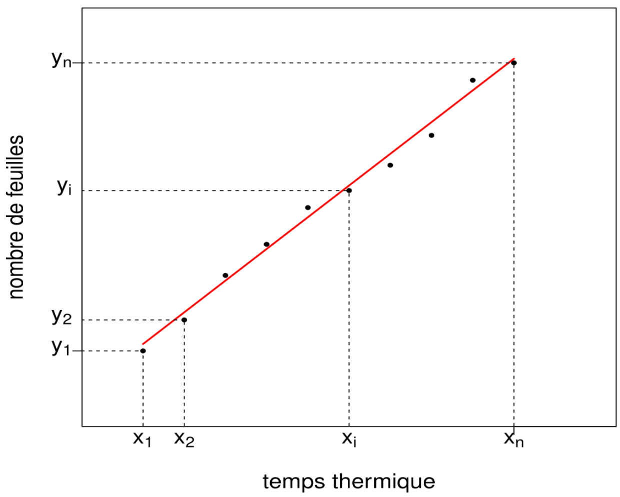

FIGURE 2 – Schéma illustrant l'évolution du nombre de feuilles en fonction du temps thermique durant l'une des trois phases.

1. Montrer que : $\displaystyle\sum_{i=1}^{n} x_i^2 - \bar{x}\sum_{i=1}^{n} x_i = \sum_{i=1}^{n}(x_i - \bar{x})^2$.
2. On définit la fonction $F : \mathbb{R}^2 \to \mathbb{R}$ par :

$$\forall (a,b) \in \mathbb{R}^2, \quad F(a,b) = \sum_{i=1}^{n}(y_i - a - bx_i)^2.$$

   a) Justifier rapidement que F est de classe $\mathscr{C}^1$ sur $\mathbb{R}^2$.
   b) Déterminer les dérivées partielle d'ordre 1 de F.
   c) Montrer que le seul point en lequel la fonction F est susceptible d'admettre un extremum est le couple $(\hat{a}, \hat{b}) \in \mathbb{R}^2$ où

$$\hat{a} = \bar{y} - \hat{b}\bar{x} \quad \text{et} \quad \hat{b} = \frac{\displaystyle\sum_{i=1}^{n} y_i(x_i - \bar{x})}{\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

3. On admet que le couple $(\hat{a}, \hat{b})$ obtenu réalise un minimum global.
   Expliquer le fonctionnement du critère des moindre carrés tout en faisant le lien avec les notations de l'énoncé.
4. On souhaite automatiser les calculs de $\hat{a}$ et $\hat{b}$ lorsqu'un grand nombre de valeurs (x_i, y_i) sont à disposition. On utilise Python, en considérant que les valeurs $x_1, \ldots, x_n$ sont stockées dans une liste `x` et que les valeurs $y_1, \ldots, y_n$ sont stockées dans une liste `y`.
   Consigne : Chaque algorithme, à écrire en Python, doit être précédé d'une phrase expliquant le raisonnement suivi pour l'écrire.
   a) Écrire une fonction `moy` qui prend en entrée `L`, une liste de nombres (non vide), et qui renvoie la moyenne des valeurs de `L`.
      On s'interdira l'appel à une fonction déjà programmée dans Python qui calcule la moyenne des valeurs d'une liste.
   b) Soit `L` une liste de longueur $n \geqslant 1$. Indiquer, sans justification, si les affirmations suivantes sont correctes :
      (i) Les éléments de `L` sont numérotés de 0 à $n-1$ (inclus) ;
      (ii) Les éléments de `L` sont numérotés de 1 à n (inclus) ;
      (iii) Les éléments de `L` sont numérotés de 0 à $n+1$ (inclus) ;
      (iv) Il y a n éléments dans `L` ;
      (v) Il y a $n+1$ éléments dans `L`.
   c) Écrire une fonction `Bchap` qui prend en entrée `x` et `y`, deux listes de nombres de même longueur $n \geqslant 2$, et qui renvoie la valeur de $\hat{b}$ associée.
   d) Écrire une fonction `Achap` qui prend en entrée `x` et `y`, deux listes de nombres de même longueur $n \geqslant 2$, et qui renvoie la valeur de $\hat{a}$ associée.

Partie II

5. Soit u une fonction connue de classe $\mathscr{C}^4$ sur $\mathbb{R}$ que l'on observe aux n temps thermiques : $x_1, x_2, \ldots, x_n$ que l'on supposera deux à deux distincts. On notera u_i la valeur de u en x_i :
$$u_i = u(x_i), \quad 1 \leqslant i \leqslant n.$$
   On supposera de plus que les données sont récoltées à des intervalles de temps réguliers $h > 0$ avec $h = x_{i+1} - x_i$, $1 \leqslant i \leqslant n-1$.
   a) Écrire la formule de Taylor-Young à l'ordre 4 au voisinage de 0 pour une fonction f de classe $\mathscr{C}^4$ sur $\mathbb{R}$.
   b) Appliquer la formule de la question précédente à la fonction $f : t \mapsto u(x_i + t)$ et en déduire une égalité du type :
$$u(x_i + t) = u(x_i) + \ldots + o(\ldots).$$
   c) En appliquant le développement précédent à $t = h$ et $t = -h$, en déduire que pour $2 \leqslant i \leqslant n-1$,
$$u''(x_i) = \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + e(h),$$
      et expliciter le terme $e(h)$.
6. À partir de maintenant on va travailler sur un modèle simplifié où les valeurs expérimentales y_i, $1 \leqslant i \leqslant n$, sont portées par la fonction linéaire par morceaux de la figure 1 et récoltées à des intervalles de temps réguliers h ($h = x_{i+1} - x_i$, $1 \leqslant i \leqslant n-1$). On aura donc :
$$\begin{cases} y_i = a_1 + b_1 x_i & \text{si } x_i \leqslant T_1 \\ y_i = a_2 + b_2 x_i & \text{si } T_1 \leqslant x_i \leqslant T_2 \\ y_i = a_3 + b_3 x_i & \text{si } x_i \geqslant T_2 \end{cases}$$
   Les valeurs des paramètres a_j et b_j, $1 \leqslant j \leqslant 3$ sont supposées inconnues.
   On notera $D_i = \dfrac{y_{i-1} - 2y_i + y_{i+1}}{h^2}$, pour $2 \leqslant i \leqslant n-1$.
   a) Trouver la valeur de D_i lorsque $x_{i+1} \leqslant T_1$ en utilisant la question 5.c. sans faire de calcul.
   b) Proposer une méthode pour détecter les deux instants séparant les différentes phases.
7. Cette question ne nécessite pas d'avoir obtenu les résultats qui précèdent pour y répondre.
   a) Que renvoie l'algorithme suivant, appliqué sur une liste `L` non vide :

```python
def Mystere(L) :
    """(L : liste non vide)."""
    n = len(L)
    p = 0
    m = L[0]
    for k in range(1,n) :
        if m < L[k] :
            m = L[k]
            p = k
    return p
```

b) On dispose dorénavant d'une liste L de nombres, de longueur supérieure ou égale à deux, dont les éléments sont distincts deux à deux. On souhaite renvoyer les positions des deux plus grands éléments de L.

(i) On suppose que l'on dispose d'un algorithme de tri. Proposer une démarche pour répondre à la question.

(ii) On souhaite désormais répondre sans utiliser d'algorithme de tri mais en utilisant la fonction `Mystere`. On commence par calculer `p1=Mystere(L)` et ensuite, on cherche le plus grand élément `L[p2]` de `L` tel que `p2` est différent de `p1`. Compléter les trois lignes manquantes de la fonction suivante, afin qu'elle renvoie la liste `[p1,p2]` des positions des deux plus grands éléments de `L`.

```python
def DeuxMax(L) :
    """...""" #à compléter
    n = len(L)
    p1 = Mystere(L)
    p2 = 0
    for k in range(1,n) :
        ... #à compléter
        ... #à compléter
    return([p1,p2])
```

8. Dans la pratique les valeurs expérimentales sont bruitées et on a :

$$\begin{cases} y_i = a_1 + b_1 x_i + \varepsilon_i & \text{si } x_i \leqslant T_1 \\ y_i = a_2 + b_2 x_i + \varepsilon_i & \text{si } T_1 \leqslant x_i \leqslant T_2 \\ y_i = a_3 + b_3 x_i + \varepsilon_i & \text{si } x_i \geqslant T_2 \end{cases}$$

où ε_i désigne un terme d'erreur ayant la forme particulière suivante $\varepsilon_i = (-1)^i \varepsilon$, où $\varepsilon > 0$.

a) Calculer D_i dans le cas où $x_{i+1} \leqslant T_1$.

b) En supposant que ε est "suffisamment petit", proposer une méthode pour détecter les deux instants séparant les différentes phases.

9. Pour éliminer le bruit on peut faire appel à une méthode dite de lissage. Elle consiste à modifier les données initiales qui seront notées $(y_i^0)_{1\leqslant i \leqslant n}$ en répétant un certain nombre de fois la récurrence suivante pour $k \geqslant 0$:

$$\begin{cases} y_i^{k+1} = y_i^k + \alpha \dfrac{y_{i-1}^k - 2y_i^k + y_{i+1}^k}{h^2}\ , \ 2 \leqslant i \leqslant n-1, \\ y_1^{k+1} = y_1^k, \\ y_n^{k+1} = y_n^k, \end{cases}$$

où $\alpha \in \mathbb{R}$.

On se limite ici au premier intervalle de temps (avant T_1), cf. figure 2. Plus précisément,

$$y_i^0 = a_1 + b_1 x_i + \varepsilon_i, 1 \leqslant i \leqslant n,$$

où $\varepsilon_i = (-1)^i \varepsilon$, avec $\varepsilon > 0$.

a) Montrer qu'une itération de lissage appliquée au vecteur $(y_i^0)_{1<i<n}$ donne un résultat de la forme :

$$a_1 + b_1 x_i + C(\alpha, h)\varepsilon_i,$$

où $C(\alpha, h)$ dépend de α et de h.

b) En déduire un intervalle pour α tel que cette méthode permette de réduire le bruit. Existe-t-il une valeur de α permettant de supprimer ce bruit ?

c) Écrire en Python une fonction `Lisse` qui applique une itération de lissage, `Lisse` prend en entrée `Y`, `a` et `h` où `Y` est une liste de nombres contenant les valeurs $(y_i^k)_{1\leqslant i\leqslant n}$ (pour un certain k), `a` est la valeur de α et `h` est la valeur de h ; elle renvoie les valeurs $(y_i^{k+1})_{1\leqslant i\leqslant n}$ dans une liste.

d) Dans la pratique, sur le problème complet de la figure 1 *i.e.* lorsque l'on ne se limite pas au premier intervalle de temps, on applique plusieurs itérations de lissage. On obtient alors le graphe de la figure 3.

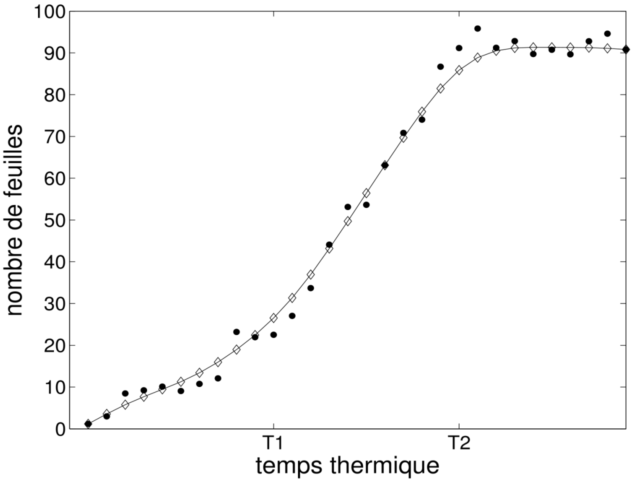

FIGURE 3 – Nombre de feuilles d'une plante en fonction du temps thermique ; y sont représentées les valeurs expérimentales initiales ('•') et les valeurs obtenues après 10 itérations de lissage (' ◇').

En s'inspirant des questions précédentes, proposer une méthode pour estimer T1 et T2.

e) Cette méthode est-elle satisfaisante pour estimer T1 et T2 ?