D.M. de Mathématiques # 3

à rendre le Jeudi 04 Octobre 2018

Lycée Chaptal BCPST2 A

Banque : Agro/Véto

Banque « Agro-Véto »
A - 0716

MATHÉMATIQUES

Modélisation Mathématique et Informatique

Durée : 3 heures 30 minutes

Chaque 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 doit alerter au plus tôt le chef de centre qui contrôlera et éventuellement remplacera le sujet.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

L'usage d'une calculatrice est autorisé pour cette épreuve. Les questions d'informatique devront être rédigées en langage Python exclusivement.

Cette épreuve s'intéresse modestement à la modélisation de phénomènes de transport, omniprésents dans notre vie : trafic routier, déplacement de lymphocytes dans le sang, invasions d'espèces ...

On pourra utiliser dans tout le sujet et sans démonstration, à condition d'y faire explicitement référence, les règles de dérivation des fonctions composées suivantes si f, a, b sont toutes des fonctions de classe $\mathcal{C}^1$:

— « **Règle 1.a.** »

$$\frac{\partial}{\partial t}\left(f(a(t,x),b(t,x))\right) = \frac{\partial f}{\partial t}(a(t,x),b(t,x))\frac{\partial a}{\partial t}(t,x) + \frac{\partial f}{\partial x}(a(t,x),b(t,x))\frac{\partial b}{\partial t}(t,x).$$

— « **Règle 1.b.** »

$$\frac{\partial}{\partial x}\left(f(a(t,x),b(t,x))\right) = \frac{\partial f}{\partial t}(a(t,x),b(t,x))\frac{\partial a}{\partial x}(t,x) + \frac{\partial f}{\partial x}(a(t,x),b(t,x))\frac{\partial b}{\partial x}(t,x).$$

De la même façon, on pourra utiliser dans tout le sujet et sans démonstration, à condition d'y faire explicitement référence, la règle de dérivation sous le signe intégral « **Règle 2.** » lorsque $f \in \mathcal{C}^1(\mathbf{R}^2, \mathbf{R})$:

$$\forall t \in \mathbf{R}, \quad \forall (a,b) \in \mathbf{R}^2, \qquad \frac{d}{dt}\left(\int_a^b f(t,x)dx\right)(t) = \int_a^b \frac{\partial f}{\partial t}(t,x)dx.$$

Le candidat est averti que ces règles devront être utilisées de nombreuses fois dans la suite.

Les différentes parties de cette épreuve sont indépendantes.

Partie 1 : Équation de transport uni-dimensionnelle

Dans cette partie, on considère l'équation dite *de transport* suivante, posée en une dimension spatiale et en temps positifs :

$$\frac{\partial u}{\partial t}(t,x) + c\frac{\partial u}{\partial x}(t,x) = 0, \qquad (t,x) \in \mathbf{R}^+ \times \mathbf{R}. \tag{1}$$

La constante $c \in \mathbf{R}^+$ est appelée *vitesse*. La fonction inconnue u dépend du temps t et de la position dans l'espace x.

On admettra dans la suite que l'équation (1) admet alors des solutions définies sur $\mathbf{R}^+ \times \mathbf{R}$. On considère une fonction u_0 définie sur $\mathbf{R}$ et à valeurs réelles, de classe $\mathcal{C}^1$:

$$u_0 \in \mathcal{C}^1(\mathbf{R}, \mathbf{R}).$$

On définit alors la fonction u de la manière suivante :

$$\forall (t,x) \in \mathbf{R}^+ \times \mathbf{R}, \qquad u(t,x) = u_0(x - ct).$$

1. **1.1** Calculer les deux dérivées partielles $\dfrac{\partial u}{\partial t}$ et $\dfrac{\partial u}{\partial x}$ de la fonction u en fonction de u_0'.

1.2 Montrer que u est solution de (1).

2. Que vaut $u(0,x)$? Justifier le nom de « condition initiale »donnée à la fonction u_0.

3. Dans cette question (seulement), on choisit la donnée initiale suivante :

$$\forall x \in \mathbf{R}, \qquad u_0(x) = \frac{1}{2} - \frac{1}{\pi}\arctan(x),$$

et on fixe $c = 1$. Tracer dans un même repère $u_0(x)$, $u(1,x)$ et $u(2,x)$ en fonction de x (échelle : 1 unité = 5 cm). Justifier au vu de ce tracé le nom *équation de transport* donné à cette équation.

On propose maintenant une situation biologique simplifiée donnant lieu à une équation du type précédent. On considère une artère rectiligne de longueur infinie, de section circulaire variable. L'aire $u(t,x)$ d'une section de l'artère dépend du temps t et de sa position x sur la droite réelle. On suppose que u est une fonction de classe $\mathcal{C}^1(\mathbf{R}^+ \times \mathbf{R})$. On suppose que la vitesse du sang selon l'axe de l'artère, notée $c \in \mathbf{R}^+$, est une constante donnée du modèle.

On se donne $x_0 \in \mathbf{R}$ un réel quelconque, et $h > 0$. Considérons une tranche de fluide située entre x_0 et $x_0 + h$ à l'instant $t = 0$. On note $x_{x_0}(t)$ *(resp.* $x_{x_0+h}(t)$*)* l'abscisse au temps $t \geq 0$ de la section de fluide qui se trouvait à l'abscisse x_0 *(resp.* $x_0 + h$*)* à l'instant 0. Ces abscisses sont donc respectivement solution des problèmes de Cauchy

$$\begin{cases} \dfrac{dx_{x_0}(t)}{dt} = c, & t \in \mathbf{R}^{+*}, \\ x_{x_0}(0) = x_0. \end{cases} \qquad \text{et} \qquad \begin{cases} \dfrac{dx_{x_0+h}(t)}{dt} = c, & t \in \mathbf{R}^{+*}. \\ x_{x_0+h}(0) = x_0 + h. \end{cases} \tag{2}$$

4. Faire un croquis de la situation. On mettra en évidence :

— La tranche de fluide initiale,

— Celle après un temps $t > 0$,

— Les différentes abscisses mises en jeu dans le problème.

5. Que signifie biologiquement l'hypothèse « $c \in \mathbf{R}^+$ » dans ce cadre ?

6. Résoudre les équations (2) qui déterminent $x_{x_0}(t)$ et $x_{x_0+h}(t)$.

7. Pour $t \in \mathbf{R}^+$, on définit la quantité

$$V(t) = \int_{x_0+ct}^{x_0+h+ct} u(t,s)\,ds.$$

7.1 Que représente biologiquement cette quantité ?

7.2 Montrer que

$$\forall t \in \mathbf{R}^+, \qquad V(t) = \int_{x_0}^{x_0+h} u(t, s+ct)\, ds.$$

7.3 Exprimer $V'(t)$ sous forme d'une intégrale.

8. On suppose que pour tout $t \in \mathbf{R}^+$, on a $V'(t) = 0$.

8.1 Comment cette hypothèse s'interprète-t-elle concernant le fluide ?

8.2 Montrer que

$$\forall t \in \mathbf{R}^+, \qquad \int_{x_0}^{x_0+h} \left[\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x}\right](t, s+ct)\, ds = 0.$$

9. Soit $t \in \mathbf{R}^+$ fixé.

9.1 On introduit la fonction G définie sur $\mathbf{R}$ par

$$\forall x \in \mathbf{R}, \qquad G(x) = \int_{0}^{x} \left[\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x}\right](t, s+ct)\, ds.$$

Montrer que

$$\forall x_0 \in \mathbf{R}, \qquad \forall h > 0, \qquad G(x_0 + h) = G(x_0).$$

9.2 En déduire que pour tout $x_0 \in \mathbf{R}$, on a $G'(x_0) = 0$.

10. En déduire que :

$$\forall (t, x) \in \mathbf{R}^+ \times \mathbf{R}, \qquad \frac{\partial u}{\partial t}(t, x) + c\frac{\partial u}{\partial x}(t, x) = 0.$$

Partie 2 : Simulation numérique de l'équation de transport

Dans cette partie, on souhaite étudier un algorithme qui fournit une solution approchée de l'équation suivante correspondant à l'équation de transport avec $c = 1$:

$$\begin{cases} \dfrac{\partial u}{\partial t}(t, x) + \dfrac{\partial u}{\partial x}(t, x) = 0 & \text{pour } (t, x) \in \mathbf{R}^+ \times \mathbf{R}, \\ u(0, x) = u_0(x) & \text{pour } x \in \mathbf{R}, \end{cases} \tag{3}$$

où t représente le temps, x est la position dans l'espace et $u_0 \colon \mathbf{R} \to \mathbf{R}$, appelée « condition initiale », est la fonction définie par :

$$\forall x \in \mathbf{R}, \quad u_0(x) = \mathbb{1}_{]-\infty, 1]}(x),$$

où $x \mapsto \mathbb{1}_I(x)$ désigne la fonction indicatrice de l'intervalle I.

Cadre et notations On restreindra la résolution à des intervalles bornés en temps et en espace. Ainsi, on fixe $T > 0$ et $L > 1$ afin de chercher une solution approchée de (3) pour $t \in [0, T]$ et $x \in [0, L]$.

Ensuite, on découpe chacun des deux intervalles $[0, T]$ et $[0, L]$ en plusieurs sous-intervalles, de la façon suivante. Soit $N_t \in \mathbf{N}^*$ et $k = \frac{T}{N_t}$; on considère alors les points $t_0, t_1, \ldots, t_{N_t} \in [0, T]$ vérifiant $t_0 < t_1 < \cdots < t_{N_t}$ et définis par :

$$\forall i \in [\![0, N_t]\!], \quad t_i = ik.$$

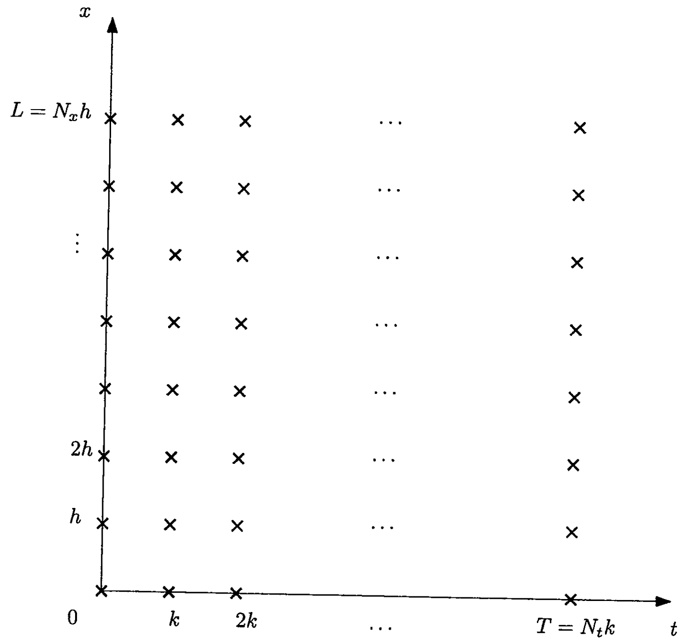

FIGURE 1 – Grille espace-temps de discrétisation utilisée pour la résolution numérique de l'équation de transport.

De même, soit $N_x \in \mathbf{N}^*$ et $h = \frac{L}{N_x}$; on considère alors les points $x_0, x_1, \dots, x_{N_x} \in [0, L]$ vérifiant $x_0 < x_1 < \cdots < x_{N_x}$ et définis par :

$$\forall j \in [\![0, N_x]\!], \quad x_j = jh.$$

Le réel k (respectivement h) est appelé le pas de discrétisation en temps (respectivement en espace). Les points $t_0, t_1, \dots, t_{N_t}$ (resp. $x_0, x_1, \dots, x_{N_x}$) forment ce qui est appelé une subdivision de $[0, T]$ (resp. de $[0, L]$). Ces notations sont résumées en Figure 1.

Objectif L'objectif de cette partie est donc, pour $i \in [\![0, N_t]\!]$ et $j \in [\![0, N_x]\!]$, de calculer une valeur approchée de u en les points (t_i, x_j), que l'on note $u_{i,j}$:

$$u(t_i, x_j) \approx u_{i,j}.$$

Consignes Chaque algorithme doit être précédé d'une phrase expliquant le raisonnement suivi pour l'écrire.

1. À partir de la question **1.2**, le module `NumPy` est utilisé pour coder les matrices et les vecteurs. Il est importé avec l'instruction `import numpy as np`, de sorte que chaque commande est à préfixer par `np.` ; en annexe A page 9 figurent quelques rappels sur ce module.

   1.1 Écrire une fonction `Uzero` qui, étant donné un réel x, renvoie la valeur de $u_0(x)$.

   1.2 Écrire une fonction `MaxiLigne` qui, étant donné une matrice M (de taille $n \times p$, avec $n, p \in \mathbf{N}^*$) et un entier $i \in [\![0, n-1]\!]$, renvoie la valeur du maximum, en valeur absolue, de la ligne n° i de M, c'est-à-dire :

$$\max_{j \in [\![0, p-1]\!]} |M[i, j]|.$$

2. Pour calculer les valeurs approchées $u_{i,j}$, on utilise les formules suivantes :

$$\forall j \in [\![0, N_x]\!], \quad u_{0,j} = u_0(x_j), \tag{4}$$

$$\forall i \in [\![0, N_t]\!], \quad u_{i,0} = 1, \tag{5}$$

$$\forall i \in [\![0, N_t - 1]\!], \forall j \in [\![1, N_x]\!], \quad u_{i+1,j} = u_{i,j} - \frac{k}{h}\left(u_{i,j} - u_{i,j-1}\right). \tag{6}$$

2.1 Pour calculer et conserver les $u_{i,j}$, on les stocke dans une matrice rectangulaire U de taille $(N_t + 1) \times (N_x + 1)$ définie par

$$U[i, j] = u_{i,j}.$$

On considère la fonction Amont suivante, qui prend en entrée les variables `T`, `L`, `Nt` et `Nx` correspondant respectivement à T, L, N_t et N_x, et qui renvoie une telle matrice U.

```
1  def Amont(T, L, Nt, Nx):
2      k = T/Nt; h = L/Nx
3      U = np.zeros((Nt+1,Nx+1))
4      for j in range(0, Nx+1):
5          U[0,j] = Uzero(j*h)
6      for i in range(0, Nt+1):
7          U[i,0] = 1
8      for i in range(_____):
9          for j in range(_____):
10             U[i+1,j] = U[i,j]-k/h*(U[i,j]-U[i,j-1])
11     return(U)
```

2.1.1 Quel est le rôle des lignes 4 à 7 ?

2.1.2 Lignes 8 et 9 : compléter, en *justifiant votre réponse*, le contenu des `range` afin que la fonction renvoie bien la matrice U voulue.

Pour la suite, on dispose d'une fonction `Norme` qui, étant donné une matrice M de taille $n \times p$, renvoie un vecteur C de taille n tel que

$$\text{pour tout } i \in [\![0, n-1]\!], \quad C[i] = \max_{j \in [\![0,p-1]\!]} |M[i, j]|.$$

Un tel vecteur contient donc les maximums, en valeur absolue, de chaque ligne de M. On cherche maintenant à savoir si l'algorithme proposé calcule des valeurs qui restent bornées.

2.2 Soit la fonction `Test` suivante qui prend en entrée un vecteur V.

```
def Test(V):
    d = np.shape(V); n = d[0]
    b = True
    for i in range(0, n):
        if V[i] > V[0]:
            b = False
    return(b)
```

Soit la fonction `Stable` suivante qui prend en entrée une matrice M.

```
def Stable(M):
    C = Norme(M)
    return(Test(C))
```

2.2.1 Pour chacune des affirmations suivantes, indiquer si elle est vraie ou fausse. *(Chaque réponse devra être justifiée.)*

(a) `Test` appliquée à un vecteur C de taille n permet de savoir si pour tout $i \in [\![0, n-1]\!]$, $C[i] \leq C[0]$.

(b) La valeur renvoyée par `Test` est de type chaîne de caractères (`str`).

(c) Dans `Test`, si `range(0,n)` est remplacé par `range(1,n)`, alors cela ne change pas le résultat renvoyé.

2.2.2 Étant donné une matrice U renvoyée par la fonction `Amont`, préciser ce que renvoie l'appel `Stable(U)`. *(La réponse sera exprimée en fonction des $u_{i,j}$.)*

3. On rappelle que, pour l'équation (3) dont on souhaite calculer une solution approchée dans cette partie 2, une solution exacte a déjà été trouvée en partie 1 : il s'agit de la fonction

$$u\colon (t,x) \mapsto u_0(x-t), \qquad (t,x) \in \mathbf{R}^+ \times \mathbf{R}.$$

Les graphiques à utiliser pour cette question figurent en annexe B, page 9.

3.1 Évolution de cette solution exacte au cours du temps. Les trois courbes $A1$, $A2$ et $A3$ (présentées en figure 3, annexe) correspondent au tracé de $x \mapsto u(t,x)$ pour trois temps différents fixés t_0, t_1 et t_2. Ces trois temps vérifient : $t_0 = 0$ et $t_0 < t_1 < t_2$. Identifier, en justifiant votre réponse, la courbe correspondant à t_0 ; de même pour t_1 et t_2.

3.2 Évolution, en fonction de la discrétisation en espace, de la solution approchée au temps final. Les deux courbes (présentées en figure 4, annexe) correspondent au tracé de la solution approchée au temps final $t = T$, pour deux valeurs de N_x différentes : $N_x^{(1)}$ et $N_x^{(2)}$, avec $N_x^{(1)} < N_x^{(2)}$. Identifier, en justifiant votre réponse, la courbe correspondant à $N_x^{(1)}$ et $N_x^{(2)}$.

A Annexe : rappel Python pour la partie 2

On suppose que le module NumPy est importé via `import numpy as np`.

A.1 Commandes

Interprétation	Python
Matrice nulle de taille $n \times p$	`np.zeros([n, p])`
Matrice identité de taille n	`np.eye(n)`
Copie de la matrice A dans une nouvelle matrice B	`B = np.copy(A)`
Dimensions de la matrice A	`d = np.shape(A)`
– nombre de lignes	`d[0]`
– nombre de colonnes	`d[1]`
Addition, soustraction et multiplication matricielles (pour A et B de tailles compatibles)	`A+B, A-B, np.dot(A, B)`
Coefficient d'indice (i, j) de la matrice A	`A[i,j]`

Pour une matrice A à n lignes et p colonnes, les indices vont de 0 à $n-1$ pour les lignes et de 0 à $p-1$ pour les colonnes.

A.2 Exemples

Pour définir la matrice $M = \begin{pmatrix} 2 & 4 & 6 \\ 1 & 3 & 7 \end{pmatrix}$, le vecteur $V = \begin{pmatrix} 2 & 4 & 6 \end{pmatrix}$ et connaître leur taille respective :

```
M = np.array([[2, 4, 6], [1, 3, 7]])
d = np.shape(M)
V = np.array([2, 4, 6])
n = np.shape(V)
```

Alors :

— `d[0]` renvoie 2, `d[1]` renvoie 3 et `M[0,2]` renvoie 6 ;

— `n[0]` renvoie 3 et `V[1]` renvoie 4.

B Annexe : tracés pour la partie 2

Cette annexe contient les graphiques à utiliser en question **3.** de la partie 2.

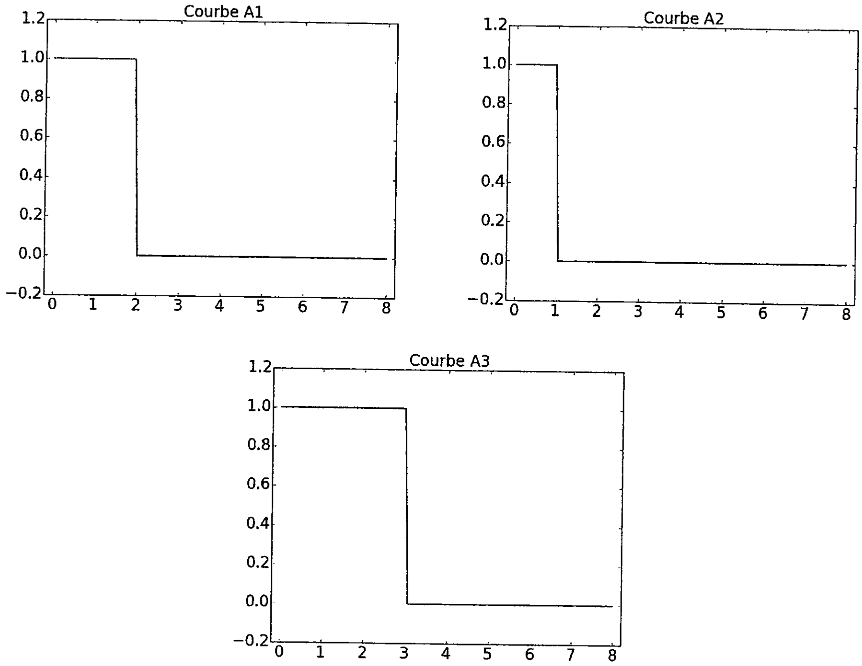

FIGURE 3 – Tracé des courbes $A1$, $A2$ puis $A3$. En abscisse figurent les valeurs de $x \in [0, L]$ avec $L = 8$.

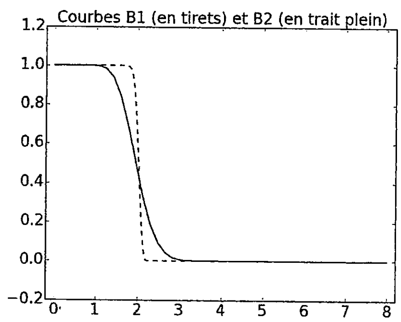

FIGURE 4 – Tracé des courbes $B1$ et $B2$. En abscisse figurent les valeurs de $x \in [0, L]$ avec $L = 8$.