Travaux pratiques - Contrôle final du 17 janvier 2019 (3h00)

Sans documents, sans calculatrice, sans portable, ...
Le barème est donné à titre indicatif

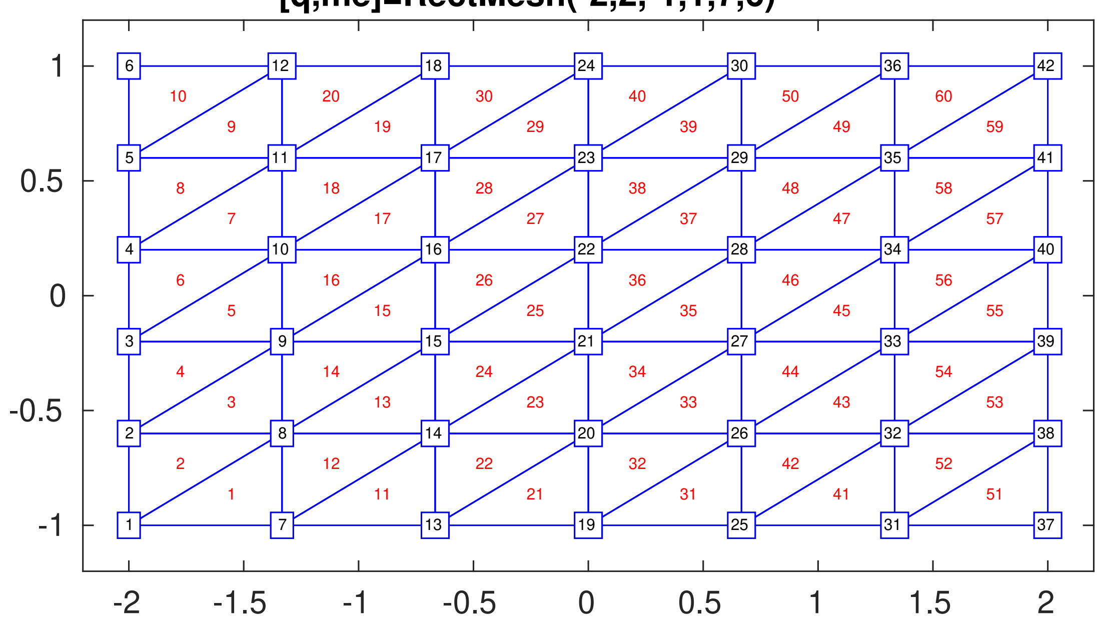
Exercice 1 (2.5 points)

Pour stocker les informations (minimales) relatives à un maillage, on utilise les tableaux $\boldsymbol{q}$ et $\boldsymbol{me}$ respectivements tableau des sommets/points et tableau de connectivité :

nom	type	dimension	descriptif
n_q	entier	1	nombre total de noeuds (sommets) du maillage
n_{me}	entier	1	nombre de triangles
q	réels	$2 \times n_q$	q(il , i) est la il-ème coordonnée du i-ème sommet, il $\in \{1, 2\}$ et i $\in \{1, \dots, n_q\}$. Le i-ème sommet sera aussi noté $\boldsymbol{q}^i = (\boldsymbol{q}^i_x, \boldsymbol{q}^i_y)$ avec $\boldsymbol{q}^i_x =$ q(1, i) et $\boldsymbol{q}^i_y =$ q(2, i)
me	entier	$3 \times n_{me}$	me(jl,k) indice de stockage, dans le tableau q , du jl-ème sommet du triangle d'indice k, jl $\in \{1, 2, 3\}$ et k $\in \{1, \dots, n_{me}\}$. Pour tout triangle la numérotation des points est dans le **sens direct**. q (:, me(1,k)) est le 1er sommet du k-ème triangle, q (:, me(2,k)) est le 2ème sommet, ...

Q. 1 *Ecrire une fonction* [q,me]=RectMesh(a,b,c,d,Nx,Ny) *retournant les tableaux de points et de connectivité associé au rectangle* $[a, b] \times [c, d]$ *avec* Nx *points suivant* x *et* Ny *points suivant* y *comme décrit sur la figure ci-dessous :*

[q,me]=RectMesh(-2,2,-1,1,7,6)



Attention, *les numérotations sont celles imposées par la figure et elles diffèrent de celles utilisées en TP. Les nombres encadrés sont les indices/numéros des sommets. Les nombres dans les triangles sont les indices/numéros des triangles.*

Exercice 2 (5.5 points)

On rappelle tout d'abord quelques définitions.

Definition 1 (problème de Cauchy) *Soit $\boldsymbol{f}$ l'application continue définie par*

$$\begin{array}{rccc} \boldsymbol{f} & : & [t^0, t^0+T] \times \mathbb{R}^d & \longrightarrow & \mathbb{R}^d \\ & & (t, \boldsymbol{y}) & \longmapsto & \boldsymbol{f}(t, \boldsymbol{y}) \end{array}$$

*avec $T \in]0, +\infty]$. Un **problème de Cauchy** revient à chercher une fonction $\boldsymbol{y} : [t^0, t^0+T] \longrightarrow \mathbb{R}^d$ continue et dérivable, telle que*

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad \forall t \in [t^0, t^0+T] \tag{1}$$

$$\boldsymbol{y}(t^0) = \boldsymbol{y}^{[0]} \in \mathbb{R}^d. \tag{2}$$

Différents schémas sont rappelés permettant de résoudre numériquement un problème de Cauchy. On note t^n, $n \in [\![0, N]\!]$, une discrétisation régulière de $[t^0, t^0+T]$, $\boldsymbol{y}^{[n]} \approx \boldsymbol{y}(t^n)$ et $\boldsymbol{f}^{[n]} = \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]})$

- Le **schéma de Runge-Kutta d'ordre 3**

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n + h, \boldsymbol{y}^{[n]} - h\boldsymbol{k}_1^{[n]} + 2h\boldsymbol{k}_2^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{6}(\boldsymbol{k}_1^{[n]} + 4\boldsymbol{k}_2^{[n]} + \boldsymbol{k}_3^{[n]}). \end{array} \tag{3}$$

- Trois **schémas d'Adams-Bashforth** :

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}\left(3\boldsymbol{f}^{[n]} - \boldsymbol{f}^{[n-1]}\right). \tag{4}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{12}\left(23\boldsymbol{f}^{[n]} - 16\boldsymbol{f}^{[n-1]} + 5\boldsymbol{f}^{[n-2]}\right). \tag{5}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{24}\left(55\boldsymbol{f}^{[n]} - 59\boldsymbol{f}^{[n-1]} + 37\boldsymbol{f}^{[n-2]} - 9\boldsymbol{f}^{[n-3]}\right). \tag{6}$$

  Ces schémas sont **explicites** et leur ordre correspond au nombre de pas.

- Trois **schémas d'Adams-Moulton** :

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}\left(\boldsymbol{f}^{[n+1]} + \boldsymbol{f}^{[n]}\right). \tag{7}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{12}\left(5\boldsymbol{f}^{[n+1]} + 8\boldsymbol{f}^{[n]} - \boldsymbol{f}^{[n-1]}\right). \tag{8}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{24}\left(9\boldsymbol{f}^{[n+1]} + 19\boldsymbol{f}^{[n]} - 5\boldsymbol{f}^{[n-1]} + \boldsymbol{f}^{[n-2]}\right). \tag{9}$$

  Ces schémas sont **implicites** et leur ordre correspond au nombre de pas plus un.

Q. 1 (Matlab) *Ecrire la fonction **Matlab*** [t,Y]=redRK3(f,a,b,yo,N) *permettant de résoudre numériquement le problème de Cauchy* (1)-(2) *par le schéma de Runge et Kutta d'ordre 3. Les paramètres d'entrée sont les données du problème de Cauchy avec* f $= \boldsymbol{f}$, a $= t^0$, b $= t^0 + T$, y0 $= \boldsymbol{y}^{[0]} \in \mathbb{R}^d$ *et* N *étant le nombre de pas de discrétisation.*

Application : On souhaite résoudre numériquement le problème de l'oscillateur libre de *Van der Pool* (1924)

$$\frac{d^2x}{dt^2}(t) - \epsilon\omega_0(1 - x^2(t))\frac{dx}{dt}(t) + \omega_0^2 x(t) = 0, \quad \forall t \in [0, T] \tag{10}$$

avec les conditions initiales

$$x(0) = x_0 \quad \text{et} \quad \frac{dx}{dt}(0) = v_0 \tag{11}$$

où x_0 et v_0 sont deux réels donnés. $x(t)$ représente l'amplitude des oscillations. ϵ, ω_0 et T sont des constantes positives. On pourra prendre, par exemple, $\epsilon = 0.1$, $\omega_0 = 1$ et $T = 10$.

On va utiliser la méthode de Runge-Kutta d'ordre 3 pour l'étude numérique des différents types de mouvements possibles, suivant les conditions de l'expérience (choix des valeurs de ϵ et de ω_0) et les conditions initiales imposées. On souhaite ensuite tracer les deux courbes discrètes $x(t)$ et $x'(t)$.

Q. 2 1. *Ecrire de manière détaillée le problème de Cauchy associé à* (10)-(11).

2. *Quelles sont les données du problème de Cauchy obtenu ? (avec leur type détaillé : entier, réel, complexe, vecteur, matrice, fonction, ...)*
3. *Quelles sont les inconnues du problème de Cauchy obtenu ? (avec leur type détaillé : entier, réel, complexe, vecteur, matrice, fonction, ...)*

Q. 3 (Matlab) *Ecrire un programme* ***Matlab*** *permettant de résoudre numériquement* (10)-(11) *par la méthode de Runge-Kutta d'ordre 3, et de représenter les deux courbes discrètes correspondant à* $x(t)$ *et* $x'(t)$.

Un **schéma prédicteur-correcteur** procède à chaque itération en deux temps : il fournit une valeur approchée de la solution au $n^{ième}$ pas en utilisant un schéma explicite (prédiction), puis il calcule la valeur correspondante de $\boldsymbol{f}$ et enfin, il substitue cette valeur dans un schéma implicite (correction).

$\boldsymbol{y}^{[0]}$ donné
pour n variant de 0 à $N-1$ faire
  Calculer $\bar{\boldsymbol{y}}$ une valeur approchée de $\boldsymbol{y}(t^{n+1})$ par un schéma explicite à partir de $\boldsymbol{y}^{[n]}$, $\boldsymbol{y}^{[n-1]}$, ...
  Evaluer $\bar{\boldsymbol{f}} = \boldsymbol{f}(t^{n+1}, \bar{\boldsymbol{y}})$
  Calculer $\boldsymbol{y}^{[n+1]}$ à l'aide d'un schéma implicite en remplaçant $\boldsymbol{f}^{[n+1]}$ par $\bar{\boldsymbol{f}}$
finpour

On suppose écrite la fonction `[t,Y]=redRK3(f,a,b,yo,N)` correspondant au schéma de Runge et Kutta d'ordre 3. Les paramètres d'entrée `f`, `a`, `b`, `yo` correspondent respectivement aux $\boldsymbol{f}$, t^0, t^0+T, $\boldsymbol{y}^{[0]}$ du problème de Cauchy (1)-(2). Enfin, `Y` est le tableau contenant les $\boldsymbol{y}^{[n]}$, $n \in \{0, \cdots, \mathrm{N}\}$ et `t` est le tableau contenant les `N+1` réels t^n, $n \in \{0, \cdots, \mathrm{N}\}$

Q. 4 (Matlab) 1. *Ecrire la fonction* `[t,Y]=redPC3(f,a,b,yo,N)` *correspondant à un schéma* ***prédicteur-correcteur*** *utilisant les schémas d'Adams-Bashforth et d'Adams-Moulton d'****ordre*** 3. *Ses paramètres d'entrée et de sortie sont identiques à ceux de la fonction* `redRK3` .

2. *Ecrire un programme permettant de représenter graphiquement l'ordre du schéma de prédiction-correction précédent.*

EXERCICE 3 (6.0 POINTS)

On souhaite résoudre numériquement l'E.D.P. suivante

$$\frac{\partial u}{\partial t}(t,x) - \kappa(x)\frac{\partial^2 u}{\partial x^2}(t,x) = f(t,x), \qquad \forall (t,x) \in]t_0; t_0+T]\times]a;b[, \tag{1}$$
$$u(t_0,x) = u_0(x), \qquad \forall x \in [a;b], \tag{2}$$
$$u(t,a) = u_a(t), \qquad \forall t \in [t_0; t_0+T], \tag{3}$$
$$\frac{\partial u}{\partial x}(t,b) + u(t,b) = v_b(t), \qquad \forall t \in]t_0; t_0+T]. \tag{4}$$

avec $\kappa : [a,b] \longrightarrow \mathbb{R}^{+*}$, $t_0 \in \mathbb{R}$, $T > 0$, $(a,b) \in \mathbb{R}^2$, $a < b$.

On note t^n, $n \in [\![0, N_t]\!]$ et x_i, $i \in [\![0, N_x]\!]$ les discrétisations régulières des intervalles $[t_0; t_0+T]$ et $[a;b]$ avec N_t pas de discrétisation en temps et N_x pas de discrétisation en espace. On souhaite résoudre l'E.D.P. à l'aide des schémas numériques

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \kappa_i \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} = f_i^{n+1}. \tag{5}$$
$$u_{N_x-2}^{n+1} - 4u_{N_x-1}^{n+1} + (3+2\Delta x)u_{N_x}^{n+1} = 2\Delta x v_b(t^{n+1}). \tag{6}$$

Q. 1 1. *Expliquer précisément comment le schéma* (5) *a été obtenu à partir de* (1) *et expliciter les valeurs* u_i^{n+1}, f_i^{n+1}, κ_i, Δt *et* Δx.

2. *Expliquer précisément comment le schéma* (6) *a été obtenu à partir de* (4).
3. *Donner une discrétisation (détaillée) du problème* (1) *à* (4) *en utilisant les schémas* (5) *et* (6).

On note $\boldsymbol{U}^n$ les vecteurs de dimension N_x+1, de composantes $\boldsymbol{U}_i^n = u_{i-1}^n$, $\forall i \in [\![1, N_x+1]\!]$.

Q. 2 1. *Comment initialiser le vecteur* $\boldsymbol{U}^0$ *?*

2. *En supposant le vecteur* $\boldsymbol{U}^n$ *déjà calculé, montrer que le vecteur* $\boldsymbol{U}^{n+1}$ *est solution du système linéaire*

$$\mathbb{A}\boldsymbol{U}^{n+1} = \boldsymbol{b}^n \tag{7}$$

en explicitant la matrice $\mathbb{A}$ *et le vecteur* $\boldsymbol{b}^n$ *(préciser les dimensions).*

Q. 3 (Matlab) *1. Ecrire la fonction* AssembleMat1D *retournant la matrice **creuse*** $\mathbb{M} \in \mathcal{M}_d(\mathbb{R})$ *définie par*

$$\mathbb{M} = \begin{pmatrix} \nu_1 & \nu_2 & \nu_3 & 0 & \dots & \dots & 0 \\ \beta_1 & \alpha_1 & \beta_1 & 0 & \dots & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \beta_{d-2} & \alpha_{d-2} & \beta_{d-2} \\ 0 & \dots & \dots & 0 & \mu_1 & \mu_2 & \mu_3 \end{pmatrix} \tag{8}$$

où $\alpha \in \mathbb{R}^{d-2}$, $\beta \in \mathbb{R}^{d-2}$ $\mu \in \mathbb{R}^3$ *et* $\nu \in \mathbb{R}^3$ *sont donnés. Cette fonction ne devra pas utiliser de boucles* for *ou* while.

2. On se donne $a = 0$, $b = 3$, $T = 10$, $f(t,x) = x^3 \sin(t)$, $u_0(x) = 10$,

$$v_b(t) = \begin{cases} 10 + 20t, & \text{si } t <= 5, \\ 110, & \text{si } t > 5. \end{cases} \quad \text{et} \quad \kappa(x) = \begin{cases} 1, & \text{si } x \in [a, (2a+b)/3] \cup [(a+2b)/3, b], \\ 5, & \text{si }](2a+b)/3, (a+2b)/3[. \end{cases}$$

Ecrire un programme Matlab complet permettant de résoudre le problème (1) *à* (4) *en utilisant les schémas* (5) *et* (6).

Exercice 4 (1.5 points)

Nous allons écrire une fonction permettant d'effectuer ce produit. Mais auparavant, quelques petits rappels (ou non) des possibilités offertes par le langage Matlab.

Il est facile sous Malab/Octave de modifier une matrice (creuse ou non). Par exemple,

- A(I,J)=M

  Les tableaux d'indices I et J sont de dimension respective n et m. La matrice M est de dimension n-par-m. La matrice A est modifiée de telle sorte que

  A(I(i),J(j))=M(i,j), $\forall$i $\in [\![1, n]\!]$, $\forall$j $\in [\![1, m]\!]$

- A(I,J)=A(I,J)+M

  Même chose en sommant.
- A(i,:)=a

  Ici a est un scalaire. Tous les éléments de la ligne i de la matrice A valent a .
- A(:,j)=a

  Ici a est un scalaire. Tous les éléments de la colonne j de la matrice A valent a .
- ...

Sous Matlab/Octave, il est possible de récupérer tous les éléments non nuls d'une matrice creuse A ainsi que leurs indices de ligne et de colonne à l'aide de la commande

[I,J,K]=find(A);

Ici les trois tableaux I, J et K ont même longueur et on a A(I(k),J(k)) == K(k) pour tout k inférieur ou égal à length(K).

Q. 1 (Matlab) *Ecrire la fonction* spMatKron *permettant à partir deux matrices creuses carrées* $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$ *et* $\mathbb{B} \in \mathcal{M}_m(\mathbb{R})$ *de retourner la matrice creuse* $\mathbb{C} = \mathbb{A} \otimes \mathbb{B}$. *Pour celà, on initialisera la matrice retournée* C *à l'aide de la commande* C=sparse(n*m,n*m) *puis on utilisera uniquement une boucle sur les éléments non-nuls de la matrice* $\mathbb{A}$ *pour «remplir» la matrice* C .

Exercice 5 (5.5 points)

Soient $\Omega =]a,b[\times]c,d[\subset \mathbb{R}^2$ et $\Gamma = \partial\Omega$ la frontière du domaine Ω. On note Γ_N, Γ_S, Γ_O et Γ_E respectivement les frontières nord, sud, ouest et est. on a

$$\Gamma = \Gamma_N \cup \Gamma_S \cup \Gamma_O \cup \Gamma_E.$$

On note $(x_i)_{i=0}^{N_x}$ et $(y_j)_{j=0}^{N_y}$ les discrétisation régulières, respectivement, des intervalles $[a,b]$ et $[c,d]$ défines par

$$x_i = a + ih_x,\ \forall i \in [\![0, N_x]\!] \quad \text{et} \quad y_j = c + jh_y,\ \forall j \in [\![0, N_y]\!] \tag{1}$$

avec $h_x = (b-a)/N_x$ et $h_y = (d-c)/N_y$. On note aussi

$$n_x = N_x + 1, \quad n_y = N_y + 1 \ \text{ et } \ N = n_x \times n_y \tag{2}$$

Soient $f : \Omega \longrightarrow \mathbb{R}$ et $g : \Gamma \longrightarrow \mathbb{R}$ deux fonctions données. On veut résoudre le problème suivant

$$-\Delta u = f, \qquad \text{dans } \Omega \tag{3}$$

$$u = g, \qquad \text{sur } \Gamma \tag{4}$$

en utilisant la discrétisation d'ordre 2 suivante :

$$-\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h_x^2} - \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h_y^2} = f(x_i, y_j), \qquad \forall (i,j) \in]\!]0, N_x[\![\times]\!]0, N_y[\![, \tag{5}$$

$$U_{0,j} = g(a, y_j), \qquad \forall j \in]\!]0, N_y[\![, \tag{6}$$

$$U_{N_x,j} = g(b, y_j), \qquad \forall j \in]\!]0, N_y[\![, \tag{7}$$

$$U_{i,0} = g(x_i, c), \qquad \forall i \in [\![0, N_x]\!], \tag{8}$$

$$U_{i,N_y} = g(x_i, d), \qquad \forall i \in [\![0, N_x]\!]. \tag{9}$$

avec $U_{i,j} \approx u(x_i, y_j)$.

Pour tout $j \in [\![0, N_y]\!]$, on note $U_{:,j}$ le vecteur de $\mathbb{R}^{n_x}$ définit par

$$U_{:,j} = \begin{pmatrix} U_{0,j} \\ \vdots \\ U_{N_x,j} \end{pmatrix}.$$

On note $\boldsymbol{V} \in \mathbb{R}^N$ le vecteur bloc

$$\boldsymbol{V} = \left(\begin{array}{c} U_{:,0} \\ \hline U_{:,1} \\ \hline \vdots \\ \hline U_{:,N_y} \end{array}\right)$$

Q. 1 *Explicitez la bijection* $\mathcal{F} : [\![0, N_x]\!] \times [\![0, N_y]\!] \longrightarrow [\![1, N]\!]$ *telle que*

$$\forall (i,j) \in [\![0, N_x]\!] \times [\![0, N_y]\!], \quad V_k = U_{i,j}, \quad \textit{avec } k = \mathcal{F}(i,j).$$

Dans le cas de la numérotation en $(i,j) \in [\![0, N_x]\!] \times [\![0, N_y]\!]$ on parlera de **numérotation 2D** et pour la numérotation en $k \in [\![1, N]\!]$ on parlera de **numérotation globale**.

Q. 2 (Matlab) *Ecrire la fonction* `k=bijF(i,j,nx)` *correspondant à la bijection* $\mathcal{F}$ *(* ***numerotation 2D*** *vers* ***numerotation globale****).*

Chacune des équations du problème discret (5)-(9) correspond à une discrétisation en un point (x_i, y_j). Nous choisissons d'écrire ces équations en utilisant la même numérotation que lors de la construction du vecteur $\boldsymbol{V}$: l'équation *écrite* au point (x_i, y_j) sera écrite en ligne $k = \mathcal{F}(i,j)$ du système.

Q. 3 *Etablir que le problème discret* (5)-(9) *peut s'écrire sous la forme du système linéaire bloc*

$$\left(\begin{array}{c|cccccc|c} \mathbb{E} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \\ \hline \mathbb{M} & \mathbb{D} & \mathbb{M} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{M} & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \mathbb{O} & \ddots & \ddots & \ddots & \mathbb{O} & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \mathbb{M} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{M} & \mathbb{D} & \mathbb{M} \\ \hline \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{E} \end{array}\right) \boldsymbol{V} = \left(\begin{array}{c} B_{:,0} \\ \hline B_{:,1} \\ \hline \vdots \\ \vdots \\ \hline B_{:,N_y} \end{array}\right) \tag{10}$$

où chaque bloc de la matrice est une matrice $(N_x + 1)$ par $(N_x + 1)$. La matrice $\mathbb{O} \in \mathcal{M}_{N_x+1}(\mathbb{R})$ est la matrice nulle. Les matrices creuses $\mathbb{D}$, $\mathbb{M}$ et $\mathbb{E}$ ainsi que les vecteurs $B_{:,j} \in \mathbb{R}^{N_x+1}$, pour tout $j \in [\![0, N_y]\!]$, devront être donnés explicitement.

On note $\mathbb{I}_n$ la matrice identité de $\mathcal{M}_n(\mathbb{R})$ et $\mathbb{J}_n$ la matrice de $\mathcal{M}_n(\mathbb{R})$ définie par

$$\mathbb{J}_n = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 0 \end{pmatrix} \in \mathcal{M}_n(\mathbb{R}).$$

Nous allons maintenant générer/assembler la matrice du système (10) **sans tenir compte des conditions aux limites** : on note $-\mathbb{A}_{xy}$ la matrice ainsi obtenue.

C'est une matrice bloc de $\mathcal{M}_N(\mathbb{R})$ avec n_y lignes bloc composées de blocs carrés de dimension n_x qui s'écrit sous la forme :

$$\mathbb{A}_{xy} = \left(\begin{array}{c|cccccc|c} \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \\ \hline \mathbb{O} & \mathbb{A}_x & \mathbb{O} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \mathbb{A}_x & \ddots & \ddots & \vdots & \vdots \\ \vdots & \mathbb{O} & \ddots & \ddots & \ddots & \mathbb{O} & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbb{A}_x & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{O} & \mathbb{A}_x & \mathbb{O} \\ \hline \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \end{array}\right) + \left(\begin{array}{c|cccccc|c} \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \\ \hline \mathbb{S}_y & \mathbb{T}_y & \mathbb{S}_y & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{S}_y & \mathbb{T}_y & \ddots & \ddots & \vdots & \vdots \\ \vdots & \mathbb{O} & \ddots & \ddots & \ddots & \mathbb{O} & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbb{T}_y & \mathbb{S}_y & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{S}_y & \mathbb{T}_y & \mathbb{S}_y \\ \hline \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \end{array}\right) \quad (11)$$

où

$$\mathbb{S}_y = \frac{1}{h_y^2}\mathbb{J}_{n_x}, \quad \mathbb{T}_y = -\frac{2}{h_y^2}\mathbb{J}_{n_x} \quad \text{et} \quad \mathbb{A}_x = \frac{1}{h_x^2}\mathbb{L}$$

avec

$$\mathbb{L} = \begin{pmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & -2 & 1 \\ 0 & \cdots & \cdots & 0 & 0 & 0 \end{pmatrix} \in \mathcal{M}_{n_x}(\mathbb{R})$$

On peut noter que les matrices $\mathbb{A}_x$, $\mathbb{T}_y$ et $\mathbb{S}_y$ sont des matrices de $\mathcal{M}_{n_x}(\mathbb{R})$.

En notant $\mathbb{A}_y \in \mathcal{M}_{n_y}(\mathbb{R})$ la matrice définie par $\mathbb{A}_y = \frac{1}{h_y^2}\mathbb{L}$, on déduit de (11)

$$\mathbb{A}_{xy} = \mathbb{I}_{n_y} \otimes \mathbb{A}_x + \mathbb{A}_y \otimes \mathbb{I}_{n_x}. \quad (12)$$

Q. 4 (Matlab) *1. Sans utiliser de boucles, écrire la fonction* Lap1DAssembling *retournant la matrice creuse* $\mathbb{L}$.

2. Ecrire la fonction Lap2DAssembling *retournant la matrice bloc creuse* $\mathbb{A}_{xy}$ *en utilisant* (12).

3. Proposer un programme permettant de tester/valider la matrice ainsi obtenue en utilisant le laplacien d'une fonction de $\mathbb{R}^2$ *à valeurs dans* $\mathbb{R}$.