

TRAVAUX PRATIQUES - EXAMEN DU 6 JANVIER 2020 (3H00)

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

EXERCICE 1 (5 POINTS)

On cherche à résoudre par un schéma de type différences finies le problème suivant

$$-u''(x) + c(x)u(x) = f(x), \forall x \in]a; b[\quad (1)$$

$$u'(a) - 2u(a) = w_a \in \mathbb{R} \quad (2)$$

$$u'(b) + 2u(b) = w_b \in \mathbb{R} \quad (3)$$

où $c : [a, b] \rightarrow \mathbb{R}^+$, $f : [a, b] \rightarrow \mathbb{R}$, w_a et w_b sont donnés.

Q. 1 Ecrire, en justifiant, un schéma d'ordre 2 associé au problème (1)-(2)-(3).

Q. 2 (Matlab) Ecrire un programme Matlab permettant de :

- résoudre le problème précédent avec des données judicieusement choisies (pour avoir une solution exacte),
- représenter graphiquement la solution exacte et la solution approchée.

Q. 3 (Matlab) Ecrire un programme Matlab permettant de retrouver graphiquement l'ordre de la méthode.

EXERCICE 2 (9 POINTS)

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

$$\frac{\partial u}{\partial t}(t, x) - \kappa \frac{\partial^2 u}{\partial x^2}(t, x) + c(x) \frac{\partial u}{\partial x}(t, x) = f(t, x), \quad \forall (t, x) \in]t_0; t_0 + T] \times]a; b[, \quad (1)$$

$$u(t_0, x) = u_0(x), \quad \forall x \in [a; b], \quad (2)$$

$$\frac{\partial u}{\partial x}(t, a) = v_a(t), \quad \forall t \in [t_0; t_0 + T], \quad (3)$$

$$u(t, b) = u_b(t), \quad \forall t \in [t_0; t_0 + T]. \quad (4)$$

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

On note t^n , $n \in \llbracket 0, N_t \rrbracket$ et x_i , $i \in \llbracket 0, N_x \rrbracket$ 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

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \kappa \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} + c_i \frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} = f_i^{n+1}. \quad (5)$$

$$u_2^{n+1} - 4u_1^{n+1} + 3u_0^{n+1} = -2\Delta x v_a(t^{n+1}). \quad (6)$$

Q. 1 (a) Expliquer précisément comment le schéma (5) (ordre 1 en temps et ordre 2 en espace) a été obtenu à partir de (1) et expliciter les valeurs u_i^{n+1} , f_i^{n+1} , c_i , Δt et Δx .

(b) Expliquer précisément comment le schéma (6) (ordre 2) a été obtenu à partir de (3).

(c) Donner une discrétisation (détaillée) du problème (1) à (4) en utilisant (entre autres) les schémas (5) et (6).

On note \mathbf{U}^n les vecteurs de dimension $N_x + 1$, de composantes $U_i^n = u_{i-1}^n$, $\forall i \in \llbracket 1, N_x + 1 \rrbracket$.

Q. 2 (a) Comment initialiser le vecteur \mathbf{U}^0 ?

(b) En supposant le vecteur \mathbf{U}^n déjà calculé, montrer que le vecteur \mathbf{U}^{n+1} est solution du système linéaire

$$\mathbb{A} \mathbf{U}^{n+1} = \mathbf{b}^n \quad (7)$$

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

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

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

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.

Q. 4 (Matlab) On se donne $a = -2$, $b = 2$, $T = 10$, $f(t, x) = x^2 \cos(t)$, $u_0(x) = 10$, $c(x) = 2 + \sin(x)$, $\kappa = 1$, $v_a(t) = \sin(t)$,

$$u_b(t) = \begin{cases} 10 + 25t, & \text{si } t \leq 2, \\ 60, & \text{si } t \in [2; 8[, \\ 60 - 25(t - 8), & \text{si } t \in [8; 10]. \end{cases}$$

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

Pour améliorer les performances du programme, nous allons réécrire la fonction **ASSEMBLEMAT1D** sans utiliser de boucles et en utilisant la commande **A=sparse(i,j,s,m,n)**.

Q. 5 (Matlab) (a) Expliquer l'usage de la commande **A=sparse(i,j,s,m,n)**.

(b) Ecrire la fonction **ASSEMBLEMAT1DVEC** retournant la matrice **creuse** $\mathbb{M} \in \mathcal{M}_d(\mathbb{R})$ définie par (8). Cette fonction ne devra pas utiliser de boucles **for** ou **while**.

EXERCICE 3 (7 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

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

La normale unitaire extérieure à Γ est notée \mathbf{n} . On note $(x_i)_{i=0}^{N_x}$ et $(y_j)_{j=0}^{N_y}$ les discrétisations régulières, respectivement, des intervalles $[a, b]$ et $[c, d]$ définies par

$$x_i = a + ih_x, \quad \forall i \in \llbracket 0, N_x \rrbracket \quad \text{et} \quad y_j = c + jh_y, \quad \forall j \in \llbracket 0, N_y \rrbracket \quad (1)$$

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

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

Soient $f : \Omega \rightarrow \mathbb{R}$, $g : \Gamma \rightarrow \mathbb{R}$ et $\kappa \in \mathbb{R}^+$ donnés. On veut résoudre le problème suivant

$$-\Delta u + \kappa u = f, \quad \text{dans } \Omega \quad (3)$$

$$u = g, \quad \text{sur } \Gamma \quad (4)$$

en utilisant la discrétisation d'ordre 2 suivante :

$$-\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h_x^2} - \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h_y^2} + \kappa U_{i,j} = f(x_i, y_j), \quad \forall (i, j) \in \llbracket 0, N_x \rrbracket \times \llbracket 0, N_y \rrbracket, \quad (5)$$

$$U_{0,j} = g(a, y_j), \quad \forall j \in \llbracket 0, N_y \rrbracket, \quad (6)$$

$$U_{N_x,j} = g(b, y_j), \quad \forall j \in \llbracket 0, N_y \rrbracket, \quad (7)$$

$$U_{i,0} = g(x_i, c), \quad \forall i \in \llbracket 0, N_x \rrbracket, \quad (8)$$

$$U_{i,N_y} = g(x_i, d), \quad \forall i \in \llbracket 0, N_x \rrbracket. \quad (9)$$

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

Pour tout $j \in \llbracket 0, N_y \rrbracket$, on note $U_{:,j}$ le vecteur de \mathbb{R}^{n_x} défini par

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

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

$$\mathbf{V} = \begin{pmatrix} \frac{U_{:,0}}{U_{:,1}} \\ \vdots \\ \frac{U_{:,N_y}}{U_{:,N_y}} \end{pmatrix}$$

Q. 1 Expliquez la bijection $\mathcal{F} : \llbracket 0, N_x \rrbracket \times \llbracket 0, N_y \rrbracket \longrightarrow \llbracket 1, N \rrbracket$ telle que

$$\forall (i, j) \in \llbracket 0, N_x \rrbracket \times \llbracket 0, N_y \rrbracket, \quad V_k = U_{i,j}, \quad \text{avec } k = \mathcal{F}(i, j).$$

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

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

$$\begin{pmatrix} \mathbb{E} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \\ \mathbb{M} & \mathbb{D} & \mathbb{M} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{M} & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \mathbb{O} & \ddots & \ddots & \ddots & \mathbb{O} & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \mathbb{M} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{M} & \mathbb{D} & \mathbb{M} \\ \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{E} \end{pmatrix} \mathbf{V} = \begin{pmatrix} \frac{B_{:,0}}{B_{:,1}} \\ \vdots \\ \frac{B_{:,N_y}}{B_{:,N_y}} \end{pmatrix} \quad (10)$$

où chaque bloc de la matrice est une matrice de $\mathcal{M}_{n_x}(\mathbb{R})$. La matrice $\mathbb{O} \in \mathcal{M}_{n_x}(\mathbb{R})$ est la matrice nulle. Les matrices creuses \mathbb{D} , \mathbb{M} et \mathbb{E} ainsi que les vecteurs $B_{:,j} \in \mathbb{R}^{n_x}$, pour tout $j \in \llbracket 0, N_y \rrbracket$, 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 :

$$\mathbb{A}_{xy} = \begin{pmatrix} \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{A}_x & \mathbb{O} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \mathbb{A}_x & \ddots & \ddots & \vdots & \vdots \\ \vdots & \mathbb{O} & \ddots & \ddots & \ddots & \mathbb{O} & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbb{A}_x & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{O} & \mathbb{A}_x & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \end{pmatrix} \begin{pmatrix} \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \\ \mathbb{S}_y & \mathbb{T}_y & \mathbb{S}_y & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{S}_y & \mathbb{T}_y & \ddots & \ddots & \vdots & \vdots \\ \vdots & \mathbb{O} & \ddots & \ddots & \ddots & \mathbb{O} & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbb{T}_y & \mathbb{S}_y & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{S}_y & \mathbb{T}_y & \mathbb{S}_y \\ \mathbb{O} & \mathbb{O} & \cdots & \cdots & \cdots & \mathbb{O} & \mathbb{O} \end{pmatrix} \quad (11)$$

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 $\mathcal{M}_{n_x}(\mathbb{R})$.

Et maintenant quelques petits rappels sur le **produit tensoriel de Kronecker** :

Soient $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$ et $\mathbb{B} \in \mathcal{M}_{p,q}(\mathbb{R})$. Le produit tensoriel de Kronecker de \mathbb{A} par \mathbb{B} , noté $\mathbb{A} \otimes \mathbb{B}$, est la matrice de $\mathcal{M}_{mp,nq}(\mathbb{R})$ définie avec des blocs de dimension $p \times q$ par

$$\mathbb{A} \otimes \mathbb{B} = \begin{pmatrix} A_{1,1}\mathbb{B} & \cdots & A_{1,n}\mathbb{B} \\ \vdots & \ddots & \vdots \\ A_{m,1}\mathbb{B} & \cdots & A_{m,n}\mathbb{B} \end{pmatrix} \quad (12)$$

Le produit de Kronecker est bilinéaire et associatif : Si les dimensions des matrices \mathbb{A} , \mathbb{B} et \mathbb{C} sont compatibles on a $\forall \lambda \in \mathbb{K}$

$$\begin{aligned} \mathbb{A} \otimes (\mathbb{B} + \lambda \cdot \mathbb{C}) &= (\mathbb{A} \otimes \mathbb{B}) + \lambda(\mathbb{A} \otimes \mathbb{C}) \\ (\mathbb{A} + \lambda \cdot \mathbb{B}) \otimes \mathbb{C} &= (\mathbb{A} \otimes \mathbb{C}) + \lambda(\mathbb{B} \otimes \mathbb{C}) \\ \mathbb{A} \otimes (\mathbb{B} \otimes \mathbb{C}) &= (\mathbb{A} \otimes \mathbb{B}) \otimes \mathbb{C} \end{aligned}$$

Par contre, il n'est pas commutatif.

Soit $\mathbb{K} = \mathbb{A} \otimes \mathbb{B}$. Cette matrice peut être calculée avec la fonction **kron** de Matlab/Octave :

$$\mathbb{K} = \mathbf{kron}(\mathbb{A}, \mathbb{B});$$

Si les matrices \mathbb{A} et \mathbb{B} sont creuses (**sparse**) alors \mathbb{K} l'est aussi.

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)

$$\mathbb{A}_{xy} = \mathbb{I}_{n_y} \otimes \mathbb{A}_x + \mathbb{A}_y \otimes \mathbb{I}_{n_x}. \quad (13)$$

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

(b) Écrire la fonction `Lap2DAssembling` retournant la matrice bloc creuse \mathbb{A}_{xy} en utilisant (13).

(c) 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} .

EXERCICE 4 (2 POINTS)

On dispose, comme en TP, des fonctions Matlab **Quadrillage** et **black** :

- **Quadrillage**(imin,imax,jmin,jmax) permet de générer un quadrillage pour les lignes imin à imax et les colonnes jmin à jmax.
- **black**(i,j) permet de représenter un carré noir en ligne i, colonne j d'un quadrillage.

Q. 1 *Ecrire une fonction Matlab `mosaïque26` de paramètre $n \in \mathbb{N}$, $n \geq 3$, permettant de créer des figures sur le quadrillage de lignes $-n$ à n et de colonnes $-n$ à n . Voici deux exemples, avec $n = 13$ et $n = 14$, des figures que l'on souhaite représenter :*

