Travaux pratiques - Examen du 8 mars 2021 (3h00)

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

Exercice 1 (6 points)

Cet exercice a pour objectif la résolution d'une E.D.O.

Q. 1 *Rappeler précisement la définition d'un problème de Cauchy (vectoriel).* ☐

Un schéma de Runge-Kutta d'ordre 3 pour la résolution d'un problème de Cauchy vectoriel est donné par

$$\left\{\begin{array}{lcl} \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{4}\big(3\boldsymbol{k}_2(t^n, \boldsymbol{y}^{[n]}, h) + \boldsymbol{k}_3(t^n, \boldsymbol{y}^{[n]}, h)\big) \\ \text{avec} & & \boldsymbol{k}_1(t, \boldsymbol{y}, h) = \boldsymbol{f}(t, \boldsymbol{y}), \\ & & \boldsymbol{k}_2(t, \boldsymbol{y}, h) = \boldsymbol{f}\big(t + \frac{2h}{3}, \boldsymbol{y} + \frac{2h}{3}\boldsymbol{k}_1(t, \boldsymbol{y}, h)\big), \\ & & \boldsymbol{k}_3(t, \boldsymbol{y}, h) = \boldsymbol{f}\big(t, \boldsymbol{y} - h\boldsymbol{k}_1(t, \boldsymbol{y}, h) + h\boldsymbol{k}_2(t, \boldsymbol{y}, h)\big), \\ \boldsymbol{y}^{[0]} & & \text{donné.} \end{array}\right. \tag{1}$$

Q. 2 (Matlab) **(a)**. *Ecrire la fonction Matlab* RedRK3 *permettant de résoudre un problème de Cauchy vectoriel par le schéma* (1).
(b). *Ecrire un programme Matlab permettant de vérifier numériquement l'ordre de cette méthode.* ☐

La **méthode de Nyström** explicite, à **pas multiples**, et d'ordre 3 est donnée par

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n-1]} + \frac{h}{3}\left(7\boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) - 2\boldsymbol{f}(t^{n-1}, \boldsymbol{y}^{[n-1]}) + \boldsymbol{f}(t^{n-2}, \boldsymbol{y}^{[n-2]})\right) \tag{2}$$

La **méthode de Gear d'ordre 3** implicite est donné par

$$\boldsymbol{y}^{[n+1]} = \frac{18}{11}\boldsymbol{y}^{[n]} - \frac{9}{11}\boldsymbol{y}^{[n-1]} + \frac{2}{11}\boldsymbol{y}^{[n-2]} + \frac{2h}{11}\boldsymbol{f}(t^{n+1}, \boldsymbol{y}^{[n+1]}) \tag{3}$$

Q. 3 (Matlab) *Ecrire la fonction Matlab* PC3 *permettant de résoudre un problème de Cauchy (vectoriel) par une méthode de prédiction-correction utilisant les schémas* (2) *et* (3)*. Cette fonction devra être* ***optimisée*** *en nombre d'appels de la fonction* $\boldsymbol{f}$ *et en place mémoire.* ☐

Application : De manière classique, un problème à N corps se modélise par

$$\frac{d^2\boldsymbol{q}_i}{dt^2}(t) = -G\sum_{j=1,\ j\neq i}^{N} m_j \frac{\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)}{\|\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)\|^3},\ \forall i \in [\![1, N]\!]$$

où $\boldsymbol{q}_i(t) \in \mathbb{R}^3$ est la position du i-ème corps à l'instant t. Les constantes strictement positives m_1, ..., m_N et G sont des constantes strictement positives supposées données.

On se propose de résoudre numériquement un problème à $N = 3$ corps avec les positions et vitesses de ces 3 corps supposées connues à l'instant initial $t = 0$.

Q. 4 *Ecrire le problème à* $N = 3$ *corps sous la forme d'un problème de Cauchy.* ☐

Q. 5 (Matlab) *Ecrire un Matlab complet permettant de résoudre ce problème à 3 corps (les données sont libres mais devront être spécifiées). Ce programme devra aussi représenter sur une figure les approximations des positions des 3 corps au cours du temps.* ☐

Exercice 2 (5 points)

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

$$\begin{aligned}
-u''(x) + \alpha u'(x) + c(x)u(x) &= f(x), \ \forall x \in]a;b[\quad (1)\\
u'(a) - u(a) &= w_a \in \mathbb{R} \quad (2)\\
u(b) &= w_b \in \mathbb{R} \quad (3)
\end{aligned}$$

où $c : [a,b] \longrightarrow \mathbb{R}^+$, $f : [a,b] \longrightarrow \mathbb{R}$, $\alpha > 0$, 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 3 (9 points)

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

$$\begin{aligned}
\frac{\partial u}{\partial t}(t,x) - \beta\frac{\partial^2 u}{\partial x^2}(t,x) + \nu\frac{\partial u}{\partial x}(t,x) + \alpha u(t,x) &= f(t,x), && \forall (t,x) \in]t_0; t_0+T] \times]a;b[, \quad (1)\\
u(t_0,x) &= u_0(x), && \forall x \in [a;b], \quad (2)\\
\frac{\partial u}{\partial x}(t,b) + 2u(t,b) &= v_b(t), && \forall t \in [t_0; t_0+T], \quad (3)\\
u(t,a) &= u_a(t), && \forall t \in]t_0; t_0+T]. \quad (4)
\end{aligned}$$

avec $\alpha > 0$, $\nu > 0$, $\beta > 0$, $t_0 \in \mathbb{R}$, $T > 0$, $(a,b) \in \mathbb{R}^2$, $a < b$.
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

$$\begin{aligned}
\frac{u_i^{n+1} - u_i^n}{h_t} - \beta\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h_x^2} + \nu\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2h_x} + \alpha u_i^{n+1} &= f_i^{n+1}. \quad (5)\\
u_{N_x-2}^{n+1} - 4u_{N_x-1}^{n+1} + (3+4h_x)u_{N_x}^{n+1} &= 2h_x v_b(t^{n+1}). \quad (6)
\end{aligned}$$

Q. 1 **(a)**. *Expliquer précisement comment le schéma (5) (ordre 1 en temps et ordre 2 en espace) a été obtenu à partir de (1) et expliquer à quoi correspondent les valeurs u_i^{n+1}, f_i^{n+1}, h_t et h_x.*

(b). *Expliquer précisement 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 $\boldsymbol{U}^n$ 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 **(a)**. *Comment initialiser le vecteur $\boldsymbol{U}^0$?*

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

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

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

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

$$\mathbb{M} = \begin{pmatrix}
a_1 & c_1 & \nu & 0 & \dots & \dots & 0\\
b_1 & a_2 & c_2 & 0 & \dots & \dots & 0\\
0 & \ddots & \ddots & \ddots & \ddots & & \vdots\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots\\
\vdots & & \ddots & \ddots & \ddots & \ddots & 0\\
0 & \dots & \dots & 0 & b_{d-2} & a_{d-1} & c_{d-1}\\
0 & \dots & \dots & 0 & \mu & b_{d-1} & a_d
\end{pmatrix} \quad (8)$$

où $\boldsymbol{a} = (a_i)_{i=1}^d \in \mathbb{R}^d$, $\boldsymbol{b} = (b_i)_{i=1}^{d-1} \in \mathbb{R}^{d-1}$, $\boldsymbol{c} = (c_i)_{i=1}^{d-1} \in \mathbb{R}^{d-1}$ $\mu \in \mathbb{R}$ et $\nu \in \mathbb{R}$ sont donnés. □

Q. 4 (Matlab) *On se donne* $a = -1$, $b = 1$, $T = 10$, $f(t,x) = x^2\cos(t)$, $u_0(x) = 20$, $\alpha = 1$, $\beta = 2$, $\nu = 1/2$, $v_b(t) = 0$,

$$u_a(t) = \begin{cases} 20 + 50t, & si\ t <= 2, \\ 120, & si\ t \in [2;9[, \\ 120 - 50(t-9), & si\ t \in [9;10]. \end{cases}$$

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**. □