Méthodes Numériques II
Chapitre 3: Equations Différentielles Ordinaires
Exercices - épisode 2

version du 2025/02/13 à 05:06:34

Exercice 1

la **méthode de Heun** est donnée par

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

Q. 1 *Ecrire la fonction algorithmique* REDHeunVec *permettant de résoudre un problème de Cauchy vectoriel par la méthode de Heun en utilisant au plus* $2N$ *évaluation de* $\boldsymbol{f}$. ▫

Q. 2 *Ecrire un programme algorithmique permettant de retrouver numériquement l'ordre de cette méthode.* ▫

Correction

R. 1 Le schéma de Heun peut s'écrire sous la forme

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}(\boldsymbol{k}_1^n + \boldsymbol{k}_2^n)$$

avec

$$\begin{aligned} \boldsymbol{k}_1^n &= \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^n &= \boldsymbol{f}(t^{n+1}, \boldsymbol{y}^{[n]} + h\boldsymbol{k}_1) \end{aligned}$$

L'algorithme de la fonction REDHeunVec s'écrit alors :

Algorithme 1: Fonction REDHeunVec : résolution d'un problème de Cauchy par le schéma de Heun

Données :
- $\boldsymbol{f}$: $\boldsymbol{f} : [t^0, t^0+T] \times \mathbb{R}^d \longrightarrow \mathbb{R}^d$ fonction d'un problème de Cauchy (scalaire)
- t^0 : réel, temps initial
- T : réel > 0
- $\boldsymbol{y}^0$: un vecteur de $\mathbb{R}^d$, donnée initiale
- N : un entier non nul (nombre de pas de discrétisation).

Résultat :
- $\boldsymbol{t}$: vecteur de $\mathbb{R}^{N+1}$, $\boldsymbol{t}(n) = t^{n-1}$, $\forall n \in [\![1, N+1]\!]$
- $\boldsymbol{Y}$: matrice réelle de dimension $d \times (N+1)$, $\boldsymbol{Y}(:,n) = \boldsymbol{y}^{(n-1)}$, $\forall n \in [\![1, N+1]\!]$

Fonction $[\boldsymbol{t}, \boldsymbol{Y}] \leftarrow$ REDHeunVec($f, t^0, T, \boldsymbol{y}^0, N$)
  $\boldsymbol{t} \leftarrow DirReg(t^0, t^0+T, N)$
  $h \leftarrow (b-a)/N$
  $\boldsymbol{Y}(:,1) \leftarrow \boldsymbol{y}^0$
  Pour $n \leftarrow 1$ **à** N **faire**
    $\boldsymbol{k}_1 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n), \boldsymbol{Y}(:,n))$
    $\boldsymbol{k}_2 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n+1), \boldsymbol{Y}(:,n) + h\boldsymbol{k}_1)$
    $\boldsymbol{Y}(:,n+1) \leftarrow \boldsymbol{Y}(:,n) + (h/2) * (\boldsymbol{k}_1 + \boldsymbol{k}_2)$
  Fin Pour
Fin Fonction

R. 2 Il est possible de vérifier/retrouver numériquement l'ordre du schéma de Heun. Pour celà on choisi un problème de Cauchy dont la solution exacte est connue et on calcule pour différentes valeurs de h (et donc différentes valeurs de N) le maximum de l'erreur commise entre la solution exacte et la solution numérique donnée par le schéma de Heun :

$$E(h) = \max_{n \in [\![0,N]\!]} \left\| \boldsymbol{y}(t^n) - \boldsymbol{y}^{[n]} \right\|_\infty$$

On représente ensuite la fonction $h \mapsto E(h)$. La méthode de Heun étant d'ordre 2, on a alors $E(h) = \mathcal{O}(h^2) \approx Ch^2$ quand h est suffisament petit. On utilise alors une échelle logarithmique pour représenter la courbe. En effet, on a

$$\log E(h) \approx \log(Ch^2) = \log(C) + 2\log(h)$$

En posant $X = \log(h)$ et $Y = \log E(h)$, coordonnées en échelle logarithmique, on a

$$Y \approx \log(C) + 2X$$

qui est l'équation d'une droite. La pente de cette droite est donc l'ordre de la méthode.

$t^0 \leftarrow 0,\ T \leftarrow 4\pi,$
$f : t, z \longrightarrow \cos(t) + 1,\ y_{ex} : t \longrightarrow \sin(t) + t$
$y^0 \leftarrow y_{ex}(t^0)$
$LN \leftarrow 100 : 50 : 1000$
$nLN \leftarrow \mathrm{length}(LN)$
$H \leftarrow \mathbb{0}_{nLN},$ ▷ pour stocker les h
$E \leftarrow \mathbb{0}_{nLN},$ ▷ pour stocker les erreurs
Pour $k \leftarrow 1$ **à** nLN **faire**
    $N \leftarrow LN(k)$
    $[t, y] \leftarrow \mathrm{REDHeunVec}(f, t^0, T, \boldsymbol{y}^0, N)$
    $E(k) \leftarrow \max(\mathrm{abs}(y - y_{ex}(t)))$
    $H(k) \leftarrow T/N$
Fin Pour
... ▷ Representation graphique

◇

Exercice 2

la **méthode de Runge-Kutta d'ordre 4** est donnée par

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_2^{[n]}) \\ \boldsymbol{k}_4^{[n]} & = & \boldsymbol{f}(t^n + h, \boldsymbol{y}^{[n]} + h\boldsymbol{k}_3^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{6}(\boldsymbol{k}_1^{[n]} + 2\boldsymbol{k}_2^{[n]} + 2\boldsymbol{k}_3^{[n]} + \boldsymbol{k}_4^{[n]}). \end{array}$$

Q. 1 *Ecrire une fonction algorithmique* REDRK4Vec *permettant de résoudre un problème de Cauchy vectoriel par la méthode de Runge-Kutta d'ordre 4.* □

Q. 2 *Ecrire un programme algorithmique permettant de retrouver numériquement l'ordre de cette méthode.* □

Correction

R. 1 Il faut noter qu'il n'est pas nécessaire de stocker l'ensemble des vecteurs $\boldsymbol{k}_1^{[n]}, \ldots, \boldsymbol{k}_4^{[n]}$, $n \in [\![1, N]\!]$. Pour minimiser l'occupation mémoire de l'ordinateur, à chaque itération, on calcule $\boldsymbol{k}_1 \leftarrow \boldsymbol{k}_1^{[n]}(t^n, \boldsymbol{y}^{[n]})$, ... $\boldsymbol{k}_4 \leftarrow \boldsymbol{k}_4^{[n]}(t^n, \boldsymbol{y}^{[n]})$. L'algorithme de la fonction REDRK4Vec s'écrit alors :

Algorithme 2: Fonction REDRK4Vec : résolution d'un problème de Cauchy par le schéma de RK4

Données :
- $\boldsymbol{f}$: $\boldsymbol{f} : [t^0, t^0 + T] \times \mathbb{R}^d \longrightarrow \mathbb{R}^d$ fonction d'un problème de Cauchy (scalaire)
- t^0 : réel, temps initial
- T : réel > 0
- $\boldsymbol{y}^0$: un vecteur de $\mathbb{R}^d$, donnée initiale
- N : un entier non nul (nombre de pas de discrétisation).

Résultat :
- $\boldsymbol{t}$: vecteur de $\mathbb{R}^{N+1}$, $\boldsymbol{t}(n) = t^{n-1}$, $\forall n \in [\![1, N+1]\!]$
- $\boldsymbol{Y}$: matrice réelle de dimension $d \times (N+1)$, $\boldsymbol{Y}(:, n) = \boldsymbol{y}^{(n-1)}$, $\forall n \in [\![1, N+1]\!]$

Fonction $[\boldsymbol{t}, \boldsymbol{Y}] \leftarrow$ REDRK4Vec($f, t^0, T, \boldsymbol{y}^0, N$)
    $\boldsymbol{t} \leftarrow DirReg(t^0, t^0 + T, N)$
    $h \leftarrow (b - a)/N$
    $\boldsymbol{Y}(:, 1) \leftarrow \boldsymbol{y}^0$
    Pour $n \leftarrow 1$ **à** N **faire**
        $\boldsymbol{k}_1 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n), \boldsymbol{Y}(:, n))$
        $\boldsymbol{k}_2 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n) + h/2, \boldsymbol{Y}(:, n) + (h/2)\boldsymbol{k}_1)$
        $\boldsymbol{k}_3 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n) + h/2, \boldsymbol{Y}(:, n) + (h/2)\boldsymbol{k}_2)$
        $\boldsymbol{k}_4 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n) + h, \boldsymbol{Y}(:, n) + h\boldsymbol{k}_3)$
        $\boldsymbol{Y}(:, n+1) \leftarrow \boldsymbol{Y}(:, n) + (h/6) * (\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)$
    Fin Pour
Fin Fonction

R. 2 *voir aussi correction Exercice ??-Q2 : l'ordre 2 étant remplacé par 4 ici!* Il est possible de vérifier/retrouver numériquement l'ordre du schéma de Runge-Kutta 4. Pour celà on choisi un problème de Cauchy dont la solution exacte est connue et on calcule pour différentes valeurs de h (et donc différentes valeurs de N) le maximum de l'erreur commise entre la solution exacte et la solution numérique donnée par le schéma de Runge-Kutta 4 :

$$E(h) = \max_{n \in [\![0, N]\!]} \left\| \boldsymbol{y}(t^n) - \boldsymbol{y}^{[n]} \right\|_\infty$$

On représente ensuite la fonction $h \mapsto E(h)$. La méthode de Runge-Kutta 4 étant d'ordre 4, on a alors théoriquement $E(h) = \mathcal{O}(h^4) \approx Ch^4$ quand h est suffisament petit. On utilise alors une échelle logarithmique pour représenter la courbe. En effet, on a

$$\log E(h) \approx \log(Ch^4) = \log(C) + 4\log(h)$$

En posant $X = \log(h)$ et $Y = \log E(h)$, coordonnées en échelle logarithmique, on a

$$Y \approx \log(C) + 4X$$

qui est l'équation d'une droite. La pente de cette droite est donc l'ordre de la méthode.

$t^0 \leftarrow 0$, $T \leftarrow 4\pi$,
$f : t, z \longrightarrow \cos(t) + 1$, $y_{ex} : t \longrightarrow \sin(t) + t$
$y^0 \leftarrow y_{ex}(t^0)$
$LN \leftarrow 100 : 50 : 1000$
$nLN \leftarrow \text{length}(LN)$
$H \leftarrow \mathbb{0}_{nLN}$, ▷ pour stocker les h
$E \leftarrow \mathbb{0}_{nLN}$, ▷ pour stocker les erreurs
Pour $k \leftarrow 1$ **à** nLN **faire**
  $N \leftarrow LN(k)$
  $[t, y] \leftarrow \text{REDRK4Vec}(f, t^0, T, \boldsymbol{y}^0, N)$
  $E(k) \leftarrow \max(\text{abs}(y - y_{ex}(t)))$
  $H(k) \leftarrow T/N$
Fin Pour
... ▷ Representation graphique

◇

Exercice 3

La **méthode de Adam-Bashforth d'ordre 4** explicite est donnée par

$$\boldsymbol{y}^{[n+1]} \quad = \quad \boldsymbol{y}^{[n]} + \tfrac{h}{24}\left(55\boldsymbol{f}^{[n]} - 59\boldsymbol{f}^{[n-1]} + 37\boldsymbol{f}^{[n-2]} - 9\boldsymbol{f}^{[n-3]}\right). \tag{3.1}$$

avec $\boldsymbol{f}^{[n]} = \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]})$.

Q. 1 *Ecrire la fonction algorithmique REDAB4Vec permettant de résoudre un problème de Cauchy vectoriel par cette méthode.* □

Correction

R. 1 Soit $(t^{[n]})_{n=0}^{N}$ la discrétisation régulière de $[t^0, t^0 + T]$ avec $h = T/N$. On a donc $t^{[n]} = t^0 + nh$, $\forall n \in [\![0, N]\!]$.
On ne peut *utiliser* le schéma à pas multiples (3.1) que pour $n \geqslant 3$. On va alors utiliser un schéma à un pas d'ordre (au moins) 4 pour calculer les 4 premiers termes $\boldsymbol{y}^{[0]}$, $\boldsymbol{y}^{[1]}$, $\boldsymbol{y}^{[2]}$ et $\boldsymbol{y}^{[3]}$ nécessaire pour *démarrer* le schéma (3.1). On choisi par exemple la méthode de Runge-Kutta d'ordre 4 pour initialiser ces 4 termes. Deux possibilités :

- on réécrit le schéma de Runge-Kutta d'ordre 4 pour ces 4 premiers termes

  $\boldsymbol{Y}(:, 1) \leftarrow \boldsymbol{y}^0$
  Pour $n \leftarrow 1$ **à** 3 **faire**
    $\boldsymbol{k}_1 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n), \boldsymbol{Y}(:, n))$
    $\boldsymbol{k}_2 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n) + h/2, \boldsymbol{Y}(:, n) + (h/2)\boldsymbol{k}_1)$
    $\boldsymbol{k}_3 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n) + h/2, \boldsymbol{Y}(:, n) + (h/2)\boldsymbol{k}_2)$
    $\boldsymbol{k}_4 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n) + h, \boldsymbol{Y}(:, n) + h\boldsymbol{k}_3)$
    $\boldsymbol{Y}(:, n+1) \leftarrow \boldsymbol{Y}(:, n) + (h/6) * (\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)$
  Fin Pour

- on utilise la fonction REDRK4Vec avec ses paramètres d'entrées judicieusement choisis :

  $[\boldsymbol{t}_{ini}, \boldsymbol{Y}_{ini}] \leftarrow \text{REDRK4Vec}(f, t^0, t^0 + 3 * h, \boldsymbol{y}^0, 3)$
  Pour $n \leftarrow 1$ **à** 4 **faire**
    $\boldsymbol{Y}(:, n) \leftarrow \boldsymbol{Y}_{ini}(:, n)$
  Fin Pour

En choisissant cette dernière solution, l'algorithme de la fonction REDAB4Vec s'écrit alors :

Algorithme 3: Fonction REDAB4Vec : résolution d'un problème de Cauchy par le schéma explicite d'Adams-Bashforth d'ordre 4

Données : $\boldsymbol{f}$: $\boldsymbol{f} : [t^0, t^0 + T] \times \mathbb{R}^d \longrightarrow \mathbb{R}^d$ fonction d'un problème de Cauchy (scalaire)
t^0 : réel, temps initial
T : réel > 0
$\boldsymbol{y}^0$: un vecteur de $\mathbb{R}^d$, donnée initiale
N : un entier non nul (nombre de pas de discrétisation).

Résultat : $\boldsymbol{t}$: vecteur de $\mathbb{R}^{N+1}$, $\boldsymbol{t}(n) = t^{n-1}$, $\forall n \in [\![1, N+1]\!]$
$\boldsymbol{Y}$: matrice réelle de dimension $d \times (N+1)$, $\boldsymbol{Y}(:, n) = \boldsymbol{y}^{(n-1)}$, $\forall n \in [\![1, N+1]\!]$

Fonction $[\boldsymbol{t}, \boldsymbol{Y}] \leftarrow$ REDAB4Vec($f, t^0, T, \boldsymbol{y}^0, N$)
  $\boldsymbol{t} \leftarrow$ DisReg$(t^0, t^0 + T, N)$
  $h \leftarrow (b - a)/N$
  $[\boldsymbol{t}_{ini}, \boldsymbol{Y}_{ini}] \leftarrow$ REDRK4Vec$(f, t^0, t^0 + 3 * h, \boldsymbol{y}^0, 3)$
  Pour $n \leftarrow 1$ **à** 4 **faire**
    $\boldsymbol{Y}(:, n) \leftarrow \boldsymbol{Y}_{ini}(:, n)$
  Fin Pour
  $\boldsymbol{k}_1 \leftarrow \boldsymbol{f}(\boldsymbol{t}(3), \boldsymbol{Y}(:, 3))$
  $\boldsymbol{k}_2 \leftarrow \boldsymbol{f}(\boldsymbol{t}(2), \boldsymbol{Y}(:, 2))$
  $\boldsymbol{k}_3 \leftarrow \boldsymbol{f}(\boldsymbol{t}(1), \boldsymbol{Y}(:, 1))$
  Pour $n \leftarrow 4$ **à** N **faire**
    $\boldsymbol{k}_0 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n), \boldsymbol{Y}(:, n))$
    $\boldsymbol{Y}(:, n+1) \leftarrow \boldsymbol{Y}(:, n) + (h/24) * (55 * \boldsymbol{k}_0 - 59 * \boldsymbol{k}_1 + 37 * \boldsymbol{k}_2 - 9 * \boldsymbol{k}_3)$
    $\boldsymbol{k}_3 \leftarrow \boldsymbol{k}_2$, $\boldsymbol{k}_2 \leftarrow \boldsymbol{k}_1$, $\boldsymbol{k}_1 \leftarrow \boldsymbol{k}_0$
  Fin Pour
Fin Fonction

◇

Exercice 4

On pose $\boldsymbol{f}^{[n]} = \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]})$. La **méthode de Adams-Bashforth d'ordre 4** explicite est donnée par

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

et la **méthode de Adams-Moulton d'ordre 4** implicite par

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

avec $\boldsymbol{f}^{[n]} = \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]})$.

Q. 1 *Ecrire la fonction algorithmique* REDPC4Vec *permettant de résoudre un problème de Cauchy (vectoriel) par une méthode de prédiction-correction utilisant ces deux schémas. On minimisera le nombre d'appel à la fonction* $\boldsymbol{f}$ *dans la boucle principale.* □

Correction

R. 1 On utilise le schéma de Runge-Kutta d'ordre 4 pour initialiser les 4 premières valeurs.
Ensuite on utilise comme prédicteur le schéma explicite et comme correcteur le schéma implicite. Le principe est donc

- Calcul à l'aide du prédicteur :

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

- Calcul à l'aide du correcteur :

$$\begin{aligned}
\hat{\boldsymbol{f}}^{[n+1]} &= \boldsymbol{f}(t^{n+1}, \hat{\boldsymbol{y}}^{[n+1]}) \\
\boldsymbol{y}^{[n+1]} &= \boldsymbol{y}^{[n]} + \tfrac{h}{24}\left(9\hat{\boldsymbol{f}}^{[n+1]} + 19\boldsymbol{f}^{[n]} - 5\boldsymbol{f}^{[n-1]} + \boldsymbol{f}^{[n-2]}\right)
\end{aligned}$$

L'algorithme de la fonction REDPC4Vec s'écrit alors :

Algorithme 4: Fonction REDPC4Vec : résolution d'un problème de Cauchy par prédiction-correction (Adams-Bashforth/Adams-Moulton) d'ordre 4

Données :
- $\boldsymbol{f}$: $\boldsymbol{f} : [t^0, t^0+T] \times \mathbb{R}^d \longrightarrow \mathbb{R}^d$ fonction d'un problème de Cauchy (scalaire)
- t^0 : réel, temps initial
- T : réel > 0
- $\boldsymbol{y}^0$: un vecteur de $\mathbb{R}^d$, donnée initiale
- N : un entier non nul (nombre de pas de discrétisation).

Résultat :
- $\boldsymbol{t}$: vecteur de $\mathbb{R}^{N+1}$, $\boldsymbol{t}(n) = t^{n-1}$, $\forall n \in [\![1, N+1]\!]$
- $\boldsymbol{Y}$: matrice réelle de dimension $d \times (N+1)$, $\boldsymbol{Y}(:,n) = \boldsymbol{y}^{(n-1)}$, $\forall n \in [\![1, N+1]\!]$

Fonction $[\boldsymbol{t}, \boldsymbol{Y}] \leftarrow$ REDPC4Vec($f, t^0, T, \boldsymbol{y}^0, N$)
  $\boldsymbol{t} \leftarrow DisReg(t^0, t^0+T, N)$
  $h \leftarrow (b-a)/N$
  $[\boldsymbol{t}_{ini}, \boldsymbol{Y}_{ini}] \leftarrow$ REDRK4Vec$(f, t^0, t^0 + 3*h, \boldsymbol{y}^0, 3)$
  Pour $n \leftarrow 1$ **à** 4 **faire**
    $\boldsymbol{Y}(:,n) \leftarrow \boldsymbol{Y}_{ini}(:,n)$
  Fin Pour
  $\boldsymbol{k}_1 \leftarrow \boldsymbol{f}(\boldsymbol{t}(3), \boldsymbol{Y}(:,3))$
  $\boldsymbol{k}_2 \leftarrow \boldsymbol{f}(\boldsymbol{t}(2), \boldsymbol{Y}(:,2))$
  $\boldsymbol{k}_3 \leftarrow \boldsymbol{f}(\boldsymbol{t}(1), \boldsymbol{Y}(:,1))$
  Pour $n \leftarrow 4$ **à** N **faire**
    $\boldsymbol{k}_0 \leftarrow \boldsymbol{f}(\boldsymbol{t}(n), \boldsymbol{Y}(:,n))$
    $\hat{\boldsymbol{Y}} \leftarrow \boldsymbol{Y}(:,n) + (h/24)*(55*\boldsymbol{k}_0 - 59*\boldsymbol{k}_1 + 37*\boldsymbol{k}_2 - 9*\boldsymbol{k}_3)$
    $\hat{\boldsymbol{F}} \leftarrow \boldsymbol{f}(\boldsymbol{t}(n+1), \hat{\boldsymbol{Y}})$
    $\boldsymbol{Y}(:,n+1) \leftarrow \boldsymbol{Y}(:,n) + (h/24)*(9*\hat{\boldsymbol{F}} + 19*\boldsymbol{k}_0 - 5*\boldsymbol{k}_1 + \boldsymbol{k}_2)$
    $\boldsymbol{k}_3 \leftarrow \boldsymbol{k}_2$
    $\boldsymbol{k}_2 \leftarrow \boldsymbol{k}_1$
    $\boldsymbol{k}_1 \leftarrow \boldsymbol{k}_0$
  Fin Pour
Fin Fonction

◇

Exercice 5 : Examen du 4 avril 2023, partie E.D.O.

Q. 1 a. *Que signifie l'abréviation E.D.O.?*

b. *Donner la définition détaillée d'un problème de Cauchy* ***vectoriel****.*

c. *Quelles sont les données d'un problème de Cauchy* ***vectoriel****?*

d. Que cherche-t'on?

□

Q. 2 *Ecrire une fonction algorithmique* `DisReg` *permettant de d'obtenir une discrétisation régulière de l'intervalle* $[a, b]$ *($a < b$) en $n + 1$ points.* □

On souhaite résoudre numériquement un problème de Cauchy par un schéma explicite à un pas (constant) du type

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + h\boldsymbol{\Phi}(t^n, \boldsymbol{y}^{[n]}, h). \tag{5.2}$$

La fonction $\boldsymbol{\Phi}$ associée à une méthode de Runge-Kutta à q évaluations de $\boldsymbol{f}$ (fonction associée au problème de Cauchy) peut s'écrire sous la forme :

$$\boldsymbol{\Phi}(t, \boldsymbol{y}, h) = \sum_{i=1}^{q} c_i \boldsymbol{k}^{[i]}(t, \boldsymbol{y}, h)$$

avec

$$\boldsymbol{k}^{[i]}(t, \boldsymbol{y}, h) = \boldsymbol{f}\left(t + ha_i, y + h\sum_{j=1}^{q} b_{i,j}\boldsymbol{k}^{[j]}(t, \boldsymbol{y}, h)\right), \ 1 \leqslant i \leqslant q$$

que l'on peut représenter sous la forme d'un tableau dit **tableau de Butcher** :

$$\begin{array}{c|c} \boldsymbol{a} & \mathbb{B} \\ \hline & \boldsymbol{c}^t \end{array} \tag{5.3}$$

avec $\mathbb{B} = (b_{i,j})_{i,j\in[\![1,q]\!]} \in \mathcal{M}_{q,q}(\mathbb{R})$, $\boldsymbol{a} = (a_i)_{i\in[\![1,q]\!]} \in \mathbb{R}^q$ et $\boldsymbol{c} = (c_i)_{i\in[\![1,q]\!]} \in \mathbb{R}^q$.
On prend pour tableau de Butcher:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{5} & \frac{1}{5} & 0 & 0 \\ 1 & -\frac{13}{7} & \frac{20}{7} & 0 \\ \hline & -\frac{1}{3} & \frac{25}{24} & \frac{7}{24} \end{array} \tag{5.4}$$

Q. 3 *Ecrire explicitement et en détail le schéma de Runge-Kutta associé au tableau de Butcher (5.4). On admettra que ce schéma est d'ordre* 3. □

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

$$\left\{\begin{array}{lcl} \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{1}{36}h(3\boldsymbol{k}_1 + 25\boldsymbol{k}_2 + 8\boldsymbol{k}_3) \\ \text{avec} & & \boldsymbol{k}_1 = \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}), \\ & & \boldsymbol{k}_2 = \boldsymbol{f}(t^n + \frac{2}{5}h, \boldsymbol{y}^{[n]} + \frac{2}{5}h\boldsymbol{k}_1), \\ & & \boldsymbol{k}_3 = \boldsymbol{f}(t^n + h, \boldsymbol{y}^{[n]} + h(-\frac{7}{8}\boldsymbol{k}_1 + \frac{15}{8}\boldsymbol{k}_2)), \\ \boldsymbol{y}^{[0]} & & \text{donné.} \end{array}\right. \tag{5.5}$$

Q. 4 *[Algorithmique] Ecrire la fonction algorithmique* RedRK3 *permettant de résoudre un problème de Cauchy vectoriel par le schéma* (5.5). □

Un schéma explicite, à **pas multiples**, et d'ordre 3 est donné par

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + h\left(\frac{23}{12}\boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) - \frac{4}{3}\boldsymbol{f}(t^{n-1}, \boldsymbol{y}^{[n-1]}) + \frac{5}{12}\boldsymbol{f}(t^{n-2}, \boldsymbol{y}^{[n-2]})\right) \tag{5.6}$$

Q. 5 *Expliquez en détail comment résoudre un problème de Cauchy vectoriel par le schéma* (5.6). *Un soin particulier sera apporté à l'«initialisation».* □

Q. 6 *[Algorithmique] Ecrire la fonction algorithmique* RedPM3 *permettant de résoudre un problème de Cauchy (vectoriel) par le schéma* (5.6).

□

Application: Soit le système d'E.D.O. suivant

$$\left\{\begin{array}{ll} \ddot{x}_1(t) - \nu_1(\dot{x}_2(t) - \dot{x}_1(t)) + x_2(t) & = \cos(t) \quad (5.7a) \\ \ddot{x}_2(t) - \nu_2(\dot{x}_1(t) - \dot{x}_2(t)) + x_1(t) & = \sin(t) \quad (5.7b) \end{array}\right.$$

On veut résoudre numériquement ce système d'E.D.O. avec pour données initiales $x_1(0) = 1$, $\dot{x}_1(0) = 0$, $x_2(0) = -1$, $\dot{x}_2(0) = 1/2$. Le temps final T sera égal à 10.

Q. 7 *Ecrire le problème précédent sous la forme d'un problème de Cauchy.* □

Q. 8 *[Algorithmique] Ecrire un algorithme complet permettant de résoudre* (5.7a)-(5.7b) *avec les données initiales spécifiées. On prendra* $\nu_1 = 1/4$ *et* $\nu_2 = 1/3$. *Ce programme devra aussi représenter les approximations des fonctions* x_1 *et* x_2. *On utilisera pour celà la fonction* Plot(X, Y) *qui relie les points* $(X(i), Y(i))$ *contenus dans les deux tableaux de même taille* X *et* Y *(function similaire à la fonction* `plot` *de Matlab).* □

Correction

R. 1 a. Equation Différentielle Ordinaire

b. Un problème de Cauchy vectoriel consiste à déterminer la fonction $\boldsymbol{y} : [t^0, t^0 + T] \longrightarrow \mathbb{R}^m$ solution de

$$\begin{cases} \boldsymbol{y}'(t) &= \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad \forall t \in [t^0, t^0 + T], \\ \boldsymbol{y}(t^0) &= \boldsymbol{y}_0 \in \mathbb{R}^m, \end{cases}$$

avec $\boldsymbol{f} : [t^0, t^0 + T] \times \mathbb{R}^m \longrightarrow \mathbb{R}^m$ donné.

c. les données d'un problème de Cauchy vectoriel sont

- $t^0 \in \mathbb{R}$,
- $T \in \mathbb{R}^{+*}$,
- $\boldsymbol{y}_0 \in \mathbb{R}^m$,
- $\boldsymbol{f} : [t^0, t^0 + T] \times \mathbb{R}^m \longrightarrow \mathbb{R}^m$.

d. On cherche la fonction $\boldsymbol{y} : [t^0, t^0 + T] \longrightarrow \mathbb{R}^m$

◇