MT09 – TP7 (équations différentielles)

On veut résoudre numériquement un système d'équations différentielles :

$$\left\{\begin{array}{l} y'(t) = f(t, y(t)) \\ y(t_0) = a \end{array}\right. , \; t \in [t_0, t_0 + T], \; y(t) \in \mathbb{R}^p, \; a \in \mathbb{R}^p, \tag{1}$$

où t_0 et T sont des réels donnés ($T > 0$), a est un vecteur de $\mathbb{R}^p$ donné, f est une fonction connue définie de $\mathbb{R} \times \mathbb{R}^p$ dans $\mathbb{R}^p$.

Dans la suite, étant donné l'entier $N > 0$, on pose $h = \dfrac{T}{N}$. On définit $t_n = t_0 + nh, \; n = 0, ..., N$.

On calcule une matrice Z dont les colonnes Z_n sont des vecteurs de $\mathbb{R}^p$ qui approchent la solution $y(t_n)$ au point t_n.

1. Écrire une fonction Scilab

   ```
   Z = pointmilieu(a, t0, T, N, f)
   ```

   qui, étant donné le vecteur a , les réels t_0 et T, ($T > 0$), l'entier non nul N et la fonction f, calcule la matrice Z solution approchée de (1) en utilisant le schéma du point milieu.

2. On appelle équation de Van Der Pol :

   $$\left\{\begin{array}{l} u''(t) = c(1 - u^2(t))u'(t) - u(t) \\ u(t_0) = \alpha, \; u'(t_0) = \beta \end{array}\right. , \; t \in [t_0, T] \tag{2}$$

   (a) Mettre cette équation (2) sous la forme d'un système de deux équations différentielles du premier ordre

   $$\left\{\begin{array}{l} x'(t) = vdp(t, x(t)) \\ x(t_0) = a \end{array}\right. , \quad t \in [t_0, T], x(t) \in \mathbb{R}^2, a \in \mathbb{R}^2 \tag{3}$$

   (b) Écrire la fonction Scilab

   ```
   Y = vdp(t, X)
   ```

   qui correspond à la résolution de (3). On fixera $c = 0.4$. Attention : dans ce cas particulier, `vdp` ne dépend pas de t.

   (c) Écrire une fonction Scilab

   ```
   tracevdp(a, t0, T, Nptmil)
   ```

   qui, étant donné le vecteur a de $\mathbb{R}^2$, étant donnés les réels t_0 et T, et l'entier `Nptmil` trace dans le plan (x_1, x_2) la courbe représentative de la solution de (3) calculée numériquement par le schéma du point milieu, mettre un titre à la figure.
   Application numérique : On choisit

   $$a = \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \; t_0 = 0, \; T = 15, \; N = 100. \tag{4}$$

   (d) Écrire une fonction Scilab

   ```
   Z = eulerexpl(a, t0, T, N, f)
   ```

   qui calcule la matrice Z solution approchée de (1) en utilisant le schéma d'Euler explicite.
   Modifier votre fonction `tracevdp` qui deviendra `tracevdp(a, t0, T, Neul, Nptmil)`, afin de tracer sur une même figure les courbes représentatives des solutions approchées de (3) obtenues respectivement par les fonctions `eulerexpl` et `pointmilieu`.

3. (a) Il existe une fonction Scilab

   ```
   X = ode(a, t0, theta, f)
   ```

   qui, étant donné a de taille $(n, 1)$, le réel t_0, le vecteur $\theta = (\theta_1, \ldots, \theta_k)$, la fonction f de $\mathbb{R}^{n+1}$ dans $\mathbb{R}^n$, calcule numériquement une solution approchée de (1), la matrice X calculée contient les approximations de $(x(\theta_1), x(\theta_2), \ldots, x(\theta_k))$, cette méthode utilise un pas adaptatif (elle est beaucoup plus sophistiquée que la méthode de point milieu que vous venez d'écrire). L'adaptation du pas amène à calculer d'autres valeurs que $(x(\theta_1), x(\theta_2), \dots, x(\theta_k))$, mais seules ces dernières sont restituées par la fonction.

Ajouter la résolution par `ode` à votre fonction `tracevdp`. Chaque méthode aura son propre nombre de points de discrétisation.

Application numérique : vous pouvez reprendre les valeurs données par (4). Vous pouvez ensuite choisir d'autres valeurs de a, d'autres valeurs de T, d'autres valeurs de N.

4. On veut résoudre (1) par la méthode de Runge Kutta d'ordre 4.

   (a) Écrire une fonction Scilab

   ```
   Z = RK4(a, t0, T, N, f)
   ```

   qui, étant donné le vecteur a , les réels t_0 et T, l'entier N, la fonction f, calcule Z solution approchée de (1) en utilisant le schéma classique de Runge Kutta d'ordre 4.

$$Z_{n+1} = Z_n + \frac{h}{6}\left(K_0 + 2K_1 + 2K_2 + K_3\right), \text{ avec } \begin{cases} K_0 = f(t_n, Z_n) \\ K_1 = f\left(t_n + \frac{h}{2}, Z_n + \frac{h}{2}K_0\right) \\ K_2 = f\left(t_n + \frac{h}{2}, Z_n + \frac{h}{2}K_1\right) \\ K_3 = f(t_n + h, Z_n + hK_2) \end{cases}$$

   (b) Modifier votre fonction en

   ```
   tracevdp(a, t0, T, Neul, Nptmil, Nrk4, Node)
   ```

   qui trace sur une même figure les courbes représentatives des solutions approchées de (3) obtenues par les méthodes ci-dessus.

   (c) Pour obtenir la matrice X, combien d'évaluations de $f(t, X)$ doit-on effectuer ? Quel était ce nombre dans le cas de la méthode du point milieu ?

   Application numérique : vous pouvez reprendre les valeurs données par (4).

   On comparera la précision des méthodes (en prenant des valeurs de N de telle sorte que chaque méthode ait le même nombre d'appels de f).

5. On veut comparer les résultats obtenus par ces méthodes à l'instant final $t = t_0 + T$. On compare la solution analytique $x(t_0 + T)$ (instant final) et les solutions numériques z_N (pour le dernier itéré $n = N$).

   On prend $f(t, u) = -u + \sin(t)$, $x(0) = 1$. Dans ce cas, (1) a pour solution exacte (vérifiez-le) :

$$x(t) = \frac{3}{2}e^{-t} - \frac{1}{2}\cos(t) + \frac{1}{2}\sin(t).$$

   On prendra $t_0 = 0,\ T = 10$ et on fera varier $N \in \{10, 100, 1\,000, 10\,000\}$.

   (a) Écrire une fonction scilab `[TV] = compar(a, t0, T)` qui remplit une tableau TV à 3 colonnes, contenant les valeurs z_N obtenues pour par les 3 méthodes ci-dessus (Euler, point milieu, RK4) et pour chaque valeur de N (on prendra le même N pour les 3 méthodes).

   (b) Compléter cette fonction en `[TV, TE] = compar(a, t0, T)`, qui remplit de plus un tableau TE à 3 colonnes, contenant les erreurs $x(t_N) - z_N$ en $t = T$, obtenues par chacune des méthodes ci-dessus et pour chaque valeur de N.

   (c) Analyser les erreurs obtenues en fonction des résultats théoriques sur l'ordre de chacune de ces méthodes.

   (d) Tracer sur un même graphique les trois courbes représentatives de $-\log_{10}(Erreur)$ en fonction de $\log_{10}(N)$.

   (e) On approche chacune de ces courbes par une droite, au sens des moindres carrés. Pour cela, on pourra utiliser la fonction `reglin` de scilab. Commenter les valeurs des pentes de ces droites.