MT09-TP3

1. **Résolution d'un sytème linéaire dont la matrice est tridiagonale**.

On a vu dans le TD2 l'algorithme de Richtmayer qui permet de résoudre un sytème d'équations linéaires $Ax = u$ lorsque la matrice A est tridiagonale, $A = \begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \dots & 0 & a_n & b_n \end{pmatrix}$.

$$\left\{ \begin{array}{l} x_i = e_i x_{i+1} + f_i, \quad \text{pour} \quad i = 1, \dots, n-1 \\ x_n = f_n \end{array} \right. , \quad \text{avec} \quad \left\{ \begin{array}{l} e_1 = -\dfrac{c_1}{b_1}, \ f_1 = \dfrac{u_1}{b_1} \\ e_i = -\dfrac{c_i}{a_i e_{i-1} + b_i}, \quad \text{pour } 2 \leq i \leq n-1 \\ f_i = \dfrac{u_i - a_i f_{i-1}}{a_i e_{i-1} + b_i}, \quad \text{pour } 2 \leq i \leq n \end{array} \right.$$

Écrire une fonction Scilab

$$[x] = rich(a, b, c, u)$$

qui, étant donné les vecteurs a, b, c, u, calcule la solution du système $Ax = u$ par la méthode précédente.

Application : utiliser la fonction précédente pour déterminer la solution de $Ax = u$ dans le cas :

$$A = \begin{pmatrix} 2 & 3 & 0 & 0 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 0 & 2 & 1 & 2 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 & 1 \end{pmatrix}, u = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 3 \\ 1 \end{pmatrix}.$$

2. **Tracé d'une fonction définie par morceaux** .

On se donne $T_1 < \dots < T_n$, n réels ordonnés, cc une matrice de taille $(n-1) \times 4$.

On définit la fonction g par :

$$\left\{ \begin{array}{lll} g(t) & = & cc_{i1} + cc_{i2}(t - T_i) + cc_{i3}(t - T_i)^2 + cc_{i4}(t - T_i)^2(t - T_{i+1}), \quad \text{pour } T_i \leq t < T_{i+1} \\ g(t) & = & cc_{n-1,1} + cc_{n-1,2}(t - T_{n-1}) + cc_{n-1,3}(t - T_{n-1})^2 + cc_{n-1,4}(t - T_{n-1})^2(t - T_n), \quad \text{pour } t = T_n \end{array} \right. \tag{1}$$

On veut tracer la courbe d'équation $z = g(t), \ T_1 \leq t \leq T_n$.

(a) Écrire une fonction Scilab

$$[i] = place(T, t)$$

qui étant donné le vecteur T (ordonné), étant donné le nombre t, détermine la valeur de l'indice i pour laquelle $T_i \leq t < T_{i+1}$ et affiche un message d'erreur dans le cas où $t < T_1$ ou $t > T_n$. Dans le cas où $t = T_n$, on veut que $i = n - 1$.

Un algorithme de dichotomie possible est le suivant :

$imin \leftarrow 1$
$imax \leftarrow n$
tant que $(imax - imin) > 1$ **faire**
  $mil =$ partie entière $((imax + imin)/2)$
  si $t \geq T_{mil}$ **alors**
    $imin \leftarrow mil$
  sinon
    $imax \leftarrow mil$
  fin si
fin tant que
$i \leftarrow imin$

Application : On définit, $T = (1 \ 3 \ 4.5 \ 5 \ 6)^T$, utilisez la fonction précédente pour déterminer i dans les cas suivants : $t = -5, t = 1.5, t = 4.5, t = 6, t = 1, t = 20$

(b) Écrire une fonction Scilab

$$[z] = calcg(t, T, cc)$$

qui, étant donné le nombre t, le vecteur T, la matrice de coefficients cc, calcule $z = g(t)$ définie par la relation (1)

Application : On choisit : $T = \begin{pmatrix} 1 \\ 3 \\ 4.5 \\ 5 \\ 6 \end{pmatrix}$, $cc = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 5 & 0 & -8/9 & 0 \\ 3 & 0 & 16 & 0 \\ 7 & 0 & -8 & 0 \end{pmatrix}$, calculer $g(t)$ pour les valeurs $t = 3, t = 5$.

(c) Écrire une fonction Scilab

$$trace(N, T, cc)$$

qui étant donné le vecteur T de taille n, étant donné la matrice cc de taille $(n-1) \times 4$, trace la courbe d'équation $z = g(t)$ à l'aide de N valeurs $(t_j, g(t_j)),\ 1 \leq j \leq N)$. Pour construire le vecteur $(t_1, t_2, ..., t_N)$, vous pouvez utiliser la commande $linspace$, pour le tracé utilisez la fonction $plot$, voir le Chapitre 5 de la documentation Scilab.

Application : Tracer la courbe d'équation $z = g(t),\ T_1 \leq t \leq T_n$ en choisissant les valeurs de T et cc précédentes et en essayant $N = 10$ puis $N = 200$.

3. **Interpolation par spline cubique**.

Étant donnés $T_1 < \ldots < T_n$, n réels ordonnés, et $y_1, \ldots, y_n$, n réels quelconques, on veut déterminer une fonction spline cubique qui interpole les points (T_i, y_i), c'est-à-dire qu'on cherche une fonction g telle que :

- $g(T_i) = y_i,\ 1 \leq i \leq n$: propriété d'interpolation.
- la fonction g est définie par (1) : g est cubique par morceaux.
- la fonction g est continue ainsi que ses dérivées premières et secondes : propriété de régularité des fonctions splines.

On montrera dans le TD5 qu'il faut alors construire les coefficients cc de la manière suivante :

- $h_i = T_{i+1} - T_i,\ 1 \leq i \leq n-1$
- les coefficients $(d_i, i = 1, \ldots, n)$ vérifient le sytème $Ad = u$, avec

$$A = \begin{pmatrix} \frac{2}{h_1} & \frac{1}{h_1} & 0 & \ldots & 0 \\ \frac{1}{h_1} & \frac{2}{h_1} + \frac{2}{h_2} & \frac{1}{h_2} & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & 0 & \frac{1}{h_{n-2}} & \frac{2}{h_{n-2}} + \frac{2}{h_{n-1}} & \frac{1}{h_{n-1}} \\ 0 & \ldots & 0 & \frac{1}{h_{n-1}} & \frac{2}{h_{n-1}} \end{pmatrix},\ u = 3 \begin{pmatrix} \frac{y_2 - y_1}{h_1^2} \\ \frac{y_2 - y_1}{h_1^2} + \frac{y_3 - y_2}{h_2^2} \\ \ddots \\ \frac{y_{n-1} - y_{n-2}}{h_{n-2}^2} + \frac{y_n - y_{n-1}}{h_{n-1}^2} \\ \frac{y_n - y_{n-1}}{h_{n-1}^2} \end{pmatrix}$$

  On peut remarquer que la matrice A est à diagonale strictement dominante (revoir le TD2).

- $cc_{i1} = y_i, cc_{i2} = d_i, cc_{i3} = \dfrac{y_{i+1} - y_i}{h_i^2} - \dfrac{d_i}{h_i}, cc_{i4} = \dfrac{d_{i+1} + d_i}{h_i^2} - 2\dfrac{y_{i+1} - y_i}{h_i^3}, 1 \leq i \leq n-1$

(a) Écrire une fonction Scilab

$$[d] = cald(T, y)$$

qui, étant donnés les vecteurs T et y de taille n, calcule les coefficients d_i à l'aide de la fonction $rich$.

(b) Écrire une fonction Scilab

$$[cc] = calcoef(T, y)$$

qui, étant donnés les vecteurs T et y de taille n, calcule la matrice cc de taille $(n-1) \times 4$.

Application : On donne $T = (1\ 3\ 4.5\ 5\ 6)^T, y = (1\ 5\ 3\ 7\ -1)^T$, utiliser tout ce qui précède pour tracer la courbe d'équation $z = g(t),\ T_1 \leq t \leq T_n$.