Université de Jijel
Département de Mathématiques

ANALYSE NUMÉRIQUE I

TP - devoir

Présentation de la méthode. L'interpolation polynômiale peut présenter des problèmes de stabilité et de convergence (cf. l'exemple de la fonction de Runge). En général, beaucoup de ces problèmes peuvent-être évités en utilisant une interpolation polynômiale par intervalles avec des polynômes de degré pas trop élévé. La fonction d'interpolation ainsi obtenue est suffisamment régulière et ne présente pas d'oscillations.

L'idée de l'interpolation par des splines cubiques est d'interpoler la fonction localement (i.e. sur chaque sous-intervalle) par des polynômes de degré 3 tout en garantissant que la fonction d'interpolation est globalement régulière.

Définition. Soit f une fonction définie sur un intervalle $I = [a, b]$ avec $a < b$ et considérons l'ensemble des points équidistants dans $[a, b]$,

$$x_i = a + (i - 1)h, \quad i = 1, ..., n + 1, \quad h = \tfrac{b-a}{n}.$$

Un spline cubique d'interpolation de f est une fonction S telle que

- $S(x_i) = f(x_i), \quad i = 1, \cdots, n + 1$
- Dans chaque sous-intervalle $[x_i, x_{i+1}]$, S coincide avec un polynôme de degré 3
- $S \in C^2[a, b]$.

On peut montrer (voir les notes de cours) que le *spline cubique naturel* S est défini par

$$S(x) = S_i(x) \qquad \text{para } x \in [x_i, x_{i+1}]$$

avec

$$\begin{aligned} S_i(x) = {} & \tfrac{\mathbf{m}_i}{6h}(x_{i+1} - x)^3 + \tfrac{\mathbf{m}_{i+1}}{6h}(x - x_i)^3 + \tfrac{1}{h}\left(f(x_i) - \mathbf{m}_i \tfrac{h^2}{6}\right)(x_{i+1} - x) \\ & + \tfrac{1}{h}\left(f(x_{i+1}) - \mathbf{m}_{i+1}\tfrac{h^2}{6}\right)(x - x_i) \qquad i = 1, \cdots, n \end{aligned}$$

où $\mathbf{m}_1 = 0$, $\mathbf{m}_{n+1} = 0$ et où les coefficients $\mathbf{m}_i = S"(x_i)$ $(i = 2, \cdots, n)$ satisfont le système linéaire suivant

$$\underbrace{\begin{pmatrix} 4 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & 4 & 1 \\ 0 & \cdots & \cdots & 0 & 1 & 4 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \mathbf{m}_2 \\ \mathbf{m}_3 \\ \mathbf{m}_4 \\ \vdots \\ \mathbf{m}_{n-1} \\ \mathbf{m}_n \end{pmatrix}}_{\mathbf{m}} = \underbrace{\begin{pmatrix} \mathbf{s}_2 \\ \mathbf{s}_3 \\ \mathbf{s}_4 \\ \vdots \\ \mathbf{s}_{n-1} \\ \mathbf{s}_n \end{pmatrix}}_{s},$$

avec

$$\mathbf{s}_i = \tfrac{6}{h^2} \left(f(x_{i+1}) - 2f(x_i) + f(x_{i-1}) \right).$$

1. Implémenter un programme qui, donnant $a,\ b \in \mathbb{R}$ avec $a < b$, une fonction f et $n \in \mathbb{N}$, permet de construire une fonction spline S.

2. Considère la fonction de Runge

$$f(x) = \frac{1}{x^2 + 1}$$

définie dans l'intervalle $[-5, 5]$. Soit $m = 100$ et considère l'ensemble des points

$$y(k) = a + \frac{k-1}{mn}(b - a) \qquad k = 1, \cdots, mn + 1$$

obtenus en divisant $[-5, 5]$ en mn sous-intervalles de même longueur (ou, de manière équivalente, divisant chaque sous-intervalle $[x_i, x_{i+1}]$ en m sous-intervalles de même longueur). Utisant l'alinéa **1.**, déterminer le vecteur z défini par

$$z(k) = S(y(k)) \qquad k = 1, \cdots, mn + 1$$

et construire un tableau de la forme

n	$E_n = \max\limits_{j=1,\cdots,nm+1} \lvert z(k) - f(y(k)) \rvert$
5	
10	
20	
40	
80	
160	
320	

3. L'objectif de la dernière partie est de déterminer numériquement l'ordre de la méthode. Rappelons qu'une méthode est d'ordre p, si l'erreur correspondant au pas $h_n = \frac{b-a}{n}$ satisfait

$$E_n \approx Ch_n^p = C\left(\frac{b-a}{n}\right)^p \tag{1}$$

où C est une constante positive indépendante de n. Déduire de (1) que

$$p \approx \frac{\ln\left(\frac{E_n}{E_{2n}}\right)}{\ln 2}. \tag{2}$$

où E_{2n} est l'erreur correspondant à $h_{2n} = \frac{h_n}{2}$. Utilisant (2) et les résultats du tableau précédent, construire un autre tableau de la forme

n	$p_n = \frac{\ln\left(\frac{E_n}{E_{2n}}\right)}{\ln 2}$
5	
10	
20	
40	
80	
160	

Commenter les résultats obtenus.

Note.

i) **Présenter un court rapport avec les sections suivantes: réponse à la question 2, réponse à la question 3 et conclusions.**

ii) Dans *Matlab*, vous pouvez utiliser la commande $A\backslash b$ pour résoudre le système $Ax = b$.

iii) Pour tester le programme, vous pouvez dessiner les graphiques de f et de S en utilisant la commande

plot($y, 1./(1 + y.\,^2)$,'r',y, z,'b');

où y et z sont les vecteurs définis dans l'alinéa **2**.