MAT-2400 Méthodes numériques
Labo 4 et Devoir 4
7 novembre 2025

- Avant 9h30 aujourd'hui, inscrire votre équipe sur monportail. Cela ne peut pas être la même équipe que pour un devoir précédent.
- Avant 10h20, il faut avoir répondu à la question 1. Déposer dans la boîte de dépôt votre programme *labo4.m* et le programme *polynome_ newton.m* (qui ne doit pas être modifié) de la banque de programmes associée au manuel, compressés en 1 seul fichier. Il n'y a pas de rapport à remettre.
- Avant jeudi 13 novembre 23h59, déposer le rapport et les programmes demandés, **en un seul fichier compressé**, dans la boite de dépôt.
- **ATTENTION :** le rapport doit être en format **PDF**. Le convertir s'il a été produit dans un autre format.
- Pour le devoir, Il y a 1 seul programme Matlab (le script *devoir4.m*) à créer mais on demande de fournir ce fichier ainsi que les fonctions *polynome_ newton.m* et *spline_ cubique.m*.
- Lorsqu'on exécute le script *labo4.m* ou *devoir4.m*, les figures demandées doivent s'afficher. Au total il y en a 2 pour le *labo4* et 5 pour *devoir4*.
- Dans le rapport, on reproduira les figures demandées avec un commentaire pour chacune d'elles, en plus des réponses aux questions posées.

La fonction Matlab *polynome_ newton.m* qui se trouve dans la banque de programmes Matlab associée au manuel du cours (voir site web du cours pour un lien vers cette banque) permet de déterminer un polynôme d'interpolation. Pour savoir comment utiliser cette fonction Matlab et comment obtenir directement des évaluations du polynôme d'interpolation, bien lire les commentaires situés en début du programme. Vous vous rendrez compte que pour réaliser ce devoir, il suffit d'utiliser seulement les évaluations du polynôme d'interpolation, que ce soit pour tracer le graphe du polynôme en question ou pour évaluer l'erreur d'interpolation. Prendre soin de **prendre beaucoup plus de points d'évaluation que de points d'interpolation**, afin que le graphe soit représentatif et l'erreur finement évaluée. Mêmes remarques pour *spline_ cubique.m*.

Soit

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5,\ 5].$$

À la suite on propose plusieurs façons d'interpoler cette fonction, on commençant par une première méthode qui ne s'avère pas efficace pour donner un approximation uniforme de

$f(x)$ sur $[-5,\ 5]$, puis deux autres qui sont des remèdes à ce problème.

1. Soit $n \geq 1$. On pose $h = 10/n$. On considère les $n+1$ noeuds $x_i = -5 + ih$, $i = 0, 1, ..., n$, uniformément répartis sur $[-5,\ 5]$. On note $p_n(x)$ le polynôme de degré $n$ interpolant les points $(x_i, f(x_i))$, $i = 0, 1, \ldots, n$.
   Créer un script *labo4.m*, faire appel à *polynome_ newton.m* pour 3 valeurs différentes de $n$ ($n = 2,\ 4,\ 8$), et faire créer une figure (Figure 1) où sont tracés le graphe de $f$ et les graphes de ces polynômes d'interpolation. Les graphes doivent illustrer la propriété que le maximum de l'erreur absolue d'interpolation sur $[-5,\ 5]$,

   $$E(h) = \max_{-5 \leq x \leq 5} |f(x) - p_n(x)|,$$

   ne tend pas vers 0 lorsque $h \to 0$, c'est-à-dire lorsque $n \to \infty$.
   Utiliser différentes couleurs pour les différents graphes ainsi que la commande Matlab *legend*.
   Sur un autre graphique (Figure 2) faire tracer $E(h)$ en fonction de $h$. À cette fin, on estimera $E(h)$ pour différentes valeurs de $n$ ($n = 2,\ 10,\ 20,\ 30,\ 40$).

   **Pour la suite, dupliquer le script *labo4.m*. Nommer la nouvelle copie *devoir4.m*, et continuer ce devoir avec *devoir4.m*.**

2. Comme à la question 1, on considère de nouveau les noeuds $x_i$ uniformément répartis sur $[-5,\ 5]$. Pour $n$ pair, on note $Q_n(x)$ la fonction d'interpolation quadratique par morceaux telle que sa restriction à chaque $[x_i,\ x_{i+2}]$, $i = 0, 2, 4, ..., n-2$, est le polynôme d'interpolation de degré 2 interpolant $(x_i, f(x_i))$, $(x_{i+1}, f(x_{i+1}))$ et $(x_{i+2}, f(x_{i+2}))$.
   Dans le script, faire créer une figure (Figure 3) où sont tracés le graphe de $f$ et le graphe des fonctions $Q_n(x)$ pour $n = 2$, $n = 4$ et $n = 6$. Utiliser la commande *legend*.
   On définit l'erreur d'interpolation

   $$E(h) = \max_{-5 \leq x \leq 5} |f(x) - Q_n(x)|.$$

   Sur une nouvelle figure (Figure 4), tracer $E(h)$ en fonction de $h$. À cette fin, on prendra les valeurs de $n$ suivantes : $n = 2, 10, 20, ..., 50$. Le graphique doit mettre en évidence une convergence d'ordre 3 (par rapport à $h$) : $E(h) \approx C\, h^3$. On utilisera donc une échelle appropriée et on expliquera dans le rapport comment le graphique met en évidence cet ordre de convergence.

**Dans le rapport, on démontrera que $E(h) \leq Ch^3$, pour une constante positive $C$. À cette fin, on supposera qu'il existe $M > 0$ tel que $|f'''(x)|| \leq M$,** $\forall x \in [-5, 5]$**.** Indication : s'inspirer de la démonstration de l'exercice fait en classe avec de l'interpolation linéaire par morceaux.

3. On considère de nouveau les noeuds $x_i$ uniformément répartis sur $[-5, 5]$. Cette fois on interpole tous les points par un spline cubique, en utilisant la fonction *spline_ cubique.m* de la banque de programmes Matlab associée au manuel.
   Effectuer d'abord les calculs avec $n = 2, 4, 6$, puis faire créer une figure (Figure 5) où sont tracés le graphe de $f$ et le graphe des 3 splines cubiques $S_n(x)$. Utiliser la commande *legend*.
   On définit l'erreur d'interpolation

$$E(h) = \max_{-5 \leq x \leq 5} |f(x) - S_n(x)|.$$

   Sur la figure 4, tracer $E(h)$ en fonction de $h$, en prenant soin de garder la courbe précédente de cette figure et d'utiliser le même type d'échelle que pour la 1ère courbe. On prendra les valeurs de $n$ suivantes : $n = 2, 10, 20, 30, 40$. Utiliser la même échelle que pour la 1ère courbe. Ne pas oublier d'utiliser la commande *legend*. Dans le rapport, commenter brièvement ces graphiques.

Pour tous les graphiques, ne pas oublier de mettre un titre et d'indiquer ce qu'il y a en abscisse et en ordonnée (commandes *title*, *xlabel* et *ylabel*). Et d'utiliser la commande *legend* lorsqu'il y a plusieurs courbes sur le même graphique.