TP5 : Interpolation de Lagrange

Commencez par créer un répertoire `TP5/` dans lequel vous travaillerez pour ce TP.

Rappel de cours :
Soient $x_0, .., x_n$ $n+1$ points équirépartis sur l'intervalle $[0, 1]$. Si on pose $h = \frac{1}{n}$, on a donc $x_i = i * h$.
Le polynome d'interpolation de Lagrange de f relativement aux points x_i s'écrit :

$$p_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

avec $L_i(x) = \prod_{j=0, i \neq j}^{n} \frac{x - x_j}{x_i - x_j}$

Première partie. Algorithmique.

Soient (x_i, f_i), $i = 0 : n$, des points d'interpolation. Le but de cette partie est d'écrire et de comparer différents algorithmes d'évaluation en un point x quelconque du polynôme d'interpolation de Lagrange associé.

1. Ecrire un algorithme utilisant les polynômes de base de Lagrange. Déterminer la complexité de cet algorithme.
2. Même question en utilisant les différences divisées. On écrira d'abord un algorithme en supposant les différences divisées déjà calculées. Cet algorithme devra avoir la complexité la plus faible possible, compte tenu de l'expression particulière du polynôme en fonction des différences divisées. On écrira ensuite un algorithme pour le calcul de ces différences divisées.
3. Lequel de ces deux algorithmes est le plus efficace ?

Deuxième partie. Programmation de l'interpolation de Lagrange de deux manières.

Exercice 1.

Le but de cet exercice est de calculer une valeur approchée d'une fonction en un point par une méthode d'interpolation de Lagrange en utilisant les fonctions de base.
Créez un répertoire `Exo1` et placez vous à l'intérieur.
Dans un fichier `exo1.f90` :

1. Définissez une constante entière nommé `n` qui vaut 10

2. Calculez le polynome d'interpolation de Lagrange p_n en $x = 0.55$ de la fonction $f(x) = \dfrac{x}{sqrt(x)}$
3. Imprimez à l'écran le résultat.
4. Vérifiez que votre programme est valide : copiez les fichiers `exo1.sh` et `gradeexo1.sh` qui se trouve dans le répertoire `/net/autre/Matmeca/dobrzyns/Public/TP5` dans votre répertoire `TP5` puis dans ce même répertoire, tapez `./gradeexo1.sh`. Une note doit apparaitre et votre programme doit s'exécuter et afficher la valeur du polynôme.

Exercice 2. Le but de cet exercice est de calculer une valeur approchée d'une fonction en un point par une méthode d'interpolation de Lagrange en utilisant les différences divisées.
Créez un répertoire `Exo2` et placez vous à l'intérieur.

Dans un fichier `exo2.f90` :

1. Définissez une constante entière nommé `n` qui vaut 10
2. Calculez le polynome d'interpolation de Lagrange en $x = 0.55$ de la fonction $f(x) = \dfrac{x}{sqrt(x)}$
3. Imprimez à l'écran le résultat.

Troisième partie. Gnuplot.
Gnuplot est un logiciel qui sert à représenter graphiquement en deux ou trois dimensions des fonctions numériques ou des données.

1. Pour lancer gnuplot, tapez `gnuplot` dans un terminal. Taper `quit` pour quitter.
2. Quelques commandes pour tracer des courbes :
   - `plot` permet de tracer une courbe.
     Exemple : `plot x*x+cos(4*x)` ou `plot[-1:1] x*x+cos(4*x)`
   - pour fixer les axes : `set xrange[-10:10]` et `set yrange[-10:10]`
   - pour effacer les valeurs définies : `reset x` et `reset y`
   - pour donner des noms aux axes : `set xlabel "axe X"`
   - pour tracer plusieurs courbes : `replot` ou `plot` suivi de toutes les fonctions à tracer espacées d'une virgule
     Exemple : `plot x*x*x` puis `replot x-4*x*x` ou `plot x*x*x, x-4*x*x`
3. Pour lire des informations dans un fichier :
   - première étape : avoir un fichier dans son répertoire contenant des colonnes de chiffres.
     Exemple : copier dans un fichier nommé `ex.dat` les lignes suivantes (attention vous devez être hors de gnuplot pour faire cela) :

```
1 1 45.4
2 0.5 18.3
3 1 3.
4 10 20.
5 3 10.
```

- dans gnuplot, on utilise la fonction `plot` de la manière suivante :
  `plot 'ex.dat' u 1:2` ce qui signifie que la première colonne de ex.dat contiendra les abscisses des points et la deuxième colonne les ordonnées.
  `plot 'ex.dat' u 2:3` ce qui signifie que la deuxième colonne de ex.dat contiendra les abscisses des points et la troisième colonne les ordonnées.

- pour relier les points, il faut rajouter `w l`.
  Exemple : `plot 'ex.dat' u 2:3 w l`
- on peut bien sûr tracer plusieurs séries de points ou courbes :
  Exemple : `plot 'ex.dat' u 2:3 w l` puis `replot 'ex.dat' u 1:3 w l` ou `plot 'ex.dat' u 2:3 w l, 'ex.dat' u 1:3 w l`

4. Pour avoir des informations sur une commande `help`
   Exemple : `help plot`

Quatrième partie. Le phénomène de Runge.

Exercice 3.

Soient $y_0, .., y_m$ $m+1$ points équirépartis sur l'intervalle $[-1, 1]$. Soient $x_0, .., x_n$ $n+1$ points équirépartis sur l'intervalle $[-1, 1]$.

Le but de cet exercice est d'évaluer les polynômes d'interpolation de Lagrange de degré 3, 5 et 15 en tous les points y_i et de les tracer avec `gnuplot`.

Créez un répertoire `Exo3` et placez vous à l'intérieur.
Dans un fichier `exo3.f90` :

1. Définissez un vecteur `y` de taille 101 et initialisez-le avec les valeurs des y_i pour $i = 0..100$.
2. Définissez un entier `n` qui représentera le degré de votre polynôme d'interpolation. La valeur de `n` sera demandé à l'utilisateur de votre programme.
3. Calculez le polynome d'interpolation de Lagrange en $x = y(i)$ pour $i = 0..100$ de la fonction $f(x) = \dfrac{1}{1+x^2}$ et imprimez à l'écran pour chaque i la valeur de y(i) et la valeur du polynôme en ce point.
4. Compilez votre programme en nommant l'exécutable `exo3`
5. Exécutez votre programme en tapant `./exo3 > data3` en donnant une valeur 3 pour `n`. Recommencez pour `n` = 5 (resp. 15) en changeant data3 en data5 (resp. data15). Des fichiers nommés data3, data5 et data15 contenant des colonnes de chiffres existent maintenant dans votre répertoire.
6. Tracez avec `gnuplot` ces fichiers ainsi que la fonction $f(x)$. Que dire de l'approximation ?

Exercice 4. Créez un répertoire `Exo4` et placez vous à l'intérieur. Reprendre l'éxercice précédent avec la fonction $f(x) = \dfrac{1}{1 + 25 \times x^2}$. Que constatez-vous ?

Exercice 5. Une façon de résoudre ce phénomène est de décomposer l'intervalle en sous-intervalles et d'approcher la fonction par un polynôme de "bas" degré sur chaque sous-intervalle.

Ecrire un programme qui découpe l'intervalle $[-1, 1]$ en 10 sous-intervalles et qui approche la fonction de l'exercice précédent par un polynôme de degré 5. Calculez la valeur de ce polynôme pour chaque case du vecteur `y`. Puis tracez le résultat avec `gnuplot`

Il est fortement conseillé de commencer par écrire l'algorithme sur papier en vérifiant bien les bornes des sous-intervalles et les valeurs de x_i avant de se mettre à programmer.