TRAVAUX PRATIQUES 7

INTERPOLATION DE LAGRANGE

Opérations sur les polynômes

On considère un polynôme de degré n sous la forme $P(X) = a_n X^n + a_{n-1} X^{n-1} + \cdots + a_1 X + a_0$.

En Python, un polynôme est implémenté sour la forme d'une liste : `P = [a_0,a_1,...,a_n]`; par exemple, le polynôme $P(X) = X^2 + 2X + 3$ sera implémenté par : `P = [3,2,1]`.

Nous allons opter pour une **programmation modulaire**; ainsi, tous les programmes de cette partie seront enregistrés dans un fichier appelé `polpy.py`, qui pourra ensuite être importé.

1. **Création d'un polynôme**

   Dans les algorithmes suivants, nous aurons besoin d'initialiser un polynôme d'un degré donné. Pour cela, nous allons écrire une fonction `Creer(n)` qui prend en argument un entier naturel n et qui revoie le polynôme de $\mathbb{R}_n[X]$ dont tous les cœfficients sont nuls.

   Par exemple : `Creer(4)` $\longrightarrow$ `[0,0,0,0,0]`.

2. **Degré d'un polynôme**

   En fonction des opérations que nous devons effectuer sur les polynômes, nous devons accéder facilement à son degré. Pour cela nous allons écrire une fonction `Degre(P)` qui prend en argument une liste `P` correspondant à un polynôme et qui revoie le degré du polynôme considéré.

   Par exemple : `Degre([1,1,1,1,1])` $\longrightarrow$ `4` et `Degre([0,1,0,1,0])` $\longrightarrow$ `3`.

3. **Multiplication par un scalaire**

   Dans l'espace vectoriel des polynômes, il est fréquent de multiplier un polynôme par un scalaire. Pour cela nous allons écrire une fonction `Scalaire(P,s)` qui prend en arguments une liste `P` et un flottant `s` et qui revoie la liste correspondant au produit du polynôme P par le scalaire s.

   Par exemple, `Scalaire([3,2,1],4)` $\longrightarrow$ `[12,8,4]`.

4. **Addition de deux polynômes**

   Dans l'espace vectoriel des polynômes, la deuxième opération incontournable est l'addition de deux polynômes. Il est possible bien sûr d'additionner deux polynômes de degrés différents; mais pour les listes il peut être préférable d'additionner terme à terme deux listes de même taille. Par exemple, si $P(X) = X^2 + 2X + 3$ et $Q(X) = X - 4$, alors on pourra créer et utiliser les listes `[3,2,1]` et `[-4,1,0]` pour effectuer l'addition.

   Ecrire alors une fonction `Additionner(P,Q)` qui prend en argument deux listes représentant deux polynôme P et Q et qui renvoie la liste correspondant à la somme de P et de Q.

   Par exemple : `Additionner([3,2,1],[-4,1])` $\longrightarrow$ `[-1,3,1]`.

   Ce qui correspond bien à l'addition $\left(X^2 + 2X + 3\right) + \left(X - 4\right) = X^2 + 3X - 1$.

5. **Produit de deux polynômes**

   Soient $P(X) = \sum_{i=0}^{p} a_i X^i$ et $Q(X) = \sum_{j=0}^{q} b_j X^j$ deux polynômes de degrés respectifs p et q, alors le produit PQ est un polynôme de degré $p + q$ et le cœfficients de X^k est donné par $\sum_{i=0}^{k} a_i b_{k-i}$.

   Ecrire alors une fonction `Multiplier(P,Q)` qui prend en argument deux listes représentant deux polynôme P et Q et qui renvoie la liste correspondant au produit de P et de Q.

   Par exemple : `Multiplier([3,2,1],[-4,1])` $\longrightarrow$ `[-12,-5,-2,1]`.

   Ce qui correspond bien à la multiplication : $\left(X^2 + 2X + 3\right)\left(X - 4\right) = X^3 - 2X^2 - 5X - 12$.

6. **Intégrer un polynôme**

   Soit $P(X) = \sum_{i=0}^{p} a_i X^i$ un polynôme de degré p. A l'instar du polynôme dérivé on peut définir le polynôme intégré comme $\sum_{i=0}^{p} \frac{a_i}{i+1} X^{i+1}$; il s'agit d'un polynôme de degré $p+1$ qui coïncide avec la primitive de $P(x)$ qui s'annule en 0.

   Ecrire alors une fonction `Integrer(P)` qui prend en argument une liste représentant un polynôme P et qui renvoie la liste correspondant au polynôme intégré de P.

   Par exemple : `Integrer([3,2,1])` $\longrightarrow$ `[0,3.0,1.0,0.3333333333333333]`.

   Ce qui correspond bien à la l'intégration de $X^2 + 2X + 3$ qui donne $\frac{1}{3}X^3 + X^2 + 3X$.

7. **Evaluer un polynôme**

   Ecrire une fonction `Evaluer(P,x)` qui prend en arguments une liste `P` correspondant à un polynôme P et un flottant `x` ; cette fonction renvoie la valeur de $P(x)$. On cherchera à écrire une fonction de complexité linéaire.

   Par exemple : `Evaluer([3,2,1],2)` $\longrightarrow$ `11`. Ce qui correspond bien au calcul $P(2) = 2^2 + 2 \times 2 + 3$.

Polynômes interpolateurs de Lagrange

On considère une liste $A = [a_0, a_1, ..., a_n]$ constituée de $n+1$ réels deux à deux distincts appelés **nœuds**. On définit alors le **polynôme de Lagrange** d'indice i par :

$$L_i(X) = \prod_{j=0, j\neq i}^{n} \frac{X - a_j}{a_i - a_j} = \frac{X - a_0}{a_i - a_0} \times \cdots \times \frac{X - a_{i-1}}{a_i - a_{i-1}} \times \frac{X - a_{i+1}}{a_i - a_{i+1}} \times \cdots \times \frac{X - a_n}{a_i - a_n}$$

Il s'agit d'un polynôme de degré n qui vérifie : $\begin{cases} L_i(a_j) = 0 & \text{si } j \neq i \\ L_i(a_i) = 1 \end{cases}$

8. **Polynômes de Lagrange**

   En utilisant les fonctions définies précédemment, écrire une fonction `Lagrange(A,i)` qui prend en arguments une liste `A` de nœuds et un entier `i` ; cette fonction renvoie le polynôme de Lagrange d'indice i.

   Par exemple :

   `Lagrange([1,2,3],0)` $\longrightarrow$ `[3.0,−2.5,0.5]`
   `Lagrange([1,2,3],1)` $\longrightarrow$ `[−3.0,4.0,−1.0]`
   `Lagrange([1,2,3],2)` $\longrightarrow$ `[1.0,−1.5,0.5]`

Maintenant, considérons $n+1$ points $\big((a_0, b_0), (a_1, b_1), ..., (a_n, b_n)\big)$ tels que les réels $(a_i)_{0\leqslant i\leqslant n}$ sont deux à deux distincts. Alors il existe un unique polynôme de degré au plus n qui passe exactement par ces n points ; il s'agit du **polynôme interpolateur de Lagrange** défini par :

$$L(X) = \sum_{i=0}^{n} b_i L_i(X) = \sum_{i=0}^{n} b_i \prod_{j=0, j\neq i}^{n} \frac{X - a_j}{a_i - a_j}$$

9. **Polynôme interpolateur de Lagrange**

   On considère deux listes, la liste $A = [a_0, a_1, ..., a_n]$ des abscisses des points (nœuds) et la liste $B = [b_0, b_1, ..., b_n]$ des ordonnées des points.

   En utilisant les fonctions définies précédemment, écrire une fonction `Interpoler(A,B)` qui prend en arguments deux liste `A` et `B`, et qui renvoie le polynôme interpolateur de Lagrange. Par exemple :

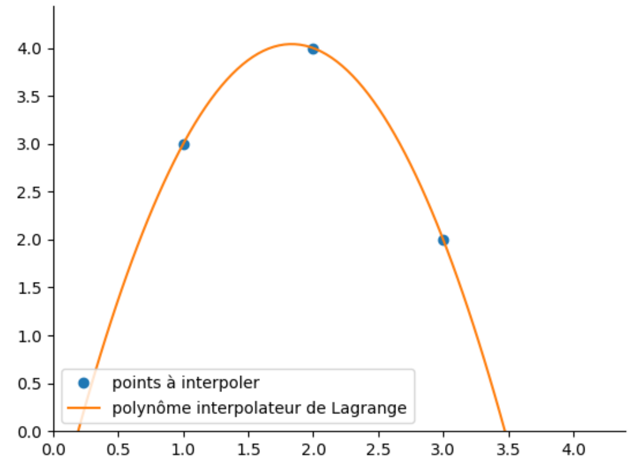
   `Interpoler([1,2,3],[3,4,2]` $\longrightarrow$ `[−1.0,5.5,−1.5]`

   Pour visualiser le résultat précédent, on peut réaliser le graphique suivant :

```
import numpy as np
import matplotlib.pyplot as plt

A=[1,2,3]
B=[3,4,2]

plt.close()
plt.plot(A,B,'o',label='points à interpoler')
X=np.linspace(0,4,100)
Y=[pl.Evaluer(Interpoler(A,B),x) for x in X]
plt.plot(X,Y,label='polynôme interpolateur de
                    Lagrange')
plt.legend()
plt.show()
```



Application : interpolation polynomiale d'une fonction

Dans cette partie, nous considérons la fonction définie sur $\mathbb{R}$ par $f(x) = \dfrac{4}{1+x^2}$.

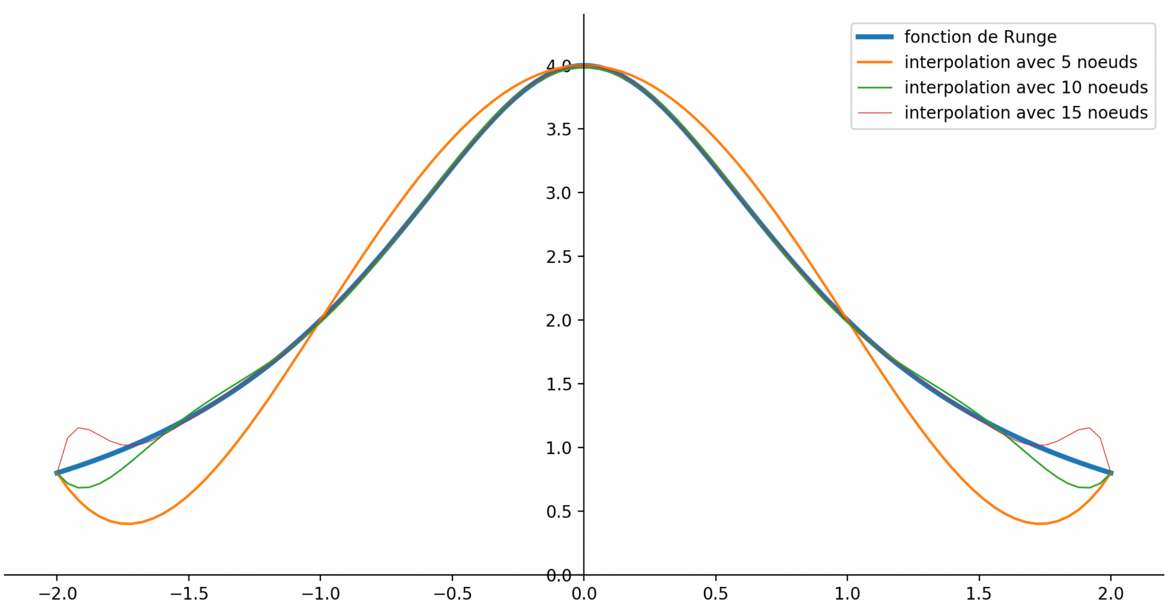
Dans un premier temps nous allons construire des polynômes interpolateurs de Lagrange qui permettent d'approcher cette fonction f et constater le **phénomène de Runge**. Dans un deuxième temps nous utilisons cette interpolation polynomiale pour calculer une intégrale.

Pour obtenir une interpolation polynomiale de la fonction f sur l'intervalle $[-2,2]$ par exemple, il suffit de définir un nombre n de nœuds répartis dans cet intervalle et calculer les images de ces nœuds par f.

Par exemple si nous choisissons 5 nœuds équirépartis, nous obtenons $A = [-2,-1,0,1,2]$ et $B = [0.8,2,4,2,0.8]$.

Le polynôme interpolateur de Lagrange correspondant aux listes A et B donnera alors l'approximation polynomiale de f à 5 nœuds.

On peut alors penser que plus le nombre de nœuds augmentent et plus l'approximation est bonne; mais en 1901, le mathématicien allemand Carl RUNGE découvrit un résultat contraire à l'intuition : il existe des configurations où l'écart maximal entre la fonction et son interpolation peut augmenter avec n, comme le montre le graphique suivant :



10. Donner une séries d'instructions Python qui permettent de réaliser un graphique mettant en évidence le phénomène de Runge.

11. **Calcul d'une intégrale**

    Dans cette question on s'intéresse à l'intégrale $\int_0^1 f(x)\mathrm{d}x = \int_0^1 \frac{4}{1+x^2}\mathrm{d}x$.
    L'objectif est de calculer cette intégrale au moyen de polynômes. En utilisant la technique d'interpolation polynomiale de f et les fonctions définies dans ce TP, donner une série d'instructions qui permettent ce calcul.
    Donner le script de la fonction qui permet de calculer l'intégrale précédente par la méthode des trapèzes. Comparer les résultats.