TP 17 informatique

BCPST 1 2019-2020

Points abordés

- Listes.
- Polynômes.

Introduction

L'objectif est d'implémenter différents algorithmes nous permettant de faire des calculs avec des polynômes. On rappelle qu'un polynôme est un objet qui s'écrit sous la forme :

$$P(X) = a_0 + a_1X + a_2X^2 + \cdots + a_nX^n.$$

Ainsi, on peut représenter le polynôme P en python à l'aide de la liste de ses coefficients :

$$L = [a_0, a_1, a_2, \cdots, a_n].$$

Sur les polynômes, il existe différentes opérations élémentaires :

1. la somme, 2. le produit, 3. la dérivation, 4. le calcul de $P(a)$ pour un réel a.

Une question naturelle est alors d'implanter ces opérations en python, pour effectuer des calculs sur les polynômes à l'aide de l'ordinateur.

Exercices

Exercice 1. Un premier problème concerne l'unicité de la représentation. En effet, deux listes différentes peuvent correspondre à un même polynôme. Par exemple, $L = [0, 0, 1, 0]$ et $M = [0, 0, 1]$ correspondent toutes les deux au polynôme X^2. De même, le polynôme nul peut être représenter par [] ou par [0].

1. Écrire une fonction `polEgaux(L,M)` qui retourne `True` si L et M correspondent au même polynôme et `False` sinon.
2. Écrire une fonction `reduire(L)` qui supprime de L les 0 "inutiles" pour représenter P.
3. Une notion fondamentale sur les polynômes est le degré. Écrire une fonction `degre(P)` qui retourne le degré du polynôme P. Par convention, le degré du polynôme nul sera -1.

Remarque 1. En mathématiques, le degré du polynôme nul est $-\infty$. Mais il est plus facile, pour l'implémentation de le représenter ici par -1.

Exercice 2. Écrire une fonction `derive(P)` qui retourne une liste représentant P'.

Exercice 3. Écrire une fonction `somme(P,Q)` qui prend en argument deux listes représentant des polynômes et qui retourne la somme des deux.

Exercice 4. Soient $P(X) = \sum_{k=0}^{m} a_kX^k$ et $Q(X) = \sum_{k=0}^{n} b_kX^k$ deux polynômes de degrés respectifs m et n. Le produit est alors de degré $m + n$. Si on note $R(X) = \sum_{k=0}^{m+n} c_kX^k$ le produit, on rappelle que l'on a : $\forall k \in \{0, \cdots, m + n\}, c_k = \sum_{i=0}^{k} a_ib_{k-i}$. Écrire une fonction `Produit(P,Q)` qui retourne le produit PQ.

Exercice 5. On rappelle que pour évaluer en un point, on remplace X par valeur a. Écrire une fonction `evaluation(P,a)` qui retourne la valeur $P(a)$.

Exercice 6. Une des méthodes pour évaluer de manière efficace un polynôme en un point est appelée méthode de Horner. Elle consiste à remarquer le fait suivant :

$$a_0 + a_1X + a_2X^2 + a_nX^n = a_0 + X(a_1 + X(a_2 + \cdots + X(a_{n-1} + a_nX))\cdots)$$

Pour calculer $P(a)$, on effectue alors les affectations suivantes :

1. $S = a_n$, 2. $S = a_na$, 3. $S = S + a_{n-1}$, 4. $S = Sa$,
5. $S = S + a_{n-2}$, 6. $\cdots$ 7. $S = S + a_1$, 8. $S = Sa$, 9. $S = S + a_0$.

Par exemple, pour évaluer le polynôme X^3+5X^2+2X+1 en 2, on a donc successivement : 1,2,7,14,16,32,33. On a donc $P(2) = 33$.

Écrire une fonction `Horner(P,a)` qui calcule $P(a)$ à l'aide de la méthode de Horner.

Exercice 7. On considère la suite de polynômes définie par la récurrence suivante :

$$P_0 = 1, P_1 = X, \forall n \in \mathbb{N}, P_{n+2} = (n + 1)XP_{n+1} + 2P_n.$$

Écrire une fonction `poly(n)` qui renvoie le polynôme P_n. On pourra utiliser les fonctions précédentes.