TP 4 : Fonctions polynomiales

On rappelle que $\mathbf{R}_n[X]$ désigne l'ensemble des polynômes de degré au plus n. C'est un espace vectoriel.

On choisit de manipuler les polynômes en utilisant les listes comme structure de données pour représenter les coefficients. Un polynôme P de degré *au plus* n peut être représenté par une liste de longueur $n+1$, où chaque élément de la liste correspond au coefficient du monôme de degré correspondant. Par exemple, le polynôme $P = 3X^3 + 2X - 5$ vu comme élément de $\mathbf{R}_3[X]$ sera représenté par la liste `[ -5, 2, 0, 3 ]`, tandis que P, vu comme un élément $\mathbf{R}_4[X]$ sera représenté par la liste `[ -5, 2, 0, 3, 0 ]`.

■ **Remarque 1.** Le polynôme nul est particulier car tous ses coefficients sont nuls. En fonction de l'espace vectoriel $\mathbf{R}_n[X]$ dans lequel il est défini, il sera représenté par une liste de $n+1$ coefficients tous nuls : par exemple, dans $\mathbf{R}_3[X]$, il sera représenté par `[0, 0, 0, 0]`. Par convention, on fixe son degré à -1.

■ **Exercice 1.** Écrire une fonction `degre(p)` qui prend en entrée une liste `p` représentant un polynôme et renvoie le degré de ce polynôme (attention, le degré d'un polynôme n'est pas forcément égal à la longueur de la liste utilisée pour le représenter). Pour le polynôme nul, la fonction doit renvoyer -1 conformément à la convention.
Testez votre fonction avec plusieurs listes, par exemple `[1, 0, 2, 3]` (qui représente $P = 3X^3 + 2X^2 + 1$), ainsi qu'avec des polynômes de degrés inférieurs, et surtout avec le polynôme nul.

■ **Exercice 2.** Écrire une fonction `eval(p, x)` qui prend en entrée (une liste représentant) un polynôme `p`, un flottant `x`, et qui renvoie en sortie la valeur de ce polynôme en `x`.
Testez la fonction avec différents flottants et polynômes.

■ **Exercice 3.** Écrire une fonction `add(p, q)` qui prend en entrée deux (listes représentant des) polynômes `p` et `q`, et renvoie une liste représentant leur somme.
La fonction doit pouvoir gérer des polynômes de degrés différents. Par exemple, pour `p = [1, 0, 3]` et `q = [0, 5, 2, 1]`, le résultat devrait être `[1, 5, 5, 1]`.
Vérifiez votre implémentation en testant la fonction avec plusieurs exemples.

■ **Exercice 4.** Écrire une fonction `scal_mult(p, alpha)` qui prend en entrée une liste `p` représentant un polynôme P, un flottant `alpha`, et renvoie en sortie une nouvelle liste représentant le polynôme αP. Par exemple, pour `p = [1, 2, 3]` et `alpha = 2`, le résultat doit être `[2, 4, 6]`.

■ **Exercice 5.** Rappeler la formule donnant le produit de deux polynômes.

■ **Exercice 6.** Écrire une fonction `multiply(p,q)` qui prend en entrée deux polynômes `p` et `q`, et renvoie une liste représentant leur produit.
Par exemple, pour `p = [1, 1]` et `q = [1, -1]`, le résultat devrait être `[1, 0, -1]`.

■ **Exercice 7.** Écrire une fonction `derive(p)` qui prend en paramètre un polynôme `p` et renvoie sa dérivée sous forme de liste.
Par exemple, pour `p = [3, 0, 5, 2]`, la fonction devrait renvoyer `[0, 10, 6]`

M. Patel
2024-2025
Classes de B_2, Rennes