Informatique en CPGE (2017-2018)
Exercices : polynômes

Un polynôme est défini par la liste de ses coefficients dans l'ordre croissant.
Par exemple, le polynôme $-X + 4X^3 + 2X^5$ est représenté par [0,-1,0,4,0,2].

Opérations de base

1. Ecrire une fonction **print_poly(p)** qui ne renvoie rien mais affiche par exemple le polynôme **p** ci-dessus sous la forme :

   `-X + 4 X^3 + 2 X^5`
2. Ecire une fonction **oppose(p)** qui renvoie l'opposé de **p**.
3. Ecire une fonction **prod_poly_reel(p, r)** qui renvoie le produit du polynôme **p** par le réel **r**.
4. Ecire une fonction **somme_poly(p, q)** qui renvoie la somme des polynômes **p** et **q**.
5. Ecire une fonction **produit_poly(p, q)** qui renvoie le produit des polynômes **p** et **q**.
6. Ecire une fonction **puissance_poly(p, n)** qui renvoie la puissance n-ème du polynôme **p**. (n est un entier)
7. Ecire une fonction **derive_poly(p)** qui renvoie le polynôme dérivé de **p**.
8. Ecire une fonction **primitive_poly(p)** qui renvoie un polynôme primitive de **p**.
9. Ecire une fonction **valeur_poly(p, x)** qui renvoie la valeur du polynôme **p** en x. (x est un réel)

Polynômes de Legendre

On définit un produit scalaire par : $< p, q >= \displaystyle\int_{-1}^{1} p(x)q(x)\mathrm{d}x$.

1. Construction de la base canonique : définir une variable **base** de type **list** contenant la liste des polynômes $1, X, X^2, \dots, X^{10}$.
2. Ecire une fonction **scalaire(p,q)** qui renvoie le produit scalaire des polynômes **p** et **q**.
3. Ecire une fonction **norme(p)** qui renvoie la norme du polynôme **p**.
4. Polynômes de Legendre normalisés : définir une variable **legendre** de type **list** contenant la liste des polynômes de Legendre normalisés (degré de 0 à 10). On utilisera pour cela le procédé d'orthonormalisation de Gram-Schmidt sur la base canonique.

   Si on souhaite obtenir les polynômes de Legendre on multipliera chaque polynôme de degré i par $\sqrt{\dfrac{2}{2i+1}}$.
5. Représenter sur un même graphique les polynômes de Legendre normalisés jusqu'au degré 6, sur l'intervalle $[-1; 1]$.
6. Tester le code suivant permettant de représenter les approximations de la fonction sinus par une combinaison linéaire des polynômes de Legendre. (degré ≤ 3, puis degré ≤ 5 et degré ≤ 7).

```
import numpy as np
import scipy.integrate as integ
def f(x):
    return np.sin(2.0*np.pi*x)
def approx(k):
    r=[0]
    for i in range(k+1):
        s=integ.quad(lambda x:f(x)*valeur_poly(legendre[i],x),-1,1)[0]
```

```
            q=prod_poly_reel(legendre[i],s)
            r=somme_poly(r,q)
        return r
    xliste=[i/100-1 for i in range(201)]
    yliste=[f(x) for x in xliste]
    plt.plot(xliste,yliste)
    yliste=[valeur_poly(approx(3),x) for x in xliste]
    plt.plot(xliste,yliste)
    yliste=[valeur_poly(approx(5),x) for x in xliste]
    plt.plot(xliste,yliste)
    yliste=[valeur_poly(approx(7),x) for x in xliste]
    plt.plot(xliste,yliste)
    plt.show()
```

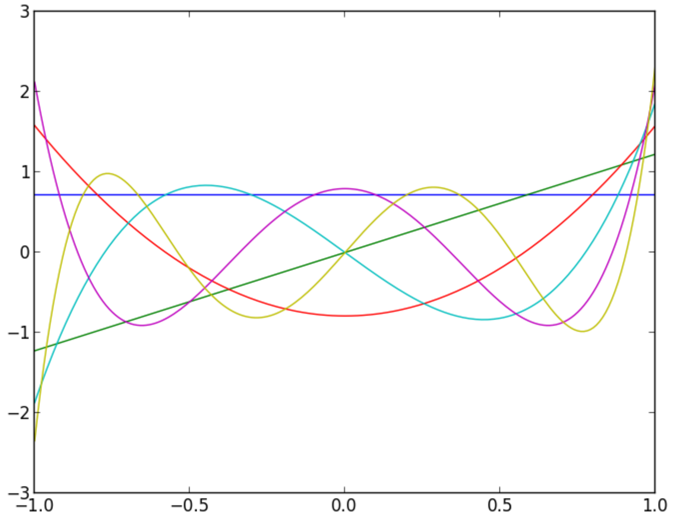

FIGURE 1 – Polynômes de Legendre normalisés

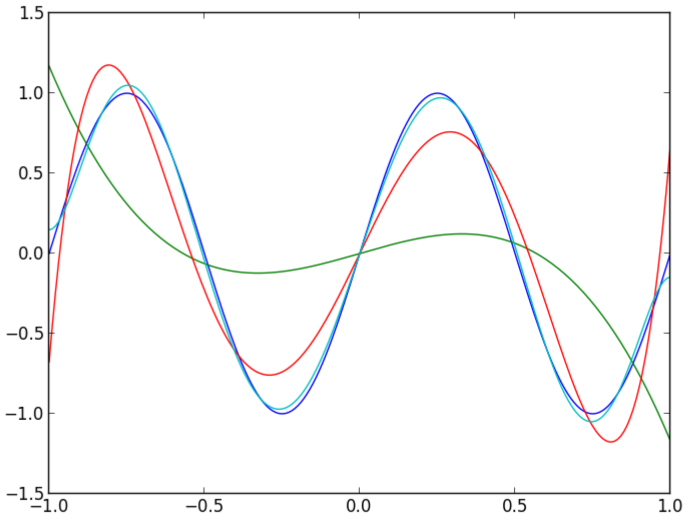

FIGURE 2 – Approximations de la fonction sinus