PBS 9 - À RENDRE AU PLUS TARD MARDI 26/03

EXERCICE 1 — UNE BIJECTION ENTRE $\mathbb{K}_n[X]$ ET $\mathbb{K}^{n+1}$

Soit $n \in \mathbb{N}$, et soit $\alpha_0, \ldots, \alpha_n$ (n+1) scalaires deux à deux distincts.

Soit l'application $\varphi : \mathbb{K}_n[X] \longrightarrow \mathbb{K}^{n+1}$

$$P \longmapsto (P(\alpha_0), P(\alpha_1), \ldots, P(\alpha_n))$$

1/ Soit k un entier de $[\![\, 0, n \,]\!]$. Montrer qu'il existe un unique polynôme L_k tel que :

$$L_k \in \mathbb{K}_n[X] \qquad \text{et} \qquad \forall\, j \in [\![\, 0, n \,]\!],\ L_k(\alpha_j) = \delta_{kj}$$

2/ Montrer que φ est bijective.

EXERCICE 2 — UNE AUTRE BIJECTION ENTRE $\mathbb{K}_n[X]$ ET $\mathbb{K}^{n+1}$

Soit $n \in \mathbb{N}$.

Soit l'application $\varphi : \mathbb{K}_n[X] \longrightarrow \mathbb{K}^{n+1}$

$$P \longmapsto \big(P(0), P'(0), \ldots, P^{(n)}(0)\big)$$

1/ Montrer que φ est injective.

2/ **Surjectivité de φ.** Soit $(y_0, y_1, \ldots, y_n) \in \mathbb{K}^{n+1}$.

Montrer qu'il existe un polynôme $P = \sum_{k=0}^{n} a_k X^k \in \mathbb{K}_n[X]$ tel que : $\varphi(P) = (y_0, y_1, \ldots, y_n)$. Exprimer les coefficients a_k en fonction des y_k.

3/ Déduire de ce qui précède que φ est bijective.

EXERCICE 3 — ETUDE D'UNE ÉQUATION DIFFÉRENTIELLE

On considère l'équation différentielle (E) : $(1-x^2)\,y''(x) - 3x\,y'(x) - y(x) = 0$

Dans cet exercice, on ne cherche pas à résoudre cette équation, mais à caractériser le développement limité en 0 d'une solution de (E).

Tout au long de cette partie, f désigne une fonction de $\mathscr{C}^2(I, \mathbb{R})$ solution de (E), avec $I =]-1, 1[$.

1/ Montrer que f est de classe $\mathscr{C}^\infty$ sur $I =]-1, 1[$.

2/ Etablir que :

$$\forall\, n \in \mathbb{N},\ \forall\, x \in I, \quad \big(1-x^2\big)\, f^{(n+2)}(x) - (2n+3)\, x f^{(n+1)}(x) - (n+1)^2\, f^{(n)}(x) = 0$$

3/ Pour tout $n \in \mathbb{N}$, on pose $a_n = f^{(n)}(0)$. Etablir une relation entre a_{n+2} et a_n.

4/ Soit p un entier naturel quelconque. Exprimer a_{2p+1} et a_{2p} en fonction de a_1 et a_0 respectivement et à l'aide de factorielles.

5/ En déduire le développement limité à l'ordre $(2n+1)$ en 0 de la fonction f (avec n entier naturel quelconque).

Exercice 4 — (Polynômes de Lagrange et Python).

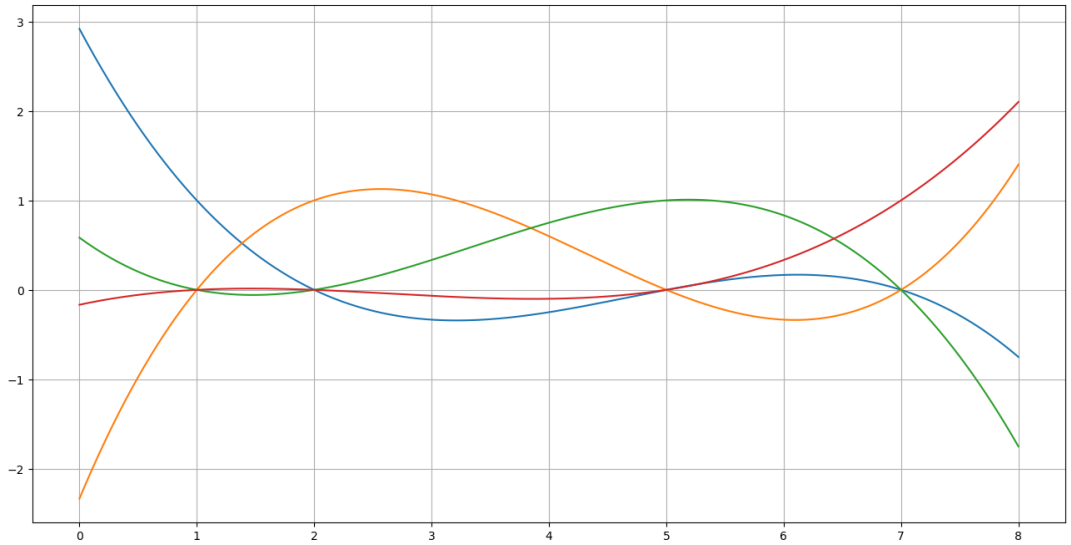
Le premier objectif de cet énoncé est de tracer les courbes représentatives des (fonctions associées aux) polynômes interpolateurs de Lagrange. Pour le principe, on rappelle la définition de ces polynômes :

- ➢ On considère $(n+1)$ réels deux à deux distincts : α_0, α_1,..., α_n
- ➢ On définit $(n+1)$ polynômes L_0, L_1,..., L_n de degré n tels que : $\forall\,(i,j) \in [\![\, 0, n \,]\!]^2, \quad L_i(\alpha_j) = \delta_{ij}$

  Une formule générale (et à connaître) permet de construire ces polynômes :

$$\forall\, i \in [\![\, 0, n \,]\!], \quad L_i = \prod_{k=0,\ k\neq i}^{n} \frac{X - \alpha_k}{\alpha_i - \alpha_k}$$

Illustration. Le graphe ci-dessous contient les courbes représentatives des polynômes interpolateurs de Lagrange L_0 (bleu), L_1 (orange), L_2 (vert) et L_3 (rouge) associés aux réels $\alpha_0 = 1$, $\alpha_1 = 2$, $\alpha_3 = 5$ et $\alpha_4 = 7$.



1/ Construction des polynômes de Lagrange. Ecrire une fonction `LAG(liste)`, qui reçoit comme paramètre une liste de flottants, et qui retourne la famille des polynômes interpolateurs de Lagrange associés à cette liste.

2/ Construction du graphique. Ecrire une fonction `RepresentLAG(liste)`, qui reçoit comme paramètre une liste de flottants, et qui fait apparaître sur un même graphique les courbes représentatives des polynômes interpolateurs de Lagrange associés à cette liste.

3/ Interpolation polynomiale appliquée. Déterminer l'unique polynôme P de degré 7 tel que :

$$P(1) = 1\,; \quad P(2) = 2\,; \quad P(3) = 3\,; \quad P(4) = 4\,; \quad P(5) = 8\,; \quad P(6) = 7\,; \quad P(7) = 6\,; \quad P(8) = 5$$

On précisera explicitement les coefficients de P, en donnant leurs valeurs approchées avec quatre chiffres significatifs.

Construire la courbe représentative de P.

Indication - Définition de polynômes avec numpy

On commence par importer la bibliothèque numpy via l'instruction classique : `import numpy as np`.

Ceci fait, on dispose d'une fonction (poly1d) qui permet de définir un polynôme, soit à l'aide de ses coefficients, soit à l'aide de ses racines. Explicitement :

le polynôme $aX^2 + bX + c$ peut être défini par l'instruction `np.poly1d([a,b,c])`.

le polynôme $(X - a)(X - b)(X - c)$ peut être défini par l'instruction `np.poly1d([a,b,c],True)`.

Dans les deux cas, l'objet obtenu est une fonction, que l'on peut ensuite évaluer numériquement.

Illustration :

```
>>>import numpy as np;

>>>P =np.poly1d([1,2,3]); # définit le polynôme P comme P = X^2 + 2X +3

>>>P(1); # Calcul de P(1)
6

>>>Q =np.poly1d([1,2,3], True); # définit le polynôme Q comme Q = (X−1)(X−2)(X−3)

>>>Q(1); # Calcul de Q(1)
0.0
```