# Devoir Surveillé
# 13 novembre 2023

On répondra aux questions en utilisant le language Python et en veillant à ce que l'indentation soit lisible.
Pour chaque question on pourra utiliser les procédures introduites aux questions précédentes, même si on n'y a pas répondu.
Les deux exercices sont indépendants, sauf la dernière question de l'exercice 2. On peut donc passer à l'exercice 2 sans avoir terminé l'exercice 1.
Tout document et toute communication sont interdites.

**Exercice 1.** On représente un polynôme $P(x) = \sum_{k=0}^{n} a_k x^k$ par la liste `L = [`$a_0, \ldots, a_n$`]` de ses coefficients.

a. Écrire une procédure `Éval(L,x)` qui calcule la valeur en `x` d'un polynôme donné par la liste `L` de ses coefficients. On utilisera de préférence la méthode de Horner pour ce calcul.

b. Écrire une procédure `Zéro(L,a,b)` qui cherche et retourne une racine dans l'intervalle $[a, b]$ pour le polynôme donné par la liste `L` de ses coefficients. On procèdera par dichotomie jusqu'à atteindre la précision `1e-10`.

c. Écrire une procédure `Dérivée(L)` qui calcule la dérivée d'un polynôme, au niveau des listes de coefficients. L'argument `L` aussi bien que la valeur retournée par la procédure sont donc des listes de coefficients.

On définit la suite des *polynômes de Legendre* $P_n$ par récurrence en posant $P_0 = 1$, $P_1 = X$ et

$$nP_n(X) = (2n-1)XP_{n-1}(X) - (n-1)P_{n-2}(X).$$

On admet que le polynôme $P_n$ est de degré $n$ et admet $n$ racines deux-à-deux distinctes dans l'intervalle $[-1, 1]$.

d. Écrire une procédure `Récurrence(Q,R,n)` qui retourne la liste des coefficients de $P_n$ si on lui passe en argument l'indice `n` et les listes `Q`, `R` des coefficients de $P_{n-1}$ et $P_{n-2}$.

e. On suppose qu'on connaît les racines de $P_n$.
Écrire une procédure `Critiques(L,Z)` qui retourne la liste des racines de $P_n'$ si on lui passe en argument la liste `L` des coefficients de $P_n$ et la liste `Z` des racines de $P_n$.

f. Écrire une procédure `Norme(L,Z)` qui retourne la norme $\|P\|_\infty = \sup_{x\in[-1,1]} |P(x)|$ du polynôme $P$ et un point `x` où cette norme est atteinte, si on lui passe en argument la liste `L` des coefficients de $P$ et la liste `Z` des racines de $P'$. On évitera de procéder à un balayage « naïf » de l'intervalle.

**Exercice 2.**

a. La procédure `Mystère` suivante admet deux arguments : une fonction Python `f`, qui représente une fonction réelle de la variable réelle, et un entier `m`. Que fait cette procédure ? On précisera notamment la grandeur mathématique dont `Mystère` calcule une valeur approchée.

```
def Mystère(f,m):
    R = 0
    h = 1 / m
    for i in range(-m,m):
        R = R + h * f(i*h)
    return R
```

b. Expliquer sur un schéma le principe de la méthode des trapèzes pour le calcul approché de $\int_a^b f(t)\mathrm{d}t$.
Écrire la formule pour l'aire d'un des trapèzes (avec des notations à préciser sur le schéma), puis donner la formule pour la valeur approchée de l'intégrale obtenue avec $m$ trapèzes.

c. Écrire une procédure `Trapèzes(f,m)` qui calcule une valeur approchée de la même quantitée que `Mystère(f,m)`, mais en utilisant la méthode des trapèzes.

On reprend la suite de polynômes $P_n$ de l'exercice précédent. On souhaite vérifier que ces polynômes sont deux-à-deux orthogonaux pour le produit scalaire

$$(f \mid g) = \int_{-1}^{1} f(t)g(t)\mathrm{d}t.$$

d. Écrire un programme qui utilise les procédures de cet exercice et de l'exercice précédent pour vérifier l'orthogonalité de $P_n$ et $P_m$ pour $0 \leq m < n \leq 6$. On calculera les intégrales avec $10^5$ trapèzes.