PRÉPARATION À L'AGRÉGATION
TP ALGORITHMIQUE EN SAGE

CHRISTOPHE RITZENTHALER

1. Évaluation, évaluation multi-points, interpolation

Exercice 1. Dans la méthode de pré-processing pour l'évaluation d'un polynôme de degré 4

$$\begin{aligned} a_4x^4 &+ a_3x^3 + a_2x^2 + a_1x + a_0 \\ &= a_4((x(x+\alpha_0)+\alpha_1)(x(x+\alpha_0)+x+\alpha_2)+\alpha_3) \\ &= a_4x^4 + a_4(2\alpha_0+1)x^3 + a_4(\alpha_1+\alpha_2+\alpha_0(\alpha_0+1))x^2 \\ &+ a_4((\alpha_1+\alpha_2)\alpha_0+\alpha_1)x + a_4(\alpha_1\alpha_2+\alpha_3). \end{aligned} \tag{1}$$

trouver les α_i en fonction des coefficients a_i.

Exercice 2. Implémenter l'interpolation de Lagrange. On ne cherchera pas à optimiser le calcul des A_i.

Exercice 3. Implémenter l'évaluation multi-points rapide ainsi que l'interpolation rapide. On pourra créer la structure arborescente de la manière suivante

```
class Tree(object):
    def __init__(self):
        self.left = None
        self.right = None
        self.data = None
```

Puis pour créer un arbre binaire A de racine un polynôme P et de sous-arbre gauche (resp. droit) A_0 (resp. A_1), on écrit

```
A = Tree()
A.data = P
A.left = A0
A.right = A1
```

Exercice 4. On considère la fonction $f(x) = \frac{1}{1+25x^2}$. On cherche à interpoler cette fonction aux points $a_0 = -1, a_1 = -1+h, \ldots, a_{i+1} = a_i + h, \ldots, 1$ avec $h = 2/n$ et $n = 2^s - 1$. Dessiner pour plusieurs valeurs de s le polynôme de Lagrange obtenu et comparer le dessin à celui de f. Qu'observe-t-on ?