Interpolation Polynomiale de Lagrange

Implementation en langage C

https://github.com/ryurina/interpolationDeLagrange

22 mai 2025

Table des matières

1 Formule de Lagrange **2**

2 Cas Particulier : Degre 2 **2**
2.1 Polynomes de Base 2
2.2 Polynome Interpolateur 2

3 Algorithme et Pseudo-code **3**
3.1 Pseudo-code de l'Interpolation de Lagrange 3

4 Implementation en C **4**

1 Formule de Lagrange

La methode de Lagrange exprime le polynome interpolateur sous la forme :

$$P(x) = \sum_{i=0}^{n} y_i L_i(x) \tag{1}$$

ou $L_i(x)$ sont les polynomes de base de Lagrange definis par :

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{2}$$

2 Cas Particulier : Degre 2

Pour l'interpolation de degre 2, nous utilisons trois points (x_0, y_0), (x_1, y_1), (x_2, y_2).

2.1 Polynomes de Base

Les trois polynomes de base sont :

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \tag{3}$$

$$L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \tag{4}$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \tag{5}$$

2.2 Polynome Interpolateur

Le polynome interpolateur de degre 2 s'ecrit :

$$P(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) \tag{6}$$

3 Algorithme et Pseudo-code

3.1 Pseudo-code de l'Interpolation de Lagrange

Voici le pseudo-code general pour l'interpolation de Lagrange de degre 2 :

Listing 1 – Pseudo-code de l'interpolation de Lagrange

```
ALGORITHME InterpolationLagrange
ENTREE:
    points[3] : tableau de 3 points (xi, yi)
    x : valeur a interpoler
SORTIE:
    P(x) : valeur du polynome interpolateur en x

DEBUT
    resultat <- 0

    POUR i DE 0 A 2 FAIRE
        Li <- 1  // Polynome de base Li(x)

        POUR j DE 0 A 2 FAIRE
            SI j != i ALORS
                Li <- Li * (x - points[j].x) / (points[i].x -
                   points[j].x)
            FIN SI
        FIN POUR

        resultat <- resultat + points[i].y * Li
    FIN POUR

    RETOURNER resultat
FIN
```

4 Implementation en C

Listing 2 – Implementation complete de l'interpolation de Lagrange en C

```c
#include <stdio.h>
#include <stdlib.h>

// Structure pour representer un point (x, f(x)=y)
typedef struct {
    double x;
    double y;
} Point;

// Fonction pour afficher le polynome interpolateur
void afficher_polynome(Point points[3]) {
    printf("Points d'interpolation:\n");
    for (int i = 0; i < 3; i++) {
        printf("P%d: (%.2f, %.2f)\n", i+1, points[i].x, points[i
           ].y);
    }

    printf("\nPolynome de Lagrange de degre 2:\n");
    printf("P(x) = %.2f * L0(x) + %.2f * L1(x) + %.2f * L2(x)\n",
           points[0].y, points[1].y, points[2].y);

    printf("\nOu:\n");
    printf("L0(x) = (x - %.2f)(x - %.2f) / ((%.2f - %.2f)(%.2f -
       %.2f))\n",
           points[1].x, points[2].x, points[0].x, points[1].x,
           points[0].x, points[2].x);
    printf("L1(x) = (x - %.2f)(x - %.2f) / ((%.2f - %.2f)(%.2f -
       %.2f))\n",
           points[0].x, points[2].x, points[1].x, points[0].x,
           points[1].x, points[2].x);
    printf("L2(x) = (x - %.2f)(x - %.2f) / ((%.2f - %.2f)(%.2f -
       %.2f))\n",
           points[0].x, points[1].x, points[2].x, points[0].x,
           points[2].x, points[1].x);
}

// Fonction pour saisir les points
void saisir_points(Point points[3]) {
    printf("Saisie des 3 points pour l'interpolation de Lagrange
       :\n");
    for (int i = 0; i < 3; i++) {
        printf("Point %d - x_%d: ", i, i);
        scanf("%lf", &points[i].x);
        printf("Point %d - f(x_%d) = y_%d: ", i, i, i);
        scanf("%lf", &points[i].y);
    }
}

```

```
44 int main() {
45     Point points[3];
46 
47     printf("INTERPOLATION DE LAGRANGE - DEGRE 2\n\n");
48 
49     saisir_points(points);
50     printf("\n");
51 
52     afficher_polynome(points);
53 
54     return 0;
55 }
```