

Glody KUTUMBAKANA

Compte Rendu Splines

Année 2024 -2025

Introduction

L'interpolation par splines cubique est une technique permettant d'approcher une fonction réelle d'une variable en la divisant en segments de polynôme de degré trois.

L'objectif de ce TP est d'implémenter une classe ***Spline*** qui permet :

- Calcule et stocke les dérivées seconde aux point de support.
- Fournit une méthode pour évaluer la fonction interpolée en une valeur donnée
- Gère les erreurs en cas d'entrée hors de l'intervalle de définition

Code :

Class Spline

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;

import org.knowm.xchart.*;
import org.knowm.xchart.style.lines.SeriesLines;

public class Splines {

    private double[] b;
    private double[] c;
    private double[] d;
    private double[] x;
    private double[] y;
    private int n;
    private Vecteur g;

    public Splines(double[] x, double[] y) throws Exception{
        if(x.length != y.length) throw new Exception("Les tableaux x et y doivent avoir la même taille");
        if(x.length < 4) throw new Exception("Il faut au moins 4 points pour faire une spline");
        this.n = x.length;
        this.x = x;
        this.y = y;
        b = new double[n - 2];
```

Glody KUTUMBAKANA

```
c = new double[n - 2];
d = new double[n - 2];
b[0] = 2 * (x[2] - x[0]);
c[0] = (x[2] - x[1]);
d[0] = 6 * (((y[2] - y[1]) / (x[2] - x[1])) - ((y[1] - y[0]) / (x[1] -
x[0])));
g = deriveSecond();
}

private Vecteur deriveSecond() throws Exception{
    Vecteur res = new Vecteur(n);
    res.replacecoef(0, 0);
    res.replacecoef(n - 1, 0);

    Mat3Diag diag = new Mat3Diag(n - 2);
    Vecteur bx = new Vecteur(n - 2);
    double[] a = new double[n - 2];

    for(int i = 2; i < n - 2; i++){
        a[i - 1] = c[i - 2];
        b[i - 1] = 2 * (x[i + 1] - x[i - 1]);
        c[i - 1] = x[i + 1] - x[i];
        d[i - 1] = 6 * ((y[i + 1] - y[i]) / c[i - 1] - (y[i] - y[i - 1]) / a[i -
1]));

        if (b[i - 1] == 0) throw new Exception("Matrice tridiagonale singulière :
b[" + i + "] est nul");
    }

    a[n - 3] = c[n - 4];
    b[n - 3] = 2 * (x[n - 1] - x[n - 3]);
    d[n - 3] = 6 * ((y[n - 1] - y[n - 2]) / (x[n - 1] - x[n - 2]) - (y[n - 2] -
y[n - 3]) / a[n - 3]);

    for(int i = 0; i < n - 2; i++){
        diag.replacecoef(0, i, a[i]);
        diag.replacecoef(1, i, b[i]);
        diag.replacecoef(2, i, c[i]);
        bx.replacecoef(i, d[i]);
    }

    Thomas thomas = new Thomas(diag, bx);
    Vecteur par = thomas.resolution();

    for(int i = 1; i <= n - 2; i++){
        res.replacecoef(i, par.getCoef(i - 1));
    }
    return res;
}

public double interpolation(double rech) throws Exception{
    double alpha , beta , gamma;
    double t1 = 0 , t2 = 0 , t3 = 0;
    int interateur = 0;
    while(interateur < n - 1 && x[interateur + 1] < rech){
        interateur++;
    }
    if(interateur == n - 1) throw new Exception("La valeur recherchée est en
dehors de l'intervalle");
    alpha = x[interateur + 1] - rech;
```

Glody KUTUMBAKANA

```
        beta = rech - x[interateur];
        gamma = x[interateur + 1] - x[interateur];
        t1 = (g.getCoef(interateur) / 6) * ((Math.pow(alpha, 3) / gamma) - alpha *
gamma);
        t2 = (g.getCoef(interateur + 1) / 6) * ((Math.pow(beta, 3) / gamma) - beta *
gamma);
        t3 = (y[interateur] * alpha + y[interateur + 1] * beta) / gamma;
        return t1 + t2 + t3;
    }

    public static void main(String[] args) throws Exception {
        try {
            double[] x = null;
            double[] y = null;
            Scanner scanner = new Scanner(System.in);
            System.out.print("Entrez le nom du fichier contenant les points de support
: ");
            String filename = scanner.nextLine();
            scanner.close();

            BufferedReader br = new BufferedReader(new FileReader(filename));
            String line = br.readLine();
            if (line != null) {
                int size = Integer.parseInt(line);
                x = new double[size];
                y = new double[size];
                line = br.readLine();
                String[] xParts = line.split(" ");
                for (int i = 0; i < size; i++) {
                    x[i] = Double.parseDouble(xParts[i]);
                }
                line = br.readLine();
                String[] yParts = line.split(" ");
                for (int i = 0; i < size; i++) {
                    y[i] = Double.parseDouble(yParts[i]);
                }
            }
            br.close();

            Splines splines = new Splines(x, y);
            XYChart chart = new
XYChartBuilder().width(800).height(600).title("Spline").xAxisTitle("X").yAxisTit
le("Y").build();
            XYSeries series = chart.addSeries("Spline", x, y);
            series.setLineStyle(SeriesLines.NONE);

            double[] interpolX = new double[110];
            double[] interpolY = new double[110];
            double minX = Arrays.stream(x).min().getAsDouble();
            double maxX = Arrays.stream(x).max().getAsDouble();
            for (int i = 0; i < 110; i++) {
                interpolX[i] = minX + i * (maxX - minX) / 109;
                interpolY[i] = splines.interpolation(interpolX[i]);
            }
            chart.addSeries("Interpolation", interpolX, interpolY);
            new SwingWrapper(chart).displayChart();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Résultat Obtenu

L'implémentation a été testée avec des fichiers (txt) de points de support contenant 4 à 100 valeurs

les résultats montrent que les courbes interpolées suivent les points de support.

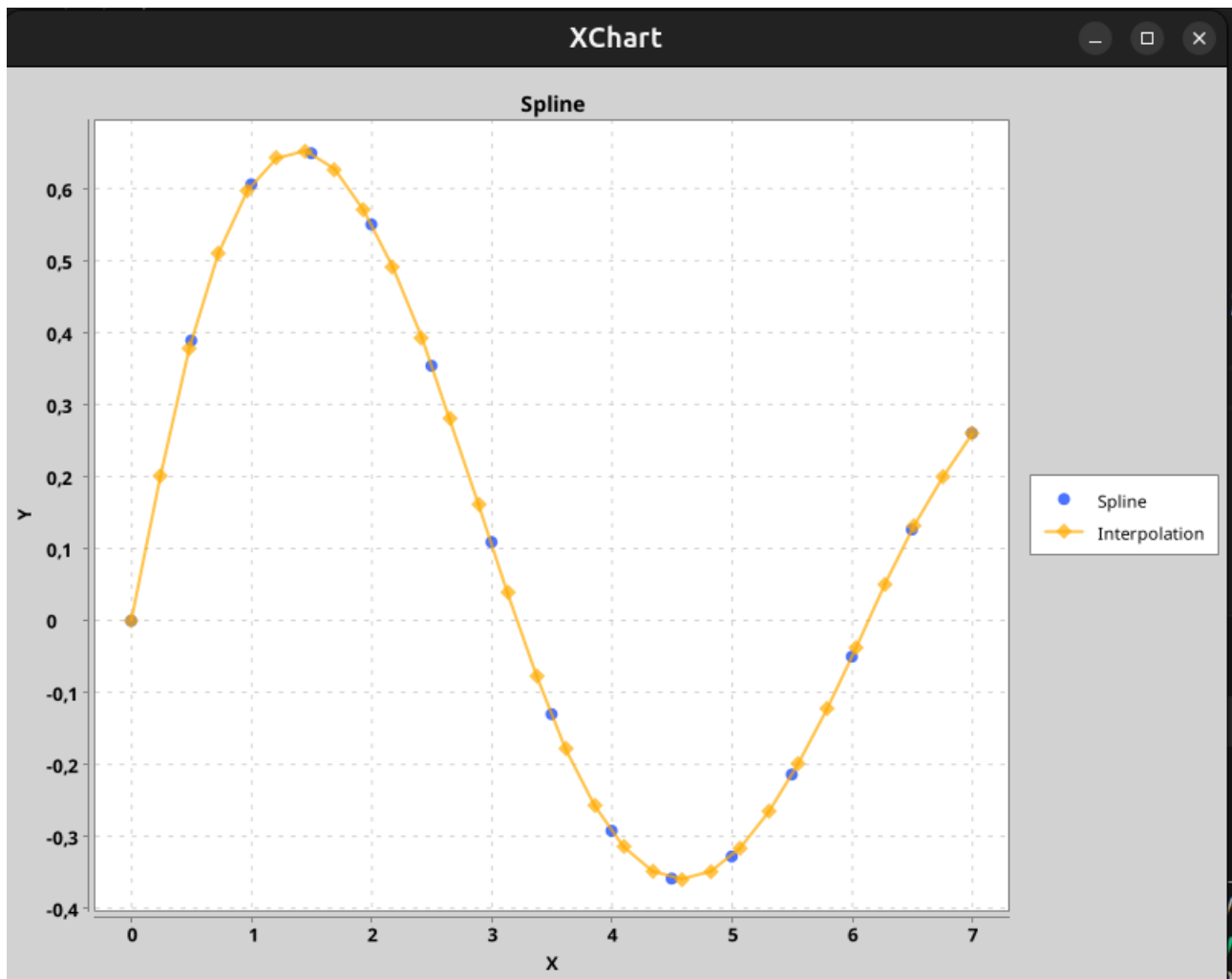
Pour chaque fichier obtiens :

Fichier Point.txt :

Le fichier est de taille 15 et contient :

- X : 0.0,0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5,7.0
- Y : 0.0,0.3894,0.6065,0.6496,0.5507,0.3543,0.1093,-0.1298,-0.2919,-0.3583,-0.3276,-0.2137,-0.0499,0.1266,0.2607

Ainsi on obtient :

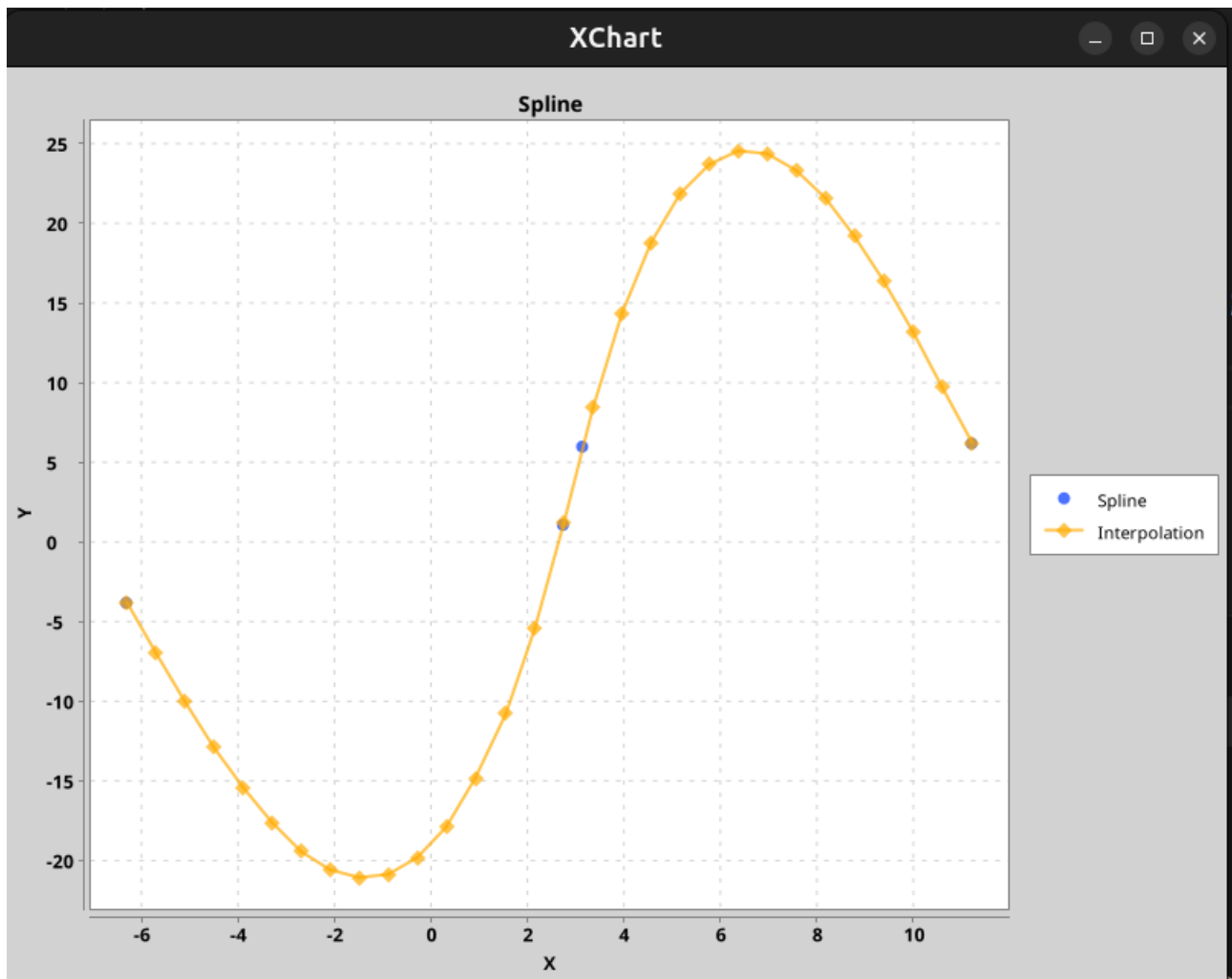


Fichier Point1.txt :

Le fichier est de taille 4 et contient :

- X : -6.3,2.75,3.15,11.22
- Y : 6,1.09,6.2,-3.8

Ainsi on obtient

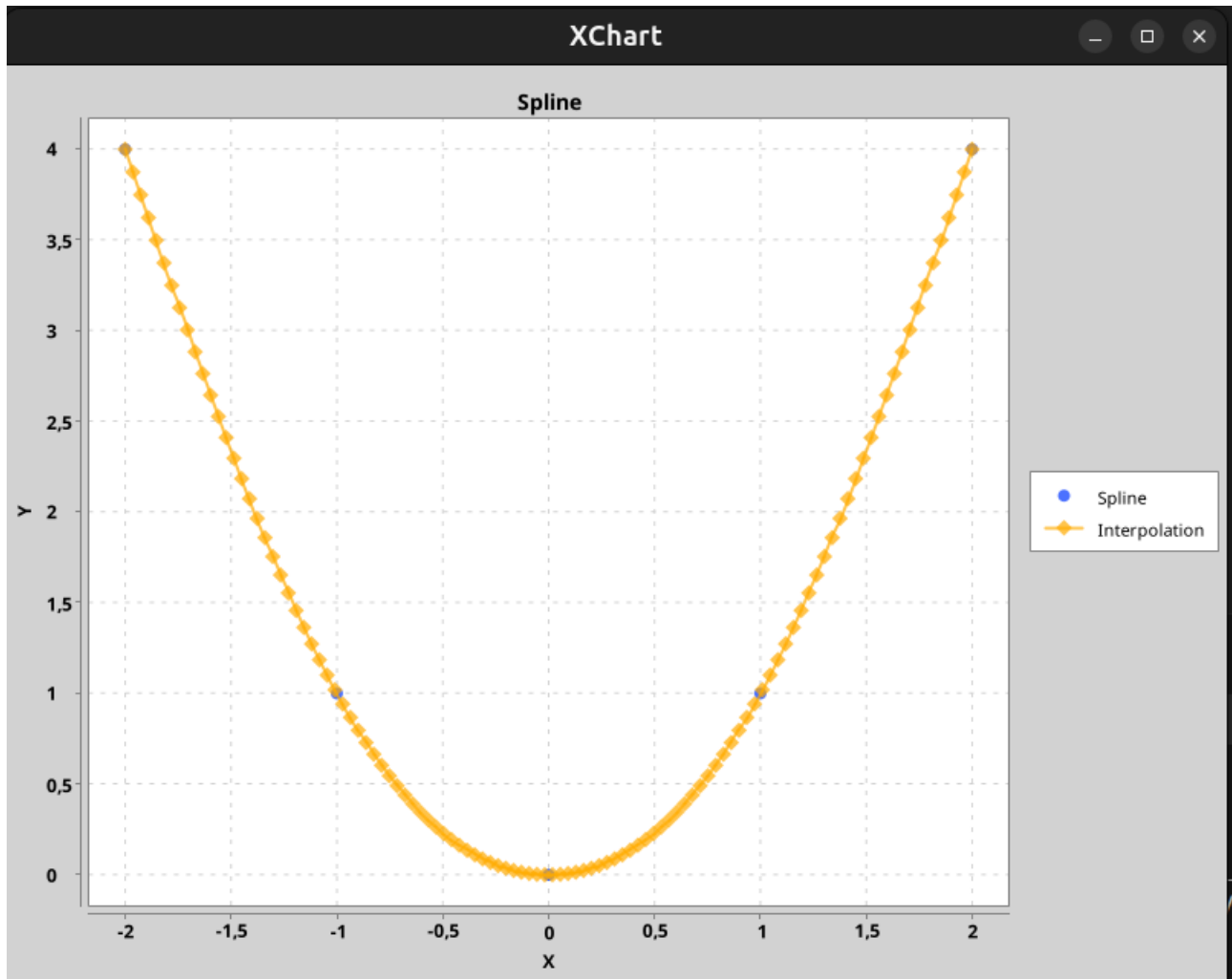


Fichier Point2.txt

Le fichier est de taille 5 et contient :

- X : -2.0,-1.0,0.0,1.0,2.0
- Y : 4.0,1.0,0.0,1.0,4.0

Ainsi on obtient :

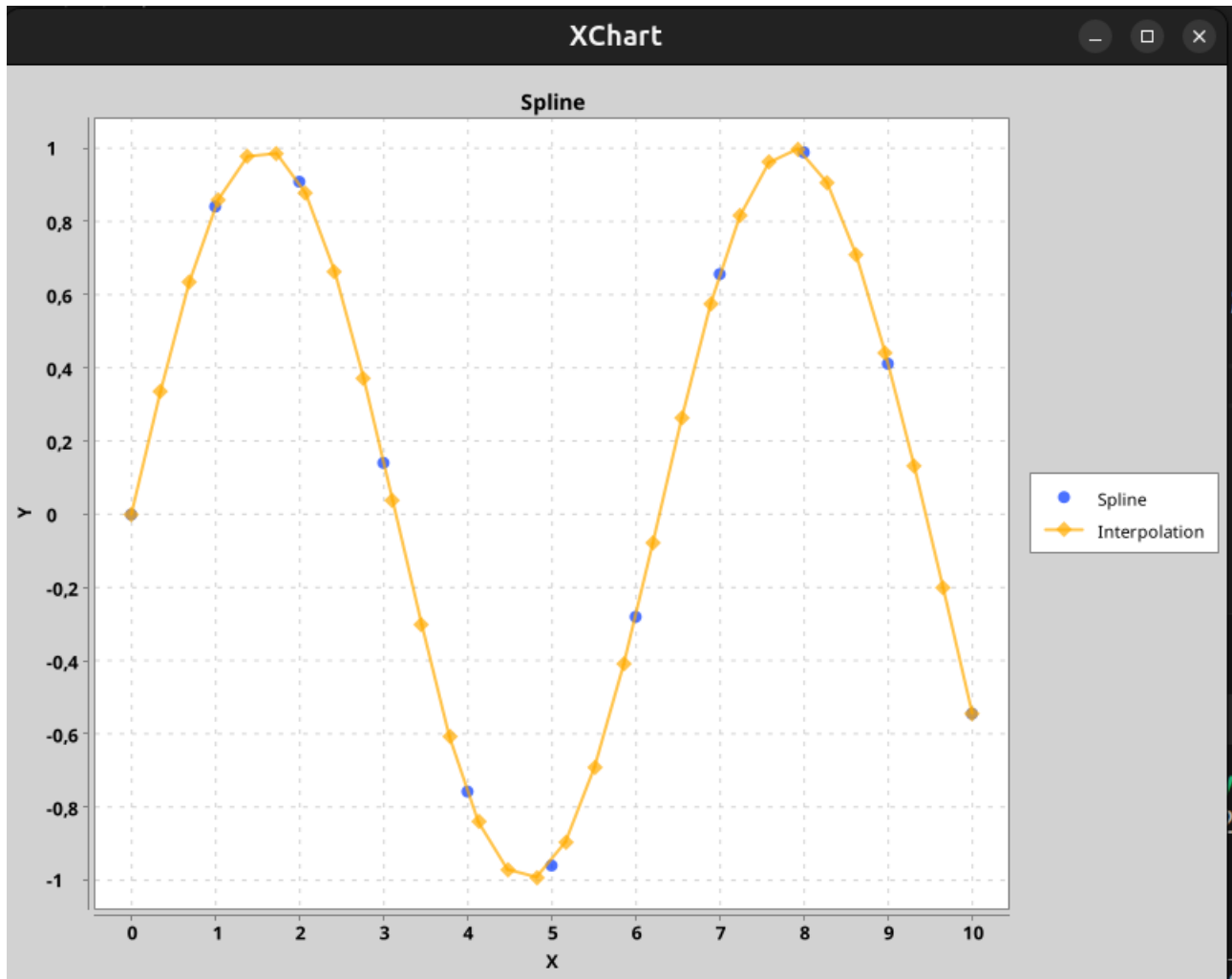


Fichier Point3.txt

Le fichier est de taille 11 et contient :

- X : 0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0
- Y : 0.0,0.8415,0.9093,0.1411,-0.7568,-0.9589,-0.2794,0.6569,0.9894,0.4121,-0.5440

Ainsi on obtient :

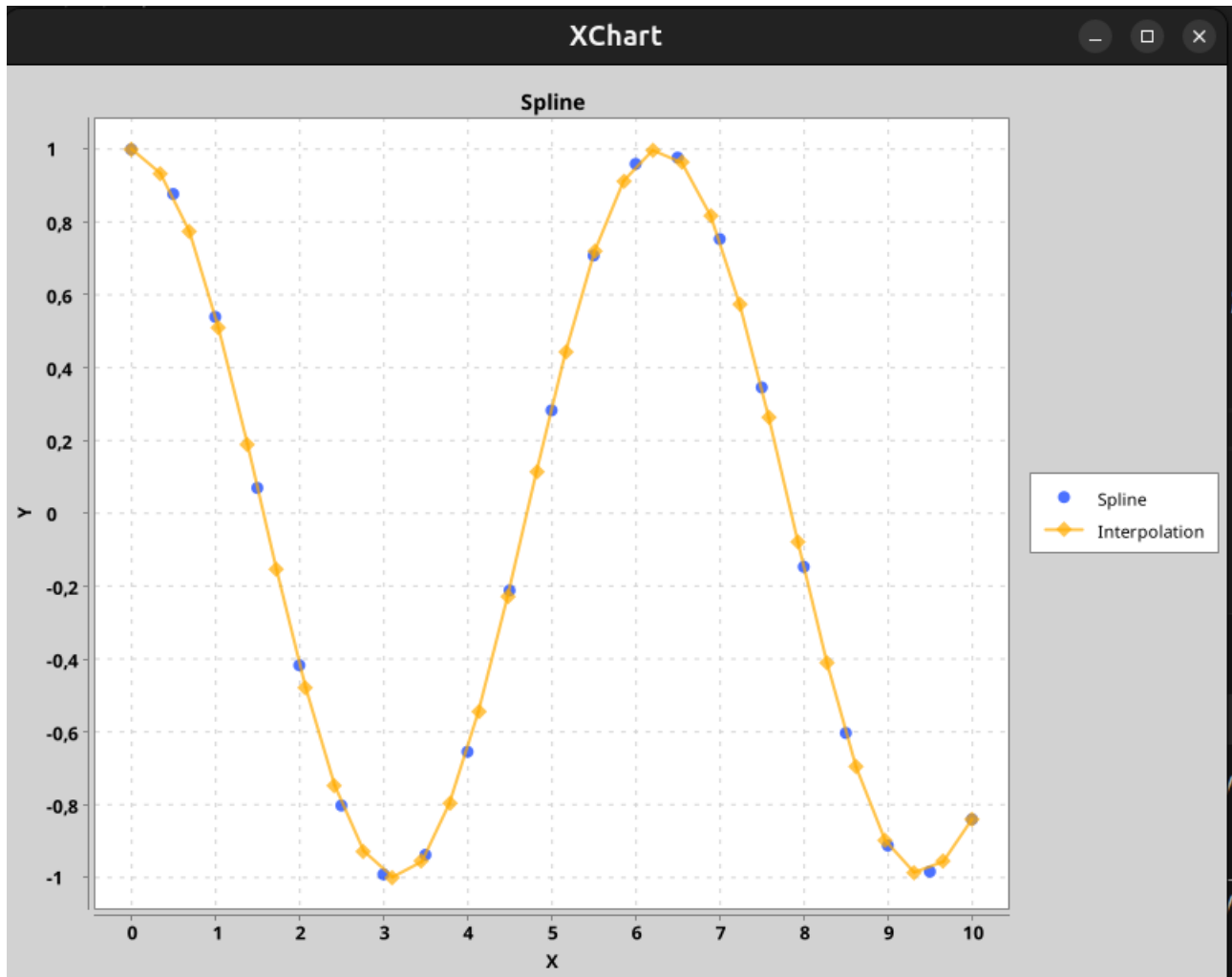


Fichier Point4.txt

Le fichier est de taille 21 et contient :

- X : 0.0,0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5,7.0,7.5,8.0,8.5,9.0,9.5,10.0
- Y : 1.0,0.8776,0.5403,0.0707,-0.4161,-0.8011,-0.9900,-0.9365,-0.6536,-0.2108,0.2837,0.7087,0.9602,0.9775,0.7539,0.3466,-0.1459,-0.6020,-0.9111,-0.9825,-0.8391

Ainsi on obtient :

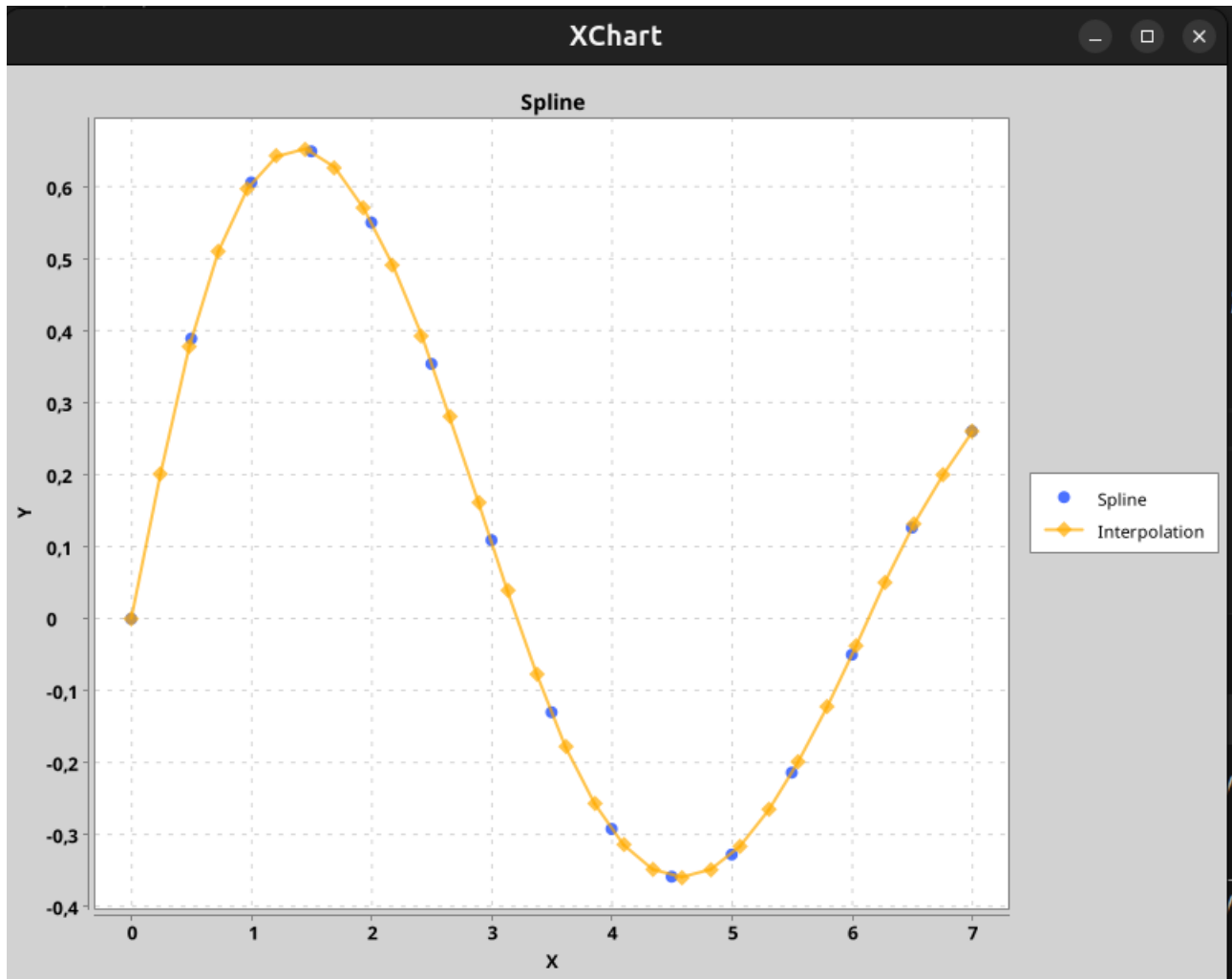


Fichier Point5.txt

Le fichier est de taille 15 et contient :

- X : 0.0,0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5,7.0
- Y : 0.0,0.3894,0.6065,0.6496,0.5507,0.3543,0.1093,-0.1298,-0.2919,-0.3583,-0.3276,-0.2137,-0.0499,0.1266,0.2607

Ainsi on obtient :



Fichier Point6.txt

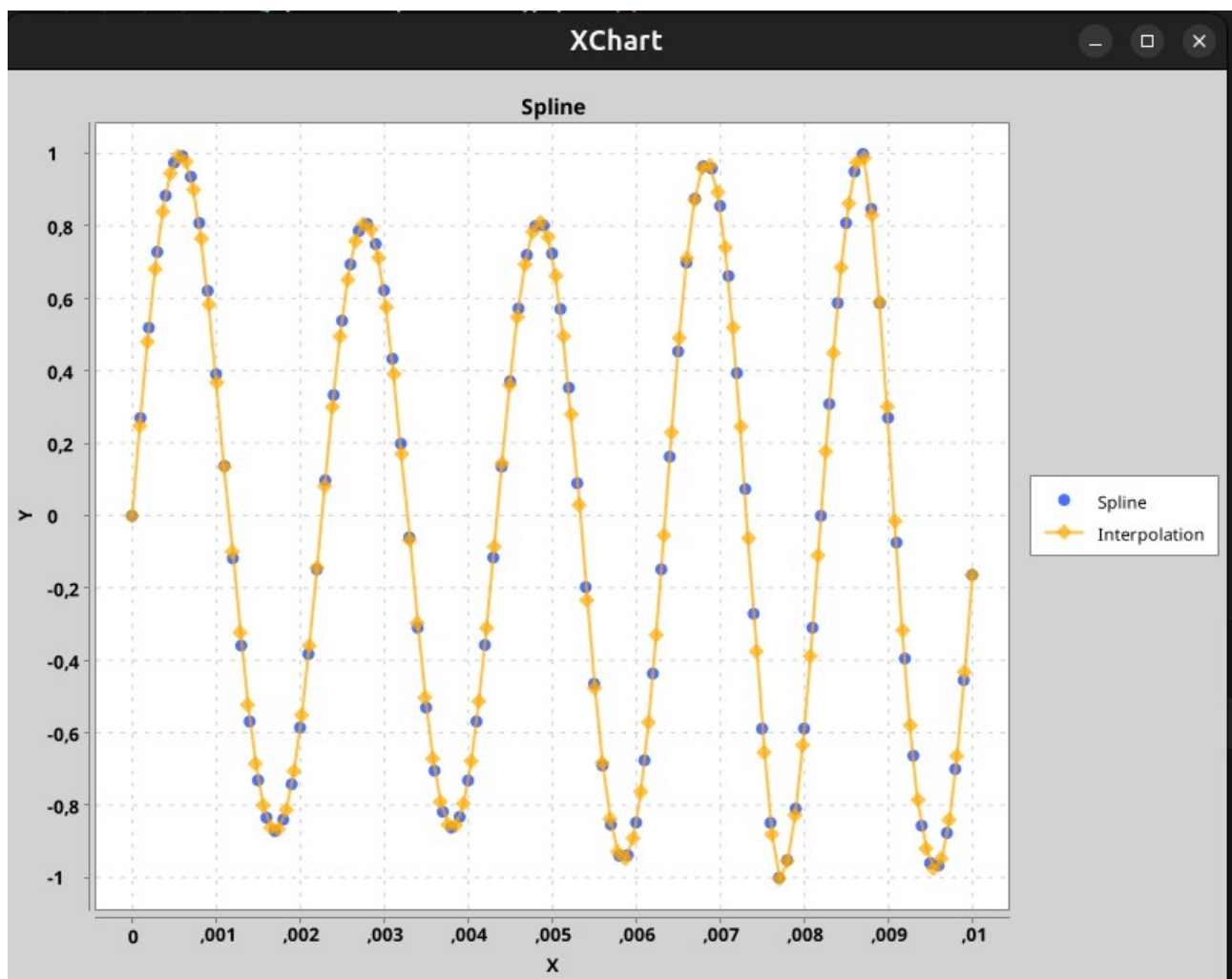
Le fichier est de taille 101 et contient :

- X : 0.0,0.0001,0.0002,0.0003,0.0004,0.0005,0.0006,0.0007,0.0008,0.0009,0.0010,0.0011,0.0012,0.0013,0.0014,0.0015,0.0016,0.0017,0.0018,0.0019,0.0020,0.0021,0.0022,0.0023,0.0024,0.0025,0.0026,0.0027,0.0028,0.0029,0.0030,0.0031,0.0032,0.0033,0.0034,0.0035,0.0036,0.0037,0.0038,0.0039,0.0040,0.0041,0.0042,0.0043,0.0044,0.0045,0.0046,0.0047,0.0048,0.0049,0.0050,0.0051,0.0052,0.0053,0.0054,0.0055,0.0056,0.0057,0.0058,0.0059,0.0060,0.0061,0.0062,0.0063,0.0064,0.0065,0.0066,0.0067,0.0068,0.0069,0.0070,0.0071,0.0072,0.0073,0.0074,0.0075,0.0076,0.0077,0.0078,0.0079,0.0080,0.0081,0.0082,0.0083,0.0084,0.0085,0.0086,0.0087,0.0088,0.0089,0.0090,0.0091,0.0092,0.0093,0.0094,0.0095,0.0096,0.0097,0.0098,0.0099,0.0100
- Y : 0.0,0.2706,0.5196,0.7290,0.8846,0.9755,0.9938,0.9368,0.8090,0.6216,0.391,0.1378,-0.1175,-0.3584,-0.5681,-0.7305,-0.8337,-0.8707,-0.8391,-0.7413,-0.5849,-0.3817,-0.1478,0.0980,0.3338,0.5390,0.6947,0.7867,0.8064,0.7507,0.6230,0.4339,0.1993,-0.0596,-0.3090,-0.5298,-0.7040,-0.8176,-0.8613,-0.8316,-0.7310,-0.5681,-0.3567,-0.1152,0.1358,0.3717,0.5728,0.7206,0.8002,0.8023,0.7245,0.5713,0.3540,0.0901,-0.1969,-0.4640,-0.6897,

Glody KUTUMBAKANA

-0.8532,-0.9389,-0.9375,-0.8474,-0.6755,-0.4357,-0.1478,0.1634,0.4540,0.6997,0.8761,
0.9659,0.9595,0.8559,0.6625,0.3944,0.0739,-0.2706,-0.5878,-0.8481,-0.9999,-0.9511,
-0.8090,-0.5878,-0.3090,0.0000,0.3090,0.5878,0.8090,0.9511,0.9999,0.8481,0.5878,0.2706,
-0.0739,-0.3944,-0.6625,-0.8559,-0.9595,-0.9659,-0.8761,-0.6997,-0.4540,-0.1634,0.1478,
0.4357,0.6755,0.8474,0.9375,0.9389,0.8532,0.6897,0.4640,0.1969,-0.0901,-0.3540,-0.5713,
-0.7245,-0.8023,-0.8002,-0.7206,-0.5728,-0.3717,-0.1358,0.1152,0.3567,0.5681,0.7310,
0.8316,0.8613,0.8176,0.7040,0.5298,0.3090,0.0596,-0.1993,-0.4339,-0.6230,-0.7507,
-0.8064,-0.7867,-0.6947,-0.5390,-0.3338,-0.0980,0.1478,0.3817,0.5849,0.7413,0.8391,
0.8707,0.8337,0.7305,0.5681,0.3584,0.1175,-0.1378,-0.3910,-0.6216,-0.8090,-0.9368,
-0.9938,-0.9755,-0.8846,-0.7290,-0.5196,-0.2706,0.0

Ainsi on obtient :



Fichier Point7.txt

Le fichier est de taille 101 et contient :

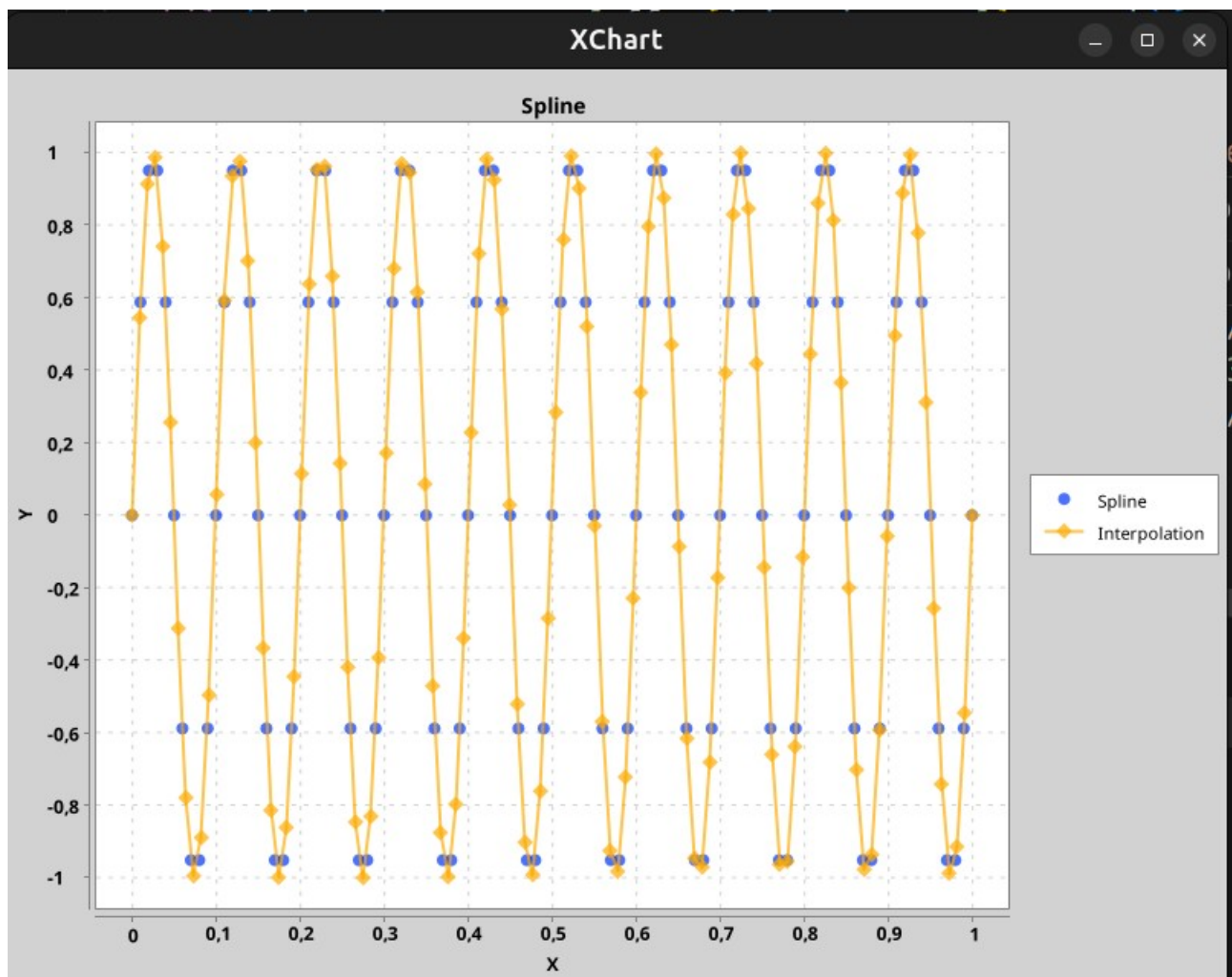
- X : 0.0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.10,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19,0.20,0.21,0.22,0.23,0.24,0.25,0.26,0.27,0.28,0.29,0.30,0.31,0.32,0.33,0.34,0.35,0.36,0.37,0.38,0.39,0.40,0.41,0.42,0.43,0.44,0.45,0.46,0.47,0.48,0.49,0.50,0.51,0.52,0.53,

Glody KUTUMBAKANA

0.54,0.55,0.56,0.57,0.58,0.59,0.60,0.61,0.62,0.63,0.64,0.65,0.66,0.67,0.68,0.69,0.70,0.71,
0.72,0.73,0.74,0.75,0.76,0.77,0.78,0.79,0.80,0.81,0.82,0.83,0.84,0.85,0.86,0.87,0.88,0.89,
0.90,0.91,0.92,0.93,0.94,0.95,0.96,0.97,0.98,0.99,1.00

- Y : 0.0,0.5878,0.9511,0.9511,0.5878,0.0,-0.5878,-0.9511,-0.9511,-0.5878,0.0,0.5878,0.9511,
0.9511,0.5878,0.0, -0.5878,-0.9511,-0.9511,-0.5878,0.0,0.5878,0.9511,0.9511,0.5878,0.0,
-0.5878,-0.9511,-0.9511,-0.5878,0.0,0.5878,0.9511,0.951,0.5878,0.0,-0.5878,-0.9511,
-0.9511,-0.5878,0.0,0.5878,0.9511,0.9511,0.5878,0.0,-0.5878,-0.9511,-0.9511,-0.5878,0.0
0.5878,0.9511,0.9511,0.5878,0.0,-0.5878,-0.9511,-0.9511,-0.5878,0.0,0.5878,0.9511,0.9511,
0.5878,0.0,-0.5878,-0.9511,-0.9511,-0.5878,0.0,0.5878,0.9511,0.9511,0.5878,0.0,-0.5878,
-0.9511,-0.9511,-0.5878,0.0,0.5878,0.9511,0.9511,0.5878,0.0,-0.5878,-0.9511,-0.9511,
-0.5878,0.0,0.5878,0.9511,0.9511,0.5878,0.0,-0.5878,-0.9511,-0.9511,-0.5878,0.0

Ainsi on obtient :



Conclusion

Cette classe nous permet de faire le calcul d'interpolation par splines cubiques ,cet approche utilisée garantit la continuité de la fonction interpolée et de sa dérivée première.