Lagrange's Interpolation Polynomial

Lucas V. Araujo <lucas.vieira.ar@disroot.org>

December 3, 2019

Part I
Explanation

Given N points $a_i(x_i, y_i)$, the Lagrange's Interpolation Polynomial is described as follows:

$$P(x) = L_0y_0 + L_1y_1 + L_2y_2 + ... + L_{N-1}y_{N-1}$$

where

$$L_0 = (\frac{x - x_1}{x_0 - x_1})(\frac{x - x_2}{x_0 - x_2})(\frac{x - x_3}{x_0 - x_3})...(\frac{x - x_{N-1}}{x_0 - x_{N-1}})$$

$$L_1 = (\frac{x - x_0}{x_1 - x_0})(\frac{x - x_2}{x_1 - x_2})(\frac{x - x_3}{x_1 - x_3})...(\frac{x - x_{N-1}}{x_1 - x_{N-1}})$$

$$L_2 = (\frac{x - x_0}{x_2 - x_0})(\frac{x - x_1}{x_2 - x_1})(\frac{x - x_3}{x_2 - x_3})...(\frac{x - x_{N-1}}{x_2 - x_{N-1}})$$

$$...$$

$$L_k = \prod_{i,j=0}^{N-1} \frac{x - x_j}{x_i - x_j} \mid i \neq j$$

As an (computational) algorithm it would be:

Algorithm 1 Lagrange's Interpolation Algorithm

```
let points = {(x_0, y_0), (x_1, y_1), (x_2, y_2), ..., (x_{N-1}, y_{N-1})};
let x = <future_value>;
let r = 0;
for (i = 0; i < N; i++)
{
    let l = 1;
    for (j = 0; j < N; j++)
    {
        if (i != j)
        {
            l = l * (x - points[j][0]) / (points[i][0] - points[j][0]);
        }
    }
    r = r + l * points[i][1];
}
```