



# Informe Laboratorio: Análisis Numérico

## Práctica No. 6

**Daniel Delgado**

**Código:** 2182066

**Grupo:** B2

*Escuela de Ingeniería de Sistemas e Informática  
Universidad Industrial de Santander*

4 de febrero de 2021

## 1. Introducción

La interpolación, específicamente la interpolación polinómica, es una práctica la cual permite el encontrar más puntos a partir de un conjunto de puntos dados. Esto puede ser por la mera aproximación de puntos que sigan la tendencia de la función o la generación de un polinomio que se ajuste a un set de puntos dados.

En el presente informe, tratará las interpolaciones polinómicas de Newton y LaGrange y el como estas operan y se diferencian entre ellas. De la misma manera, se buscará entender el como cada una de estas se comporta en un contexto computacional y las ventajas de cada uno de estos.

La comprensión de las diferentes maneras de realizar interpolaciones, al igual que el desarrollo de la algoritmia relacionada, son los principales temas a a tratar durante el desarrollo del presente informe, así como la resolución de los problemas propuestos a manera de pregunta orientadora durante el desarrollo del componente práctico del mismo.

## 2. Desarrollo

### 1. Preguntas propuestas

De manera inicial, la guía de trabajo plantea algunas preguntas base con el fin de establecer los conceptos básicos que serán trabajados a lo largo del desarrollo del presente informe.

#### a) ¿Qué es la interpolación?

En términos simples, la interpolación se define como la obtención de puntos de una función indeterminada a partir de algunos puntos conocidos. Esto se hace comúnmente para la búsqueda de tendencias dadas por algunos datos o con el fin de realizar una aproximación de alguna función hacia algo más simple.

#### b) ¿Cómo se calcula un polinomio del Taylor de grado N?

Los polinomios de Taylor de grado N, aunque existen varias maneras de calcularlos, dentro de un dominio determinado  $[a, b]$ , es posible calcularlos a partir de la siguiente sumatoria la cual es usada para aproximar una función:

$$f(x) \approx P(x) = \sum_{k=0}^N \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

Es gracias a esto que podemos realizar aproximaciones polinómicas de algunas funciones más complejas como lo vienen siendo las funciones sinusoidales o exponenciales.

c) ¿Cómo se calculan las interpolaciones de Newton y LaGrange?

Estas interpolaciones son calculadas de dos maneras bastante diferentes pero toma datos similares para realizar este proceso.

La interpolación de LaGrange se realiza en dos partes, la primera, es el cálculo de los productos de los puntos que serán usados para realizar la generación del polinomio:

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

Seguidamente, empleando los valores obtenidos para  $L_i$ , se calculará el polinomio resultante de la interpolación a partir de la siguiente sumatoria:

$$f_n(x) = \sum_{i=0}^n L_i(x) f(x_i)$$

En cuanto a la interpolación de Newton, también conocida como diferencias divididas, es calculada a partir del siguiente proceso recursivo:

$$f_i(x_0, x_1, x_2, \dots, x_{i-1}, x_i) = \frac{f_{i-1}(x_0, x_1, \dots, x_{i-1}, x_i) - f_i(x_0, x_1, x_2, \dots, x_{i-1})}{x_i - x_0}$$

d) ¿Qué aplicaciones tiene la interpolación?

La interpolación tiene múltiples aplicaciones. Estas aplicaciones están principalmente relacionadas con la búsqueda de una función aproximada a partir de un polinomio de grado  $n$ . Estas aproximaciones pueden ser usadas principalmente para la simplificación de cálculos y la reducción de tiempos de computación para funciones donde la precisión no debe ser extremadamente alta.

## 2. Aplicando

a) Para encontrar el polinomio cuadrático de LaGrange  $P_2(x)$  para la función  $y = f(x) = \sqrt{x}$ , debemos realizar siguiente sumatoria:

$$P_n(x) = \sum_{i=0}^n L_i(x) f(x_i)$$

Con el fin de realizar esto, tenemos que encontrar los valores para  $L_i(x)$  los cuales son obtenidos a partir del siguiente producto:

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

Partiendo de los valores para  $x_i$  dados en el enunciado, podemos realizar el cálculo de  $L_0(x)$ ,  $L_1(x)$ ,  $L_2(x)$ :

$$\begin{aligned} L_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - 1.25)(x - 1.5)}{(1 - 1.25)(1 - 1.5)} = 8(x - 1.25)(x - 1.5) \\ L_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x - 1)(x - 1.5)}{(1.25 - 1)(1.25 - 1.5)} = -16(x - 1)(x - 1.5) \\ L_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x - 1)(x - 1.25)}{(1.5 - 1)(1.5 - 1.25)} = 8(x - 1)(x - 1.25) \end{aligned}$$

Teniendo ya los valores para  $L_i(x)$ , podemos expandir la sumatoria planteada inicialmente:

$$\begin{aligned} P_2(x) &= \sum_{i=0}^2 L_i(x) f(x_i) = L_0(x) f(x_0) + L_1(x) f(x_1) + L_2(x) f(x_2) \\ P_2(x) &= 8(x - 1.25)(x - 1.5) \cdot f(1) + [-16(x - 1)(x - 1.5)] \cdot f(1.25) + 8(x - 1)(x - 1.25) \cdot f(1.5) \end{aligned}$$

Y evaluando los valores de la función en los diferentes valores de  $x$  dados:

$$P_2(x) = 8(x - 1.25)(x - 1.5) \cdot 1 + [-16(x - 1)(x - 1.5)] \cdot 1.1180339 + 8(x - 1)(x - 1.25) \cdot 1.2247448$$

Y simplificando, dando nuestro resultado final para el polinomio de LaGrange:

$$P_2(x) = 8(x - 1.25)(x - 1.5) - 17.8885438(x - 1)(x - 1.5) + 9.7979589(x - 1)(x - 1.25)$$

- b) Para encontrar los valores indeterminados de la tabla dada, será necesario, a partir de los valores que tenemos en la tabla, usar la regla recursiva de las diferencias divididas. Esta está definida de la siguiente manera:

$$f[x_{k-j}, x_{k-j+1}, \dots, x_k] = \frac{f[x_{k-j+1}, \dots, x_k] - f[x_{k-j}, \dots, x_{k-1}]}{x_k - x_{k-j}}$$

En este sentido, para poder encontrar los valores indeterminados en la tabla, vamos a requerir las siguientes "versiones" de las diferencias divididas:

$$\begin{aligned} k_0 &\rightarrow f[x_0] = f(x_0) \\ k_1 &\rightarrow f[x_1, x_0] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} \\ k_2 &\rightarrow f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} \\ k_3 &\rightarrow f[x_3, x_2, x_1, x_0] = \frac{f[x_3, x_2, x_1] - f[x_2, x_1, x_0]}{x_3 - x_0} \end{aligned}$$

Seguidamente, vamos a darle nombres a cada uno de los valores indeterminados con el fin de ubicarnos mejor:

$x_k$	$f[x_k]$	$f[.,]$	$f[.,.]$	$f[.,.,]$
$x_0 = 1.0$	3.5	-	-	-
$x_1 = 1.5$	$x_a$	$x_b$	-	-
$x_2 = 3.5$	103	45.5	11.4	-
$x_3 = 5.0$	491.5	259	61	$x_c$

El primer valor que podemos calcular es  $x_c$  debido a que conocemos los dos valores necesarios para el cálculo de este:

$$\begin{aligned} f[x_3, x_2, x_1, x_0] &= \frac{f[x_3, x_2, x_1] - f[x_2, x_1, x_0]}{x_3 - x_0} \\ f[5, 3.5, 1.5, 1] &= \frac{f[5, 3.5, 1.5] - f[3.5, 1.5, 1]}{5 - 1} \\ f[5, 3.5, 1.5, 1] &= \frac{61 - 11.4}{5 - 1} \\ f[5, 3.5, 1.5, 1] &= \frac{49.6}{4} = \frac{62}{5} = 12.4 = x_c \end{aligned}$$

Para calcular los valores de  $x_a$  y  $x_b$ , tendremos, es posible desperjarlos de las siguientes ecuaciones:

$$\begin{aligned} 11.4 &= f[3.5, 1.5, 1] = \frac{f[3.5, 1.5] - f[1.5, 1]}{3.5 - 1} \\ 11.4 &= \frac{f[3.5, 1.5] - f[1.5, 1]}{2.5} \\ f[1.5, 1] &= \frac{f[1.5] - f[1]}{1.5 - 1} & f[3.5, 1.5] &= \frac{f[3.5] - f[1.5]}{3.5 - 1.5} \\ x_b &= 2(x_a - 3.5) & 45.5 &= \frac{103 - x_a}{2} \end{aligned}$$

$$\begin{aligned}
 45.5 &= \frac{103 - x_a}{2} \\
 91 &= 103 - x_a \\
 x_a &= 103 - 91 = 12 \\
 x_b &= 2(x_a - 3.5) \\
 x_b &= 2((12) - 3.5) = 2(8.5) = 17
 \end{aligned}$$

Ya con los valores despejados, podemos remplazar en la tabla para completarla:

$x_k$	$f[x_k]$	$f[,]$	$f[, ,]$	$f[, , ,]$
$x_0 = 1.0$	3.5	-	-	-
$x_1 = 1.5$	12	17	-	-
$x_2 = 3.5$	103	45.5	11.4	-
$x_3 = 5.0$	491.5	259	61	12.4

### 3. Implementando

- a) De manera inicial, se plantea la necesidad de tener un algoritmo el cual realice el cálculo de un polinomio de LaGrange a partir de una cantidad  $n$  de puntos dada. Con el fin de dar solución a esto, se desarrolló la función `lagrangePoly(pointMatrix)` la cual permite calcular el polinomio de LaGrange correspondiente a unos puntos dados.

```

1 function output = lagrangePoly(pointMatrix)
2     [vS, hS] = size(pointMatrix);
3     output = NaN;
4     if(hS == 2 && vS > 1) %%Las entradas son correctas cuando los puntos son de (x,y)
5         for index = 1:vS
6             tempFun = @(x) 1;
7             for pain = 1:vS
8                 if(pain == index)
9                     %% skip!
10                else
11                    tempFun = @(x) tempFun(x) * ((x-pointMatrix(pain,1))/(pointMatrix(index,1) -
12                        pointMatrix(pain,1)));
13                end
14            end
15            L{index} = tempFun;
16        end
17        fullFunc = @(x) 0;
18        for finex = 1:vS
19            fullFunc = @(x) fullFunc(x) + pointMatrix(finex,2) * L{finex}(x);
20        end
21        output = fullFunc;
22    end

```

Este proceso se realiza en 3 partes. La primera es una verificación básica de la forma de la matriz de entrada, de tamaño  $n \times 2$ , donde  $n$  es mayor a 1. La segunda la cual se basa en el cálculo de los valores de  $L_i(x)$ . Esto se realiza de una manera, aunque poco eficiente por la necesidad de 2 ciclos iterativos, simple de entender.

Inicialmente se define una función anónima temporal `tempfun = @(x) 1`, esta se usará para poder guardar los valores de  $L_i(x)$ . Seguidamente, dentro del segundo ciclo iterativo, se realizará el cálculo de todos los valores que compondrán  $L_i$  y se evitarán los valores donde daría un valor dividido entre 0.

```
1 for index = 1:vS
2     tempFun = @(x) 1;
3     for pain = 1:vS
4         if(pain == index)
5             %% skip!
6         else
7             tempFun = @(x) tempFun(x) * ((x-pointMatrix(pain,1))/(pointMatrix(index,1) -
8                 pointMatrix(pain,1)));
9         end
10    end
11    L{index} = tempFun;
12 end
```

Tras calcular todos los valores respecto de  $L_i(x)$ , en la tercera parte, se construirá la función final en el tercer y último ciclo iterativo en la cual se multiplicarán los valores calculados de  $L_i(x)$  con los valores de  $f(x)$  correspondientes.

```
1 fullFunc = @(x) 0;
2 for finex = 1:vS
3     fullFunc = @(x) fullFunc(x) + pointMatrix(finex,2) * L{finex}(x);
4 end
5 output = fullFunc;
6 end
```

### 3. Anexos