Centro de Investigación en Matemáticas A.C.
Métodos Numéricos(2019)

Tarea 11 - Interpolación

Marco Antonio Esquivel Basaldua (MCMESQ)

El objetivo de la Interpolación es obtener un polinomio de aproximación para un conjunto dado de puntos discretos. La Interpolación construye un polinomio $P(x)$ de menor grado posible que pase por el conjunto dado de puntos, $(x_i, y_i), i = 0, 1, \ldots, n$. Este polinomio es conocido como Polinomio de Interpolación. El polinomio de Interpolación puede ser usado para la obtención de nuevos puntos partiendo del conocimiento del conjunto discreto de puntos dados; es usado en ingeniería y algunas ciencias en las que frecuentemente se dispone de un cierto número de puntos que se obtienen por muestreo en un experimento y se pretende construir una función que los ajuste, así como es utilizado para aproximar una función complicada por otra más simple.

En este reporte se presentan los métodos de Interpolación de Lagrange, Diferencias Divididas de Newton por diferencias divididas y el método de Hermite. Aunque a partir de un conjunto de puntos, un polinomio de menor grado posible, menor o igual a n, que pase por todos ellos es único, es decir que si se aplica el método de Lagrange o Newton se obtiene el mismo polinomio, la forma de construirlo es distinta e involucra distintas implicaciones numéricas que se discutirán en este reporte. En cuanto al método de Hermite, el polinomio obtenido es de grado menor o igual a $2n + 1$, ya que además de que este método construye un polinomio que pasa por los puntos dados, la derivada del polinomio y de la función de la que se obtienen los puntos de Interpolación también coinciden por lo que se obtiene una curva que se ajusta mejor a la función original.

La implementación de los métodos de Lagrange, Diferencias Divididas de Newton y Hermite se lleva a cabo de la siguiente manera:

- En un archivo de texto llamado "Enter points.txt" se encuentran los valores de x_i, $i = 0, 1, \ldots, n$ del conjunto de puntos (x_i, y_i), $i = 0, 1, \ldots, n$. El primero valor leído en este archivo corresponde al valor de $(n + 1)$, es decir el total de puntos de Interpolación. El usuario puede cambiar los valores de estas x_i según lo requiera.
- Se despliega un menú de funciones, $f(x)$, que servirán para encontrar los puntos de Interpolación $(x_i, y_i) = (x_i, f(x_i))$. La función elegida corre a cargo del usuario y esta será la función a la que el polinomio $P(x)$ se aproximará. Las opciones disponibles y sus respectivos intervalos son:
  - $f(x) = 2 + x$ en el intervalo $[-10, 10]$
  - $f(x) = x^2 - 5$ en el intervalo $[-10, 10]$
  - $f(x) = sin(x)$ en el intervalo $[-\pi, \pi]$
  - $f(x) = x^2(5x - 3) - 2x^4 + 4x - 5$ en el intervalo $[-2, 4]$

  Debido a que cada opción está definida sobre un intervalo distinto, el usuario debe verificar que los valores dentro del archivo "Enter points.txt" estén dentro de el intervalo dado para la función, $f(x)$, con la que se desee trabajar.
- Se calcula y se genera una gráfica que contiene las funciones $f(x)$, $P(x)$ así como los puntos de Interpolación $(x_i, f(x_i)) = (x_i, P(x_i))$ para su visualización y comparación.

- Se incluye un análisis del error entre la función $f(x)$ y el polinomio $P(x)$, dado por:

$$error = \frac{\sum_{j=0}^{m} |P(x_j) - f(x_j)|}{m}$$

  para un valor de muestras m y donde las x_j, $j = 0, 1, \ldots m$ están equiespaciadas.

- Para cada método se incluye también su tiempo de ejecución.

Método de Lagrange

Sea $y = f(x)$ una función real evaluada en el intervalo $I = [a, b]$ y sean $x_0, x_1, \ldots, x_n$ $(n+1)$ en el intervalo I, entonces existe un único polinomio de Interpolación $P(x)$ de grado menor o igual a n tal que $(x_i, f(x_i)) = (x_i, P(x_i))$ para $i = 0, 1..., n$. La siguiente es la forma del polinomio de Lagrange para la obtención del polinomio de Interpolación.

$$P(x) = \sum_{i=0}^{n} a_i l_i(x) = a_0 l_0(x) + a_1 l_1(x) + \ldots + a_n l_n(x)$$

donde:

$$l_i(x) = \prod_{(k=0),(k\neq i)}^{n} \frac{x - x_k}{x_i - x_k} = \frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)}$$

La expresión para el polinomio de Interpolación $P(x)$ es:

$$\begin{gathered} P(x) = a_0 l_0(x) + a_1 l_1(x) + \ldots + a_n l_n(x) \\ = a_0 \frac{(x-x_1)(x-x_2)\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)} + a_1 \frac{(x-x_0)(x-x_2)\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)} + \\ \ldots + a_n \frac{(x-x_0)(x-x_1)\ldots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\ldots(x_n-x_{n-1})} \end{gathered}$$

Ya que el polinomio $P(x)$ coincide con $f(x)$ en los $(n+1)$ puntos $(x_i, f(x_i))$, $i = 0, 1, \ldots, n$, entonces, $P(x_i) = f(x_i)$, $i = 0, 1, \ldots n$.

Para cualquier punto $x = x_i$, los coeficientes $a_i, i = 0, 1, \ldots, n$ se pueden calcular como:

$$a_i = P(x_i) = f(x_i)$$

El polinomio de Interpolación resultante es de la forma:

$$P(x) = \sum_{i=0}^{n} f(x_i) l_i(x) = \sum_{i=0}^{n} f(x_i) \prod_{(k=0),(k\neq i)}^{n} \frac{x - x_k}{x_i - x_k}$$

Podemos concluir que existe un único polinomio de grado igual o menor a n que interpola a $f(x)$ en los $(n+1)$ puntos dados.

Implementación del algoritmo

Para mostrar una prueba de la implementación del algoritmo se usará $f(x) = sin(x)$ en el intervalo $[-\pi, \pi]$ usando cuatro puntos de Interpolación con $x_0 = -3$, $x_1 = -2.6$, $x_2 = 0$, $x_3 = 1.2$. Este caso es usado igualmente para ejemplificar el método de Newton y el método de Hermite para posteriormente realizar una comparativa entre los tres métodos.

Al ejecutar el algoritmo se obtiene la gráfica mostrada en la figura 1 donde se pueden apreciar la función $f(x)$, el polinomio de Interpolación $P(x)$ y los puntos de Interpolación $(x_i, f(x_i)) = (x_i, P(x_i)), i = 0, 1, 2, 3$. El tiempo tomado para generar los resultados es de 0.869 milisegundos y el polinomio presenta una medida de error de 0.285062.

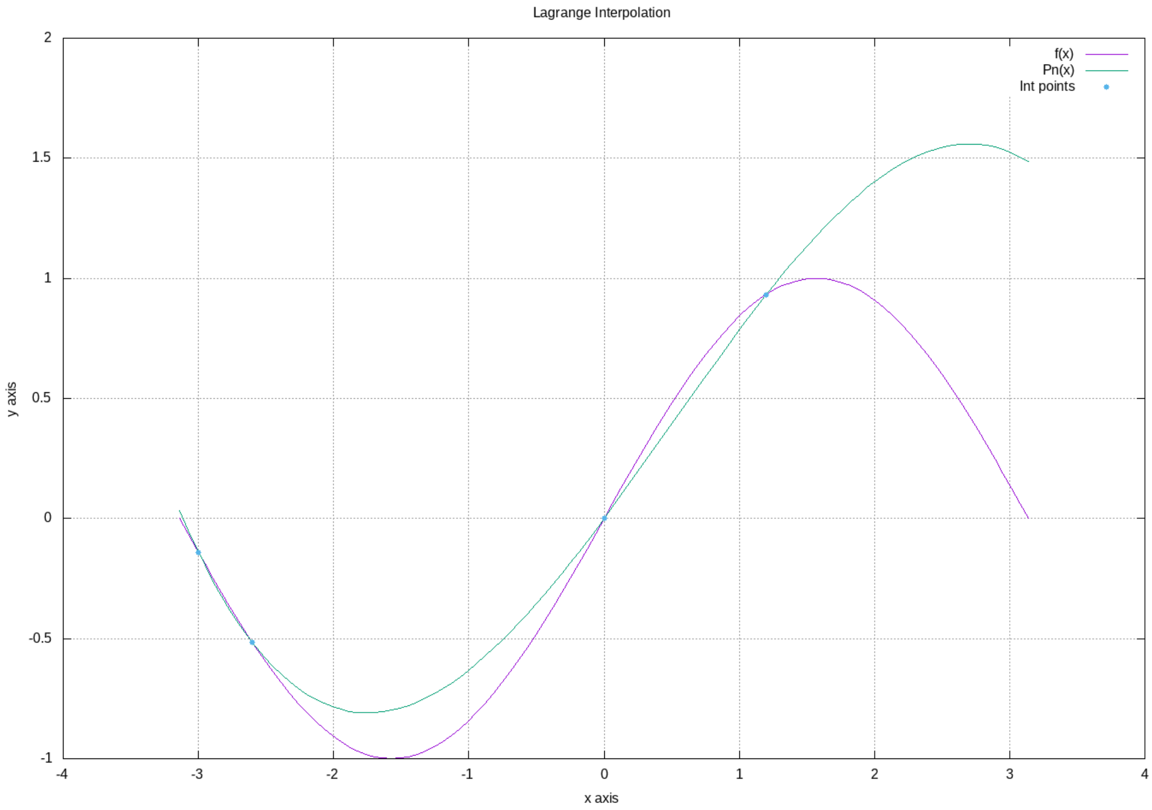

Figura 1. Polinomio de interpolación usando el método de Lagrange

Método de Diferencias Divididas de Newton

Otra forma de Interpolación polinomial es la conocida como Diferencias Divididas de Newton (DDN). Consideremos el polinomio de Interpolación de la forma de Newton con centros $x_0, x_1, \ldots x_n$.

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Ya que se trata de un polinomio de Interpolación, este pasa por los $(n + 1)$ puntos $(x_i, f(x_i))$, $i = 0, 1, 2, 3$. Por tanto $P(x_i) = f(x_i)$.

En el punto $x = x_0$, se obtiene $P(x_0) = f(x_0) = a_0$.

En el punto $x = x_1$, se obtiene $P(x_1) = f(x_1) = f(x_0) + a_1(x_1 - x_0)$, por tanto $a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$.

El coeficiente a_1 depende únicamente de los puntos x_0 y x_1. Por esta razón, tomamos la notación $f[x_0, x_1]$ para esta primer diferencia dividida. De manera similar, se tiene la siguiente formula para las primeras diferencias divididas:

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j} = f[x_j, x_i]$$

En el punto $x = x_2$, se obtiene $P(x_2) = f(x_2) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$. A partir de las diferencias divididas se obtiene la siguiente expresión para a_2:

$$a_2 = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

Ahora, a_2 depende de de los puntos x_0, x_1 y x_2 teniendo la notación para segundas diferencias divididas como $f[x_0, x_1, x_2]$, y así sucesivamente, por lo que las diferencias divididas de órdenes superiores pueden ser expresadas en términos de diferencias divididas de órdenes inferiores de la siguiente forma:

$$f[x_i, x_{i+1}, x_{x+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$$
$$f[x_i, x_{i+1}, x_{i+2}, \dots, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, x_{i+3}, \dots, x_{i+k}] - f[x_i, x_{i+1}, x_{i+2}, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Entonces el polinomio de Interpolación está dado por:

$$P(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots$$
$$+ f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1)\dots(x - x_{n-1})$$

Esta forma es conocida como la fórmula para el polinomio de Interpolación por diferencias divididas de Newton.

Divided Difference Table

x	$f(x)$	$[\,,\,]$	$[\,,\,,\,]$	$[\,,\,,\,,\,]$
x_0	$f(x_0)$			
		$f[x_0,x_1]=\frac{f(x_1)-f(x_0)}{x_1-x_0}$		
x_1	$f(x_1)$		$f[x_0,x_1,x_2]=\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$	
		$f[x_1,x_2]=\frac{f(x_2)-f(x_1)}{x_2-x_1}$		$f[x_0,x_1,x_2,x_3]=\frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$
x_2	$f(x_2)$		$f[x_1,x_2,x_3]=\frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	
		$f[x_2,x_3]=\frac{f(x_3)-f(x_2)}{x_3-x_2}$		$\vdots$
x_3	$f(x_3)$		$\vdots$	
		$\vdots$		
$\vdots$	$\vdots$			

Para la implementación del algoritmo se hará uso de la versión recursiva del polinomio de Newton también conocida como forma anidada de Newton. Dicha forma se expresa de la siguiente manera:

$$P[x] = a_0 + (x - x_0)[a_1 + (x - x_1)[a_2 + \dots + (x - x_{n-2})[a_{n-2} + a_{n-1}(x - x_{n-1})]\dots]]$$

En el polinomio de Newton de grado n, el total de multiplicaciones y sumas son $\frac{n(n+1)}{2}$ y $n + \frac{n(n+1)}{2}$, respectivamente. Al usar la forma anidada del polinomio de Newton se realizan únicamente n multiplicaciones y $2n$ sumas, por lo que se tiene un algoritmo más rápido y eficiente.

Implementación del Algoritmo

Para mostrar una prueba de la implementación del algoritmo se usará $f(x) = sin(x)$ en el intervalo $[-\pi, \pi]$ usando cuatro puntos de Interpolación con $x_0 = -3$, $x_1 = -2.6$, $x_2 = 0$, $x_3 = 1.2$. Este caso es usado igualmente para ejemplificar el método de Lagrange y el método de Hermite para posteriormente realizar una comparativa entre los tres métodos.

Al ejecutar el algoritmo se obtiene la gráfica mostrada en la figura 2 donde se pueden apreciar la función $f(x)$, el polinomio de Interpolación $P(x)$ y los puntos de Interpolación $(x_i, f(x_i)) = (x_i, P(x_i))$, $i = 0, 1, 2, 3$. El tiempo tomado para generar los resultados es de 0.659 milisegundos y el polinomio presenta una medida de error de 0.285062.

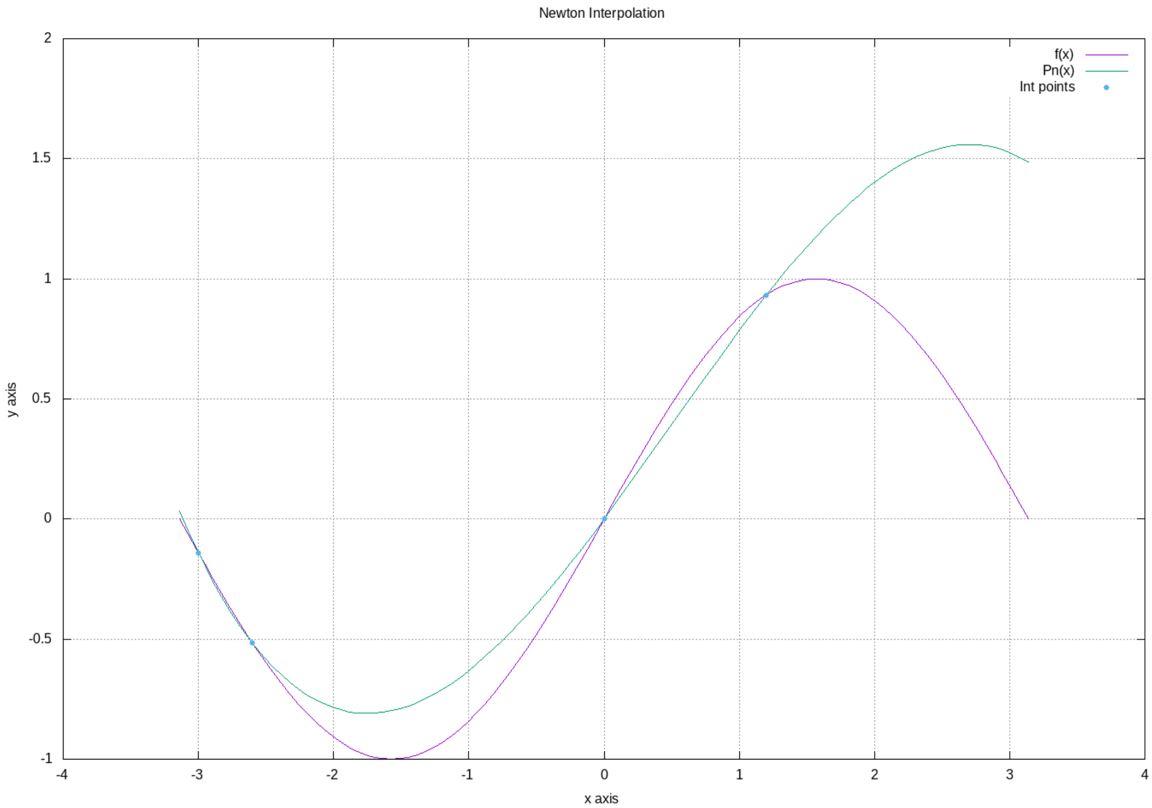

Figura 2. Polinomio de interpolación usando el método de las Diferencias Divididas de Newton

Método de Hermite

El método de Hermite permite encontrar un polinomio de Interpolación que ademas de coincidir con un conjunto de puntos dados de una función $f(x)$, también coincide con la derivada de la función $f(x)$ en estos puntos, por lo que se tiene:

$$
\begin{gathered}
P(x_i) = f(x_i) \\
P'(x_i) = f'(x_i) \\
i = 0, 1, 2, \ldots, n
\end{gathered}
$$

El polinomio de Interpolación de Hermite es de grado menor o igual a $(2n+1)$ para una colección de $(n+1)$ puntos de Interpolación. El polinomio está dado por la fórmula:

$$P(x) = \sum_{i=0}^{n}[(-2l_i'(x_i)x + 1 + 2x_i l_i'(x_i))l_i^2(x)f(x_i n+) + (x - x_i)l_i^2(x)f'(x_i)])$$

Donde:

$$l_i(x) = \prod_{k=0,k\neq i}^{n} \frac{x - x_k}{x_i - x_k} = \frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)}$$

$$l_i'(x) = \sum_{p=0,p\neq i}^{n}\left[\frac{1}{x_i - x_p}\prod_{m=0,m\neq(p,i)}^{n}\frac{x - x_m}{x_i - x_m}\right]$$

Ya que en la forma del polinomio de Hermite solo se trabaja $l_i'(x_i)$, su fórmula está dada por:

$$l_i'(x_i) = \sum_{p=0,p\neq i}^{n} \frac{1}{x_i - x_p}$$

Implementación del Algoritmo

Para mostrar una prueba de la implementación del algoritmo se usará $f(x) = sin(x)$ en el intervalo $[-\pi, \pi]$ usando cuatro puntos de Interpolación con $x_0 = -3$, $x_1 = -2.6$, $x_2 = 0$, $x_3 = 1.2$. Este caso es usado igualmente para ejemplificar el método de Lagrange y el método de las Diferencias Divididas de Newton para posteriormente realizar una comparativa entre los tres métodos.

Al ejecutar el algoritmo se obtiene la gráfica mostrada en la figura 3 donde se pueden apreciar la función $f(x)$, el polinomio de Interpolación $P(x)$ y los puntos de Interpolación $(x_i, f(x_i)) = (x_i, P(x_i)), i = 0, 1, 2, 3$. El tiempo tomado para generar los resultados es de 0.622 milisegundos y el polinomio presenta una medida de error de 0.036569.

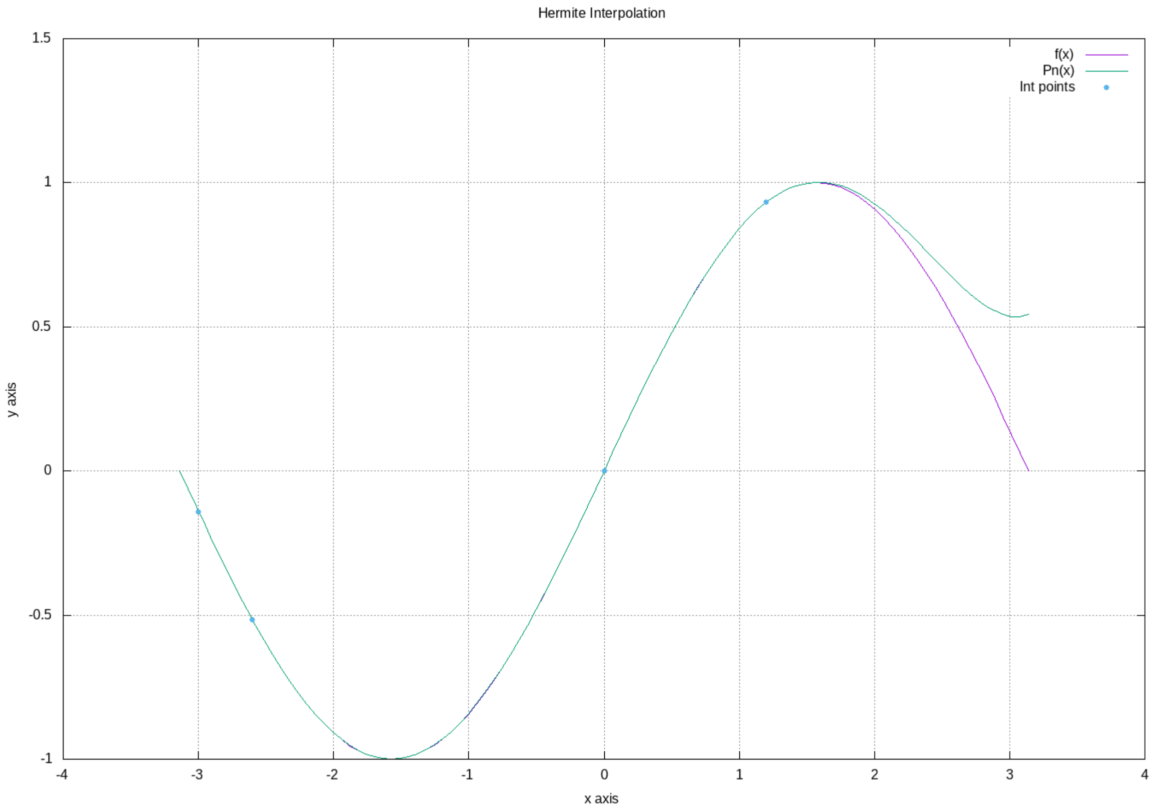

Figura 2. Polinomio de interpolación usando el método de Hermite

Comparativa Entre los Métodos

A continuación se presenta una tabla comparativa entre los tres métodos descritos anteriormente. En esta tabla se compara el tiempo tomado por cada algoritmo y su tiempo de ejecución para las funciones presentadas en la introducción de este documento tomando los mismos puntos por función para cada uno de los métodos. El tiempo de ejecución mostrado en la tabla corresponde al tiempo llevado a cabo solo por el algoritmo en cuestión y no por todo el programa.

Comparativa	**Lagrange**	**Newton**	**Hermite**
$f(x) = 2 + x$ Para: $x_0 = -9.8, x_1 = 3.4$			
Tiempo de ejecución (milisegundos)	0.651	0.501	0.463
Error	0.0	0.0	0.0
$f(x) = x^2 + 5$ Para: $x_0 = -10, x_1 = 0, x_2 = 10$			
Tiempo de ejecución (milisegundos)	0.766	0.560	0.574
Error	0.0	0.0	0.0
$f(x) = sin(x)$ Para: $x_0 = -3, x_1 = -2.6, x_2 = 0, x_3 = 1.2$			
Tiempo de ejecución (milisegundos)	0.869	0.659	0.622
Error	0.285062	0.285062	0.036569
$f(x) = x^2(5x - 3) - 2x^4 + 4x - 5$ Para: $x_0 = -1.9, x_1 = -0.4, x_2 = 1, x_3 = 2.3$			
Tiempo de ejecución (milisegundos)	0.567	0.549	0.607
Error	29.206790	29.206790	0.000024

Como se puede apreciar en la tabla cuando se trabaja con un número de puntos mayor al grado de la función de la que se obtienen las coordenadas $(x_i, f(x_i))$ se obtiene un valor de error igual a cero, es decir, el polinomio ajustado $P(x)$, coincide con la función $f(x)$, solo variando el tiempo de ejecución de un algoritmo a otro. Cuando este caso no se presenta, ya sea que se trabaje con un número de puntos menor o igual al grado de la función $f(x)$ o se trabaje con una función que no sea polinomial, existe una medida de error mayor a cero.

También se puede notar que el error entre el método de Lagrange y el método de Diferencias Divididas de Newton es siempre el mismo, pero varia el tiempo de ejecución entre ambos algoritmos siendo el tiempo de ejecución del algoritmo de Lagrange siempre mayor. Esto es debido a que el utilizar la forma anidada del método de Newton se reduce considerablemente el número de multiplicaciones y sumas usadas para este método.

En cuanto al método de Interpolación de Hermite se aprecia que el error siempre es menor o igual al error generado por los métodos de Lagrange y Newton, esto es debido a que el método de Hermite hace que no solamente los puntos sean comunes para $f(x)$ y $P(x)$ sino también las pendientes en estos puntos para $f(x)$ y $P(x)$. Sin embargo la disminución del error es compensada con un tiempo de ejecución mayor que para los métodos de Lagrange y Newton.