



Centro de Investigación en Matemáticas, A.C.

Descripción tarea 9 - Métodos numéricos

Erick Salvador Alvarez Valencia

26 de Octubre de 2017

1. Introducción

En el presente reporte se describirán tres métodos de interpolación de puntos, los cuales se enfocan en generar los coeficientes de el polinomio que pasará por el conjunto de puntos dado. Se describirá a grandes rasgos en qué consiste cada método, un ejemplo de ejecución, y al finalizar se mostrarán los pasos para compilar y ejecutar cada uno de los programas.

2. Interpolación polinomial

2.1. Descripción

Se tiene un conjunto de $n + 1$ puntos distintos $\{(x_i, y_i)\}_{i=0}^n$ y se quiere encontrar un polinomio de grado m a lo mucho que pase por ellos tal que $p_m(x_i) = y_i$. Una idea es construir un polinomio de grado $m = n + 1$ que sea combinación lineal de las abscisas y que su resultado sea una de las ordenadas por las que se quiere interpolar.

$$p_m(x_i) = a_m x_i^m + a_{m-1} x_i^{m-1} + \dots + a_1 x_i + a_0 = y_i$$

Al hacer esto para cada y_i obtendremos un sistema de $(n + 1) * (n + 1)$ ecuaciones, que al resolverlo podremos conocer el valor de los coeficientes a_i del polinomio interpolador.

2.2. Ejemplo de ejecución

A continuación se mostrará el resultado de la ejecución del método de interpolación polinomial con un conjunto de 7 puntos, a su vez se adjuntarán las gráficas generadas para ver qué tan buena

1.00000000e+00	1.05840000e+04
2.28571429e+00	-3.10696432e+03
3.57142857e+00	-5.51295541e+03
4.85714286e+00	1.97779050e+04
6.14285714e+00	5.04288422e+04
7.42857143e+00	3.22807622e+04
8.71428571e+00	-4.86741102e+04
1.00000000e+01	0.00000000e+00

(a) Puntos iniciales.

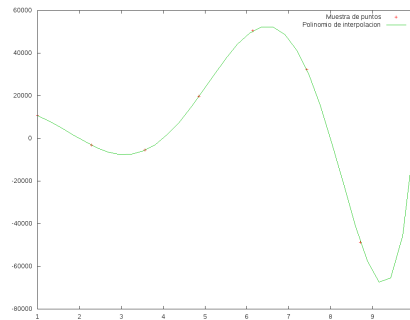
1.000000	10584.000000
1.281250	8140.278283
1.562500	5121.366411
1.843750	1827.775250
2.125000	-1411.032379
2.406250	-4257.433428
2.687500	-6386.826123
2.968750	-7508.499645
3.250000	-7385.490417
3.531250	-5852.723422
3.812500	-2832.736946
4.093750	1651.710807
4.375000	7468.854825
4.656250	14373.657711
4.937500	22008.766977
5.218750	29911.396819
5.500000	37526.835938
5.781250	44229.282983
6.062500	49350.711224
6.343750	52218.464003
6.625000	52202.282568
6.906250	48771.467872
7.187500	41562.877895
7.468750	30460.462104
7.750000	15687.034607
8.031250	-2091.012407
8.312500	-21645.353359
8.593750	-41053.030529
8.875000	-57548.710263
9.156250	-67361.192584
9.437500	-65533.472598
9.718750	-45725.652130
10.000000	0.000000

(b) Puntos de interpolación.

Figura 1: Tablas de puntos para la interpolación polinomial.

fue la aproximación.

En la Figura 1. podemos observar que la discretización generada en el intervalo $[x_1, x_n]$ pasa de manera muy cercana a los puntos propuestos en la tabla 1.a.



(a) Puntos iniciales.

Figura 2: Tablas de puntos para la interpolación polinomial.

3. Polinomios de Lagrange

3.1. Descripción

Una base polinómica de Lagrange se define de la siguiente manera: $l_j(x) = \prod_{j=0, j \neq i}^n \frac{x-x_j}{x_i-x_j}$. Si se tiene una serie de puntos distintos x_0, x_1, \dots, x_n y $f(x)$ es una función cuyos valores están dados en esos puntos por $f(x_0), f(x_1), \dots, f(x_n)$ entonces existe un único polinomio $L(x)$ de grado al menos n con la propiedad siguiente: $f(x_k) = L(x_k)$, $k = 0, 1, 2, \dots, n$ donde el polinomio $L(x)$ es una combinación lineal de las bases polinómicas de Lagrange: $L(x) = \sum_{j=0}^n y_j l_j(x)$. Se puede demostrar que las bases polinómicas de Lagrange son una base del espacio de los polinomios, de esta forma se garantiza la existencia de $L(x)$.

Para este programa se leyeron los $(n + 1)$ puntos y se calculó el polinomio $p_n(x)$ como la combinación lineal de las bases polinómicas de Lagrange donde los coeficientes de la combinación venían siendo representados por los valores de las abscisas de los puntos.

3.2. Ejemplo de ejecución

A continuación se mostrará el resultado de la ejecución del método de interpolación por polinomios de Lagrange con un conjunto de 7 puntos.

1.00000000e+00	1.05840000e+04
2.28571429e+00	-3.18696432e+03
3.57142857e+00	-5.51295541e+03
4.85714286e+00	1.97779050e+04
6.14285714e+00	5.04288422e+04
7.42857143e+00	3.22807622e+04
8.71428571e+00	-4.86741102e+04
1.00000000e+01	0.00000000e+00

(a) Puntos iniciales.

1.000000	10584.000000
1.281250	8140.278283
1.562500	5121.366411
1.843750	1827.775250
2.125000	-1411.032379
2.406250	-4257.433428
2.687500	-6386.826123
2.968750	-7508.499645
3.250000	-7385.490417
3.531250	-5852.723422
3.812500	-2832.736946
4.093750	1651.710807
4.375000	7468.854825
4.656250	14373.657711
4.937500	22008.766977
5.218750	29911.396819
5.500000	37526.835937
5.781250	44229.282983
6.062500	49350.711224
6.343750	52218.464003
6.625000	52202.282568
6.906250	48771.467872
7.187500	41562.877895
7.468750	30460.462104
7.750000	15687.034607
8.031250	-2091.012407
8.312500	-21645.353359
8.593750	-41053.030529
8.875000	-57548.710263
9.156250	-67361.192584
9.437500	-65533.472598
9.718750	-45725.652130
10.000000	0.000000

(b) Puntos de interpolación.

Figura 3: Tablas de puntos para la interpolación de Lagrange.

Podemos ver que para el conjunto de puntos usado en el método de interpolación polinomial, este método nos arroja exactamente el mismo conjunto de puntos de interpolación, claro que hay que mencionar que la discretización se hizo de la misma forma. Por lo anterior, se puede ver que la gráfica que se generaría sería la misma, por lo que se omitirá.

4. Polinomio de Newton de diferencias divididas

El método de interpolación mediante diferencias divididas es un método de construcción de polinomios interpoladores que utiliza un esquema recursivo. Se considera a $P_n(x)$ un polinomio de grado al menos n , que se aproxima con la función $f(x)$ en los puntos distintos x_0, x_1, \dots, x_n . Las diferencias divididas de $f(x)$ respecto a estos puntos pueden ser obtenidas demostrando que $P_n(x)$ viene definido por:

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

en donde a_0, a_1, \dots, a_n son los coeficientes polinomiales. Para este método hay que aplicar un algoritmo que calcule los términos de las diferencias divididas, las cuales denotamos por:

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

Vemos que hay una relación recursiva en ellas, por lo cual podemos ir guardando las que ya se calcularon en una tabla y posteriormente usarlas para calcular las próximas. Encontraremos el polinomio interpolador como una combinación lineal del polinomio de Newton y de los coeficientes que serán las diferencias divididas.

$$P_n(x) = \sum_{i=0}^n a_{0,i} \prod_{j=0}^{i-1} (x - x_j)$$

Donde $a_{0,i}$ era la i -ésima columna de la matriz de diferencias divididas.

4.1. Ejemplo de ejecución

A continuación se mostrará el resultado de la ejecución del método de interpolación por polinomios de Newton con un conjunto de 7 puntos.

1.00000000e+00	1.05840000e+04
2.28571429e+00	-3.10696432e+03
3.57142857e+00	-5.51295541e+03
4.85714286e+00	1.97779050e+04
6.14285714e+00	5.04288422e+04
7.42857143e+00	3.22807622e+04
8.71428571e+00	-4.86741102e+04
1.00000000e+01	0.00000000e+00

(a) Puntos iniciales.

1.000000	10584.000000
1.281250	8140.278283
1.562500	5121.366411
1.843750	1827.775250
2.125000	-1411.032379
2.406250	-4257.433428
2.687500	-6386.826123
2.968750	-7508.499645
3.250000	-7385.490417
3.531250	-5852.723422
3.812500	-2832.736946
4.093750	1651.710807
4.375000	7468.854825
4.656250	14373.657711
4.937500	22008.766977
5.218750	29911.396819
5.500000	37526.835937
5.781250	44229.282983
6.062500	49350.711224
6.343750	52218.464003
6.625000	52202.282568
6.906250	48771.467872
7.187500	41562.877895
7.468750	30460.462104
7.750000	15687.034607
8.031250	-2091.012407
8.312500	-21645.353359
8.593750	-41053.030529
8.875000	-57548.710263
9.156250	-67361.192584
9.437500	-65533.472598
9.718750	-45725.652130
10.000000	-0.000000

(b) Puntos de interpolación.

Figura 4: Tablas de puntos para la interpolación de Newton.

Al igual que en los dos métodos anteriores, este método de interpolación volvió a generar la misma tabla de puntos y por ende la misma gráfica. Podríamos ver diferencias en estos métodos con usando un conjunto de puntos inicial más grande, ya que con el método de interpolación polinomial la matriz se va mal condicionando y el método de Lagrange requiere muchas operaciones matemáticas.

5. Compilación y ejecución

Para esta tarea, los programas se dividieron en tres carpetas, mediante las cuales se encuentran los archivos del código y los de prueba, hay que mencionar que para cada programa se usará la misma forma de compilar, por lo que solo es necesario describir los pasos una vez.

Para compilar: En la carpeta encontraremos los archivos `.c` y `.h` con los que se podrá compilar el ejecutable. De la misma forma, en conjunto con los archivos anteriores, también podremos encontrar un Makefile para, en caso de encontrarse en linux, compilar de manera sencilla.

1. **Compilar usando Makefile:** En la terminal, nos colocamos en el directorio donde se encuentre el programa, y ejecutamos el comando `make`, automáticamente se realizará la compilación y se generará el ejecutable. El Makefile también contiene el comando `make clean` el cual limpiará los archivos generados por la compilación, incluyendo el ejecutable.
2. **Compilar directamente:** De la misma forma, podemos compilar directamente usando los siguientes comandos (en terminal):

- `gcc -c main.c -o obj/main.o`

- `gcc -c memo.c -o obj/memo.o`
- `gcc -c reader.c -o obj/reader.o`
- `gcc -c matriz_vector.c -o obj/matriz_vector.o`
- `gcc -c met_num.c -o obj/met_num.o`
- `gcc -o main obj/main.o obj/memo.o obj/reader.o obj/matriz_vector.o obj/met_num.o -lm`

Para ejecutar: Únicamente debemos de usar el comando `./main` para ejecutar el programa en consola, este recibe un argumento:

- **Un string:** El nombre del archivo binario donde se encuentra la matriz de puntos.

El programa ejecutará el método de interpolación polinomial con los puntos iniciales y generará un archivo de texto que contendrá los puntos discretos obtenidos evaluando el polinomio interpolador.