



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA

REPORTE DE ALGORITMOS

INTERPOLACION DE NEWTON

Nombre	Expediente
Zuñiga Fragoso Diego Joel	317684

Asignatura: Método Numéricos 2023-2

Docente: Vargas Vázquez Damián



I. Antecedentes teóricos

La interpolación de Newton es un método ampliamente utilizado para aproximar una función desconocida a partir de un conjunto discreto de puntos conocidos. Este método fue desarrollado por Sir Isaac Newton y es una técnica fundamental en el campo de la interpolación polinómica. La idea principal detrás de la interpolación de Newton es construir un polinomio que pase exactamente por los puntos dados.

En lugar de utilizar un único polinomio para todos los puntos, la interpolación de Newton se basa en polinomios de diferencias divididas, que son más eficientes y permiten la fácil adición de nuevos puntos de interpolación sin tener que recalcular todo el polinomio.

Este método ha demostrado ser muy útil en diversas disciplinas, desde la aproximación de funciones matemáticas hasta la representación de datos experimentales. Su versatilidad y eficiencia lo convierten en una herramienta valiosa para abordar problemas de interpolación en diversas áreas de la ciencia y la ingeniería.

II. Algoritmos y sus resultados

Cada algoritmo esta seccionado e incluye descripciones de lo que sucede. Además de contar con capturas de sus resultados

Código

```
#include <stdio.h>
#include <iostream>

using namespace std;

double NewtonInterpolation(int n, double* x, double* y, double xvalue)
{
    double result = 0;

    // Creamos matriz de memoria dinamica
    double** divDiff = new double* [n];
    for (int i = 0; i < n; i++)
        divDiff[i] = new double[n];

    // Inicializacion de la tabla de diferencias divididas
    for (int i = 0; i < n; i++)
        divDiff[i][0] = y[i];
```



```
// Cálculo de las diferencias divididas
for (int j = 1; j < n; j++)
    for (int i = 0; i < n - j; i++)
        divDiff[i][j] = (divDiff[i + 1][j - 1] - divDiff[i][j - 1]) /
            (x[i + j] - x[i]);

// Calcula la ecuación de interpolación
cout << "\n\nLa ecuacion de interpolacion de Newton es:\n";
printf("\nf(x) = %g", divDiff[0][0]);

for (int j = 1; j < n; j++)
{
    cout << " + ";

    printf("(%g)", divDiff[0][j]);

    for (int i = 0; i < j; i++)
        printf("(x - %g)", x[i]);
}
cout << endl;

// Calcula el resultado de la interpolación
for (int j = 0; j < n; j++)
{
    double term = divDiff[0][j];

    for (int i = 0; i < j; i++)
        term *= (xvalue - x[i]);
    result += term;
}

// Liberamos memoria dinamica
for (int i = 0; i < n; i++)
    delete[] divDiff[i];
delete[] divDiff;

return result;
}

int main()
{
    cout << "Programa hecho para calcular la interpolacion de newton.\n\n";

    Points:
    int n;
    cout << "Ingresa el numero de puntos:\t\t";
    cin >> n;

    if (n <= 0)
        goto Points;

    // Creamos arreglos de memoria dinamica
    double* x = new double[n];
    double* y = new double[n];

    // Pedimos al usuario que ingrese los puntos
```



```
for (int i = 0; i < n; i++)
{
    cout << "\n\nIngrese x_" << i << " =\t\t";      cin >> x[i];
    cout << "Ingrese f(x_" << i << ") =\t";          cin >> y[i];
}

// Ingreso de valor de x
double xvalue;
cout << "\n\nIngresa el valor de x para la interpolacion: ";
cin >> xvalue;

double result = NewtonInterpolation(n, x, y, xvalue);

printf("\n\nEl valor interpolado en x = %g es: %g\n", xvalue, result);

// Liberamos memoria dinamica
delete[] x;
delete[] y;

return 0;
}
```

Resultado

```
Programa hecho para calcular la interpolacion de newton.
Ingresa el numero de puntos:      5

Ingresa x_0 =      0.4
Ingresa f(x_0) =      1

Ingresa x_1 =      2.5
Ingresa f(x_1) =      0.5

Ingresa x_2 =      4.3
Ingresa f(x_2) =      2

Ingresa x_3 =      5
Ingresa f(x_3) =      2.55

Ingresa x_4 =      6
Ingresa f(x_4) =      4

Ingresa el valor de x para la interpolacion: 5.2

La ecuacion de interpolacion de Newton es:
f(x) = 1 + (-0.238095)(x - 0.4) + (0.274725)(x - 0.4)(x - 2.5) + (-0.0638637)(x - 0.4)(x - 2.5)(x - 4.3) + (0.0323126)(x - 0.4)(x - 2.5)(x - 4.3)(x - 5)

El valor interpolado en x = 5.2 es: 2.74806
```

III. Conclusiones

En conclusión, la interpolación de Newton se destaca como una técnica eficaz y versátil para aproximarse a funciones desconocidas a partir de un conjunto discreto de puntos conocidos. Desarrollada por Sir Isaac Newton, esta metodología de interpolación polinómica ha demostrado su utilidad en una variedad de campos, desde la matemática pura hasta la ingeniería y las ciencias aplicadas.



La interpolación de Newton ofrece una ventaja significativa al utilizar polinomios de diferencias divididas, permitiendo una mayor flexibilidad y eficiencia en la aproximación. Además, su capacidad para incorporar nuevos puntos de interpolación sin tener que recalcular todo el polinomio lo hace especialmente práctico en situaciones donde se añaden datos adicionales.