Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра Автоматизованих Систем Обробки Інформації та Управління

Лабораторна робота № 5

з дисципліни «Чисельні методи»

на тему

«**Інтерполяційні поліноми**»

Виконав:

студент гр. ІС-03

Філоненко Андрій

Викладач:

доц. Рибачук Л.В.

Київ – 2021

### Зміст

[Зміст 2](#_Toc190845267)

[1 Постановка задачі 3](#_Toc190845268)

[2 Розв’язок 4](#_Toc190845269)

[3 Розв’язок у Mathcad 5](#_Toc190845270)

[4 Лістинг програми](#_Toc190845271) 7

[5 Висновок](#_Toc190845270) 10

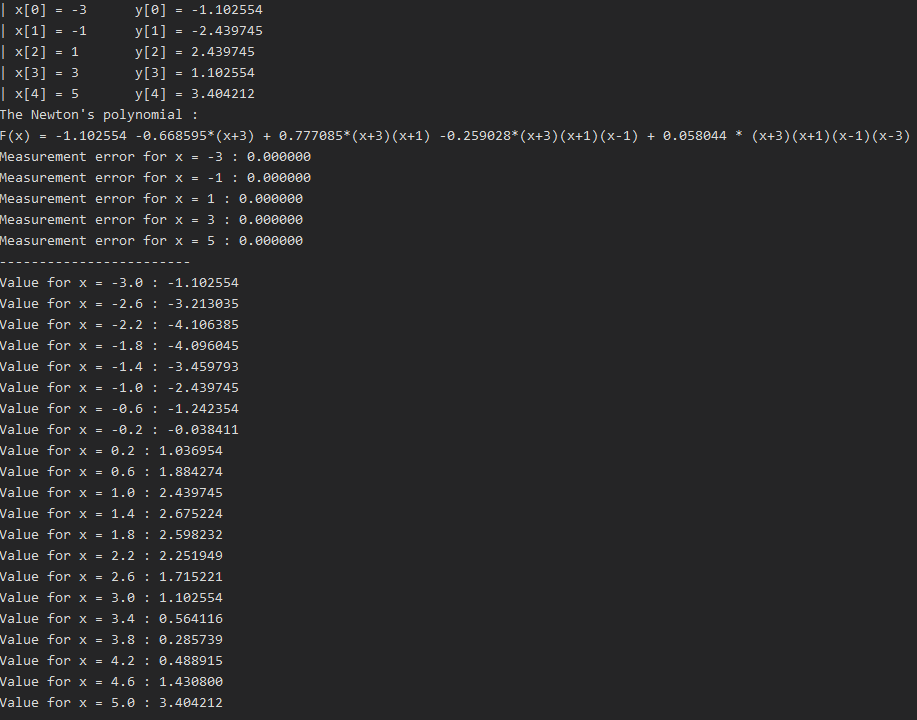
### 1 Постановка задачі

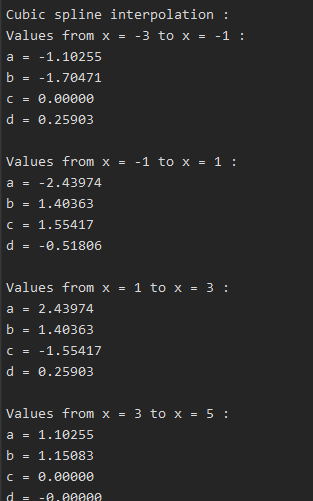
### 

### Вузли інтерполяції :

Функція : 

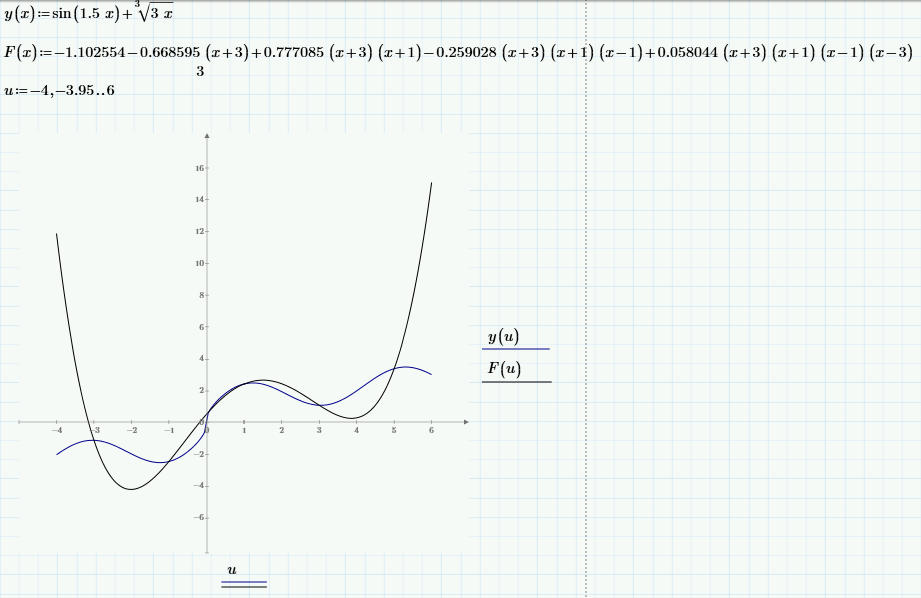
### 2 Розв’язок

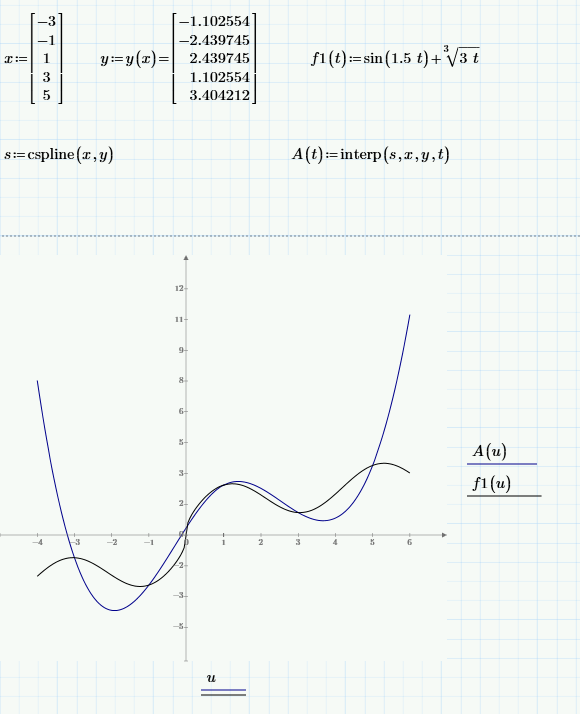




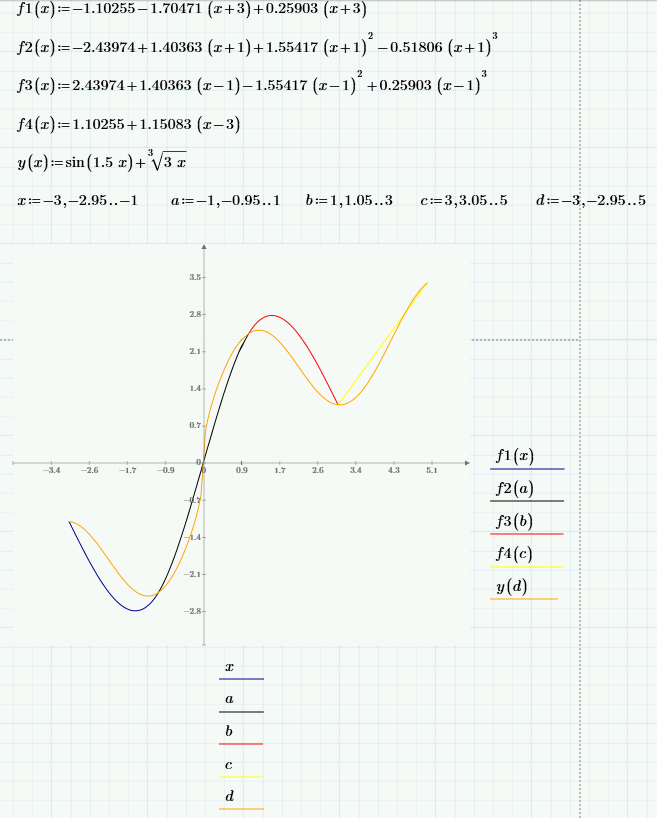
**3 Розв’язок у MathCad**

Порівняння графіку функції та інтерп. Поліному :



Порівняння графіку функції та інтерполяції сплайнами середовища MathCad :

Порівняння графіку функції та інтерполяції сплайнами програми :



**4 Лістинг програми**

**С# Main :**

using System;  
  
namespace NumMethods5  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 double[] x = {-3, -1, 1, 3, 5};  
 double[] y = {-1.102554, -2.439745, 2.439745, 1.102554, 3.404212};  
 for (int i = 0; i < x.GetLength(0); i++)  
 {  
 Console.WriteLine($"| x[{i}] = {x[i]} \t y[{i}] = {y[i]}");  
 }  
  
 Console.WriteLine("\n\nThe Newton's polynomial : ");  
 Functions.newtMethod(x, y, 5);  
  
 Functions.krokMalenky(x, y, 5);  
 Console.WriteLine("\nCubic spline interpolation : ");  
 for (int i = 0; i < 4; i++)  
 {  
 Console.WriteLine($"Values from x = {x[i]} to x = {x[i + 1]} : ");  
 Functions.cubSplineInterp(x, y, 5, x[i]);  
 }  
 }  
 }  
}

**C# Functions :**

using System;  
  
namespace NumMethods5  
{  
 internal class Functions  
 {  
 static public void krokMalenky(double[] x, double[] y, int n)  
 {  
 double[] firstDifference = new double[ --n];  
 double[] secondDifference = new double[--n];  
 double[] thirdDifference = new double[--n];  
 double[] fourthDifference = new double[--n];  
  
 for (int i = 0; i < firstDifference.GetLength(0); i++)  
 {  
 firstDifference[i] = (y[i] - y[i + 1]) / (x[i] - x[i + 1]);  
 }  
 for (int i = 0; i < secondDifference.GetLength(0); i++)  
 {  
 secondDifference[i] = (firstDifference[i] - firstDifference[i + 1]) / (x[i] - x[i + 2]);  
 }  
 for (int i = 0; i < thirdDifference.GetLength(0); i++)  
 {  
 thirdDifference[i] = (secondDifference[i] - secondDifference[i + 1]) / (x[i] - x[i + 3]);  
 }  
 for (int i = 0; i < fourthDifference.GetLength(0) ; i++)  
 {  
 fourthDifference[i] = (thirdDifference[i] - thirdDifference[i + 1]) / (x[i] - x[i + 4]);  
 }  
  
 Console.WriteLine("------------------------");  
 for (double i = x[0]; i <= x[4]; i += 0.4)  
 {  
 double znachennya = y[0] + firstDifference[0] \* (i - x[0]) +  
 secondDifference[0] \* (i - x[0]) \* (i - x[1]) +  
 thirdDifference[0] \* (i - x[0]) \* (i - x[1]) \* (i - x[2]) + fourthDifference[0] \*  
 (i - x[0]) \* (i - x[1]) \* (i - x[2]) \* (i - x[3]);  
 Console.Write($"Value for x = {i:f1} : ");  
 Console.Write($"{znachennya:f6}");  
 Console.WriteLine();  
 }  
 }  
 static public void newtMethod(double[] x, double[] y, int n)  
 {  
 double[] firstDifference = new double[ --n];  
 double[] secondDifference = new double[--n];  
 double[] thirdDifference = new double[--n];  
 double[] fourthDifference = new double[--n];  
  
 for (int i = 0; i < firstDifference.GetLength(0); i++)  
 {  
 firstDifference[i] = (y[i] - y[i + 1]) / (x[i] - x[i + 1]);  
 }  
 for (int i = 0; i < secondDifference.GetLength(0); i++)  
 {  
 secondDifference[i] = (firstDifference[i] - firstDifference[i + 1]) / (x[i] - x[i + 2]);  
 }  
 for (int i = 0; i < thirdDifference.GetLength(0); i++)  
 {  
 thirdDifference[i] = (secondDifference[i] - secondDifference[i + 1]) / (x[i] - x[i + 3]);  
 }  
 for (int i = 0; i < fourthDifference.GetLength(0) ; i++)  
 {  
 fourthDifference[i] = (thirdDifference[i] - thirdDifference[i + 1]) / (x[i] - x[i + 4]);  
 }  
 Console.Write("F(x) = ");  
 Console.Write($"{y[0]:f6} {firstDifference[0]:f6}\*(x+3) + {secondDifference[0]:f6}\*(x+3)(x+1) {thirdDifference[0]:f6}\*");  
 Console.Write($"(x+3)(x+1)(x-1) + {fourthDifference[0]:f6} \* (x+3)(x+1)(x-1)(x-3)");  
   
 Console.WriteLine();  
 double measErr = 0;  
 for (int i = 0; i < x.GetLength(0); i++) {  
 Console.Write($"Measurement error for x = {x[i]} : ");  
 measErr = y[i] - (y[0] + firstDifference[0] \* (x[i] - x[0]) +  
 secondDifference[0] \* (x[i] - x[0]) \* (x[i] - x[1]) +  
 thirdDifference[0] \* (x[i] - x[0]) \* (x[i] - x[1]) \* (x[i] - x[2]) +  
 fourthDifference[0] \* (x[i] - x[0]) \* (x[i] - x[1]) \* (x[i] - x[2]) \* (x[i] - x[3]));  
 Console.WriteLine($"{measErr:f6}");  
 }  
  
 }  
   
  
 static public void cubSplineInterp(double[] X, double[] Y, int n, double x0)  
 {  
 int s = n - 1;  
   
 double[] a = new double[s];  
 double[] b = new double[s];  
 double[] d = new double[s];  
 double[] h = new double[s];  
  
 double[,] A = new double[n-1,n];  
  
 double[] vectorTDM = new double[n];  
  
 for (int i = 0; i < s; i++)  
 {  
 a[i] = Y[i];  
 h[i] = X[i + 1] - X[i];  
 }  
 A[0, 0] = 1;  
 A[n - 2, n - 2] = 1;  
 for (int i = 1; i < n - 2; i++)  
 {  
 A[i,i - 1] = h[i - 1];  
 A[i,i] = 2 \* (h[i - 1] + h[i]);  
 A[i,i + 1] = h[i];  
 vectorTDM[i] = 3 \* (((Y[i + 1] - Y[i]) / h[i]) - ((Y[i] - Y[i - 1]) / h[i - 1]));  
 }  
 double [] c = runTriMatrix(A, vectorTDM, s);  
  
 for (int i = 0; i < s; i++)  
 {  
 if (i != s-1)  
 {  
 d[i] = (c[i + 1] - c[i]) / (3 \* h[i]);  
 b[i] = ((a[i + 1] - a[i]) / h[i]) - h[i] \* (c[i + 1] + 2 \* c[i]) / 3;  
 }  
 else  
 {  
 d[i] = (-1) \* (c[i] / (3 \* h[i]));  
 b[i] = ((a[i] - a[i - 1]) / h[i]) - ((2 \* h[i] \* c[i]) / 3);  
  
 }  
 }  
 d[n - 2] = -c[n - 2] / (3 \* h[n - 2]);  
 b[n - 2] = ((Y[n - 1] - Y[n - 2]) / h[n - 2]) - 2 \* h[n - 2] \* c[n - 2] / 3;  
 int m = 0;  
 for (int i = 1; i < n; i++)  
 {  
 if (x0 >= X[i-1] && x0 <= X[i])  
 {  
 m = i-1;  
 }  
 }  
   
 Console.WriteLine($"a = {a[m]:f5}");  
 Console.WriteLine($"b = {b[m]:f5}");  
 Console.WriteLine($"c = {c[m]:f5}");  
 Console.WriteLine($"d = {d[m]:f5}\n");  
 }  
  
 static public double [] runTriMatrix(double[,]A, double [] b, int n)  
 {  
 double[] K = new double[n];  
 int size = n - 1;  
  
 double y = A[0, 0];  
 double[] a = new double[n];  
 double[] B = new double[n];  
 a[0] = -A[0,1] / y;  
 B[0] = b[0] / y;  
 for (int i = 1; i < size; i++)  
 {  
 y = A[i,i] + A[i,i - 1] \* a[i - 1];  
 a[i] = -A[i,i + 1] / y;  
 B[i] = (b[i] - A[i,i - 1] \* B[i - 1]) / y;  
 }  
 K[size] = B[size];  
 for (int i = size - 1; i >= 0; i--)  
 {  
 K[i] = a[i] \* K[i + 1] + B[i];  
 }  
 return K;  
 }  
 }  
}

**5 Висновок**

Програма виконує алгоритм будування поліному Ньютона, а також знаходження коефіцієнтів для кубічної інтерполяції сплайнами. Застосована мова програмування С# а також середовище розрахунків MathCad.