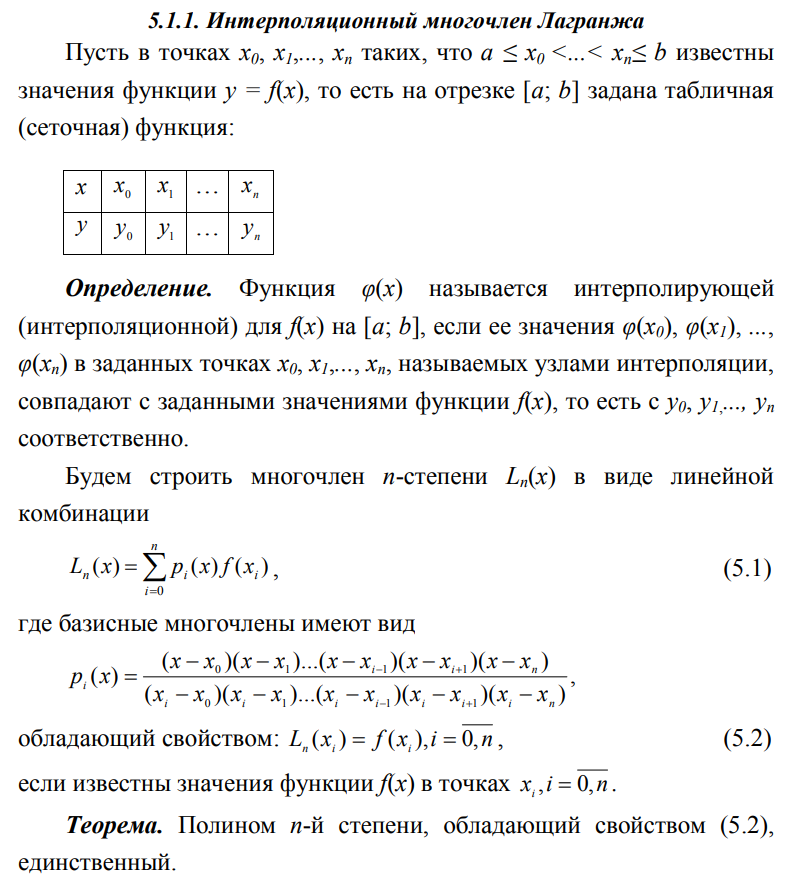
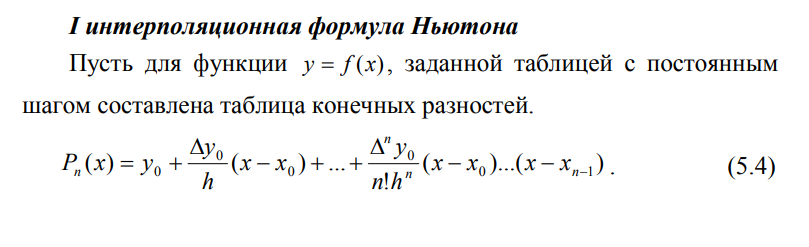
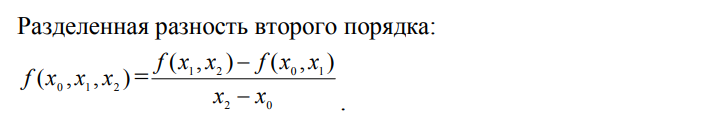
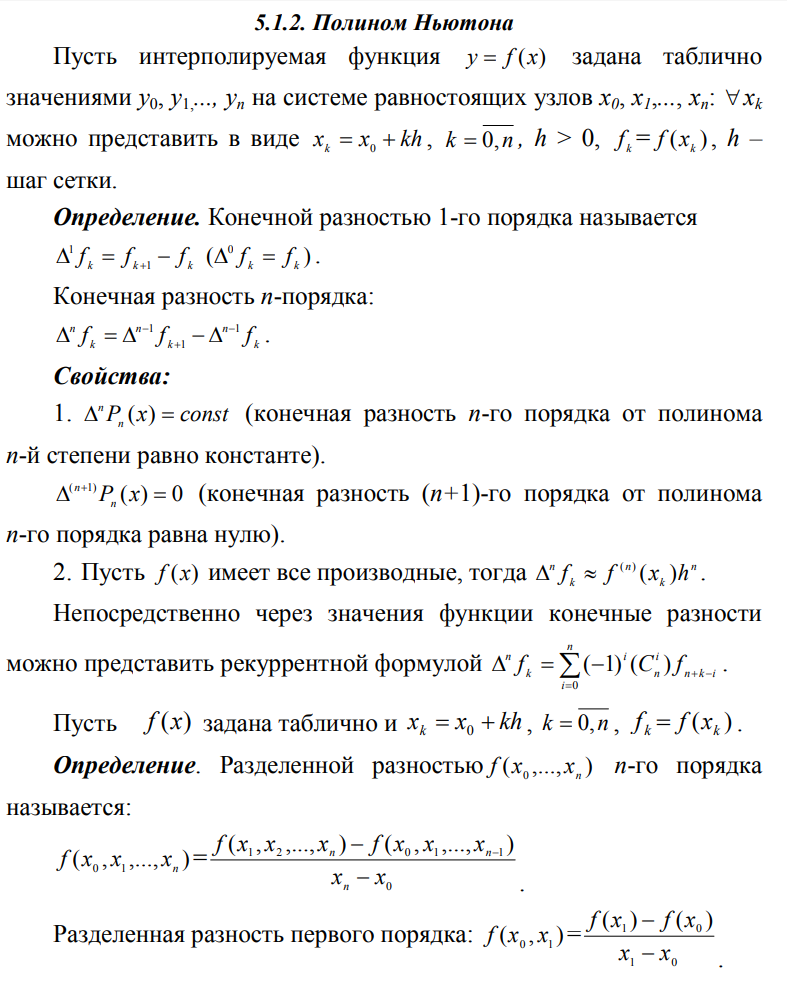
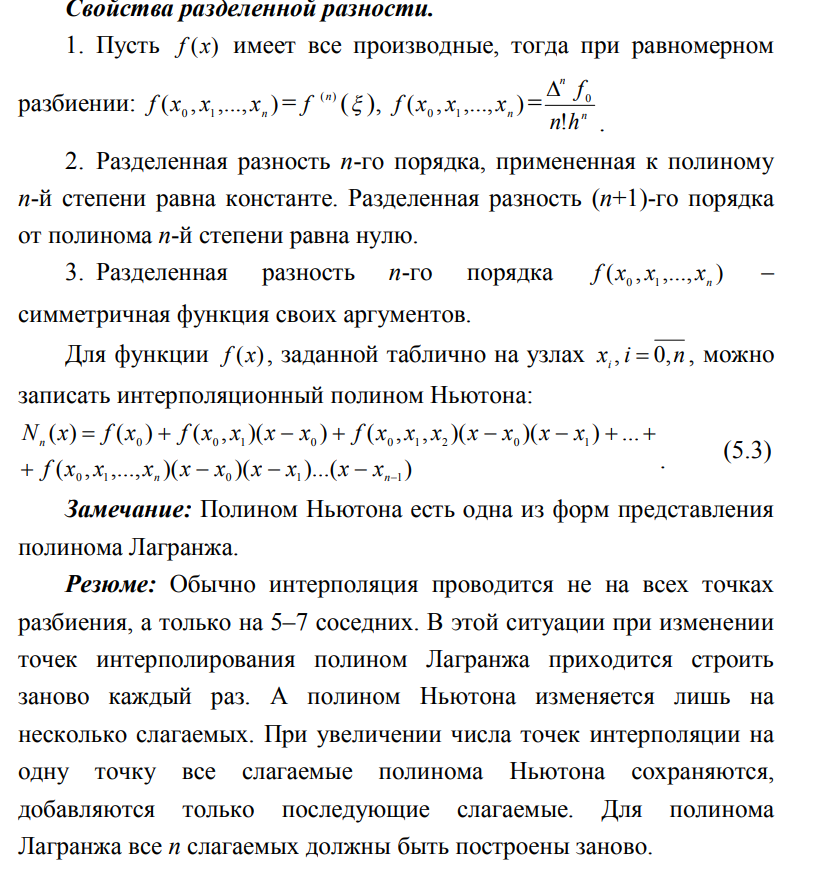
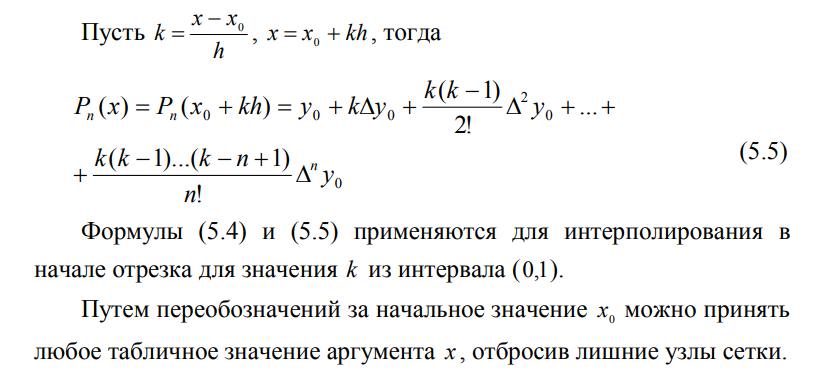
********





**Решение:**

Функция y = f (x) задана таблично в узлах

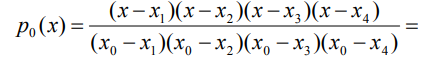
****

1. Построим интерполяционный многочлен Лагранжа 4-й степени L4(x) в виде



линейной комбинации

Вычислим базисные многочлены.



=0,2025

Интерполяционный многочлен Лагранжа 4-й степени будет иметь вид

L4(x) = -0,49) +

+ 3,034) +

+ 0,71) +

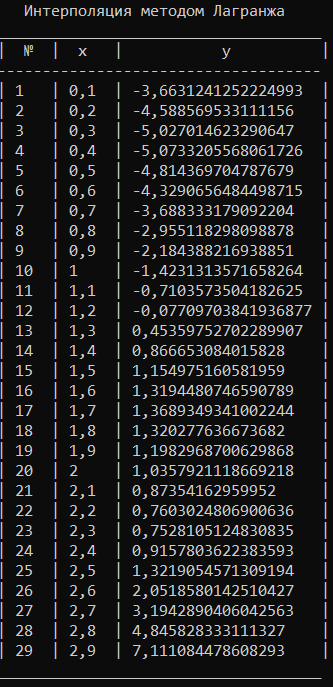
- 0,654) +

+ 0,725)

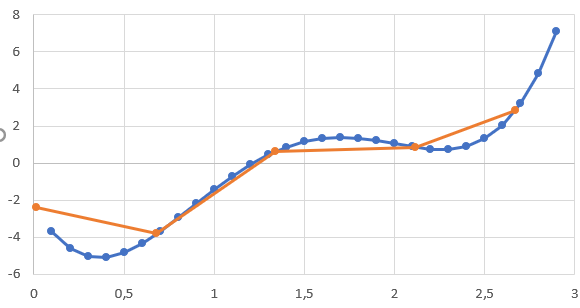
Вычислим значение полинома в точке

N4(x1+х2) = L4(0,681+1,342) = 0,9968100161895291

Вычислим значения L4(x) с шагом 0,1



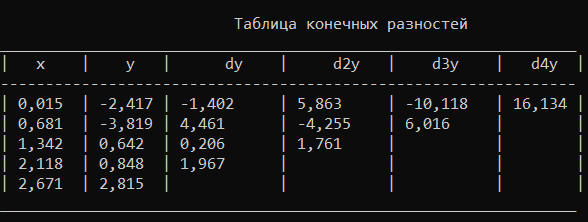
И построим график

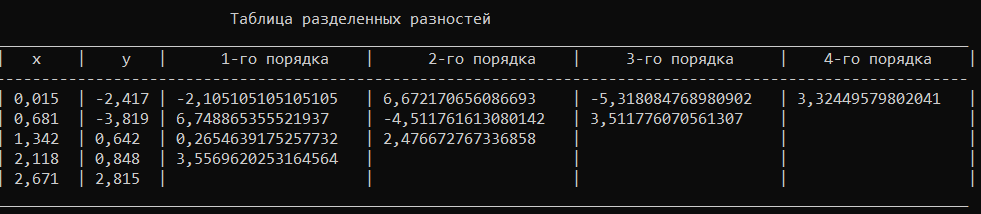


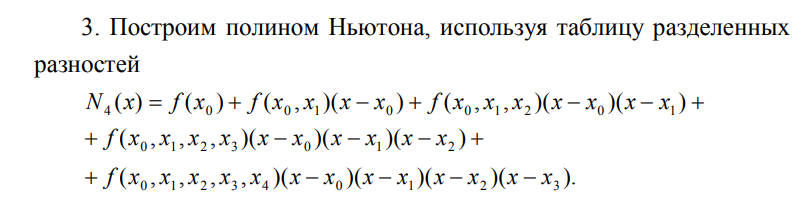
Зеленая линия график по значениям данным в задании

Синяя линия построена по интерполированным значениям

Построим таблицы конечных и разделенных разностей







Получим

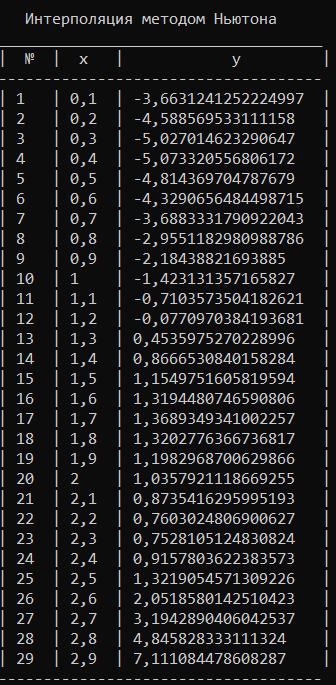
N4(x) = -2,417 – 2,105(x-0,015) + 6,672(x-0,015)(x-0681)

– 5,318(x-0,015)(x-0681)(x-3,315) + 3,324(x-0,015)(x-0681)(x-1,342)(x-2,118)

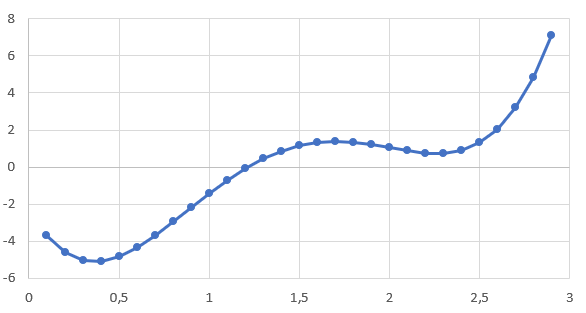
Вычислим значение полинома Ньютона в точке

N4(x1+х2) = L4(0,681+1,342) = 0,9968100161895291

И вычислим значения полинома с шагом 0,1



И построим график по этим значениям



**Код:**

double[,] m = new double[4, 4];

double[,] r = new double[4, 4];

double[] x = { 0.015, 0.681, 1.342, 2.118, 2.671 };

double[] y = { -2.417, -3.819, 0.642, 0.848, 2.815 };

lagrangh(x, y, m, r);

ntn(x, y, m);

void lagrangh(double[] x, double[] y, double[,] m, double[,] r) {

double[] a =

{

y[0] / ((x[0] - x[1]) \* (x[0] - x[2]) \* (x[0] - x[3]) \* (x[0] - x[4])),

y[1] / ((x[1] - x[0]) \* (x[1] - x[2]) \* (x[1] - x[3]) \* (x[1] - x[4])),

y[2] / ((x[2] - x[0]) \* (x[2] - x[1]) \* (x[2] - x[3]) \* (x[2] - x[4])),

y[3] / ((x[3] - x[0]) \* (x[3] - x[1]) \* (x[3] - x[2]) \* (x[3] - x[4])),

y[4] / ((x[4] - x[0]) \* (x[4] - x[1]) \* (x[4] - x[2]) \* (x[4] - x[3])),

};

Console.WriteLine(" Интерполяция методом Лагранжа");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("| № | x | y |");

Console.WriteLine("------------------------------------");

for (var i = 1; i < 30; i++)

{

double z = 3 \* i / 30.0;

double L = (z - x[1]) \* (z - x[2]) \* (z - x[3]) \* (z - x[4]) \* a[0] +

(z - x[0]) \* (z - x[2]) \* (z - x[3]) \* (z - x[4]) \* a[1] +

(z - x[0]) \* (z - x[1]) \* (z - x[3]) \* (z - x[4]) \* a[2] +

(z - x[0]) \* (z - x[1]) \* (z - x[2]) \* (z - x[4]) \* a[3] +

(z - x[0]) \* (z - x[1]) \* (z - x[2]) \* (z - x[3]) \* a[4];

Console.WriteLine("| {0,-3} | {1,-4} | {2,-20} |", i, z, L);

}

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

for (int i = 1; i < 5; i++)

{

r[i-1, 0] = y[i] - y[i - 1];

m[i - 1, 0] = (y[i] - y[i - 1]) / (x[i] - x[i - 1]);

}

for (int i = 2; i < 5; i++)

{

r[i-2, 1] = r[i - 1, 0] - r[i - 2, 0];

m[i-2, 1] = (m[i-1, 0] - m[i - 2, 0]) / (x[i] - x[i - 2]);

}

for (int i = 3; i < 5; i++)

{

r[i - 3, 2] = r[i - 2, 1] - r[i - 3, 1];

m[i - 3, 2] = (m[i - 2, 1] - m[i - 3, 1]) / (x[i] - x[i - 3]);

}

r[0, 3] = r[1, 2] - r[0, 2];

m[0, 3] = (m[1, 2] - m[0, 2]) / (x[4] - x[0]);

Console.WriteLine(" Таблица конечных разностей");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("| x | y | dy | d2y | d3y | d4y |");

Console.WriteLine("----------------------------------------------------------------");

Console.WriteLine("| {0,-6} | {1,-6} | {2,-10} | {3,-9} | {4,-9} | {5,-5} |", x[0], y[0], (float)r[0,0], r[0, 1], (float)r[0, 2], (float)r[0, 3]);

Console.WriteLine("| {0,-6} | {1,-6} | {2,-10} | {3,-9} | {4,-9} | |", x[1], y[1], r[1, 0], (float)r[1, 1], (float)r[1, 2], (float)r[1, 2]);

Console.WriteLine("| {0,-6} | {1,-6} | {2,-10} | {3,-9} | | |", x[2], y[2], (float)r[2, 0], (float)r[2, 1]);

Console.WriteLine("| {0,-6} | {1,-6} | {2,-10} | | | |", x[3], y[3], r[3, 0]);

Console.WriteLine("| {0,-6} | {1,-6} | | | | |", x[4], y[4]);

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Console.WriteLine(" Таблица разделенных разностей");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("| x | y | 1-го порядка | 2-го порядка | 3-го порядка | 4-го порядка |");

Console.WriteLine("------------------------------------------------------------------------------------------------------------");

Console.WriteLine("| {0,-6} | {1,-6} | {2,-20} | {3,-20} | {4,-20} | {5,-18} |", x[0], y[0], m[0, 0], m[0,1], m[0,2], m[0,3]);

Console.WriteLine("| {0,-6} | {1,-6} | {2,-20} | {3,-20} | {4,-20} | |", x[1], y[1], m[1, 0], m[1, 1], m[1, 2], m[1,3]);

Console.WriteLine("| {0,-6} | {1,-6} | {2,-20} | {3,-20} | | |", x[2], y[2], m[2, 0], m[2, 1]);

Console.WriteLine("| {0,-6} | {1,-6} | {2,-20} | | | |", x[3], y[3], m[3, 0]);

Console.WriteLine("| {0,-6} | {1,-6} | | | | |", x[4], y[4]);

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

void ntn(double[] x, double[] y, double[,] m)

{

Console.WriteLine("\n Интерполяция методом Ньютона");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("| № | x | y |");

Console.WriteLine("------------------------------------");

for (var i = 1; i < 30; i++)

{

double z = 3 \* i / 30.0;

double L = y[0] + m[0,0] \* (z - x[0]) + m[0,1] \* (z - x[0]) \* (z - x[1]) + m[0,2] \* (z - x[0]) \* (z - x[1]) \* (z - x[2])+ m[0, 3] \* (z - x[0]) \* (z - x[1]) \* (z - x[2])\* (z - x[3]);

Console.WriteLine("| {0,-3} | {1,-4} | {2,-20} |", i, z, L);

}

Console.WriteLine("------------------------------------");

}

**Вывод:** Метод интерполяции Лагранжа более универсальный, имеет возможность обратной интерполяции, но более трудоемкий по вычислениям чем метод Ньютона, который хоть и легче, но не имеет возможности обратной интерполяции