МИНОБРНАУКИ РФ

Федеральное государственное бюджетное

образовательное учреждение высшего образования  
«Тульский государственный университет»

Институт прикладной математики и компьютерных наук  
Кафедра вычислительной техники

Отчёт по практической работе №3

«Интерполирование с помощью многочленов»

по дисциплине «Численные методы»

Выполнил Проверил

Ст. гр. 220681 Доц. каф. ВТ

Шайхаттаров Д.В.\_\_\_\_\_ Волошко А. Г.\_\_\_\_

Тула 2020

# ЗАДАНИЕ НА РАБОТУ

Получить навык построения интерполяционного многочлена и разработки программных средств для интерполяции функции интерполяционным многочленом Бесселя

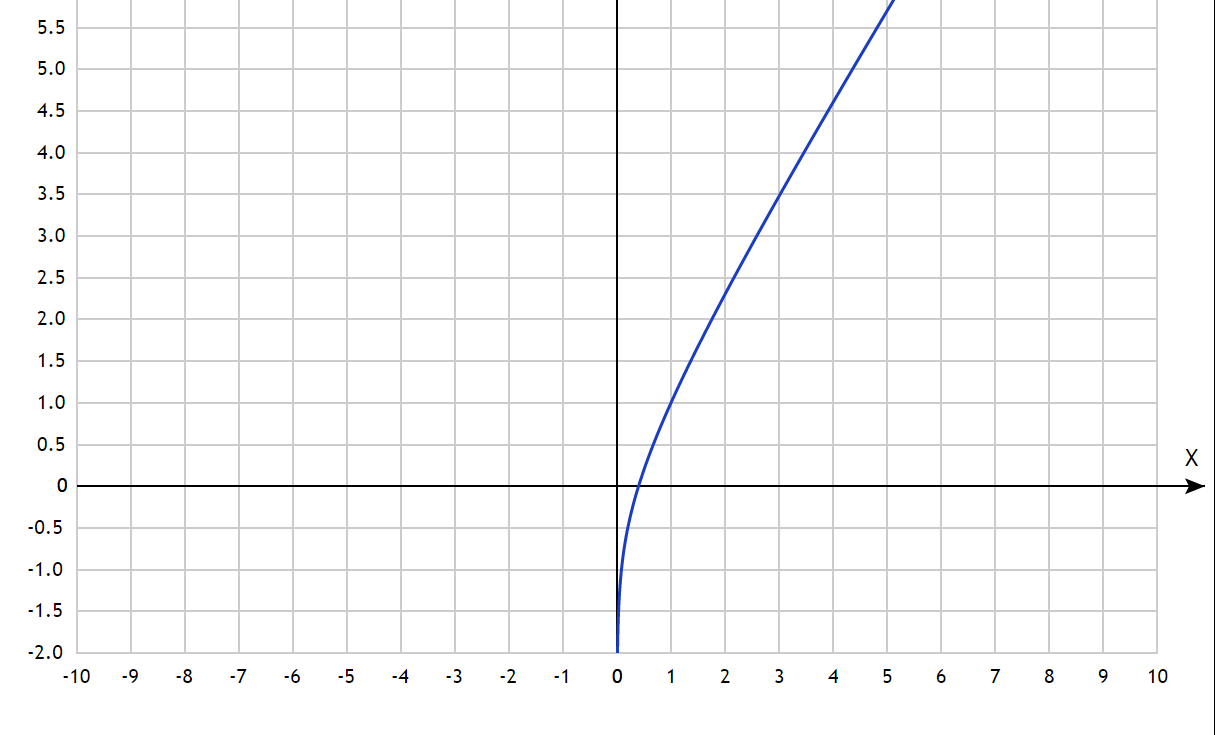
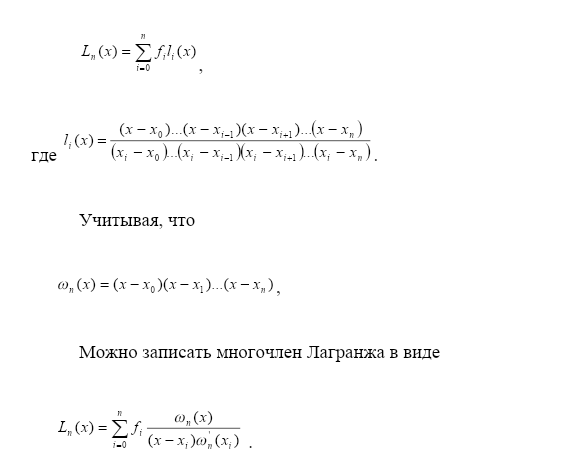


Рисунок 1 –

# МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ МЕТОДА

В интерполяционном многочлене Интерполяционный многочлен Лагранжа с произвольной сеткой узлов.



# ВХОДНЫЕ/ВЫХОДНЫЕ ДАННЫЕ

Входные данные: 2 строки содержащие значения y(x) и x.

Выходные данные: значение корня, строка – полином.

# АЛГОРИТМ РАСЧЕТА

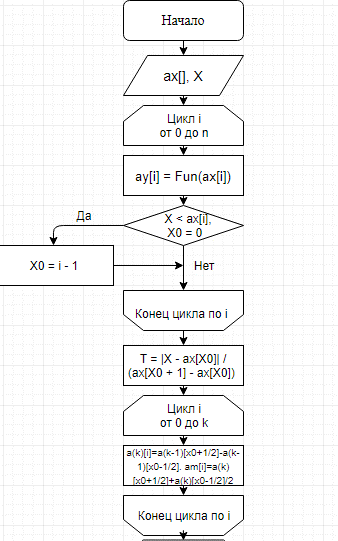


Рисунок 2 - Алгоритм подсчёта значения полинома в Т многочленом Бесселя

# ПРОГРАМНАЯ РЕАЛИЗАЦИЯ

Разработанное ПО получает точки и выводит полином

static string **Polinom**(int k){

string dop = "";

if (k == 0) return "";

dop += "\*t\*";

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

{

dop += "(t-" + i.ToString() +")";

if (i-1 !=k-1)

dop += "\*(t+" + i.ToString() + ")";

if (i != k)

dop += "\*";

}

return dop;

}

static double **PolinomWithT**(int k,double t){

double dop = 0;

if (k == 0) return 1;

dop = t;

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

{

dop \*= (t-i);

if (i - 1 != k - 1)

dop \*= (t+i);

}

return dop;

}

static void Main(string[] args){

Console.WriteLine("Введите точку Х\* (для нецелых запятая)");

double X = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите количество точек Х, что вы хотите ввести");

int n = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите точки Х с переносом строки");

double[] ax = new double[n];

double[] ay = new double[n];

double[] a;

int X0 = 0;

string fun = "y=f(x): ";

for (int k = 0; k < n; k++)

{

ax[k] = Convert.ToDouble(Console.ReadLine());

ay[k] = Fun(ax[k]);

fun += ay[k].ToString() + "; ";

if (X < ax[k] && X0 == 0 && k == 0) X0 = k;

if (X < ax[k] && X0 == 0) X0 = k - 1;

}

Console.WriteLine(fun);

if (X0 + 1 == n - 1) a = new double[2];

else a = new double[4];

double T = Math.Abs(X - ax[X0]) / (ax[X0 + 1] - ax[X0]);

a[0] = (ay[X0 + 1] + ay[X0])/2;

a[1] = ay[X0 + 1] - ay[X0];

if (a.Length == 4)

{

a[2] = (ay[m + 2] - ay[m + 1] - a[1] + a[1] - ay[m] + ay[m - 1]) / 2;

a[3] = ay[m + 2] - ay[m + 1] - a[1] - a[1] + ay[m] - ay[m - 1];

}

string polinom = "";

double dop = 0;

for (int i = 0, k = 0; i+1 < a.Length; i+=2, k++)

{

dop = a[i]/ Fac(i);

if (dop < 0 && polinom!="")

polinom = polinom.Remove(polinom.Length-2,2);

polinom += dop.ToString() + Polinom(k);

dop = a[i+1] / Fac(i + 1);

if (dop > 0)

polinom += " +";

polinom += " " + dop.ToString() + Polinom(k) + "\*(t - 0, 5) ";

if (i + 3 < a.Length)

polinom += "+ ";

}

Console.WriteLine("B{0}(t) = {1}", a.Length-1, polinom);

double polinomT = 0;

for (int i = 0, k = 0; i + 1 < a.Length; i += 2, k++)

{

polinomT += a[i] / Fac(i)\*PolinomWithT(k, T);

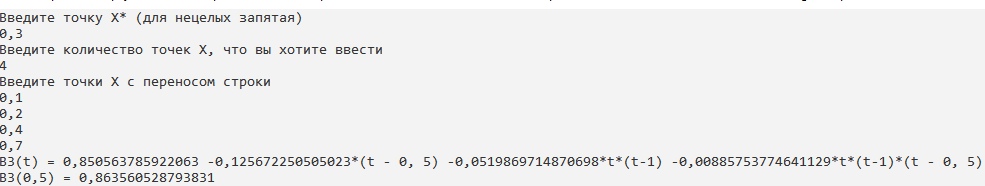
polinomT += a[i + 1] / Fac(i + 1)\* PolinomWithT(k,T)\*(T - 0.5);

}

Console.WriteLine("B{0}({1}) = {2}", a.Length-1, T, polinomT);

# ТЕСТИРОВАНИЕ

Тест 1. Входные данные 0.1 0.2 0.4 0.7 и х\*=0.3 => Ограничен членом 3 степ



Тест 2. Входные данные 10, 11, 15 и х\*=13 => Ограничен членом 1 степени

