МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 2

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Доцент                                                               С.Л. Козенко

должность, уч. Степень, звание           подпись, дата                     инициалы, фамилия

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №3

ИНТЕРПОЛЯЦИЯ

по курсу: ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА



РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. 4134К                                                                             Столяров Н.С.

                                                                        подпись, дата                      инициалы, фамилия

Санкт-Петербург 2023

**Цель работы**

а) освоение методов интерполяции функций;

б) совершенствование навыков по алгоритмизации и программированию вычислительных задач.

**Задание**

Составить схему алгоритма и программу на языке C/C++ решения задачи по теме «Интерполяция» в соответствии с индивидуальным заданием.

**Вариант 5**

Значения x: 1.1 0.7 0.4 0.1 -0.2

Значения y: 2.5 3.7 4.2 2.0 0.0

Метод интерполяции: метод Ньютона

**Описание метода решения**

Для интерполяции функций, заданных таблицами с равноотстоящими значениями аргумента

𝑋𝑖+1−𝑋𝑖=ℎ,где 𝑖=0,1,2…𝑛−1

построение интерполяционных формул и вычисление по формулам заметно упрощается. В записях этих интерполяционных алгоритмов используются разности между значениями функции в соседних узлах интерполяции.

Конечной разностью первого порядка называется

∆𝑦𝑖=(𝑌𝑖+1−𝑌𝑖),где 𝑖=0,1,2…𝑛−1

Из конечных разностей первого порядка образуются конечные разности второго порядка:

∆2𝑦𝑖=∆𝑦𝑖+1−∆𝑦𝑖,где 𝑖=0,1,2…𝑛−1

Аналогично определяются конечные разности третьего, четвёртого и более высоких порядков.

Для функций, заданных таблицами с постоянным шагом изменения аргумента, наиболее часто используются первая или вторая формулы Ньютона, в которых интерполяционная функция определяется как многочлен вида:

𝑃𝑛(𝐼)(𝑥)=𝑎0+𝑎1(𝑥−𝑋0)+𝑎2(𝑥−𝑋0)(𝑥−𝑋1).+⋯+𝑎𝑛(𝑥−𝑋0)…(𝑥−𝑋𝑛−1) (1)

при интерполяции от нулевого узла 𝑋0 или

𝑃𝑛(𝐼𝐼)(𝑥)=𝑏0+𝑏1(𝑥−𝑋𝑛)+𝑏2(𝑥−𝑋𝑛)(𝑥−𝑋𝑛−1).+⋯+𝑏𝑛(𝑥−𝑋𝑛)…(𝑥−𝑋0) (2)

при интерполяции от узла 𝑋𝑛.

Значения коэффициентов 𝑎𝑖 и 𝑏𝑖 в формулах (1) или (2) находятся из условий Лагранжа, определяющих в узлах интерполяции совпадение значений интерполирующей функции со значением табличнозаданной функции

𝑃𝑛(𝑥𝑖)=𝑌𝑖

Полагая 𝑋=𝑋0 , в формуле (1) получим

𝑃𝑛(𝑋0)=𝑎0=𝑌0

Аналогично для

𝑋=𝑋1 𝑃𝑛(𝑋1)=𝑎0+𝑎1(𝑋1−𝑋0)=𝑌1

и далее 𝑎

1=𝑌1−𝑌0𝑋1−𝑋0

или, используя введённые обозначения,

𝑎1= 𝛥𝑦01!ℎ

Продолжая подстановки значений Хi , получим

𝑃𝑛(Х2)=𝑎0+𝑎1(𝑋2−𝑋0)+ 𝑎2 (𝑋2−𝑋0)(𝑋2−𝑋1) =𝑌2

и далее 𝑎2∗2ℎ2=𝑌2−𝑎0−𝑎1∗2ℎ=𝑌2−𝑌0−𝛥𝑦0ℎ∗2ℎ=𝑌2−2𝑌1+ 𝑌0=𝛥2𝑦0

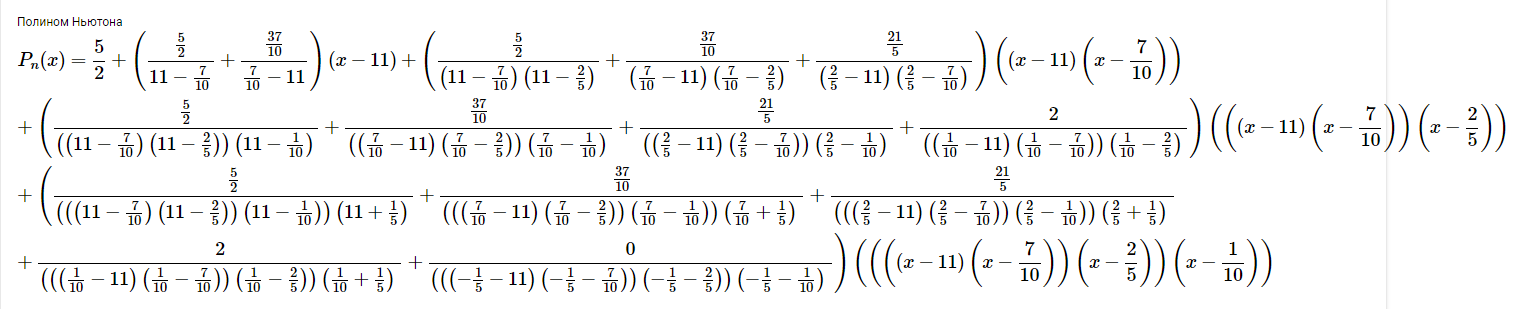
откуда 𝑎2= 𝛥2𝑦02!ℎ2

Проведя аналогичные преобразования для 𝑋=𝑋3 и 𝑋=𝑋4, получим

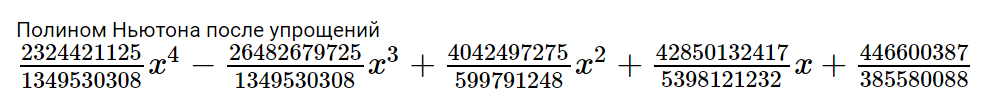
𝑎3= 𝛥3𝑦03!ℎ3, 𝑎4= 𝛥4𝑦04!ℎ4, … , 𝑎𝑘= 𝛥𝑘𝑦0𝑘!ℎ𝑘

Найденные коэффициенты подставляются в формулы (1) или (2) для вычисления значений интерполируемой функции.

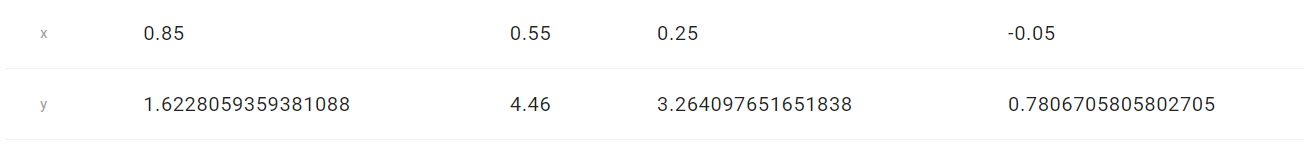
**Аналитические расчеты**

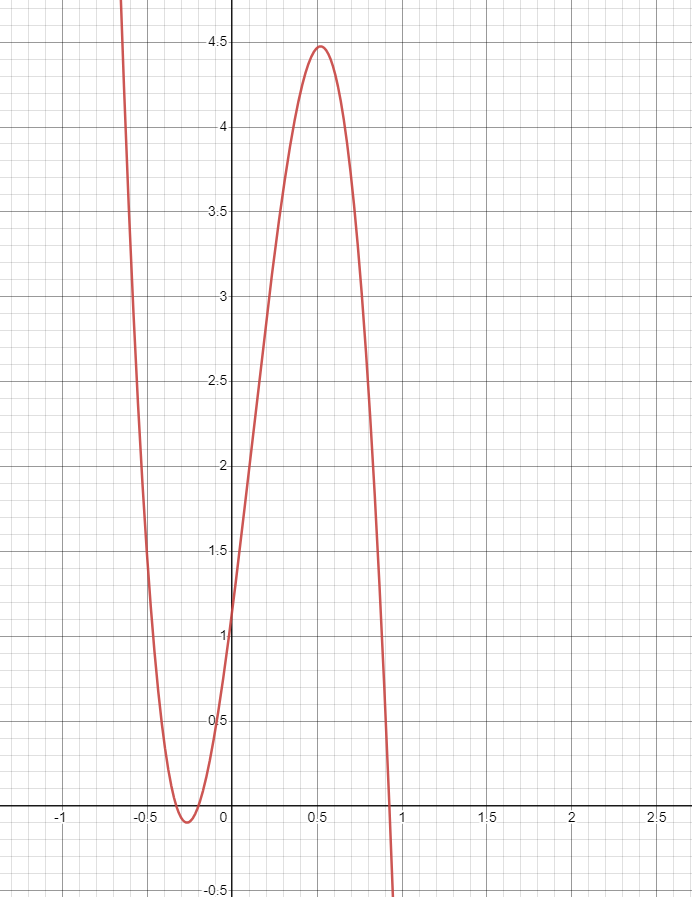
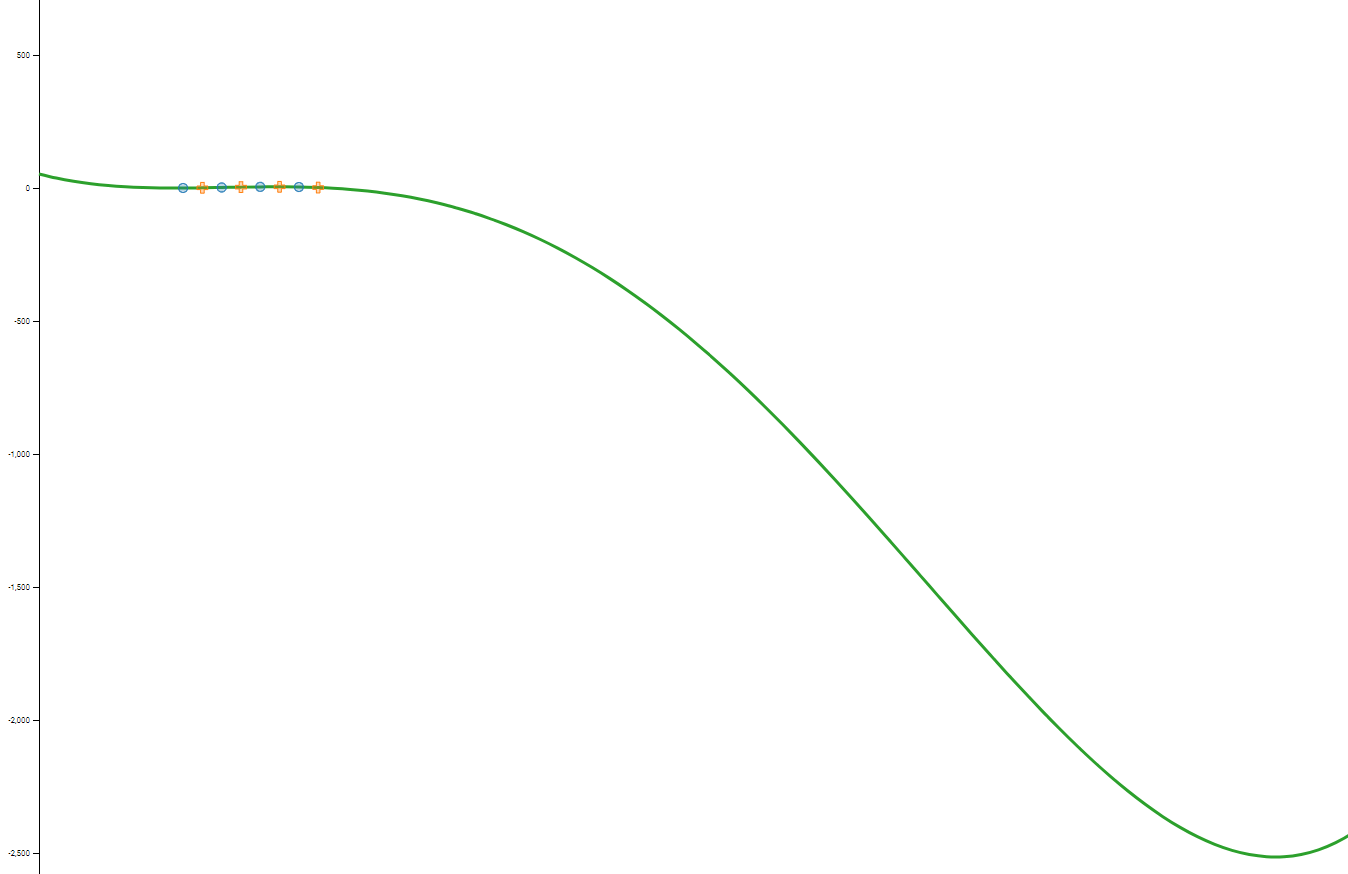


После упрощения

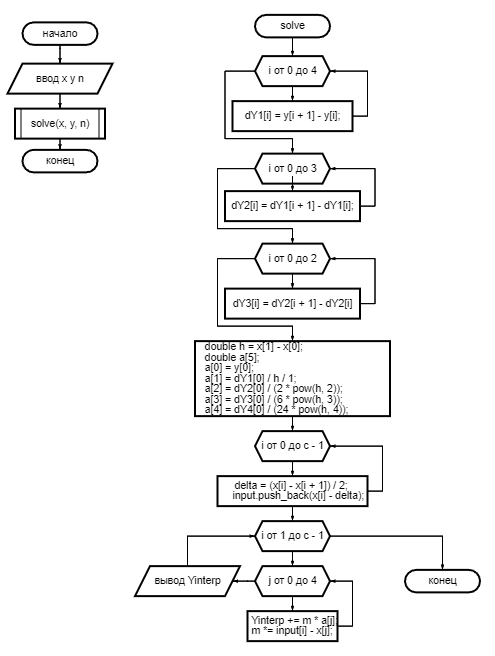


Интерполированные значения



**Схема алгоритма решения задачи**



**Листинг кода программы**

#include <iostream>

#include <vector>

#include <cmath>

**using** **namespace** std**;**

#define AUTO\_INPUT true

void solve**(**double x**[],** double y**[],** int n**)** **{**

//вычисление конечных разностей

double dY1**[**4**];**

double dY2**[**3**];**

double dY3**[**2**];**

double dY4**[**1**];**

**for** **(**int i **=** 0**;** i **<** 4**;** i**++)** **{**

dY1**[**i**]** **=** y**[**i **+** 1**]** **-** y**[**i**];**

**}**

**for** **(**int i **=** 0**;** i **<** 3**;** i**++)** **{**

dY2**[**i**]** **=** dY1**[**i **+** 1**]** **-** dY1**[**i**];**

**}**

**for** **(**int i **=** 0**;** i **<** 2**;** i**++)** **{**

dY3**[**i**]** **=** dY2**[**i **+** 1**]** **-** dY2**[**i**];**

**}**

dY4**[**0**]** **=** dY3**[**1**]** **-** dY3**[**0**];**

//вычисление значений коэффициента полинома

double h **=** x**[**1**]** **-** x**[**0**];**

double a**[**5**];**

a**[**0**]** **=** y**[**0**];**

a**[**1**]** **=** dY1**[**0**]** **/** h **/** 1**;**

a**[**2**]** **=** dY2**[**0**]** **/** **(**2 **\*** pow**(**h**,** 2**));**

a**[**3**]** **=** dY3**[**0**]** **/** **(**6 **\*** pow**(**h**,** 3**));**

a**[**4**]** **=** dY4**[**0**]** **/** **(**24 **\*** pow**(**h**,** 4**));**

int c**;**

double iX**,** Yinterp**,** m**;**

vector **<**double**>** input**;**

input**.**clear**();**

double delta **=** 0**;**

c **=** n**;**

**for** **(**int i **=** 0**;** i **<** c **-** 1**;** i**++)** **{**

delta **=** **(**x**[**i**]** **-** x**[**i **+** 1**])** **/** 2**;**

input**.**push\_back**(**x**[**i**]** **-** delta**);**

**}**

// cout << "Введите количество дополнительных точек: ";

// cin >> c;

// for (int i = 0; i < c; i++) {

// cout << "x[" << i << "]= ";

// cin >> iX;

// input.push\_back(iX);

// }

cout << endl;

//вычисления y по первой формуле ньютона

for (int i = 0; i < c - 1; i++) {

m = 1;

Yinterp = 0;

for (int j = 0; j < 5; j++) {

Yinterp += m \* a[j];

m \*= input[i] - x[j];

}

cout << "Yinterp[" << input.at(i) << "]= " << Yinterp << endl;

}

}

int main() {

system("chcp 65001");

int n;

double \*x;

double \*y;

if (AUTO\_INPUT) {

n = 5;

x = (double\*)malloc(n \* sizeof(double));

y = (double\*)malloc(n \* sizeof(double));

x[0] = 1.0;

x[1] = 0.7;

x[2] = 0.4;

x[3] = 0.1;

x[4] = -0.2;

y[0] = 2.5;

y[1] = 3.7;

y[2] = 4.2;

y[3] = 2.0;

y[4] = 0.0;

cout << "x = { ";

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

cout << x[i] << " ";

cout << "}\ny = { ";

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

cout << y[i] << " ";

cout << "}";

} else {

cout << "количество уловых точек: ";

cin >> n;

x = (double\*)malloc(n \* sizeof(double));

y = (double\*)malloc(n \* sizeof(double));

for (int i = 0; i < n; i++) {

cout << "x[" << i << "] = ";

cin >> x[i];

cout << "y[" << i << "] = ";

cin >> y[i];

cout << endl;

}

}

solve(x, y, n);

return 0;

}

**Результат работы программы**

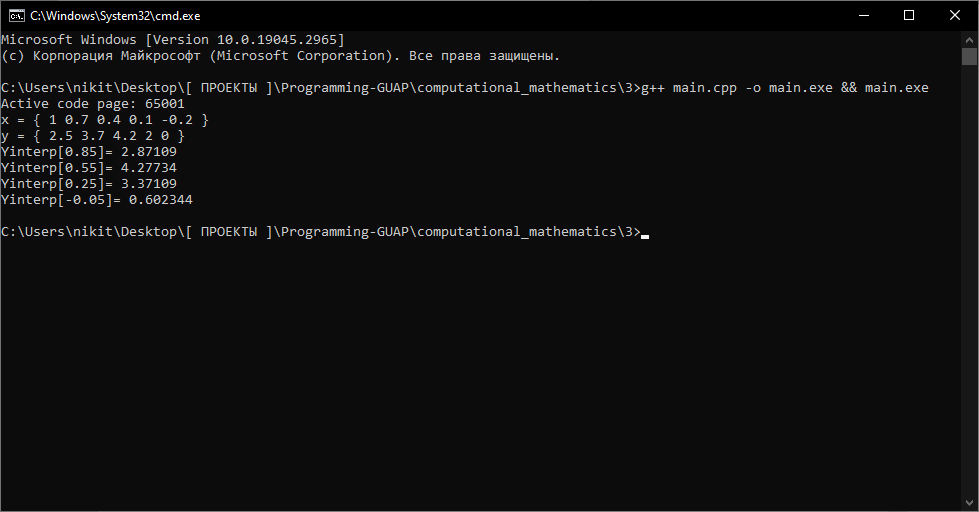


Рисунок 16 – Результат работы программы

**Сравнение результатов аналитического и программного расчета**

Таблица 1 – Сравнение результатов аналитического и программного расчета

|  |  |  |
| --- | --- | --- |
|  | аналитический | программный |
| x=0 | 2.79375 | 2.79375 |
| x=0.2 | 1.58125 | 1.58125 |
| x=0.4 | 0.91875 | 0.91875 |
| x=0.6 | 0.50625 | 0.50625 |
| x=0.8 | 0.04375 | 0.04375 |
| x=1 | -0.76875 | -0.76875 |
| x=1.2 | -2.23125 | -2.23125 |
| x=1.4 | -4.64375 | -4.64375 |

Исходя из данных таблицы 1 (все данные совпадают), мы можем сделать вывод, что программа работает верно.

**Вывод**

В ходе выполнения практической работы №3 была написана программа, которая решает задачу интерполяции функции используя при этом метод Ньютона. Результат программного расчета при одинаковых входных данных полностью совпал с результатом аналитического расчета, что говорит о том, что программа работает корректно и выдает правильные результаты.