**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Лабораторная №3

**«Метод наименьших квадратов»**

**Выполнил:**

Крученков Евгений Андреевич

студент 2 курса 9 группы,

специальность

“прикладная математика”

**Преподаватель:**

Ассистент кафедры вычислительной

математики ФПМИ,

Ю.Н. Горбачёва

Минск, 2022

**Содержание:**

Постановка задачи ------------------------------------------------------------------ 2

Краткие теоретические сведения ------------------------------------------------ 2-3

Листинг программы ---------------------------------------------------------------- 3-9

Результаты --------------------------------------------------------------------------- 9-11

**Постановка задачи**

На отрезке [a, b] задана таблица значений функции f(x) с шагом h = 0,1. По заданной таблице значений найти наилучшие среднеквадратичные приближения при n = 4, 6.

Найти

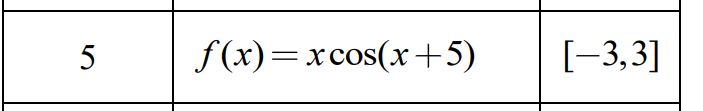
В содержание отчета должна быть включена следующая информация:

• Метод наименьших квадратов.

• Графики аппроксимирующих функций и график заданной функции (по множеству точек).

• .

• Листинг программы с комментариями.

****

**Краткие теоретические сведения**

**Метод наименьших квадратов**

Заданы N+1 узел и значения в них:

Решая систему, найдем коэффициенты многочлена наилучшей аппроксимации:

…

**Листинг программы**

**Header.h**

#pragma once

#include <iostream>

#include <stdio.h>

#include <iomanip>

#include <fstream>

#include <string>

#include <time.h>

using namespace std;

double E = pow(10, -6);

// Гаусс

class Matr

{

friend class Sist;

double\*\* M;

int n;

public:

Matr()

{

this->n = 1;

M = new double\* [n];

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

{

M[i] = new double[n];

}

srand(time(0));

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

{

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

{

M[i][j] = (((double)rand() / RAND\_MAX) \* 2 - 1) \* 100;

}

}

}

Matr(int n, double\*\* M1)

{

this->n = n;

M = new double\* [n];

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

{

M[i] = new double[n];

}

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

{

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

{

M[i][j] = M1[i][j];

}

}

}

Matr(const Matr& B)

{

this->n = B.n;

M = new double\* [n];

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

{

M[i] = new double[n];

}

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

{

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

{

M[i][j] = B.M[i][j];

}

}

}

~Matr()

{

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

{

delete[] M[i];

}

delete[] M;

}

friend ostream& operator <<(ostream& os, Matr& A)

{

os << fixed << showpoint << setprecision(3);

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

{

for (int j = 0; j < A.n; j++)

{

os << A.M[i][j] << " ";

}

os << endl;

}

return os;

}

Matr& operator = (const Matr& B)

{

if (this == &B)

return \*this;

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

{

delete[] M[i];

}

n = B.n;

delete[] M;

M = new double\* [n];

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

{

M[i] = new double[n];

}

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

{

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

{

M[i][j] = B.M[i][j];

}

}

return \*this;

}

};

class Sist

{

friend class Matr;

private:

Matr A;

double\* X;

double\* F;

double\* X\_;

double pogr;

public:

int size()

{

return A.n;

}

void ShowPogr()

{

cout << scientific << pogr << fixed << showpoint << setprecision(3);

}

Sist(const Matr& B, double\* F1)

{

A = B;

X = new double[B.n];

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

{

X[i] = 1 + i;

}

F = F1;

X\_ = new double[B.n];

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

{

X\_[i] = 0;

}

}

~Sist()

{

delete[] X;

delete[] X\_;

delete[] F;

}

double\* Gauss()

{

//выбор главного элемента по столбцу

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

{

double max = abs(A.M[i][i]);

int gl = i;

for (int j = i; j < size(); j++)

{

if (abs(A.M[j][i]) > max)

{

max = A.M[j][i];

gl = j;

}

}

if (max == 0) throw "Все главные элементы нулевые";

//исключение в случае, когда все главные элементы нулевые

for (int j = i; j < size(); j++)

{

swap(A.M[i][j], A.M[gl][j]);

}

swap(F[i], F[gl]);

//прямой ход метода Гаусса

for (int j = i + 1; j < size(); j++)

{

A.M[i][j] /= A.M[i][i];

}

F[i] /= A.M[i][i];

A.M[i][i] /= A.M[i][i];

for (int j = i + 1; j < size(); j++)

{

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

{

A.M[j][k] -= A.M[i][k] \* A.M[j][i];

}

F[j] -= F[i] \* A.M[j][i];

A.M[j][i] -= A.M[i][i] \* A.M[j][i];

}

}

//обратный ход метода Гаусса

for (int i = (size() - 1); i >= 0; i--)

{

X\_[i] = F[i];

for (int j = size() - 1; j > i; j--)

{

X\_[i] -= A.M[i][j] \* X\_[j];

}

}

//вычисление относительной погрешности

double maxX = abs(X[0]), max\_X = abs(X[0] - X\_[0]);

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

{

if (max\_X < abs(X[i] - X\_[i]))

{

max\_X = abs(X[i] - X\_[i]);

}

if (maxX < abs(X[i]))

{

maxX = abs(X[i]);

}

}

pogr = max\_X / maxX;

return X\_;

}

};

double F(double x) {

return (x\*cos(x+5));

}

void MNK() {

int N = 60, N1 = 5, N2 = 7;

double\* X\_k = new double[N+1];

double\* F\_k = new double[N+1];

double x = -3;

//cout << "Таблица значений:" << endl << "X\_k" << " " << "F\_k" << endl;

for(int i = 0; i < N+1; x = x + 0.1, i++) {

X\_k[i] = x;

F\_k[i] = F(x);

//cout <<X\_k[i]<<" "<< endl;

}

double\* S1 = new double[N1\*2];

double\* M1 = new double[N1];

double\* S2 = new double[N2 \* 2];

double\* M2 = new double[N2];

//4

for (int i = 0; i < N1; i++) {

M1[i] = 0;

for (int j = 0; j < N + 1; j++) {

M1[i] += F\_k[j] \* pow(X\_k[j], i);

}

}

for (int i = 0; i < N1\*2; i++) {

S1[i] = 0;

for (int j = 0; j < N + 1; j++) {

S1[i] += pow(X\_k[j], i);

}

}

//6

for (int i = 0; i < N2; i++) {

M2[i] = 0;

for (int j = 0; j < N + 1; j++) {

M2[i] += F\_k[j] \* pow(X\_k[j], i);

}

}

for (int i = 0; i < N2 \* 2; i++) {

S2[i] = 0;

for (int j = 0; j < N + 1; j++) {

S2[i] += pow(X\_k[j], i);

}

}

double\*\* Matr1 = new double\* [N1];

for (int i = 0; i < N1; i++) {

Matr1[i] = new double[N1];

}

for (int i = 0; i < N1; i++) {

for (int j = 0; j < N1; j++) {

Matr1[i][j] = S1[i + j];

//cout << i + j << " ";

}

//cout << endl;

}

double\*\* Matr2 = new double\* [N2];

for (int i = 0; i < N2; i++) {

Matr2[i] = new double[N2];

}

for (int i = 0; i < N2; i++) {

for (int j = 0; j < N2; j++) {

Matr2[i][j] = S2[i + j];

//cout << i + j << " ";

}

//cout << endl;

}

Matr Mt1(N1, Matr1);

Matr Mt2(N2, Matr2);

Sist Sist1(Mt1, M1);

Sist Sist2(Mt2, M2);

double\* Rez1 = Sist1.Gauss();

double\* Rez2 = Sist2.Gauss();

cout << "Коэффициенкты аппроксимирующего многочлена 4 степени:" << endl;

for (int i = 0; i < N1; i++) {

cout << "c"<<i<<" "<< Rez1[i] << endl;

}

cout << "Наилучшее среднеквадратичное приближение для аппроксимирующего многочлена 4 степени:" << endl;

double delta1 = 0;

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

double fi\_k = 0;

for (int j = 0; j < N1; j++) {

fi\_k += Rez1[j] \* pow(X\_k[i], j);

}

delta1 += pow((F\_k[i] - fi\_k), 2);

}

cout << delta1 << endl;

cout << endl;

cout << "Коэффициенкты аппроксимирующего многочлена 6 степени:" << endl;

for (int i = 0; i < N2; i++) {

cout << "c" << i << " " << Rez2[i] << endl;;

}

cout << "Наилучшее среднеквадратичное приближение для аппроксимирующего многочлена 6 степени:" << endl;

double delta2 = 0;

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

double fi\_k = 0;

for (int j = 0; j < N2; j++) {

fi\_k += Rez2[j] \* pow(X\_k[i], j);

}

delta2 += pow((F\_k[i] - fi\_k), 2);

}

cout << delta2 << endl;

cout << endl;

}

**Main.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include "Header.h"

#include <fstream>

#include<time.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Метод наименьших квадратов:" << endl;

MNK();

system("pause");

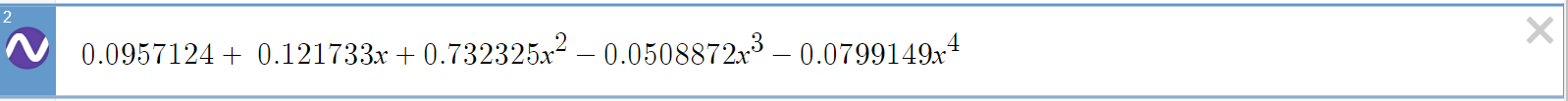
return 0;

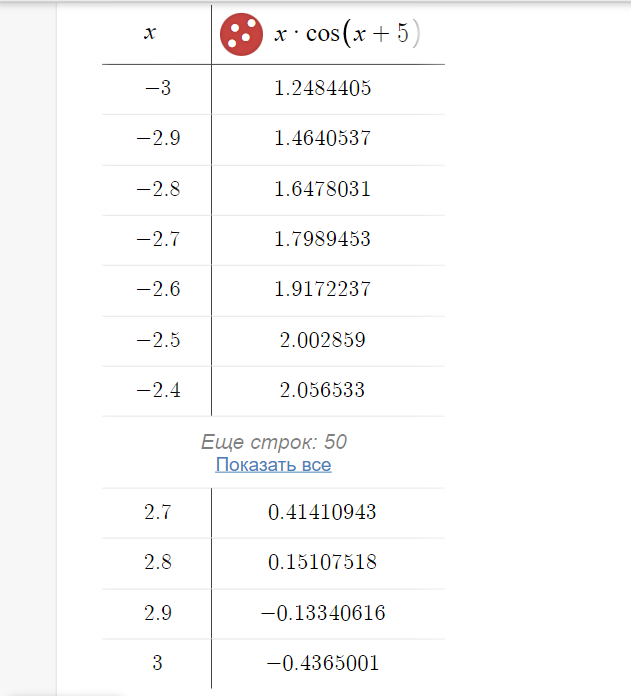
}

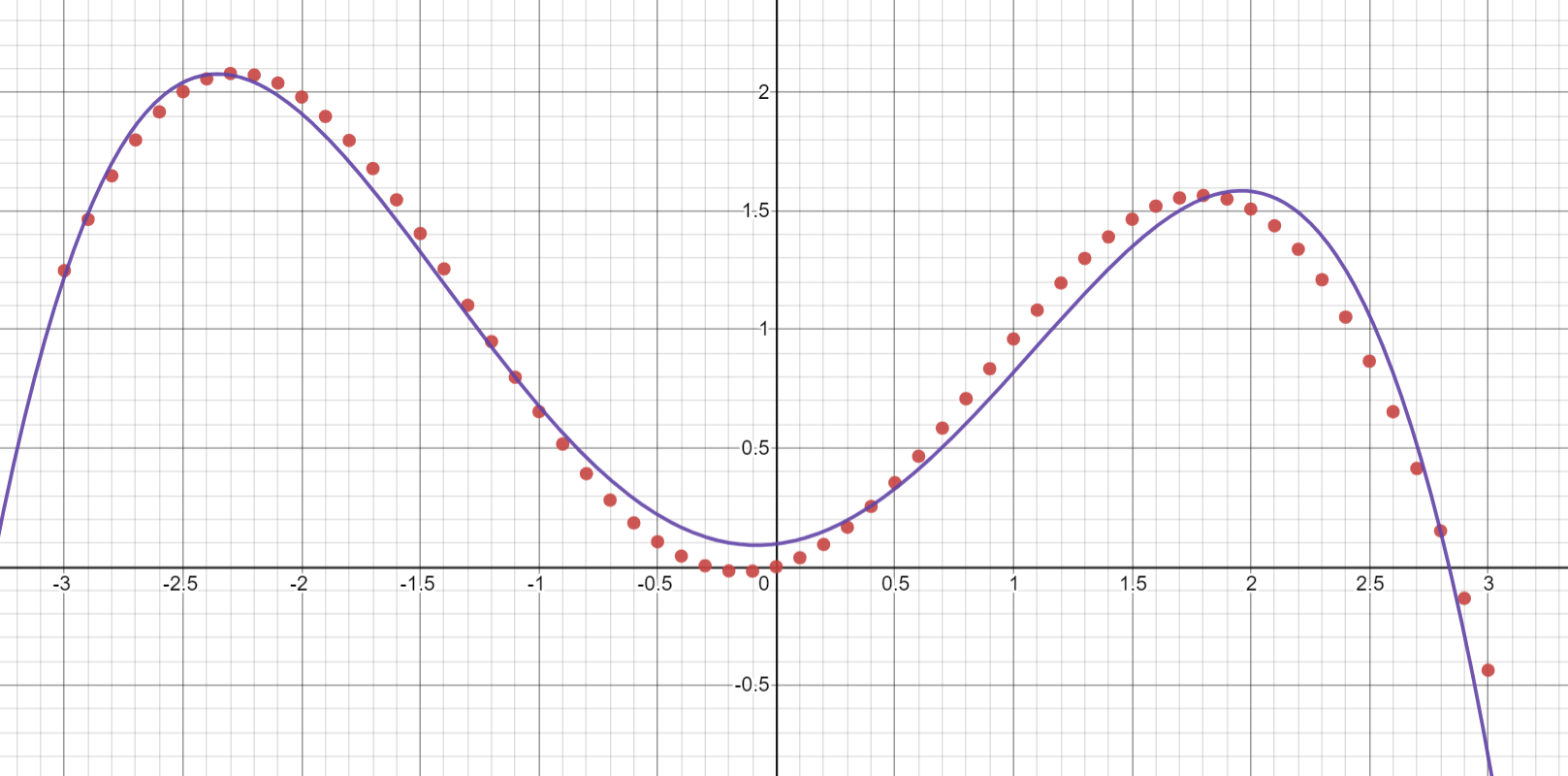
**Результаты**

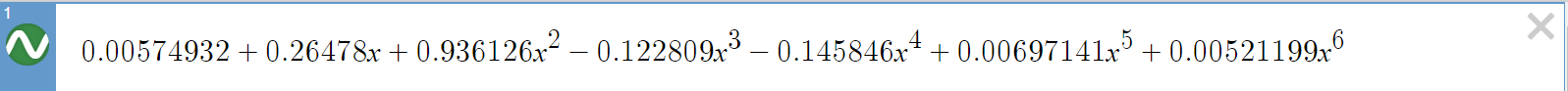
*Графики функций:*

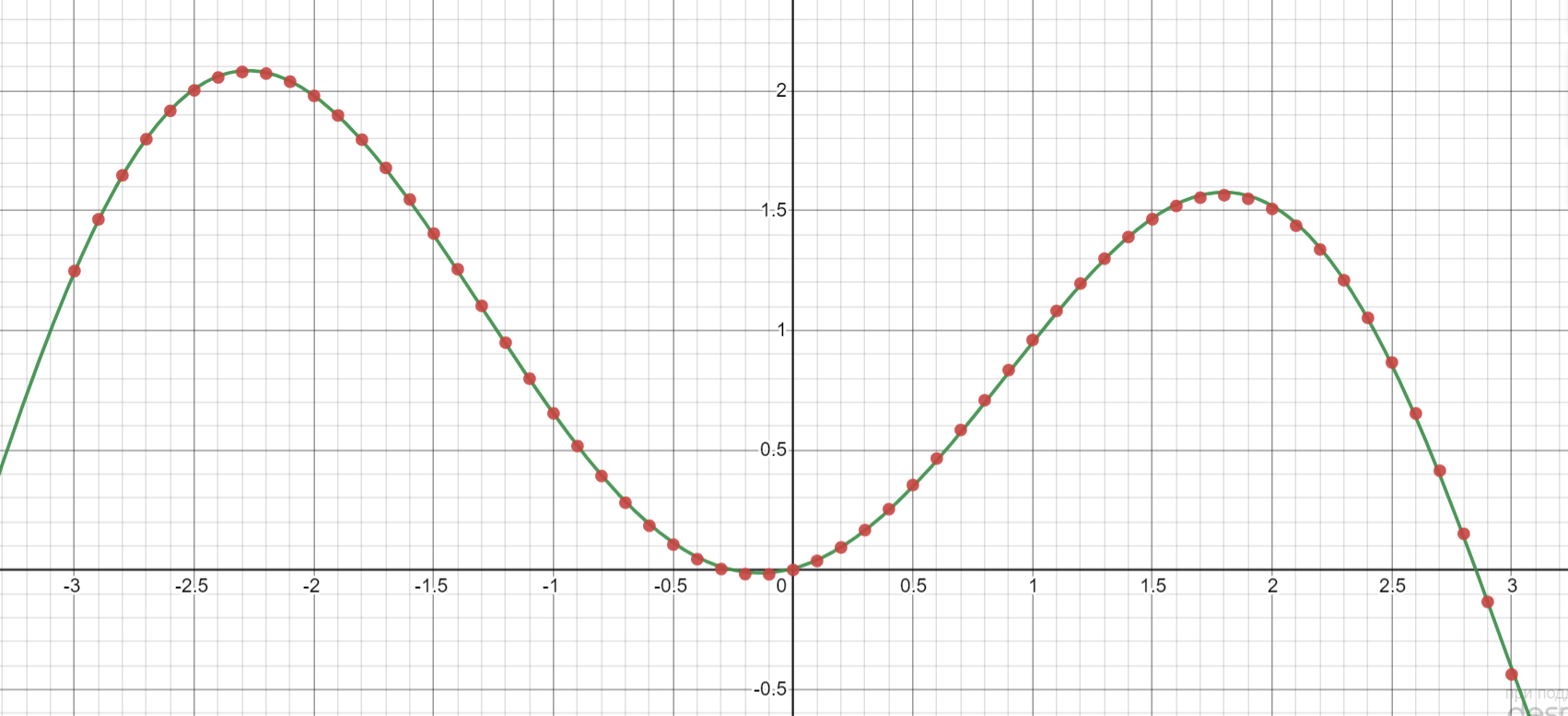
1. *Сравнение с графиком функции, построенной поточечно:*



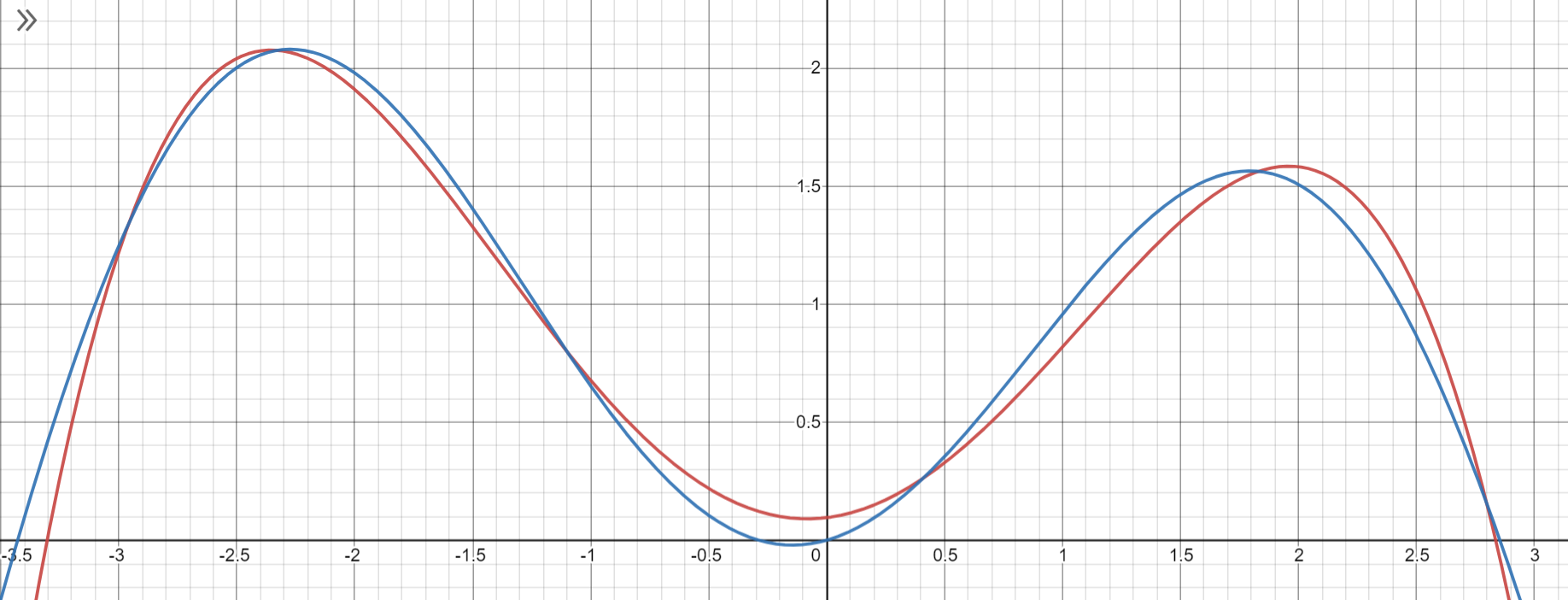


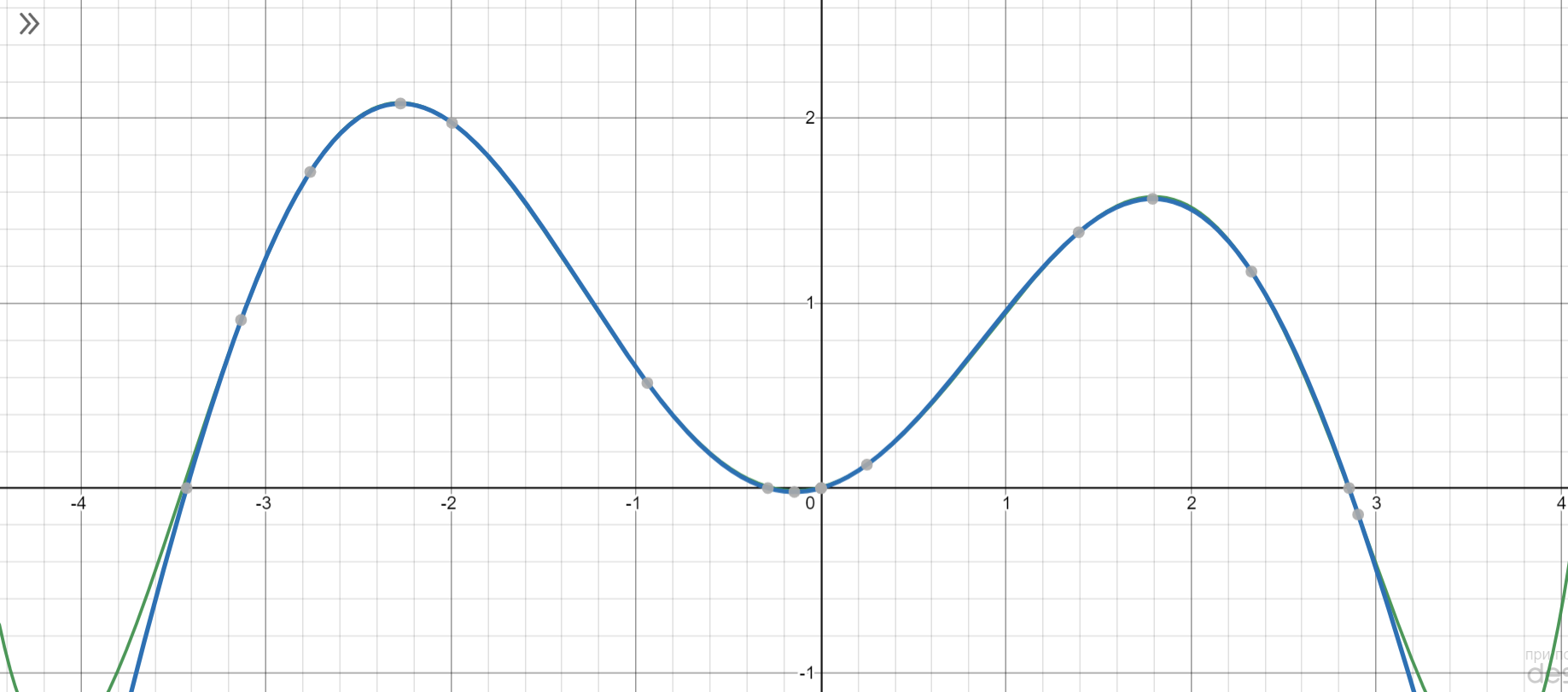




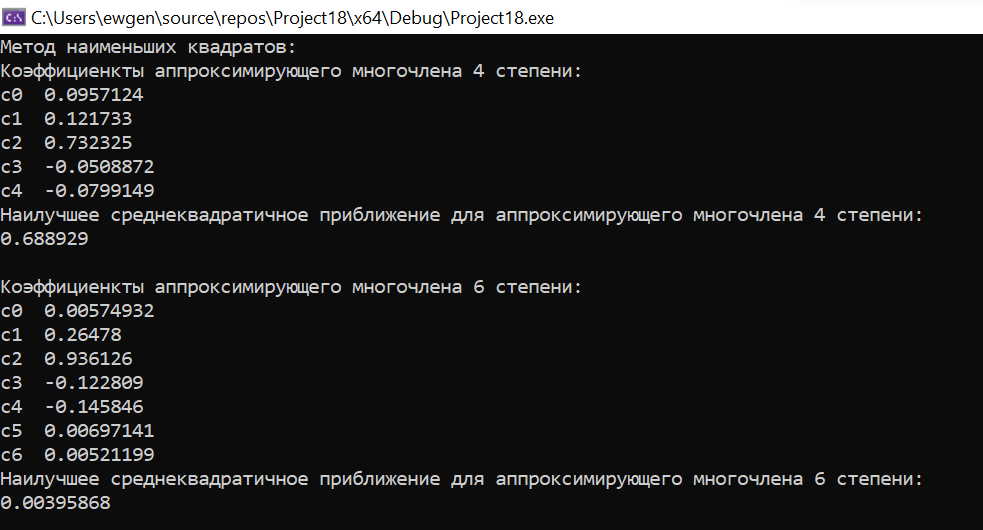


1. *Сравнение с графиком функции:*





1. *Вывод программы:*

**