**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Лабораторная №3

**«Интерполяция алгебраическими многочленами»**

**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 ------------------------------------------------------------------ 2-3

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

Листинг программы ---------------------------------------------------------------- 3-12

Результаты --------------------------------------------------------------------------- 12-20

**Постановка задачи**

На отрезке [a, b] заданы функции f1(x) и f2(x). Постройте многочлены степени n = 5, 10, 15, 20, интерполирующие каждую из них

а) на сетке равноотстоящих узлов;

б) на сетке чебышёвских узлов.

Постройте графики функции f1(x) (f2(x) и интерполяционных многочленов для каждого n. Оцените погрешность интерполирования в узлах сетки x = a + i(b-a)/n, i = 0, …, 100 . Сравните полученные результаты. Сделайте выводы о сходимости интерполяционного процесса по равноотстоящим и чебышёвским узлам. В содержание отчета должна быть включена следующая информация:

• Способ выбора узлов.

• Представление, использованное при построении интерполяционных многочленов.

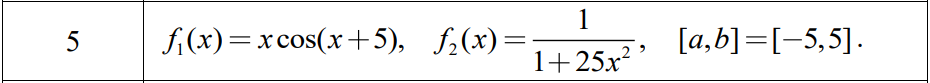
• Графики функции f1(x) и интерполяционных многочленов по равноотстоящим узлам. Графики функции f1(x) и интерполяционных многочленов по чебышёвским узлам.

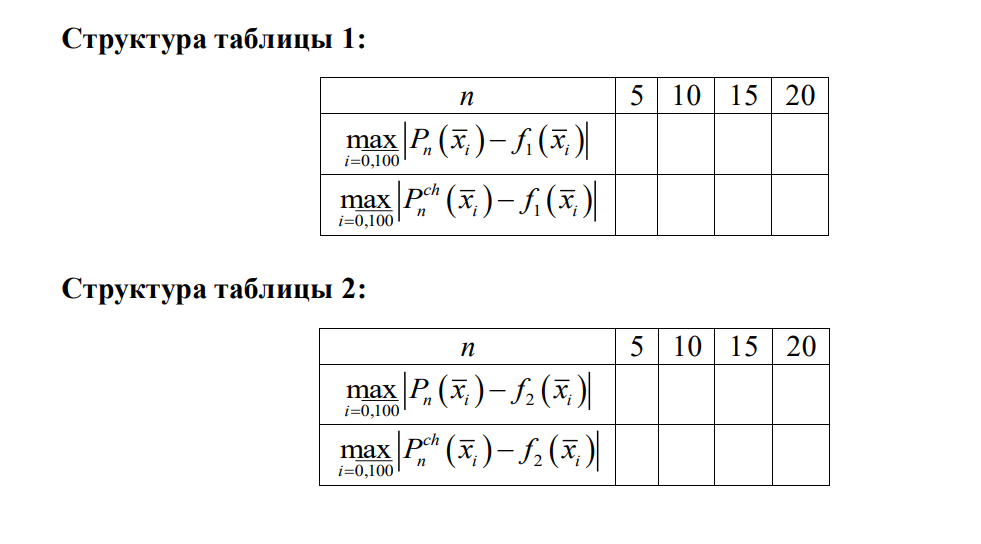
• Графики функции f2(x) и интерполяционных многочленов по равноотстоящим узлам. Графики функции f2(x) и интерполяционных многочленов по чебышёвским узлам.

• Оценка погрешности интерполирования функции f1(x), оформленная в виде таблицы 1 (см. ниже).

• Оценка погрешности интерполирования функции f2(x), оформленная в виде таблицы 2 (см. ниже).

• Листинг программы с комментариями

****

****

**Краткие теоретические сведения**

Равноотстоящие узлы:

Чебышёвские узлы:

Интерполяционные многочлены построим методом Лагранжа:

**Листинг программы**

**Header.h**

#pragma once

#define \_USE\_MATH\_DEFINES

#include<cmath>

#include <iostream>

#include <stdio.h>

#include <iomanip>

#include <fstream>

#include <string>

#include <time.h>

using namespace std;

double E = pow(10, -6);

double PI = acos(-1.0);

// 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 F1(double x) {

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

}

double F2(double x) {

return (1 / (1 + 25 \* x \* x));

}

//double Cheb(double x) {

// return (x/5);

//}

void Cheb1(int N) {

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

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

double a = -5, b = 5;

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

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

X\_k[i] = (a + b) / 2 + ((b - a) / 2) \* cos(((2 \* i + 1) \* PI) / (2 \* (N + 1)));

F\_k[i] = F1(X\_k[i]);

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

}

double\* Rez1 = new double[N + 1];

cout << "Интерполяционный многочлен " << N << " степени для функции f1:" << endl;

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

Rez1[i] = F\_k[i];

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

if (i != j) {

Rez1[i] /= (X\_k[i] - X\_k[j]);

}

}

}

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

if (Rez1[i] < 0) cout << fixed << showpoint << setprecision(20) << Rez1[i];

else cout << "+" << Rez1[i];

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

if (i != j) {

if (X\_k[j] < 0) cout << "(x+" << abs(X\_k[j]) << ")";

else cout << "(x-" << X\_k[j] << ")";

}

}

}\*/

/\*cout <<endl<< "Оценка погрешности:" <<endl;

double max = 0;

for (int i = 0; i < 100; i++) {

int x\_k = a + i \* (b - a) / 100;

double Pn\_xk = 0;

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

Pn\_xk += Rez1[j] \* pow(x\_k, j);

}

if (abs(F1(x\_k) - Pn\_xk) > max) max = abs(F1(x\_k) - Pn\_xk);

}

cout << max << endl;

cout << endl;\*/

cout << "Оценка погрешности:" << endl;

double max = 0;

for (int k = 0; k < 100; k++) {

double x\_k = a + k \* (b - a) / 100;

double summ = 0;

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

double el = Rez1[i];

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

if (i != j) {

el \*= (x\_k - X\_k[j]);

}

}

summ += el;

}

if (abs(summ-F1(x\_k)) > max) max = abs(summ-F1(x\_k));

}

cout << max << endl;

}

void Cheb2(int N) {

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

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

double a = -5, b = 5;

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

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

X\_k[i] = (a + b) / 2 + ((b - a) / 2) \* cos(((2 \* i + 1) \* PI) / (2 \* (N + 1)));

F\_k[i] = F2(X\_k[i]);

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

}

double\* Rez1 = new double[N + 1];

cout << "Интерполяционный многочлен " << N << " степени для функции f2:" << endl;

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

Rez1[i] = F\_k[i];

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

if (i != j) {

Rez1[i] /= (X\_k[i] - X\_k[j]);

}

}

}

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

if (Rez1[i] < 0) cout << fixed << showpoint << setprecision(20) << Rez1[i];

else cout << "+" << Rez1[i];

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

if (i != j) {

if (X\_k[j] < 0) cout << "(x+" << abs(X\_k[j]) << ")";

else cout << "(x-" << X\_k[j] << ")";

}

}

}\*/

/\*cout <<endl<< "Оценка погрешности:" <<endl;

double max = 0;

for (int i = 0; i < 100; i++) {

int x\_k = a + i \* (b - a) / 100;

double Pn\_xk = 0;

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

Pn\_xk += Rez1[j] \* pow(x\_k, j);

}

if (abs(F1(x\_k) - Pn\_xk) > max) max = abs(F1(x\_k) - Pn\_xk);

}

cout << max << endl;

cout << endl;\*/

cout << "Оценка погрешности:" << endl;

double max = 0;

for (int k = 0; k < 100; k++) {

double x\_k = a + k \* (b - a) / 100;

double summ = 0;

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

double el = Rez1[i];

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

if (i != j) {

el \*= (x\_k - X\_k[j]);

}

}

summ += el;

}

if (abs(summ - F2(x\_k)) > max) max = abs(summ - F2(x\_k));

}

cout << max << endl;

}

void Ravn1(int N) {

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

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

double a = -5, b = 5;

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

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

X\_k[i] = a + i \* (b - a) / N;

F\_k[i] = F1(X\_k[i]);

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

}

double\* Rez1 = new double[N + 1];

cout << "Интерполяционный многочлен " << N << " степени для функции f1:" << endl;

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

Rez1[i] = F\_k[i];

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

if (i != j) {

Rez1[i] /= (X\_k[i] - X\_k[j]);

}

}

}

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

if (Rez1[i] < 0) cout << fixed << showpoint << setprecision(20) << Rez1[i];

else cout << "+" << Rez1[i];

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

if (i != j) {

if (X\_k[j] < 0) cout << "(x+" << abs(X\_k[j]) << ")";

else cout << "(x-" << X\_k[j] << ")";

}

}

}\*/

/\*cout <<endl<< "Оценка погрешности:" <<endl;

double max = 0;

for (int i = 0; i < 100; i++) {

int x\_k = a + i \* (b - a) / 100;

double Pn\_xk = 0;

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

Pn\_xk += Rez1[j] \* pow(x\_k, j);

}

if (abs(F1(x\_k) - Pn\_xk) > max) max = abs(F1(x\_k) - Pn\_xk);

}

cout << max << endl;

cout << endl;\*/

cout << "Оценка погрешности:" << endl;

double max = 0;

for (int k = 0; k < 100; k++) {

double x\_k = a + k \* (b - a) / 100;

double summ = 0;

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

double el = Rez1[i];

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

if (i != j) {

el \*= (x\_k - X\_k[j]);

}

}

summ += el;

}

if (abs(summ - F1(x\_k)) > max) max = abs(summ - F1(x\_k));

}

cout << max << endl;

}

void Ravn2(int N) {

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

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

double a = -5, b = 5;

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

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

X\_k[i] = a + i \* (b - a) / N;

F\_k[i] = F2(X\_k[i]);

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

}

double\* Rez1 = new double[N + 1];

cout << "Интерполяционный многочлен " << N << " степени для функции f2:" << endl;

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

Rez1[i] = F\_k[i];

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

if (i != j) {

Rez1[i] /= (X\_k[i] - X\_k[j]);

}

}

}

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

if (Rez1[i] < 0) cout << fixed << showpoint << setprecision(20) << Rez1[i];

else cout << "+" << Rez1[i];

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

if (i != j) {

if (X\_k[j] < 0) cout << "(x+" << abs(X\_k[j]) << ")";

else cout << "(x-" << X\_k[j] << ")";

}

}

}\*/

/\*cout <<endl<< "Оценка погрешности:" <<endl;

double max = 0;

for (int i = 0; i < 100; i++) {

int x\_k = a + i \* (b - a) / 100;

double Pn\_xk = 0;

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

Pn\_xk += Rez1[j] \* pow(x\_k, j);

}

if (abs(F1(x\_k) - Pn\_xk) > max) max = abs(F1(x\_k) - Pn\_xk);

}

cout << max << endl;

cout << endl;\*/

cout << "Оценка погрешности:" << endl;

double max = 0;

for (int k = 0; k < 100; k++) {

double x\_k = a + k \* (b - a) / 100;

double summ = 0;

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

double el = Rez1[i];

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

if (i != j) {

el \*= (x\_k - X\_k[j]);

}

}

summ += el;

}

if (abs(summ - F2(x\_k)) > max) max = abs(summ - F2(x\_k));

}

cout << max << 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 << "Интерпаляционный многочлен в узлах Чебышёва:" << endl;

Cheb1(5);

Cheb1(10);

Cheb1(15);

Cheb1(20);

cout << endl;

Cheb2(5);

Cheb2(10);

Cheb2(15);

Cheb2(20);

cout <<endl<< "Интерпаляционный многочлен в равностоящих узлах:" << endl<<endl;

Ravn1(5);

Ravn1(10);

Ravn1(15);

Ravn1(20);

cout << endl;

Ravn2(5);

Ravn2(10);

Ravn2(15);

Ravn2(20);

system("pause");

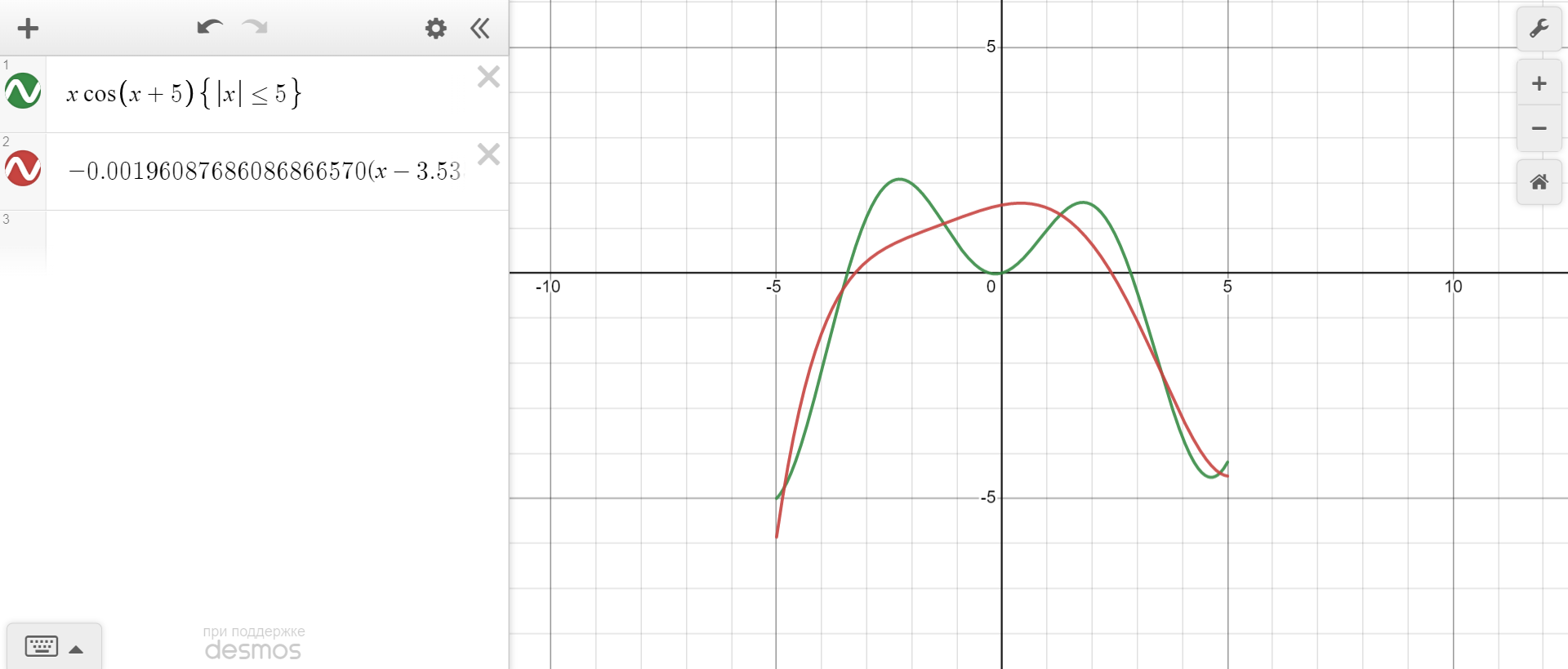
return 0;

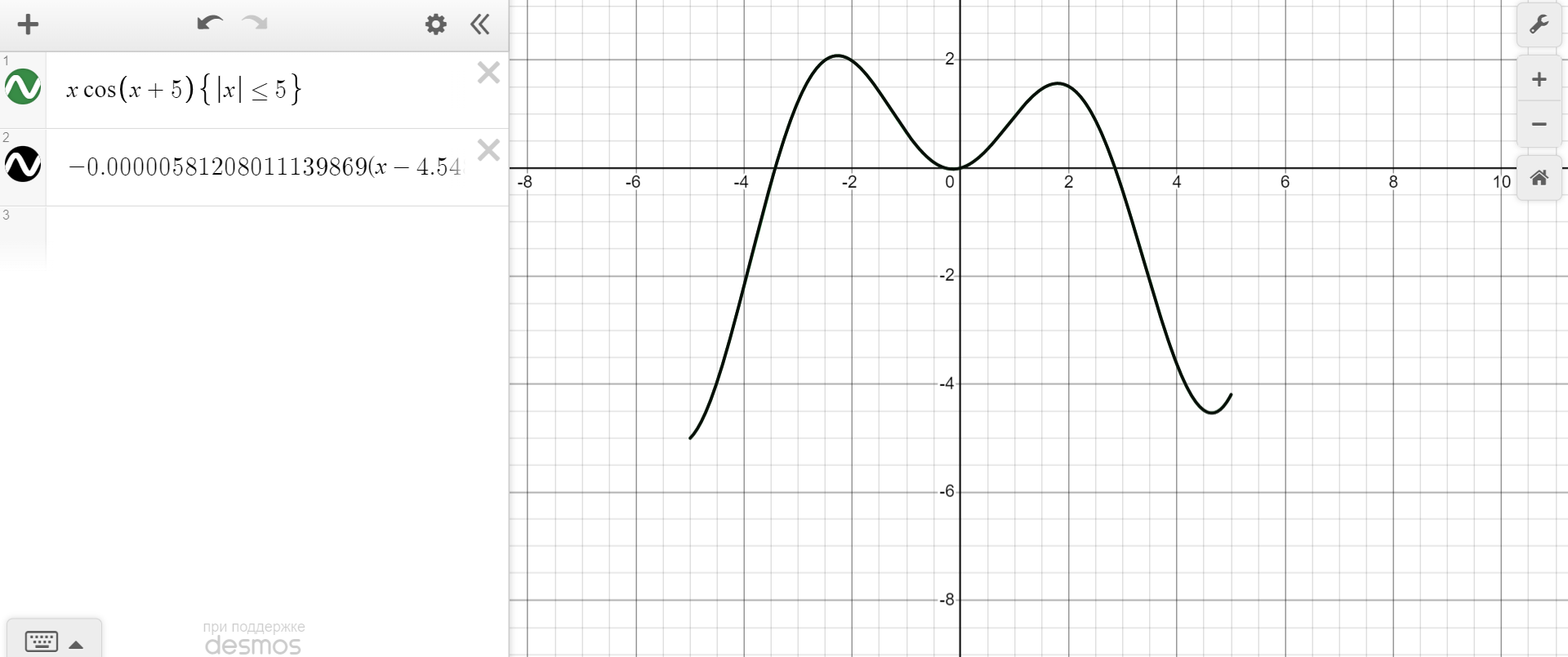
}

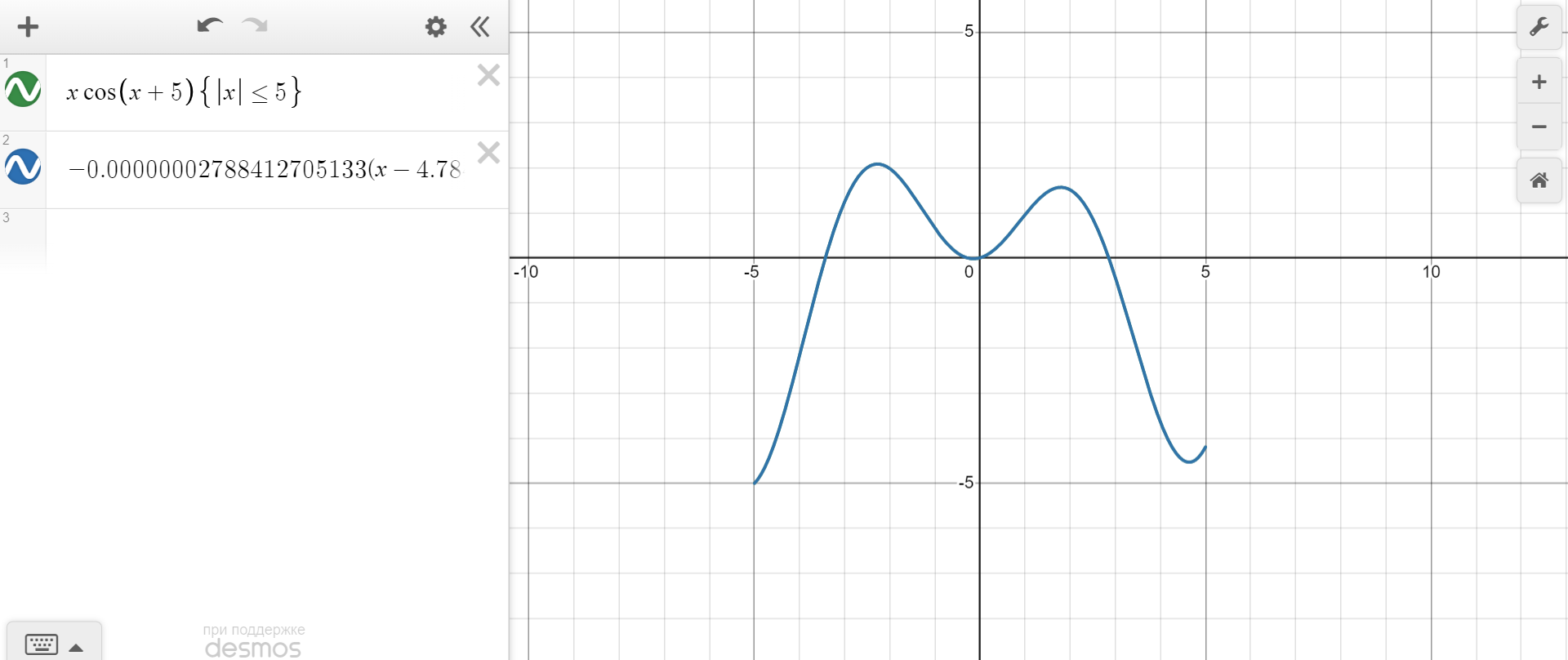
**Результаты**

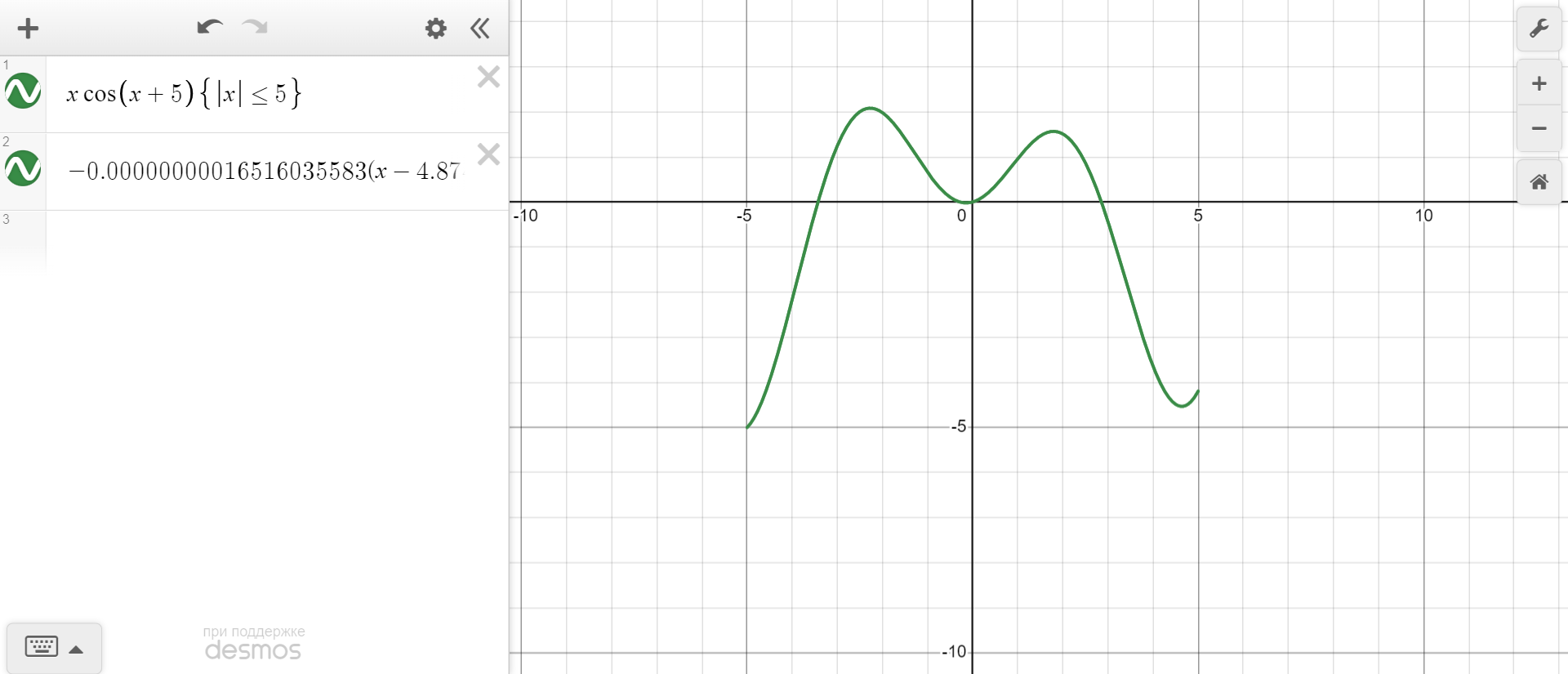
*Графики функций:*

1. Чебышёвские узлы, функция первая

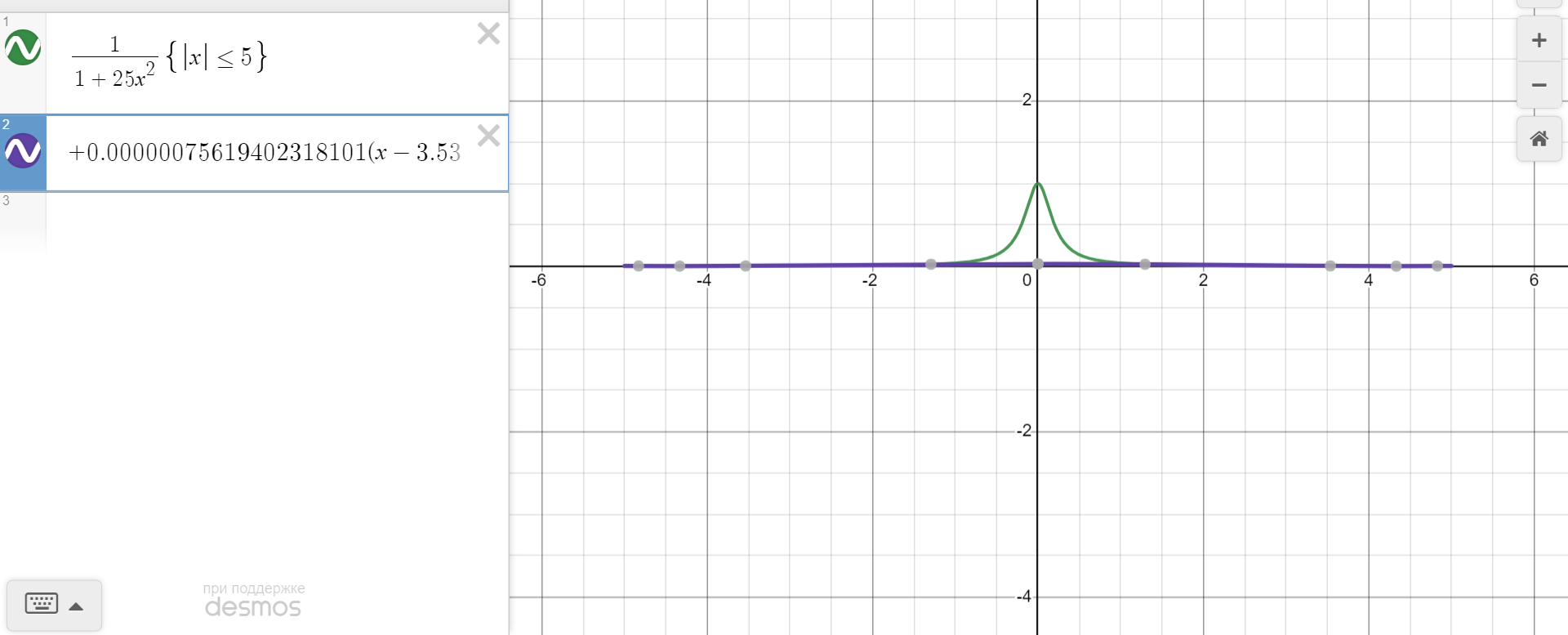


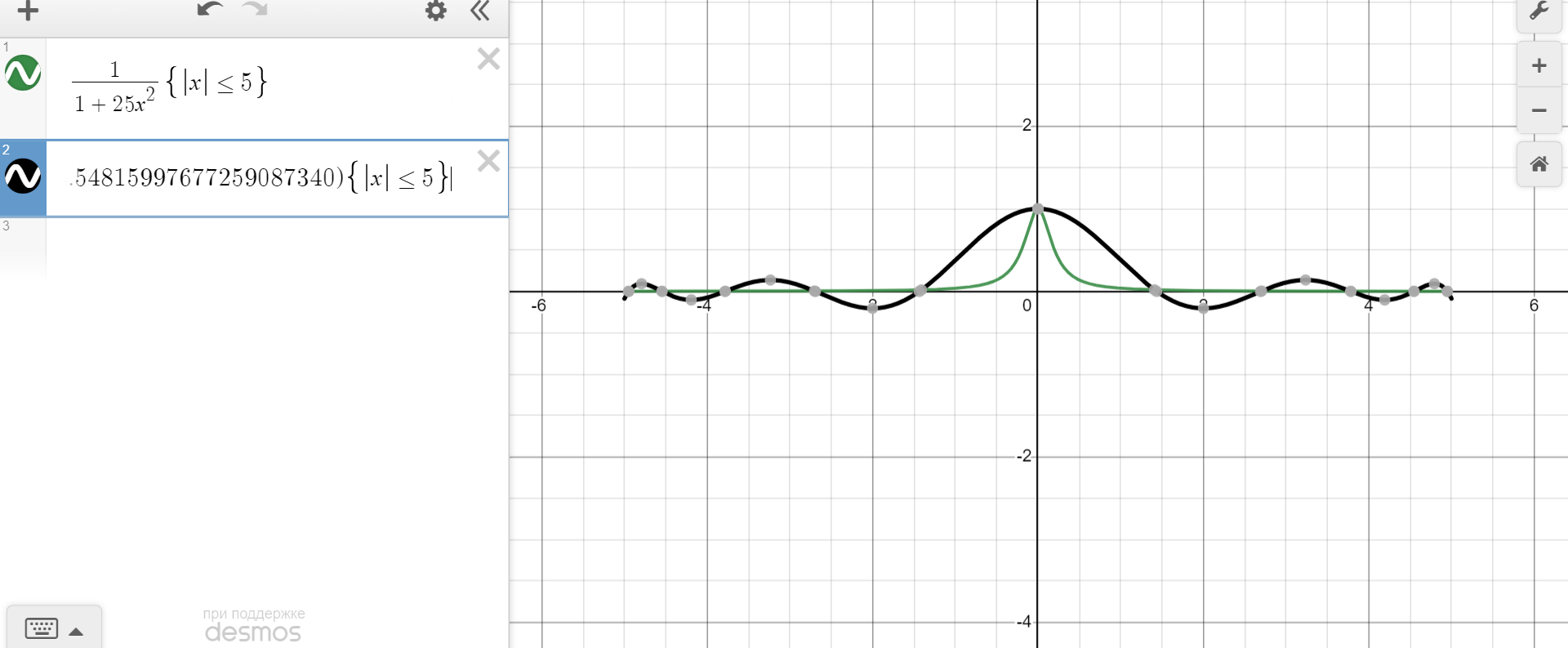




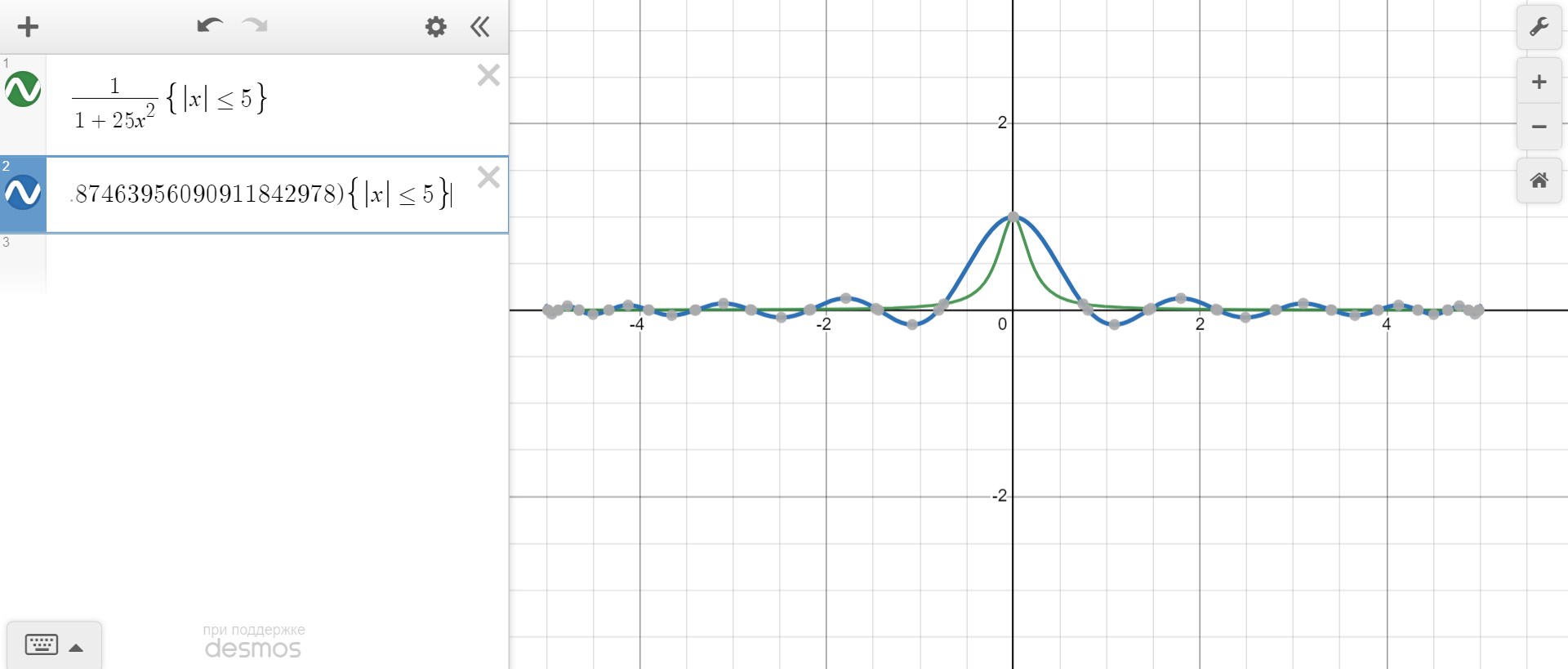


1. *Чебышёвские узлы, функция вторая*

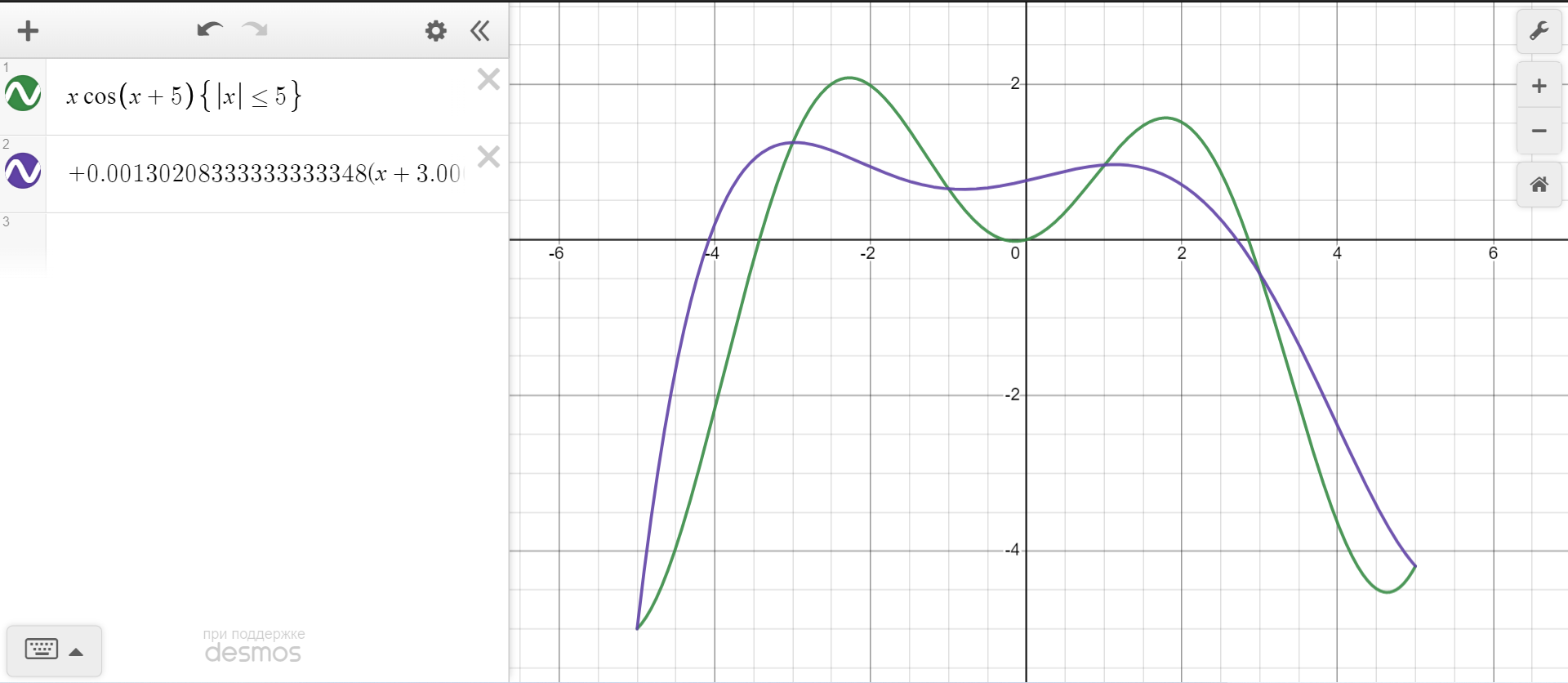


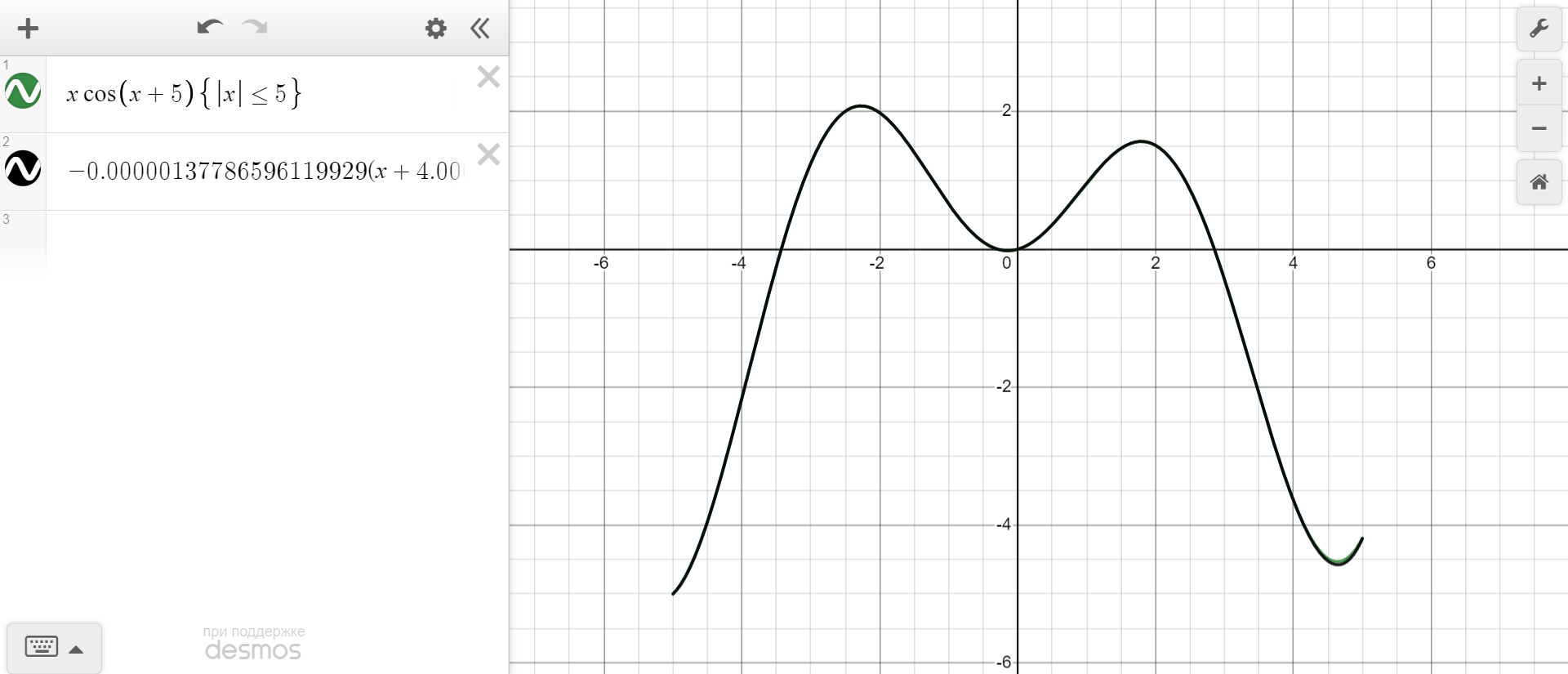


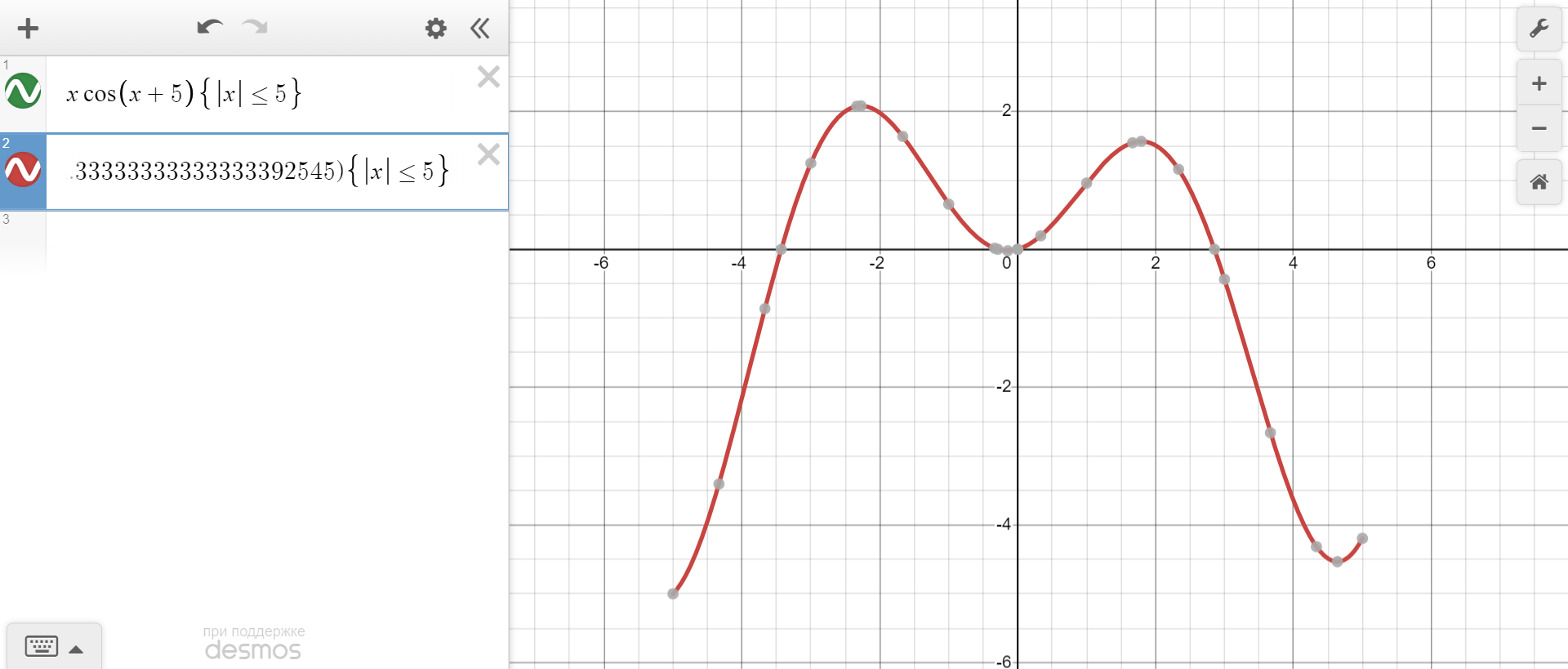


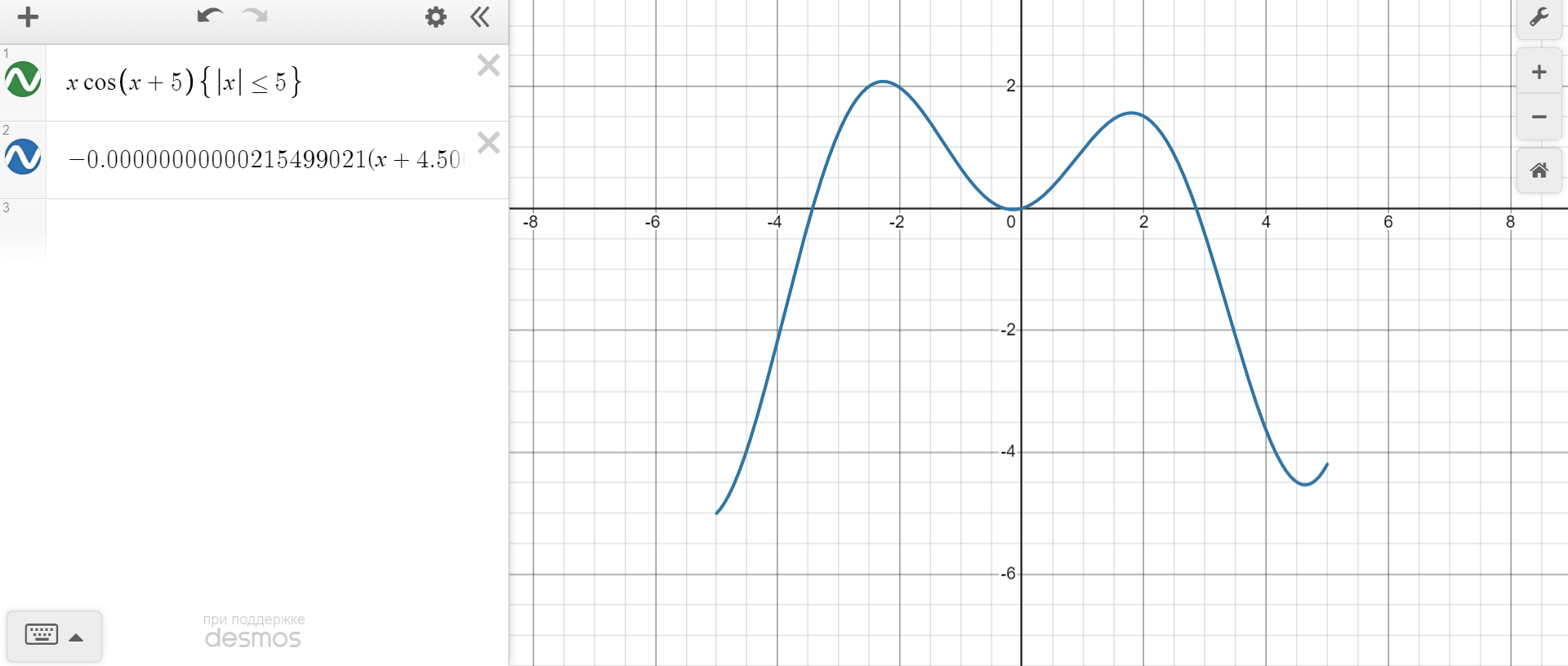


1. *Равностоящие узлы, функция первая*

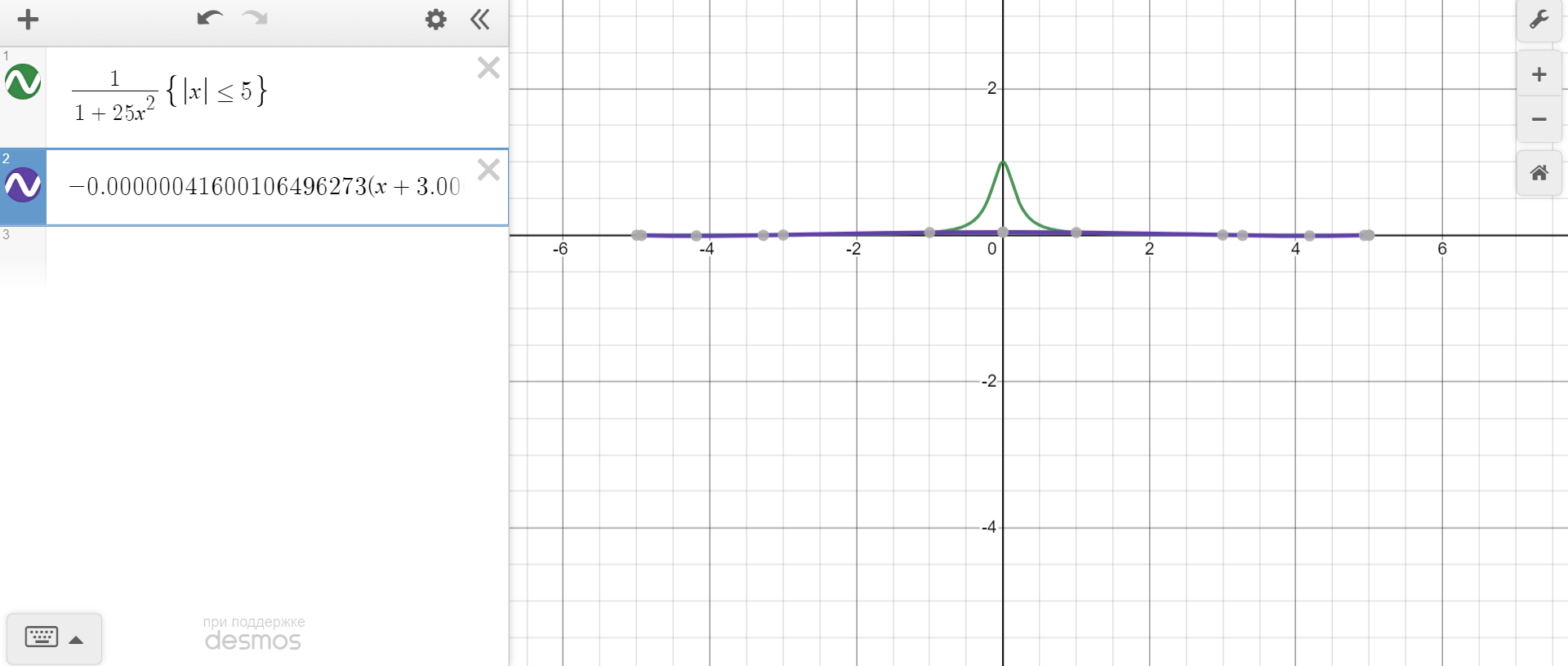


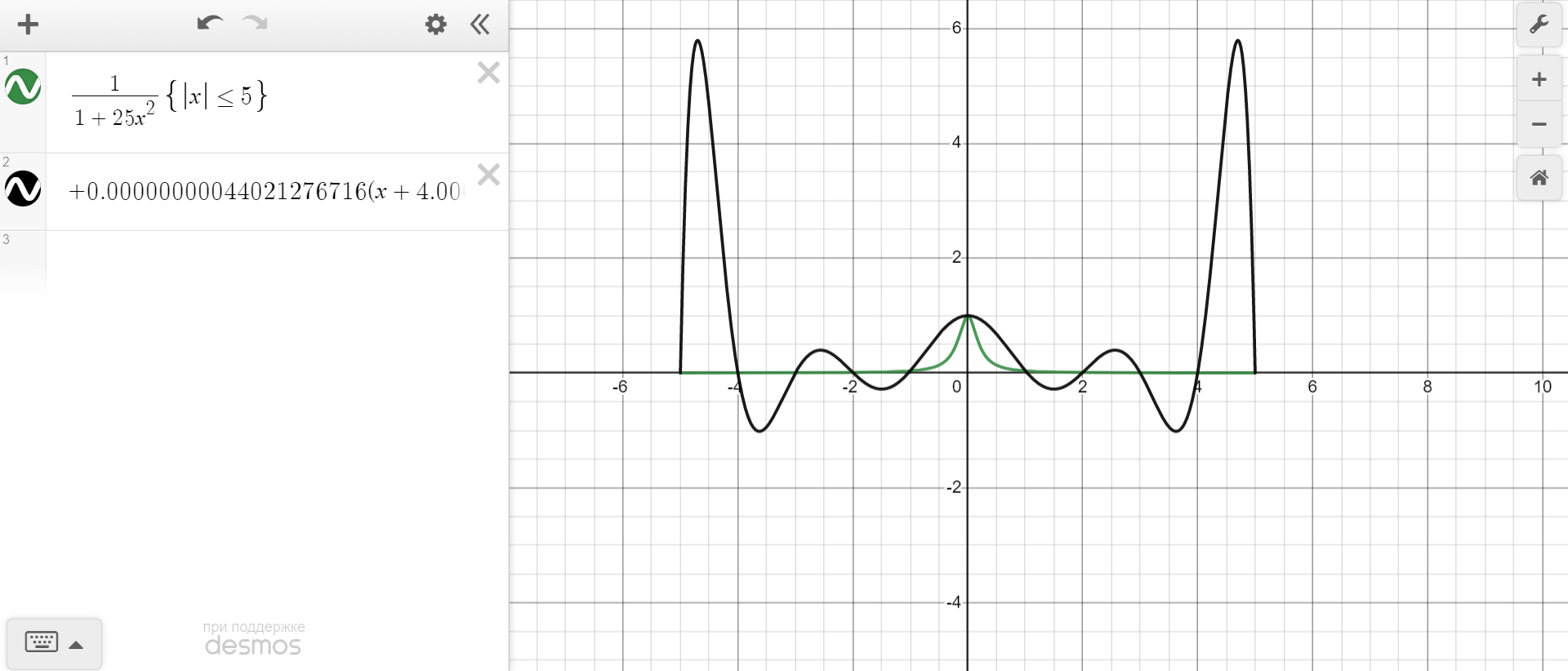


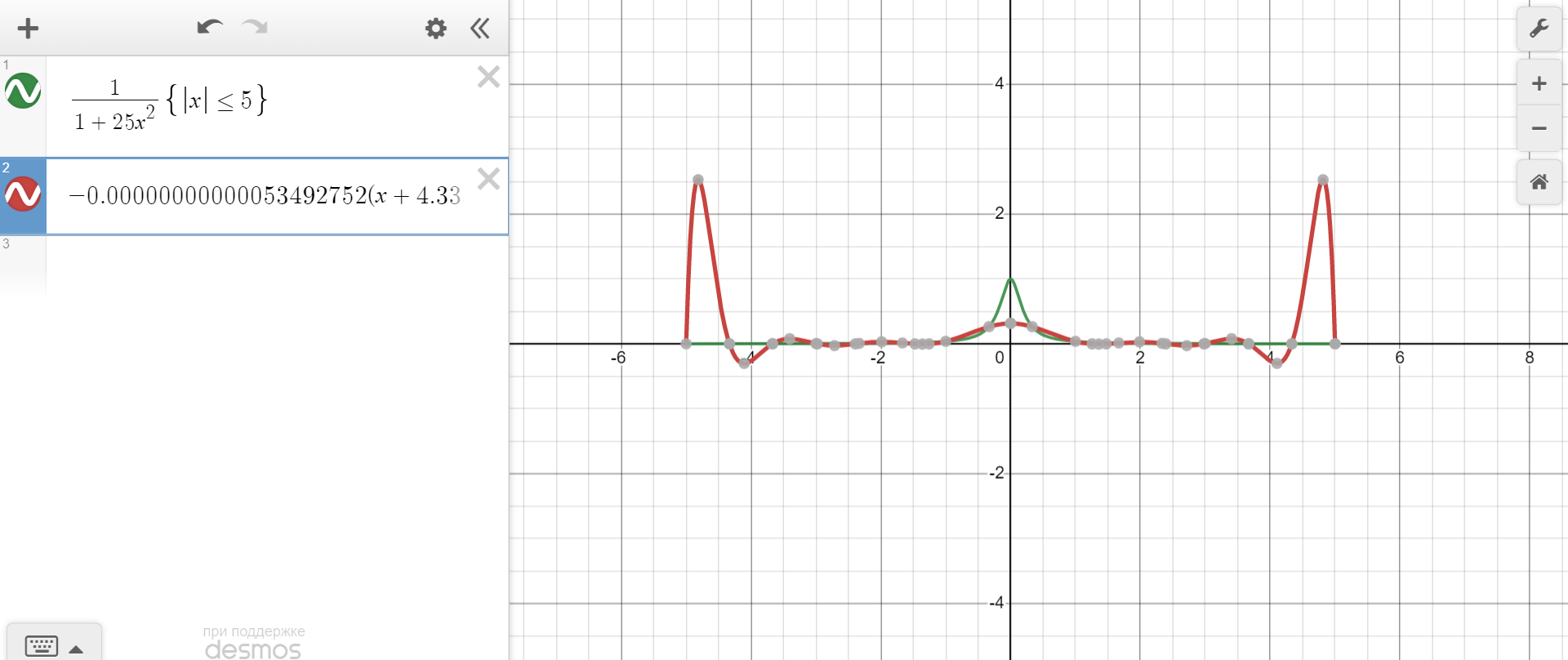


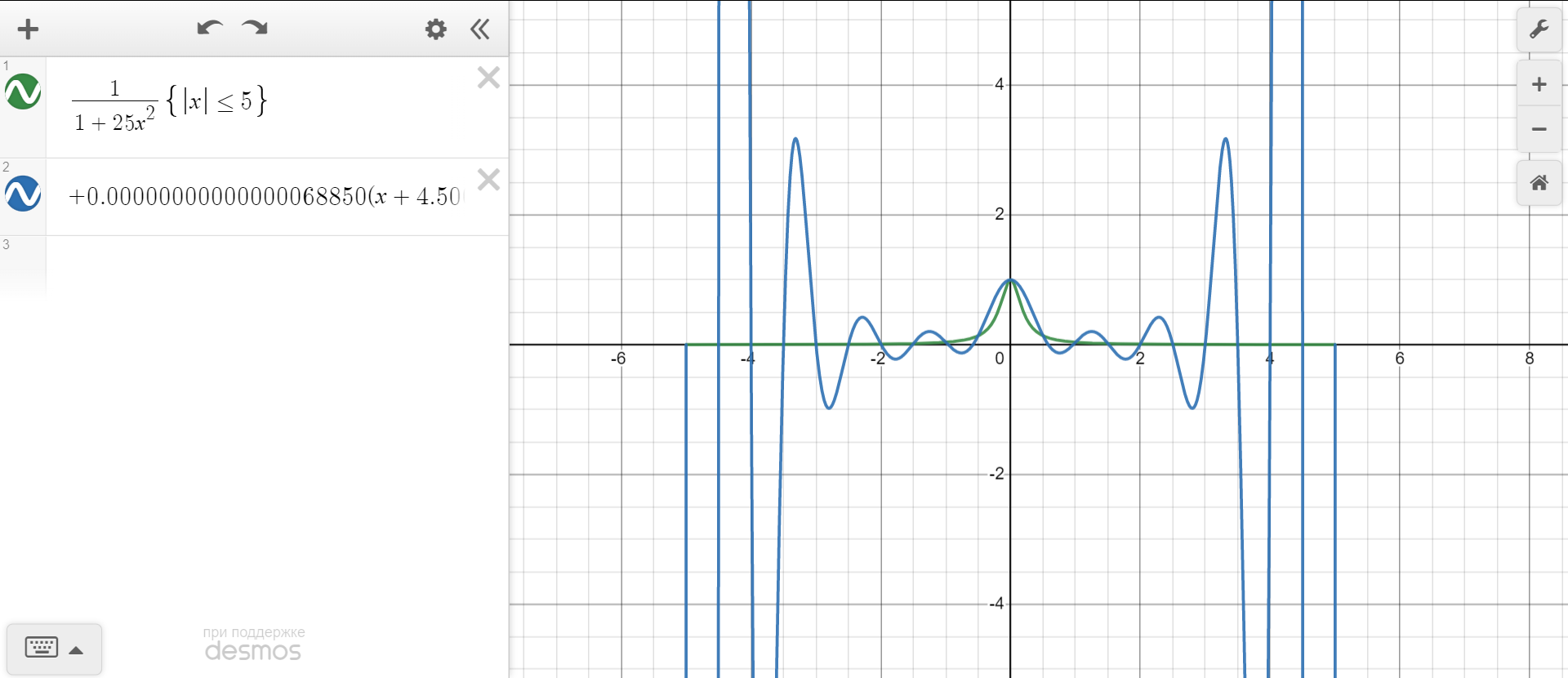


1. *Равностоящие узлы, функция вторая*









1. *Вывод программы:*

