Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа  
“Методы решения нелинейных уравнений”**

**Вариант 4**

Выполнил:   
студент группы ИВТ-23-2Б  
Жигалов Владислав Андреевич

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

2023 г.

**Задача**

**Постановка задачи**

Решить тремя методами уравнение 0,25x^3 + x – 1,2502 = 0

Отрезок, содержащий корень: [0;2]

Точное значение: 0,10001

**Анализ задачи**

**Метод половинного деления:**

Задаем начальные значения левой и правой границ отрезка: a = 0, b = 1.

Проверяем условие f(a) \* f(b) < 0, где f(x) = 0,25x^3 + x – 1,2502 = 0:

если условие не выполняется, выходим из метода, так как на данном отрезке нет корней, иначе выполняем итерации метода:

Вычисляем среднюю точку отрезка: c = (a + b) / 2.

Проверяем условие f(c) = 0 или |b - a| <= ɛ.

Если условие выполняется, то выводим значение x = c как приближенный корень, иначе, проверяем знак f(a) \* f(c):

Если f(a) \* f(c) < 0, обновляем правую границу b = c, иначе обновляем левую границу a = c.

Возвращаемся к шагу 4 и продолжаем итерации до достижения заданной точности или нахождения корня.

**Метод Ньютона:**

Задаем начальное приближение x0 = 2 (можно выбрать любое значение из отрезка [0, 2]).

Выполняем итерации метода:

На каждой итерации вычисляем значение функции и ее производной в точке x: f(x) и f'(x).

Обновляем значение x по формуле: x = x - f(x) / f'(x).

Проверяем условие |x – x-1| < ɛ, Если условие выполняется, выводим значение x как приближенный корень, в противном случае возвращаемся к шагу 2 и продолжаем итерации до достижения заданной точности или нахождения корня.

**Метод итераций:**

1. Найти первую производную f’(x) исходного уравнения f(x)=0.
2. Найти значение производной в точках границ отрезка f’(a), f’(b).
3. Найти экстремумы первой производной f’(x): найти вторую производную f’’(x), приравнять её к 0 f’’(x) = 0.
4. На заданном отрезке [a; b] выбирается начальное приближение х0 (x0 ∈ [a; b]).
5. Следующее приближение вычисляется по формуле xn+1=φ(хn).
6. 3 шаг повторяется пока |xn+1 - xn| > ε.
7. Как только |xn+1 - xn| <= ε, цикл заканчивается, xn+1 — искомое значение.

**Смысловые значения:**

**a –** левая граница отрезка [a;b]

**b** – правая граница отрезка [a;b]

ɛ – заданная точность

x – корень уравнения

fb – значение функции в точке b

fc – значение функции в точке середины отрезка [a;b]

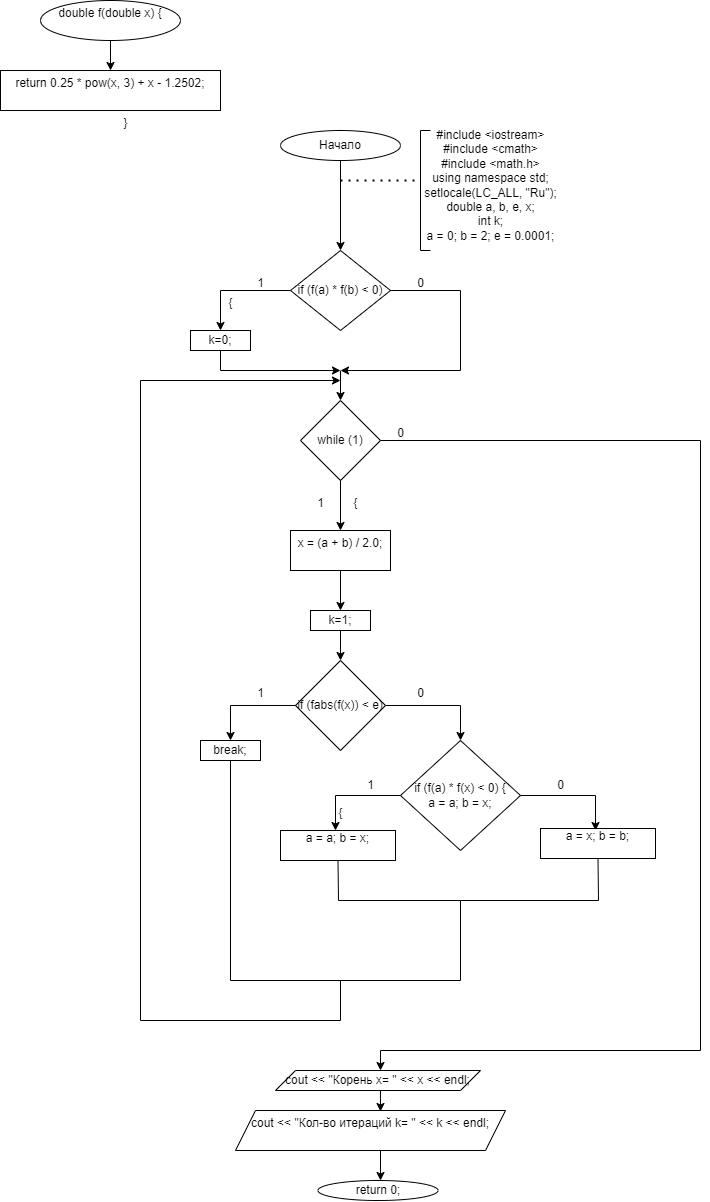
f1 – первая производная функции

f2 – вторая производная функции (производная от первой производной)

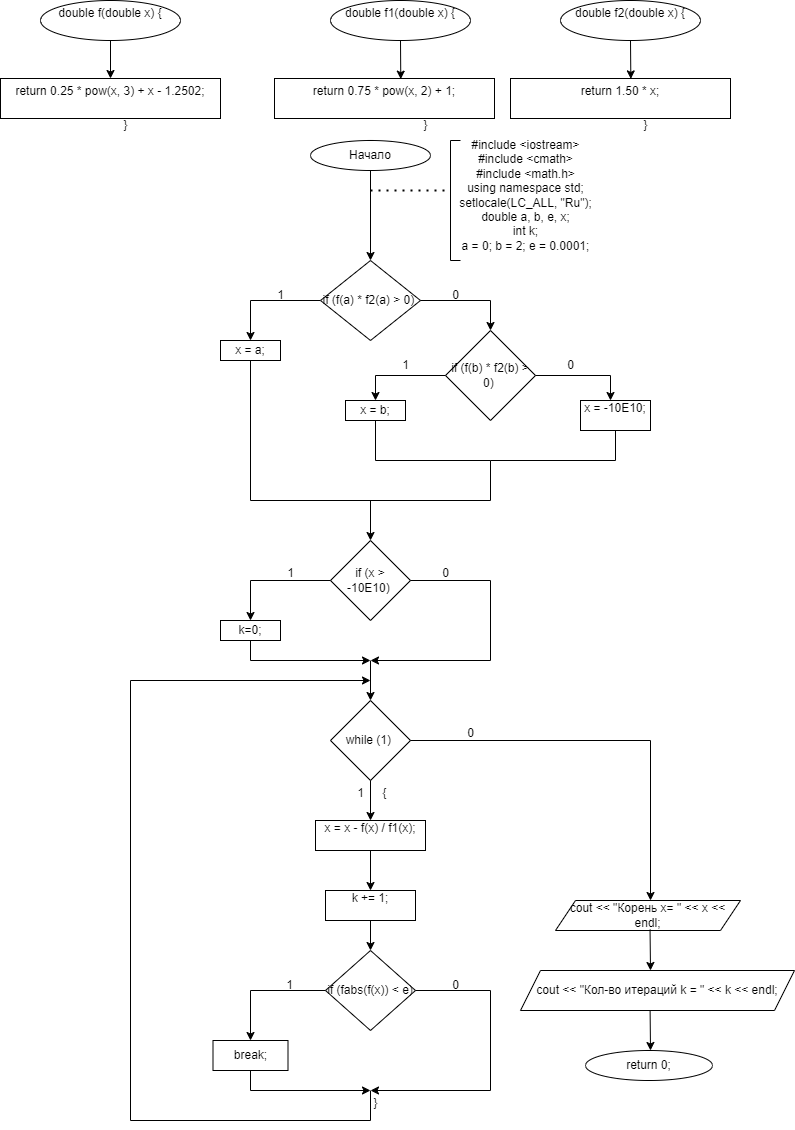
g – тоже самое, что и фи(x)

**Блок схема:**

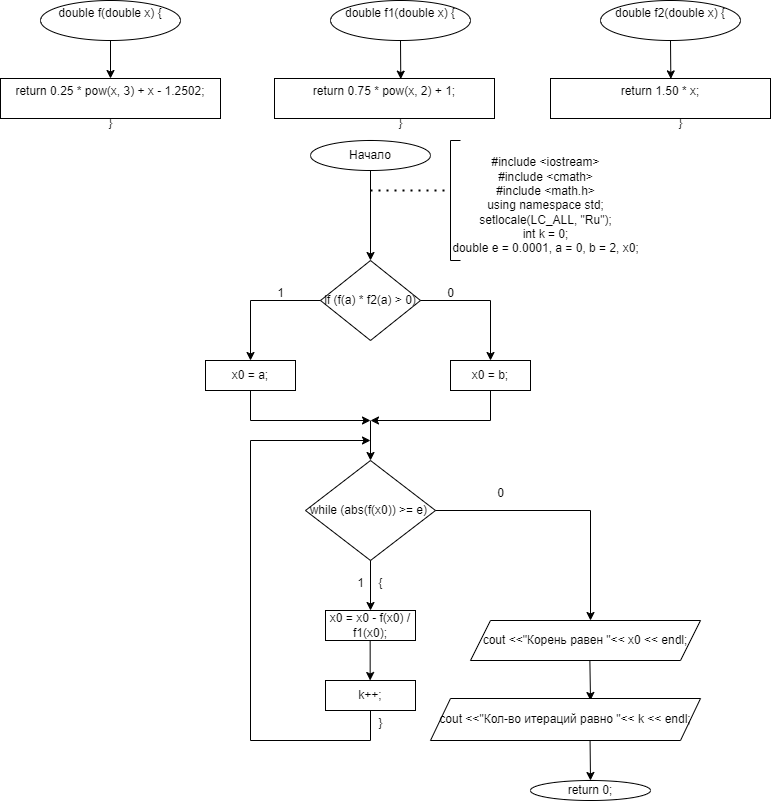
**Метод половинного деления:**



**Метод Ньютона:**



**Метод Итераций:**



**Код программы:**

**1) Метод половинного деления:**

#include <iostream>

#include <cmath>

#include <math.h>

using namespace std;

double f(double x) {

return 0.25 \* pow(x, 3) + x - 1.2502;

}

int main() {

setlocale(LC\_ALL, "Ru");

double a, b, e, x;

int k;

a = 0; b = 2; e = 0.0001;

cout << "a = " << a << endl;

cout << "b = " << b << endl;

cout << "e = " << e << endl;

if (f(a) \* f(b) < 0) {

cout << " Условие сходимости выполнено" << endl;

k = 0;

while (1) {

x = (a + b) / 2.0;

k += 1;

if (fabs(f(x)) < e)

break;

if (f(a) \* f(x) < 0) {

a = a; b = x;

}

else {

a = x; b = b;

}

}

cout << "Корень x= " << x << endl;

cout << "Кол-во итераций k= " << k << endl;

}

return 0;

}

**2) Метод Ньютона:**

#include <iostream>

#include <cmath>

#include <math.h>

using namespace std;

double f(double x) {

return 0.25 \* pow(x, 3) + x - 1.2502;

}

double f1(double x) {

return 0.75 \* pow(x, 2) + 1;

}

double f2(double x) {

return 1.50 \* x;

}

int main() {

setlocale(LC\_ALL, "Ru");

double a, b, e, x;

int k;

a = 0; b = 2; e = 0.0001;

if (f(a) \* f2(a) > 0) {

x = a;

}

else {

if (f(b) \* f2(b) > 0) {

x = b;

}

else {

cout << "УСловие сходимости не выполнено" << endl;

x = -10E10;

}

}

if (x > -10E10) {

k = 0;

while (1) {

x = x - f(x) / f1(x);

k += 1;

if (fabs(f(x)) < e)

break;

}

cout << "Корень x= " << x << endl;

cout << "Кол-во итераций k = " << k << endl;

}

return 0;

}

**3) Метод Итераций:**

#include <iostream>

#include <cmath>

#include <math.h>

using namespace std;

double f(double x) {

return 0.25 \* pow(x, 3) + x - 1.2502;

}

double f1(double x) {

return 0.75 \* pow(x, 2) + 1;

}

double f2(double x) {

return 1.50 \* x;

}

int main() {

setlocale(LC\_ALL, "Ru");

int k = 0;

double e = 0.0001, a = 0, b = 2, x0;

if (f(a) \* f2(a) > 0)

x0 = a;

else

x0 = b;

while (abs(f(x0)) >= e) {

x0 = x0 - f(x0) / f1(x0);

k++;

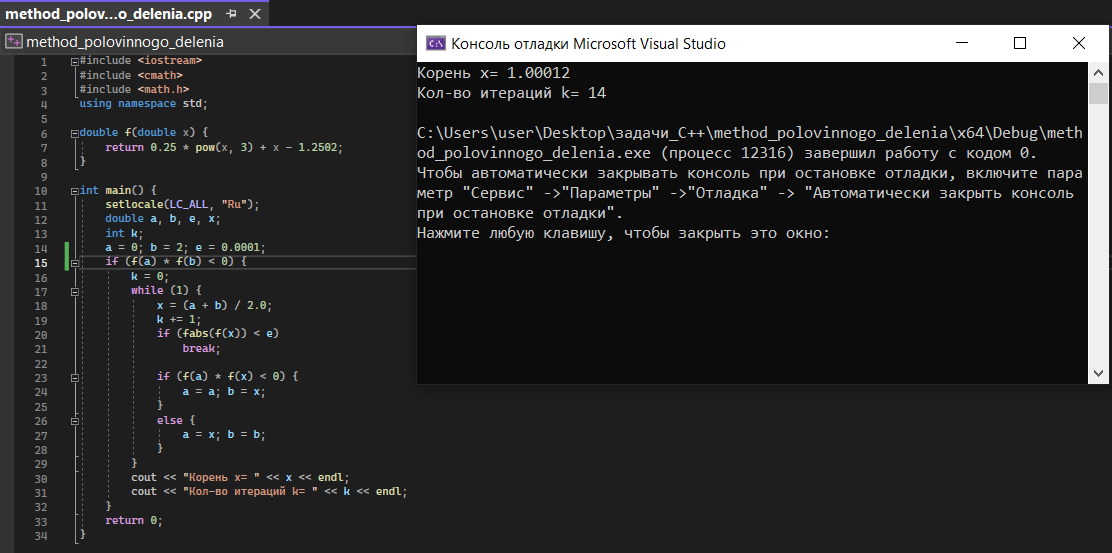
}

cout <<"Корень x= "<< x0 << endl;

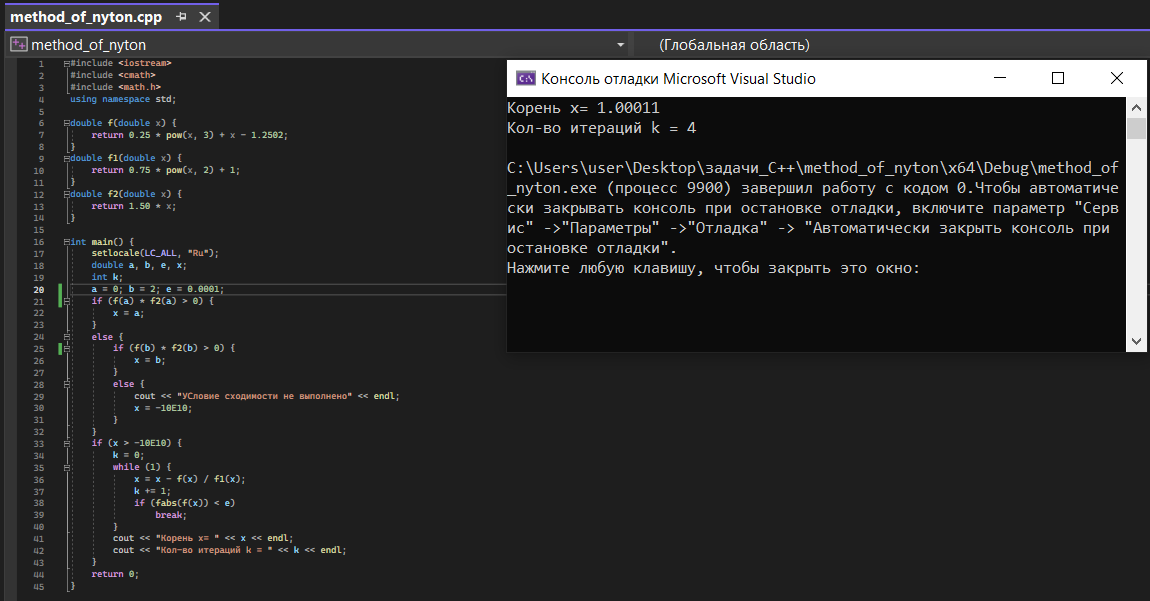
cout <<"Кол-во итераций k = "<< k << endl;

}

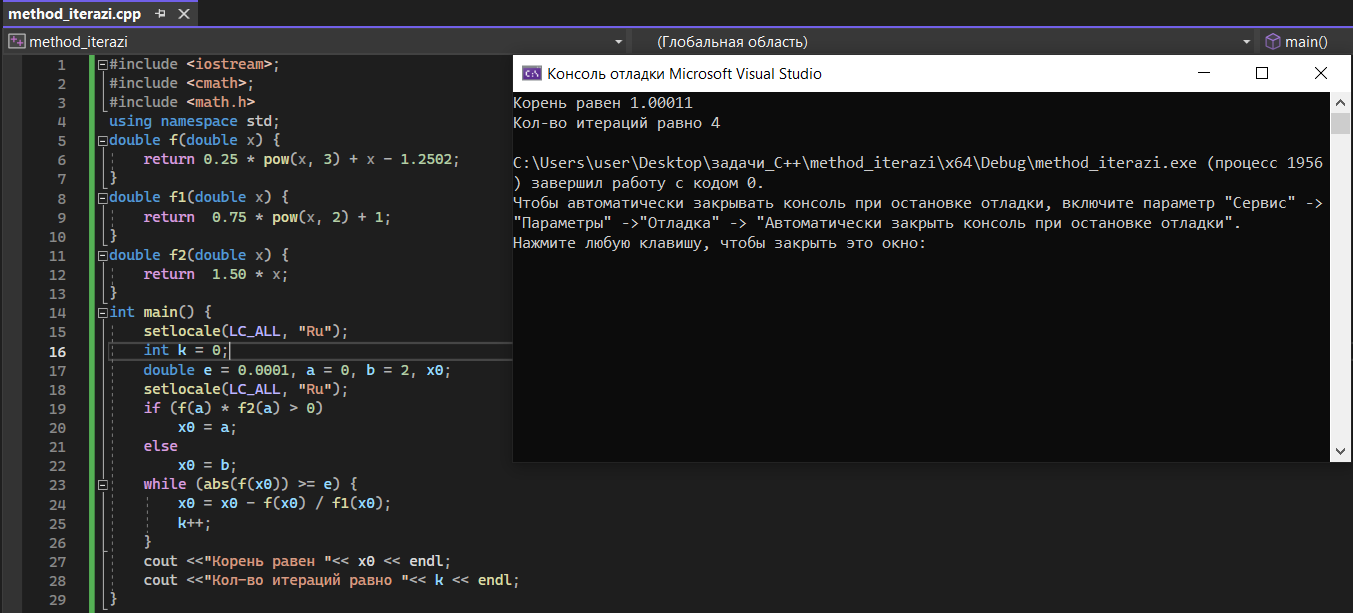
Скриншоты из Visual Studio  
**1) Метод половинного деления:**



**2) Метод Ньютона:**



**3) Метод Итераций:**



Графики:  
1) Метод половинного деления:

2)Метод Ньютона:

3)Метод Итераций:

**Вывод**: Задачи были выполнили. Всё получилось, программа сработала

корректно и вывела желаемый результат.