«Пермский национальный исследовательский политехнический университет» ПНИПУ

**Лабораторная работа**

**“Методы решения нелинейных уравнений”**

Выполнил:

студент группы РИС-23-1б

Фаезов Александр Рамисович

Проверила:

доцент кафедры ИТАС

Ольга Андреевна Полякова

Пермь

2023

Задача

Решить уравнение ex-e-x-2=0 тремя методами. Отрезок, содержащий корень: [0;1]. Точное значение: 0,8814.

Анализ задачи

**Метод половинного деления:**

Проверяем условие f(a) \* f(b) < 0, где f(x) = ex-e-x-2=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| <= epsilon.

Если условие выполняется, 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:**

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| <epsilon, Если условие выполняется, выводим значение x как приближенный корень.

Иначе, 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]

epsilon – заданная точность

x – корень уравнения

fb – значение функции в точке b

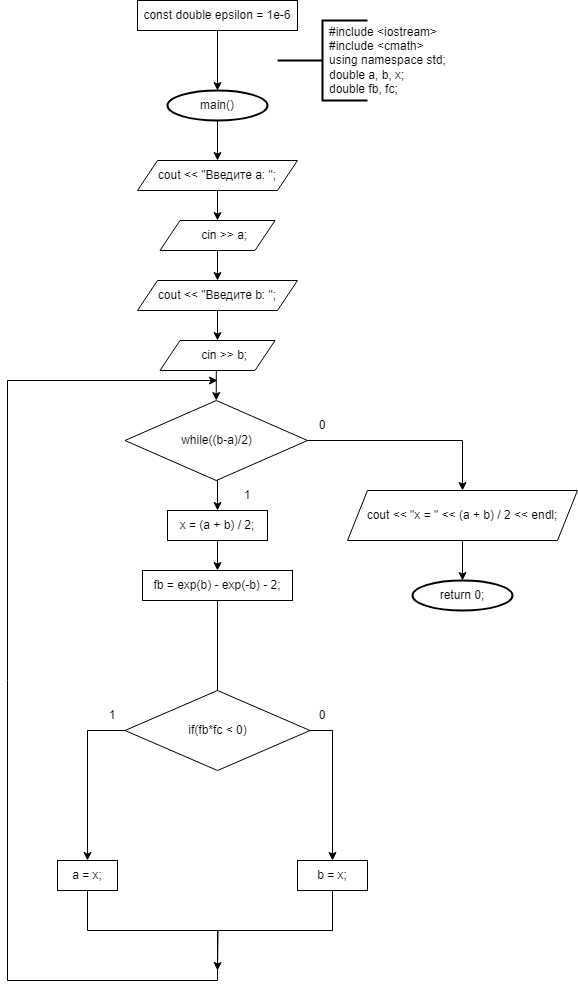
fc – значение функции в точке середины отрезка [a;b]

f1 – первая производная функции

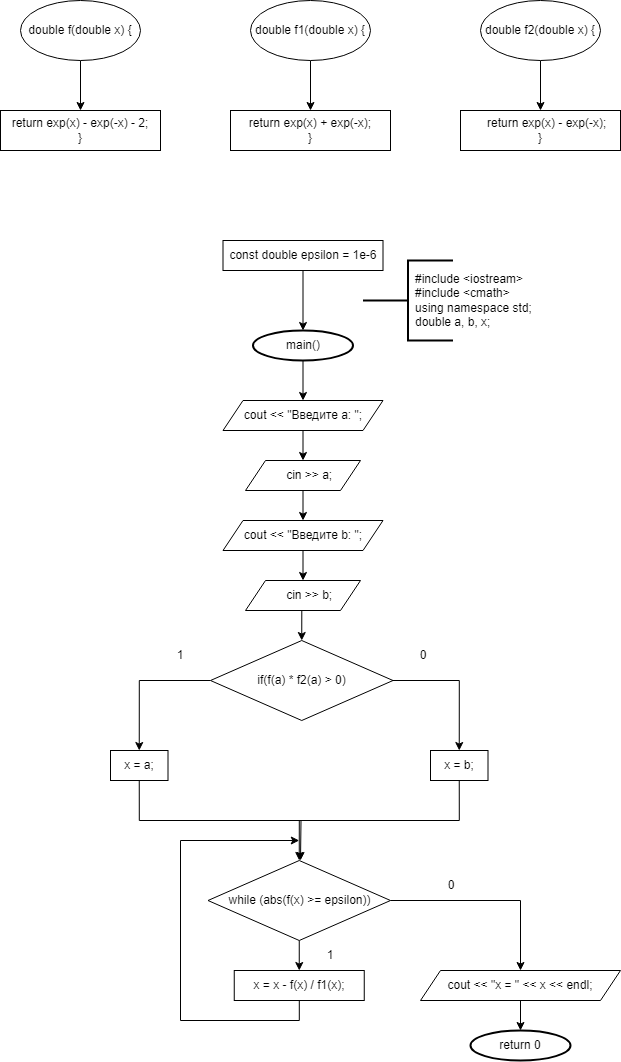
f2 – вторая производная функции (производная от первой производной)

g – тоже самое, что и фи(x)

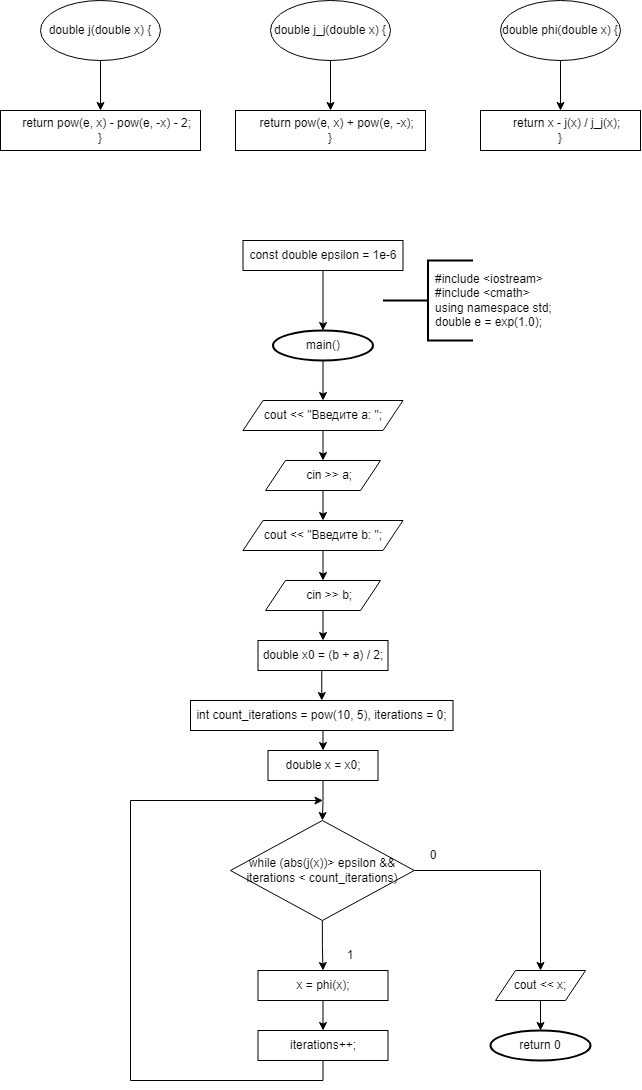
Блок-схемы



Блок-схема 1 – Метод половинного деления



Блок-схема 2 – Метод Ньютона



Блок-схема 2 – Метод Итераций

Код

#include <iostream>

#include <cmath>

using namespace std;

const double epsilon = 1e-6;

void method1();

void method2();

void method3();

int main()

{

setlocale(LC\_ALL, "ru");

int methodNumber;

cout << "Выберите метод: " << endl;

cout << "1. Метод половинного деления: " << endl;

cout << "2. Метод Ньютона: " << endl;

cout << "3. Метод итераций: " << endl;

cin >> methodNumber;

switch (methodNumber)

case 1: {

method1();

break;

case 2:

method2();

break;

case 3:

method3();

break;

default:

break;

}

}

//Метод половинного деления

void method1() {

double a, b, x;

double fb, fc;

cout << "Введите а: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

while (b - a > epsilon)

{

x = (a + b) / 2;

fb = exp(b) - exp(-b) - 2;

fc = exp(x) - exp(-x) - 2;

if (fb \* fc < 0)

{

a = x;

}

else

{

b = x;

}

}

cout << "x = " << (a + b) / 2 << endl;

}

//Метод Ньютона

double f(double x) {

return exp(x) - exp(-x) - 2;

}

double f1(double x) {

return exp(x) + exp(-x);

}

double f2(double x) {

return exp(x) - exp(-x);

}

void method2() {

double a, b, x;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

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

{

x = a;

}

else

{

x = b;

}

while (abs(f(x) >= epsilon))

{

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

}

cout << "x = " << x << endl;

}

//Метод итераций

double e = exp(1.0);

double j(double x) {

return pow(e, x) - pow(e, -x) - 2;

}

double j\_j(double x) {

return pow(e, x) + pow(e, -x);

}

double phi(double x) {

return x - j(x) / j\_j(x);

}

void method3() {

double a, b;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

double x0 = (b + a) / 2;

int count\_iterations = pow(10, 5), iterations = 0;

double x = x0;

while (abs(j(x))> epsilon && iterations < count\_iterations)

{

x = phi(x);

iterations++;

}

cout << x;

}

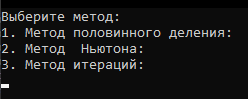


Рис. 1 – пример работы программы

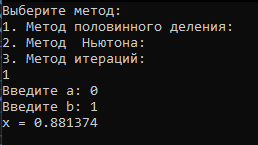


Рис. 2 – пример работы программы