Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**Лабораторная работа №1**

**Методы решения нелинейных уравнений**

**Вариант: 4**

Выполнил работу

студент группы ИВТ-24-2б

К.Д.Горбушин

Проверил

Доцент кафедры ИТАС

О.А.Полякова

Пермь, 2024

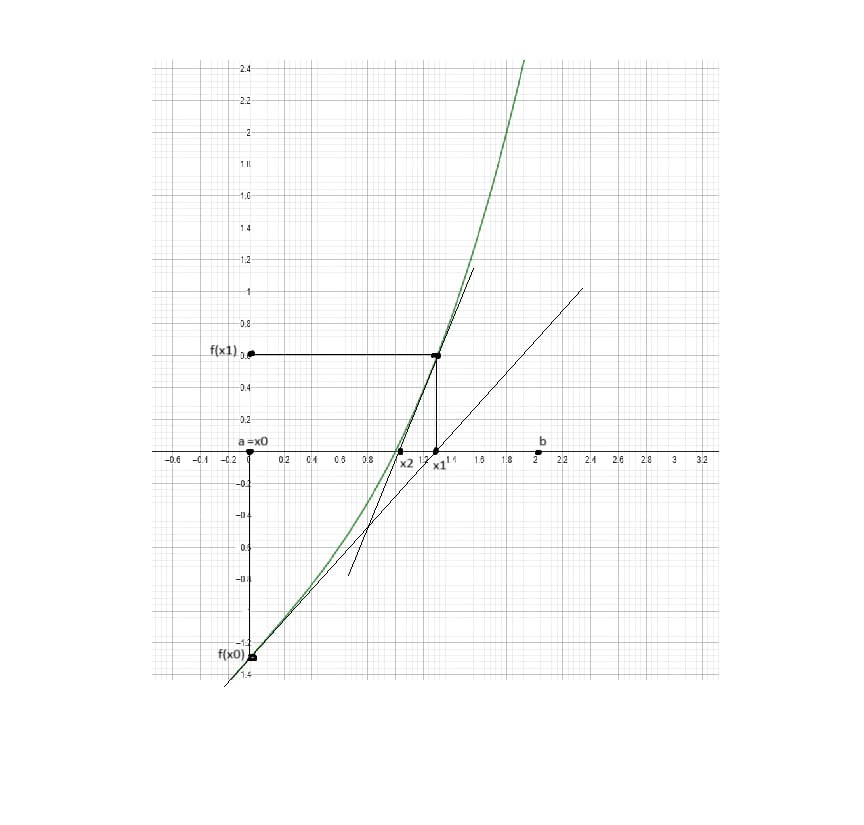
**Постановка задачи**

Решить уравнение тремя методами.

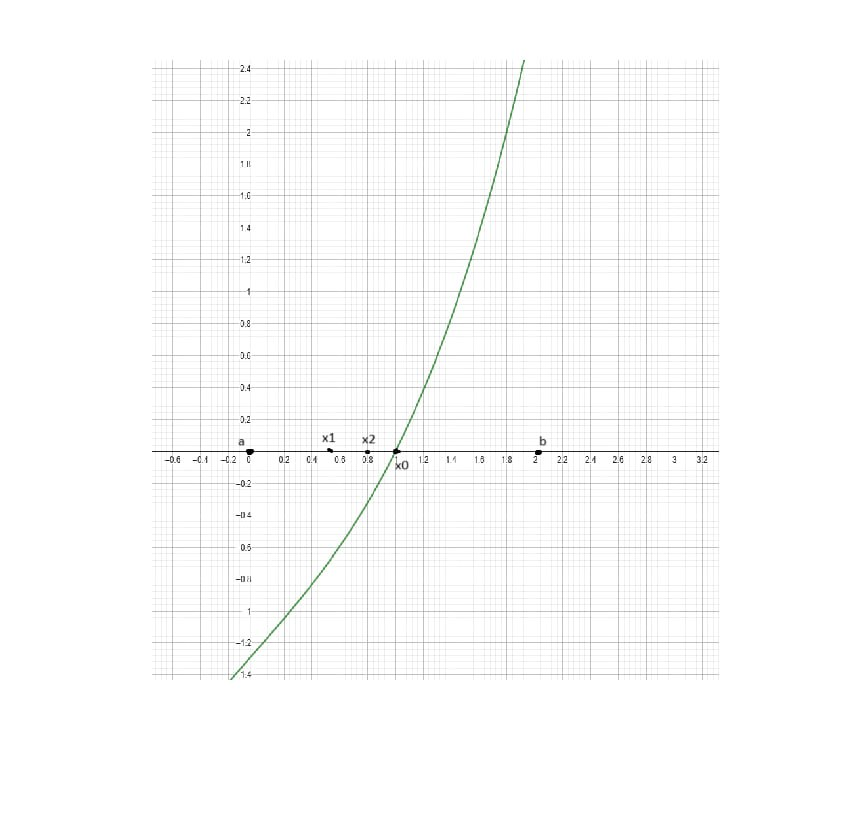
Решить нелинейное уравнение: 0,25x3 + x - 1,2502 = 0, используя три метода: Ньютона, половинного деления (Дихотомии). С заданной точностью ε = 1e^-8, на отрезке [0;2].

**Геометрическая интерпретация**

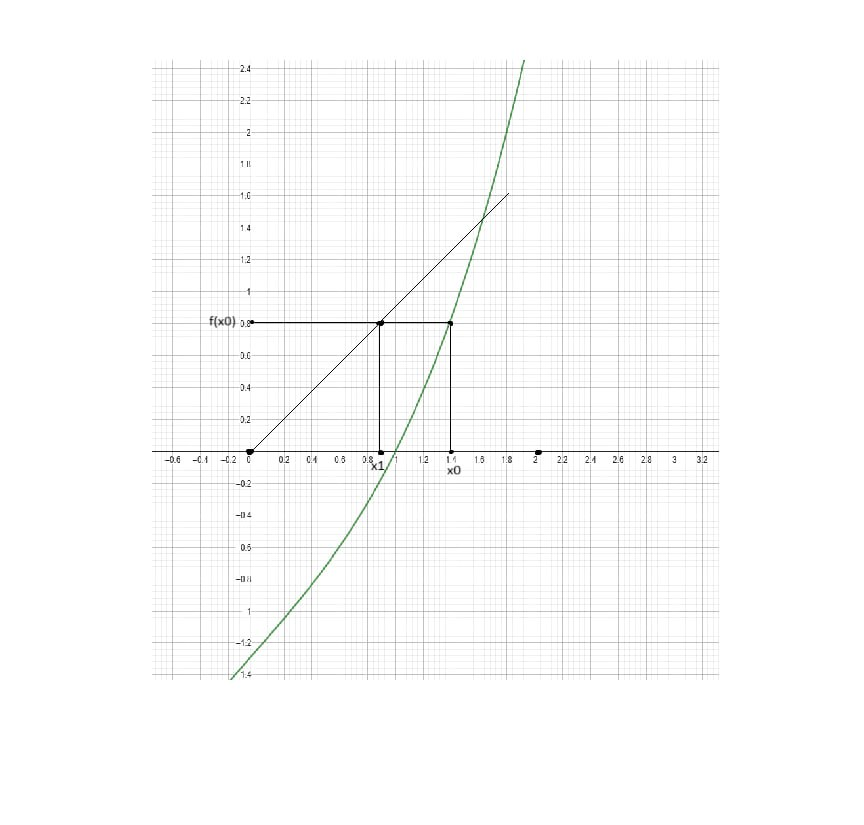
Метод Ньютона:



Метод половинного деления:



Метод итераций:



**Анализ задачи**

Метод Ньютона:

1. Определение функции:
2. Определение производных:

А)

Б) = 1.5x

1. Инициализация: задаем начальное приближение со стороны одной из граней интервала, где выполняется условие
2. Итерации: Находим значения хn, по формуле:
3. Продолжаем действия до того момента, когда будет <

Метод половинного деления:

1. Определение функции:
2. Инициализация: начальное значение находится на середине заданного отрезка [1;2]
3. Итерации: проверяем интервалы (a, ) и (, b).

А) Если – отбрасываем

Б) Если – берем   
В) Для пунктов А и Б, действует то же самое, но в обратную сторону

Г) После того, как мы взяли определенный интервал – начинаем его делить пополам

1. Продолжаем действия до того момента, когда будет <

Метод итераций:

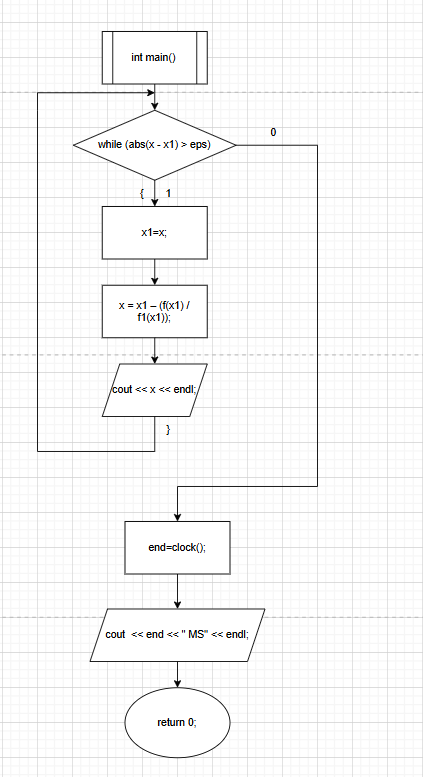
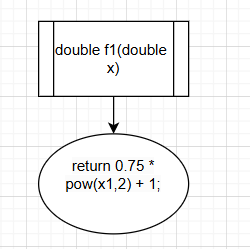
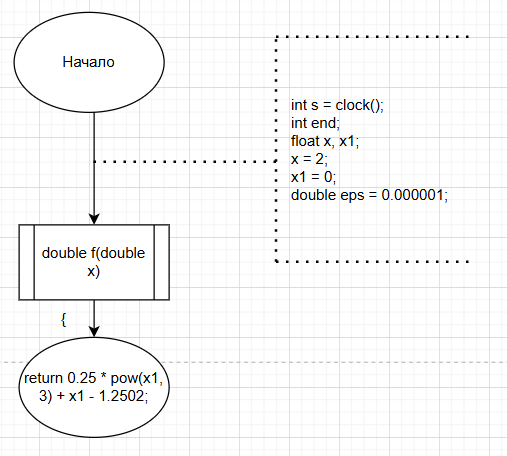
1. Определение функции :
2. Выражаем
3. Выбираем произвольное значение из отрезка [0;2]
4. Итерации:

А) Находим новое значение

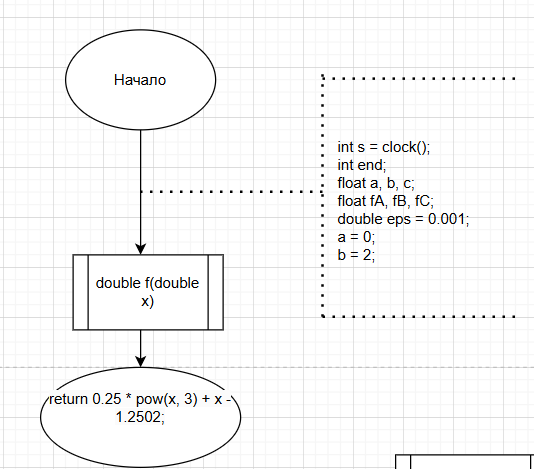
Б) Продолжаем действия до того момента, когда будет <

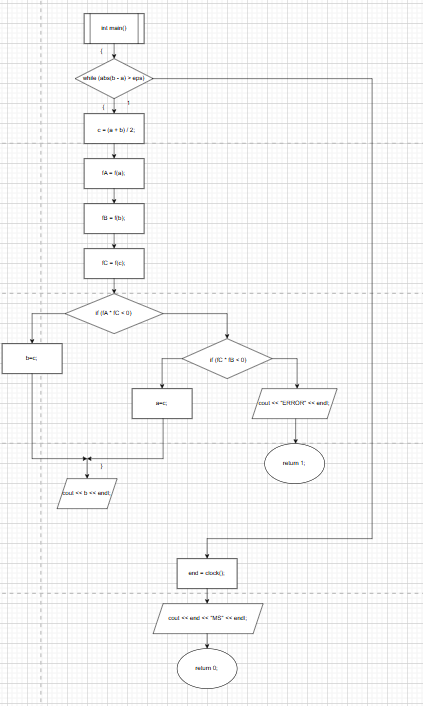
**Блок-схемы**

Метод Ньютона:

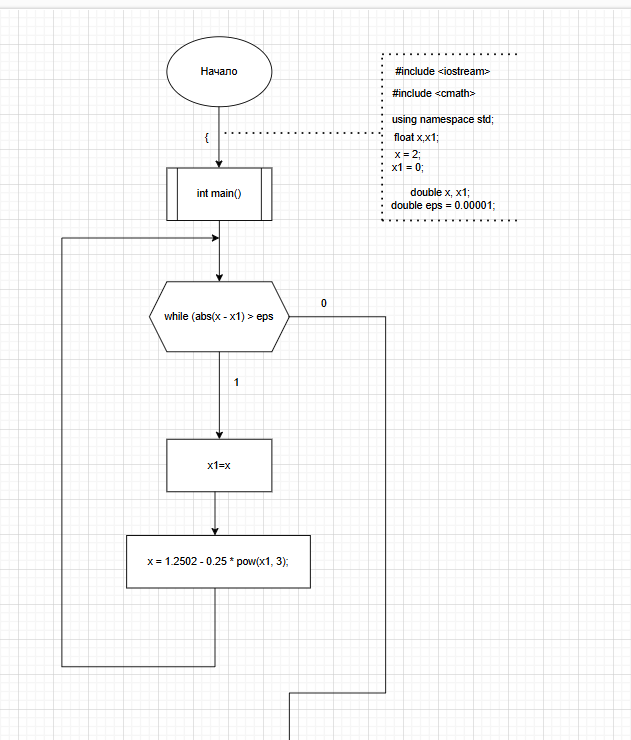
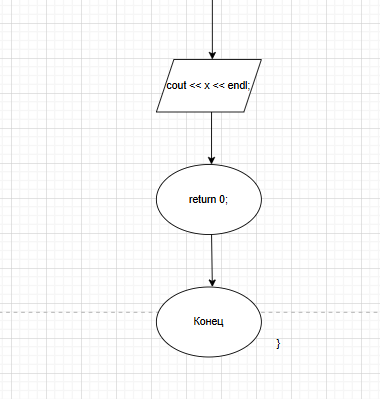


Метод половинного деления:

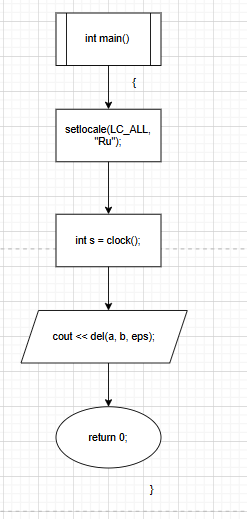
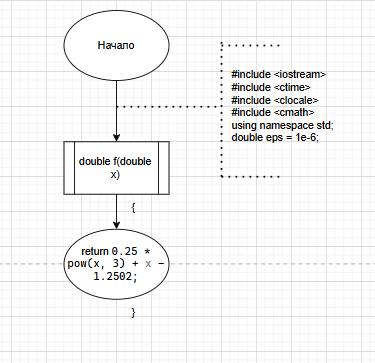


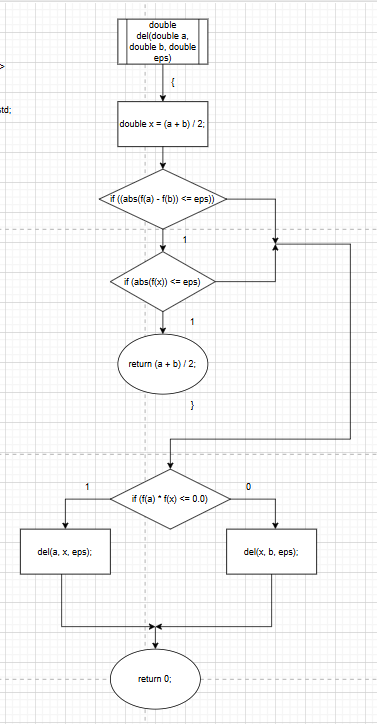


Метод итераций:

Половинное деление(рекурсия):





**Код программы**

Метод Ньютона:

#include <iostream>

#include <cmath>

#include <ctime>

using namespace std;

double f(double x1){

return 0.25 \* pow(x1, 3) + x1 - 1.2502;

}

double f1(double x1){

return 0.75 \* pow(x1,2) + 1;

}

int main() {

int s = clock();

int end;

float x, x1;

x = 2;

x1 = 0;

double eps = 0.000001;

while (abs(x - x1) > eps) {

x1 = x;

x = x1 - (f(x1) / f1(x1));

cout << x << endl;

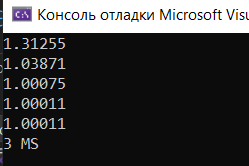
}

end = clock();

cout << end << " MS" << endl;

return 0;

}



Метод половинного деления:

#include <iostream>

#include <cmath>

#include <ctime>

using namespace std;

double f(double x) {

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

}

int main() {

int s = clock();

int end;

float a, b, c;

float fA, fB, fC;

double eps = 0.001;

a = 0;

b = 2;

while (abs(b - a) > eps) {

c = (a + b) / 2;

fA = f(a);

fB = f(b);

fC = f(c);

if (fA \* fC < 0) {

b = c;

}

else if (fC \* fB < 0) {

a = c;

}

else {

cout << "ERROR" << endl;

return 1;

}

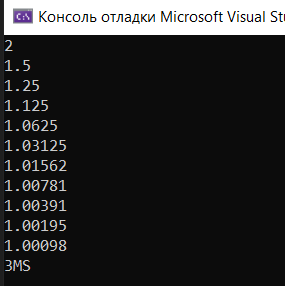
cout << b << endl;

}

end = clock();

cout << end << "MS" << endl;

return 0;

}

Метод итераций:

#include <iostream>

#include <cmath>

#include <ctime>

using namespace std;

int main() {

int s = clock();

int end;

double x, x1;

double eps = 0.00001;

x = 1.0;

x1 = 0.0;

while (abs(x - x1) > eps) {

x1 = x;

x = 1.2502 - 0.25 \* pow(x1, 3);

cout << x << endl;

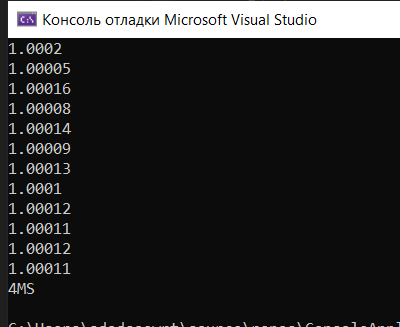
}

end = clock();

cout << end << "MS" << endl;

return 0;

}



Половинное деление (рекурсия)

#include <iostream>

#include <ctime>

#include <clocale>

#include <cmath>

using namespace std;

double eps = 1e-6;

double f(double x)

{

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

}

double del(double a, double b, double eps)

{

double x = (a + b) / 2;

if ((abs(f(a) - f(b)) <= eps))

{

if (abs(f(x)) <= eps)

{

return (a + b) / 2;

}

}

if (f(a) \* f(x) <= 0.0) {

del(a, x, eps);

}

else {

del(x, b, eps);

}

return 0;

}

int main()

{

setlocale(LC\_ALL, "Ru");

int s = clock();

double a = 0.0;

double b = 1.0;

cout << del(a, b, eps);

}

**Ссылка на GitHub**

<https://github.com/kirill2209/ConsoleApplication8>