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: 21**

Выполнил работу

студент группы ИВТ-24-2б

К.С.Фофанов

Проверил

Доцент кафедры ИТАС

О.А.Полякова

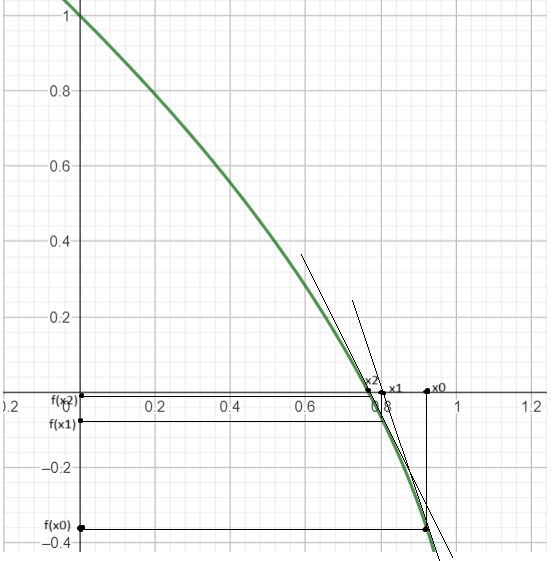
Пермь, 2024

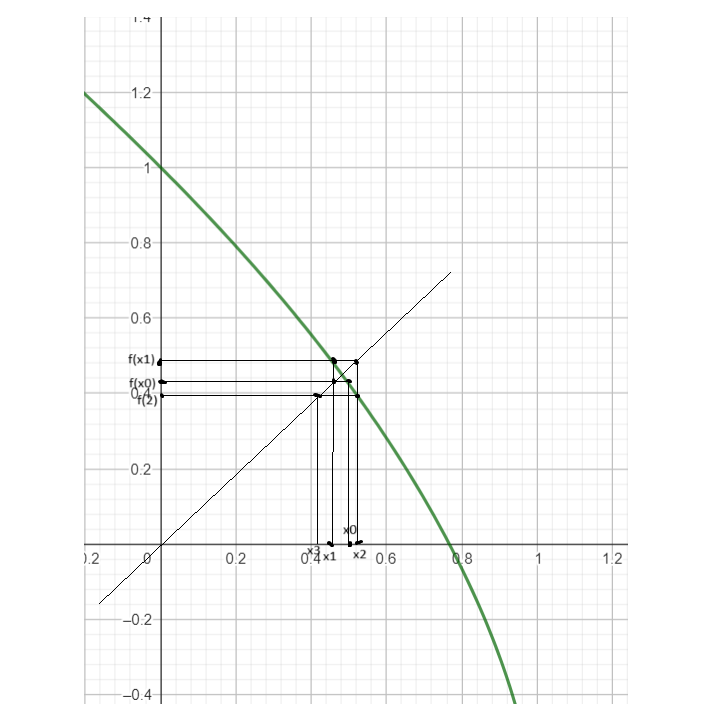
**Постановка задачи**

Решить нелинейное уравнение: используя три метода: Ньютона, итерации и половинного деления (Дихотомии). С заданной точность eps = 1e-6, на отрезке [0,1].

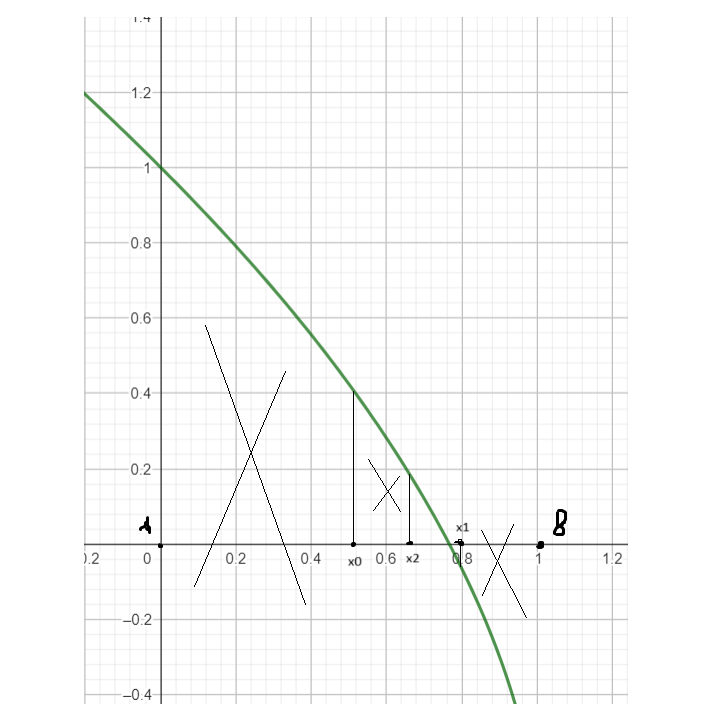
**Геометрическая интерпретация**

Метод Ньютона:



Метод итерации:

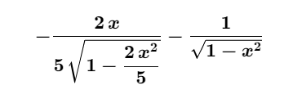
Метод Дихотомии:

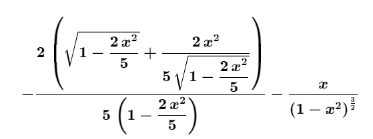


**Анализ задачи**

Метод Ньютона:

1. Функция: f(x) =
2. Производные функции:

* f’(x)=



* f”(x)=

1. Инициализация: берем начально значение х0 со стороны грани интервала, где выполняется условие f(x)\*f”(x)>0.
2. Итерации:

* Находим значения хn, по формуле: xn = x(n-1) – f(x(n-1))/f’(x(n-1))
* Проверяем Ix(n-1)-xnI<=eps, если корни не подходят, то алгоритм повторяем с третьего шага.

Метод итерации:

1. Функция: f(x) =
2. Выражаем х: g(x)=
3. Выбираем любое значение начального х из промежутка [0,1]
4. Итерации:

* Находим новое значение х1: xn = g(x(n-1))
* Продолжаем итерации, пока разница между значениями не станет меньше eps.

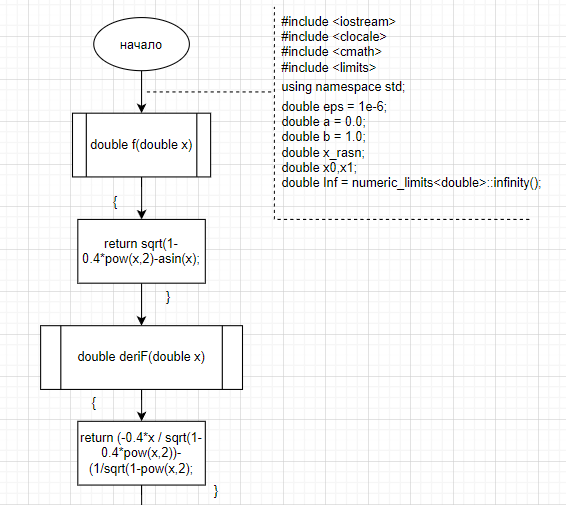
Метод Дихотомии:

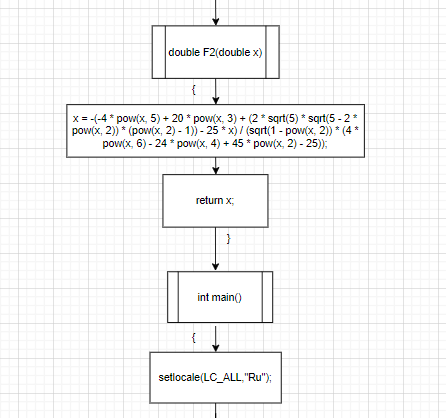
1. Функция: f(x) =
2. Инициализация: берем начальное х0 со значение середины отрезка [0,1]
3. Итерации:

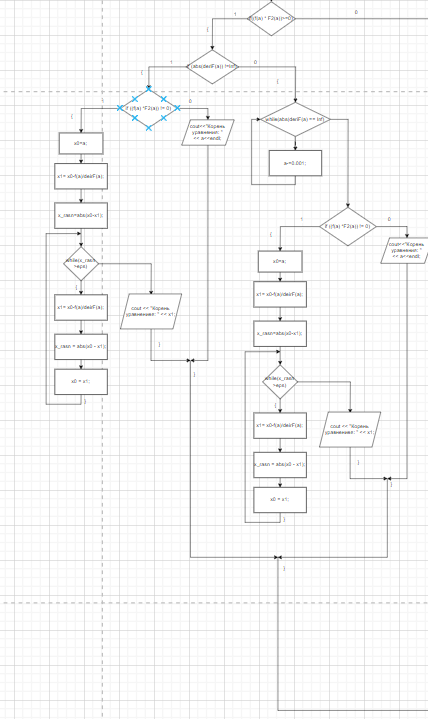
* Проверяем два интервала (а,х0) и (х0, b), если функция на концах интервала имеет разные знаки, то тот интервал где знаки одинаковы отбрасываем
* Отброс половины отрезка происходит путем перестановки границы либо а, либо b на очередную точку xn найденную по формуле: xn=(b-a)/2
* Окончание происходит, когда Ib-aI <= eps.

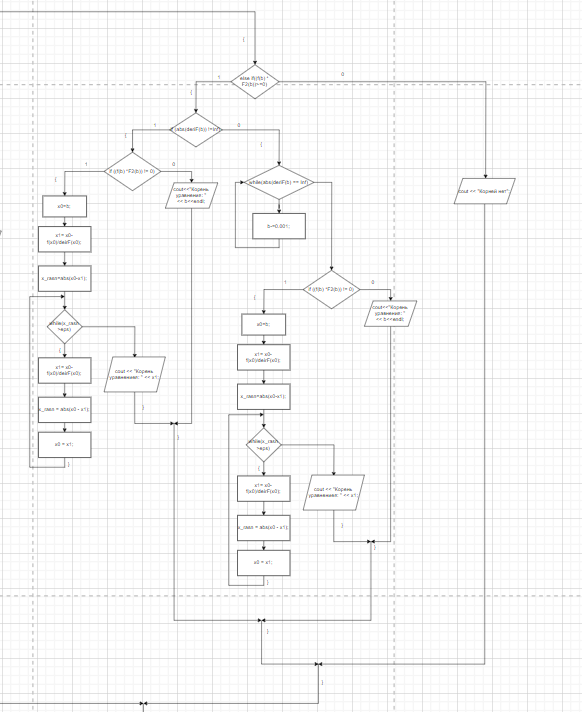
**Блок-схемы**

Метод Ньютона:

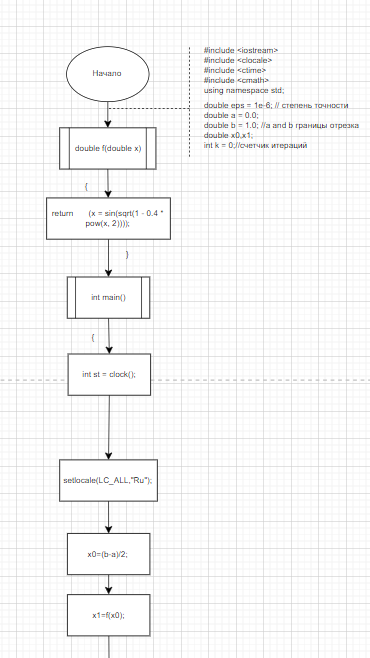


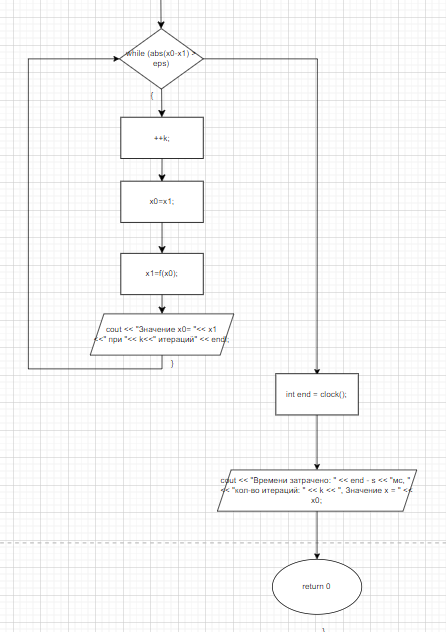




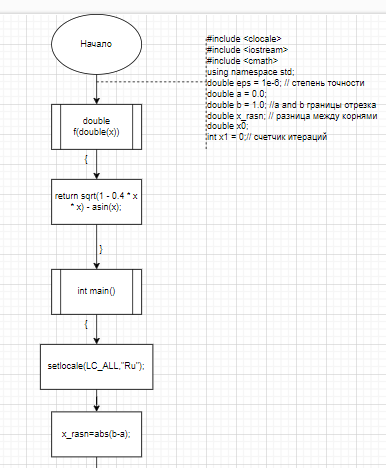


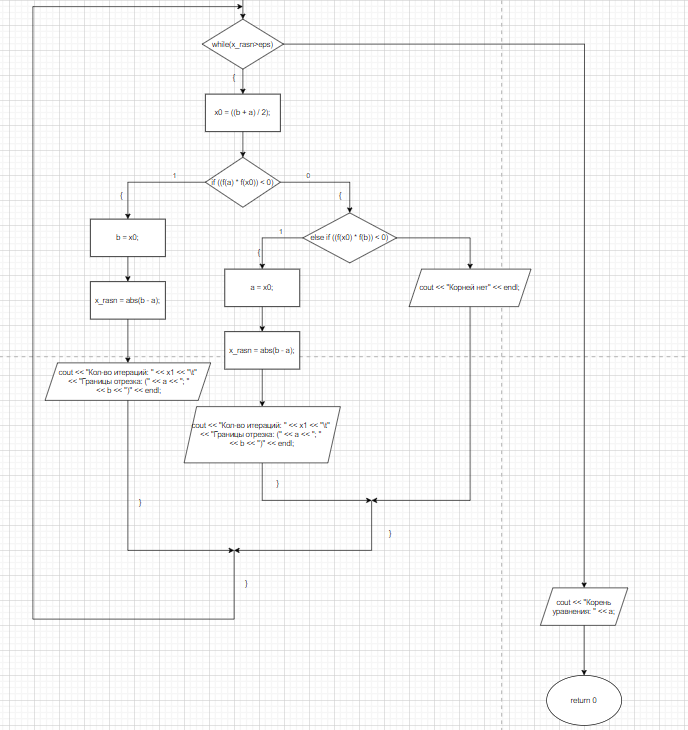
Метод итерации:





Метод Дихотомии:





**Код программы**

Метод Ньютона:

#include <iostream>

#include <clocale>

#include <cmath>

#include <limits>

#include <ctime>

using namespace std;

double F2(double x)

{

x = -(-4 \* pow(x, 5) + 20 \* pow(x, 3) + (2 \* sqrt(5) \* sqrt(5 - 2 \* pow(x, 2)) \* (pow(x, 2) - 1)) - 25 \* x) / (sqrt(1 - pow(x, 2)) \* (4 \* pow(x, 6) - 24 \* pow(x, 4) + 45 \* pow(x, 2) - 25));

return x;

}

double f(double x)

{

return sqrt(1 - 0.4 \* x \* x) - asin(x);

}

double deriF(double x)

{

return (-0.4 \* x / sqrt(1 - 0.4 \* x \* x)) - (1 / sqrt(1 - x \* x));

}

int main()

{

int st = clock();

setlocale(LC\_ALL, "Ru");

double eps = 1e-6; // степень точности

double a = 0.0;

double b = 1.0; //a and b границы отрезка

double x\_rasn; // разница между корнями

double x0, x1;

int k = 1;

int end;

double Inf = numeric\_limits<double>::infinity();

if ((f(a) \* F2(a)) >= 0)

{

if (abs(deriF(a)) != Inf)

{

if ((f(a) \* F2(a)) == 0)

{

cout << "Корень уравнения: " << a << endl;

}

else {

x0 = a;

x1 = x0 - f(a) / deriF(a);

x\_rasn = abs(x0 - x1);

while (x\_rasn <= eps)

{

++k;

x1 = x0 - f(x0) / deriF(x0);

x\_rasn = abs(x0 - x1);

x0 = x1;

cout << "Итерация: " << k << ". х1= " << x1 << endl;

}

end = clock();

cout <<"Количество итераций: "<<k<<". Время выполнения: "<< end-st<< ". Корень уравнениея: " << x1;

}

}

else {

while (abs(deriF(a)) == Inf)

{

a -= 0.001;

}

if ((f(a) \* F2(a)) == 0)

{

cout << "Корень уравнения: " << a << endl;

}

else

{

x0 = a;

x1 = x0 - f(a) / deriF(a);

x\_rasn = abs(x0 - x1);

while (x\_rasn > eps)

{

x1 = x0 - f(x0) / deriF(x0);

x\_rasn = abs(x0 - x1);

x0 = x1;

cout << "Итерация: " << k << ". х1= " << x1 << endl;

}

end = clock();

cout << "Количество итераций: " << k << ". Время выполнения: " << end - st << ". Корень уравнениея: " << x1;

}

}

}

else if ((f(b) \* F2(a)) >= 0)

{

if (abs(deriF(b)) != (Inf))

{

if ((f(b) \* F2(b)) == 0)

{

cout << "Корень уравнения: " << b;

}

else

{

x0 = b;

x1 = (x0 - f(b) / deriF(b));

x\_rasn = abs(x0 - x1);

while (x\_rasn > eps)

{

x1 = x0 - f(b) / deriF(b);

x0 = x1;

x\_rasn = abs(x0 - x1);

cout << "Итерация: " << k << ". х1= " << x1 << endl;

}

end = clock();

cout << "Количество итераций: " << k << ". Время выполнения: " << end - st << ". Корень уравнениея: " << x1;

}

}

else {

while (abs(deriF(b)) == Inf)

{

b -= 0.001;

}

if ((f(b) \* F2(b)) == 0)

{

cout << "Корень уравнения: " << b;

}

else

{

x0 = b;

x1 = (x0 - f(b) / deriF(b));

x\_rasn = abs(x0 - x1);

while (x\_rasn > eps)

{

++k;

x1 = x0 - (f(x0) / deriF(x0));

x\_rasn = abs(x0 - x1);

x0 = x1;

cout << "Итерация: " << k << ". х1= " << x1 << endl;

}

end = clock();

cout << "Количество итераций: " << k << ". Время выполнения: " << end - st<<"мс" << ". Корень уравнениея: " << x1;

}

}

}

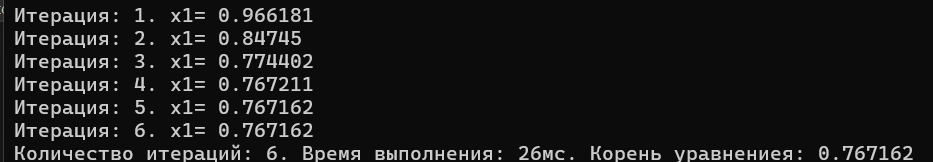
else {

cout << "Корней нет";

}

return 0;

}



Метод итерации:

#include <iostream>

#include <clocale>

#include <ctime>

#include <cmath>

using namespace std;

double f(double x)

{

return x = sin(sqrt(1 - 0.4 \* pow(x, 2)));

}

int main()

{

int s = clock();

setlocale(LC\_ALL, "Ru");

double eps = 1e-6; // степень точности

double a = 0.0;

double b = 1.0; //a and b границы отрезка

double x0,x1;

int k = 0;

x0 = (b-a)/2;

x1 = f(x0);

while (abs(x0-x1) > eps)

{

++k;

x0 = x1;

x1 = f(x0);

cout << "Значение х0= "<< x1<<" при "<< k<<" итераций" << endl;

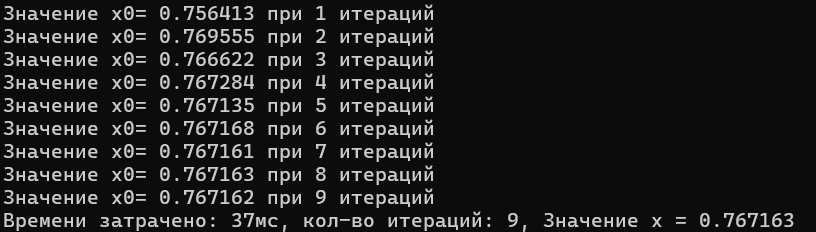
}

int end = clock();

cout << "Времени затрачено: " << end - s << "мс, " << "кол-во итераций: " << k << ", Значение x = " << x0;

return 0;

}



Метод Дихотомии:

#include <clocale>

#include <iostream>

#include <cmath>

#include <ctime>

using namespace std;

double f(double x)

{

return sqrt(1 - 0.4 \* x \* x) - asin(x);

}

int main()

{

int st = clock();

setlocale(LC\_ALL, "Ru");

double eps = 1e-6; // степень точности

double a = 0.0;

double b = 1.0; //a and b границы отрезка

double x\_rasn; // разница между корнями

double x0;

int x1 = 0;// счетчик итераций

int end;

x\_rasn = abs(b - a);

while (x\_rasn > eps)

{

++x1;

x0 = ((b + a) / 2);

;

if ((f(a) \* f(x0)) < 0)

{

b = x0;

x\_rasn = abs(b - a);

end =clock();

cout << "Итерация: " << x1 << "\t" << "Границы отрезка: (" << a << "; " << b << ")" << endl;

}

else if ((f(x0) \* f(b)) < 0)

{

a = x0;

x\_rasn = abs(b - a);

cout << "Итерация: " << x1 << "\t" << "Границы отрезка: (" << a << "; " << b << ")" << endl;

}

else

{

cout << "Корней нет" << endl;

}

}

end = clock();

cout << "Всего итераций: "<< x1<<". Время действия программы: "<< end-st<<"мс"<<". Корень равен = "<<a<<endl;

return 0;

}

