Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное

образовательное учреждение высшего образования

Санкт-Петербургский университет аэрокосмического приборостроения

КАФЕДРА № 2

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук		С.Л. Козенко
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

Нелинейные уравнения

по дисциплине: ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4134к	10.03.2023	Д.В. Шумков
		подпись, дата	инициалы, фамилия

Санкт – Петербург, 2023

Цель работы

а)Освоение методов решения нелинейных уравнений;

б)Совершенствование навыков по алгоритмизации и программированию вычислительных задач

Постановка задачи

Составить схему алгоритма и программу на языке С/С++ решения задачи по теме «Нелинейные уравнения»: $a^x - bxsin(x)$ методом хорд. a = 2.01, b = -1.

Математическая часть

Общие методические указания к численным методам решения нелинейных уравнений

Уравнением называется равенство

$$f(x) = 0, \qquad (2.1)$$

справедливое при некоторых значениях $x{=}x*$, называемыми корнями этого уравнения или нулями функции $f(x)$. Решение уравнения заключается в определении его корней. Среди корней $x*$ могут быть и комплексные, однако в данной работе вычисляются только действительные корни.

Вычисление каждого из действительных корней складывается из двух этапов:

1) отделение корня, т.е. нахождение возможно малого интервала $[a, b]$, в пределах которого находится один и только один корень $x*$ уравнения;

2) уточнение значения корня, т.е. вычисление с заданной степенью точности.

При использовании рассматриваемых ниже методов решения уравнения (2.1) к функции $f(x)$ на интервале $[a,b]$ предъявляются следующие требования:

а) функция $f(x)$ непрерывна и дважды дифференцируема (т.е. существует первая и вторая производные);

b) первая производная $f'(x)$ непрерывна, сохраняет знак и не обращается в нуль;

c) вторая производная $f''(x)$ непрерывна и сохраняет знак.

Отделение корней может производиться аналитическим или графическим способами. Аналитический способ основывается на теореме Коши, утверждающей, что для непрерывной функции $f(x)$ (первое требование "a"), принимающей на концах интервала $[a, b]$ разные знаки, т.е. $f(a)\cdot f(b)<0$, уравнение (2.1) имеет внутри этого интервала хотя бы один корень (рис. 1). Если к этому добавить второе требование "b", означающее монотонность функции $f(x)$, то этот корень оказывается единственным.

В этих условиях отделение корня сводится к вычислению значений функции $f(x)$ для последовательности точек α_1, α_2, …, α_n и сопоставлению знаков $f(\alpha_k)$, $f(\alpha_{k+1})$ в соседних точках α_k и α_{k+1}. Каждый интервал $[\alpha_k, \alpha_{k+1}]$, для которого $f(\alpha_k)\cdot f(\alpha_{k+1})<0$, содержит, по крайней мере, один корень уравнения. Этот корень является единственным, если на этом интервале выполняется второе требование "b". В противном случае следует интервал $[\alpha_k, \alpha_{k+1}]$ разделить на меньшие интервалы, повторяя для каждого из них указанные действия.

При использовании графического способа уравнение (2.1) можно также представить в виде

$$f_1(x) = f_2(x) \qquad (2.2)$$

и построить графики функций $y=f_1(x)$ и $y=f_2(x)$. Абсцисса точки пересечения этих графиков дает приближенное значение x^0 корня x^* уравнения

$$f(x) = f_1(x) - f_2(x) = 0.$$

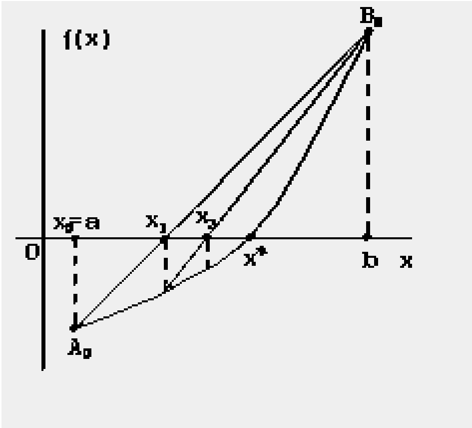
Представление уравнения (2.1) в форме (2.2) не является, естественно, однозначным и его следует подбирать так, чтобы построение графиков было возможно простым.

Из того же чертежа следует определить и тот интервал $[a, b]$, в пределах которого данный корень является единственным (если это необходимо для выбранного метода последующего уточнения значения корня x^0);

Метод хорд.

Пусть определен интервал $[a, b]$, в котором лежит один корень x^* уравнения $f(x)=0$.

Учитывая, что $f(a)\cdot f(b)<0$, определяем первое приближение как точку пересечения с осью абсцисс хорды A_0B_0, соединяющей точки $A_0[a, f(a)]$ и $B_0[b, f(b)]$



Для нахождения последующего приближения вычислим значение *f(x₁)* и сопоставим со значениями *f(a)* и *f(b).* Выберем тот из интервалов *[a,x₁]* или [x₁,b], на концах которого функция *f*(x) имеет разные знаки (именно внутри этого интервала лежит искомый корень *x∗*). Применим предыдущий прием к этому интервалу, получая последующее приближение – точку *x₂*.

Заметим, что производные *f'(x)* и *f''(x)* сохраняют положительный знак *(f'(x)>0, f''(x)>0; f'(x)·f''(x)>0)* и все приближения *x₁,x₂,…* образуют возрастающую последовательность, ограниченную значением x=x∗. Следовательно, $\lim\limits_{n\to\infty} x_n = x^*,$ и при этом в любом из приближений соответствующая хорда проходит через начальную точку *B₀[b,f(b)].*

Для получения формулы, определяющей последующие приближения, рассмотрим переход от x_n и x_{n+1}. В этом случае уравнение хорды B_nB_0 как прямой, проходящей через точки B_n,B_0, имеет вид

$$\frac{y-f(x_n)}{f(b)-f(x_n)}=\frac{x-x_n}{b-x_n}.$$

Если для определения x_{n+1} положить $y(x_{n+1})=0$, то получим

$$x_{n+1}=x_n-f(x_n)\frac{b-x_n}{f(b)-f(x_n)} \qquad (n=0,1,2,...)$$

Согласно требованиям:

a) функция *f*(*x*) непрерывна и дважды дифференцируема (т.е. существует первая и вторая производные);

b) первая производная *f*'(*x*) непрерывна, сохраняет знак и не обращается в нуль;

c) вторая производная *f*''(*x*) непрерывна и сохраняет знак.

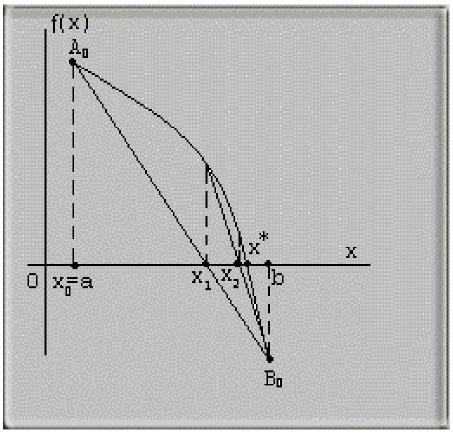
для оценки погрешностей вычислений используется неравенство:

$$|x_{n+1} - x^*| \le \frac{M-m}{m}|x_{n+1} - x_n|,$$ где $0 < m \le |f'(x)| \le M < 1$.

Если при этом $M \le 2m$, то $|x_{n+1}-x^*| \le |x_{n+1}-x_n|$, и для заданной погрешности ε вычисления прекращаются при $|x_{n+1}-x_n| \le \varepsilon$ (как это имело место и для методов последовательных приближений и метода касательных).

При выполнении упомянутых требований возможны и иные картины построений для метода хорд, определяемые сочетаниями знаков производных $f'(x)$ и $f''(x)$.

Рисунок выше соответствует рассмотренному случаю $f'(x)>0$, $f''(x)>0$ (функция $f(x)$ монотонно возрастает и выпукла вниз). Случай $f'(x), f''(x)<0$ приводит к аналогичным построениям, и последовательность $x_1, x_2, \ldots$ оказывается так же возрастающей.



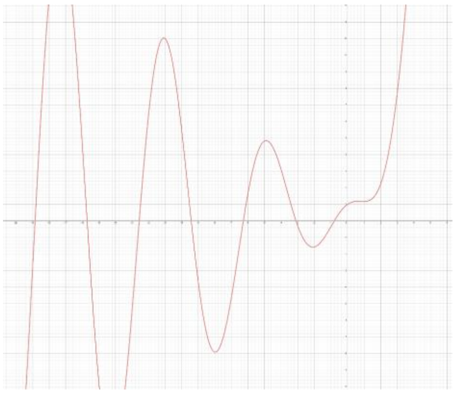
Однако в случаях $f'(x)>0$, $f''(x)<0$ и $f'(x)<0$, $f''(x)>0$ после определения каждого x_n различными оказываются знаки значений функций $f(a)$ и $f(x_n)$ (а не $f(x_n)$ и $f(b)$, как ранее). Поэтому "неподвижной" для всех хорд оказывается точка $A_0[a, f(a)]$ (а не $B_0[b, f(b)]$). В результате расчетными являются формулы

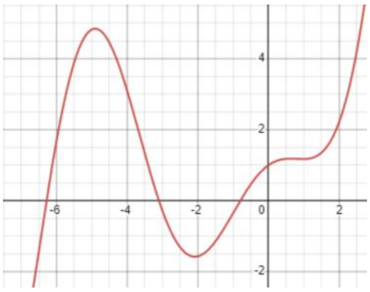
$$x_{n+1} = x_n - f(x_n)\frac{x_n - a}{f(x_n) - f(a)}$$ (n=0,1,2,…),

а последовательность $x_1, x_2, \ldots$ оказывается убывающей.

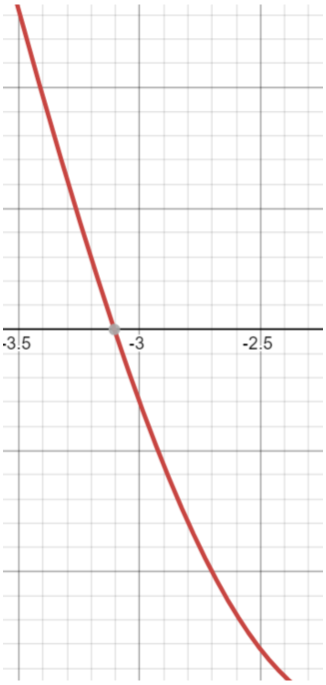
Аналитические расчеты

$f(x) = a^x - bx\sin(x)$, a = 2.01, b = -1





Возьмем интервал [-3,4; -2.5]



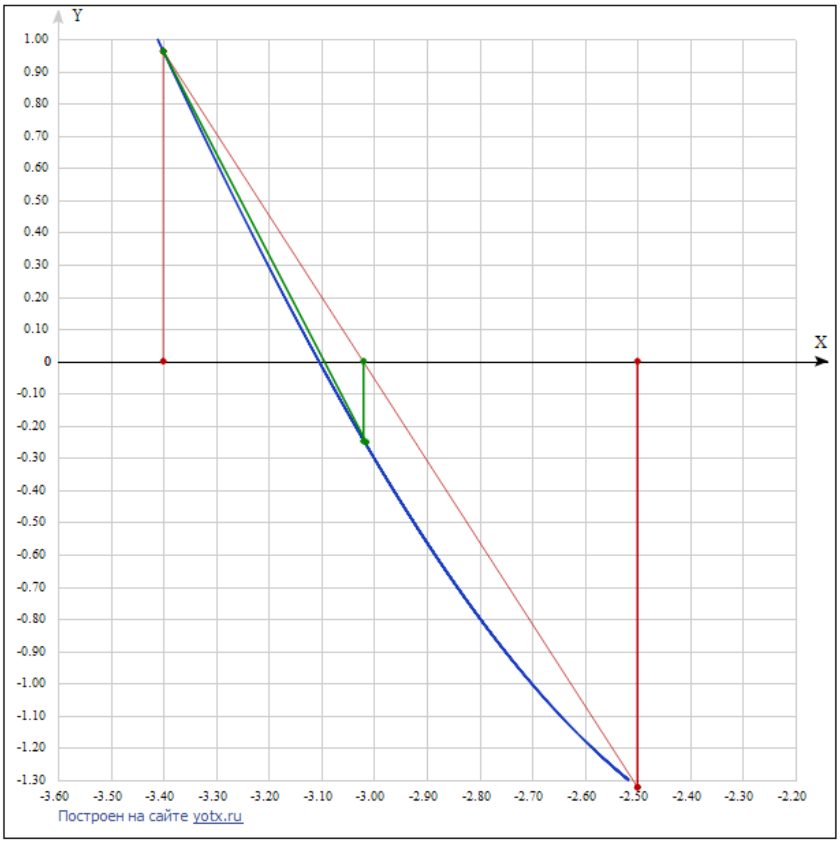
$f^{`}(x) = \ln(2.01) * 2.01^{x} - \sin(x) - xcos(x);$

$f^{``}(x) = \ln(2.01)^{2} * 2.01^{x} - 2\cos(x) + xsin(x);$

f(-3.4) = 2.22 > 0, f(-2,5) = -1.32 < 0 $\rightarrow$ корень на данном промежутке есть

при -3.5 ≤ x ≤ -2.5

-3.6 ≤ $f^{`}(x)$ ≤-1.3, 0.11 ≤ $f^{``}(x)$ ≤ 0.31.



Следовательно, $f^{`}(x) * f^{``}(x) < 0$ при -3.4 ≤ x ≤ -2.5 будем работать по такой формуле:

$$x_{n+1} = x_n - f(x_n)\frac{x_n - a}{f(x_n) - f(a)};$$

$$x_0 = b = -2.5;$$

$$x_1 = -2.5 - (-1.32)\frac{2.5-(-3.4)}{-1.32-2.22} = -2{,}9661;$$

$$x_2 = -2.9661 - (-0.39176)\frac{-2.9661-(-3.4)}{-0.39176-2.22} = -3.08368;$$

$$x_3 = -3.08368 - (-0.06233)\frac{-3.08368-(-3.4)}{-0.06233-2.22} = -3.10186;$$

$$x_4 = -3.10186 - (-0.00852)\frac{-3.10186-(-3.4)}{-0.00852-2.22} = -3.10434;$$

$$x_5 = -3.10434 - (-0.00112)\frac{-3.10434-(-3.4)}{-0.00112-2.22} = -3.10467;$$

$$x_6 = -3.10467 - (-0.00014)\frac{-3.10467-(-3.4)}{-0.00014-2.22} = -3.10471;$$

$$x_7 = -3.10471 - (-0.00002)\frac{-3.10471-(-3.4)}{-0.00002-2.22} = -3.10472;$$

Так как $|x_7 - x_6| = |-3.10472 - 3.10471| \leq 10^{-5}$, то за приближенное значение корня следует принять $x \approx * x_7 = -3.10472$(Точное значение корня x = -3.105).

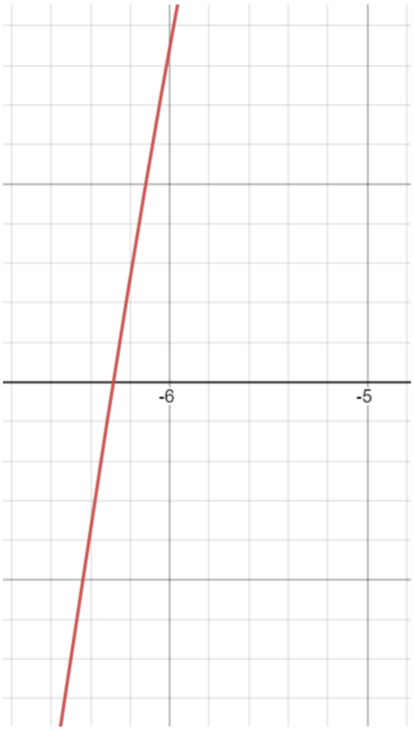
Проверим тоже самое на интервале [-6.5;-5.5]

13.1175 ≤ f`(x) ≤ 6.40839

-0.054 ≤ f``(x) ≤ 0.528

Первая и вторая производные разных знаков, поэтому работаем по формуле

$x_{n+1} = x_n - f(x_n)\frac{x_n - a}{f(x_n) - f(a)}$;



$x_0 = b = -5.5$;

$x_1 = -5.5 - 3.90197\frac{-5.5+6.5}{3.90197+1.38758} = -6.23768$;

$x_2 = -6.23768 - 0.2966\frac{-6.23768+6.5}{0.2966+1.38758} = -6.28388$;

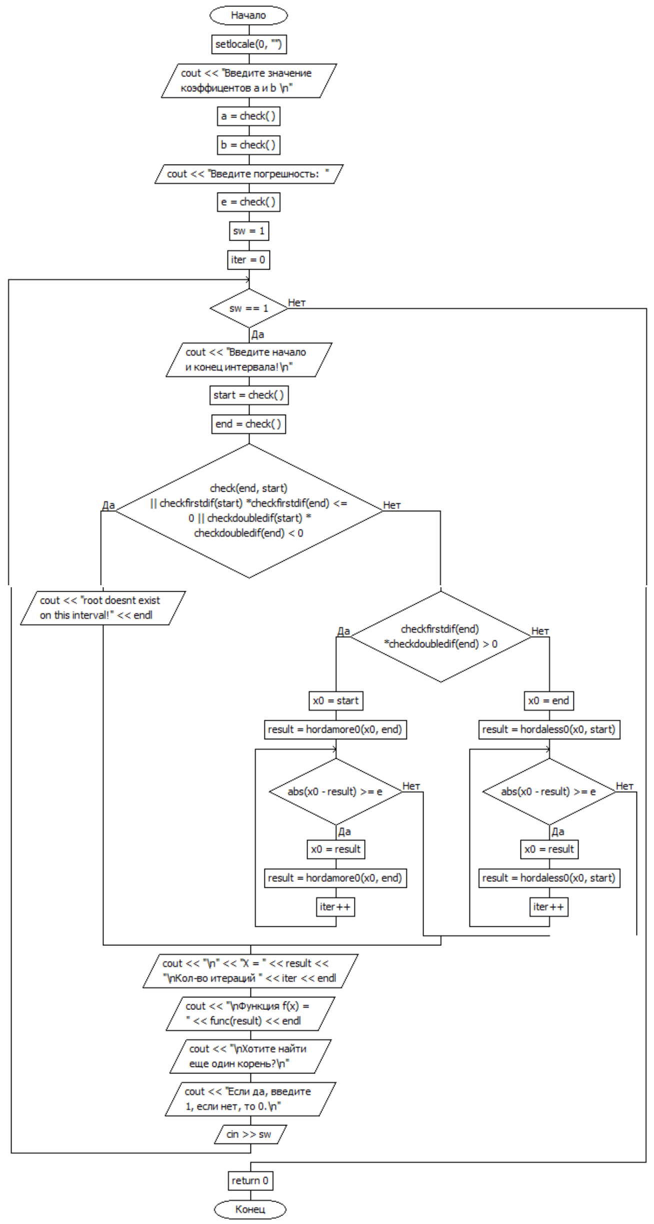
$x_3 = -6.28388 - 0.00807\frac{-6.28388+6.5}{0.00807+1.38758} = -6.28513$;

$x_4 = -6.28513 - 0.0002\frac{-6.28513+6.5}{0.0002+1.38758} = -6.28516$;

$x_5 = -6.28513 - 0.00002\frac{-6.28516+6.5}{0.00002+1.38758} = -6.28516$;

$|x_5 - x_4|$ = |-6.28516 + 6.28516| ≤ 10^{-5}, то за приближенное значение корня следует принять $x \approx * x_5 = -6.28516$(Точное значение корня x = -6.285)

Блок схема:



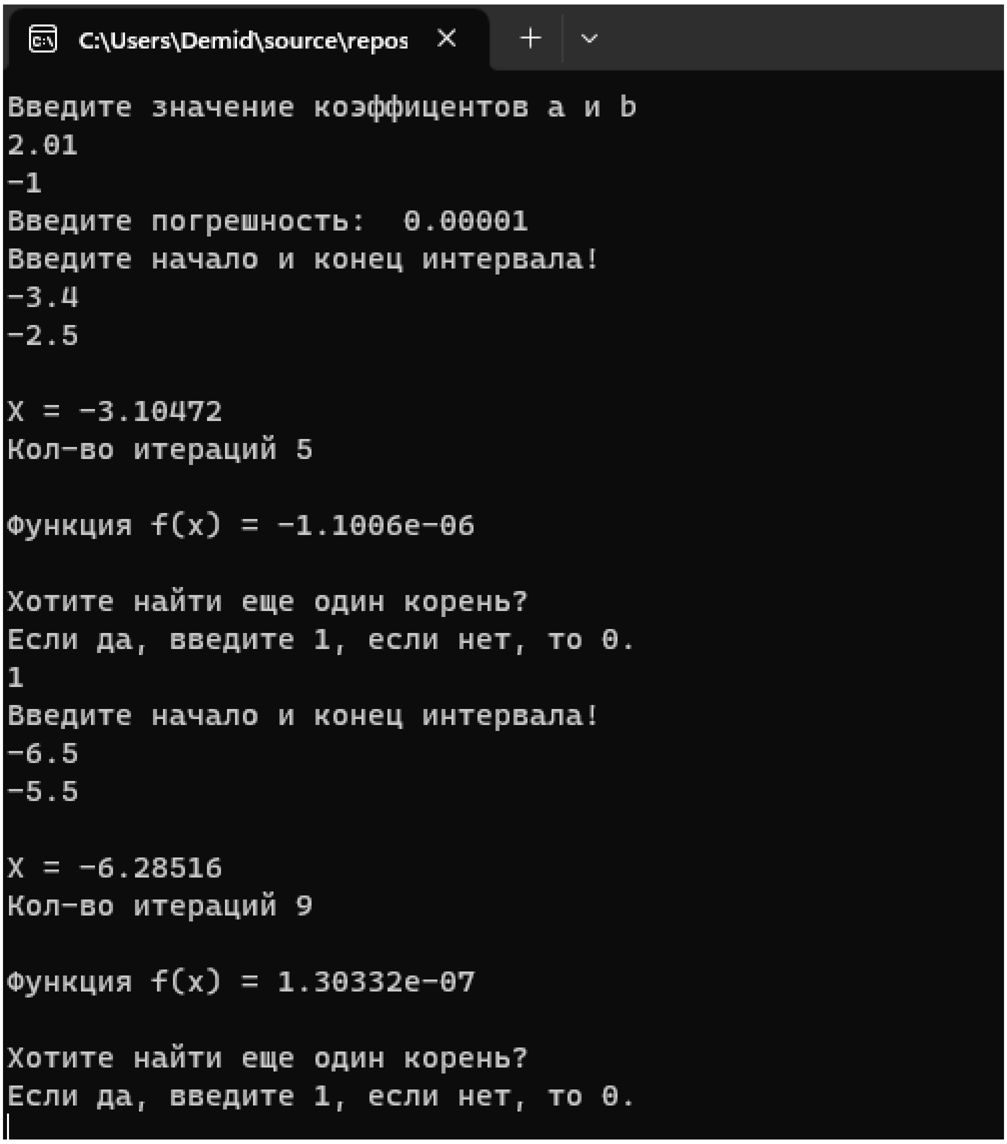
Код:

```
#include <iostream>
#include <cmath>
using namespace std;

double a, b;

double check() {
    double n;
    cin >> n;
    while (cin.fail() || cin.get() != '\n') {
        cin.clear();
        cin.ignore(cin.rdbuf()->in_avail());
        cout << "Error!";
        cin >> n;
    }
    return n;
}

double func(double x)
```

```
{
    double f;
    f = pow(a, x) + b * x * sin(x);
    return f;
}

double hordamore0(double x, double end)
{
    double f;
    f = x - func(x) * ((end - x) / (func(end) - func(x)));
    return f;
}

double hordaless0(double x, double start)
{
    double f;
    f = x - func(x) * ((x - start) / (func(x) - func(start)));
    return f;
}

bool check(double end, double start)
{
    if (end <= start || func(end) * func(start) > 0)
        return true;
    else
        return false;
}

double checkfirstdif(double x)
{
    double h = 0.1;
    double f;
    f = (func(x + h) - func(x - h)) / h;
    return f;

}

double checkdoubledif(double x)
{
    double h = 0.1;
    double f;
    f = (func(x + h) - 2 * func(x) + func(x - h)) / h;
    return f;
}

int main()
{
    setlocale(0, "ru");
    double result;
    double start, end;
    double x0, x, e;
    cout << "Введите значение коэффицентов a и b \n";
    a = check();
    b = check();
    cout << "Введите погрешность:  ";
    e = check();
    int sw = 1;
    int iter = 0;

    while (sw == 1)
    {
        cout << "Введите начало и конец интервала!\n";
        start = check();
        end = check();
        if (check(end, start) || checkfirstdif(start) * checkfirstdif(end) <= 0 ||
checkdoubledif(start) * checkdoubledif(end) < 0)
        {
```

```
            cout << "root doesnt exist on this interval!" << endl;
        }

        else
        {
            if (checkfirstdif(end) * checkdoubledif(end) > 0)
            {
                x0 = start;
                result = hordamore0(x0, end);
                while (abs(x0 - result) >= e)
                {
                    x0 = result;
                    result = hordamore0(x0, end);
                    iter++;
                }
            }
            else
            {
                x0 = end;
                result = hordaless0(x0, start);
                while (abs(x0 - result) >= e)
                {
                    x0 = result;
                    result = hordaless0(x0, start);
                    iter++;
                }
            }
        }



        cout << "\n" << "X = " <<  result << "\nКол-во итераций " << iter << endl;

        cout << "\nФункция f(x) = " << func(result) << endl;
        cout << "\nХотите найти еще один корень?\n";
        cout << "Если да, введите 1, если нет, то 0.\n";
        cin >> sw;

    }

    return 0;
}
```

Скриншоты работы программы:



Значения полученные аналитически не отличаются от значений, полученных программным путем.

Вывод

В ходе работы был освоен метод хорд для решения нелинейных уравнений и усовершенствованы навыки по алгоритмизации и программированию вычислительных задач.