Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное

образовательное учреждение высшего образования

Санкт-Петербургский университет аэрокосмического приборостроения

ГУАП

КАФЕДРА № 2

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Кандидат тех. Наук, доцент |  |  |  | С.Л Козенко |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №1 |
| по дисциплине: Вычислительная математика |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Костяков НА |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

**Цели работы:** а) освоение методов решения нелинейных уравнений; б)совершенствование навыков по алгоритмизации и программированию вычислительных задач.

**Вариант 4**



**Математический анализ**

**Уравнением называется равенство**

**(*x*) = 0**

справедливое при некоторых значениях *x=x*, называемыми корнями этого уравнения или нулями функции (*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 (*x*) на интервале *a*,*b* предъявляются следующие требования:

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

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

c) вторая производная "(*x*) непрерывна и сохраняет знак.

Отделение корней может производиться аналитическим или графическим способами. Аналитический способ основывается на теореме Коши, утверждающей, что для непрерывной функции (*x*) (первое требование “a”), принимающей на концах интервала *a*, *b* разные знаки, т.е. **(*a*)**(*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 **(*x*), то этот корень оказывается единственным.

В этих условиях отделение корня сводится к вычислению значений функции **(*x*) для последовательности точек **1, **2, …, *n* и сопоставлению знаков (*k*), **(*k*+1) в соседних точках *k* и *k*+1. Каждый интервал *k , k*+1, для которого **(*k*) **(*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 *k* , *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

**1(*x*) =**2(*x*) (2.2)

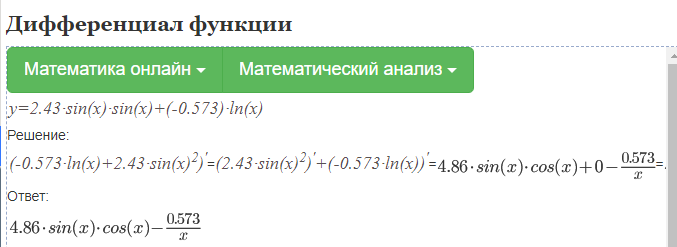
и построить графики функций *y*=**1(*x*) и *y*=**2(*x*)*.* Абсцисса точки пересечения этих графиков дает приближенное значение *x*0 корня *x* уравнения

**(*x*) =**1(*x*) - **2(*x*) = 0.

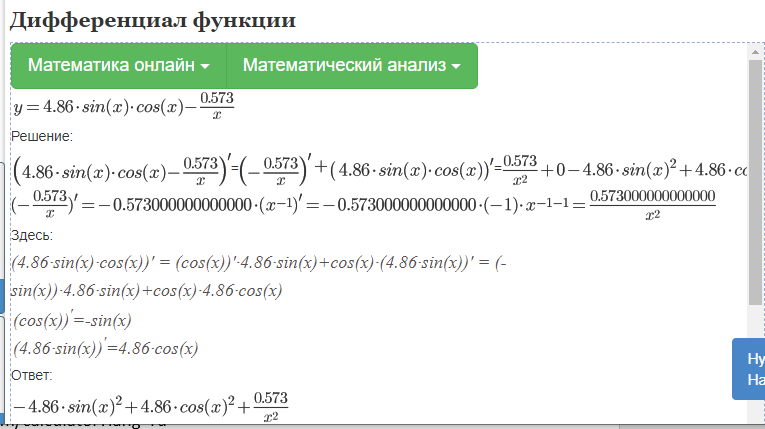
Представление уравнения (2.1) в форме (2.2) не является, естественно, однозначным и его следует подбирать так, чтобы построение графиков было возможно простым.

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);

**Первая производная**

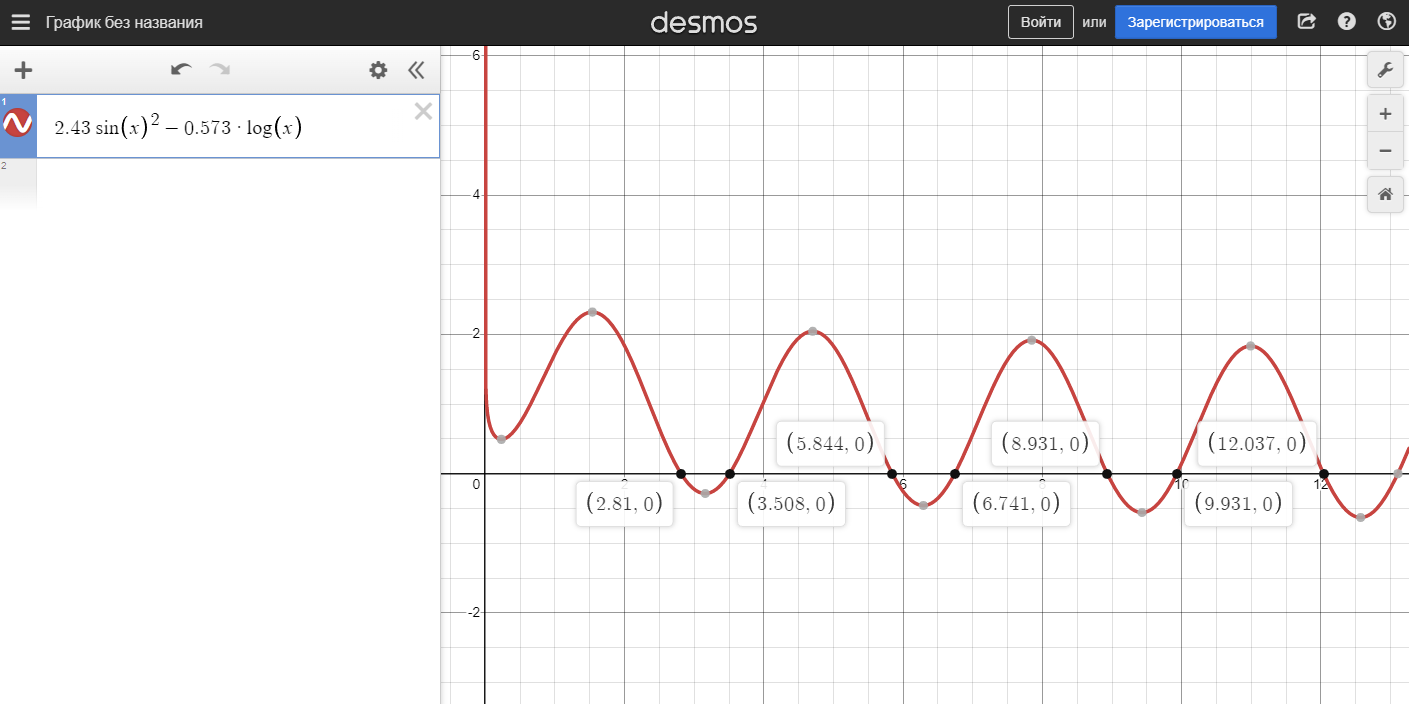


**Вторая производная**



**График**

https://www.desmos.com/calculator?lang=ru



**Алгоритм комбинированного поиска корней нелинейного уравнения можно описать следующим образом:**

1 Выбрать начальное приближение корня уравнения.

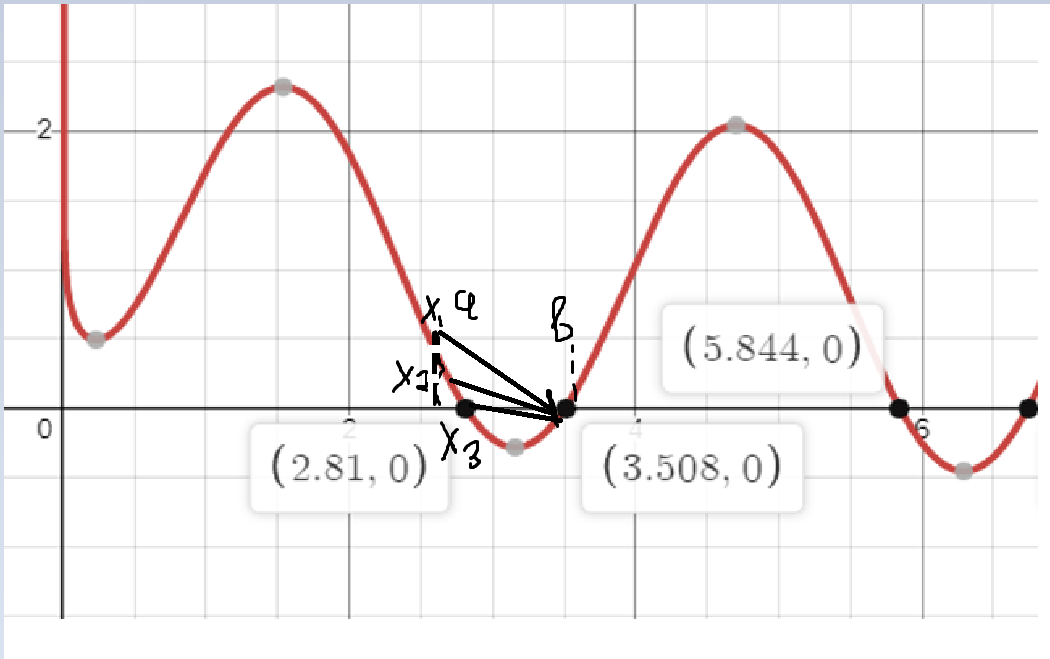
2 Применить один из методов поиска корней (например, метод Ньютона или метод бисекции) для нахождения корня с заданной точностью.

3 Если метод не сошелся к корню, то применить другой метод поиска корней.

4 Если и этот метод не сошелся к корню, то вернуть ошибку.

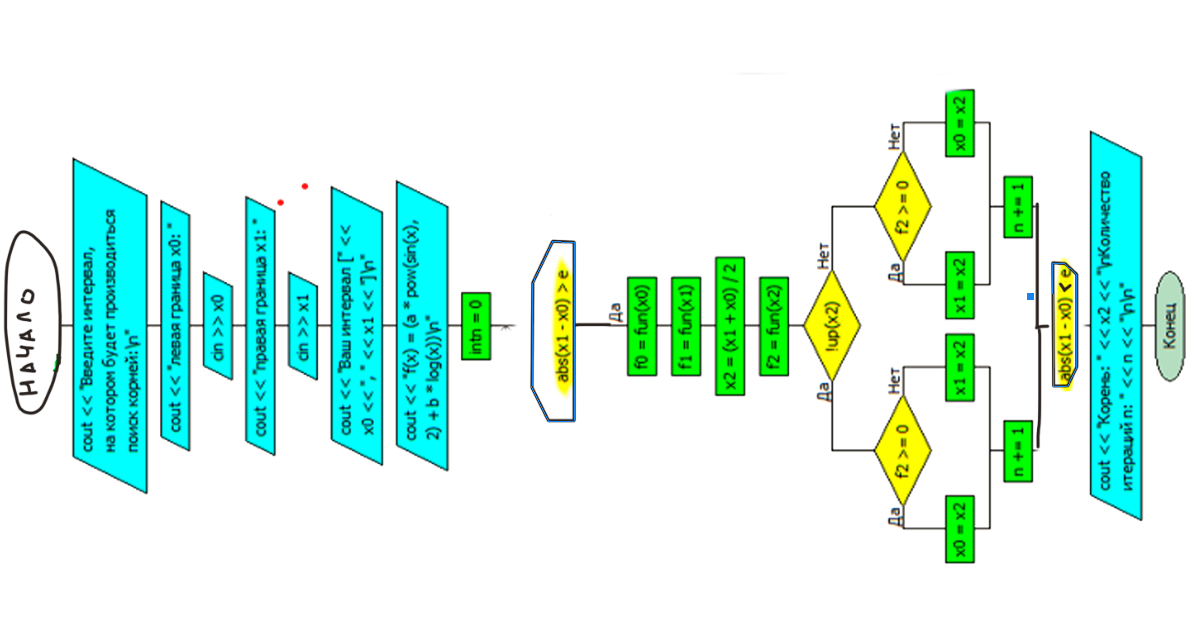
5 Повторять шаги 2-4 до тех пор, пока не будет найден корень с заданной точностью.

**Графическая визуализация метода**

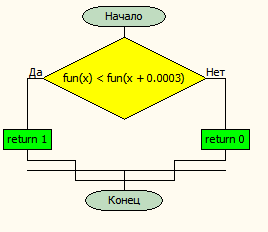
****

**Схемы алгоритмов:**

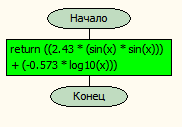
**Int main()**



**Double fun()**



**Bool up()**



**Листинг**

#include <iostream>

#include <cmath>

double fun(double x, double a, double b) {

return ((a \* (sin(x)\*sin(x))) + (b \* log10(x)));

}

bool up(double x, double a, double b, double e) {

if (fun(x, a, b) < fun(x + e, a, b)) {

return 1;

}

else {

return 0;

}

}

using namespace std;

int main()

{

//продолжение

setlocale(LC\_ALL, "Russian");

double e = 0.0003;

double x0, x1, x2, a, b;

double f2;

int n=0;

while (1)

{

cout << "Введите A: ";

cin >> a;

cout << "Ввелите B: ";

cin >> b;

cout << "Введите Епсилон: ";

cin >> e;

cout << "Введите интервал, на котором будет производиться поиск корней:\n";

cout << "левая граница x0: ";

cin >> x0;

cout << "правая граница x1: ";

cin >> x1;

cout << "Ваш интервал [" << x0 << ", " << x1 << "]\n";

cout << "f(x) = (a \* pow(sin(x), 2) + b \* log(x))\n";

while (abs(x1 - x0) > e) {

double f0 = fun(x0, a, b); //левая часть

double f1 = fun(x1, a, b); //правая часть

/\* if (f0 \* f1 > 0) {

cout << "Ошибка, функция не меняет знак. Метод биссекции может быть некорретным\n";

break;

}\*/

x2 = (x1 + x0) / 2;

f2 = fun(x2, a, b); //середина

if (!up(x2, a, b, e))

{

if (f2 >= 0) {

x0 = x2;

}

else

{

x1 = x2;

}

n += 1;

}

else

{

if (f2 >= 0) {

x1 = x2;

}

else

{

x0 = x2;

}

n += 1;

}

}

cout << "Корень: " << x2 << " F(x)= " << f2 << "\nКоличество итераций n: " << n << "\n\n";

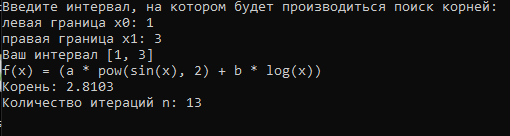
}

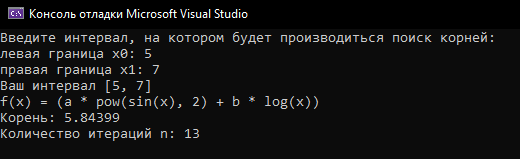
cout << "\nДля продолжения нажмите Enter\n";

getchar();

}

**Результат работы программы**

****

****

**Сравнение результатов**

Если сравнивать результаты моей программы и калькулятора десмоса, можно заметить, что мои результаты более точные.

**Вывод**

Я освоил комбинированный метод решения уравнений, совершенствовал свои навыки программирования на c++