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 № 1

Нелинейные уравнения

Вариант 5

по дисциплине: ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4128	25.02.2023	В.А. Воробьёв
		подпись, дата	инициалы, фамилия

Санкт – Петербург, 2023

СОДЕРЖАНИЕ

1 Цели и постановка задачи

1.1 Цели работы

а) Освоение методов решения нелинейных уравнений;

б) Совершенствование навыков по алгоритмизации и программированию вычислительных задач.

1.2 Задание

Составить схему алгоритма и программу на языке C/C++ решения задачи по теме «Нелинейные уравнения» в соответствии с индивидуальным заданием.

Уравнение	***Метод численного решения, точность***	***Параметры***
$\sqrt{a-x^2}+bx^2=0$	Дихотомии $\varepsilon = 4\cdot10^{-5}$	*a = 1.23; b = - 3.14*

Рисунок 1 – Вариант задания

2 Описание метода решения

Метод половинного деления (метод дихотомии)

Пусть в уравнении функция $f(x)$ является непрерывной (первое требование ”а”) на интервале $[a, b]$, в котором расположен один искомый корень x^*. Для нахождения этого корня разделим отрезок $[a, b]$ пополам точкой

$$\varepsilon = \frac{a+b}{2}.$$

Если теперь $f(\xi_1)=0$, то ξ_1 и является корнем уравнения. В противном случае выбираем тот из отрезков $[a, \xi_1]$ или $[\xi_1, b]$, на концах которого функция $f(x)$ имеет разные знаки. В пределах этого отрезка согласно предыдущим рассуждениям лежит искомый корень. Таким образом, оказывается определенным интервал $[a_1, b_1]$, (где $a_1=a$, $b_1=\xi_1$ или $a_1=\xi_1$, $b_1=b$), меньший первоначального $[a, b]$ и содержащий x^*. Повторяя подобные построения, получаем последовательность уменьшающихся интервалов $[a_n, b_n]$ таких, что

$$b_n - a_n = \frac{1}{2^n}(b-a)$$

и в каждом из которых заключен корень x^*. Точность вычисления корня x^* определяется размерами интервала $[a_n, b_n]$ после n-го деления исходного интервала $[a, b]$, так как ошибка определения корня x^* не превышает величины $b_n - a_n$.

Следовательно, если ε есть заданная точность вычисления, то должно выполняться условие

$$\frac{1}{2^n}(b-a) \le \varepsilon.$$

Отсюда можно определить и необходимое число шагов половинного деления интервала $[a, b]$, если задано ε:

$$n \ge \frac{\log \frac{b-a}{\varepsilon}}{\log 2} \approx 3.32 \log \frac{b-a}{\varepsilon}$$

3 Аналитические расчеты

Сначала строим график нашей функции (рис. 2).

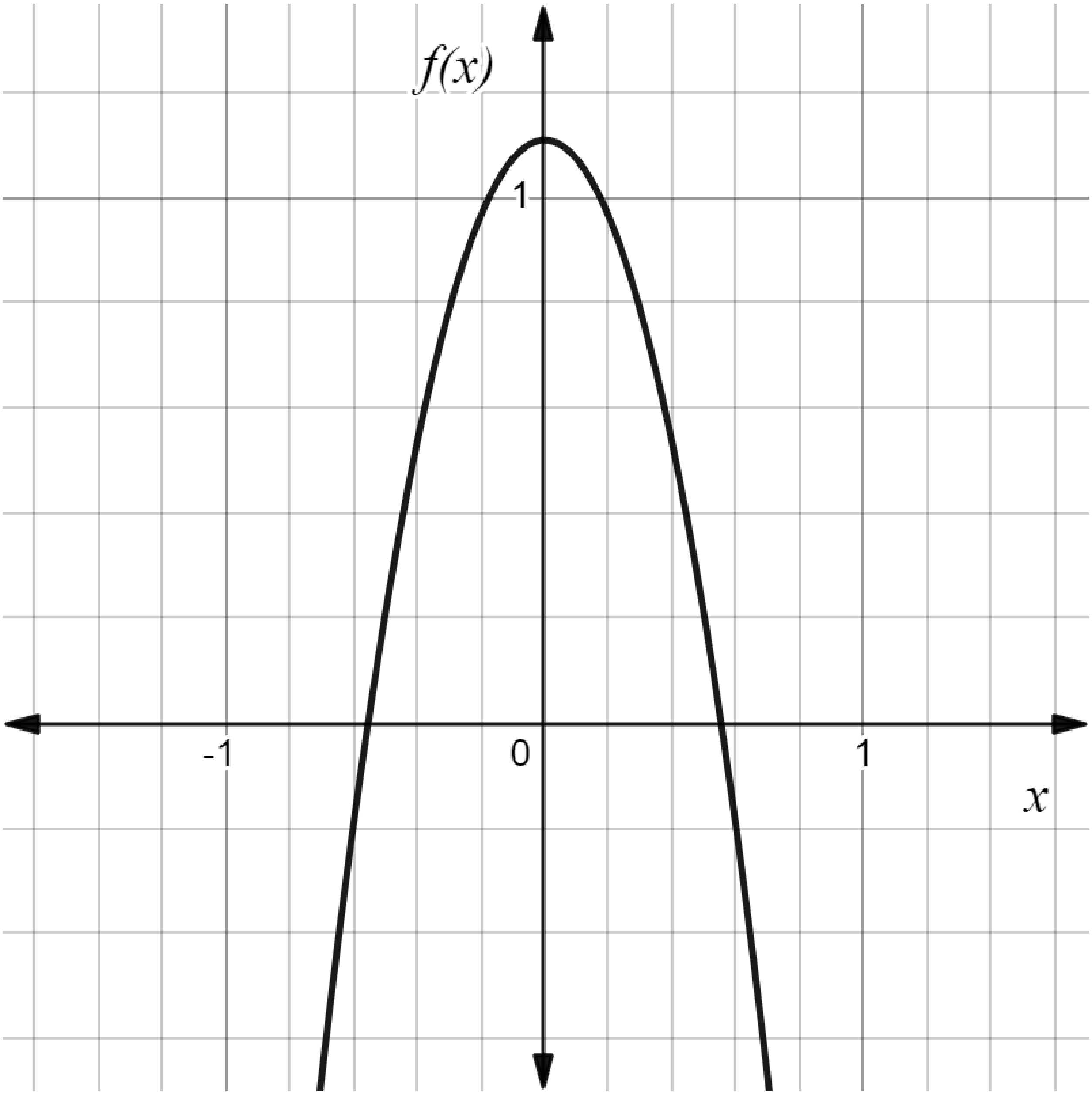


Рисунок 2 – График исходной функции

На графике видно, что f(x) пересекает ось X с в двух промежутках: [-1;0] и [0;1]. Проведем расчеты для этих промежутков с помощью онлайн калькулятора:

3.1 Расчеты для корня на промежутке [-1;0]

Найдем корни уравнения:

$\sqrt{1.23-x\cdot x}-3.14\cdot x\cdot x=0$

ε = 0.00004

Используем для этого **Метод половинного деления (метод дихотомии).**.

Считаем, что отделение корней произведено и на интервале [a,b] расположен один корень, который необходимо уточнить с погрешностью ε.

Итак, имеем f(a)f(b)<0. Метод дихотомии заключается в следующем.

Определяем половину отрезка $c={}^1/_2(a+b)$ и вычисляем f(c). Проверяем следующие условия:

1. Если |f(c)| < ε, то c – корень. Здесь ε - заданная точность.
2. Если f(c)f(a)<0, то корень лежит в интервале [a,c].
3. Если f(c)f(b)<0, то корень лежит на отрезке[c,b].

Продолжая процесс половинного деления в выбранных подынтервалов, можно дойти до сколь угодно малого отрезка, содержащего корень ξ.

Так как за каждую итерацию интервал, где расположен корень уменьшается в два раза, то через n итераций интервал будет равен:

$b_n-a_n={}^1/_2{}^n(b-a)$

В качестве корня ξ. возьмем ${}^1/_2(a_n+b_n)$. Тогда погрешность определения корня будет равна $(b_n – a_n)/2$. Если выполняется условие:

$(b_n – a_n)/2 < ε$

то процесс поиска заканчивается и $ξ = {}^1/_2(a_n+b_n)$.

Решение.

Число шагов, необходимых для достижения заданной точности определяется неравенством:

$h\geq(log_2\frac{b-a}{\epsilon})+1=(log_2(25000))+1=15$

F(-1)=-2.66; F(0)=1.109

Поскольку F(-1)*F(0)<0 (т.е. значения функции на его концах имеют противоположные знаки), то корень лежит в пределах [-1;0].

Итерация 1.

Находим середину отрезка: c = (-1 + 0)/2 = -0.5

F(x) = 0.205

F(c) = -2.66

Поскольку F(c)•F(a) < 0, то b=-0.5

Итерация 2.

Находим середину отрезка: c = (-1 -0.5)/2 = -0.75

F(x) = -0.949

F(c) = 0.205

Поскольку F(c)•F(b) < 0, то a=-0.75

Итерация 3.

Находим середину отрезка: c = (-0.75 -0.5)/2 = -0.625

F(x) = -0.31

F(c) = -0.949

Поскольку F(c)•F(b) < 0, то a=-0.625

Рисунок 3 – решение с помощью онлайн-калькулятора

Итерация 4.

Находим середину отрезка: c = (-0.625 -0.5)/2 = -0.563

F(x) = -0.0377

F(c) = -0.31

Поскольку F(c)•F(b) < 0, то a=-0.563

Остальные расчеты сведем в таблицу.

N	c	a	b	f(c)	f(x)	ε
1	-0.5	-0.5	0	-2.6604	0.2049	0.5
2	-0.75	-0.75	-0.5	0.2049	-0.9492	0.25
3	-0.625	-0.625	-0.5	-0.9492	-0.3104	0.125
4	-0.5625	-0.5625	-0.5	-0.3104	-0.03769	0.0625
5	-0.5313	-0.5313	-0.5	-0.03769	0.08735	0.03125
6	-0.5469	-0.5469	-0.5313	0.08735	0.02576	0.01563
7	-0.5547	-0.5547	-0.5469	0.02576	-0.00573	0.00781
8	-0.5508	-0.5508	-0.5469	-0.00573	0.01007	0.00391
9	-0.5527	-0.5527	-0.5508	0.01007	0.00218	0.00195
10	-0.5537	-0.5537	-0.5527	0.00218	-0.00177	0.000977
11	-0.5532	-0.5532	-0.5527	-0.00177	0.000207	0.000488
12	-0.5535	-0.5535	-0.5532	0.000207	-0.000782	0.000244
13	-0.5533	-0.5533	-0.5532	-0.000782	-0.000288	0.000122
14	-0.5533	-0.5533	-0.5532	-0.000288	-4.0E-5	6.1E-5

Таким образом, в качестве корня можно принять:

x=(-0.5533-0.5532)/2 = -0.5533

Ответ:x = -0.5533; F(x) = -4.0E-5

Рисунок 4 – решение с помощью онлайн-калькулятора

Итерация 4.

Находим середину отрезка: c = (-0.625 -0.5)/2 = -0.563

F(x) = -0.0377

F(c) = -0.31

Поскольку F(c)•F(b) < 0, то a=-0.563

Остальные расчеты сведем в таблицу.

N	c	a	b	f(c)	f(x)	ε
1	-0.5	-0.5	0	-2.6604	0.2049	0.5
2	-0.75	-0.75	-0.5	0.2049	-0.9492	0.25
3	-0.625	-0.625	-0.5	-0.9492	-0.3104	0.125
4	-0.5625	-0.5625	-0.5	-0.3104	-0.03769	0.0625
5	-0.5313	-0.5313	-0.5	-0.03769	0.08735	0.03125
6	-0.5469	-0.5469	-0.5313	0.08735	0.02576	0.01563
7	-0.5547	-0.5547	-0.5469	0.02576	-0.00573	0.00781
8	-0.5508	-0.5508	-0.5469	-0.00573	0.01007	0.00391
9	-0.5527	-0.5527	-0.5508	0.01007	0.00218	0.00195
10	-0.5537	-0.5537	-0.5527	0.00218	-0.00177	0.000977
11	-0.5532	-0.5532	-0.5527	-0.00177	0.000207	0.000488
12	-0.5535	-0.5535	-0.5532	0.000207	-0.000782	0.000244
13	-0.5533	-0.5533	-0.5532	-0.000782	-0.000288	0.000122
14	-0.5533	-0.5533	-0.5532	-0.000288	-4.0E-5	6.1E-5

Таким образом, в качестве корня можно принять:

x=(-0.5533-0.5532)/2 = -0.5533

Ответ:x = -0.5533; F(x) = -4.0E-5

Рисунок 5 – решение с помощью онлайн-калькулятора

3.2 Расчеты для корня на промежутке [0;1]

Найдем корни уравнения:

$\sqrt{1.23-x\cdot x}-3.14\cdot x\cdot x=0$

ε = 0.00004

Используем для этого **Метод половинного деления (метод дихотомии).**.

Считаем, что отделение корней произведено и на интервале [a,b] расположен один корень, который необходимо уточнить с погрешностью ε.

Итак, имеем f(a)f(b)<0. Метод дихотомии заключается в следующем.

Определяем половину отрезка $c={}^{1}/_{2}(a+b)$ и вычисляем f(c). Проверяем следующие условия:

1. Если |f(c)| < ε, то c **–** корень. Здесь ε - заданная точность.
2. Если f(c)f(a)<0, то корень лежит в интервале [a,c].
3. Если f(c)f(b)<0, то корень лежит на отрезке[c,b].

Продолжая процесс половинного деления в выбранных подынтервалов, можно дойти до сколь угодно малого отрезка, содержащего корень ξ.

Так как за каждую итерацию интервал, где расположен корень уменьшается в два раза, то через n итераций интервал будет равен:

$b_n-a_n={}^{1}/_{2^n}(b-a)$

В качестве корня ξ. возьмем $^{1}/_{2}(a_n+b_n)$. Тогда погрешность определения корня будет равна $(b_n – a_n)/2$. Если выполняется условие:

$(b_n – a_n)/2 < ε$

то процесс поиска заканчивается и $ξ = {}^{1}/_{2}(a_n+b_n)$.

Решение.

Число шагов, необходимых для достижения заданной точности определяется неравенством:

$h\geq(log_2\frac{b-a}{\epsilon})+1=(log_2(25000))+1=15$

F(0)=1.109; F(1)=-2.66

Поскольку F(0)*F(1)<0 (т.е. значения функции на его концах имеют противоположные знаки), то корень лежит в пределах [0;1].

Итерация 1.

Находим середину отрезка: c = (0 + 1)/2 = 0.5

F(x) = 0.205

F(c) = 1.109

Поскольку F(c)•F(b) < 0, то a=0.5

Итерация 2.

Находим середину отрезка: c = (0.5 + 1)/2 = 0.75

F(x) = -0.949

F(c) = 0.205

Поскольку F(c)•F(a) < 0, то b=0.75

Итерация 3.

Находим середину отрезка: c = (0.5 + 0.75)/2 = 0.625

F(x) = -0.31

F(c) = -0.949

Поскольку F(c)•F(a) < 0, то b=0.625

Рисунок 6 – решение с помощью онлайн-калькулятора

Итерация 4.

Находим середину отрезка: c = (0.5 + 0.625)/2 = 0.563

F(x) = -0.0377

F(c) = -0.31

Поскольку F(c)•F(a) < 0, то b=0.563

Остальные расчеты сведем в таблицу.

N	c	a	b	f(c)	f(x)	ε
1	0.5	0.5	1	1.1091	0.2049	0.5
2	0.75	0.75	1	0.2049	-0.9492	0.25
3	0.625	0.625	0.75	-0.9492	-0.3104	0.125
4	0.5625	0.5625	0.625	-0.3104	-0.03769	0.0625
5	0.5313	0.5313	0.5625	-0.03769	0.08735	0.03125
6	0.5469	0.5469	0.5625	0.08735	0.02576	0.01563
7	0.5547	0.5547	0.5625	0.02576	-0.00573	0.00781
8	0.5508	0.5508	0.5547	-0.00573	0.01007	0.00391
9	0.5527	0.5527	0.5547	0.01007	0.00218	0.00195
10	0.5537	0.5537	0.5547	0.00218	-0.00177	0.000977
11	0.5532	0.5532	0.5537	-0.00177	0.000207	0.000488
12	0.5535	0.5535	0.5537	0.000207	-0.000782	0.000244
13	0.5533	0.5533	0.5535	-0.000782	-0.000288	0.000122
14	0.5533	0.5533	0.5533	-0.000288	-4.0E-5	6.1E-5

Таким образом, в качестве корня можно принять:

x=(0.5532+0.5533)/2 = 0.5533

Ответ:x = 0.5533; F(x) = -4.0E-5

Рисунок 7 – решение с помощью онлайн-калькулятора

4 Схема алгоритма решения задачи

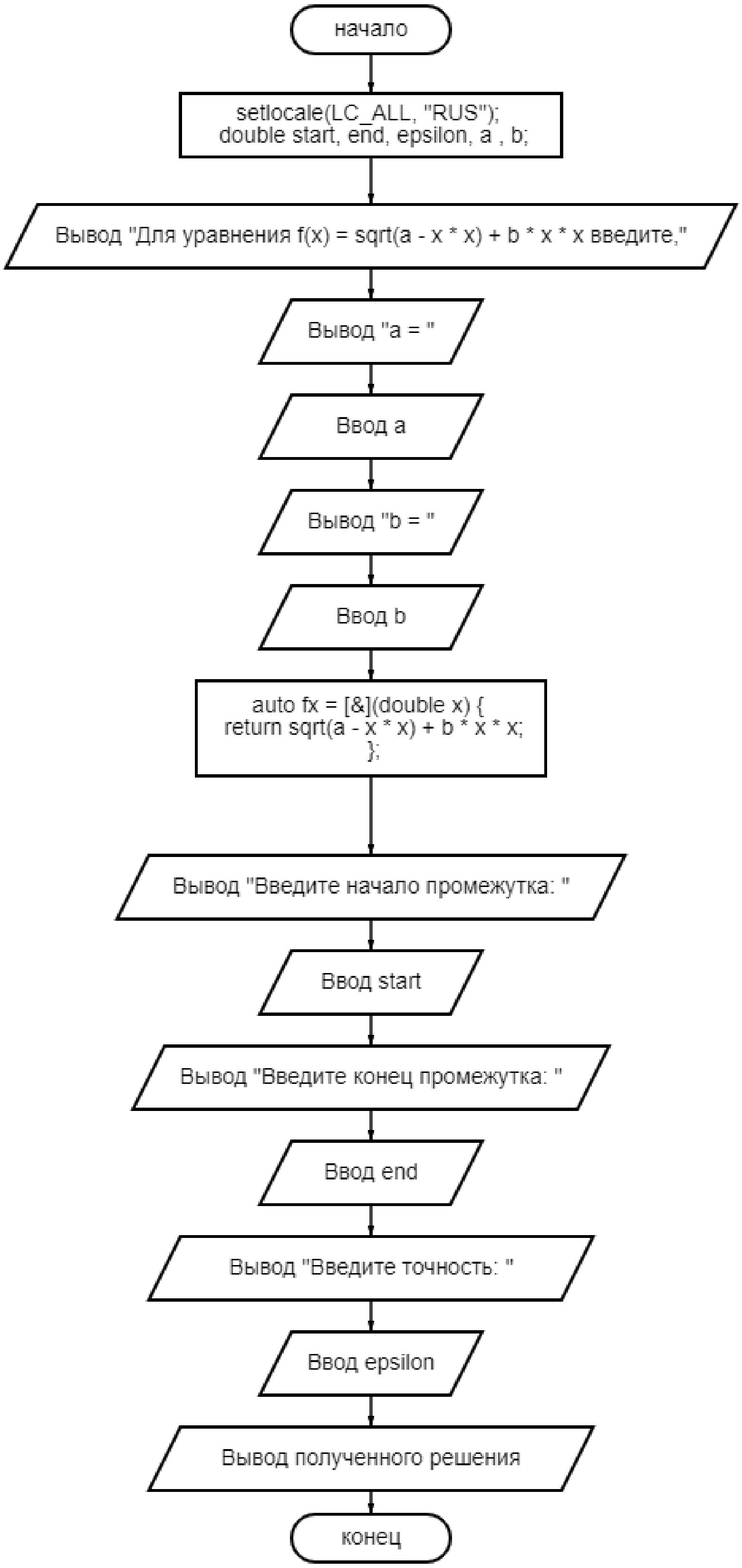


Рисунок 8 – Блок-схема основной функции

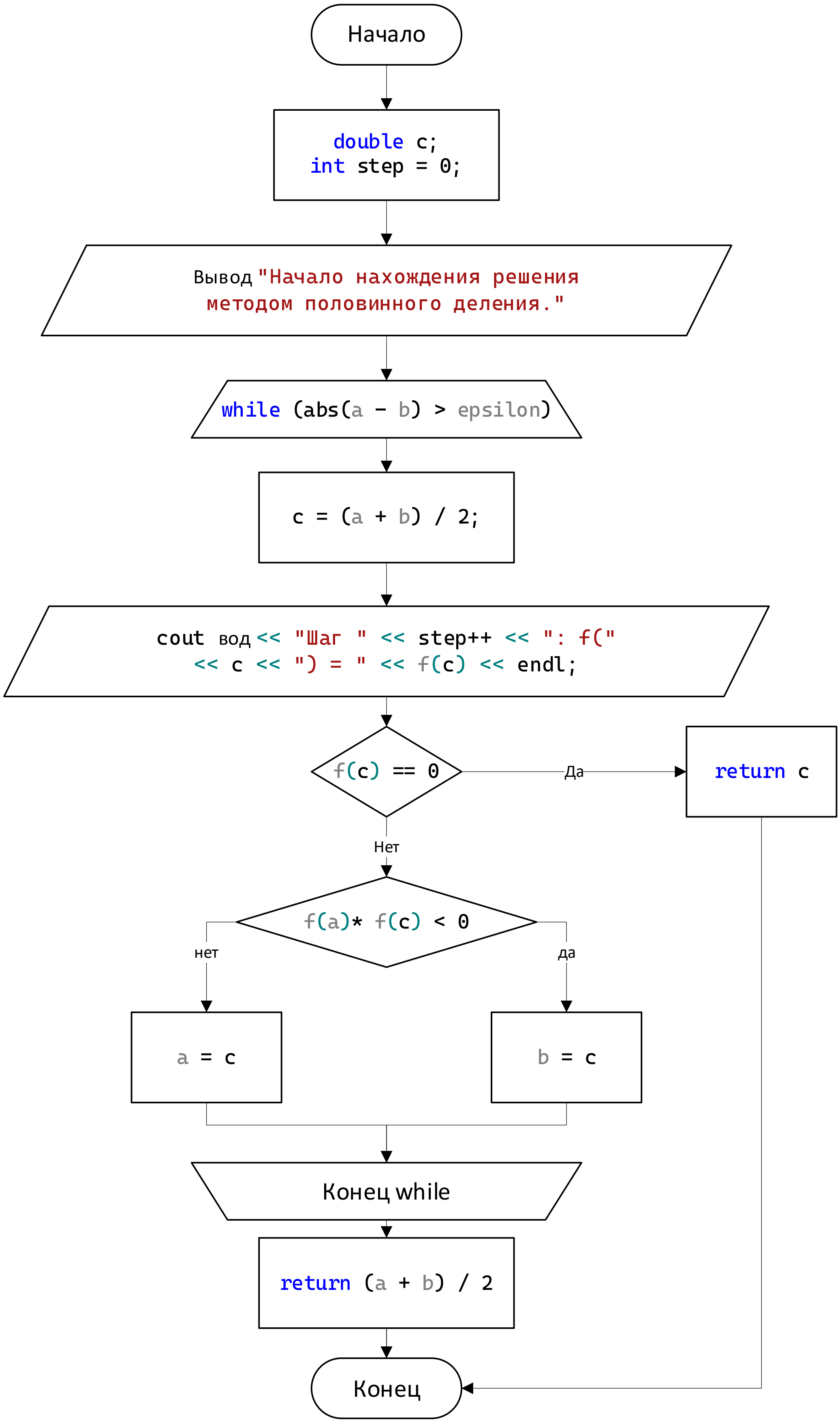


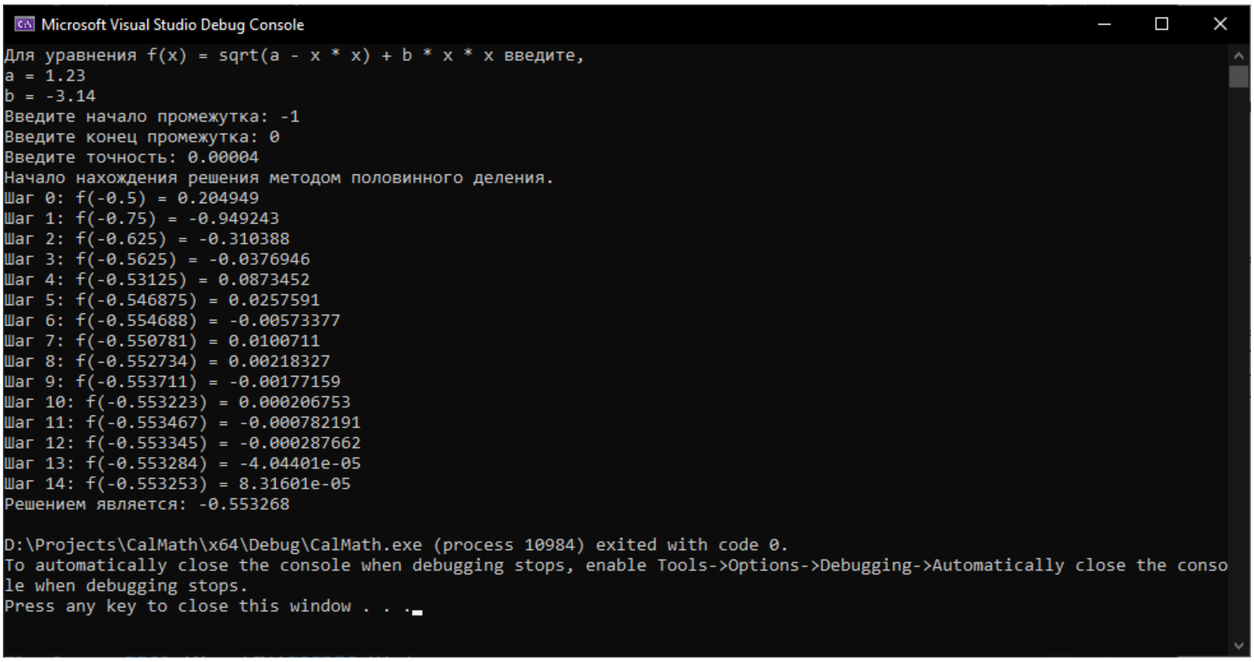
Рисунок 9 – Блок-схема dichotomy функции

5 Текст программы на C++

Исходный код доступен на GitHub: https://github.com/vladcto/SUAI_homework/blob/a635fedd450ca83705278b949d5cc37749177647/4_semester/ComputationalMathematics/program.cpp

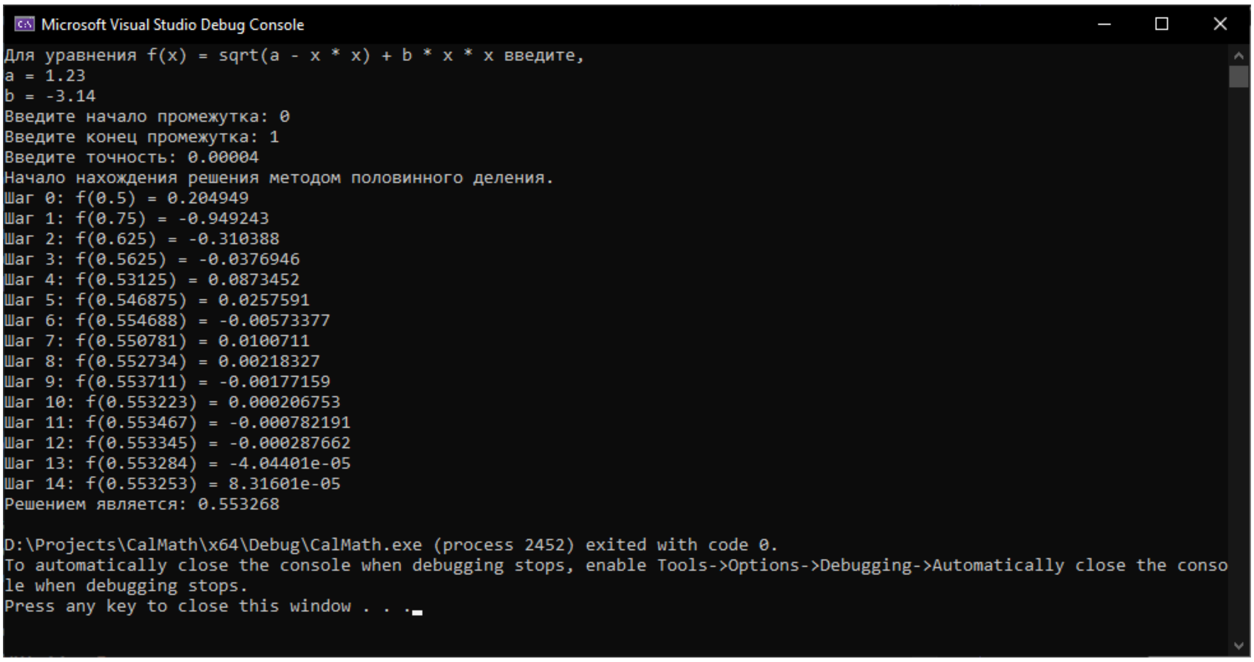
```cpp
#include <iostream>
#include <cmath>
#include <functional>

using namespace std;

double dichotomy(std::function<double(double)> f ,double a, double b, double
epsilon) {
	double c;
	int step = 0;
	cout << "Начало нахождения решения методом половинного деления." << endl;
	while (abs(a - b) > epsilon) {
		c = (a + b) / 2;
		cout << "Шаг " << step++ << ": f(" << c << ") = " << f(c) << endl;
		if (f(c) == 0) return c;
		f(a)* f(c) < 0 ? b = c : a = c;
	}
	return (a + b) / 2;
}

int main() {
	setlocale(LC_ALL, "RUS");
	double start, end, epsilon, a , b;

	cout << "Для уравнения f(x) = sqrt(a - x * x) + b * x * x введите," << endl;
	cout << "a = ";
	cin >> a;
	cout << "b = ";
	cin >> b;
	auto fx = [&](double x) {
		return sqrt(a - x * x) + b * x * x;
	};

	cout << "Введите начало промежутка: ";
	cin >> start;
	cout << "Введите конец промежутка: ";
	cin >> end;
	cout << "Введите точность: ";
	cin >> epsilon;
	cout << "Решением является: " << dichotomy(fx, start, end, epsilon) << endl;
	return 0;
}
```

6 Результаты программных расчетов



```
Microsoft Visual Studio Debug Console
Для уравнения f(x) = sqrt(a - x * x) + b * x * x введите,
a = 1.23
b = -3.14
Введите начало промежутка: -1
Введите конец промежутка: 0
Введите точность: 0.00004
Начало нахождения решения методом половинного деления.
Шаг 0: f(-0.5) = 0.204949
Шаг 1: f(-0.75) = -0.949243
Шаг 2: f(-0.625) = -0.310388
Шаг 3: f(-0.5625) = -0.0376946
Шаг 4: f(-0.53125) = 0.0873452
Шаг 5: f(-0.546875) = 0.0257591
Шаг 6: f(-0.554688) = -0.00573377
Шаг 7: f(-0.550781) = 0.0100711
Шаг 8: f(-0.552734) = 0.00218327
Шаг 9: f(-0.553711) = -0.00177159
Шаг 10: f(-0.553223) = 0.000206753
Шаг 11: f(-0.553467) = -0.000782191
Шаг 12: f(-0.553345) = -0.000287662
Шаг 13: f(-0.553284) = -4.04401e-05
Шаг 14: f(-0.553253) = 8.31601e-05
Решением является: -0.553268

D:\Projects\CalMath\x64\Debug\CalMath.exe (process 10984) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Рисунок 10 – Результат работы программы с предоставленными исходными данными



```
Microsoft Visual Studio Debug Console
Для уравнения f(x) = sqrt(a - x * x) + b * x * x введите,
a = 1.23
b = -3.14
Введите начало промежутка: 0
Введите конец промежутка: 1
Введите точность: 0.00004
Начало нахождения решения методом половинного деления.
Шаг 0: f(0.5) = 0.204949
Шаг 1: f(0.75) = -0.949243
Шаг 2: f(0.625) = -0.310388
Шаг 3: f(0.5625) = -0.0376946
Шаг 4: f(0.53125) = 0.0873452
Шаг 5: f(0.546875) = 0.0257591
Шаг 6: f(0.554688) = -0.00573377
Шаг 7: f(0.550781) = 0.0100711
Шаг 8: f(0.552734) = 0.00218327
Шаг 9: f(0.553711) = -0.00177159
Шаг 10: f(0.553223) = 0.000206753
Шаг 11: f(0.553467) = -0.000782191
Шаг 12: f(0.553345) = -0.000287662
Шаг 13: f(0.553284) = -4.04401e-05
Шаг 14: f(0.553253) = 8.31601e-05
Решением является: 0.553268

D:\Projects\CalMath\x64\Debug\CalMath.exe (process 2452) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Рисунок 11 – Результат работы программы с произвольными исходными данными

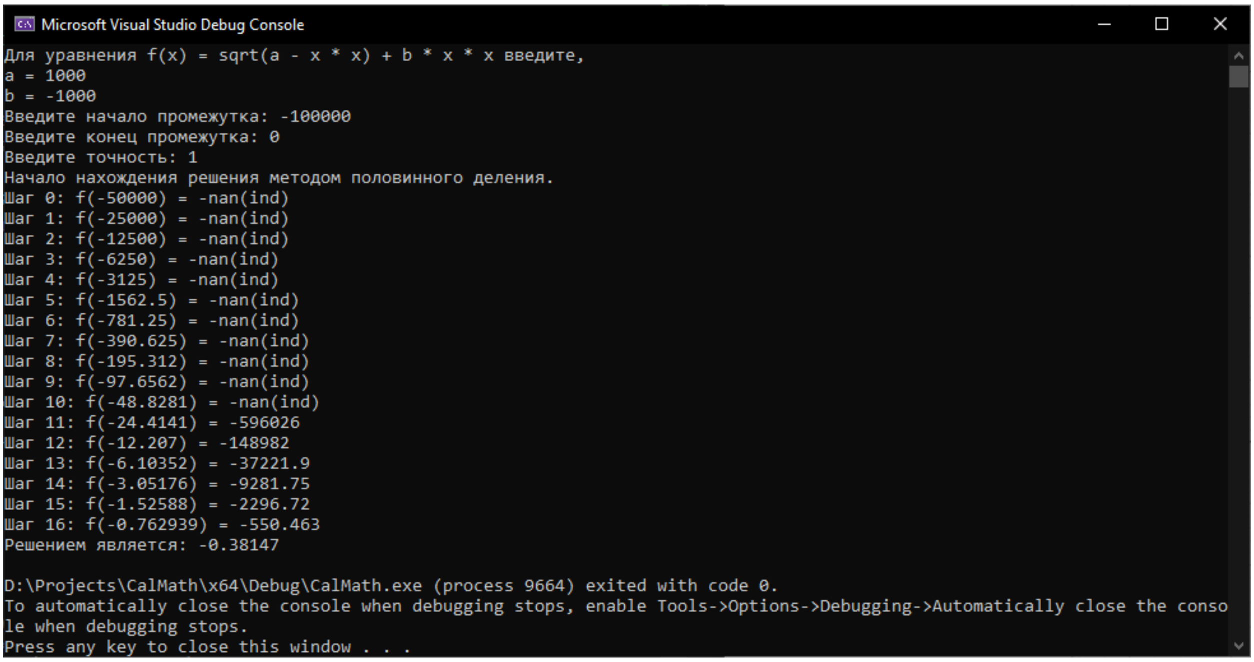


Рисунок 12 – Результат работы программы с произвольными исходными данными

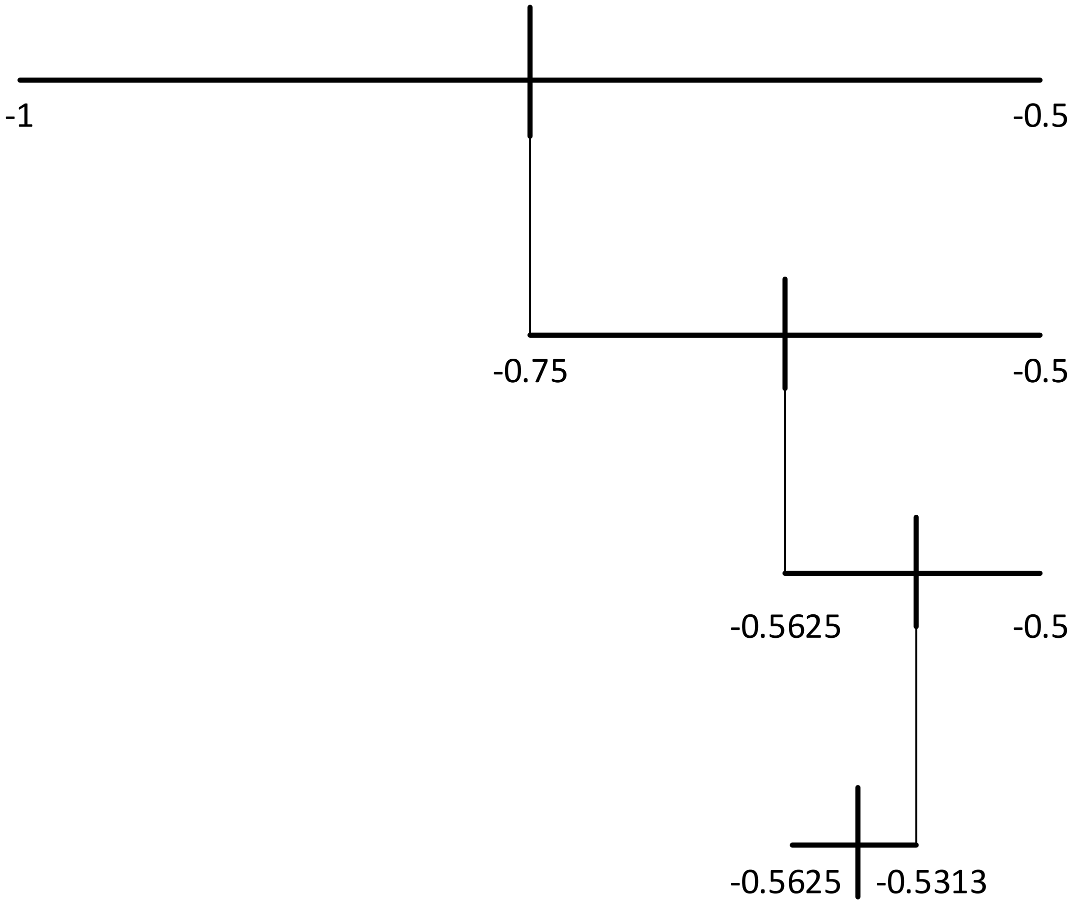


Рисунок 13 – Пример работы алгоритма для корня -0.5533

7 Сравнение программных и аналитических расчётов

Результаты сравниваемых программных и аналитических расчетов были представлены в прошлых разделах.

Сейчас сравним нахождения корня для промежутка [-1;0] (см. Рис. 14). Все шаги совпадают с решением онлайн-калькулятора, тем не менее наш метод вследствие отсутствия ограничения на количество выводимых символов типа double, более точно показывает шаги. Полученные значения совпадают с графиком (см. рис. 2).

№	x	x*
1	-0.5	-0.5
2	-0.75	-0.75
3	-0.625	-0.625
4	-0.5625	-0.5625
5	-0.5313	-0.5313
6	-0.5469	-0.5469
7	-0.5547	-0.5547
8	-0.5508	-0.5508
9	-0.5527	-0.5527
10	-0.5537	-0.5537
11	-0.5532	-0.5532
12	-0.5535	-0.5535
13	-0.5533	-0.5533
14	-0.5533	-0.5533

Рисунок 14 – Таблица сравнения для корня в промежутке [-1;0]

Теперь сравним нахождения корня для промежутка [0;1] (см. рис. 15). Аналогично сопоставлению промежутка [-1;0], на промежутке [0;1] все шаги совпадают. Мы можем утверждать, что реализованный нами алгоритм является верным.

N	x	x*
1	0.5	0.5
2	0.75	0.75
3	0.625	0.625
4	0.5625	0.5625
5	0.5313	0.5313
6	0.5469	0.5469
7	0.5547	0.5547
8	0.5508	0.5508
9	0.5527	0.5527
10	0.5537	0.5537
11	0.5532	0.5532
12	0.5535	0.5535
13	0.5533	0.5533
14	0.5533	0.5533

Рисунок 15 – Таблица сравнения для корня в промежутке [0;1]

8 Выводы

В ходе выполнения лабораторной работы были составлены схема алгоритма и программа на языке C++, решающая поставленную задачу в соответствии с вариантом. Результаты при аналитических и программных расчётах оказались одинаковы – итог и все промежуточные значения оказались равны.

В результате вычислений были получены два корня, что соответствует графику:

x1 = -0.5533;

x2 = 0.5533.

В результате выполнения работы был освоен метод дихотомии для решения нелинейных уравнений. Этот метод основан на последовательном разделение отрезка на два одинаковых, что схоже с алгоритмом бинарного поиска. Также в результате выполнения работы были усовершенствованы навыки по алгоритмизации и программированию вычислительных задач.