

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра вычислительной техники

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 24
Приближенные методы решения уравнений и вычисления
определенного интеграла
Вариант № 2

Преподаватель

подпись, дата

Пушкарев К. В.

Студент КИ18-096, 031830645

подпись, дата

Котов С.А.

Студент КИ18-096, 031831293

подпись, дата

Овсянников В.А.

Студент КИ18-096, 031831050

подпись, дата

Машковская Н.В.

Красноярск 2019

1 Дополнительное упражнение № 1 (вариант 2)

Напечатать в порядке возрастания корни уравнений, вычисленные с заданной точностью, двумя способами: методом хорд и методом половинного деления:

$$1 / (1 + x \times x) = x$$

$$4x - \ln x - 5 = 0$$

$$x - \sin x - 0.25 = 0$$

Точность и интервал вводятся с клавиатуры. Полученные результаты сравнить.

2 Цель работы

Получить практические навыки использования указателей на функции как параметров функции, получить опыт командной работы.

3 Графическая схема алгоритма

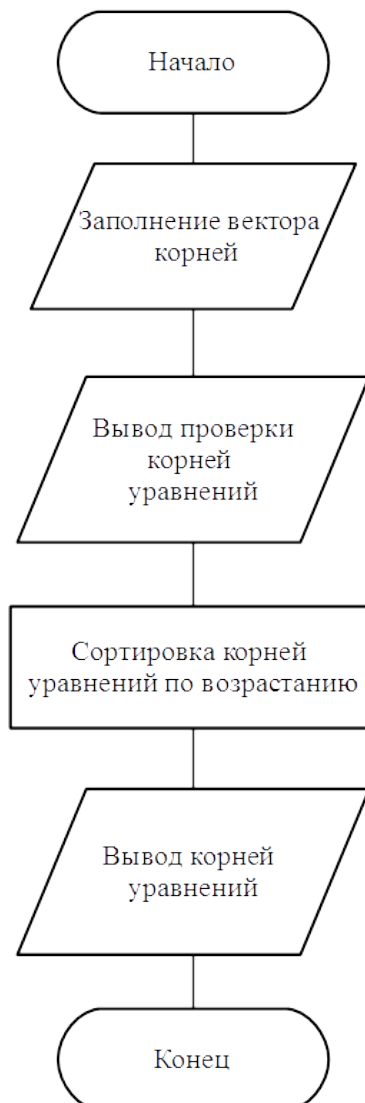


Рисунок 1 – Графическая схема алгоритма

4 Код программы

```
1 #include "stdio.h"
2 #include <iostream>
3 #include <clocale>
4 #include <cmath>
5 #include <algorithm>
6 #include <vector>
7
8 using namespace std;
9
10 typedef double(*left_part)(const double);
11
12 double func1(const double x) {
13     return (1 / (1 + x * x)) - x;
14 }
15
16 double func2(const double x) {
17     return 4 * x - log(x) - 5;
18 }
19
20 double func3(const double x) {
21     return x - sin(x) - 0.25;
22 }
23 // Метод половинного деления(бисекции)
24 double solution_div2(left_part f, double a, double b, const double eps) {
25     double x;
26     do {
27         x = (a + b) / 2;
28         if (f(a) * f(x) > 0) {
29             a = x;
30         }
31         else {
32             b = x;
33         }
34     } while (fabs(b - a) > eps);
35     return x;
36 }
37 // Метод хорд
38 double solution_chord(left_part f, double a, double b, const double eps) {
39     while (fabs(b - a) > eps) {
40         a = b - (b - a) * f(b) / (f(b) - f(a));
41         b = a - (a - b) * f(a) / (f(a) - f(b));
42     }
43     return b;
44 }
45
46 int main() {
47     setlocale(LC_ALL, "");
48
49     double a, b, eps;
50     vector<left_part> f = { func1, func2, func3 }; // Вектор указателей на функции
51     vector<double> solutions; // Вектор результатов
52
53     for (int i = 0; i < f.size(); i++) {
54         cout << "Введите a, b и eps для " << i + 1 << " уравнения: ";
55         cin >> a >> b >> eps;
56         if (f[i](a) * f[i](b) > 0) {
57             cout << "Нет корней" << endl;
58         }
59         else {
60             double temp = solution_div2(f[i], a, b, eps);
```


```

61         solutions.push_back(temp); // Запись корней функции (метод бисекции)
62         cout << "Проверка корней уравнений: " << f[i](temp) << " ";
63         temp = solution_chord(f[i], a, b, eps);
64         solutions.push_back(temp); // Запись корней функции (метод хорд)
65         cout << f[i](temp) << endl;
66     }
67 }
68
69 // Сортировка корней уравнений
70 sort(solutions.rbegin(), solutions.rend());
71
72 cout << "Корни уравнений: " << endl;
73 for (int i = 0; i <= f.size(); i++) {
74     cout << solutions[i] << " ";
75 }
76
77 return 0;
78 }


```

5 Результаты выполнения экспериментальной части работы


Результаты запуска программы с различными входными значениями приведены ниже.

 Консоль отладки Microsoft Visual Studio

```
Введите a, b и eps для 1 уравнения: -10 10 0.000001
Проверка корней уравнений: -8.82372e-07 1.58502e-11
Введите a, b и eps для 2 уравнения: 1 10 0.000001
Проверка корней уравнений: -1.01756e-06 0
Введите a, b и eps для 3 уравнения: 0 10 0.000001
Проверка корней уравнений: -1.04358e-07 2.5413e-13
Корни уравнений:
1.31927 1.31927 1.17123 1.17123 0.682328 0.682328
```

 Консоль отладки Microsoft Visual Studio

```
Введите a, b и eps для 1 уравнения: 0 10 0.0001
Проверка корней уравнений: 5.0779e-05 -4.77396e-15
Введите a, b и eps для 2 уравнения: 0 10 0.0001
Нет корней
Введите a, b и eps для 3 уравнения: 0 10 0.0001
Проверка корней уравнений: -2.52315e-05 2.5413e-13
Корни уравнений:
1.17123 1.17119 0.682328 0.682297
```

 Консоль отладки Microsoft Visual Studio

```
Введите a, b и eps для 1 уравнения: 1 10 0.00001
Нет корней
Введите a, b и eps для 2 уравнения: 1 10 0.00001
Проверка корней уравнений: -1.66699e-05 0
Введите a, b и eps для 3 уравнения: -10 10 0.00001
Проверка корней уравнений: 3.90156e-06 2.5413e-13
Корни уравнений:
1.31927 1.31926 1.17124 1.17123
```