Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский Авиационный Институт»
Национальный Исследовательский Университет

Факультет №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

КУРСОВОЙ ПРОЕКТ

По дисциплине «Вычислительные системы»

По курсам: «Основы информатики», «Алгоритмы структуры и данных»

Задание: «Процедуры и функции в качестве параметров»

Студент:	Хренникова А. С.
Группа:	М80-108-19
Преподаватель:	Поповкин А. В.
Подпись:	
Оценка:	
Дата:	

Москва, 2019

Содержание

Задание

Составить программу на языке Си с процедурами решения алгебраических уравнений различными численными методами (итераций, Ньютона и половинного деления – дихотомии). Нелинейные уравнения оформить как параметры-функции, разрешив относительно неизвестной величины в случае необходимости. Применить каждую процедуру к решению двух уравнений. Если метод не применим, дать математическое обоснование и графическую иллюстрацию. Корень вычисляется на определённом для каждого из вариантов отрезке. Кроме того, в задании указывается приближённое значение корня (для проверки правильности вычислений).

Таблица 1 – Задание 23 и 24 вариантов

Вариант №	Уравнение	Отрезок, содержащий корень	Базовый метод	Приближенное значение корня
23	$3x - 4\ln(x) - 5 = 0$	[2,4]	Ньютона	3.23
24	$\cos\left(\frac{2}{x}\right) - 2\sin\left(\frac{1}{x}\right) + \frac{1}{x} = 0$	[1,2]	дихотомии	1.8756

Общий метод решения

При решении реальных задач, где поведение функции F(x) неизвестно, сначала производят исследование функции (аналитическое, численное, графическое), и т. н. отделение корней, т. е. разбивают область определения функции на отрезки монотонности, на каждом из которых имеется ровно один корень и выполняются другие условия применимости численных методов (гладкость). Разные численные методы предъявляют свои определённые требования к функции F(x), обладают различной скоростью сходимости и поведением.

В данном задании рассматривается уравнение вида F(x) = 0. Предполагается, что функция F(x) на заданном отрезке [a;b] достаточно гладкая, монотонная, и здесь существует единственный корень уравнения $x_0 \in [a; b]$. На отрезке [a;b] ищется приближенное решение x с точностью ε, т. е., такое, что $|x - x_0| < k * \varepsilon$, где k – константа, определяющая точность вычислений. Она задаётся вручную в тексте программы. Например, в виде степени десятки.

Поиск приближённого значения производится с помощью трёх простейших численных методов решения алгебраических уравнений: метода итераций, метода Ньютона и метода половинного деления – дихотомии. Необходимо провести вычислительные эксперименты по определению корней уравнений на указанных в задании отрезках монотонности. Т. е. нужно запрограммировать несколько функций и попробовать с помощью них вычислить корень каждым из способов.

Общие сведения о программе

Необходимое программное и аппаратное обеспечение: ОС семейства UNIX (Linux Ubuntu), среда программирования Си (язык Си, компилятор gcc), процессор с 64-битной архитектурой (как на лабораторном компьютере).

Система программирования: GUN C.

Строк в программе: 63.

Местонахождение файлов на домашнем компьютере: /home/lina_tucha/dir/kp1.c – файл с программой. Сам файл компилируется с помощью написания «gcc kp1.c -lm -o 123» в командной строке интерпретатора команд. Запуск файла осуществляется вызовом исполняемого файла «./123». -lm - это математическая библиотека, -o задает имя исполняемого файла.

Функциональное назначение

Программа предназначена для нахождения корня уравнения на заданном отрезке. Причём точно должно быть известно, что корень существует и он не единственный на данном промежутке. Входные данные задаются в самой программе: отрезок, на котором будет вычисляться корень, само уравнение и коэффициент k, определяющий точность вычислений. Данные ограничиваются вместимостью базовых типов: в double помещается 8 байт, соответственно, концы отрезка, его длина, k и результат вычисляемой функции должны помещаться в диапазон $|[1.797693e + 308; 2.225074e - 308]|$. Максимальные значения можно вычислить путём подключения библиотеки <float.h> и вывода на печать DBL_MAX и DBL_MIN, лучше со спецификатором “%e” – в научной форме.

Три разных алгоритма приближённо находят корень уравнения. Каждый из методов имеет разную скорость сходимости и выдаёт разный ответ. Возможность выбирать оптимальный метод вычисления ответа помогает при решении реальных задач.

Метод дихотомии (половинного деления)

Метод заключается в делении отрезка пополам и его сужения на каждом шаге итерационного процесса в зависимости от знака функции в каждой из двух частей. Начальные значения концов отрезка – концы заданного отрезка [a;b]. находится середина отрезка: $c = (a_0 + b_0) / 2$. Так, корень будет находиться в одной из частей: (a;c) или (c;b). Если корень находится в левой части отрезка, то значения функции на его концах будут различны и F(a)*F(c)<0. Тогда мы должны выбирать именно этот отрезок, т. е. сдвинуть правую границу: b=c. Соответственно, если мы определим, что корень в другой части, то пишем: a=c.

Мы не рассматриваем отдельно случаи, когда корень равен a, b или c – F(a)=0, F(b)=0, F(c)=0. Условие подразумевает включение искомой точки в отрезок, который мы берём на следующем шаге. И в конце итерационного процесса, по достижении заданной точности, мы не будем сильно далеко от правильного ответа – например, b – корень уравнения, F(b)=0, а наш ответ : $x_0 = (a + b) / 2$. Точность вычислений характеризуется длиной отрезка: $|b - a| < k * \varepsilon$, а ответ будет отличаться от правильного лишь на половину отрезка, что будет удовлетворять заданной точности.

Метод итераций

Идея метода заключается в замене исходного уравнения F(x)=0 уравнением вида x=f(x). Достаточное условие сходимости метода: |f ‘(x)|<1, $x \in [a;b]$. Функция может быть выбрана неоднозначно, т. е., выразить x через x можно разными способами, и в случае неверного выбора функции метод расходится. Нужно найти f(x) такую, чтобы для неё выполнялось условие сходимости.

За начальное приближение корня возьмём середину исходного отрезка: x=(a+b)/2. На каждом шаге – новое значение x, ещё больше приближающее нас к правильному ответу: $\left|x_{(i+1)} - x_{(i)}\right| < k * \varepsilon$. Приближенным значением корня будет x_0, который нам даёт достижение заданной точности, т. е., после вывода из цикла над полученным на последней итерации x нужно провести ещё одну операцию: $x_0 = f(x)$.

Таблица 2 - Варианты выражения уравнений вида x = f(x)

№	f(x) для уравнения $3x - 4\ln(x) - 5 = 0$	Выполняется ли условие сходимости |f ’(x)|<1 на отрезке [a;b]		f(x) для уравнения $\cos\left(\frac{2}{x}\right) - 2\sin\left(\frac{1}{x}\right) + \frac{1}{x} = 0$
1.	$x = \frac{4\ln(x) + 5}{3}$	да	нет	$x = \frac{1}{2\sin\frac{1}{x} - \cos\frac{2}{x}}$
2.	$x = 4x - 4\ln x - 5$	нет	да	$x = \frac{1}{\sin^{-1}\left(\frac{\cos\frac{2}{x} + \frac{1}{x}}{2}\right)}$
3.	$x = e^{\frac{3x-5}{4}}$	нет	нет	$x = \frac{2}{\cos^{-1}\left(2\sin\frac{1}{x} - \frac{1}{x}\right)}$

Для вычисления в программе были выбраны варианты $f_1 = \frac{4\ln(x)+5}{3}$ и $f_2 = \frac{1}{\sin^{-1}\left(\frac{\cos\frac{2}{x}+\frac{1}{x}}{2}\right)}$.

Метод Ньютона

Метод Ньютона является частным случаем метода итераций.

Условие сходимости метода: |F(x)*F’’(x)| - (F’(x))^2 < 0 на отрезке [a;b].

Начальное значение – середина отрезка: x=(a+b)/2.

Итерационный процесс: $\mathrm{X}_{(\mathrm{i}+1)} = \mathrm{X}_{(\mathrm{i})} - \mathrm{F}\left(\mathrm{x}_{(\mathrm{i})}\right) / \mathrm{F}'\left(\mathrm{x}_{(\mathrm{i})}\right)$.

Условие вывода: $\left|\mathrm{x}_{(\mathrm{i}+1)} - x_{(\mathrm{i})}\right| < \mathrm{k} * \varepsilon$.

Для получения ответа после выхода из цикла над полученным на последней итерации x проведём ещё одну операцию: $\mathrm{x}_0 = \mathrm{x} - \mathrm{F}(\mathrm{x}) / \mathrm{F}'(\mathrm{x})$.

Таблица 3 - Проверка сходимости метода для 23 варианта

$F_1(\mathrm{x}) = 3\mathrm{x} - 4\ln(\mathrm{x}) - 5$	$F_1'(\mathrm{x}) = 3 - \frac{4}{x}$	$F_1''(\mathrm{x}) = -\frac{4}{3x^2}$

$$f_{N1}(\mathrm{x}) = |(3\mathrm{x} - 4\ln(\mathrm{x}) - 5\left(-\frac{4}{3x^2}\right)| - (3 - \frac{4}{x})^2 < 0$$

Условие $f_{N1}(\mathrm{x}) < 0$ выполняется на всём отрезке [2;4], метод на нем применим.

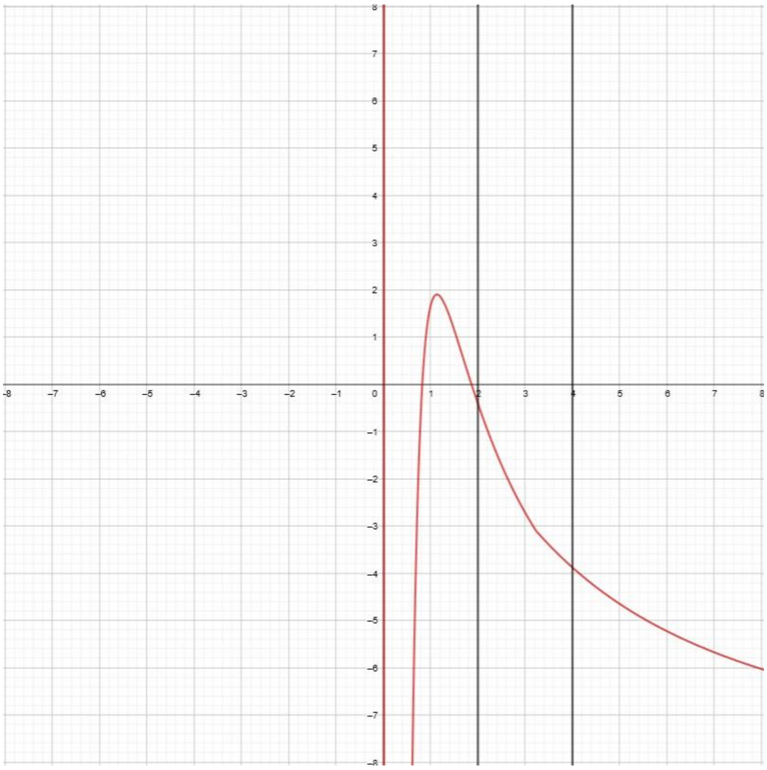

Рисунок 1 – график функции $f_{N1}(\mathrm{x})$

Таблица 4 - Проверка сходимости метода для 24 варианта

$F_2(\text{x}) = \cos\left(\frac{2}{x}\right) - 2\sin\left(\frac{1}{x}\right) + \frac{1}{x}$	$F_2'(\text{x}) = \frac{2\left(\sin\left(\frac{2}{x}\right) + \cos\left(\frac{1}{x}\right)\right) - 1}{x^2}$	$F_2''(\text{x}) = \frac{2(-2x\sin\left(\frac{2}{x}\right) - 2x\cos\left(\frac{1}{x}\right) + x + \sin\left(\frac{1}{x}\right) - 2\cos\left(\frac{2}{x}\right))}{x^4}$

$$f_{N2}(\text{x}) = \left|\left(\cos\left(\frac{2}{x}\right) - 2\sin\left(\frac{1}{x}\right) + \frac{1}{x}\right)\left(\frac{2\left(-2x\sin\left(\frac{2}{x}\right) - 2x\cos\left(\frac{1}{x}\right) + x + \sin\left(\frac{1}{x}\right) - 2\cos\left(\frac{2}{x}\right)\right)}{x^4}\right)\right| - \left(\frac{2\left(\sin\left(\frac{2}{x}\right) + \cos\left(\frac{1}{x}\right)\right) - 1}{x^2}\right)^2 < 0$$

Условие $f_{N2}(\text{x}) < 0$ выполняется на всём отрезке [1;2], метод на нем применим.

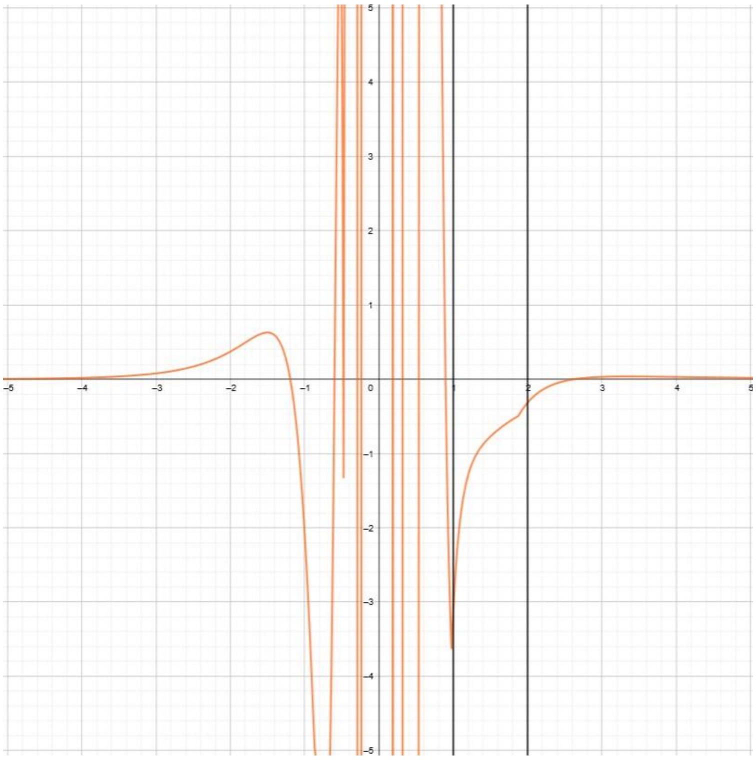

Рисунок 2 – График функции $f_{N2}(\text{x})$

Описание логической структуры

- Задание и вычисление констант (k, ε).
- Задание функций и их производных в соответствии с вариантом (F1, F2, xfx1, xfx2, производные f1, f2).
- Задание численных методов в виде функций (DIH – метод дихотомии, IT – метод итераций, N – метод Ньютона). В качестве параметров данных функций выступают ранее заданные функции.
- В функции main вызываются функции и выводятся на печать константы и вычисление значения корней каждого уравнения.

Описание переменных, функций, входные и выходные данные

Переменные и константы вспомогательных функций

Таблица 5 - Переменные и константы функции main(основы программы)

Имя	Тип	Вид	Назначение
eps	double	Глобальная переменная	Машинное эпсилон
k	const int	Глобальная константа	Коэффициент точности
DIH	double	Вызываемая функция	Вычисляет корень заданного уравнения
IT	double	Вызываемая функция	Вычисляет корень заданного уравнения
N	double	Вызываемая функция	Вычисляет корень заданного уравнения

Вспомогательные функции

Таблица 6 – Функция F1, данная по условию 23 варианта

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Параметр данной функции

Таблица 7 – Функция f1 - первая производная функции F1

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Параметр данной функции

Таблица 8 – Функция xfx1 - выражение вида x=f(x) для функции F1

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Параметр данной функции

Таблица 9 – Функция F2 - функция, данная по условию 24 варианта

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Параметр данной функции

Таблица 10 – Функция f2 - первая производная функции F2

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Параметр данной функции

Таблица 11 – Функция xfx2 - выражение вида x=f(x) для функции F2

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Параметр данной функции

Таблица 12 – Функция DIH(Находит корень методом половинного деления – дихотомии)

Имя	Тип	Вид	Назначение
a	double	Входной параметр	Начало отрезка
b	double	Входной параметр	Конец отрезка
f	double	Входной параметр-функция	Заданная функция F(x)
c	double	Локальная переменная	Середина отрезка [a;b]
eps	double	Глобальная переменная	Машинное эпсилон
k	const int	Глобальная константа	Коэффициент точности

Таблица 13 - Функция IT(Находит корень методом итераций)

Имя	*Тип*	*Вид*	*Назначение*
a	double	Входной параметр	Начало отрезка
b	double	Входной параметр	Конец отрезка

Продолжение таблицы 13

f	double	Входной параметр-функция	Функция вида x=f(x)
x	double	Локальная переменная	Вычисляемый корень
eps	double	Глобальная переменная	Машинное эпсилон
k	const int	Глобальная константа	Коэффициент точности

Таблица 14 - Функция N(Находит корень методом Ньютона)

Имя	Тип	Вид	Назначение
a	double	Входной параметр	Начало отрезка
b	double	Входной параметр	Конец отрезка
f	double	Входной параметр-функция	Первая производная функции F(x)
F	double	Входной параметр-функция	Заданная функция F(x)
x	double	Локальная переменная	Вычисляемый корень
eps	double	Глобальная переменная	Машинное эпсилон
k	const int	Глобальная константа	Коэффициент точности

Встроенные функции

Функция fabs: вычисляет абсолютное значение (модуль) и возвращает его |x|. В Си, определён только один прототип данной функции, с типом данных double.

Функция acos: вычисляет арккосинус и возвращает значение арккосинуса параметра с плавающей точкой в интервале [-1,1]. В тригонометрии, арккосинус является обратной тригонометрической функцией косинуса. В Си, определён только один прототип этой функции, с типом данных double.

Входные данные:

Фактических входных данных нет – они задаются сразу в коде программы. Конкретно, это отрезок, на котором будет вычисляться корень и заданное в соответствии с вариантом уравнение. Значения концов отрезка, его длина и результат вычисляемой функции должны укладываться в диапазон значений около |[1,7*10^308; 2.2*10^-308]|.

Концы отрезка как параметры при вызове функций DIH, IT и N. Исходные функции и их производные заносятся в код в качестве подпрограмм и тоже передаются как параметры.

Стоит заметить, что в заданных вариантах уравнений (при вычислении производных, при проверке условий сходимости и т. д.) есть области определений функций.

Таблица 15 – Области определения функций

Вариант №23		Вариант №24	
Функция	ООФ	Функция	ООФ
$F_1(\mathrm{x}) = 3\mathrm{x} - 4\ln(\mathrm{x}) - 5$	$\mathrm{x} \in \mathbb{R}$	$F_2(\mathrm{x}) = \cos\left(\frac{2}{x}\right) - 2\sin\left(\frac{1}{x}\right) + \frac{1}{x}$	$\mathrm{x} \neq 0$
$F_1'(\mathrm{x}) = 3 - \frac{4}{x}$	$\mathrm{x} \in \mathbb{R}$	$F_2'(\mathrm{x}) = \frac{2\left(\sin\left(\frac{2}{x}\right) + \cos\left(\frac{1}{x}\right)\right) - 1}{x^2}$	$\mathrm{x} \neq 0$
$F_1''(\mathrm{x}) = -\frac{4}{3x^2}$	$\mathrm{x} \neq 0$	$F_2''(\mathrm{x}) = \frac{2(-2x\sin\left(\frac{2}{x}\right) - 2x\cos\left(\frac{1}{x}\right) + x + \sin\left(\frac{1}{x}\right) - 2\cos(\frac{2}{x}))}{x^4}$	$\mathrm{x} \neq 0$
$f_1 = \frac{4\ln(\mathrm{x}) + 5}{3}$	$\mathrm{x} \in \mathbb{R}$	$f_2 = \frac{1}{\sin^{-1}\left(\frac{\cos\frac{2}{x} + \frac{1}{x}}{2}\right)}$	$\mathrm{x} \neq 0$

Выходные данные:

Выходные данные печатаются в виде таблицы с четырьмя колонками. Выводится исходная функция и значения её [a;b], вычисленные тремя методами: дихотомии, итераций, методом Ньютона. Перед таблицей печатается вычисленное значение машинного эпсилон и коэффициент точности k.

Протокол

lina_tucha@LAPTOP-44CRFC1U:~$ cd dir

lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano kp1.c

lina_tucha@LAPTOP-44CRFC1U:~/dir$ cat kp1.c

```
#include <stdio.h>
#include <math.h>

const int k=10000;

double F1(double x) {
    return (3*x-4*log(x)-5); }

double f1(double x) {
    return (3-(4/x)); }

double xfx1(double x) {
    return ((4*log(x)+5)/3); }

double F2(double x) {
    return (cos(2/x)-2*sin(1/x)+1/x); }

double f2(double x) {
    return ((2*(sin(2/x)+cos(1/x))-1)/(x*x)); }

double xfx2(double x) {
    return (2/acos(2*sin(1/x)-(1/x))); }

double DIH(double a,double b, double (*f)(double)) {
      double c;
      double eps=1.0;
      while ((eps/2+1)>1) {eps/=2;}
      while (fabs(b-a)>=k*eps) {
           c=(a+b)/2;
```

```
            if (f(a)*f(c)>0) a=c;
                else b=c;
        }
        return ((a+b)/2);
}

double IT(double a, double b, double (*f)(double)) {
        double x=(a+b)/2;
        double eps=1.0;
        while ((eps/2+1)>1) {eps/=2;}
        while (fabs(f(x)-x)>=(k*eps)) {x=f(x);}
        return (f(x));
}

double N(double a, double b, double (*F)(double), double (*f)(double)) {
        double x=(a+b)/2;
        double eps=1.0;
        while ((eps/2+1)>1) {eps/=2;}
        while (fabs(x-F(x)/f(x)-x)>=k*eps) {x=x-F(x)/f(x);}
        return (x-F(x)/f(x));
}

int main() {
    double eps=1.0;
    while ((eps/2+1)>1) {eps/=2;}
    printf("Epsilon for double = %.30lf, k = %d\n",eps,k);

printf("______________________________________________________________________________________\n");
    printf("|          Function          |  Dihot       |  Iter        |  Newton      |\n");

printf("|___________________________|________________|________________|________________|\n");
    printf("|        3*x-4*log(x)-5
|%16.13lf|%16.13lf|%16.13lf|\n",DIH(2.0,4.0,F1),IT(2.0,4.0,xfx1),N(2.0,4.0,F1,f1));
```

```
printf("|__________________________________|_________________|_________________|________________|\n");

    printf("|     cos(2/x)-2*sin(1/x)+1/x          |%16.13lf|%16.13lf|%16.13lf|\n",DIH(1.0,2.0,F2),IT(1.0,2.0,xfx2),N(1.0,2.0,F2,f2));

printf("|__________________________________|_________________|_________________|________________|\n");

    }
```

lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc kp1.c -lm -o 123

lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123

Epsilon for double = 0.000000000000000222044604925031, k = 10000

Function	Dihot	Iter	Newton
3*x-4*log(x)-5	3.2299594397273	3.2299594397272	3.2299594397279
cos(2/x)-2*sin(1/x)+1/x	1.8756173924912	1.8756173924910	1.8756173924905

Таблица значений

Таблица 16 – Значения корней

Function	Dihot	Iter	Newton
3*x-4*log(x)-5	3.2299594397273	3.2299594397272	3.2299594397279
cos(2/x)-2*sin(1/x)+1/x	1.8756173924912	1.8756173924910	1.8756173924905

Заключение

Цель задания достигнута – найдены корни двух уравнений. Была проверена правильность вычислений путём сравнения трёх результатов – между собой и со значениями, данными в условиях Вариантов №23 и №24. В процессе выполнения работы мы познакомились с некоторыми условиями возможности вычисления корней данными методами. Также получены полезные навыки работы с функциями: передача функций как параметров, присваивание глобальной переменной значения функции. Кроме того, работа с несколькими функциями и параметрами помогает научиться правильно давать им названия и не путаться. Нахождение корней с определённой точностью с помощью ресурсов ЭВМ имеет практический смысл для решения вычислительных задач в реальной жизни.

В абсолюте ни один из методов не идеален, так как требуются заранее определенные границы искомого корня и при увеличении параметра точности затраты по времени растут слишком быстро. Метод итераций не является универсальным из-за того, что необходимо подбирать нужную функцию для замены исходной. Метод Ньютона имеет условие сходимости, вследствие чего не универсален, хотя и дает быстрый результат. Метод дихотомии самый простой для понимания и реализации.

Список использованных источников

1. РосДиплом, Оформление таблиц в дипломной работе, особенности и требования ГОСТ/Электронный диплом/Режим доступа: https://www.rosdiplom.ru/rd/pubdiplom/view.aspx?id=288
2. Диплом Журнал, Оформление курсовой работы по ГОСТу 2019(образец)/Электронный диплом/Режим доступа: https://journal.duplom.ru/kursovaya/oformlenie-kursovoj-raboty-po-gostu-2019-obrazec/
3. Vyuchit.work – универсальная методичка/Электронный диплом/Режим доступа: https://vyuchit.work/samorazvitie/sekretyi/oformlenie-risunkov-po-gostu.html
4. Архив вопросов и ответов для программистов/Электронный диплом/Режим доступа: https://qarchive.ru/320864_parametry_gcc__lm__lz__lrt___o_chem_oni_
5. Компилятор GCC/Электронный диплом/Режим доступа: http://parallel.uran.ru/book/export/html/25
6. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. :Пер. с англ. – М. : Издательский дом «Вильямс», 2009. – 304 с. : ил. – Парал. тит. англ.
7. Диссертация от профессоров и докторов наук/Электронный диплом/Режим доступа: https://dissertatsija.com/poleznoe/oformlenie-rabot/oformlenie-risunkov-i-tablic-po-gost/#i-9
8. SppStudio/Электронный диплом/Режим доступа: http://cppstudio.com/post/1079/
9. Словари и энциклопедии на Академике/Электронный диплом/Режим доступа: https://dic.academic.ru/dic.nsf/ruwiki/1034689