

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет прикладной математики и информатики

Курсовая работа

по курсам

«Основы информатики», «Алгоритмы и структуры данных»

I семестр

Задание 4:

Процедуры и функции в качестве параметров

Студент: Тулин И.Д.

Группа: М8О-101Б-21

Руководитель: Титов В.К.

Оценка: _____

Дата: 04.01.2022

Москва, 2022

1. Задача

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными численными методами (итераций, Ньютона (касательных), половинного деления – дихотомии и хорд). Нелинейные уравнения оформить как параметры-функции, разрешив относительно неизвестной величины в случае необходимости.

2. Вариант

Вар	Функция	Отрезок		Приближенный ответ
22	$\arccos(x) - \sqrt{(1 - 0.3x^3)}$	0.0	1.0	0.5629
собст.	$\sin(\sqrt{x} + 1) - x$	0.5	1.0	0.9246
собст.	$\frac{1}{(\cos(x) + 6)} - x$	0.0	0.64	0.1431

3. Общий метод решения

Описание методов дихотомии, итерации, касательных и хорд для вычисления приближенных значений корней функции при помощи языка Си, подстановка значений в функции и вывод полученных корней.

4. Общие сведения о программе

Аппаратное обеспечение: домашний ноутбук

Операционная система: Linux Mint

Язык и система программирования: GNU C

Число строк программы: 69

Местонахождение файлов: /home/yusayu/Рабочий стол/cppProjects

Компиляция программы в консоли UNIX: `g++ -o kr4.out kr4.cpp`

Вызов программы: `./kr4.cpp`

5. Функциональное назначение

Программа предназначена для решения трансцендентных алгебраических уравнений различными численными методами (итерации, дихотомии, касательных и хорд). В программе используется переменные типа `double`, из чего следует диапазон значений в границах `1,7E +/- 308` (15 знаков).

6. Описание логической структуры

Программа содержит функции, описывающие каждый из четырех требуемых методов вычисления приближенного значения корней. Все четыре функции вызываются по три раза, получая на вход математическую функцию и границы поиска ее корня. Вычисление корней происходит при условии равенства нулю самой математической функции. В конце работы программы функции выводят в общей сложности двенадцать полученных значений корней, которые впоследствии печатаются на экран.

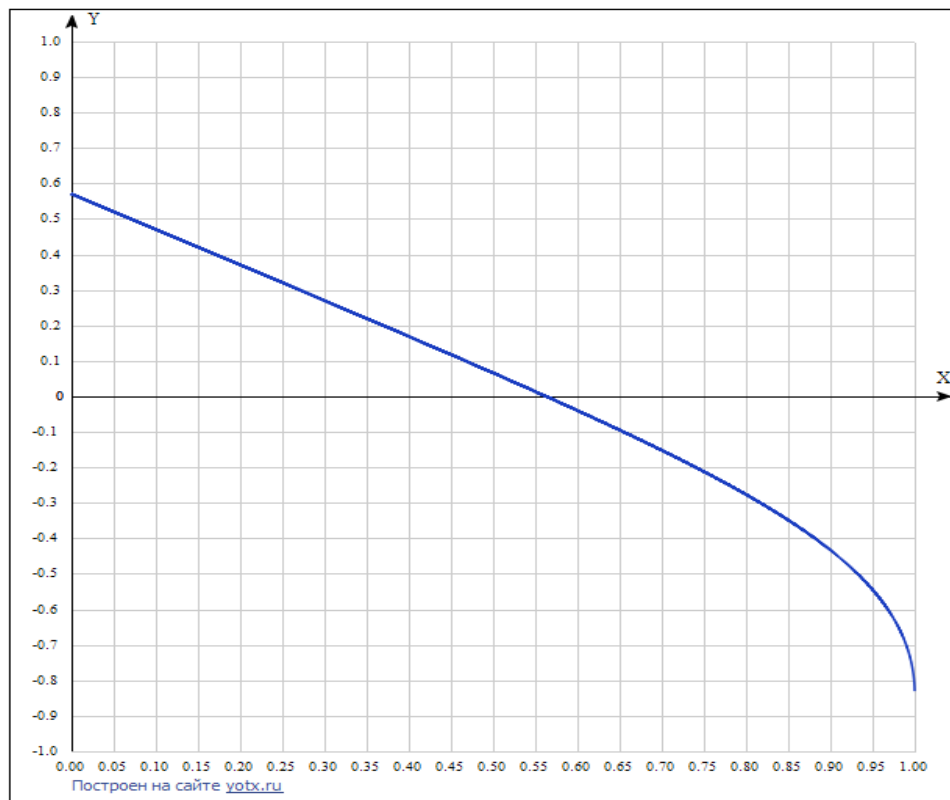
7. Описание переменных и констант

Имя	Тип	Назначение
eps	double	Достаточное эпсилон
a		Левая граница отрезка
b		Правая граница отрезка

8. Описание функций

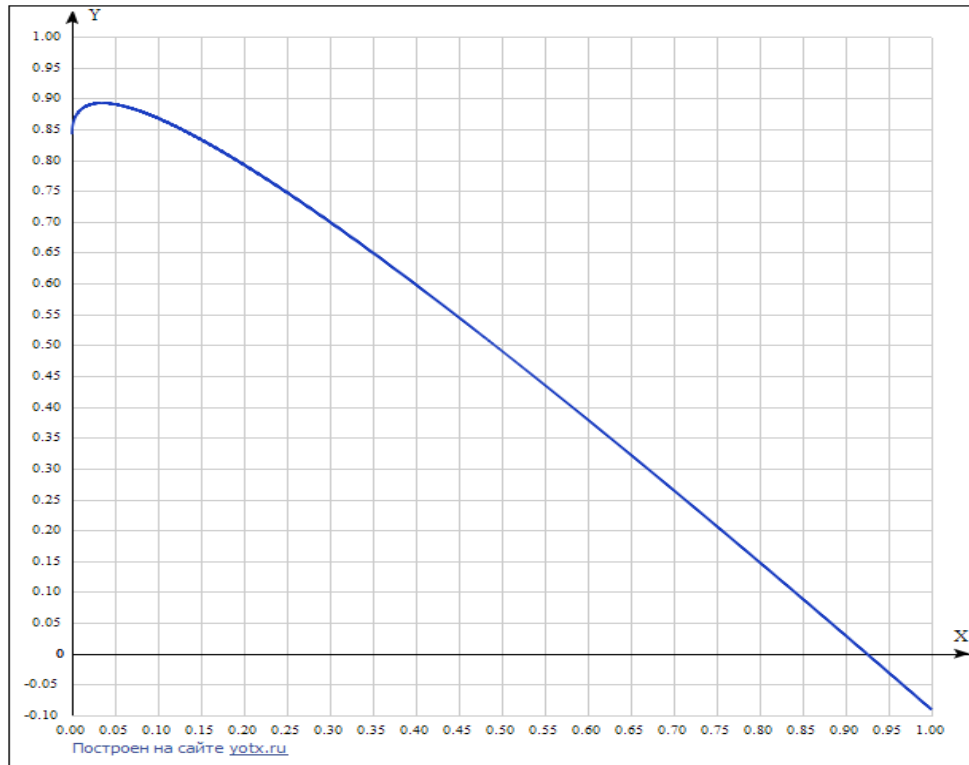
double f1(double x) — первая функция (22 вариант)

$$\arccos(x) - \sqrt{1 - 0.3x^3} = 0. \text{ Отрезок } [0; 1]$$



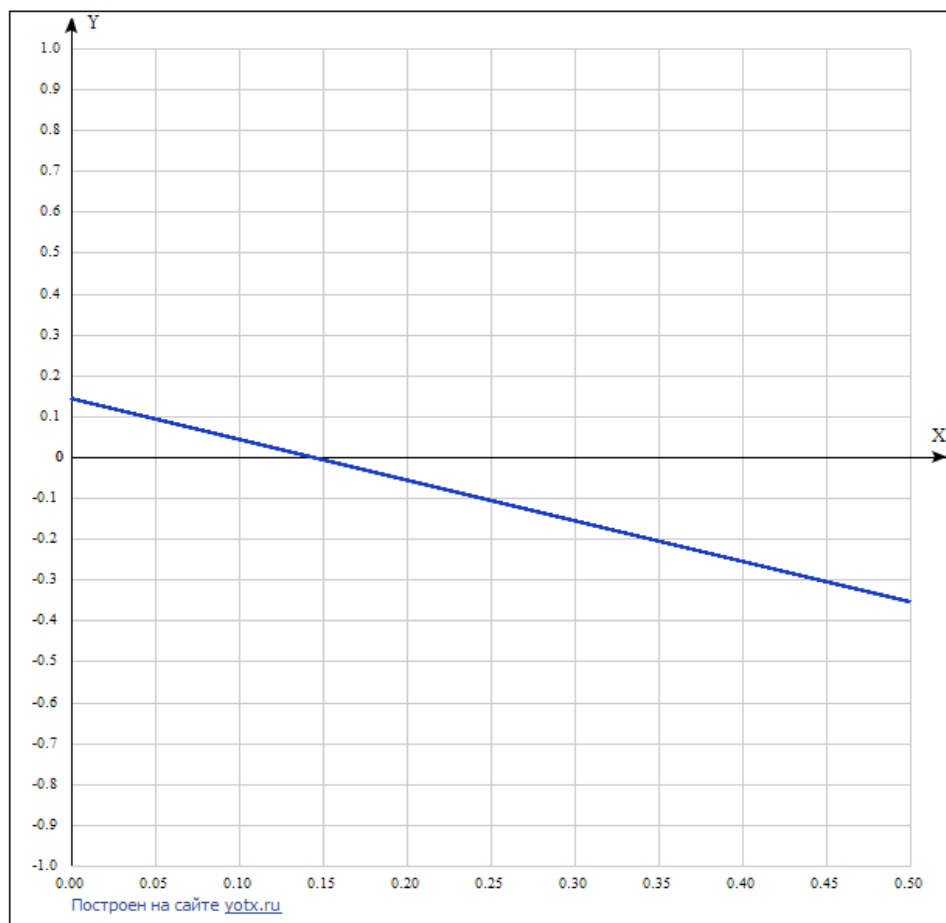
double f2(double x) — вторая функция (собственный вариант)

$$\sin(\sqrt{x} + 1) - x = 0. \text{ Отрезок } [0.5; 1]$$



double f3(double x) — третья функция (собственный вариант)

$$\frac{1}{(\cos(x)+6)} - x = 0. \text{ Отрезок } [0; 0.64]$$



Имя функ.	Тип	Назначение
dabs	double	Возвращает модуль своего аргумента
dichotomy		Описание метода дихотомии
iteration		Описание метода итераций
chord		Описание метода хорд
tangent		Описание метода касательных
F1		Функция f1 в виде $F(x)=x$
F2		Функция f2 в виде $F(x)=x$
F3		Функция f3 в виде $F(x)=x$
Fp1		Производная функции f1
Fp2		Производная функции f1
Fp3		Производная функции f1

9. Входные данные

Нет входных данных

10. Тестовые примеры

Не предусмотрены

11. Дневник отладки

Дата	Место	Событие	Действие по исправлению
04.01.17	дом	Вывод nan(not a number) вместо вывода корня первого уравнения по методу касательных в результате потери минуса при вычислении и занесении в программу производной от f1	Возвращение потерянного знака минус

12. Выводы по задаче

Я составил программу на Си с процедурами решений трансцендентных алгебраических уравнений различными численными методами и научился реализовывать эти методы.

13. Протокол

yusayu@YS:~/Рабочий стол/cppProjects\$ cat head

```
*****
*
* Курсовая работа №4
* Процедуры и функции в качестве параметров
* Выполнил: Тулин Иван Денисович
* (номер по списку: 22)
* Группа: М8О-101Б-21
*
*****
```

yusayu@YS:~/Рабочий стол/cppProjects\$ cat kr4.cpp

```
#include<stdio.h>
#include<math.h>
const double eps=0.000001;
double dabs(double);
double dichotomy(double f(double), double, double);
double iteration(double f(double), double, double);
double chord(double f(double), double, double);
double tangent(double f(double), double fp(double), double, double);
double f1(double);
double F1(double);
double Fp1(double);
double f2(double);
double F2(double);
double Fp2(double);
double f3(double);
double F3(double);
double Fp3(double);
int main() {
printf("Корень функции f1 методом деления пополам = %.4f\n", dichotomy(f1,0.,1.));
printf("Корень функции f1 методом итераций = %.4f\n", iteration(F1,0.,1.));
printf("Корень функции f1 методом касательных = %.4f\n", tangent(f1,Fp1,0.,1.));
printf("Корень функции f1 методом хорд = %.4f\n", chord(f1,0.,1.));
printf("Корень функции f2 методом деления пополам = %.4f\n", dichotomy(f2,0.5,1.));
printf("Корень функции f2 методом итераций = %.4f\n", iteration(F2,0.5,1.));
printf("Корень функции f2 методом касательных = %.4f\n", tangent(f2,Fp2,0.5,1.));
printf("Корень функции f2 методом хорд = %.4f\n", chord(f2,0.5,1.));
printf("Корень функции f3 методом деления пополам = %.4f\n", dichotomy(f3,0.,0.64));
printf("Корень функции f3 методом итераций = %.4f\n", iteration(F3,0.,0.64));
printf("Корень функции f3 методом касательных = %.4f\n", tangent(f3,Fp3,0.,0.64));
printf("Корень функции f3 методом хорд = %.4f\n", chord(f3,0.,0.64));
return 0;
}

double dabs(double x){return (x > 0? x : -x);}

double f1(double x){return acos(x)-sqrt(1-0.3*x*x*x);}
double F1(double x){return cos(sqrt(1-0.3*x*x*x));}
double Fp1(double x){return -1/sqrt(1-x*x)+9*x*x/20/sqrt(1-0.3*x*x*x);}
double f2(double x){return sin(sqrt(x)+1)-x;}
double F2(double x){return sin(sqrt(x)+1);}
double Fp2(double x){return cos(sqrt(x)+1)/2/sqrt(x)-1;}
double f3(double x){return 1/(cos(x)+6)-x;}
```

```
double F3(double x){return 1/(cos(x)+6);}
double Fp3(double x){return sin(x)/(cos(x)+6)/(cos(x)+6)-1;}
```

```
double dichotomy(double f(double), double a, double b){double x, oldx;
x=(a+b)/2.;oldx=b;
while(dabs(oldx-x)>eps){if(f(a)*f(x)>0) a=x; else b=x; oldx=x; x=(a+b)/2.;}
return x;
}
```

```
double iteration(double f(double), double a, double b){double x, oldx;
oldx=(a+b)/2.; x=f(oldx);
while(dabs(oldx-x)>eps){oldx=x;x=f(x);}
return x;
}
```

```
double tangent(double f(double), double fp(double), double a, double b){double oldx, x;
oldx = (a+b)/2.; x=oldx-f(oldx)/fp(oldx);
while(dabs(oldx-x)>eps){oldx=x; x=oldx-f(oldx)/fp(oldx);}
return x;
}
```

```
double chord(double f(double), double a, double b){double oldx, x;
oldx=b; x=(b*f(a)-a*f(b))/(f(a)-f(b));
while(dabs(oldx-x)>eps){if(f(a)*f(x)>0) a=x; else b=x; oldx=x; x=(b*f(a)-a*f(b))/(f(a)-f(b));}
return x;
}
```

```
yusayu@YS:~/Рабочий стол/cppProjects$ g++ -o kr4.out kr4.cpp
```

```
yusayu@YS:~/Рабочий стол/cppProjects$ ./kr4.out
```

```
Корень функции f1 методом деления пополам = 0.5629
```

```
Корень функции f1 методом итераций = 0.5629
```

```
Корень функции f1 методом касательных = 0.5629
```

```
Корень функции f1 методом хорд = 0.5629
```

```
Корень функции f2 методом деления пополам = 0.9246
```

```
Корень функции f2 методом итераций = 0.9246
```

```
Корень функции f2 методом касательных = 0.9246
```

```
Корень функции f2 методом хорд = 0.9246
```

```
Корень функции f3 методом деления пополам = 0.1431
```

```
Корень функции f3 методом итераций = 0.1431
```

```
Корень функции f3 методом касательных = 0.1431
```

```
Корень функции f3 методом хорд = 0.1431
```

```
yusayu@YS:~/Рабочий стол/cppProjects$
```