Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра 806 «Вычислительная информатика и программирование»

Факультет: «Информационные технологии и прикладная математика»

Курсовая работа

Дисциплина: «Вычислительная системы»

I семестр

Задание 4: «Процедуры и функции в качестве параметров»

|  |  |
| --- | --- |
| Группа: | М8О-108Б-18, №12 |
| Студент: | Коростелев Дмитрий Васильевич |
| Преподаватель: | Поповкин Александр Викторович |
| Оценка: |  |
| Дата: | 06.12.2018 |

Москва, 2018

**Содержание**

1. Задание.........................................................................................................2
2. Решение........................................................................................................2
   1. Краткие сведения из численных методов.....................................2
   2. Метод итераций...............................................................................3
   3. Метод Ньютона...............................................................................3
   4. Метод половинного деления..........................................................3
   5. Вариант.............................................................................................4
   6. Код программы................................................................................4
   7. Пояснительная записка к коду программы...................................7
   8. Результат работы программы.........................................................7
   9. Примечание......................................................................................7
3. Заключение...................................................................................................9
4. Список использованной литературы..........................................................9
5. **Задание**

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными численными методам (итераций, Ньютона и половинного деление — дихотомии). Уравнения оформить как функции параметры, разрешив относительно неизвестной величины в случае необходимости. Применить каждую процедуру к решению двух уравнений — заданного вариантом и следующего за ним. Если метод неприменим, дать математическое обоснование и графическую иллюстрацию, например, с использованием gnuplot.

1. **Решение**
   1. **Краткие сведения из численных методов**

Рассматривается уравнение вида F(x)=0. Предполагается, что функций F(x) достаточно гладкая, монотонная на этом отрезке и существует единственный корень уравнения x ∈ [a,b]. На отрезке [a, b] ищется приближенное решение x с точностью ε, т. е. такое, что |x - x| < ε.

При решении реальных задач, где поведение функции F(x) неизвестно, сначала производят исследование функции (аналитическое, численное или графическое), например, с помощью программ gnuplot, MathLab, MathCAD, Maple. Также выполняют т. н. отделение корней, т. е. разбивают область опреденения функции на отрезки монотонности, на каждом из которых имеется ровно один корень и выполняются другие условия применимости численных методов (гладкость). Различные численные методы предъявляют разные требования к функции F(x), обладают различной скоростью сходимости и поведением.

В данном задании предлагается изучить и запрограммировать три простейших численных метода решения алгебраических уравнений и провести вычислительные эксперименты по опредению корней уравнений на указанных в задании отрезках монотонности и, в качестве дополнительного упражнения, вне их.

* 1. **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 ∈ [a,b]. Это условие необходимо проверить перед началом решения задачи, так как функция f(x) может быть выбрана неоднозначно, причем в случае неверного выбора указанной функции метод расходится.

Алгоритм:

1. Начальное приближение корня: x0 = (a + b) / 2.
2. Итерационный процесс: xk+1 = f(xk).
3. Условие окончания: |xk – xk-1| < ε.
4. Приближенное значение корня: x\* ≈ xконечное.
   1. **Метод Ньютона**

Метод Ньютона является частным случаем метода итераций.

Условие сходимости метода: |F(x) ⋅ F’’(x)| < (F'(x))2 на отрезке [a, b].

Итерационный процесс: xk+1 = xk - F(xk)/F'(xk).

* 1. **Метод половинного деления**

Очевидно, что если на отрезке [a, b] существует корень уравнения, то значения функции на концах отрезка имеют разные знаки: F(a) ⋅ F(b) < 0. Метод заключается в делении отрезка пополам и его сужении в два раза на каждом шаге итерационного процесса в зависимости от знака функции в середине отрезка.

Алгоритм:

1. За начальное приближение принимаются границы исходного отрезка a0 = a, b0 = b.

2. Итерационный процесс:

2.1. ak+1 = (ak + bk) / 2, bk+1 = bk, если F(ak) ⋅ F((ak + bk) / 2) > 0;

2.2. ak+1 = ak, bk+1 = (ak + bk) / 2, если F(bk) ⋅ F((ak + bk) / 2) > 0.

3. Условие окончания: |ak - bk| < ε.

* 1. **Вариант**

Таблица 1 – задание 12-ого и 13-ого варианта

|  |  |
| --- | --- |
| **Вариант 12** | |
| Уравнение | ln x – x +1,8 |
| Интервал | [2,3] |
| Требуемый ответ | 2,8459 |
| **Вариант 13** | |
| Уравнение | x\*tg(x) – 1/3 |
| Интервал | [0,2;1,0] |
| Требуемый ответ | 0.5472 |

* 1. **Код программы**

dmitry@dmitry-VirtualBox:~/lubs$ cat 4kp.c

#include<stdio.h>

#include<math.h>

double Epsilon(){

double epsilon = 1.0;

while(1.0 + epsilon/2 > 1.0)

epsilon /=2;

return epsilon;

}

double function\_1(double x){

return log(x) - x + 1.8;

}

double function\_2(double x){

return ((x\*tan(x)) - 0.33333333);

}

double function\_1\_der(double x){

return -1.0 + 1.0/x;

}

double function\_2\_der(double x){

return ((x/cos(x)/cos(x)) + tan(x));

}

double function\_1\_x(double x){

return log(x) + 1.8;

}

double function\_2\_x(double x){

return 1/3/tan(x);

}

double Iteration(double f(double), double a, double b, double eps){

double x = (a+b)/2.0;

while(fabs(x - f(x))>eps){

x = f(x);

}

return x;

}

double Nuton(double f(double),double d(double),double a, double b, double eps){

double x0 = (a+b)/2.0;

double x = 0.0;

while(fabs(x0-x)>eps){

x = x0;

x0 -= (f(x0)/d(x0));

}

return x0;

}

double Dihotomii(double f(double),double a, double b, double eps){

double x;

while(fabs(a-b)>eps){

x = (a+b)/2.0;

if(f(a)\*f(x)>0){

a = x;

}

else{

b = x;

}

}

return x;

}

int main(void){

double eps = 10000000\*Epsilon();

double a1 = Iteration(function\_1\_x,2.0,3.0,eps);

double a2 = Nuton(function\_1,function\_1\_der,2.0,3.0,eps);

double a3 = Dihotomii(function\_1,2.0,3.0,eps);

printf("Function Iterations\t Newton\t Dichotomy\t\n");

printf("1\t %lf\t %lf\t %lf\t\n",a1,a2,a3);

a2 = Nuton(function\_2,function\_2\_der,0.2,1.0,eps);

a3 = Dihotomii(function\_2,0.2,1.0,eps);

printf("2\t NO\t %lf\t %lf\t\n",a2,a3);

return 0;

}

* 1. **Пояснительная записка к коду программы**

1. double Epsilon() – вычисляет машинное эпсилон
2. double function\_1(double x) – функция F1(x) = ln x – x +1,8
3. double function\_2(double x) - функция F2(x) = x\*tg(x) – 1/3
4. double function\_1\_der(double x) – функция F’1(x)
5. double function\_2\_der(double x) – функция F’2(x)
6. double function\_1\_x(double x) – функция F1(x) вида x = f(x)
7. double function\_1\_x(double x) – функция F2(x) вида x = f(x)
8. double Iteration(double f(double), double a, double b, double eps) – функция поиска корня уравнения f(x) = 0 методом итераций
9. double Nuton(double f(double),double d(double),double a, double b, double eps) – функция поиска корня уравнения f(x) = 0 методом Ньютона
10. double Dihotomii(double f(double),double a, double b, double eps) – функция поиска корня уравнения f(x) = 0 методом дихотомии.
    1. **Результат работы программы**

dmitry@dmitry-VirtualBox:~/lubs$ ./4kp.out

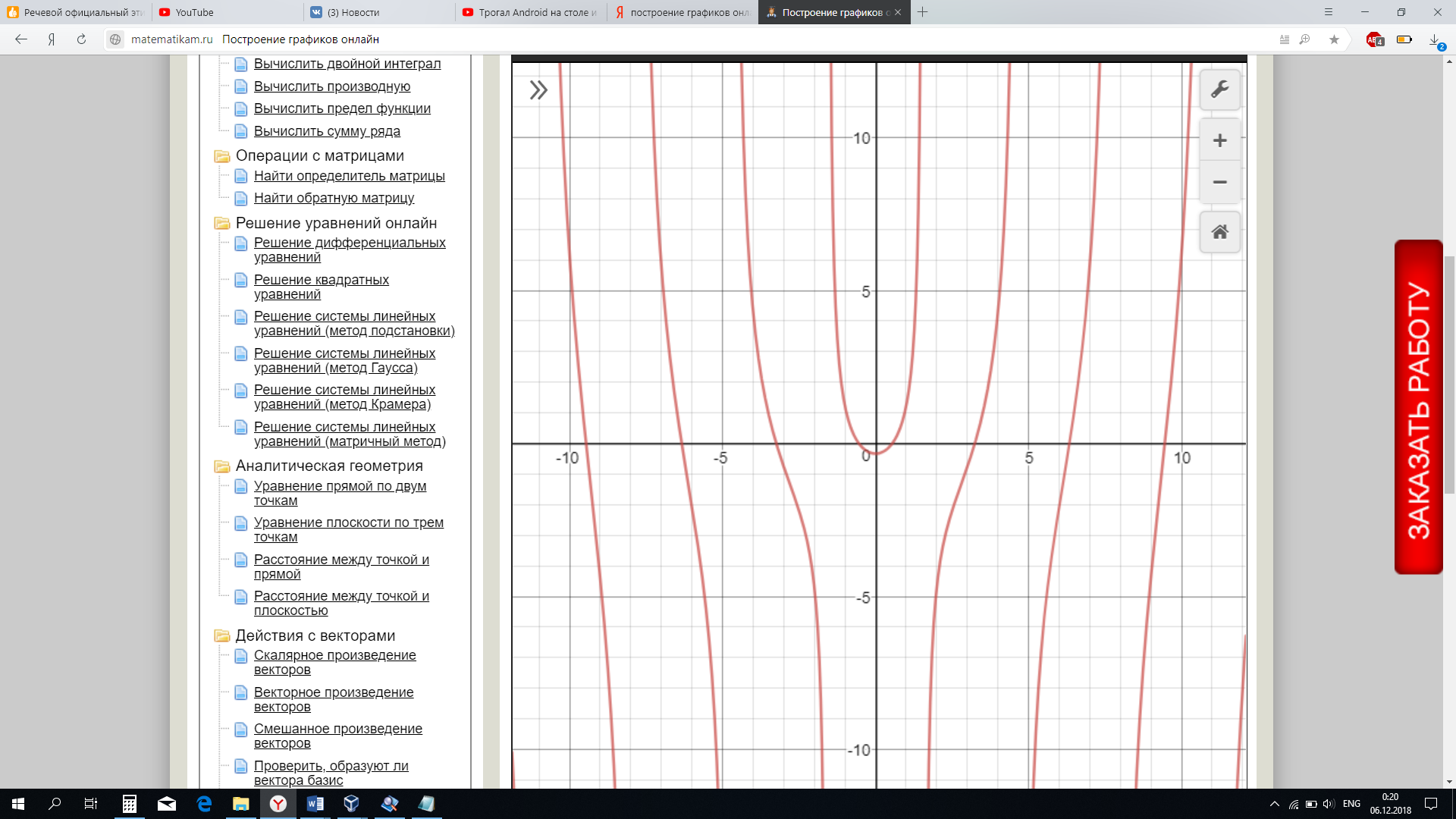
Function Iterations Newton Dichotomy

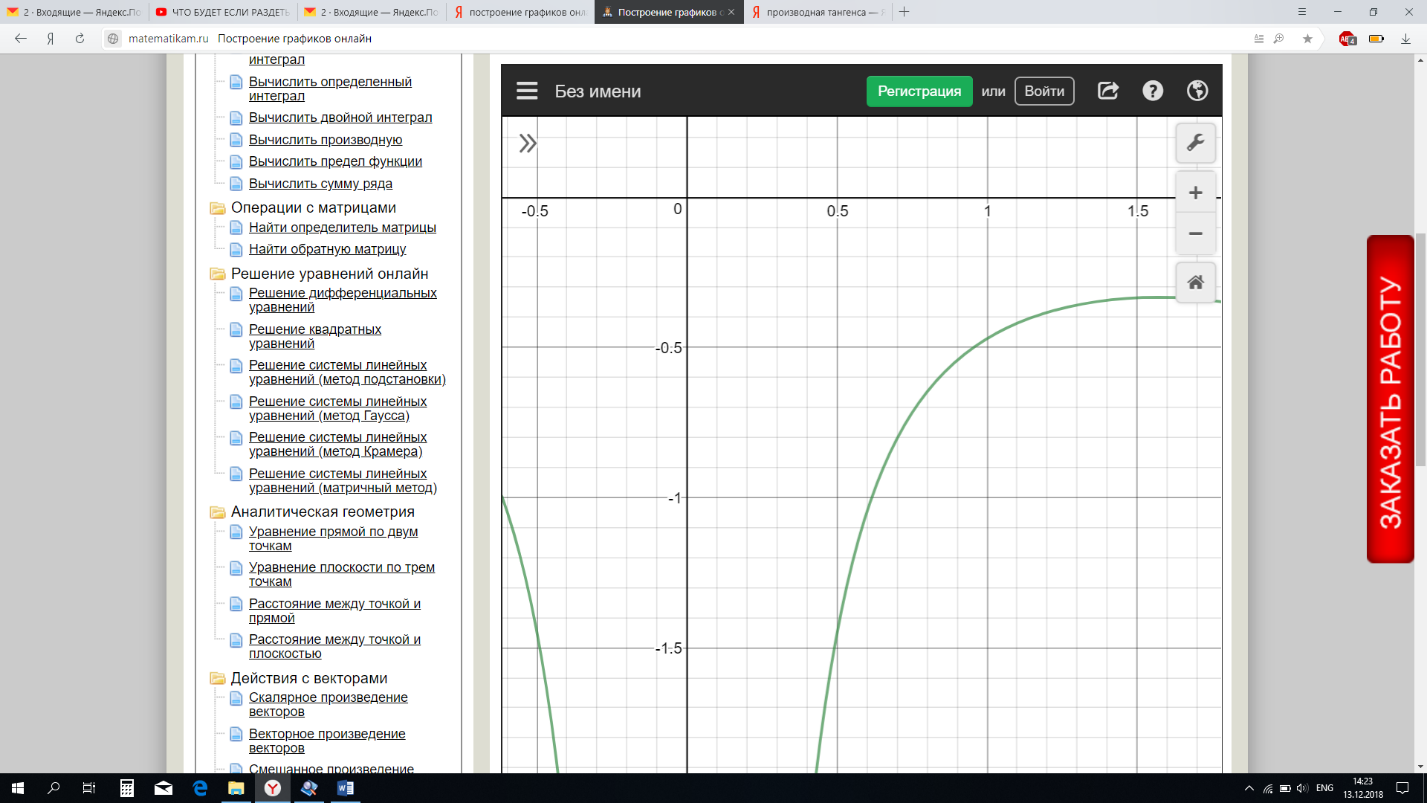
1 2.845868 2.845868 2.845868

2 NO 0.547161 0.547161

* 1. **Примечание**

Значение функции x\*tg(x) – 1/3 невозможно найти методом итераций, т.к. она не выполняется достаточное условие сходимости метода итераций. К данному выводу можно прийти, проанализировав график функции y = x\*tg(x) -1/3 представленным на рисунке 1, и график функции (1/3\*1/tan(x)) представленным на рисунку 2.

Рисунок 1 – график функции F(x) = x\*tan(x) -1/3

Рисунок 2 – график производной функции F(x)

1. **Заключение**

Изучая различные методы решения тех или иных задач, можно прийти к выводу, что иногда не все способы решения подходят для точного выполнения определённой задачи. Именно поэтому очень важно и нужно знать различные приемы, правила, с помощью которых можно реализовать решение одной задачи, а также нужно уметь выбрать среди них самый точный и эффективный алгоритм. Именно этим навыкам и обучает данная курсовая работа.

1. **Список использованной литературы**
2. Брайан Керниган, Деннис Ритчи. Язык программирования Си. Второе издани: Серия книг по программированию от Prentice Hall. 2009. – 292с.:ил.
3. Алгоритм метода половинного деления [Электронный ресурс]// <https://math.semestr.ru/> URL: https://math.semestr.ru/optim/dichotomy-algorithm.php
4. Математический энциклопедический словарь. — М.: «Сов. энциклопедия », 1988. — С. 847
5. Псевдографика [Эдектронный ресурс]// <https://ru.wikipedia.org/> URL: https://ru.wikipedia.org/wiki/Псевдографика