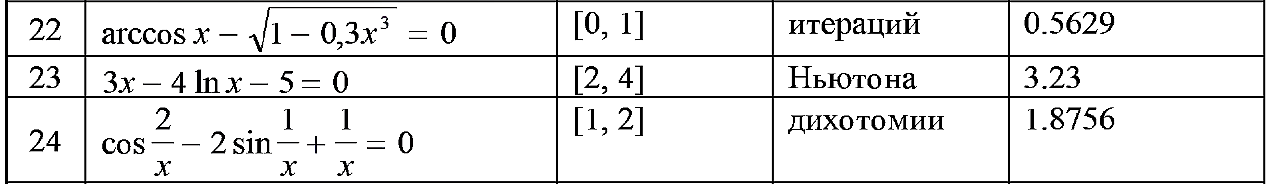
**Задача**

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений методами итераций, Ньютона и дихотомии. Нелинейные уравнения оформить как параметры-функции, разрешив относительно неизвестной величины в случае необходимости.

**Вариант**

****

**Общий метод решения**

Решить каждое из уравнений методом дихотомии, итераций, Ньютона или хорд. Передать в соответствующий метод в качестве аргументов исходную функцию (в случае метода Ньютона передать также производную) и отрезок, на котором будет производиться поиск корня.

Выразим для каждого варианта заданий функций вида *f(x) = x* для метода итераций и найдем производные для их использования в методе Ньютона.

Вариант 22:

f(x) = x:

Производная:

Вариант 23:

f(x) = x:

Производная:

Вариант 24:

f(x) = x:

Производная:

**Общие сведения о программе**

Аппаратное обеспечение: ноутбук

Операционная система: Ubuntu 16.04 LTS

Язык и система программирования: GNU C

Число строк программы: 88

Местонахождение файлов: */home/zero/УНИВЕР/МАИ/*

Компиляция программы в терминале: *gcc 4kp22.c -lm -o 4kp22.byebyeworld*

Запуск программы: *./4kp22.byebyeworld*

**Функциональное значение**

Программа предназначена для решение трансцендентных алгебраических уравнений различными численными методами (итераций, Ньютона и дихотомии). Значения ограничены переменными типа *double.*

**Описание логической структуры**

В необходимый метод поиска корня на отрезке подается сама функция, у которой нужно найти корень и значения концов самого отрезка. Далее в зависимости от самого метода осуществляется поиск корня до тех пор, пока погрешность измерений не будет меньше эпсилон, либо пока корень не будет найден точно.

**Описание переменных и констант**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| *eps* | *double* | Машинное эпсилон |
| *a* | Левая граница отрезка |
| *b* | Правая граница отрезка |
| *x, x1* | Вспомогательные переменные для методов итераций, Ньютона, хорд |
| *c* | Вспомогательная переменная для метода дихотомии |

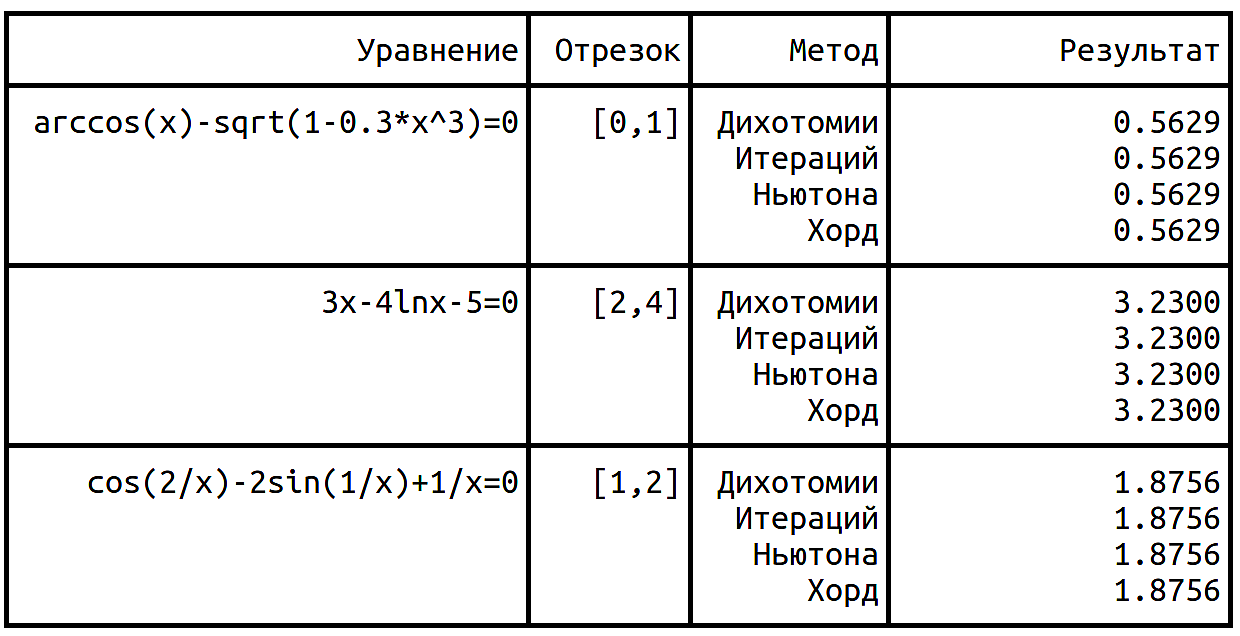
**Описание функций**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| *func22v1* | *double* | Функция из 22 варианта в исходном виде |
| *func22v2* | Функция из 22 варианта в виде *f(x)=x* |
| *deriv22* | Производная функции из 22 варианта |
| *func23v1* | Функция из 23 варианта в исходном виде |
| *func23v2* | Функция из 23 варианта в виде *f(x)=x* |
| *deriv23* | Производная функции из 23 варианта |
| *func24v1* | Функция из 24 варианта в исходном виде |
| *func24v2* | Функция из 24 варианта в виде *f(x)=x* |
| *deriv24* | Производная функции из 24 варианта |
| *dichotomy* | Нахождение корня методом дихотомии |
| *iteration* | Нахождение корня методом итераций |
| *newton* | Нахождение корня методом Ньютона |
| *chord* | Нахождение корня методом хорд |

**Входные данные**

Отсутствуют

**Выходные данные**



**Тестовые примеры**

Не предусмотрены.

**Дневник отладки**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Место** | **Событие** | **Действия по исправлению** |
| 18.12.2019 | дом | Ошибка в программе по синтаксису | *Исправление синтаксиса* |

**Выводы по задаче**

Написал программу на языке Си с использованием библиотеки «*math.h»* по решению трансцендентных алгебраических уравнений различными методами.

**Протокол**

daniel@daniel:~/$ cat head.txt

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

Курсовая работа №4

Процедуры и функции в качестве параметров.

Выполнил: Пивницкий Даниэль Сергеевич

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

daniel@daniel:~/$ cat 4kp22.c

#include <stdio.h>

#include <math.h>

double eps = 0.00001;

double func22v1(double x){

return(acos(x)-sqrt(1-0.3\*x\*x\*x));

}

double func22v2(double x){

return(cos(sqrt(1-0.3\*x\*x\*x)));

}

double deriv22(double x){

return((0.45\*x\*x)/sqrt(1-0.3\*x\*x\*x)-1/(sqrt(1-x\*x)));

}

double func23v1(double x){

return(3\*x-4\*log(x)-5);

}

double func23v2(double x){

return((4\*log(x)+5)/3);

}

double deriv23(double x){

return(3-4/x);

}

double func24v1(double x){

return(cos(2/x)-2\*sin(1/x)+1/x);

}

double func24v2(double x){

return(2/(acos(2\*sin(1/x)-(1/x))));

}

double deriv24(double x){

return((2\*sin(2/x)+2\*cos(1/x)-1)/(x\*x));

}

double dichotomy(double (double), double, double);

double iteration(double (double), double, double);

double newton(double (double), double (double), double, double);

double chord(double (double), double, double);

int main(){

printf("┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━┳━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━┓\n");

printf("┃ Уравнение┃ Отрезок┃ Метод┃ Результат┃\n");

printf("┣━━━━━━━━━━━━━━━━━━━━━━━━━━━━╋━━━━━━━━╋━━━━━━━━━━╋━━━━━━━━━━━━━━━━━━┫\n");

printf("┃ arccos(x)-sqrt(1-0.3\*x^3)=0┃ [0,1]┃ Дихотомии┃ %.4lf┃\n",dichotomy(func22v1,0,1));

printf("┃ ┃ ┃ Итераций┃ %.4lf┃\n",iteration(func22v2,0,1));

printf("┃ ┃ ┃ Ньютона┃ %.4lf┃\n",newton(func22v1, deriv22,0,1));

printf("┃ ┃ ┃ Хорд┃ %.4lf┃\n",chord(func22v1,0,1));

printf("┣━━━━━━━━━━━━━━━━━━━━━━━━━━━━╋━━━━━━━━╋━━━━━━━━━━╋━━━━━━━━━━━━━━━━━━┫\n");

printf("┃ 3x-4lnx-5=0┃ [2,4]┃ Дихотомии┃ %.4lf┃\n",dichotomy(func23v1,2,4));

printf("┃ ┃ ┃ Итераций┃ %.4lf┃\n",iteration(func23v2,2,4));

printf("┃ ┃ ┃ Ньютона┃ %.4lf┃\n",newton(func23v1, deriv23,2,4));

printf("┃ ┃ ┃ Хорд┃ %.4lf┃\n",chord(func23v1,2,4));

printf("┣━━━━━━━━━━━━━━━━━━━━━━━━━━━━╋━━━━━━━━╋━━━━━━━━━━╋━━━━━━━━━━━━━━━━━━┫\n");

printf("┃ cos(2/x)-2sin(1/x)+1/x=0┃ [1,2]┃ Дихотомии┃ %.4lf┃\n",dichotomy(func24v1,1,2));

printf("┃ ┃ ┃ Итераций┃ %.4lf┃\n",iteration(func24v2,1,2));

printf("┃ ┃ ┃ Ньютона┃ %.4lf┃\n",newton(func24v1,deriv24,1,2));

printf("┃ ┃ ┃ Хорд┃ %.4lf┃\n",chord(func24v1,1,2));

printf("┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━┻━━━━━━━━┻━━━━━━━━━━┻━━━━━━━━━━━━━━━━━━┛\n");

}

double dichotomy(double func(double), double a, double b){

double c;

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

c = (a+b)/2;

if(func(a)\*func(c) > 0)

a = c;

else

b = c;

}

return c;

}

double iteration(double Func(double), double a, double b){

double x = (a+b)/2;

double x1 = x+1;

while(fabs(x-x1) > eps){

x1 = x;

x = Func(x);

}

return x;

}

double newton(double func(double), double deriv(double), double a, double b){

double x = (a+b)/2;

double x1 = x+1;

while(fabs(x-x1) > eps){

x1 = x;

x = x - func(x)/deriv(x);

}

return x;

}

double chord(double func(double), double a, double b){

double x = a;

double x1 = b;

while(fabs(x-x1) > eps){

x1 = x;

x = (func(a)\*b-func(b)\*a)/(func(a)-func(b));

if(func(x)\*func(a) > 0)

a = x;

else

b = x;

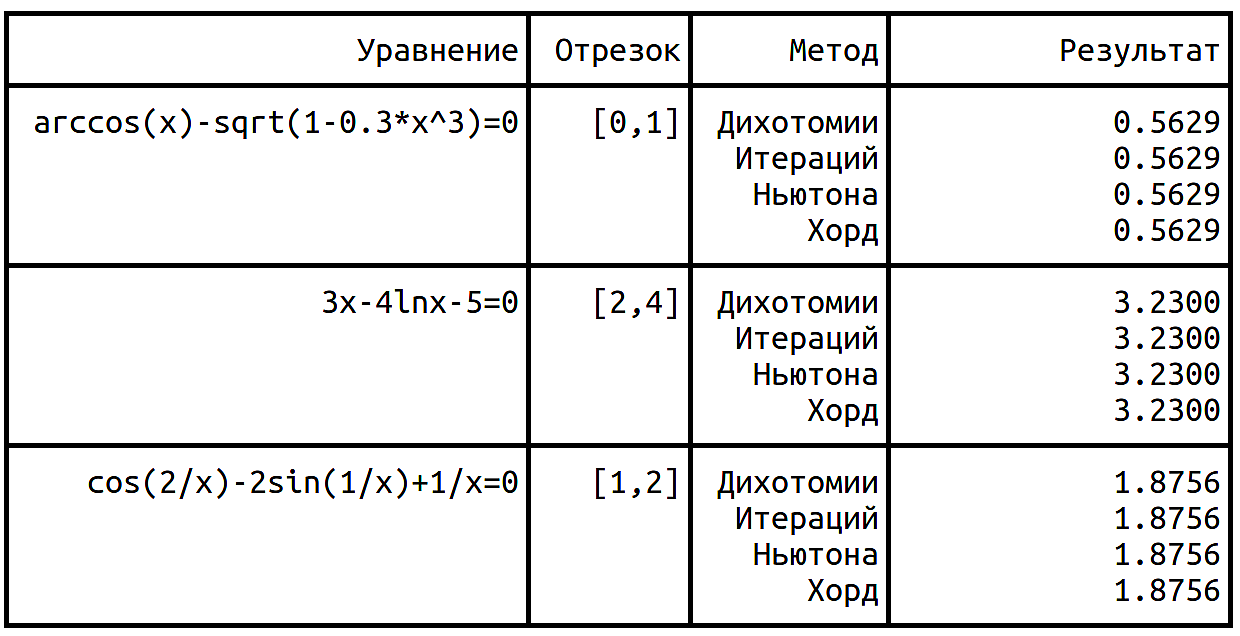
}

return(x);

}

daniel@daniel:~/$ gcc 4kp22.c -lm -o 4kp22.byebyeworld

daniel@daniel:~/$ ./4kp22.byebyeworld



**Заключение**

Мы научились составлять программу на языке Си с процедурами решения трансцендентных алгебраических уравнений методами итераций, Ньютона и дихотомии.

Написал программу на языке Си с использованием библиотеки «*math.h»* по решению трансцендентных алгебраических уравнений различными методами.