МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт»

(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Курсовая работа

по курсу «Вычислительные системы»

I семестр

Курсовой проект №4

«Процедуры и функции в качестве параметров»

|  |  |
| --- | --- |
| Группа: | М8О-107Б-22 |
| Студент: | Брюханов З.Д. |
| Преподаватель: | Аносова Н.П. |
| Оценка: |  |
| Дата: |  |

Москва, 2022

Оглавление

[Постановка задачи 3](#_Toc92671305)

[Вариант 5: 3](#_Toc92671306)

[Вариант 6: 3](#_Toc92671307)

[Теоретическая часть 4](#_Toc92671308)

[Метод дихотомии (половинного деления) 4](#_Toc92671309)

[Метод итераций (последовательных приближений) 5](#_Toc92671310)

[Метод Ньютона 8](#_Toc92671311)

[Графики 9](#_Toc92671312)

[Вариант 5 9](#_Toc92671313)

[Вариант 6 12](#_Toc92671314)

[Использованные в программе переменные 15](#_Toc92671315)

[Использованные в программе функции 16](#_Toc92671316)

[Исходный код программы: 17](#_Toc92671317)

[Входные данные 19](#_Toc92671318)

[Выходные данные 19](#_Toc92671319)

[Протокол с тестами 19](#_Toc92671320)

[Вывод 20](#_Toc92671321)

[Список литературы 20](#_Toc92671322)

# Постановка задачи

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными численными методами (итераций, Ньютона и половинного деления — дихотомия). Нелинейные уравнения оформить как параметры-функции, разрешив относительно неизвестной величины в случае необходимости. Применить каждую процедуру к решению двух уравнений, заданных двумя строками таблицы, начиная с варианта с заданным номером.

# Вариант 5:

Уравнение:



Отрезок: [0, 1];

Базовый метод: Дихотомии;

Приближённое значение корня: 0.5768;

# Вариант 6:

Уравнение:



Отрезок: [0.5, 1];

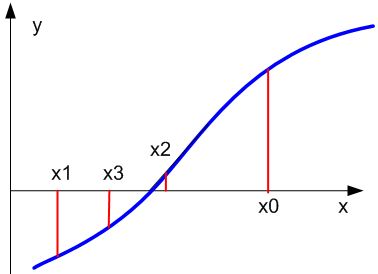
Базовый метод: итераций;

Приближённое значение корня: 0.9892;

# Теоретическая часть

## Метод дихотомии (половинного деления)

Уравнение имеет действительные корни на заданном отрезке, если функция на нем непрерывна и её значения на концах отрезка имеют разные знаки. Последовательно будем делить отрезок пополам и выбирать ту половину, на которой лежит корень (то есть значения функции на концах которого имеют разные знаки), пока не будет достигнута заданная точность.



*Рисунок 1 Геометрическая интерпретация метода*

## Метод итераций (последовательных приближений)

Метод итерации — численный метод решения математических задач, используемый для приближённого решения алгебраических уравнений и систем. Суть метода заключается в нахождении по приближённому значению величины следующего приближения (являющегося более точным). Метод позволяет получить решение с заданной точностью в виде предела последовательности итераций. Характер сходимости и сам факт сходимости метода зависит от выбора начального приближения решения.

Пусть задана функция *f(x),* требуется найти корни уравнения



Представим уравнение ( 1 ) в виде

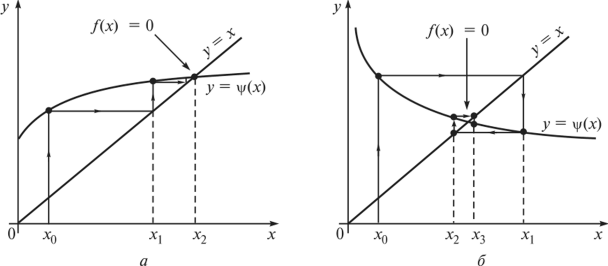


Рассмотрим последовательность чисел хk, которая определяется следующим образом:



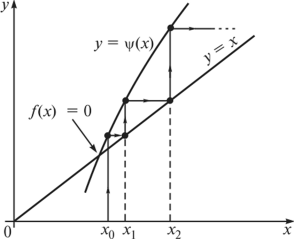
Метод простых итераций имеет следующую наглядную геометрическую интерпретацию (рис. 2).

Решением уравнения ( 2 ) будет абсцисса точки пересечения прямой с кривой . При выполнении итераций значение функции в точке *xi* необходимо отложить по оси абсцисс. Это можно сделать, если провести горизонталь до пересечения с прямой и из точки их пересечения опустить перпендикуляр на ось абсцисс. На рис. 2 показаны разные ситуации: а) сходимость к корню односторонняя; б) сходимость с разных сторон.



*Рисунок 2 Приближение к корню методом простой итерации*

На рис. 3 показан расходящийся процесс, при котором метод простой итерации не находит решения уравнения*.*



*Рисунок 3 Расходящийся процесс при методе итерации*

Достаточное условие сходимости метода:

,

Это условие необходимо проверить перед началом решения задачи, так как функция может быть выбрана неоднозначно, причём в случае неверного выбора указанной функции метод расходится.

В частности, сохраняет корни уравнение вида , если на исследуемом отрезке. Если в качестве выбрать константу того же знака, что и производная в окрестности корня, то мы получаем простейший метод итерации.

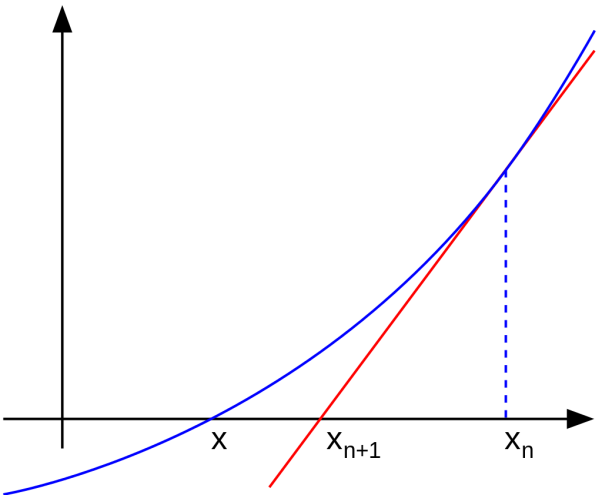
## Метод Ньютона

Метод Ньютона, алгоритм Ньютона (также известный как метод касательных) — это итерационный численный метод нахождения корня (нуля) заданной функции. Метод был впервые предложен английским физиком, математиком и астрономом Исааком Ньютоном (1643—1727). Поиск решения осуществляется путём построения последовательных приближений и основан на принципах простой итерации.

Если выбрать , то нахождение корня проходит более быстро, но требует вычисления производной.

Основная идея метода заключается в следующем: задаётся начальное приближение вблизи предположительного корня, после чего строится касательная к графику исследуемой функции в точке приближения, для которой находится пересечение с осью абсцисс. Эта точка берётся в качестве следующего приближения. И так далее, пока не будет достигнута необходимая точность.

Условие сходимости: на отрезке

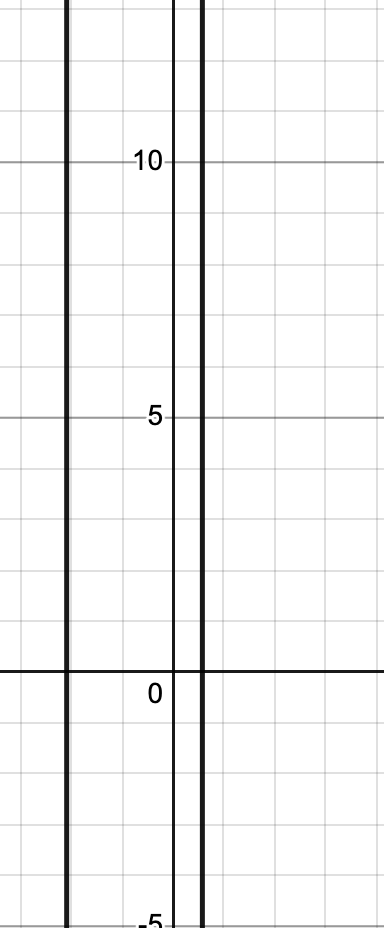


*Рисунок 4 Геометрическая интерпретация метода Ньютона*

# Графики

## Вариант 5

 [0;1]

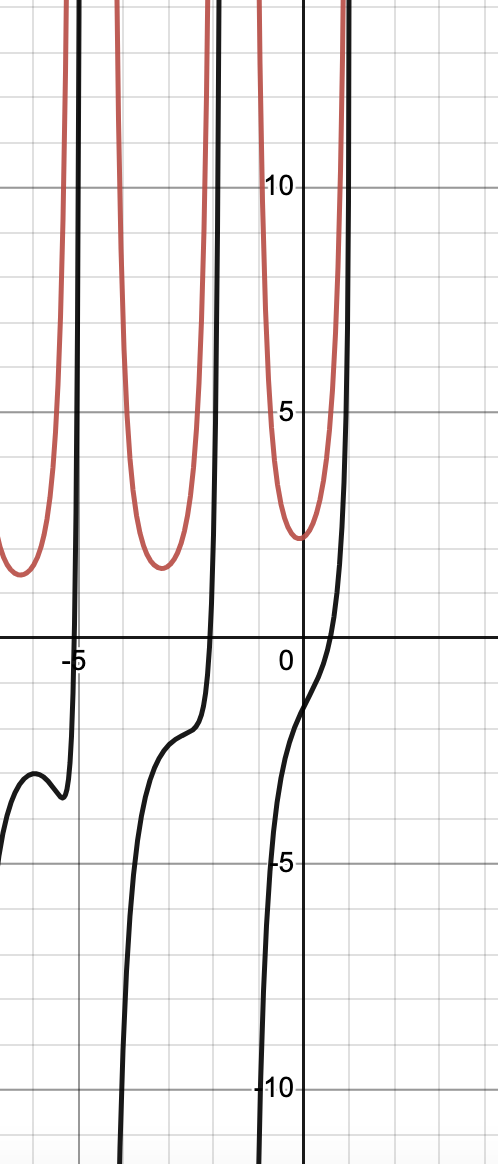
**

График

****

*< 1* на всём отрезке , поэтому условие сходимости метода итераций выполнено.

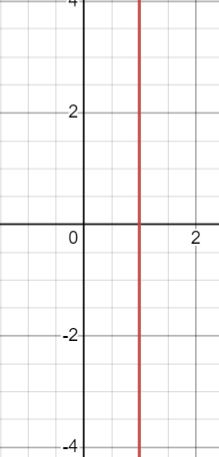
Графики (черный) и (красный)

****

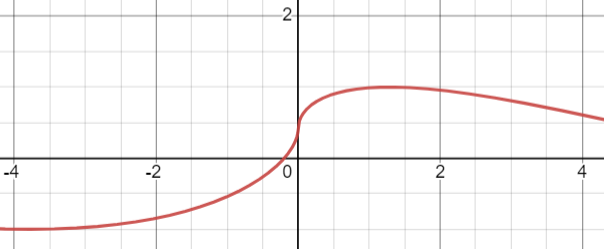
Условия сходимости по Ньютону на отрезке [0,1] выполняется.

## Вариант 6

 [0.5, 1]

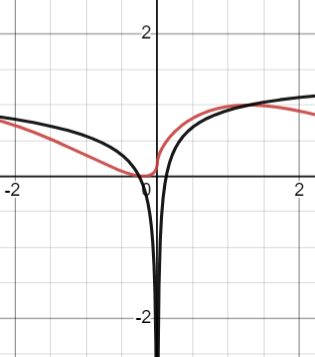
**

График

****

*<= 1* на всём отрезке , поэтому условие сходимости метода итераций выполнено.

Графики (черный) и (красный)

****

Условия сходимости по Ньютону на отрезке [0.5, 1] выполняется.

# Использованные в программе переменные

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Смысл переменной |
| eps | double | Машинное эпсилон. |

# Использованные в программе функции

|  |  |  |
| --- | --- | --- |
| Название функции | Тип возвращаемой переменной | Смысл функции |
| func1 | double | Возвращает функцию 5 варианта |
| func2 | double | Возвращает функцию 6 варианта |
| newton\_func1 | double | Возвращает производную функции 1 |
| newton\_func2 | double | Возвращает производную функции 2 |
| iter\_func1 | double | Возвращает выраженный x из функции 1 |
| iter\_func2 | double | Возвращает выраженный x из функции 2 |
| itter | double | Возвращает результат работы метода итераций |
| newton | double | Возвращает результат работы метода Ньютона |
| dth | double | Возвращает результат работы метода Дихотомии |
| ans | void | Выводит результат работы трёх методов |

# Исходный код программы:

#include <stdbool.h>

#include <ctype.h>

#include <stdio.h>

#include <math.h>

double func1(double x) {

return pow(1 - x, 0.5) - tan(x);

}

double func2(double x) {

return x + cos(pow(x, 0.52) + 2);

}

double iter\_func1(double x) {

return atan(pow(1-x,0.5));

}

double iter\_func2(double x) {

return -cos(pow(x, 0.52) + 2);

}

double newton\_func1(double x) {

return -0.5 \* pow(1 - x, -0.5) - (1 / (cos(x) \* cos(x)));

}

double newton\_func2(double x) {

return 1 - sin(pow(x, 0.52) + 2) \* 0.52 \* pow(x, -0.48);

}

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

double x;

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

x = (a + b) / 2;

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

b = x;

} else {

a = x;

}

}

return (a + b) / 2;

}

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

double x = (a + b) / 2;

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

x = f(x);

}

return x;

}

double newton(double F(double), double F1(double), double a, double b, double eps) {

double x = (a + b / 2);

while (fabs(F(x) / F1(x)) > eps) {

x -= F(x) / F1(x);

}

return x;

}

void ans(double F(double), double F1(double), double f(double), double a, double b, double eps) {

printf("Корень, полученный методом дихотомии: %11.7f\n", dth(F, a, b, eps));

printf("Корень, полученный методом итераций: %11.7f\n", itter(f, a, b, eps));

printf("Корень, полученный методом Ньютона: %11.7f\n", newton(F, F1, a, b, eps));

}

int main() {

double eps = 1e-5;

printf("Функция pow(1 - x, 0.5) - tan(x)\n");

ans(func1, newton\_func1, iter\_func1, 0, 1, eps);

printf("Функция x + cos(pow(x, 0.52) + 2)\n");

ans(func2, newton\_func2, iter\_func2, 0.5, 1, eps);

return 0;

}

# Входные данные

# Выходные данные

Программа должна вывести метод, корень, полученный этим методом.

# Протокол с тестами

# Var 5

Функция **pow(1 - x, 0.5) - tan(x)**

Корень, полученный методом дихотомии: **0.5767708**

Корень, полученный методом итераций: **0.5767660**

Корень, полученный методом Ньютона: **0.5767772**

**var 6**

Функция **x + cos(pow(x, 0.52) + 2)**

Корень, полученный методом дихотомии: **0.9891777**

Корень, полученный методом итераций: **0.9891797**

Корень, полученный методом Ньютона: **0.9891807**

# Вывод

В работе описаны идеи и принципы трёх численных методов решения уравнений: дихотомии, итераций и Ньютона. Проведены нужные вычисления для использования методов, построены графики данных заданиями функций. Составлена программа на языке Си. Описан формат ввода и вывода, составлен протокол тестирования программы.

В ходе написания курсовой работы №3 мной были изучены три численных метода решения уравнений, которые могут быть использованы, например, в приложении простого калькулятора для решения подобных уравнений. Помимо этого, я усовершенствовал свой навык написания программ на языке Си, что также крайне полезно.

# Список литературы

1. Информационный портал, метод дихотомии <https://www.km.ru/referats/2923FEFE93A540F6801A8AFA0BAE264A>
2. Электронный справочник, метод бисекции

<https://ru.wikipedia.org/wiki/Метод_бисекции>

1. Электронный справочник, метод Ньютона

<https://ru.wikipedia.org/wiki/Метод_Ньютона>

1. Информационный портал, метод Ньютона

<https://helpiks.org/3-15720.html>

1. Информационный портал, численные методы решения нелинейных уравнений

<https://prog-cpp.ru/digital-find/>

1. Информационный портал, метод простых итераций

<https://studref.com/654041/matematika_himiya_fizik/metod_iteratsiy>