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: | Диёров Д.У |
| Преподаватель: | Аносова Н.П. |
| Оценка: |  |
| Дата: | 18.12.2022 |

Москва, 2022

# Постановка задачи

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

# Вариант 10:

Уравнение:



Отрезок: [0.4, 1.0];

Приближённое значение корня: 0.6533

Теоретическая часть

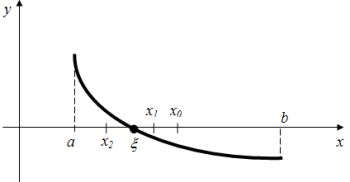
## Метод дихотомии (половинного деления)

Самым простейшим из методов уточнения корней является метод половинного деления, или метод дихотомии, предназначенный для нахождения корней уравнений, представленных в виде f (x)=0.

Пусть непрерывная функция f(x) на концах отрезка [a,b] имеет значения разных знаков, тогда на отрезке имеется хотя бы один корень.

Возьмем середину отрезка с=(a+b)/2. Если f(a)<f(c), то корень явно принадлежит отрезку от aдо (a+b)/2 и в противном случае от (a+b)/2 до b.

Поэтому берем подходящий из этих отрезков, вычисляем значение функции в его середине и т.д. до тех пор, пока длина очередного отрезка не окажется меньше заданной предельной абсолютной погрешности (b-a)<0



## Метод итераций (последовательных приближений)

это численный метод нахождения (одного) решения (с заданной точностью **ε**) нелинейного уравнения вида **f(x) = 0**.

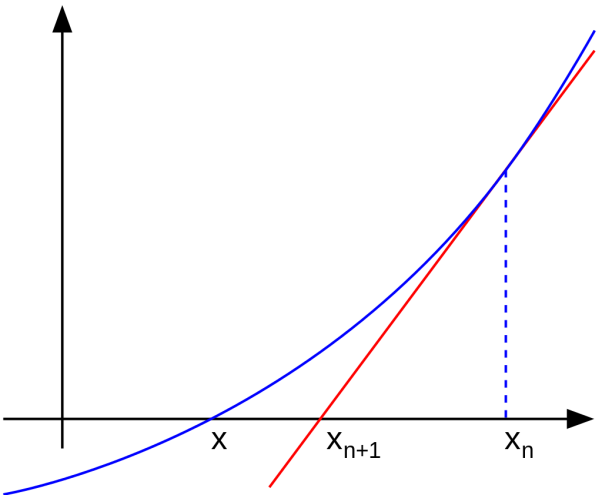
Суть метода итераций состоит в расчётах новой точки **x** (итерациях) по формуле **x = φ(x)**, которая выводится из уравнения **f(x)=0**. Итерации продолжаются до достижения необходимой точности решения **ε**. Метод итераций применим, если уравнение вида **f(x) = 0** сводится к уравнению вида **x = φ(x)** такому, что функция **φ(x)** непрерывна и дифференцируема на отрезке **[a; b]** и **max|φ’(x)|<1**. Для решения рассчитываются вспомогательные параметры **q** и **δ**, где **δ** — уточнённая точность. Сначала находим отрезок **[a; b]** такой, что функция **f(x)** непрерывна и меняет знак на отрезке, то есть **f(a)f(b)<0**.

## Метод Ньютона

предназначен для приближенного нахождения нулей функции, и сегодня мы не только узнаем его суть, но и научимся быстро решать тематическую задачу! В которой чаще всего фигурирует «обычная» функция одной переменной  и соответствующее уравнение . Например:

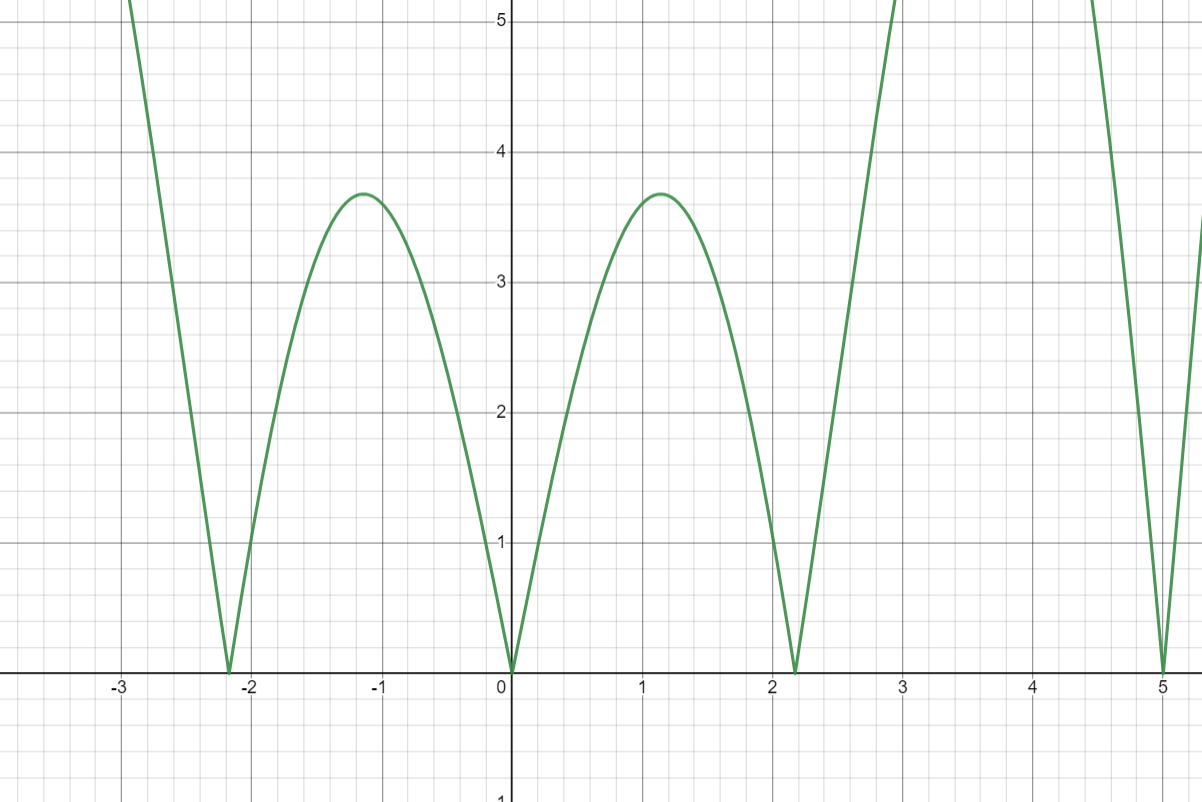


Поставим задачу отыскать действительные корни данного уравнения. График функции-многочлена  нечётной степени хотя бы один раз пересекает ось , следовательно, наше уравнение имеет по меньшей мере один действительный корень. Один. Или два. Или три.



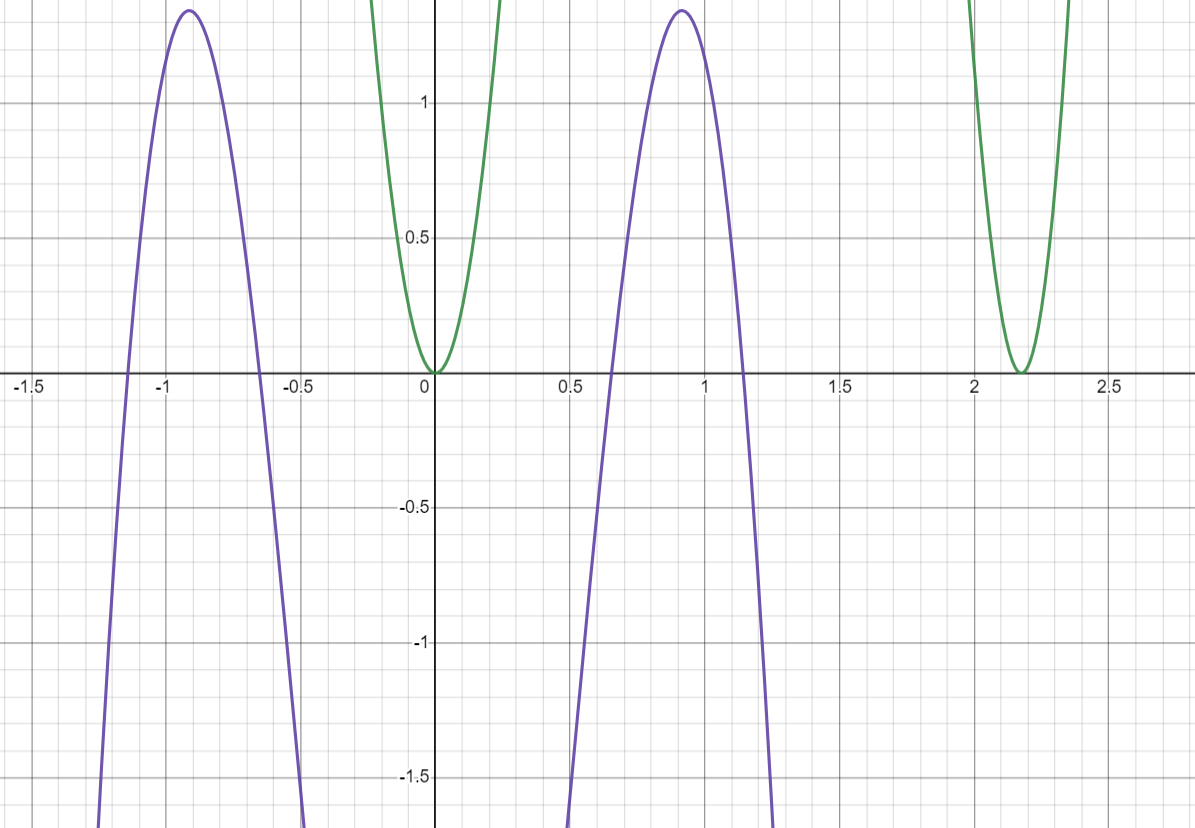
# Графики

## Вариант 10

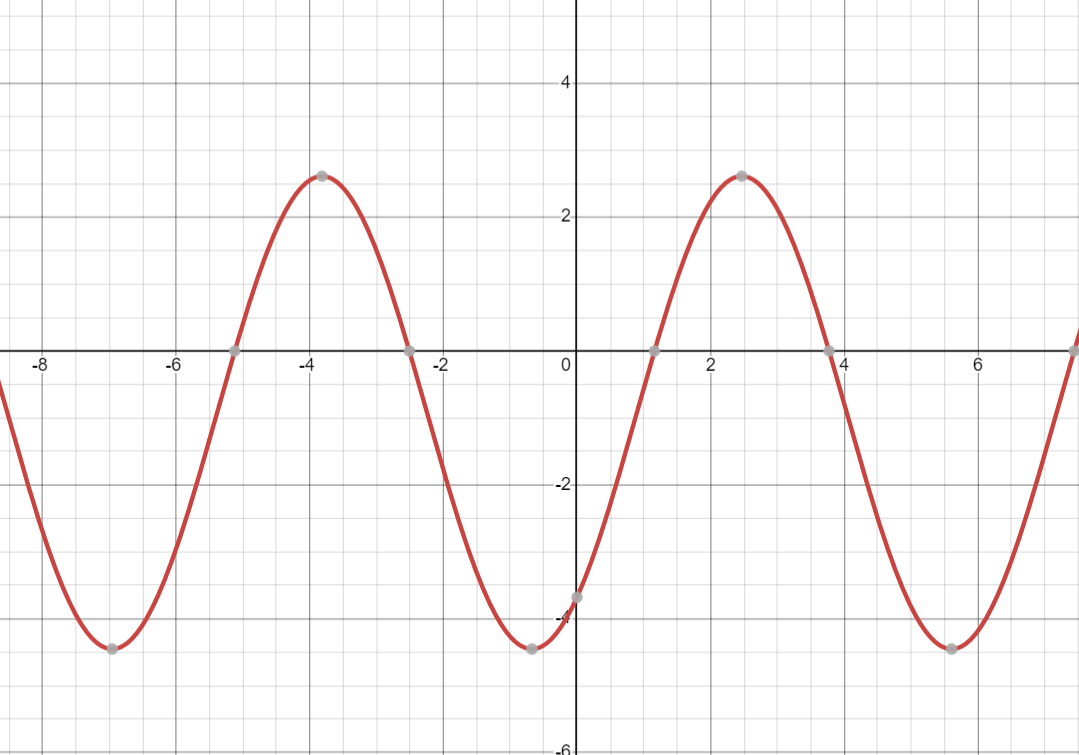


*< 1* на всём отрезке , поэтому условие сходимости метода итераций выполнено.

Графики (фиолетовый) и (зеленый)



< на всём отрезке поэтому условие сходимости метода Ньютона выполнено.



F(a)F(b)<0 на всём отрезке поэтому условие сходимости метода дихотомии выполнено

# Использованные в программе функции

|  |  |  |
| --- | --- | --- |
| Название функции | Тип возвращаемой переменной | Смысл функции |
| fun1 | double | Возвращает функцию 10 варианта |
| F1 | double | Возвращает производную функции |
| F2 | double | Возвращает 2 производную функции |
| itter | result | Возвращает результат работы метода итераций |
| newton | result | Возвращает результат работы метода Ньютона |
| dth | result | Возвращает результат работы метода Дихотомии |

# Исходный код программы:

double fun(double x)

{

return cos(x)/(2\*sin(x));

}

double F1(double x) {

return (2\*x\*sin(x)-cos(x));

}

double F2(double x) {

return cos(x) - exp(-pow(x, 2) / 2) + x - 1;

}

double f1(double x) {

return cos(x)/(2\*sin(x));

}

double f2(double x) {

return 1 + exp(-pow(x, 2) / 2) - cos(x);

}

double F1\_1(double x) {

return (3\*sin(x)+2\*x\*cos(x));

}

double F2\_1(double x) {

return -sin(x) + x \* exp(-pow(x, 2) / 2) + 1;

}

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

}

int main() {

double eps = 1;

while (1 + eps/2 > 1){

eps /= 2;

}

printf("Функция exp(x) + ln(x) - 10x\n");

ans(F1, F1\_1, f1, 0.4, 1.0, eps);

{

double a = 0.4, b = 1.0, r = 0.6532712, eps = 1.0, x = (a + b) / 2, c = fun(x);

int k = 0;

while (1.0 + eps > 1.0)

eps /= 2.0;

eps \*= 100;

while (fabs(x - c) >= eps)

{

k++;

x = c;

c = fun(x);

if(fabs(x-r)<=1e-7){

printf("Корень, полученный методом итераций: %11.7f\n", x);

return 0;

}

}

return 0;

}

}

# Входные данные

# Выходные данные

Программа должна вывести корень тремя методами

# Протокол с тестами

# satori@Satori-ubuntu:~/c$ gcc -std=c99 -Wall -pedantic kp3.c -lm -o main

# satori@Satori-ubuntu:~/c$ ./main

# Функция 2\*x\*sin(x)-cos(x)

# Корень, полученный методом дихотомии: 0.6532712

# Корень, полученный методом Ньютона: 0.6532712

# Корень, полученный методом итераций: 0.6532712

# 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

https://ru.wikipedia.org/wiki/Метод\_итерации