Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСТИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

РЕШЕНИЕ НЕЛИНЕЙНОГО УРАВНЕНИЯ С ОДНОЙ ПЕРЕМЕННОЙ

Отчет по лабораторной работе №2

по дисциплине «Численные методы»

Студент гр.728-2

\_\_\_\_\_\_\_\_\_ Д. Р. Геворгян

Принял

Доцент кафедры КИБЭВС

\_\_\_\_\_\_\_\_\_ С. С. Катаева

\_\_\_\_\_\_\_\_\_

1 Введение

Целью лабораторной работы является нахождение корней индивидуального уравнения вида f(x)=0 в два этапа – отделение и уточнение корня.

2 Ход работы

Индивидуальная функция имеет вид:

2.1 Отделение корней

Было проанализировано, что функция пересекает ось ОХ в точках -4.45, -0.913 и 0.712. Возьмём корень 0.712 (Рисунок 2.1). Интервал был взят [0 ; 1].

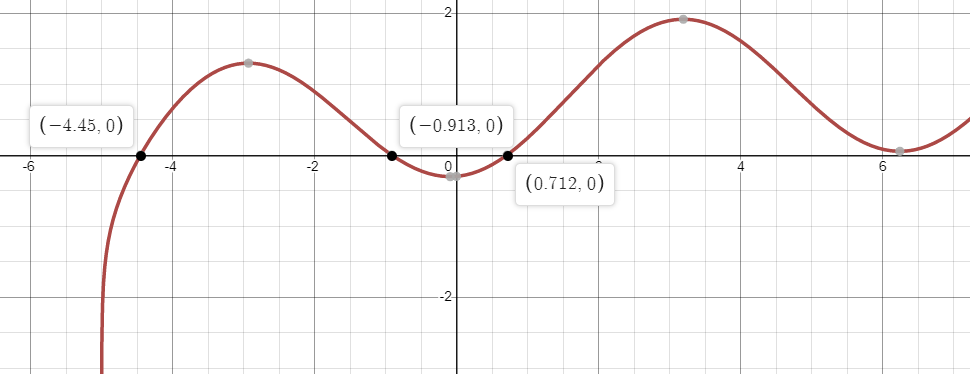


Рисунок 2.1 – График функции

2.2 Уточнение корня

2.2.1 Метод деления отрезка пополам

Суть метода деления отрезка пополам состоит в разбиении отрезка [a,b] на два отрезка,  определении знака функции в середине отрезка (a+b)/2 и выборе отрезка, на котором функция меняет знак и содержит решение. Деление продолжается до последующего необходимого приближения по формуле:

Была написана программа, реализующая данный метод, на С++:

#include "stdio.h"

#include <iostream>

#include <cmath>

using namespace std;

double func(double x)

{

return log10(x+5) - cos(x);

}

int main()

{

setlocale(LC\_ALL, "rus");

cout << "a=0; b=1:" << endl;

double a, b, e, x;

int i;

a = 0;

b = 1;

e = 0.01;

i = 0;

while (1) {

x = (a + b) / 2;

i++;

if (fabs(func(x)) < e)

break;

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

a = a; b = x;

}

else

{

a = x; b = b;

}

}

cout << "Корень уравнения: х=" << x << endl;

cout << "Количество итераций: " << i << endl;

system("pause");

return 0;

}

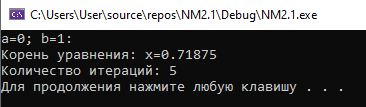
Результат работы программы продемонстрирован на рисунке 2.2.

Рисунок 2.1 - Результат

2.2.2 Метод Ньютона

В этом методе на отрезке [a; b] выбирается такое число , при котором имеет тот же знак, что и . Затем находится первое приближение по формуле .

Вычисления продолжаются пока не будет выполнено условие, где требуемая точность.

Была написана программа, реализующая данный метод, на С++:

#include "stdio.h"

#include <iostream>

#include <cmath>

using namespace std;

double f(double x)

{

return log10(x + 5) - cos(x);

}

double df(double x)

{

return 1 / ((x + 5) \* log(10)) + sin(x);

}

double d2f(double x)

{

return cos(x) - 1 / (pow((x + 5), 2)\*log(10));

}

int main()

{

setlocale(LC\_ALL, "rus");

cout << "Метод Ньютона\n";

double a = 0, b = 1, e = 0.01;

double x = 0;

double xn = 0;

int count = 0;

if (f(a) \* d2f(a) > 0) {

x = a;

}

else {

x = b;

}

double x0 = x;

xn = x - f(x) / df(x);

while (abs(f(x) - f(xn)) > e && abs(f(x)) > e) {

x = xn;

xn = x - f(x) / df(xn);

++count;

cout << "x = " << xn << endl;

}

cout << "Начальная точка: " << x0 << "\nКоличество итераций: " << count << "\nx = " << xn << endl;

system("pause");

return 0;

}

Результат работы программы продемонстрирован на рисунке 2.2.

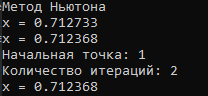


Рисунок 2.2 - Результат

2.2.3 Метод простых итераций

Метод состоит в приведении функции к виду x = φ(x). Это можно сделать, обеспечив сходимость с помощью формулы 2.3.

(2.3)

можно вычислить для данного уравнения путём следующих вычислений:

Точка является точкой начального приближения.

В общем виде формулу уточнения корня можно описать таким образом:

Корень будет уточняться до тех пор, пока не будет выполнено условие останова:

, где – допустимая погрешность. В данном случае он равна 0.01.

Была написана программа, реализующая данный метод, на С++:

#include "stdio.h"

#include <iostream>

#include <cmath>

using namespace std;

double f(double x)

{

return log10(x + 5) - cos(x);

}

double df(double x)

{

return 1 / ((x + 5) \* log(10)) + sin(x);

}

void simple\_iter(double lambda, double x1, double epsilon)

{

int iter = 0;

while(true)

{

double x0 = x1;

x1 = x0 - lambda \* f(x0);

iter += 1;

if (abs(x0 - x1) < epsilon)

break;

}

cout << "Корень равен " << x1 << "\nКоличество итераций равно " << iter << endl;

}

int main()

{

setlocale(LC\_ALL, "rus");

double x0 = 0.5;

double dfx0 = df(x0);

double lambda = 1 / dfx0;

cout << "Решение методом простых итераций\nlg(x+5)-cos(x)=0\nНачальное приближение равно " << x0 << endl;

simple\_iter(lambda, x0, 0.01);

system("pause");

return 0;

}

Результат работы программы (Рисунок 2.3):

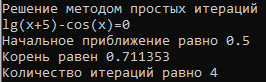


Рисунок 2.3 - Результат

Таблица 2.1 – Результаты измерений

|  |  |  |  |
| --- | --- | --- | --- |
|  | Решение | Начальная точка | Количество итераций |
| Метод деления отрезка пополам | 0.71875 | a=0  b=1 | 5 |
| Метод Ньютона | 0.712368 | 1 | 2 |
| Метод простых итераций | 0.711353 | 0.5 | 4 |

3 Заключение

В результате выполнения лабораторной работы были изучены теоретический материал по теме нелинейные уравнения и написаны программы, которые решают нелинейное уравнение тремя разными способами.