МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования



НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Вычислительные системы и технологии»

# ОТЧЁТ

по лабораторной работе №1

" РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ С ОДНОЙ НЕИЗВЕСТНОЙ"

по дисциплине

*Вычислительная математика*

(наименование дисциплины)

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Панкратова А.З.\_\_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Халеев А.А. \_

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_21-ВМз-4\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2023

**Тема работы:** решение нелинейных уравнений с одной неизвестной.

**Цель работы**: изучить численные методы и алгоритмы решения нелинейных уравнений.

**Постановка задачи:** решить нелинейное уравнение с одним неизвестным с использованием трех методов (метод половинного деления, метод Ньютона, метод простой итерации). Задание по вариантам. Точность ε=0.001

**Вариант №7:**



***Шаговый метод***Дано уравнение *f* (*x*) = 0 . Задана точка начала поиска *x*0 и длина интервала h*.* Требуется найти интервал [*a*,*b*] длиной h, содержащий первый корень уравнения, начиная с левой границы интервала поиска.  
*Алгоритм метода:*1. Установить интервал [a,b] на начало интервала поиска (a=x0).  
2. Определить координату точки b (b=a+h), а также значения функции в точках a и b:

F(a) и F(b).  
3. Проверить условие F(a)\*F(b)<0. Если условие не выполнено - передвинуть  
интервал [*a*,*b*]на один шаг (a=b) и перейти к пункту 2. Если условие выполнено -  
закончить алгоритм.  
 Рассмотрим нелинейное уравнение 

Отделим и локализуем корни, используя ***шаговый метод*** на отрезке [0; 1] с шагом 0.01  
***Программа на Python:***

def f(x: int) -> float:  
 *"""  
 Вычисляет значение функции f(x) = x \*\* 3 + 0.2 \* x \*\* 2 + 0.5 \* x - 1.2  
  
 Параметры:  
 x: Значение аргумента функции.  
  
 Возвращаемое значение:  
 Значение функции в точке x.  
 """* return x \*\* 3 + 0.2 \* x \*\* 2 + 0.5 \* x - 1.2  
  
  
def step\_method(func: callable, start: int, stop: int, step: float) -> tuple[int, int]:  
 *"""  
 Находит отрезок, содержащий корень функции.  
  
 Параметры:  
 func: Функция f(x), корень которой необходимо найти.  
 start: Начало отрезка поиска.  
 step: Шаг при переборе точек на отрезке.  
  
 Возвращаемое значение:  
 Кортеж с началом и концом сегмента, содержащего корень функции.  
  
 """* a, b = start, start + step  
 while func(a) \* func(b) > 0:  
 if b > stop:  
 raise ValueError("Функция не имеет корней в заданном отрезке")  
 a, b = b, b + step  
 return a, b  
  
  
def main():  
 *# Использование методов* a, b = step\_method(func=f, start=0, stop=1, step=0.01)  
 print(f"Корень уравнения находится на отрезке [{a:.2f}, {b:.2f}]")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Вывод программы:



Таким образом, на отрезке [0.85; 0.86] существует единственный корень уравнения  
 рассмотренного на интервале [0; 1].

После того как найден интервал, содержащий корень, применяют итерационные методы уточнения корня с заданной точностью.

Мы разберем следующие методы:  
1. метод половинного деления  
2. метод Ньютона (метод касательных)  
3. метод итерации

**Вывод:**

В данной лабораторной работе были произведены измерения и вычисления для определения сопротивлений трех резисторов.

Полученная погрешность незначительна. Следовательно, грубые ошибки при измерениях и вычислениях отсутствуют и данный метод вычисления можно использовать для определения сопротивления резистора.