**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э.Баумана)**

**Мытищинский филиал**

**ФАКУЛЬТЕТ КОСМИЧЕСКИЙ**

**КАФЕДРА К-1 САУ**

**ЛАБОРАТОРНАЯ РАБОТА №3**

**ПО ДИСЦИПЛИНЕ**

**“Численные методы”**

**НА ТЕМУ:**

**«Решение нелинейных уравнений»**

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**Студент К1-61Б**  **20.04.24 Тимофеев К. А.**

(Группа) (Подпись, дата) (ФИО)

**Руководитель**  **Чернова Т.В.**

(Подпись, дата) (ФИО)

*2024 г.*

# Задачи

1. Методом бисекций найти корень уравнения  с точностью  на отрезке [a, b] = [1, 2].
2. Ввести такие a, b, чтобы отрезок не содержал корней.
3. Найти второй корень на отрезке [a, b] = [-2, -1].
4. Округлить значения до 1е-4.
5. Выполнить вариант 12 из приложения к ЛР.
6. Найти корень  методом хорд.
7. Выполнить вариант 12 из приложения к ЛР методом хорд.

## Код программы

import numpy as np; import pandas as pd; import math; from tabulate import tabulate as tab

'''

Лабораторная работа №3

РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ МЕТОДАМИ БИСЕКЦИЙ(ДЕЛЕНИЯ ОТРЕЗКА ПОПОЛАМ)И ХОРД

'''

# 1) x^2-sqrt(x+4) = 0; e = 0.001, метод бисекций

eps: float = 0.001

a: float = 1

b: float = 2

def func(x: float) -> float:

    return math.pow(x, 2) - math.sqrt(x+4)

def bis(func, start, end, epsilon):

    # объявления массивов

    A = [start];  B = [end];  C = [(end+start)/2]

    funcA = []; funcB = []; funcC = [];  Err = [];  comm = []

    while True:  # цикл поиска корня на отрезке

        funcA.append(func(A[len(A)-1]))  # вычисляем значения функции

        funcB.append(func(B[len(B)-1]))  # в точках A[i], B[i], C[i]

        # len(A)-1 = i - последний элемент в массиве

        funcC.append(func(C[len(C)-1]))

        # проверка на наличие корня на отрезке

        if (funcA[len(funcA)-1] \* funcB[len(funcB)-1]) > 0.0:

            A[len(A)-1] = start

            B[len(B)-1] = end

            C[len(C)-1] = (end+start)/2

            Err.append('xxxxxxxx')

            comm.append("Roots does`t secured")  # корней на отрезке нет

            break

        # проверяем не достигли ли уже необходимой точности

        if abs(A[len(A)-1]-B[len(B)-1]) < epsilon:

            # round(x, 4) - округляем значения до 10^-4 для 4 задания

            Err.append(f'Root is: {round(C[len(C)-1], 4)}')

            comm.append("~('\*'-'\*')~")

            break

        else:  # если точность не достигнута записываем результат и идём дальше

            Err.append(round(abs(B[len(B)-1]-A[len(A)-1]), 4))

            comm.append(":-)")

        # если произведение значений функции в точках A[i] и С[i],

        if funcA[len(funcA)-1] \* funcC[len(funcC)-1] < 0.0:

            # продолжаем вычисление на отрезке A[i]C[i]

            A.append(A[len(A)-1])

            B.append(C[len(C)-1])

        # если произведение значений функции в точках B[i] и С[i],

        if funcB[len(funcB)-1] \* funcC[len(funcC)-1] < 0.0:

            # продолжаем вычисление на отрезке C[i]B[i]

            A.append(C[len(C)-1])

            B.append(B[len(B)-1])

        # новое C[i] для выбранного отрезка

        C.append((B[len(B)-1]+A[len(A)-1])/2)

    # датафрейм для хранения и визуализации результатов

    dt = pd.DataFrame({

        "A": A, "B": B, "C": C, "F(A)": funcA, "F(B)": funcB, "F(C)": funcC,

        "Error": Err, "Comment": comm})

return dt

print("-------Task №1-------\n",

      "x^2 - sqrt(x+4) = 0; \n",

      tab(bis(func, a, b, eps),

          headers='keys', tablefmt='psql', showindex=False))

# 2) x^2-sqrt(x+4) = 0; e = 0.001, метод бисекций / проверка на корректность

print("-------Task №2-------\n",

      "x^2 - sqrt(x+4) = 0; \n",

      tab(bis(func, 8, 10, eps),

          headers='keys', tablefmt='psql', showindex=False))

# 3) x^2-sqrt(x+4) = 0; e = 0.001, метод бисекций / левый корень

print("-------Task №3-------\n",

      "x^2 - sqrt(x+4) = 0; \n",

      tab(bis(func, -2, -1, eps),

          headers='keys', tablefmt='psql', showindex=False))

# 4) x^2 - sqrt(x + 4) = 0; e = 0.001, метод бисекций / округляем до 10^-4

print("-------Task №4-------\n",

      "x^2 - sqrt(x+4) = 0; Rounded to 10^-4\n",

      tab(bis(func, a, b, eps),

          headers='keys', tablefmt='psql', showindex=False))

# 5) (12 вар.) x^3 + 3\*x^2 - 1 = 0; e = 0.001, метод бисекций / левый корень

def func\_12(x):

    return math.pow(x, 3) + 3 \* math.pow(x, 2) - 1

print("-------Task №5-------\n",

      "x^3 + 3\*x^2  - 1 = 0; \n",

      tab(bis(func\_12, -3, 1, eps),

          headers='keys', tablefmt='psql', showindex=False))

# 6) x^2 - sqrt(x + 4) = 0; e = 0.001, метод хорд

def hord(func, start, end, epsilon):  # поиск корня ур-ия методом хорд по заданной точности

    X = [start];  Err = ['xxxxxxxx'];  comm = [':-0']

    if func(start) \* func(end) > 0:

        return None

    while True:

        X.append(X[len(X)-1] - (func(X[len(X)-1]) \*

                 (end - X[len(X)-1]) / (func(end) - func(X[len(X)-1]))))

      if abs(X[len(X)-2] - X[len(X)-1]) < epsilon:

            Err.append(f'Root is: {round(X[len(X)-1], 4)}')

            comm.append("~('\*'-'\*')~")

            break

        else:

            Err.append(abs(X[len(X)-1] - X[len(X)-2]))

            comm.append(":-)")

    dt = pd.DataFrame({

        "X": X, "Error": Err,  "Comment": comm

    })

    return dt

print("-------Task №6-------\n",

      "x^2 - sqrt(x+4) = 0;\n",

      tab(hord(func, 1, 2, eps),

          headers='keys', tablefmt='psql', showindex=False))

# 7) (12 вар.) x^3 + 3\*x^2 - 1 = 0; e = 0.001, метод хорд

print("-------Task №7-------\n",

      "x^3 + 3 \* x^2 - 1 = 0;\n\tRoots:")

roots = np.roots([1, 3, 0, -1])

print(roots)

j: int = 1

for i in roots:

    print(f"------- for {j}`st root in [{i-1} ; {i+1}]-------\n",

          tab(bis(func\_12, i-1, i+1, eps),

              headers='keys', tablefmt='psql', showindex=False))

j += 1

## Вывод программы:







