



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ**

**Кафедра информатики, математического и компьютерного
моделирования**

ОТЧЕТ

к лабораторной работе №1 по дисциплине
«Вычислительная математика»

Направление подготовки
01.03.02 «Прикладная математика и информатика»

Выполнил студент

гр. Б9120-01.03.02

Агличиев А.О.

(ФИО)

(подпись)

« 5 » марта 2022 г.

г. Владивосток
2023

Содержание

1	Задание 1	3
1.1	Постановка задачи	3
1.2	Решение	3

1 Задание 1

1.1 Постановка задачи

Найти число обусловленности(μ) и зависимость μ от \mathcal{E} системы линейных уравнений: $(A + N\mathcal{E})X = E$, где A – матрица Гильберта, N – размерность матрицы, $\mathcal{E} \in [0, 1]$

1.2 Решение

Матрица Гильберта - квадратная матрица с элементами:

$$A_{ij} = \frac{1}{i + j - 1}$$

Для нахождения чисел обусловленности напишем программу на Python. Непосредственно вычислением чисел обусловленности будет производить функция `cond` из библиотеки `numpy`. Она вычисляет путем умножения второй нормы матрицы на вторую норму обратной матрицы. В результате получили следующие значения:

1. $\mathcal{E} = 0.0, \quad \mu = 16024413500363.82$
2. $\mathcal{E} = 0.1, \quad \mu = 89912684538852.02$
3. $\mathcal{E} = 0.2, \quad \mu = 168778690000890.22$
4. $\mathcal{E} = 0.3, \quad \mu = 247345485434015.62$
5. $\mathcal{E} = 0.4, \quad \mu = 326288866751882.1$
6. $\mathcal{E} = 0.5, \quad \mu = 406609448878607.1$
7. $\mathcal{E} = 0.6, \quad \mu = 484440225993016.06$
8. $\mathcal{E} = 0.7, \quad \mu = 565004615280073.2$

9. $\mathcal{E} = 0.8, \quad \mu = 641287468678821.8$
10. $\mathcal{E} = 0.9, \quad \mu = 721077365256902.4$
11. $\mathcal{E} = 1.0, \quad \mu = 801803862815113.0$

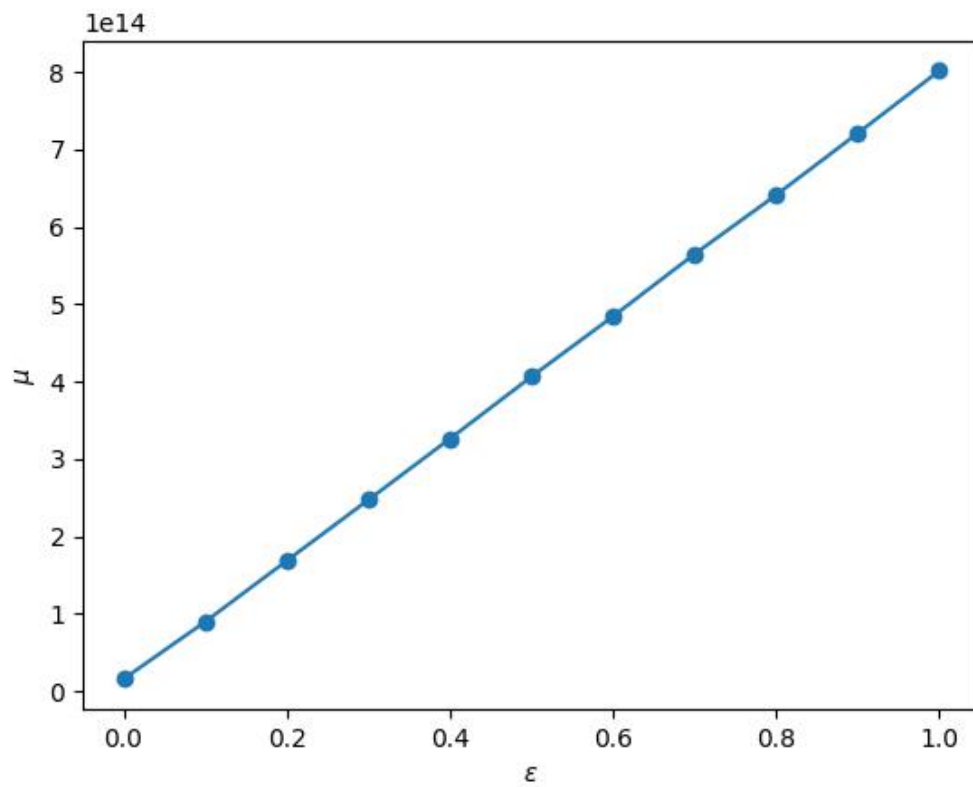


Рис. 1: График зависимости μ от \mathcal{E}

Код программы:

```
import numpy as np
import matplotlib.pyplot as plt

N = 10
START = 0
END = 1.1
STEP = .1

def build_hilbert_matrix(epsilon):
    hilbert_matrix = [[(1 / (i + j + 1) + N*epsilon) for j in range(N)] for i
        in range(N)]
    return np.matrix(hilbert_matrix)

def get_conds():
    epsilons = np.arange(START, END, STEP)
    return [np.linalg.cond(build_hilbert_matrix(epsilon)) for epsilon in
        epsilons]

if __name__ == '__main__':
    epsilons = np.arange(START, END, STEP)
    conds = get_conds()
    print(conds)
    plt.plot(epsilons, conds, marker='o')
    plt.ticklabel_format(useOffset=False)
    plt.xlabel('$\epsilon$')
    plt.ylabel('$\mu$')
    plt.savefig('1.jpg')
```