Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе № 2**

Дисциплина: Вычислительная математика

Выполнил студент гр. 3530901/10003 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Я.А. Иванов

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Н. Цыган

(подпись)

“27” марта 2023 г.

Санкт-Петербург

2023

Оглавление

[Задание: 2](#__RefHeading___Toc1279_301334972)

[Инструменты: 2](#__RefHeading___Toc1281_301334972)

[Ход выполнения работы: 2](#__RefHeading___Toc1283_301334972)

[*Порядок действий:* 2](#__RefHeading___Toc1285_301334972)

[*Первая задача:* 2](#__RefHeading___Toc1287_301334972)

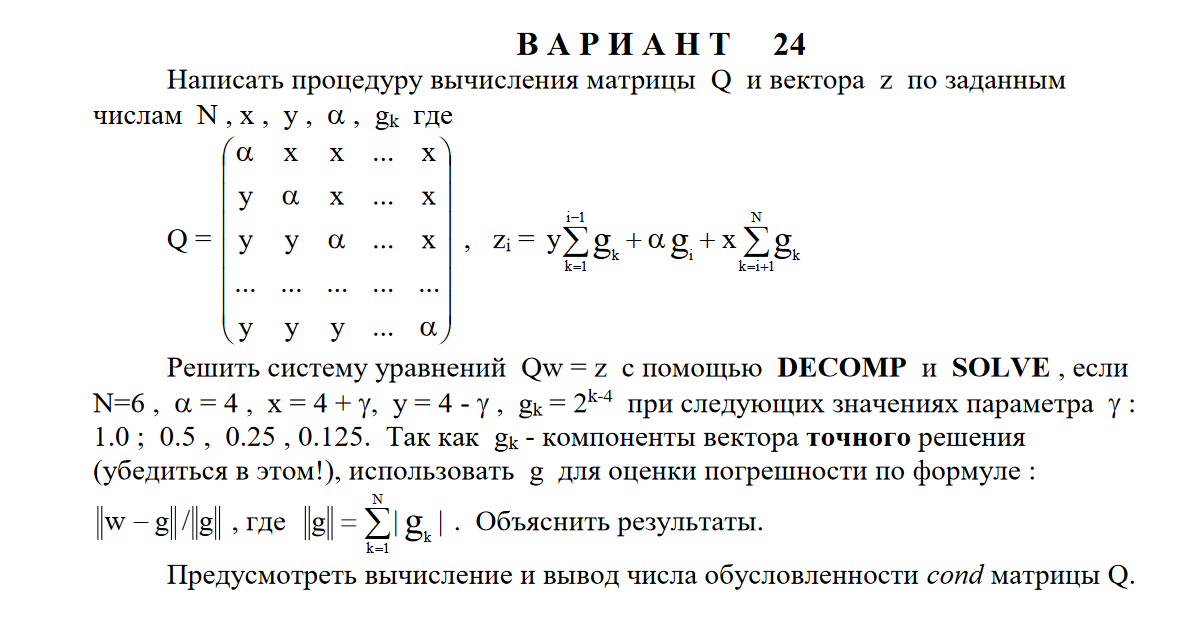
[*Вторая задача:* 2](#__RefHeading___Toc1289_301334972)

[Третья задача 3](#__RefHeading___Toc1541_301334972)

[Вывод: 5](#__RefHeading___Toc1291_301334972)

[Ссылки: 5](#__RefHeading___Toc1293_301334972)

# Задание:

**Вариант 24:**

# Инструменты:

Для работы был выбран язык программирования Python версии 3.09 ввиду наличия необходимых библиотек для выполнения поставленной задачи, а именно:

* NumPy – для большей скорости расчетов
* pandas – для красивого вывода в консоль таблицы
* MatplotLib – для вывода графиков

# Ход выполнения работы:

Установим константы в соответствии с условиями задачи:

# устанавливает размерность матрицы Q и вектора z.  
N = 6  
# устанавливает значение элемента a матрицы Q  
a = 4  
# устанавливает значения параметра gamma, для которых нужно решить систему уравнений.  
gamma\_values = [1.0, 0.5, 0.25, 0.125]

создает матрицу Q и вектор z

Q = numpy.zeros((N, N))  
z = numpy.zeros(N)

заполняет матрицу Q соответствующими значениями из заданных формул.

x = 4 + gamma  
y = 4 - gamma  
g = [2 \*\* (k - 4) for k in range(1, N + 1)]

for i in range(N):  
 z[i] = y \* sum(g[:i]) + a \* g[i] + x \* sum(g[i + 1:])  
 for j in range(N):  
 if i > j:  
 Q[i][j] = y  
 elif i == j:  
 Q[i][j] = a  
 else:  
 Q[i][j] = x

вычисляет числа обусловленности матрицы Q.

cond = numpy.linalg.cond(Q)  
P, R = numpy.linalg.qr(Q)

использует для решения системы уравнений Qw = z с помощью метода LU-разложения

w = numpy.linalg.solve(R, numpy.dot(P.T, z))

Выполним оценку погрешности по формуле и сохраним результат

error = numpy.linalg.norm(w - g) / numpy.linalg.norm(g)  
# добавляет данные о текущем значении gamma в соответствующие списки  
plot\_data["gamma"].append(gamma)  
plot\_data["cond"].append(cond)  
plot\_data["error"].append(error)

Перейдём к представлению результатов:

# Построение таблицы  
results = pd.DataFrame({  
 'Gamma': plot\_data["gamma"],  
 'cond(Q)': plot\_data["cond"],  
 'Error': plot\_data["error"]  
})  
  
print(results.to\_string(index=False))  
  
# Построение графика  
plt.plot(plot\_data["gamma"], plot\_data["cond"], label="cond(Q)")  
plt.plot(plot\_data["gamma"], plot\_data["error"], label="Error")  
plt.xlabel("Gamma")  
plt.legend()  
plt.savefig("img.jpg")  
plt.show()

Вывод программы:

Таблица:

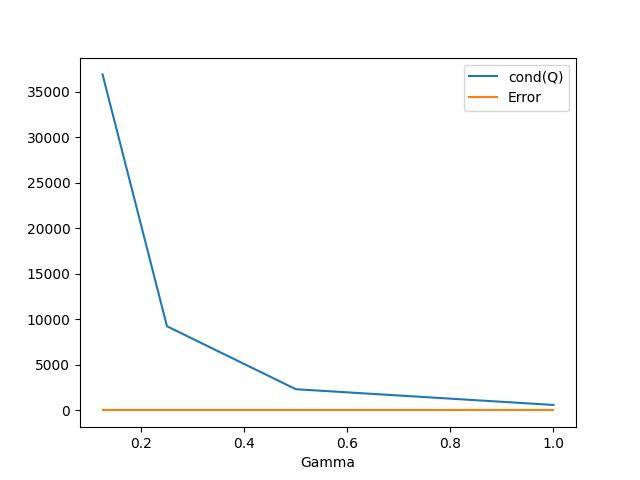
Gamma cond(Q) Error

1.000 599.188026 9.792577e-15

0.500 2327.296157 6.435579e-15

0.250 9239.323984 3.065154e-13

0.125 36887.330993 1.168568e-12

График:

# Вывод:

Из данной работы можно сделать вывод, что числа обусловленности матрицы могут сильно варьироваться в зависимости от выбранного параметра гамма, что может повлиять на точность решения системы уравнений. В данном конкретном примере, при значениях гамма, близких к 1, числа обусловленности матрицы Q были крайне большими, что привело к большой относительной погрешности решения системы уравнений.

# Ссылки:

Листин кода на github: <https://github.com/vitaya-para/lab2_2023/blob/main/main.py>