

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
Инструменты:	2
Ход выполнения работы:	2
<i>Порядок действий:</i>	2
<i>Первая задача:</i>	2
<i>Вторая задача:</i>	2
<i>Третья задача:</i>	3
Вывод:	5
Ссылки:	5

Задание:

ВАРИАНТ 24

Написать процедуру вычисления матрицы Q и вектора z по заданным числам N, x, y, α, g_k где

$$Q = \begin{pmatrix} \alpha & x & x & \dots & x \\ y & \alpha & x & \dots & x \\ y & y & \alpha & \dots & x \\ \dots & \dots & \dots & \dots & \dots \\ y & y & y & \dots & \alpha \end{pmatrix}, \quad z_i = y \sum_{k=1}^{i-1} g_k + \alpha g_i + x \sum_{k=i+1}^N g_k$$

Решить систему уравнений $Qw = z$ с помощью **DECOMP** и **SOLVE**, если $N=6, \alpha=4, x=4+\gamma, y=4-\gamma, g_k=2^{k-4}$ при следующих значениях параметра γ : 1.0; 0.5, 0.25, 0.125. Так как g_k - компоненты вектора **точного** решения (убедиться в этом!), использовать g для оценки погрешности по формуле:

$$\|w - g\| / \|g\|, \text{ где } \|g\| = \sum_{k=1}^N |g_k|. \text{ Объяснить результаты.}$$

Предусмотреть вычисление и вывод числа обусловленности *cond* матрицы Q .

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.

В данной задаче будем использовать норму Фробениуса. Для этого используем `numpy.linalg.norm`. Согласно документации, вычисление нормы нам не нужно задавать дополнительных параметров

```
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)
```

Для проверки корректности работы программы, выведем значения z,w,g

```
#выведем промежуточные значения
print("z:")
print(z)

print("g:")
print(g)

print("w:")
print(w)
```

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.xlabel("Gamma")
plt.legend()
```

```
plt.savefig("img1.jpg")
plt.show()

plt.plot(plot_data["gamma"], plot_data["error"], label = "Error")
plt.xlabel("Gamma")
plt.legend()
plt.savefig("img2.jpg")
plt.show()
```

Вывод программы:

Gamma 1.0

z:

[39.25 38.875 38.125 36.625 33.625 27.625]

g:

[0.125 0.25 0.5 1. 2. 4.]

w:

[0.125 0.25 0.5 1. 2. 4.]

Gamma 0.5

z:

[35.375 35.1875 34.8125 34.0625 32.5625 29.5625]

g:

[0.125 0.25 0.5 1. 2. 4.]

w:

[0.125 0.25 0.5 1. 2. 4.]

Gamma 0.25

z:

[33.4375 33.34375 33.15625 32.78125 32.03125 30.53125]

g:

[0.125 0.25 0.5 1. 2. 4.]

w:

[0.125 0.25 0.5 1. 2. 4.]

Gamma 0.125

z:

[32.46875 32.421875 32.328125 32.140625 31.765625 31.015625]

g:

[0.125 0.25 0.5 1. 2. 4.]

w:

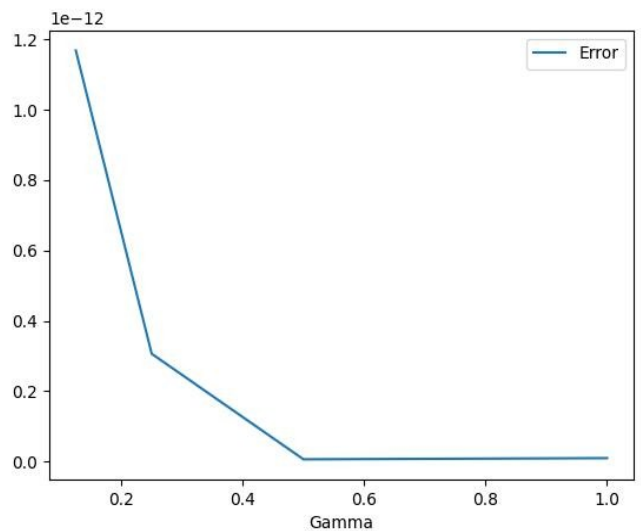
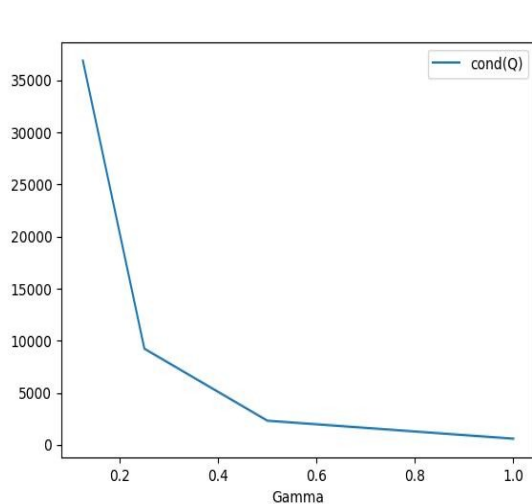
[0.125 0.25 0.5 1. 2. 4.]

Result table:

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

По промежуточным значениям можно понять, что точное решение совпадает с решения системы уравнений

Графики:



///

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 стремительно уменьшается, также стремительно уменьшается и значение погрешности

Ссылки:

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