Лабораторная работы 5

Вычисления числа обусловленности матрицы

Задание 1 Исследовать зависимость относительной погрешности решение системы уравнения от величины числа обусловленности при различных нормах погрешности A и f

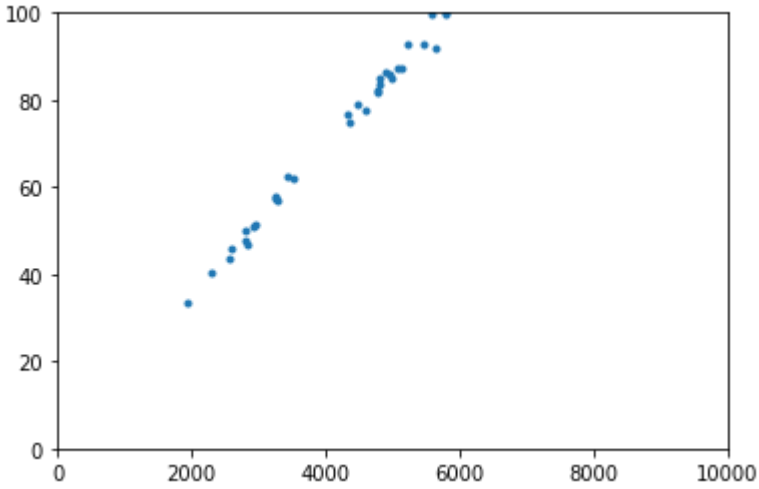
Задание 2 Построить трехмерные графики зависимости ||delta x||/||x|| от числа обсуловленности и ||delta A||/||A||, ||delta f||/||f||

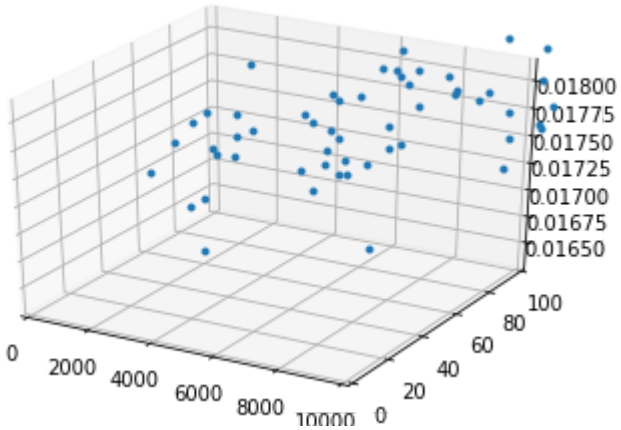
Задание 3 Добавить в решение задания 1 график зависимости погрешностей вычисления ||delta x||/||x|| от числа обсуловленности и сравнить

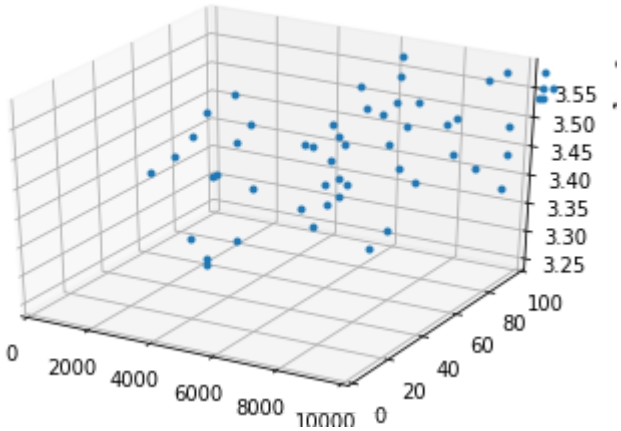
```
import random

import numpy as np
from matplotlib import pyplot as plt


def norma_matrix(m):
    max = -1
    sum_vec = []
    for i in range(0, len(m)):
        sum = 0
        for j in range(0, len(m)):
            sum += abs(m[i][j])
        sum_vec.append(sum)

    for x in sum_vec:
        if max <= x:
            max = x
    return max


def fill_matrix(n, a, b):
    m = [[random.uniform(-a, b) for i in range(n)] for j in range(n)]
    return m


def norma_vec(vec):
    max = -1
    for x in vec:
        if max <= abs(x):
            max = x
    return max


def calc_diff(delta_f, delta_a, x, n):
    a = fill_matrix(n, 0, 10)
    f = np.matmul(a, x)
```

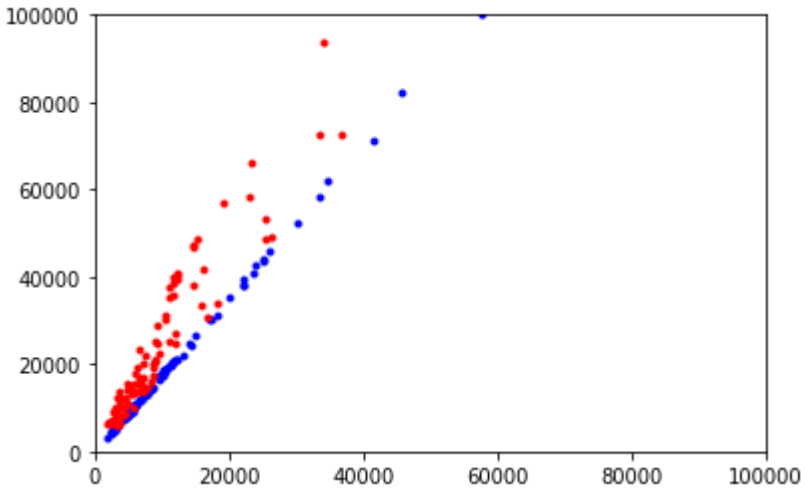
```
    norma_a = norma_matrix(a)
    norma_delta_f = norma_vec(delta_f)
    norma_f = norma_vec(f)
    norma_delta_a = norma_matrix(delta_a)

    a_inv = np.linalg.inv(a)
    mu = norma_matrix(a_inv)*norma_a

    return mu, mu * ((norma_delta_f / norma_f) + (norma_delta_a / norma_a)) * 100,
           norma_delta_a, norma_a, norma_delta_f, norma_f


def generate_graphic_diff_and_mu():
    n = 100
    delta_f = np.empty(n)
    delta_f.fill(0.1)
    delta_a = np.empty((n,n))
    delta_a.fill(0.1)
    # print(delta_a)
    x = np.linspace(0, 100, 100)
    x_axis = []
    y_axis = []
    norma_delta_a_vec = []
    norma_a_vec = []
    norma_delta_f_vec = []
    norma_f_vec = []
    for i in range(0, 100):
        xi, yi, norma_delta_a, norma_a, norma_delta_f, norma_f = calc_diff(delta_f,
        x_axis.append(xi)
        y_axis.append(yi)
        norma_delta_a_vec.append(norma_delta_a)
        norma_a_vec.append(norma_a)
        norma_delta_f_vec.append(norma_delta_f)
        norma_f_vec.append(norma_f)

    draw_chart(x_axis, y_axis)
    draw_3d_chart(x_axis, y_axis, norma_delta_a_vec, norma_a_vec)
    draw_3d_chart(x_axis, y_axis, norma_delta_f_vec, norma_f_vec)
    return x_axis, y_axis


def generate_graphic_with_step_diff():
    n = 100
    x = np.ones(n)
    x_axis = []
    y_axis = []
    for i in range(0, 100):
        delta_f = np.empty(n)
        delta_f.fill(0.1 + (i * 0.001))
        delta_a = np.empty((n, n))
        delta_a.fill(0.1 + (i * 0.001))
        xi, yi, norma_delta_a, norma_a, norma_delta_f, norma_f = calc_diff(delta_f,
        x_axis.append(xi)
        y_axis.append(yi)
```

```
    x = np.array(x_axis)
    y = np.array(y_axis)
    return x, y


def draw_3d_chart(x, y, norma_delta, norma):
    x = np.array(x)
    y = np.array(y)
    y /= 100
    nd = np.array(norma_delta)
    n = np.array(norma)
    z = np.divide(nd, n)
    fig = plt.figure()
    ax = fig.add_subplot(projection='3d')
    ax.set_ylim([0, 100])
    ax.set_xlim([0, 10000])
    ax.plot(x, y, z, '.')

    plt.show()


def draw_chart(x, y):
    x = np.array(x)
    y = np.array(y)
    y /= 100
    plt.xlim(0, 10000)
    plt.ylim(0, 100)

    plt.plot(x, y, '.')
    plt.show()


def draw_two(x1, y1, x2, y2):
    plt.xlim(0, 100000)
    plt.ylim(0, 100000)
    plt.plot(x1, y1, '.', color="blue")
    plt.plot(x2, y2, '.', color="red")
    plt.show()


if __name__ == '__main__':
    x1, y1 = generate_graphic_diff_and_mu()
    x2, y2 = generate_graphic_with_step_diff()
    draw_two(x1, y1, x2, y2)
```


100
80
60
40
20
0
0
2000
4000
6000
8000
10000


0.01800
0.01775
0.01750
0.01725
0.01700
0.01675
0.01650
0
2000
4000
6000
8000
10000
0
20
40
60
80
100


3.55
3.50
3.45
3.40
3.35
3.30
3.25
0
2000
4000
6000
8000
10000
0
20
40
60
80
100


100000
80000
60000
40000
20000
0
0
20000
40000
60000
80000
100000

Платные продукты Colab - Отменить подписку

✓ 7 сек. выполнено в 20:19