Федеральное государственное автономное образовательное учреждение высшего образования
Национальный исследовательский университет
"Высшая школа экономики"

Московский институт электроники и математики имени А. Н. Тихонова
Программа "Прикладная математика"

Лабораторная работа №2

Решение систем линейных алгебраических уравнений прямыми методами. Теория возмущений

Группа БПМ213
Вариант 8
Номера выполняемых задач: 3.1.8, 3.2, 3.8.2

Выполнила:
Варфоломеева Анастасия Андреевна
Преподаватель:
Токмачев Михаил Геннадьевич

Москва, 2024 г.

1 Погрешности решения от погрешностей правой части системы

1.1 Формулировка задачи

Дана система уравнений $Ax = b$ порядка n. Исследовать зависимость погрешности решения x от погрешностей правой части системы b.

1.2 Порядок решения задачи

1. Задать матрицу системы A и вектор правой части b. Используя встроенную функцию, найти решение x системы $Ax = b$ с помощью метода Гаусса.
2. С помощью встроенной функции вычислить число обусловленности матрицы A.
3. Принимая решение x, полученное в п.1, за точное, вычислить вектор $d = (d_1, ..., d_n)^T$, $d_i = \frac{\|x - x^i\|_\infty}{\|x\|_\infty}$, $i = 1, ..., n$, относительных погрешностей решений x^i систем $Ax^i = b^i$, $i = 1, ..., n$, где компоненты векторов b^i вычисляются по формулам:

$$b_k^i = \begin{cases} b_k + \Delta, & k = i \\ b_k, & k \neq i \end{cases}$$

$k = 1, ..., n$ (Δ - произвольная величина погрешности).

4. На основе вычисленного вектора d построить гистограмму. По гистограмме определить компоненту b_m вектора b, которая оказывает наибольшее влияние на погрешность решения.

5. Оценить теоретически погрешность решения x^m по формуле: $\delta(x^m) \leq cond(A) \cdot \delta(b^m)$. Сравнить значение $\delta(x^m)$ со значением практической погрешности d_m. Объяснить полученные результаты.

УКАЗАНИЕ. Пусть функция $cond(A)$ возвращает число обусловленности матрицы A, основанное на ∞-норме. Для вычисления $\|\cdot\|_\infty$ вектора удобно воспользоваться встроенной функцией, возвращающей максимальную компоненту вектора v.

Компоненты вектора b во всех вариантах задаются формулой $b_i = N$, $\forall i = 1, ..., n$ коэффициенты $c = c_{ij} = 0.1 \cdot N \cdot i \cdot j$, $\forall i, j = 1, ..., n$, N - номер варианта

$$N = 8$$

$$n = 6$$

$$a_{ij} = \frac{1}{\sqrt{c^2 + 0.58 \cdot c}}$$

1.3 Код на Python

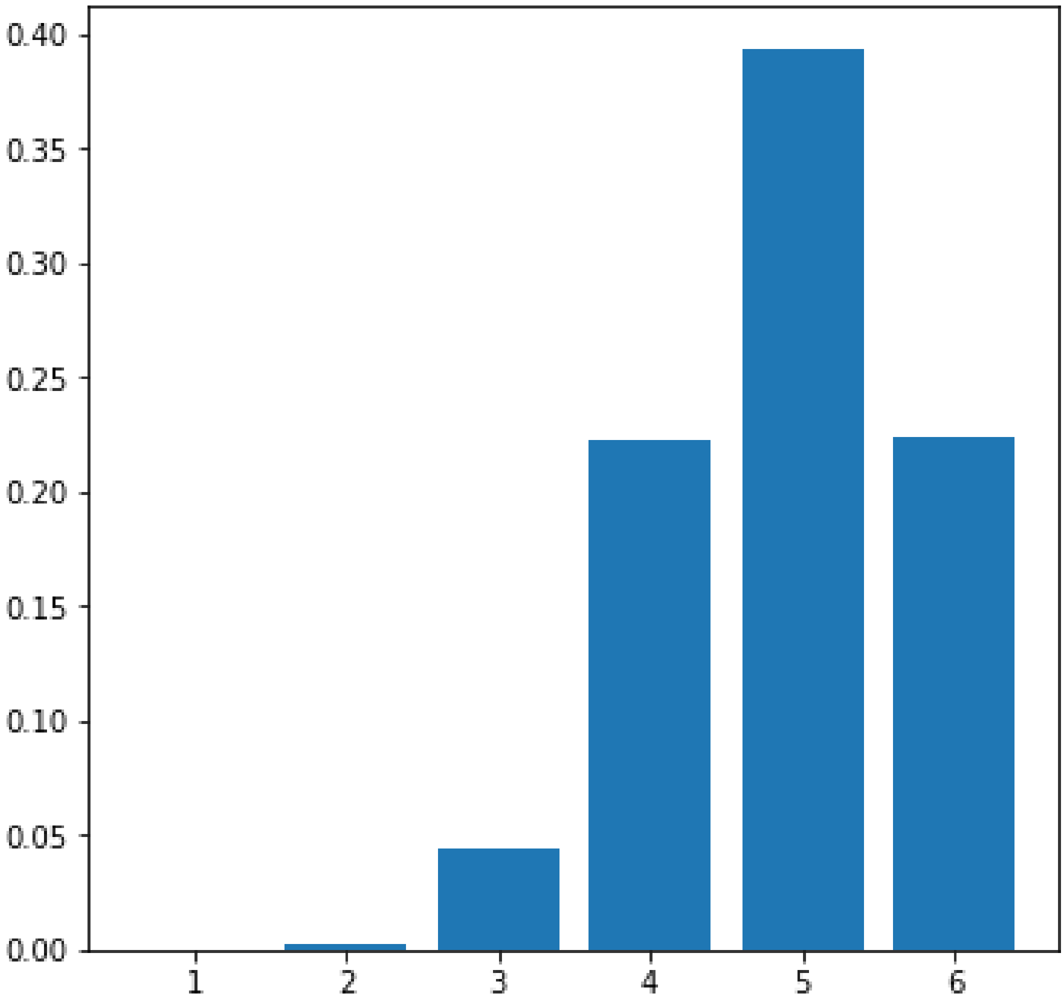
```
import numpy as np
import matplotlib.pyplot as plt

b = np.full(6, fill_value=8, dtype=float)

A = np.zeros((6, 6))
C = np.zeros((6, 6))
for i in range(6):
    for j in range(6):
        C[i, j] = 0.1 * 8 * (i + 1) * (j + 1)
```

```
        A[i, j] = 1 / (C[i,j]**2 + 0.58 * C[i, j])

x = np.linalg.solve(A, b)
cond_value = np.linalg.cond(np.abs(A), p=np.inf)

delta = 0.08
new_x = np.empty((6, 6))
for i in range(6):
    new_b = b.copy()
    new_b[i] += delta
    new_x[i] = np.linalg.solve(A, new_b)

d = []
for i in new_x:
    d.append(np.linalg.norm(x - i, ord=np.inf) / np.linalg.norm(x, ord=np.inf))

plt.figure(figsize=(6, 6))
plt.bar(range(1, 7), d)
plt.show()

new_b = b.copy()
new_b[np.argmax(d)] += delta

delta_b = (np.linalg.norm(new_b - b, ord=np.inf) / np.linalg.norm(b, ord=np.inf))
print(f'The greatest influence on the error has b_{np.argmax(d) + 1}')
print(f'Vector d = {d}')
print(f'The inequality delta(x^m) <= cond(A) * delta(b^m) is fulfilled: {d[np.argmax(c
```

1.4 Результат работы программы

```
The greatest influence on the error has b_5
Vector d = [1.4227340159743037e-05, 0.002458281201822218, 0.044530417947518594, 0.22188869703656877, 0.39304077475900123, 0.22323844180828958]
The inequality delta(x^m) <= cond(A) * delta(b^m) is fulfilled: 0.39304077475900123 <= 45807660729.06723
```

1.5 Гистограмма точности результата



1.6 Вывод

Из полученных данных видно, что наибольшее влияние на погрешность решения оказывает пятая компонента вектора b. Можно также заметить, что теоретическая оценка покрешности в несколько раз больше чем полученная на практике.

2 Погрешности решения системы от погрешностей коэффициентов матрицы

2.1 Формулировка задачи

Для системы уравнений $Ax = b$ из задачи 3.1 исследовать зависимость погрешности решения системы от погрешностей коэффициентов матрицы A (аналогично задаче 3.1). Теоретическая оценка погрешности в этом случае имеет вид: $\delta(x^*) \leq cond(A) \cdot \delta(A^*)$, где x^*- решение системы с возмущенной матрицей A^*.

2.2 Код на Python

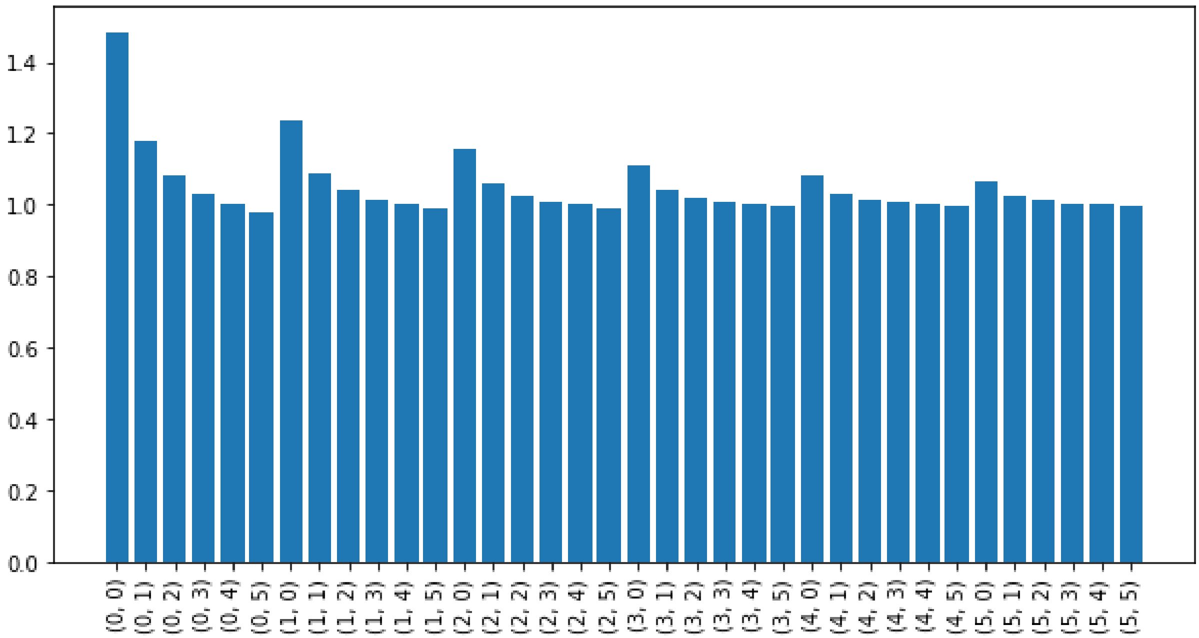
```
import numpy as np
import matplotlib.pyplot as plt
```

```
b = np.full(6, fill_value=8, dtype=float)

A = np.zeros((6, 6))
C = np.zeros((6, 6))
for i in range(6):
    for j in range(6):
        C[i, j] = 0.1 * 8 * (i + 1) * (j + 1)
        A[i, j] = 1 / (C[i,j]**2 + 0.58 * C[i, j])

x = np.linalg.solve(A, b)
cond_value = np.linalg.cond(A, p=np.inf)

delta = 0.08
new_x = {}
for i in range(6):
    for j in range(6):
        new_A = A.copy()
        new_A[i, j] += delta
        new_x[(i, j)] = np.linalg.solve(new_A, b)

d = {key: np.linalg.norm(x - x_i, ord=np.inf) / np.linalg.norm(x, ord=np.inf) for key

plt.figure(figsize=(10, 5))
plt.bar([str(i) for i in d.keys()], d.values())
plt.xticks(rotation=90)
plt.show()

d_i, d_j = max(d, key=d.get)
new_A = A.copy()
new_A[d_i, d_j] += delta

delta_A = (np.linalg.norm(new_A - A, ord=np.inf) / np.linalg.norm(A, ord=np.inf))
print(f'The greatest influence on the error has element on position: {d_i, d_j}')
print(f'The inequality delta(x^*) <= cond(A) * delta(A^*) is fulfilled: {d[(d_i, d_j)]
```

2.3 Результат работы программы

```
The greatest influence on the error has element on position: (0, 0)
The inequality delta(x^*) <= cond(A) * delta(A^*) is fulfilled: 1.4821367200256994 <= 242963747017.2493
```

2.4 Гистограмма точности результата



2.5 Вывод

Так же как в 3.1 получившееся теоретическое значение сильно больше практического. И элемент из первой строки и первого столбца матрицы (в программе будет (0,0), так как массивы индексируются с нуля) больше всего влияет на погрешность (видно из гистограммы).

3 Метод Гаусса

3.1 Формулировка задачи

Дана система уравнений $Az(x) = b(x)$ порядка n. Построить график функции $y(x) = \sum_{i=1}^{n} z_i(x)$ на отрезке $[a, b]$, здесь $z(x) = (z_1(x), ..., z_n(x))^T$ - решение системы. Для решения системы уравнений использовать метод Гаусса (схема полного выбора).

Элементы матрицы А вычисляются по формулам:

$$A_{ij} = \begin{cases} q_M^{i+j} + 0.1 \cdot (j - i), & i \neq j \\ (q_M - 1)^{i+j}, & i = j \end{cases}$$

где $q_M = 1.001 - 2 \cdot M \cdot 10^{-3}$, $i, j = 1, ..., n$

$$[a, b] = [-5, 5]$$

$$N = M = 2$$

$$n = 40$$

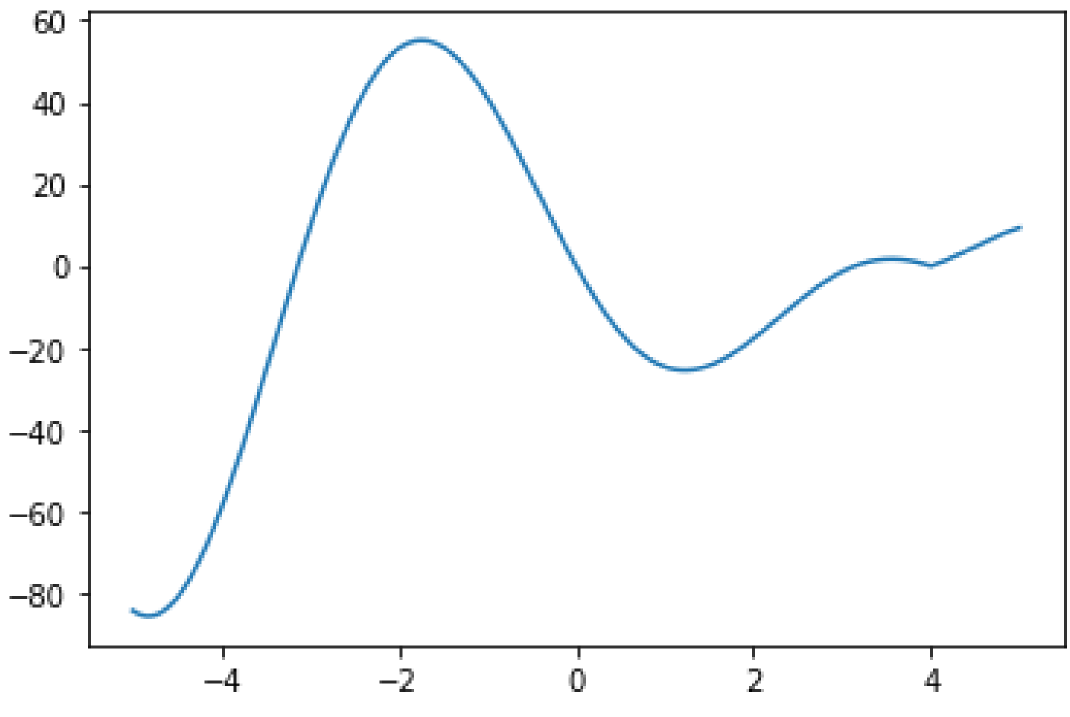
$$b_i = |x - \frac{n}{10}| \cdot i \cdot \sin x, i = 1, .., n$$

3.2 Код на Python

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

def solve(A, b):
    if len(A.shape)!=2:
        raise Exception
    if len(b.shape)!=1:
        raise Exception
    if A.shape[0]!=b.shape[0]:
        raise Exception
    A = A.copy()
    b = b.copy()
    n = A.shape[0]
    b_order = []
    for i in range(n):
        max_value, max_i, max_j = abs(A[i, 0]), i, 0
        for j in range(i, n):
            for k in range(0, n):
                if abs(A[j, k])> max_value:
                    max_value, max_i, max_j = abs(A[j, k]), j, k
        b_order.append(max_j)
        if i!=max_i:
            tmp = A[i].copy()
            A[i] = A[max_i]
            A[max_i] = tmp
            tmp = b[i]
            b[i] = b[max_i]
            b[max_i] = tmp
        for k in range(i + 1, n):
            b[k] -= b[i] * A[k,max_j]/A[i,max_j]
            A[k] -= A[i] * A[k,max_j]/A[i,max_j]
    result = b.copy()
    for i in range(n-1, -1, -1):
        j = b_order[i]
        b[i]/=A[i,j]
        result[j] = b[i]
        for k in range(0, i):
            b[k] -= b[i] * A[k,j]
    return result

def find_b(x):
    b = np.zeros(40)
    for i in range(40):
        b[i] = np.linalg.norm(x - 40/10) * i * np.sin(x)
    return b


x = np.linspace(-5, 5, 1000)
```

```
A = np.zeros((40, 40))
q = 1.001 - 2*2*10**(-3)
for i in range(40):
    for j in range(40):
        if i == j:
            A[i, j] = (q - 1)**(i+j)
        else:
            A[i, j] = q**(i+j) + 0.1*(j - i)

b = np.zeros(40)
y = []
for x_i in x:
    y.append(np.sum(solve(A, find_b(x_i))))

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

3.3 Результат работы программы



3.4 Вывод

Построенный график по равномерно распределенным значениям x показывает, что полученные значения z при суммировании имеют представленное распределение. Немного после значения -2 наблюдается максимальное значение, потом идет спад, а потом решения опять увеличиваются.