

Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота № 2

з дисципліни «Спеціальні розділи математики-2.
Чисельні методи»

на тему

**«Розв’язання систем лінійних алгебраїчних рівнянь
(СЛАР) прямими методами. Звичайний метод Гауса та
метод квадратних коренів»**

Виконав:

студент гр. ІС-34

Колосов Ігор

Викладач:

доц. Рибачук Л.В.

Київ – 2024

Зміст

1. Постановка задачі

Розв'язати систему рівнянь з кількістю значущих цифр $m = 6$.

Якщо матриця системи симетрична, то розв'язання проводити за методом квадратних коренів, якщо матриця системи несиметрична, то використати метод Гауса.

Вивести всі проміжні результати (матриці A , що отримані в ході прямого ходу методу Гауса, матрицю зворотного ходу методу Гауса, або матрицю T та вектор y для методу квадратних коренів) та розв'язок системи.

Навести результат перевірки: вектор нев'язки $r = b - Ax$, де x – отриманий розв'язок.

Розв'язати задану систему рівнянь за допомогою програмного забезпечення Mathcad. Навести результат перевірки: вектор нев'язки $r = b - Ax_m$, де x_m – отриманий у Mathcad розв'язок.

Порівняти корені рівнянь, отримані у Mathcad, із власними результатами за допомогою методу середньоквадратичної похибки:

$$\delta = \sqrt{\frac{1}{n} \sum_{k=1}^n (x_k - x_{mk})^2},$$

де x – отриманий у програмі розв'язок, x_m – отриманий у Mathcad розв'язок.

Зазвичай при використанні для обчислень 4-байтових чисел (тип *float* у Visual C++) порядок δ :

- у методі Гауса - $10^{-4} - 10^{-6}$,
- у методі квадратних коренів - $10^{-5} - 10^{-7}$, бувають і повні співпадіння рішень до 6 знаків після коми.

2. Вихідна система рівнянь

10	$\begin{pmatrix} 1,00 & 0,42 & 0,54 & 0,66 \\ 0,42 & 1,00 & 0,32 & 0,44 \\ 0,54 & 0,32 & 1,00 & 0,22 \\ 0,66 & 0,44 & 0,22 & 1,00 \end{pmatrix}$	$\begin{pmatrix} 0,3 \\ 0,5 \\ 0,7 \\ 0,9 \end{pmatrix}$
----	--	--

3. Проміжні результати

1. matrix A

1.00, 0.42, 0.54, 0.66
0.42, 1.00, 0.32, 0.44
0.54, 0.32, 1.00, 0.22
0.66, 0.44, 0.22, 1.00

2. vector b

0.30, 0.50, 0.70, 0.90

6. matrix U

1.00000, 0.00000, 0.00000, 0.00000
0.42000, 0.90752, 0.00000, 0.00000
0.54000, 0.10270, 0.83538, 0.00000
0.66000, 0.17939, -0.18533, 0.70560

4. Вектор нев'язки

7. vector y

0.30, 0.41, 0.59, 1.05

8. vector x

-1.25779, 0.04349, 1.03917, 1.48239

9. a @ x (should be aprox equal b)

0.30, 0.50, 0.70, 0.90

10. a @ x - b

-0.000000, 0.000000, -0.000000, -0.000000

5. Розв'язок задачі у Mathcad

$$a := \begin{pmatrix} 1.00 & 0.42 & 0.54 & 0.66 \\ 0.42 & 1.0 & 0.32 & 0.44 \\ 0.54 & 0.32 & 1.0 & 0.22 \\ 0.66 & 0.44 & 0.22 & 1.0 \end{pmatrix}$$

$$b := (0.3 \ 0.5 \ 0.7 \ 0.9)$$

$$u := \text{cholesky}(a)$$

$$u = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.42 & 0.908 & 0 & 0 \\ 0.54 & 0.103 & 0.835 & 0 \\ 0.66 & 0.179 & -0.185 & 0.706 \end{pmatrix}$$

$$y := \frac{b}{u^T} = (0.3 \ 0.412 \ 0.593 \ 1.046)$$

$$x := \frac{y}{u} = (-1.258 \ 0.043 \ 1.039 \ 1.482)$$

$$r := b^T - a \cdot x^T = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{Вектор нев'язки}$$

6. Порівняння розв'язку у Mathcad та власного

8. vector x

-1.25779, 0.04349, 1.03917, 1.48239

Програмне рішення

$\text{sol} := (-1.25779 \ 0.04349 \ 1.03917 \ 1.48239)$

Середньоквадратична похибка

$$\text{error} := \sqrt{\frac{1}{4} \cdot \sum_{k=1}^4 [(\text{sol}^T) - x^T]^2} = \begin{pmatrix} 2.808 \times 10^{-11} \\ 1.453 \times 10^{-11} \\ 2.81 \times 10^{-11} \\ 1.663 \times 10^{-11} \end{pmatrix}$$

7. Лістинг програми

```
import numpy as np
from lib_print import ANSI, printer, highlight, pidx
counter = pidx()

vector_b = np.array([0.3, 0.5, 0.7, 0.9]).T
matrix_a = np.array([
    [1.00, 0.42, 0.54, 0.66],
    [0.42, 1.00, 0.32, 0.44],
    [0.54, 0.32, 1.00, 0.22],
    [0.66, 0.44, 0.22, 1.00]
])

a = matrix_a.copy()
b = vector_b.copy()
n = a.shape[0]

u = np.zeros_like(a)

print(printer(
    a, default_style=ANSI.Style.ITALIC,
    text=f"{ANSI.Style.BOLD}{counter.str()}matrix
A{ANSI.Style.RESET}", "\n")
```

```

print(printer(
    b.reshape(-1, 1), default_style=ANSI.Style.ITALIC,
    text=f"{ANSI.Style.BOLD}{counter.str()}vector
b{ANSI.Style.RESET}"), "\n")

h_ij = highlight([], ANSI.FG.GREEN, 0, "case i == j")
h_else = highlight([], ANSI.FG.BLUE, 0, "case else")
h_ki = highlight([], ANSI.FG.YELLOW, 0, "u[k, i] at else case")
h_kj = highlight([], ANSI.FG.CYAN, 0, "u[k, j] at else case")

for i in range(n):
    for j in range(i, n):
        if i == j:
            sum = np.sum([u[k, i]**2 for k in range(i)])
            u[i, i] = np.sqrt(a[i, i] - sum)
            h_ij.indicies.append((i, i))
        else:
            sum = np.sum([u[k, i] * u[k, j] for k in range(i)])
            u[i, j] = (a[i, j] - sum) / u[i, i] # чому тут верхня
трикутна матриця

            for k in range(i):
                h_kj.indicies.append((k, j))
                h_ki.indicies.append((k, i))
            h_else.indicies.append((i, j))
print(printer(a,
    f"{ANSI.Style.BOLD}{counter.str()}matrix A\n -> (values taken
from A at decomposition step) {ANSI.Style.RESET}", [h_ij, h_else],
    print_description=True, default_style=ANSI.FG.BRIGHT_BLACK),
"\n")
print(printer(
    u, formatting="5.5f", highlights=[h_ki],
default_style=ANSI.FG.BRIGHT_BLACK, print_description=True,
    text=f"{ANSI.Style.BOLD}{counter.str()}matrix U (during
decomposition) {ANSI.Style.RESET}"), "\n")
print(printer(
    u, formatting="5.5f", highlights=[h_kj],
default_style=ANSI.FG.BRIGHT_BLACK, print_description=True,
    text=f"{ANSI.Style.BOLD}{counter.str()}matrix U (during

```

```

decomposition){ANSI.Style.RESET}"), "\n")
print(printer(
    u, formatting="5.5f", default_style=ANSI.Style.ITALIC,
    text=f"{ANSI.Style.BOLD}{counter.str()}matrix
U{ANSI.Style.RESET}"), "\n")

# T'y = b
y = np.zeros_like(b)
for i in range(n):
    sum = np.sum([u[k, i] * y[k] for k in range(i)])
    y[i] = (b[i] - sum) / u[i, i]

print(printer(
    y.reshape(-1, 1), default_style=ANSI.FG.GREEN,
    text=f"{ANSI.Style.BOLD}{counter.str()}vector
y{ANSI.Style.RESET}"), "\n")

# Tx = y
x = np.zeros_like(b)
for i in range(n-1, -1, -1):
    sum = np.sum([u[i, k] * x[k] for k in range(i+1, n)])
    x[i] = (y[i] - sum) / u[i, i]

print(printer(
    x.reshape(-1, 1), default_style=ANSI.Style.ITALIC,
    formatting="5.5f",
    text=f"{ANSI.Style.BOLD}{counter.str()}vector
x{ANSI.Style.RESET}"), "\n")
print(printer(
    (a @ x).reshape(-1, 1), default_style=ANSI.Style.ITALIC,
    text=f"{ANSI.Style.BOLD}{counter.str()}a @ x (should be aprox
equal b) {ANSI.Style.RESET}"), "\n")
print(printer(
    (a @ x - b).reshape(-1, 1), default_style=ANSI.Style.ITALIC,
    formatting=".6f",
    text=f"{ANSI.Style.BOLD}{counter.str()}a @ x - b
{ANSI.Style.RESET}"), "\n")
print(printer(

```

```
        a - u.T @ u, default_style=ANSI.Style.ITALIC, formatting=".6f",
        text=f"{ANSI.Style.BOLD}{counter.str()}a - u.T @ u
{ANSI.Style.RESET})", "\n")
```

8. Лістинг програми; Додаток: libprint.py

```
import numpy as np
import dataclasses
from typing import List, Tuple

@dataclasses.dataclass
class highlight:
    indices: List[Tuple[int, int]]
    style: str
    priority: int # 0 heighest
    description: str

def printer(
    array: np.ndarray,
    text: str = "",
    highlights: List[highlight] = [],
    formatting: str = "0.2f",
    separator: str = ", ",
    pre_row_str: str = "  ",
    print_description: bool = False,
    print_text: bool = True,
    default_style: str = "",
    reset_style: str = "\033[0m",
):

    highlights = sorted(highlights, key=lambda h: h.priority,
reverse=True)
    if not highlights:
        highlights = [highlight([], ANSI.Style.RESET, 0, "")]
```



```

        output = [["" for _ in range(array.shape[0])] for _ in
range(array.shape[1])]

        descriptions = ""
        for n, hlght in enumerate(higlhts):
            for j, row in enumerate(array):
                for i, val in enumerate(row):
                    if (i, j) in hlght.indicies:
                        output[i][j] =
f"{f'{hlght.style}{val:{formatting}}'}{reset_style}"
                        elif output[i][j] == "":
                            output[i][j] =
f"{f'{default_style}{val:{formatting}}}{reset_style}'"
                            descriptions += f" * {n: 2d}.
{hlght.style}{hlght.description}{reset_style}\n"
                            mat_str = pre_row_str +
f"\n{pre_row_str}".join(separator.join(map(str, row)) for row in output)

        out = ""
        if print_text:
            out += text + ANSI.Styles.RESET + "\n"
        if print_description:
            out += descriptions + ""
        out += mat_str

        return out

class pidx:
    def __init__(self):
        self.counter = 0

    def get(self):
        self.counter += 1
        return self.counter

    def str(self):
        return f"{self.get(): 2d}. "

```

```
class ANSI:
    class Styles:
        RESET = "\033[0m"
        BOLD = "\033[1m"
        DIM = "\033[2m"
        ITALIC = "\033[3m"
        UNDERLINE = "\033[4m"
        BLINK = "\033[5m"
        REVERSE = "\033[7m"
        HIDDEN = "\033[8m"
        STRIKETHROUGH = "\033[9m"

    class FG:
        BLACK = "\033[30m"
        RED = "\033[31m"
        GREEN = "\033[32m"
        YELLOW = "\033[33m"
        BLUE = "\033[34m"
        MAGENTA = "\033[35m"
        CYAN = "\033[36m"
        WHITE = "\033[37m"
        # Bright colors
        BRIGHT_BLACK = "\033[90m"
        BRIGHT_RED = "\0ANSI.Sty33[91m"
        BRIGHT_GREEN = "\033[92m"
        BRIGHT_YELLOW = "\033[93m"
        BRIGHT_BLUE = "\033[94m"
        BRIGHT_MAGENTA = "\033[95m"
        BRIGHT_CYAN = "\033[96m"
        BRIGHT_WHITE = "\033[97m"

    class BG:
        BLACK = "\033[40m"
        RED = "\033[41m"
        GREEN = "\033[42m"
        YELLOW = "\033[43m"
        BLUE = "\033[44m"
        MAGENTA = "\033[45m"
        CYAN = "\033[46m"
        WHITE = "\033[47m"
```

```
# Bright colors
BRIGHT_BLACK = "\033[100m"
BRIGHT_RED = "\033[101m"
BRIGHT_GREEN = "\033[102m"
BRIGHT_YELLOW = "\033[103m"
BRIGHT_BLUE = "\033[104m"
BRIGHT_MAGENTA = "\033[105m"
BRIGHT_CYAN = "\033[106m"
BRIGHT_WHITE = "\033[107m"

if __name__ == "__main__":
    mat = np.random.random((5, 5))
    h1 = highlight([(0, 0), (4, 4)], ANSI.Styles.BOLD + ANSI.BG.GREEN,
0, "aboba1")
    h2 = highlight([(4, 2), (2, 1)], ANSI.Styles.ITALIC + ANSI.BG.RED,
0, "aboba2")
    mat_str = printer(mat, "cool text phronebius id k ksk kssk sk", [h1,
h2])
    print(mat_str)
```