Звіт до Лабораторної роботи №2
Чиківчука Миколи, ІПС-31
Варіант 11

1. Завдання:

Матрицю X будемо інтерпретувати як двовимірне вхідне зображення, а матрицю Y – як вихідне зображення. Потрібно побудувати лінійний оператор перетворення вхідного сигналу X у вихідний сигнал Y на основі формули (2.9).

1. Вивчити означення псевдооберненої матриці і її основні властивості.

2. Створити програму, яка за заданими двома зображеннями знаходить лінійний оператор переходу між цими зображеннями. Основою для програми є формула (2.9), де V – довільна матриця (наприклад, нульова). Псевдообернену матрицю в (2.9) шукати двома методами: на основі формули Мура-Пенроуза (див. (2.3) або (2.4)) і на основі формули Гревіля. Правильність знаходження псавдооберненої матриці перевірити за допомогою теореми 2.1 про характеристичну властивість псевдооберненої матриці.

3. Вивести вихідне зображення і образ вхідного зображення при одержаному перетворенні. Зробити порівняння. Проаналізувати одержаний результат.

2. Розв’язок:

```
import numpy as np

from matplotlib import image
import matplotlib.pyplot as plt


def greville(M):
    M = np.array(M, dtype=float)
    ai = M[0:1]
    if np.count_nonzero(ai[0]) == 0:
        res = np.zeros_like(ai.T)
    else:
        res = ai.T/(ai @ ai.T)

    n = M.shape[0]
    for i in range(1, n):
        z_a = np.eye(res.shape[0]) - (res @ M[:i])
        r_a = res @ res.T
        ai = M[i:i+1]

        condition = (ai @ z_a) @ ai.T
```

```
            if np.count_nonzero(condition) != 1:
                a_inv = (r_a @ ai.T) / (1 + (ai @ r_a) @ ai.T)
            else:
                a_inv = (z_a @ ai.T) / condition

            res -= a_inv @ (ai @ res)
            res = np.concatenate((res, a_inv), axis=1)
        return res


def moore_penrose(M, sigma0, eps=1e-5):
        M = np.array(M, dtype=float)
        e = np.eye(M.shape[0])
        sigma_k = sigma0

        plus_matrix = M.T @ np.linalg.inv(M @ M.T + sigma0 * e)
        while True:
            sigma_k = sigma_k / 2
            previous = plus_matrix
            plus_matrix = M.T @ np.linalg.inv(M @ M.T + sigma_k * e)
            if np.linalg.norm(plus_matrix - previous) < eps:
                return plus_matrix



X, Y = image.imread('x2.bmp'), image.imread('y2.bmp')

MP = moore_penrose(X, 1)
G = greville(X)
A_moore = Y @ MP
A_greville = Y @ G
```

3. Результат:

x2

y2

greville

moore penrose