

Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем
Моделювання складних систем

Звіт
з лабораторної роботи №2
студентки 3-го курсу
групи ІС-31
Ляшенка Матвія Олексійовича
Варіант №14

Київ
2025

Хід роботи

1. Завантаження вхідного та вихідного зображень:

Завантажено зображення **x3.bmp** (вхідний сигнал) та **y8.bmp** (вихідний сигнал), які були оброблені у вигляді числових матриць X та Y відповідно. Проведено попереднє опрацювання, включаючи нормалізацію пікселів зображень до діапазону [0, 1].

2. Формування розширеної матриці:

До матриці X додано рядок одиниць для врахування вільного члена у моделі лінійного оператора. Таким чином, отримано розширену матрицю (\tilde{X}), яка має розмірність $(m+1) \times n$, де m — кількість рядків у матриці (X), а (n) — кількість стовпців.

3. Обчислення псевдооберненої матриці двома методами:

Для кожного з методів обчислено псевдо обернену матрицю X^+ :

- **Метод Мура-Пенроуза** (ітераційний підхід на основі граничного переходу).
- **Метод Гревіля** (побудова псевдооберненої матриці шляхом поступового додавання рядків).

4. Для кожного методу перевірено виконання чотирьох умов Мура-Пенроуза:

- $A A^+ A \approx A$
- $A^+ A A^+ \approx A^+$
- $A A^+$ симетрична
- $A^+ A$ симетрична

5. Побудова лінійного оператора (A):

Лінійний оператор побудовано за формулою:

$$A = Y \cdot X^+$$

де X^+ — псевдообернена матриця, отримана відповідним методом, а Y — матриця вихідних сигналів.

6. Обчислення відновленого вихідного сигналу:

Для кожного методу отримано апроксимацію вихідного сигналу:

$$\hat{Y} = A \cdot \tilde{X}$$

Це значення є відновленим вихідним сигналом, який обчислюється на основі отриманого лінійного оператора A та розширеної матриці \tilde{X} .

7. Оцінювання точності та порівняння методів:

Для результатів обох методів обчислено:

- **MSE (середньоквадратичну помилку)** — середнє значення квадратів різниць між фактичними і відновленими значеннями.
- **RMSE (корінь середньоквадратичної помилки)** — корінь квадратний з MSE.
- **Час виконання** кожного методу.

8. На основі цих метрик проведено порівняльний аналіз точності та ефективності методів:

- Визначено, який з методів дає найменшу помилку та найбільш ефективний за часом виконання.

Програмна реалізація

Програму реалізовано мовою **Python** та розділено на три логічні модулі:

- **main.py** — основний керуючий файл;
- **image_io.py** — модуль роботи із зображеннями;
- **pseudoinverse.py** — модуль обчислення псевдообернених матриць двома методами (Мура–Пенроуза та Гревіля).

Для реалізації використано бібліотеки:

- **NumPy** — для лінійної алгебри та обчислень з матрицями;
- **Pillow** — для зчитування та запису зображень у форматі BMP;
- **Matplotlib** — для побудови графіків та візуалізації результатів;
- Стандартні модулі **time** та **os** — для вимірювання часу виконання та роботи з файлами.

1. Модуль **image_io.py**

У модулі **image_io.py** реалізовано дві функції:

- **read_grayscale_image(path)** — читує BMP-файл, переводить його у формат **grayscale** та перетворює на матрицю типу **float** з нормалізацією пікселів у діапазон $[0,1]$. Саме ці матриці використовуються як матриці **X** та **Y** у задачі ($AX = Y$).
- **save_grayscale_image(path, matrix)** — перетворює отриману матрицю назад у 8-бітне зображення (обрізаючи значення до $[0,1]$) і зберігає результат на диск. Це дозволяє візуально порівняти вихідні та відновлені зображення.

2. Модуль **pseudoinverse.py**

Модуль **pseudoinverse.py** містить реалізацію двох методів для обчислення псевдообернених матриць:

- **pseudo_inverse_moore_penrose(A)** — ітераційно обчислює псевдообернену матрицю за формулою Мура–Пенроуза. На кожній ітерації будеться матриця, параметр δ поступово зменшується, а цикл зупиняється при досягненні заданої точності ϵ .
- **pseudo_inverse_greville(A)** — реалізує метод Гревіля, який формує псевдообернену матрицю послідовно, додаючи рядки матриці **A** та оновлюючи A^+ за рекурсивними формулами.
- **is_pseudoinverse(A, A_plus)** — перевіряє виконання чотирьох умов Мура–Пенроуза для отриманої матриці A^+ , що дозволяє переконатися в коректності реалізації алгоритмів.

3. Основний файл **main.py**

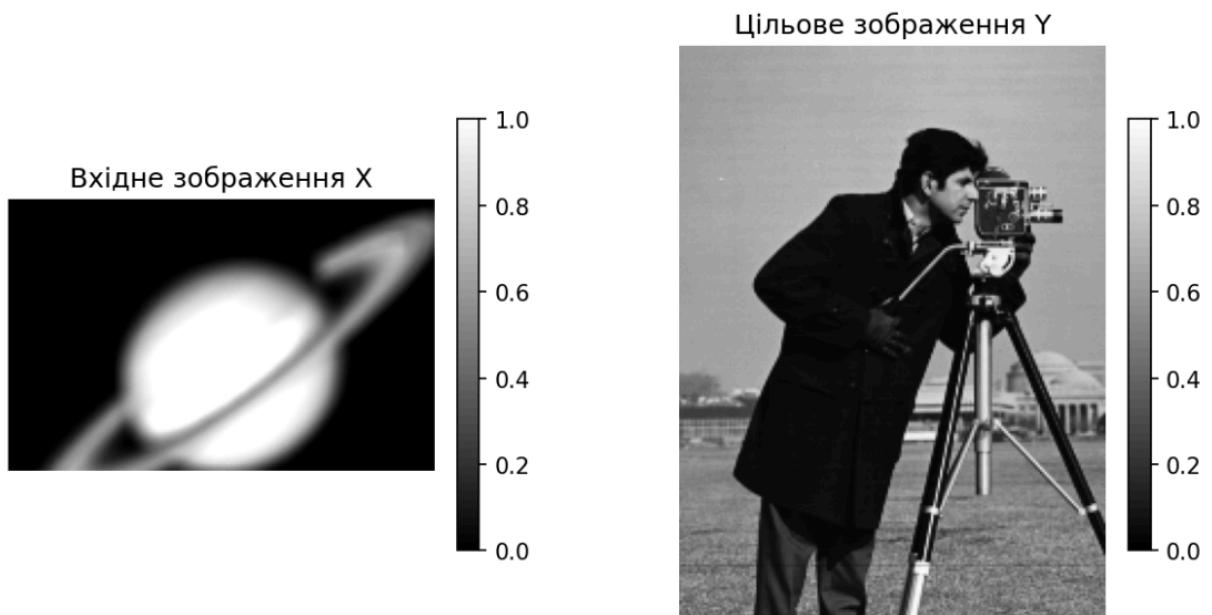
У файлі **main.py** реалізується основний процес:

1. Зчитування вхідного зображення (x3.bmp) та відповідного вихідного (y8.bmp). Перевіряється узгодженість розмірностей.
2. Формування розширеної матриці \tilde{X} за допомогою функції `add_bias_row`.
3. Функція `build_operator_and_predict()` для кожного з методів:
 - Обчислює псевдо обернену матрицю X^+ .
 - Перевіряє умови Мура-Пенроуза.
 - Будує оператор $A = Y \cdot X^+$.
 - Відновлює вихідне зображення.
 - Обчислює метрики похибки: MSE та RMSE.
 - Фіксує час виконання кожного методу.
4. Отримані результати для методів Мура-Пенроуза та Гревіля зберігаються у вигляді зображень `result_moore_penrose.bmp` та `result_greville.bmp`, що дає можливість якісно оцінити відновлення.

Підсумок:

Програмна реалізація виконує всі необхідні кроки, щоб відновити вихідне зображення з допомогою лінійного оператора та псевдообернених матриць, використовуючи два різні методи. Код порівнює методи за метриками точності та часу виконання, а також забезпечує візуалізацію результатів.

Результат:



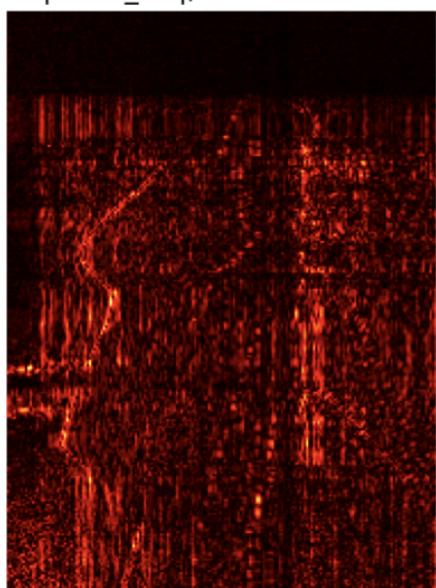
Мур-Пенроуз
RMSE = 0.0417



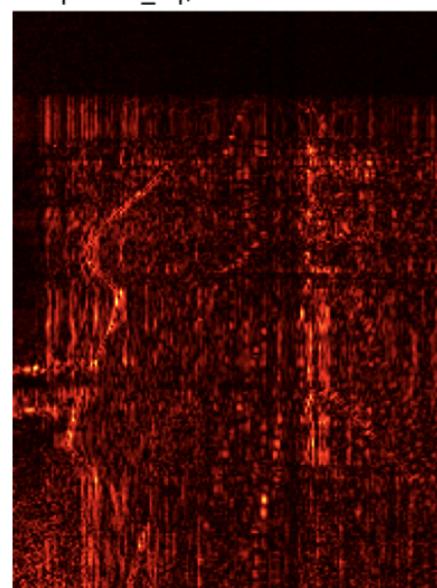
Гревіль
RMSE = 0.0417



$|Y - \hat{Y}_{MP}|$, max = 0.3193



$|Y - \hat{Y}_G|$, max = 0.3193



Метод	Час виконання (с)	RMSE
Мур-Пенроуз	0.002199	0.042468
Гревіль	0.000967	0.042468

Згідно з отриманими результатами, обидва методи відновили зображення з майже однаковою точністю ($RMSE \approx 0.042468$), що вказує на незначну похибку і високу точність наближення до оригіналу.

Однак суттєва різниця спостерігається в часі виконання: метод Гревіля продемонстрував значно більшу швидкість, працюючи майже вдвічі швидше за метод Мура-Пенроуза. Це зумовлено різними підходами в алгоритмах: метод Мура-Пенроуза потребує обчислення інверсій матриць, що є ресурсозатратним, в той час як метод Гревіля поступово оновлює псевдообернену матрицю, що робить його більш ефективним при додаванні нових даних. Таким чином, з огляду на однакову точність, метод Гревіля є **більш вигідним** для практичних застосувань.