Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

**Лабораторна робота №2** з предмету «Моделювання систем»

Виконала студентка 3-го курсу Групи ІПС-31

Гаврада Дарина Ігорівна

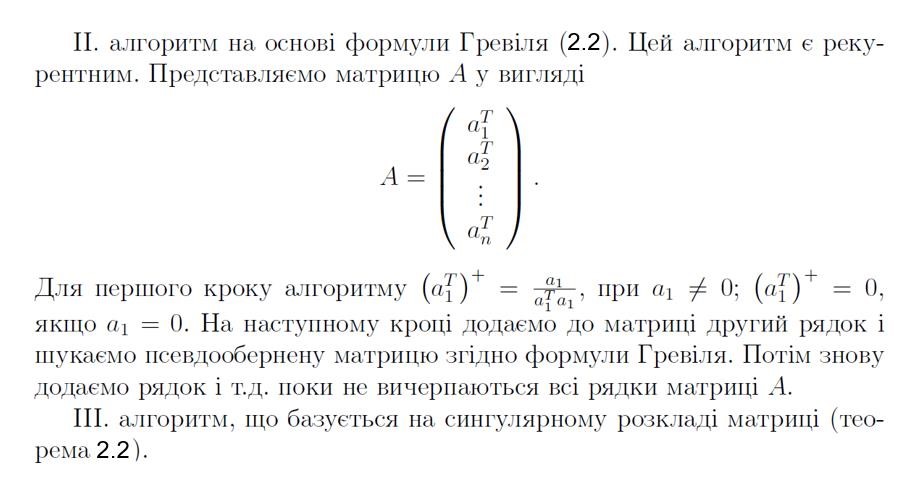
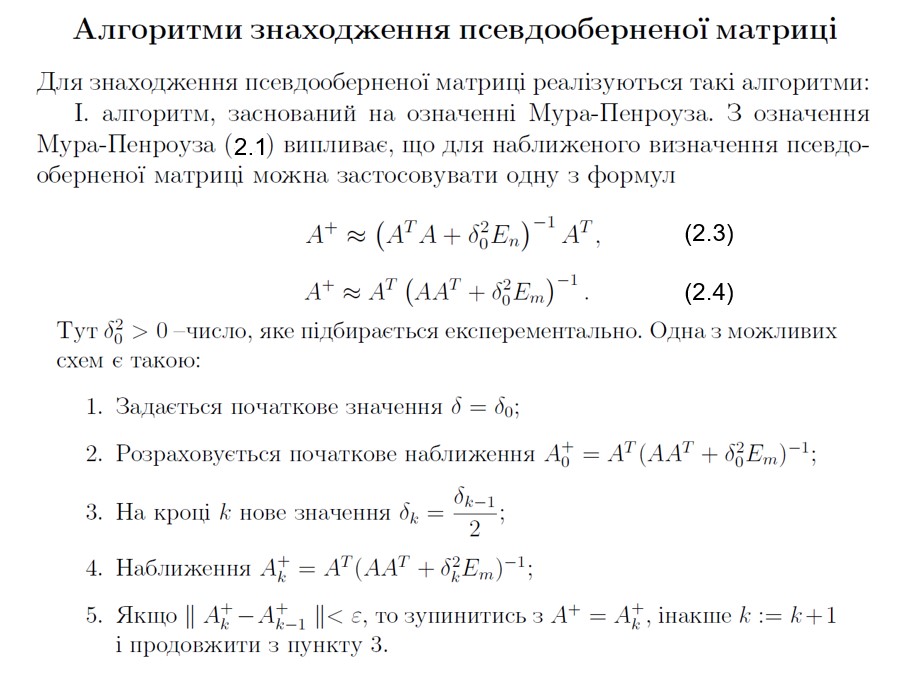
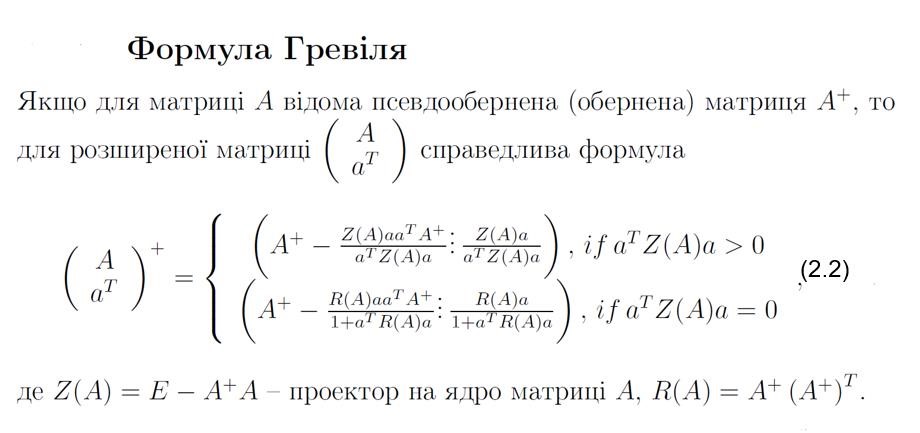
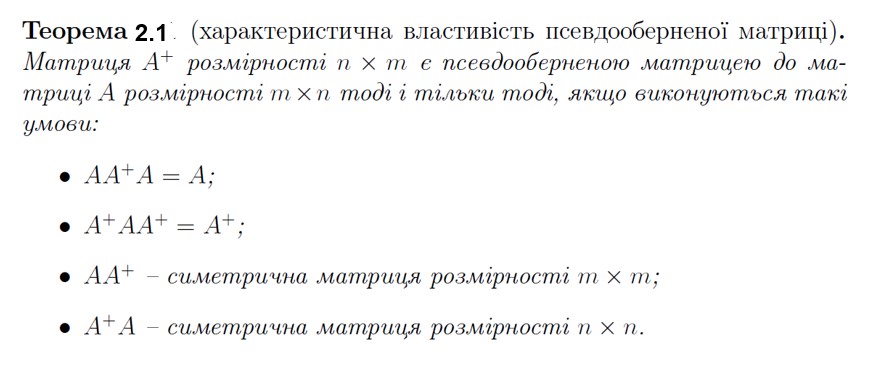
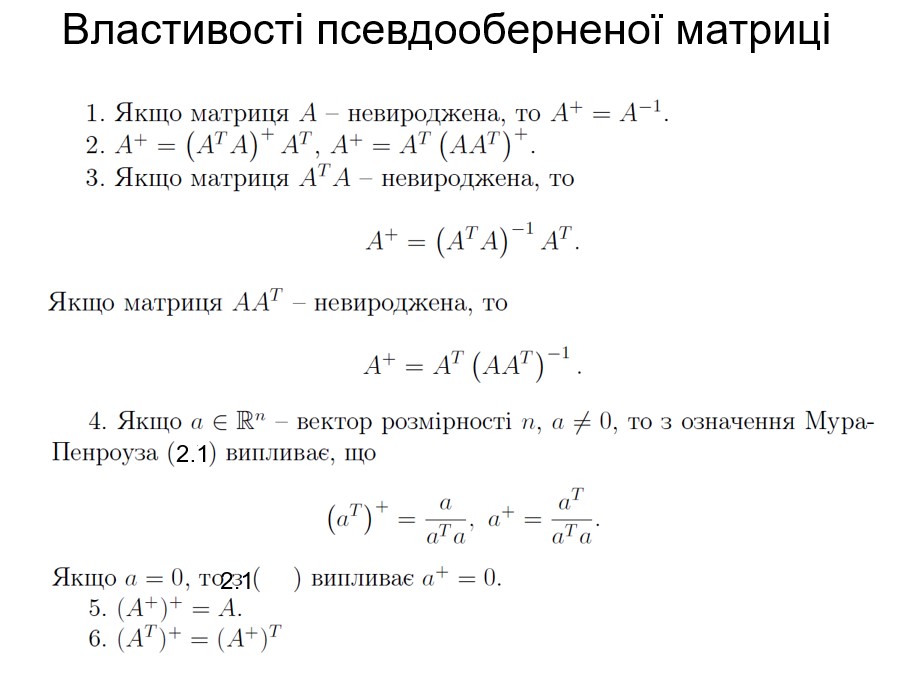
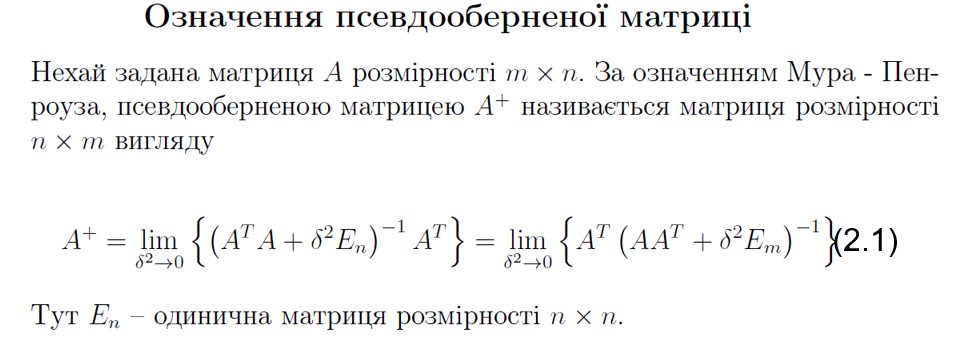
2023, Київ

# Завдання – Варіант 7 (x1.bmp та y7.bmp)

Text

Description automatically generated

# Теорія

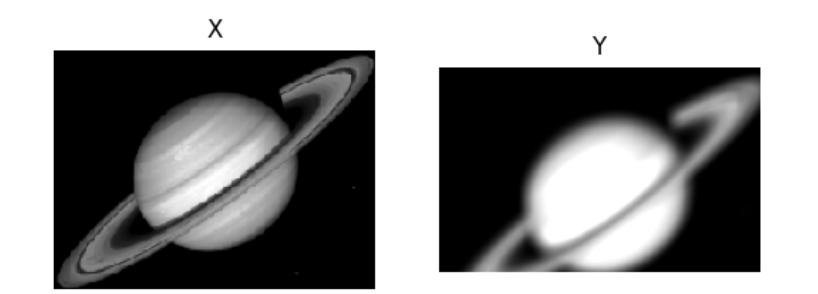


# Розв’язок

## Зчитування зображення з файлів

x\_image, y\_image = image.imread('x1.bmp'), image.imread('y7.bmp')

## Вивід даних зображень на екран



## Реалізація методу Мура-Пенроуза

def moore\_penrose\_method(matrix, sigma0, eps=1e-5):  
 # Step 1 (sigma0 init by user)  
  
 matrix = np.array(matrix, dtype=float)  
 e = np.eye(matrix.shape[0])  
 sigma\_k = sigma0  
  
 # Step 2  
 plus\_matrix = matrix.T @ np.linalg.inv(matrix @ matrix.T + sigma0 \* e)  
  
 while True:  
 # Step 3  
 sigma\_k = sigma\_k / 2  
  
 previous = plus\_matrix  
  
 # Step 4  
 plus\_matrix = matrix.T @ np.linalg.inv(matrix @ matrix.T + sigma\_k \* e)  
  
 # Step 5  
 if np.linalg.norm(plus\_matrix - previous) < eps:  
 return plus\_matrix

## Реалізація методу на основі формули Гревіля

def greville\_method(matrix):  
 matrix = np.array(matrix, dtype=float)  
  
 # Get first row  
 a = matrix[0:1]  
  
 if np.count\_nonzero(a[0]) == 0:  
 result = np.zeros\_like(a.T)  
 else:  
 result = a.T / a @ a.T  
  
 # Greville formula  
 for i in range(1, matrix.shape[0]):  
 z\_a = np.eye(result.shape[0]) - result @ matrix[:i]  
 r\_a = result @ result.T  
 a = matrix[i:i + 1]  
  
 dot\_product = (a @ z\_a) @ a.T  
  
 if np.count\_nonzero(dot\_product) == 0:  
 part\_a = (r\_a @ a.T) / (1 + (a @ r\_a) @ a.T)  
 else:  
 part\_a = (z\_a @ a.T) / dot\_product  
  
 result = np.concatenate((result - part\_a @ (a @ result), part\_a), axis=1)  
  
 return result

## Реалізація Теореми 2.1 (характеристична властивість матриці)

def pseudoinverse\_matrix\_check(x\_plus, x):  
 result = True  
  
 result = result and ((x @ x\_plus) @ x).all() == x.all()  
 result = result and ((x\_plus @ x) @ x\_plus).all() == x\_plus.all()  
 result = result and np.allclose(x @ x\_plus, (x @ x\_plus).T)  
 result = result and np.allclose(x\_plus @ x, (x\_plus @ x).T)  
  
 return result

Отримано матрицю за допомогою метода Мура-Пенроуза

moore\_penrose\_matrix = moore\_penrose\_method(x\_image, 1)

## Отримано матрицю за допомогою метода на основі формули Гревіля

greville\_matrix = greville\_method(x\_image)

Перевірка отриманої матриці moore\_penrose\_matrix за допомогою метода pseudoinverse\_matrix\_check

moore\_penrose\_status = 'Ok' if pseudoinverse\_matrix\_check(moore\_penrose\_matrix, x\_image) else 'Bad'  
print(f'Status Moore-Penrose method: {moore\_penrose\_status}')

Перевірка отримаої матриці greville\_matrix за допомогою метода pseudoinverse\_matrix\_check

greville\_status = 'Ok' if pseudoinverse\_matrix\_check(greville\_matrix, x\_image) else 'Bad'  
print(f'Status Greville method: {greville\_status}')

### Отриманий результат у консолі

Status Moore-Penrose method: Ok

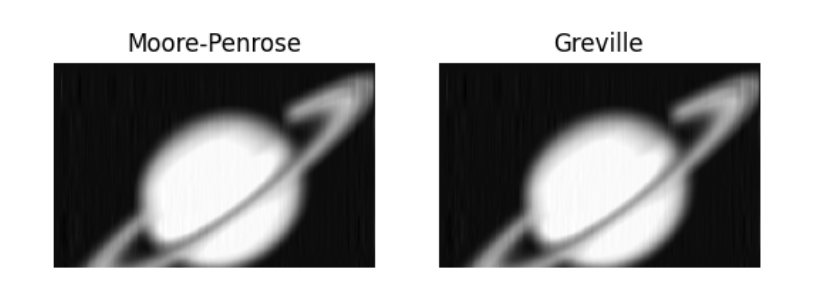
Status Greville method: Ok

### Пошук операторів відповідних матриць

moore\_penrose\_operator = y\_image @ moore\_penrose\_matrix  
greville\_operator = y\_image @ greville\_matrix

### Вивід отриманого вихідного зображення за допомогою операторів

ax = fig.add\_subplot(2, 2, 3)  
ax.set\_title('Moore-Penrose')  
plt.imshow(moore\_penrose\_operator @ x\_image, cmap='gray')  
plt.axis('off')  
  
ax = fig.add\_subplot(2, 2, 4)  
ax.set\_title('Greville')  
plt.imshow(greville\_operator @ x\_image, cmap='gray')  
plt.axis('off')



# Висновок

Як бачимо, дане зображення та зображень, які отримані за допомогою методу Мура-Пенроуза та методу

Гревіля майже збігаються. Це свідчить про правильність отриманих операторів.