Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Моделювання систем

Лабораторна робота №2

Виконав студент 3-го курсу

Групи ІПС-31

Аніканов Олександр Олександрович

2022

Зміст

Завдання 3

Вхідні дані 4

Теорія 5

Програмний розв`язок 9

Вихідні дані 12

Висновок 13

Завдання

,

Матрицю *X* будемо iнтерпретувати як двовимiрне вхiдне зображення,  
а матрицю *Y* – як вихiдне зображення. Потрiбно побудувати лiнiйний  
оператор перетворення вхiдного сигналу *X* у вихiдний сигнал *Y* на основi  
формули (3.9).

1. Вивчити означення псевдооберненої матрицi i її основнi властивостi.

2. Створити програму, яка за заданими двома зображеннями знаходить лiнiйний оператор переходу мiж цими зображеннями. Основою для  
програми є формула (3.9), де *V* – довiльна матриця (наприклад, нульова). Псевдообернену матрицю в (3.9) шукати двома методами: на основi  
формули Мура-Пенроуза (див. (3.3) або (3.4)) i на основi формули Гревiля. Правильнiсть знаходження псавдооберненої матрицi перевiрити за  
допомогою теореми 3.1 про характеристичну властивiсть псевдооберненої матрицi.

3. Вивести вихiдне зображення i образ вхiдного зображення при одержаному перетвореннi. Зробити порiвняння. Проаналiзувати одержаний  
результат.

4. Оформити в друкованiй формi звiт про виконання роботи, в якому  
викласти результати проведених обчислень.

Вхідні дані  


Теорія

Псевдооберненою називається узагальнення оберненої матриці в лінійній алгебрі. називається псевдооберненою до матриці А, якщо вона задовольняє такі умови:

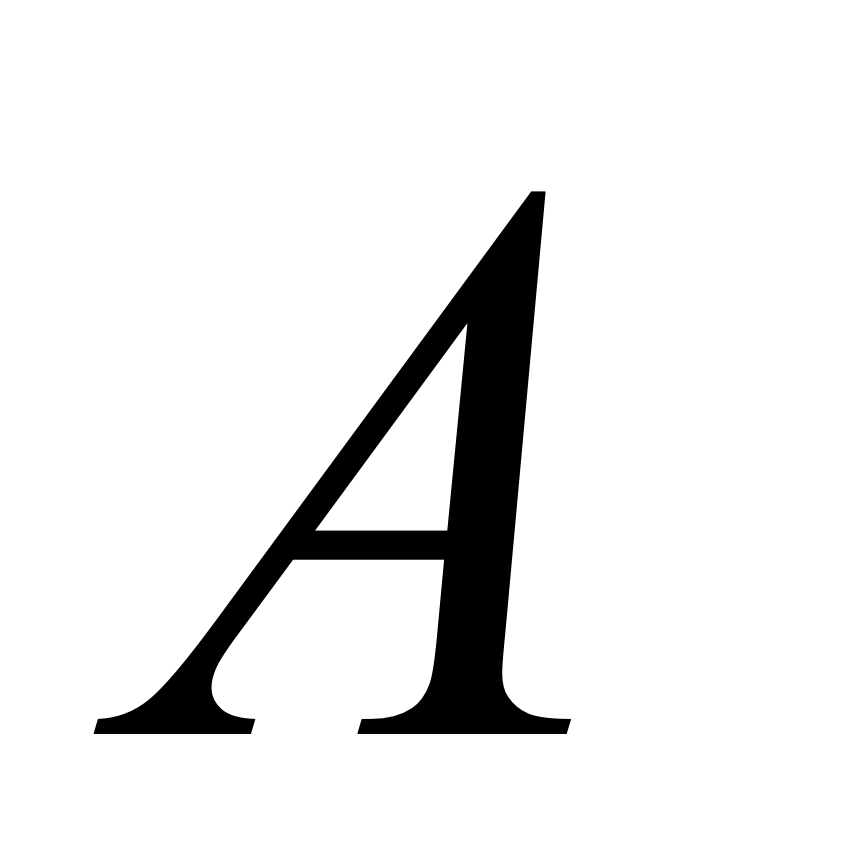
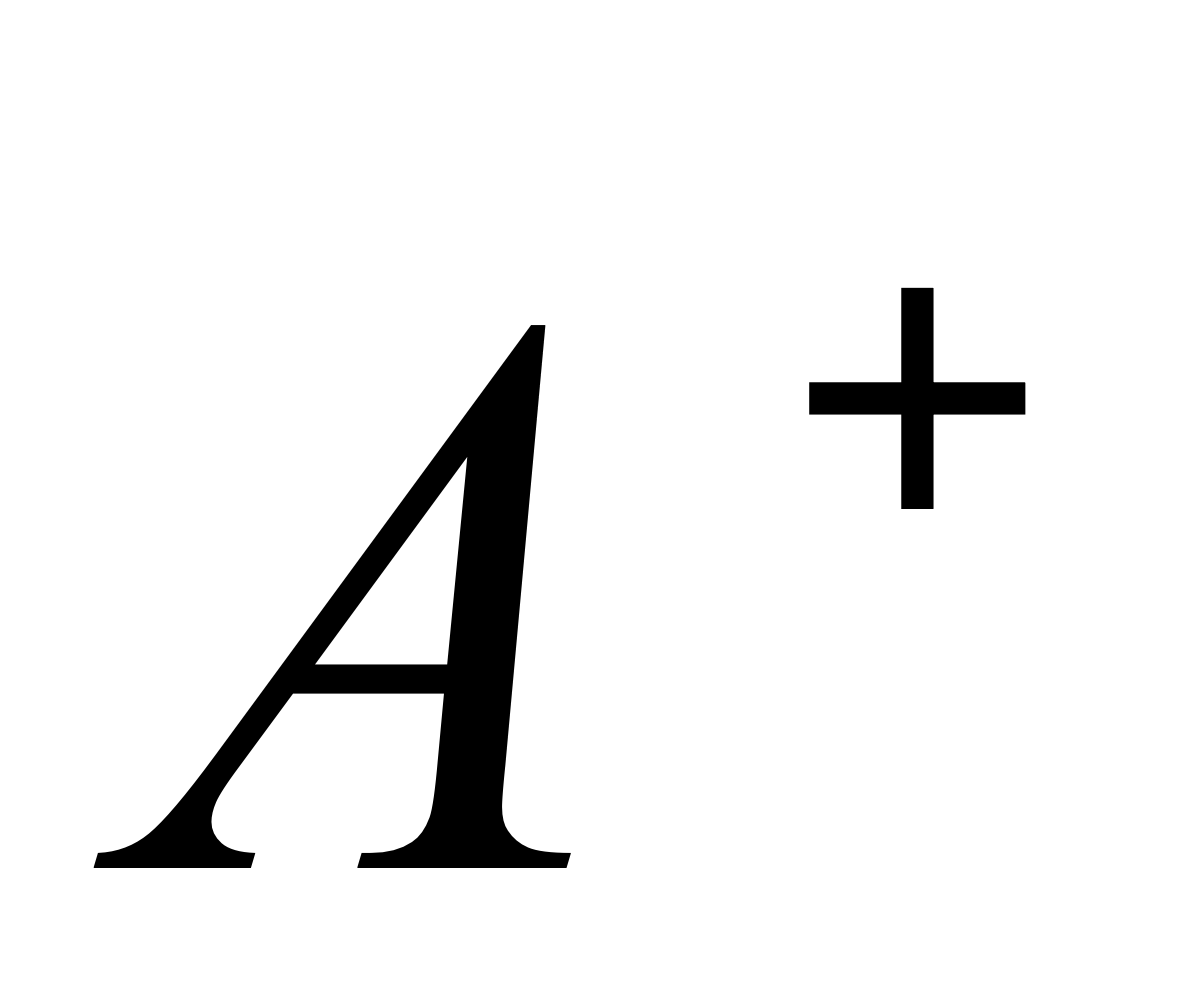
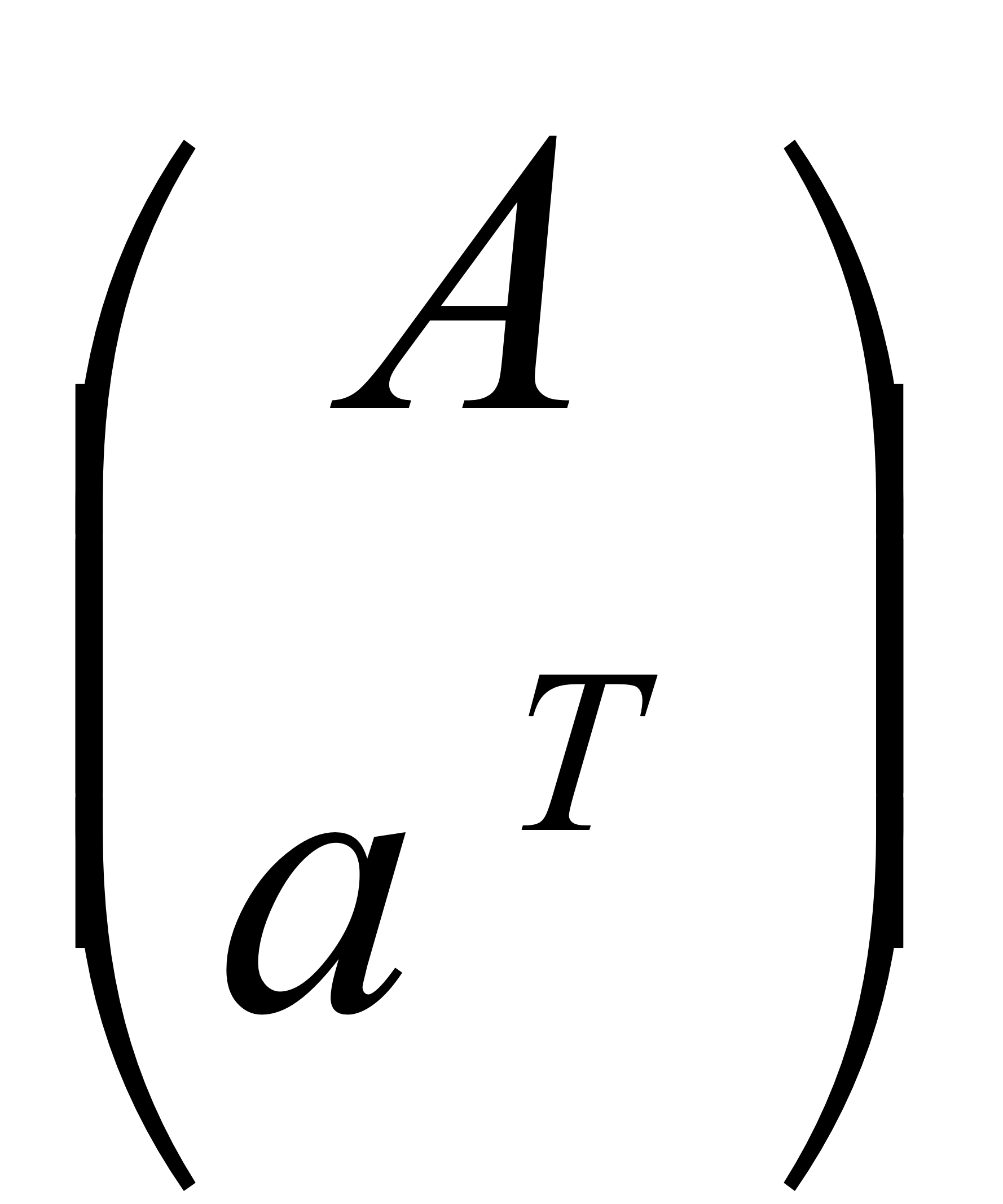
1. (чи не обов’язково дорівнюватимуть одиничній матриці)
2. ( – також ермітова матриця)

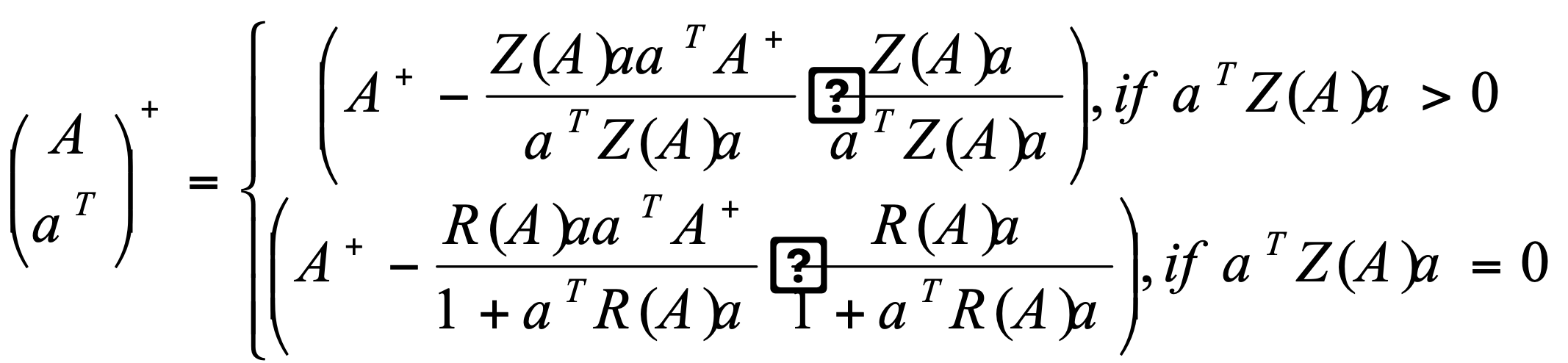
Де ермітово-спряжена матриця до матриці А.

Властивості:

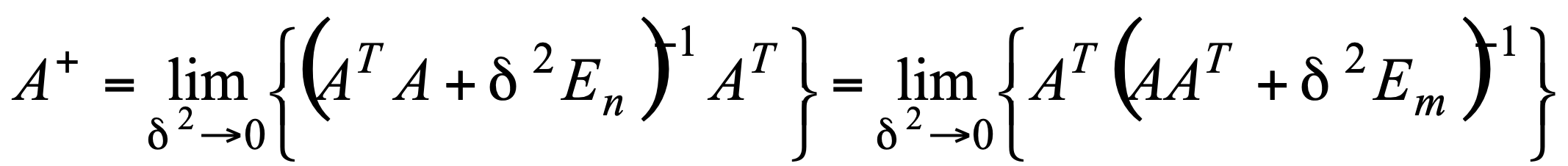
1. Псевдообернена матриця існує і вона єдина.
2. Псевдообернення нульової матриці дорівнює її транспонуванню
3. Псевдообернення є оборотним до самого себе
4. Псевдообернення комутує з транспонуванням, спряженням і ермітовим спряженням: =
5. Ранг матриці дорівнює рангу її псевдооберненої
6. Псевдообернення добутку матриці на скаляр a дорівнює добутку матриці на обернене число .
7. Якщо вже відома матриця чи матриця
8. Якщо матриця утворена за матриці за допомогою вставки ще одного нульового рядка/стовпця в і-ту позицію, то буде утворюватись з додаванням нульового стовпця/рядка в і-ту позицію.
9. Якщо рядок/стовпець в попередній процедурі не є нульовим то існує формула Гревіля для вираження через

Формула Гревіля

Якщо для матриці  відома псевдообернена (обернена) матриця , то для розширеної матриці  справедлива формула



Визначення Мура-Пенроуза



Код

X = imread("x1.bmp");

X = double(X);

X = [X; ones(size(X, 2), 1)'];

imshow(uint8(X));

figure

Y = imread("y1.bmp");

Y = double(Y);

imshow(uint8(Y));

PseudoX = Grevil(X);

A = Y \* PseudoX + rand(size(Y,1), size(X,1)) \* Z(PseudoX, X);

isCorrect = A \* X;

figure

imshow(uint8(isCorrect));

PseudoX = MurPenroze(X);

A = Y \* PseudoX + rand(size(Y,1), size(X,1)) \* Z(PseudoX, X);

isCorrect = A \* X;

figure

imshow(uint8(isCorrect));

function valReturn = Z(A, PseudoA)

valReturn = eye(size(PseudoA, 1)) - PseudoA \* A;

end

function valReturn = Grevil(A)

isSwap = 0;

if(size(A,1) > size(A,2))

isSwap = 1;

A = A';

end

currVector = A(1,:)';

PseudoA = 0;

currVectorScalar = currVector' \* currVector;

if(currVectorScalar == 0)

PseudoA = currVector;

else

PseudoA = currVector/currVectorScalar;

end

iterA = currVector';

for i = 2: size(A,1)

currVector = A(i,:)';

ZA = Z(iterA,PseudoA);

iterA = [iterA;currVector'];

denomZ = currVector' \* ZA \* currVector;

if(denomZ > 0.000001)

PseudoA = [(PseudoA-(ZA\*currVector\*currVector'\*PseudoA)/denomZ),(ZA\*currVector)/denomZ];

else

RA = PseudoA \* PseudoA';

denomR = 1 + currVector' \* RA \* currVector;

PseudoA = [(PseudoA - (RA \* currVector \* currVector' \* PseudoA)/denomR),(RA \* currVector)/denomR];

end

end

if (isSwap)

PseudoA = PseudoA';

end

valReturn = PseudoA;

end

function valReturn = J(currA, nextA)

valReturn = max(max((currA-nextA).^2));

end

function valueReturn = MurPenroze(A)

isSwap = 0;

if(size(A,1) > size(A,2))

isSwap = 1;

A = A';

end

epsilon = 1e-8;

inf = 1e9;

currPseudoA = inf \* ones(size(A))';

nextPseudoA = -inf \* ones(size(A))';

delta = 10.0;

counter = 0;

while(J(currPseudoA, nextPseudoA) > epsilon)

currPseudoA = nextPseudoA;

nextPseudoA = A' \* inv(A \* A' + (delta) \* eye(size(A, 1)));

delta = delta/2.0;

counter = counter + 1;

fprintf("%d\n", counter);

end

if(isSwap)

nextPseudoA = nextPseudoA';

end

valueReturn = nextPseudoA;

end

Програмний розв`язок

1) Знаходимо Z функцію:

X = imread("x1.bmp");

X = double(X);

X = [X; ones(size(X, 2), 1)'];

imshow(uint8(X));

figure

Y = imread(«y1.bmp");

Y = double(Y);

imshow(uint8(Y));

PseudoX = Grevil(X);

A = Y \* PseudoX + rand(size(Y,1), size(X,1)) \* Z(PseudoX, X);

isCorrect = A \* X;

figure

imshow(uint8(isCorrect));

PseudoX = MurPenroze(X);

A = Y \* PseudoX + rand(size(Y,1), size(X,1)) \* Z(PseudoX, X);

isCorrect = A \* X;

figure

imshow(uint8(isCorrect));

function valReturn = Z(A, PseudoA)

valReturn = eye(size(PseudoA, 1)) - PseudoA \* A;

end

2) Знаходимо псевдообернену матриці використовуючи формулу Гревіля:

function valReturn = Grevil(A)

isSwap = 0;

if(size(A,1) > size(A,2))

isSwap = 1;

A = A';

end

currVector = A(1,:)';

PseudoA = 0;

currVectorScalar = currVector' \* currVector;

if(currVectorScalar == 0)

PseudoA = currVector;

else

PseudoA = currVector/currVectorScalar;

end

iterA = currVector';

for i = 2: size(A,1)

currVector = A(i,:)';

ZA = Z(iterA,PseudoA);

iterA = [iterA;currVector'];

denomZ = currVector' \* ZA \* currVector;

if(denomZ > 0.000001)

PseudoA = [(PseudoA-(ZA\*currVector\*currVector'\*PseudoA)/denomZ),(ZA\*currVector)/denomZ];

else

RA = PseudoA \* PseudoA';

denomR = 1 + currVector' \* RA \* currVector;

PseudoA = [(PseudoA - (RA \* currVector \* currVector' \* PseudoA)/denomR),(RA \* currVector)/denomR];

end

end

if (isSwap)

PseudoA = PseudoA';

end

valReturn = PseudoA;

end

3) Знаходимо J функцію

function valReturn = J(currA, nextA)

valReturn = max(max((currA-nextA).^2));

end

4) Знаходимо псевдообернену матрицю використовуючи формулу Мура-Пенроуза:

function valueReturn = MurPenroze(A)

isSwap = 0;

if(size(A,1) > size(A,2))

isSwap = 1;

A = A';

end

epsilon = 1e-8;

inf = 1e9;

currPseudoA = inf \* ones(size(A))';

nextPseudoA = -inf \* ones(size(A))';

delta = 10.0;

counter = 0;

while(J(currPseudoA, nextPseudoA) > epsilon)

currPseudoA = nextPseudoA;

nextPseudoA = A' \* inv(A \* A' + (delta) \* eye(size(A, 1)));

delta = delta/2.0;

counter = counter + 1;

fprintf("%d\n", counter);

end

if(isSwap)

nextPseudoA = nextPseudoA';

end

valueReturn = nextPseudoA;

end

5) Маючи вхідні дані знаходимо лінійний оператор переходу вхідного сигналу у вихідний для Гревіля та Мура-Пенроуза:

X = imread("x1.bmp");

X = double(X);

X = [X; ones(size(X, 2), 1)'];

imshow(uint8(X));

figure

Y = imread("y4.bmp");

Y = double(Y);

imshow(uint8(Y));

PseudoX = Grevil(X);

A = Y \* PseudoX + rand(size(Y,1), size(X,1)) \* Z(PseudoX, X);

isCorrect = A \* X;

figure

imshow(uint8(isCorrect));

PseudoX = MurPenroze(X);

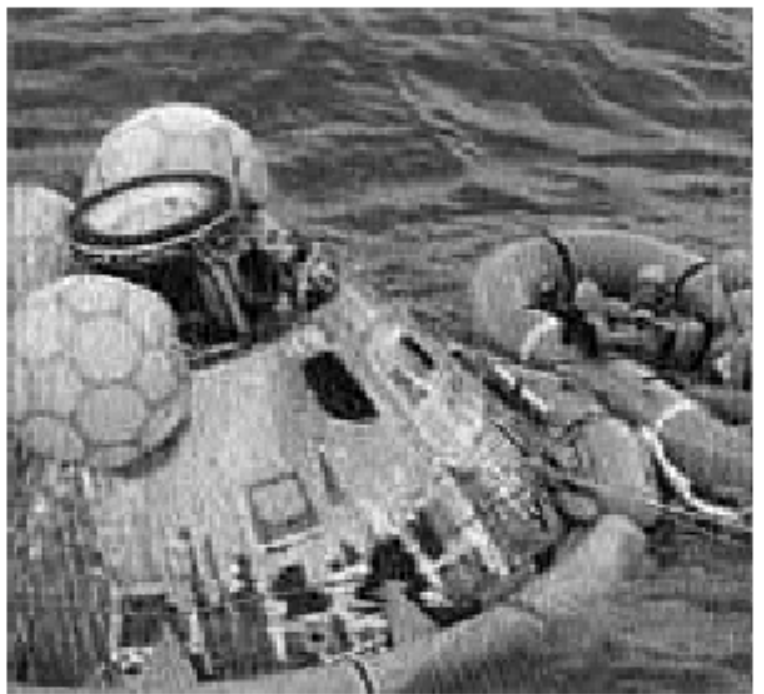
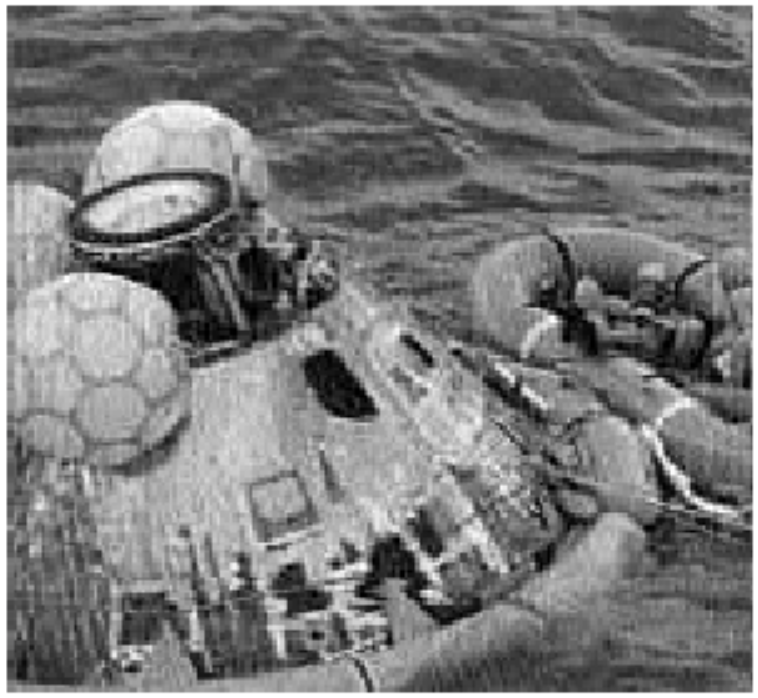
A = Y \* PseudoX + rand(size(Y,1), size(X,1)) \* Z(PseudoX, X);

isCorrect = A \* X;

figure

imshow(uint8(isCorrect));

Вихідні дані



Висновок

Навчився будувати лінійну модель з допомогою псевдообернених операторів користуючись формулами Гревіля та Мура-Пенроуза.