МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Звіт

з лабораторної роботи №4

з дисципліни «Чисельні методи програмування»

Виконав:

Студент групи ФІТ 2-16

Пархоменко Іван Дмитрович

Київ 2024

Завдання

1. Знайти матрицю C = $AB - BA$:

$$2)\ A = \begin{pmatrix} 2 & 3 & 1 \\ -1 & 1 & 0 \\ 1 & 2 & -1 \end{pmatrix},$$

```
import numpy as np
m_sqr_arr = np.array([[2, 3, 1], [-1, 1, 0], [1, 2, -1]])
print("Матриця:")
print(m_sqr_arr)
```

2.

$$2)\ \begin{pmatrix} -1 & 0 & 2 \\ 0 & 1 & 0 \\ 1 & 2 & -1 \end{pmatrix}^2;$$

```
import numpy as np

m_sqr_arr = np.array([[-1, 0, 2], [-1, 1, 0], [1, 2, -1]])

power = 2
result_matrix = np.power(m_sqr_arr, power)

print("Matrix raised to the power of", power, ":")
print(result_matrix)
```

$$2)\ \begin{pmatrix} 5 & 8 & -4 \\ 6 & 9 & -5 \\ 4 & 7 & -3 \end{pmatrix} \cdot \begin{pmatrix} 3 & 2 & 5 \\ 4 & -1 & 3 \\ 9 & 6 & 5 \end{pmatrix};$$

3.

```
import numpy as np

# Define the matrices
m1 = np.array([[5, 8, -4], [6, 9, -5], [4, 7, -3]])
m2 = np.array([[3, 2, 5], [4, -1, 3], [9, 6, 5]])

# Find the product of matrices
product_matrix = np.dot(m1, m2)

print("Product of matrices:")
print(product_matrix)
```

```
Product of matrices:
[[ 11 -22  29]
 [  9 -27  32]
 [ 13 -17  26]]
```

1) $\begin{vmatrix} 2 & 3 & 4 \\ 1 & 0 & 6 \\ 7 & 8 & 9 \end{vmatrix}$;

4.

```python
import numpy as np
# Визначення 3x3 матриць
m1 = np.array([[2, 3, 4], [1, 0, 6], [7, 8, 9]])  # Перша матриця
# Обчислення визначників
det_m1 = np.linalg.det(m1)  # Визначник першої матриці
print("Визначник матриці m1:", det_m1)
```

```
Визначник матриці m1: 35.00000000000001
```

2) $\begin{vmatrix} 2 & 3 & 4 & 1 \\ 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{vmatrix}$

5.

```python
import numpy as np

# Визначення 4x4 матриці
m = np.array([[2, 3, 4, 1],
              [1, 2, 3, 4],
              [3, 4, 1, 2],
              [4, 1, 2, 3]])

# Обчислення визначника
det_m = np.linalg.det(m)

print("Визначник матриці m:", det_m)
```

```
Визначник матриці m: -160.00000000000009
```

$$2)\begin{pmatrix}2 & 5 & 7\\ 6 & 3 & 4\cdot\\ 5 & -2 & -3\end{pmatrix};$$

6.

```python
import numpy as np

# Визначення 3x3 матриці
m = np.array([[2, 5, 7],
              [6, 3, 4],
              [5, -2, -3]])

# Знаходження оберненої матриці
inv_m = np.linalg.inv(m)

print("Обернена матриця:")
print(inv_m)
```

```
C:\Users\Иван\PycharmProjects\pythonProject\.venv\
Обернена матриця:
[[  1.  -1.   1.]
 [-38.  41. -34.]
 [ 27. -29.  24.]]

Process finished with exit code 0
```

7.

$$;2)\begin{pmatrix}1 & -1 & 3 & 4\\ 0 & -1 & 2 & 1\\ 1 & 1 & -1 & 2\\ 2 & 3 & -5 & 3\end{pmatrix};$$

```python
import numpy as np

# Визначення 4x4 матриці
m = np.array([[1, -1, 3, 4],
              [0, -1, 2, 1],
              [1, 1, -1, 2],
              [2, 3, -5, 3]])

# Знаходження рангу матриці
rank_m = np.linalg.matrix_rank(m)

print("Ранг матриці m:", rank_m)
```

```
Ранг матриці m: 3
```

8. $15)\begin{cases} x-2y+z=4, \\ 2x-y+z=3, \\ 3x+2y+2z=2. \end{cases}$ варіант 15

```python
import numpy as np

# Визначення матриці коефіцієнтів (A) та вектора значень (B)
A = np.array([[1, -2, 1],
              [2, -1, 1],
              [3, 2, 2]])

B = np.array([5, -3, 1])
# Метод Крамера
X_kramer = np.linalg.solve(A, B)
print("Розв'язок за допомогою методу Крамера:", X_kramer)
# Матричний метод (обчислення оберненої матриці)
A_inv = np.linalg.inv(A)
X_matrix = np.dot(A_inv, B)
print("Розв'язок за допомогою матричного методу:", X_matrix)
# Метод Гауса
X_gauss = np.linalg.solve(A, B)
print("Розв'язок за допомогою методу Гауса:", X_gauss)
# Перевірка за допомогою функції solve() пакету linalg
X_check = np.linalg.solve(A, B)
print("Перевірка за допомогою solve():", X_check)
```

```
Розв'язок за допомогою методу Крамера: [-7.8 -0.2 12.4]
Розв'язок за допомогою матричного методу: [-7.8 -0.2 12.4]
Розв'язок за допомогою методу Гауса: [-7.8 -0.2 12.4]
Перевірка за допомогою solve(): [-7.8 -0.2 12.4]
```

2.

1.Створіть прямокутну матрицю A, яка має N рядків і M стовпців з випадковими елементами. Знайдіть найменший стовпчастий елемент матриці A, для якого сума абсолютних значень елементів максимальна

```python
import numpy as np

# Встановлення розмірності матриці
N = 5  # кількість рядків
M = 4  # кількість стовпців
```

```python
# Створення прямокутної матриці з випадковими елементами
A = np.random.rand(N, M)

# Знайти стовпчик з максимальною сумою абсолютних значень
column_sum_abs = np.sum(np.abs(A), axis=0)  # сума абсолютних значень для кожного стовпця
max_sum_column_idx = np.argmax(column_sum_abs)  # індекс стовпця з максимальною сумою

# Знайти найменший елемент у знайденому стовпчику
min_element = np.min(A[:, max_sum_column_idx])

# Вивести результат
print("Матриця A:")
print(A)
print("Найменший стовпчастий елемент з максимальною сумою абсолютних значень:", min_element)
```

https://github.com/Bloorel/Numerical-Methods