

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт информационных технологий и телекоммуникаций

Кафедра инфокоммуникаций.

**Дисциплина: Кроссплатформенное программирование**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

**Исследование методов работы с матрицами и векторами с  
помощью библиотеки NumPy**

Выполнила:  
студентка 2 курса  
ИВТ-б-о-19-1  
Хубиева Аида

Проверил:  
Воронкин  
Роман Александрович

Работа защищена с оценкой:

---

Ставрополь, 2021

Цель работы: исследовать базовые возможности библиотеки NumPy языка программирования Python.

Ход работы:

Ссылка на репозиторий: <https://github.com/hubieva-a/lab3.1>

Задание 1. Самостоятельно изучите матричный метод для решения систем линейных уравнений. Приведите алгоритм решения системы линейных уравнений методом Крамера средствами библиотеки NumPy.

Матричный метод

```
In [12]: import numpy as np
A = np.matrix('10 9 -90; 8 18 60; 4 7 13')
B = np.matrix('7; 6; 5')
X = np.matrix('0; 0; 0')
detA = np.linalg.det(A)
invA = np.linalg.inv(A)
X = invA * B
print(X)

[[ 6.23880597]
 [-3.3681592 ]
 [ 0.27860647]]
```

Рисунок 1. Метод решения системы линейных уравнений матричным методом

Задание 2. Самостоятельно изучите матричный метод для решения систем линейных уравнений. Приведите алгоритм решения системы линейных уравнений методом Крамера средствами библиотеки NumPy.

## Метод Крамера

```
In [120]: C = np.matrix('10 9 90; 8 18 60; 4 7 13')
D = np.matrix('10; 20; 30')
X = np.matrix('0; 0; 0')
detC = np.linalg.det(C)
C1 = np.zeros((3,3))
C1 = C.copy()
C1[:, 0] = D
detC1 = np.linalg.det(C1)
C2 = C.copy()
C2[:, 1] = D
detC2 = np.linalg.det(C2)
C3 = C.copy()
C3[:, 2] = D
detC3 = np.linalg.det(C3)
X1 = detC1 / detC
X2 = detC2 / detC
X3 = detC3 / detC
X = np.vstack((X1, X2, X3))
print(X)

[[ 59.70149254]
 [ 15.17313433]
 [ 1.         ]]
```

Рисунок 2. Метод решения системы линейных уравнений методом Крамера

Вопросы для защиты работы:

1. Приведите основные виды матриц и векторов. Опишите способы их создания в языке Python.

`ar = np.array(5, 5)` – матрица с заданными размерами

`ar = np.zeros(5, 5)` – матрица из нулей

`ar = np.eye(5, 5)` – матрица из единиц

`vec = np.array([1, 2])` – вектор (одномерная матрица)

`vec = np.zeros((5, ))` – нулевой вектор

`vec = np.eye((5, ))` – единичный вектор

2. Как выполняется транспонирование матриц?

`ar_t = ar.transpose()`

3. Приведите свойства операции транспонирования матриц.

Свойство 1. Дважды транспонированная матрица равна исходной матрице:

Свойство 2. Транспонирование суммы матриц равно сумме транспонированных матриц:

Свойство 3. Транспонирование произведения матриц равно произведению транспонированных матриц расставленных в обратном порядке

Свойство 4. Транспонирование произведения матрицы на число равно произведению этого числа на транспонированную матрицу

Свойство 5. Определители исходной и транспонированной матрицы совпадают:

4. Какие имеются средства в библиотеке NumPy для выполнения транспонирования матриц?

```
ar_t = ar.transpose()
```

```
ar_t = ar.T
```

5. Какие существуют основные действия над матрицами?

Умножение матрицы на число --  $C = 3 * A$

Сложение матриц --  $C = A + B$

Умножение матриц --  $C = A.dot(B)$

Определитель матрицы -- `np.linalg.det(A)`

6. Как осуществляется умножение матрицы на число?

```
C = 3 * A
```

7. Какие свойства операции умножения матрицы на число?

Свойство 1. Произведение единицы и любой заданной матрицы равно заданной матрице

Свойство 2. Произведение нуля и любой матрицы равно нулевой матрице, размерность которой равна исходной матрицы

Свойство 3. Произведение матрицы на сумму чисел равно сумме произведений матрицы на каждое из этих чисел

Свойство 4. Произведение матрицы на произведение двух чисел равно произведению второго числа и заданной матрицы, умноженному на первое число

Свойство 5. Произведение суммы матриц на число равно сумме произведений этих матриц на заданное число

8. Как осуществляется операции сложения и вычитания матриц?

$$C = A + B, C = A - B$$

9. Каковы свойства операций сложения и вычитания матриц?

Свойство 1. Коммутативность сложения. От перестановки матриц их сумма не изменяется

Свойство 2. Ассоциативность сложения. Результат сложения трех и более матриц не зависит от порядка, в котором эта операция будет выполняться

Свойство 3. Для любой матрицы существует противоположная ей, такая, что их сумма является нулевой матрицей

11. Как осуществляется операция умножения матриц?

$$C = A \cdot B$$

12. Каковы свойства операции умножения матриц?

Свойство 1. Ассоциативность умножения. Результат умножения матриц не зависит от порядка, в котором будет выполняться эта операция

Свойство 2. Дистрибутивность умножения. Произведение матрицы на сумму матриц равно сумме произведений матриц

Свойство 3. Умножение матриц в общем виде не коммутативно. Это означает, что для матриц не выполняется правило независимости произведения от перестановки множителей

Свойство 4. Произведение заданной матрицы на единичную равно исходной матрице

13. Какие имеются средства в библиотеке NumPy для выполнения операции умножения матриц?

$$C = A.\text{dot}(B)$$

14. Что такое определитель матрицы? Каковы свойства определителя матрицы?

Свойство 1. Определитель матрицы остается неизменным при ее транспонировании

Свойство 2. Если у матрицы есть строка или столбец, состоящие из нулей, то определитель такой матрицы равен нулю

Свойство 3. При перестановке строк матрицы знак ее определителя меняется на противоположный

Свойство 4. Если у матрицы есть две одинаковые строки, то ее определитель равен нулю

Свойство 5. Если все элементы строки или столбца матрицы умножить на какое-то число, то и определитель будет умножен на это число

15. Какие имеются средства в библиотеке NumPy для нахождения значения определителя матрицы?

`np.linalg.det(A)`

16. Что такое обратная матрица? Какой алгоритм нахождения обратной матрицы?

Обратная матрица — такая матрица  $A^{-1}$ , при умножении на которую исходная матрица  $A$  даёт в результате единичную матрицу  $E$ .

`np.linalg.inv(A)`

17. Каковы свойства обратной матрицы?

Свойство 1. Обратная матрица обратной матрицы есть исходная матрица

Свойство 2. Обратная матрица транспонированной матрицы равна транспонированной матрице от обратной матрицы

Свойство 3. Обратная матрица произведения матриц равна произведению обратных матриц

18. Какие имеются средства в библиотеке NumPy для нахождения обратной матрицы?

`np.linalg.inv()`

19. Самостоятельно изучите метод Крамера для решения систем линейных уравнений. Приведите алгоритм решения системы линейных уравнений методом Крамера средствами библиотеки NumPy.

20. Самостоятельно изучите матричный метод для решения систем линейных уравнений. Приведите алгоритм решения системы линейных уравнений матричным методом средствами библиотеки NumPy.