# Векторы и матрицы в Python

Для расчетов линейной алгебры в Python имеется ряд библиотек, наиболее распространенной из которых является библиотека

## Векторы (numpy)

### Создание вектора

# Вектор можно задать простым перечислением его элементов:

a = np.array([1, 2, 3])

print(f'Целочисленный вектор:\n{a}')

b = np.array([1, 2, 3, 4, 5], dtype=float)

print(f'Вектор с вещественными значениями:\n{b}')

# Еще один вариант задания вектора – это метод arange, который задает последовательность чисел заданного типа из заданного промежутка

print(a)

# Размер вектора можно вывести при помощи метода .shape.

print(a.shape)

В результате работы кода будет выведен вектор:

[ 1. 2. 3. 4. 5.]  
(5,)

### Арифметика векторов

# Рассмотрим, как в numpy реализуются основные операции векторной алгебры.

import numpy as np

a = np.array([1, 2, 3])

b = np.array([4, 5, 6])

print(a + b, a – b, 5 \* a)

## Матричная алгебра

Матрица в numpy – это двумерный массив.

# Аналогичные операции для матриц:

a = np.array([[1, 2], [8, 9]])

# функция для создания матрицы размера (x,y), заполненной единицами

b = np.ones((2,2))

print(f'Матрица a\n{a}\n')

print(f'Матрица b\n{b}\n')

#print("a + b =", a + b)

#print("2\*a =", 2 \* a)

# Умножение матриц и поэлементное умножение:

c = np.dot(a, b)

print("a ∙ b =", с)

print("поэлементное умножение: ", a \* b)

# Доступ к элементу матрицы А:

A = np.array([[1, 2], [0, 5]])

print (A)

print (A[0][0])

# Транспонирование матрицы А:

f = А.transpose()

print(f)

# Размер матрицы:

d.shape, А.shape

# Нахождение обратной матрицы А и проверка умножением А∙А1=Е:

A1 = np.linalg.inv(A)

print(A1)

print (A \* A1)

D = np.dot(A,A1)

print (np.dot(A,a))

# Создание вектора-строки и вектора-столбца, как матрицы:

a = np.array([[1, 2, 3]], float)

print(a)

b = a.transpose()

print(b)

Результат:

[[ 1. 2. 3.]]

[[ 1.]

[ 2.]

[ 3.]]

# Перемножение матрицы-строки и матрицы-столбца:

np.dot(a, b)

Результат: array([[ 14.]])

np.dot(b, a)

Результат:

array([[ 1., 2., 3.],

[ 2., 4., 6.],

[ 3., 6., 9.]])

# Создание трехмерного массива:

F = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])