

Лабораторная работа № 3

Методы линейной алгебры на Python

Теоретические сведения

1.1. Подключение библиотеки Numpy

Библиотека Numpy содержит большое количество функций для работы с матрицами. Для того чтобы импортировать данный модуль, в самое начало программы нужно добавить следующую строку:

```
import numpy as np
```

После этого можно обращаться к функциям библиотеки через новое имя np.

Пример. Вызов функции формирования единичной матрицы заданной размерности 4:

```
E = np.eye(4) # из модуля numpy вызываем функцию eye
print(E)      # выводим полученную матрицу на экран
```

Модуль numpy.linalg содержит алгоритмы линейной алгебры: нахождение определителя матрицы, решений системы линейных уравнений, обращение матрицы, нахождение собственных чисел и собственных векторов матрицы, разложение матрицы на множители: Холецкого, сингулярное, метод наименьших квадратов и т.д.

1.2. Описание векторов и матриц

Для работы с массивами в Python по умолчанию используется тип данных список, но с точки зрения представления матриц и проведения вычислений с ними стандартные списки – не самый удобный инструмент.

Numpy содержит специальную структуру данных array, которую и будем использовать для создания векторов и матриц.

Описание векторов и матриц

| Команда | Комментарий |
|--|---|
| <code>vr = np.array([1, 2])</code> | Создание вектора-строки $vr = (1 \ 2)$ |
| <code>v0 = np.zeros((5,))</code> | Создание нулевого вектора-строки размера 5 |
| <code>v1 = np.ones((5,))</code> | Создание вектора-строки размера 5, заполненного единицами |
| <code>vc = np.array([[1], [2]])</code> | Создание вектора-столбца $v = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ |

| | |
|--|---|
| m_sqr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) | Создание квадратной матрицы |
| m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]] m_sqr = np.array(m) | $M_{sqr} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ |
| m_sqr = np.matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) | Ещё один способ создания матриц – это построение объекта типа <i>matrix</i> |
| m_sqr = np.matrix('1 2 3; 4 5 6; 7 8 9') | доступен стиль Matlab, когда между элементами ставятся пробелы, а строки разделяются точкой с запятой, при этом такое описание должно быть передано в виде строки |

Пусть в программе определены матрицы

A = np.matrix('1 2 3; 4 5 6; 7 8 9')

B = np.matrix('7 8 9; 0 7 5; 3 2 1')

Действия с матрицами

| <i>Команда</i> | <i>Комментарий</i> |
|--------------------------|--------------------------|
| A.transpose() A.T | Транспонирование матрицы |
| A + B | Сумма матриц |
| A.dot(B) | Произведение <i>AB</i> |
| B.dot(A) | Произведение <i>BA</i> |
| A_det = np.linalg.det(A) | Определитель матрицы |
| A_obr = np.linalg.inv(A) | Обратная матрица |

Указания к выполнению лабораторной работы

В качестве отчета по работе преподавателю предъявляются решения в электронном виде. При необходимости нужно ответить на дополнительные вопросы.

Задание на лабораторную работу 3

Задание 1. Выполнить задания 1–3 из лабораторной работы 2, используя стандартные функции из модуля `numpy.linalg`. Сравнить результаты с полученными ранее.

Задание 2. Найти собственные значения для матрицы своего варианта из задания 3 лабораторной работы 2.

Для этого используйте функцию `linalg.eigvals(A)`

Определить среди них действительные и комплексные значения.

Записать спектр матрицы.

Справочная информация

Numpy. Матричные вычисления:

<https://physics.susu.ru/vorontsov/language/numpy.html>