

- (5 баллов) Найти ближайшую ортогональную матрицу (в смысле нормы Фробениуса) к матрице

$$A = \begin{pmatrix} 0.5 & 2.16506351 & 0.4330127 \\ -0.8660254 & 1.25 & 0.25 \\ 0 & 0.5 & 2.5 \end{pmatrix}$$

Посмотрев на результат, найти соответствующие этому преобразованию угол и направление поворота. Матрицу, угол и направление поворота распечатать в `stdout`.

- (5 баллов) При помощи SVD найти обратную матрицу к квадратной матрице $n \times n$, состоящей из элементов

$$A_{ij} = \frac{1}{i + j - 1},$$

для размеров матрицы $n=3$ и $n=10$. Результат распечатать в `stdout`.

- (5 баллов) Для квадратной матрицы $n \times n$, состоящей из элементов

$$A_{ij} = i + j - 1,$$

найти с использованием SVD все решения уравнений $AX=0$ для $n=4$ Результат распечатать в `stdout`.

- (5 баллов) Написать программу, находящую все точки пересечения двух прямых на плоскости методом однородных координат. Прямые заданы уравнениями $ax+by+c=0$, $cx+dy+e=0$. Коэффициенты задать константами на ваше усмотрение. Результат распечатать в `stdout`.
- Необязательная задача (15 баллов) Найти матрицу B ранга r (для любого значения r), ближайшую (в смысле нормы Фробениуса) к заданной квадратной матрице A (использовать SVD).

Рекомендованные функции: `cv2.SVDecomp` или `numpy.linalg.svd`, `numpy.matmul`, `numpy.divide`, `numpy.squeeze`