1. Найдите собственные векторы и собственные значения для линейного оператора, заданного матрицей:

$$A=\begin{pmatrix}-1 & -6\\ 2 & 6\end{pmatrix}.$$

Решение

Найдём собственные значения линейного оператора, составив и решив характеристическое уравнение:

$$\left|\begin{matrix}-1-\lambda & -6\\ 2 & 6-\lambda\end{matrix}\right|=0,$$

$$(-1-\lambda)(6-\lambda)+2\cdot 6=0,$$

$$\lambda^2-5\lambda+6=0,$$

$$D=1$$

$$\lambda_{1,2}=\frac{5\pm\sqrt{1}}{2}$$

$$\lambda_1=3$$

$$\lambda_2=2$$

Найдем собственные вектора: Для $\lambda_1=2$ $\begin{pmatrix}-1 & -6\\ 2 & 6\end{pmatrix}\cdot\begin{pmatrix}x\\ y\end{pmatrix}=2\begin{pmatrix}x\\ y\end{pmatrix}$ $\begin{cases}-x-6y=2x\\ 2x+6y=2y\end{cases}$, $\begin{cases}x=-2y\\ y=y\end{cases}$ Первый собственный вектор в общем виде $\begin{pmatrix}-2y\\ y\end{pmatrix}$ Для $\lambda_2=3$ $\begin{pmatrix}-1 & -6\\ 2 & 6\end{pmatrix}\cdot\begin{pmatrix}x\\ y\end{pmatrix}=3\begin{pmatrix}x\\ y\end{pmatrix}$ $\begin{cases}-x-6y=3x\\ 2x+6y=3y\end{cases}$, $\begin{cases}x=-\frac{3}{2}y\\ y=y\end{cases}$ Второй собственный вектор в общем виде $\begin{pmatrix}-1.5y\\ y\end{pmatrix}$

```
import numpy as np
A = np.array([[-1, -6], [2, 6]])
w, v = np.linalg.eig(A)
w, v
```

```
(array([2., 3.]),
 array([[-0.89442719,  0.83205029],
        [ 0.4472136 , -0.5547002 ]]))
```

2. Дан оператор поворота на 180 градусов, задаваемый матрицей:

$$A=\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}.$$

Покажите, что **любой** вектор считается для него собственным.

Решение

$$\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix}=\lambda\begin{pmatrix}x\\ y\end{pmatrix}$$

Оператор имеет единственное собственное значение, равное $\lambda_1=\lambda_2=-1$. Запишем систему для нахождения собственных векторов:

$$\begin{cases}-x=-x,\\ -y=-y\end{cases}$$

Система верна при любых значениях x и y, это означает, что любой вектор является для данного оператора собственным

3. Пусть линейный оператор задан матрицей:

$$A=\begin{pmatrix}1 & 1\\ -1 & 3\end{pmatrix}.$$

Установите, считается ли вектор $x=(1,1)$ собственным вектором этого линейного оператора.

Решение

Найдем собственные значения матрицы:

$$\begin{vmatrix}1-\lambda & 1\\ -1 & 3-\lambda\end{vmatrix}=(1-\lambda)(3-\lambda)+1=0$$

$$\lambda_1=\lambda_2=2$$

Подставим в систему наш вектор и проверим, подойдут ли заданные значения:

$$\begin{pmatrix}1 & 1\\ -1 & 3\end{pmatrix}\begin{pmatrix}1\\ 1\end{pmatrix}=2\begin{pmatrix}1\\ 1\end{pmatrix}$$

$$\begin{cases}2=2\\ 2=2\end{cases}$$

Система верна при заданных значениях, значит, вектор является собственным

4. Пусть линейный оператор задан матрицей:

$$A=\begin{pmatrix}0 & 3 & 0\\ 3 & 0 & 0\\ 0 & 0 & 3\end{pmatrix}.$$

Установите, считается ли вектор $x=(3,-3,-4)$ собственным вектором этого линейного оператора.

Решение

Найдем собственные значения

```
A = np.array([[0, 3, 0], [3, 0, 0], [0, 0, 3]])
v, _ = np.linalg.eig(A)
print(v)
```

```
[ 3. -3.  3.]
```

Подставим собственные значения и заданный вектор в систему $Ax=\lambda x$; если вектор является собственным, должно выполняться равенство

$$\begin{pmatrix}0 & 3 & 0\\ 3 & 0 & 0\\ 0 & 0 & 3\end{pmatrix}\begin{pmatrix}3\\ -3\\ -4\end{pmatrix}=\lambda\begin{pmatrix}3\\ -3\\ -4\end{pmatrix}$$

$$\Rightarrow\begin{cases}3\lambda=-9\\ -3\lambda=9\\ -4\lambda=-12\end{cases}\Rightarrow\begin{cases}\lambda=-3\\ \lambda=3\end{cases}.$$

Такая система не имеет смысла, следовательно, **вектор** $x=(3,-3,-4)$ **не считается собственным вектором** линейного оператора, заданного матрицей A.

```
vect = np.array([3, -3, -4])
print(f'A * vect = {A.dot(vect)}')
print(f'lambda1 * vect = {v[0] * vect}')
print(f'lambda2 * vect = {v[1] * vect}')
print(f'lambda3 * vect = {v[2] * vect}')
```

```
A * vect = [ -9   9 -12]
lambda1 * vect = [  9.  -9. -12.]
lambda2 * vect = [-9.  9. 12.]
lambda3 * vect = [  9.  -9. -12.]
```

Вектор не является собственным

```
b = np.array([[1, 1], [-1, 3]])
v, w = np.linalg.eig(A)
w
```

```
array([[ 0.70710678, -0.70710678,  0.        ],
       [ 0.70710678,  0.70710678,  0.        ],
       [ 0.        ,  0.        ,  1.        ]])
```