Практическое задание к уроку 5 "Сингулярное разложение матриц".

1. Найти с помощью NumPy SVD для матрицы

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 5 \\ 3 & -4 & 2 \\ 1 & 6 & 5 \\ 0 & 1 & 0 \end{pmatrix}.$$

In [1]:
```python
import numpy as np
import math
np.set_printoptions(precision=2, suppress=True)
```

Зададим матрицу A:

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 5 \\ 3 & -4 & 2 \\ 1 & 6 & 5 \\ 0 & 1 & 0 \end{pmatrix}$$

In [2]:
```python
A = np.array([[1, 2, 0], [0, 0, 5], [3, -4, 2], [1, 6, 5], [0, 1, 0]])
print(f'Матрица A:\n\n{A}')
```

```
Матрица A:

[[ 1  2  0]
 [ 0  0  5]
 [ 3 -4  2]
 [ 1  6  5]
 [ 0  1  0]]
```

In [3]:
```python
# SVD для матрицы A
U, S, W = np.linalg.svd(A)
V = W.T
D = np.zeros_like(A, dtype=float)
D[np.diag_indices(min(A.shape))] = S
```

In [4]:
```python
print(f'   Элементы, лежащие на главной диагонали матрицы D,\n\
являются сингулярными числами матрицы A:\n\n{D}')
```

```
   Элементы, лежащие на главной диагонали матрицы D,
являются сингулярными числами матрицы A:

[[8.82 0.   0.  ]
 [0.   6.14 0.  ]
 [0.   0.   2.53]
 [0.   0.   0.  ]
 [0.   0.   0.  ]]
```

$$D = \begin{pmatrix} 8.82 & 0 & 0 \\ 0 & 6.14 & 0 \\ 0 & 0 & 2.53 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

In [5]:
```python
print(f'   Столбцы матриц U и V называются левыми и правыми\n\
сингулярными векторами матрицы A:\n\nU:\n{U}\n\nV:\n{V}')
```

```
   Столбцы матриц U и V называются левыми и правыми
сингулярными векторами матрицы A:

U:
[[ 0.17  0.16 -0.53 -0.8  -0.16]
 [ 0.39 -0.53  0.61 -0.43  0.03]
 [-0.14 -0.82 -0.52  0.14  0.07]
 [ 0.89  0.06 -0.25  0.38 -0.06]
 [ 0.08  0.11 -0.08 -0.11  0.98]]

V:
[[ 0.07 -0.37 -0.93]
 [ 0.72  0.67 -0.21]
 [ 0.69 -0.65  0.31]]
```

$$U = \begin{pmatrix} 0.17 & 0.16 & -0.53 & -0.8 & -0.16 \\ 0.39 & -0.53 & 0.61 & -0.43 & 0.03 \\ -0.14 & -0.82 & -0.52 & 0.14 & 0.07 \\ 0.89 & 0.06 & -0.25 & 0.38 & -0.06 \\ 0.08 & 0.11 & -0.08 & -0.11 & 0.98 \end{pmatrix}; V = \begin{pmatrix} 0.07 & -0.37 & -0.93 \\ 0.72 & 0.67 & -0.21 \\ 0.69 & -0.65 & 0.31 \end{pmatrix}$$

In [6]:
```python
print(f'Проверим на ортоганальность матрицу U:\n\n{np.dot(U.T, U)}')
```

```
Проверим на ортоганальность матрицу U:

[[ 1.  0. -0. -0. -0.]
 [ 0.  1.  0.  0.  0.]
 [-0.  0.  1. -0. -0.]
 [-0.  0. -0.  1. -0.]
 [-0.  0. -0. -0.  1.]]
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

In [7]:
```python
print(f'Проверка:\n\n{np.dot(np.dot(U, D), V.T)}')
```

```
Проверка:

[[ 1.  2.  0.]
 [ 0. -0.  5.]
 [ 3. -4.  2.]
 [ 1.  6.  5.]
 [-0.  1. -0.]]
```

2. Для матрицы из предыдущего задания найти:

а) евклидову норму;

In [8]:
```python
print(f'Евклидова норма равна: {round(max(S),2)}')
```

```
Евклидова норма равна: 8.82
```

б) норму Фробениуса.

In [9]:
```python
norm_f = [item**2 for item in S]
print(f'Норма Фробениуса равна:{round(math.sqrt(sum(norm_f)),2)}')
```

```
Норма Фробениуса равна:11.05
```