

# SPRAWOZDANIE

Zajęcia: Matematyka konkretna

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 01 Data 28-02-2026 Temat: "Analiza macierzowa w informatyce" Wariant 7	Łukasz Pezda Informatyka II stopień, niestacjonarne, 2 semestr
---	---

## 1. Cel sprawozdania

Zadanie dotyczy kompresji obrazu metodą SVD zgodnie z wariantem zadania.

Jaka powinna być użyta liczba wartości singularnych żeby zachować 90% informacji na obrazie.

## 2. Opis programu opracowanego

Link do GitHub <https://github.com/Pezdus/mk>

Rozkład SVD to jedna z najważniejszych technik faktoryzacji macierzy. Dowolną macierz rzeczywistą  $A$  o wymiarach  $m \times n$  można rozłożyć na iloczyn trzech macierzy:

$$A = U\Sigma V^T$$

gdzie:

- $U$  – macierz ortogonalna o wymiarach  $m \times m$  (lewe wektory singularne),
- $\Sigma$  – macierz diagonalna o wymiarach  $m \times n$ , zawierająca na głównej przekątnej nieujemne wartości singularne  $\sigma_i$  ułożone malejąco,
- $V^T$  – transponowana macierz ortogonalna o wymiarach  $n \times n$  (prawe wektory singularne).

W kontekście analizy obrazów, macierz  $A$  reprezentuje natężenie pikseli. Ze względu na fakt, że wartości singularne szybko maleją, odrzucenie najmniejszych z nich pozwala na znaczącą redukcję rozmiaru danych przy minimalnej utracie jakości wizualnej.

### Podstawowy rozkład SVD i pseudoodwrotność macierzy

W pierwszej fazie ćwiczenia poddano analizie proste macierze numeryczne przy wykorzystaniu biblioteki `scipy.linalg` oraz `numpy`. Obliczono macierz pseudoodwrotną korzystając z gotowej funkcji `pinv()` oraz algorytmu opartego na SVD.

Obliczenie pseudoodwrotności  $A^+$  metodą SVD opiera się na relacji:

$$A^+ = V\Sigma^+U^T$$

Macierz  $\Sigma^+$  tworzy się poprzez odwrócenie niezerowych wartości singularnych. Wyniki uzyskane z implementacji własnej za pomocą SVD były tożsame z wynikami wbudowanej funkcji `pinv()`, co dowodzi poprawności zaimplementowanego algorytmu numerycznego.

```
from numpy import array
from scipy.linalg import svd
A = array([[1, 2], [3, 4], [5, 6]])
print(A)
U, s, VT = svd(A)
print(U)
print(s)
print(VT)
```

```
from numpy import array
from numpy.linalg import pinv
A = array([
    [0.1, 0.2],
    [0.3, 0.4],
    [0.5, 0.6],
    [0.7, 0.8]])
print(A)
B = pinv(A)
print(B)
```

## Przetwarzanie i kompresja obrazu

Właściwa część zadania polegała na wczytaniu pliku graficznego, konwersji do skali szarości i zastosowaniu algorytmu SVD przy użyciu funkcji `np.linalg.svd`.

Następnie przeprowadzono rekonstrukcję obrazu z wykorzystaniem jedynie  $r$  początkowych wartości singularnych dla progów  $r = 5, 20, 100$  oraz 650. Do rekonstrukcji posłużyło mnożenie obciętych macierzy:

$$A_r = U_r \Sigma_r V_r^T$$

Wyniki wizualne wygenerowane za pomocą biblioteki `matplotlib` potwierdzają, że dla małych wartości  $r$  (np.  $r=5$ ) obraz jest całkowicie rozmyty i składa się z bazowych bloków (pionowych/poziomych przejść tonalnych). Wraz ze wzrostem  $r$  (np.  $r=100$ ), jakość drastycznie rośnie, stając się trudną do odróżnienia od oryginału dla ludzkiego oka.

```
from matplotlib.image import imread
import matplotlib.pyplot as plt
import numpy as np
import os
plt.rcParams['figure.figsize'] = [16,8]

A = imread('7.webp')

img = plt.imshow(X)
img.set_cmap('gray')
plt.axis('off')
plt.show()

U, S, VT = np.linalg.svd(X,full_matrices=False)
print(S.shape)
S = np.diag(S)
```

```

j=0
for r in (5,20,100,650):
    Xapprox = U[:, :r]@S[0:r, :r]@VT[:, :]
    plt.figure(j+1)
    j += 1
    img = plt.imshow(Xapprox)
    img.set_cmap('gray')
    plt.axis('off')
    plt.title('r='+str(r))
    plt.show()

```

### Wykonanie wariantu zadania: Zachowanie 90% informacji

Aby rozwiązać główny problem badawczy – znalezienie liczby  $r$  pozwalającej zachować 90% informacji – wykorzystano własności macierzy  $\Sigma$ . Ilość "energii" lub informacji w obrazie jest wprost proporcjonalna do sumy wartości singularnych.

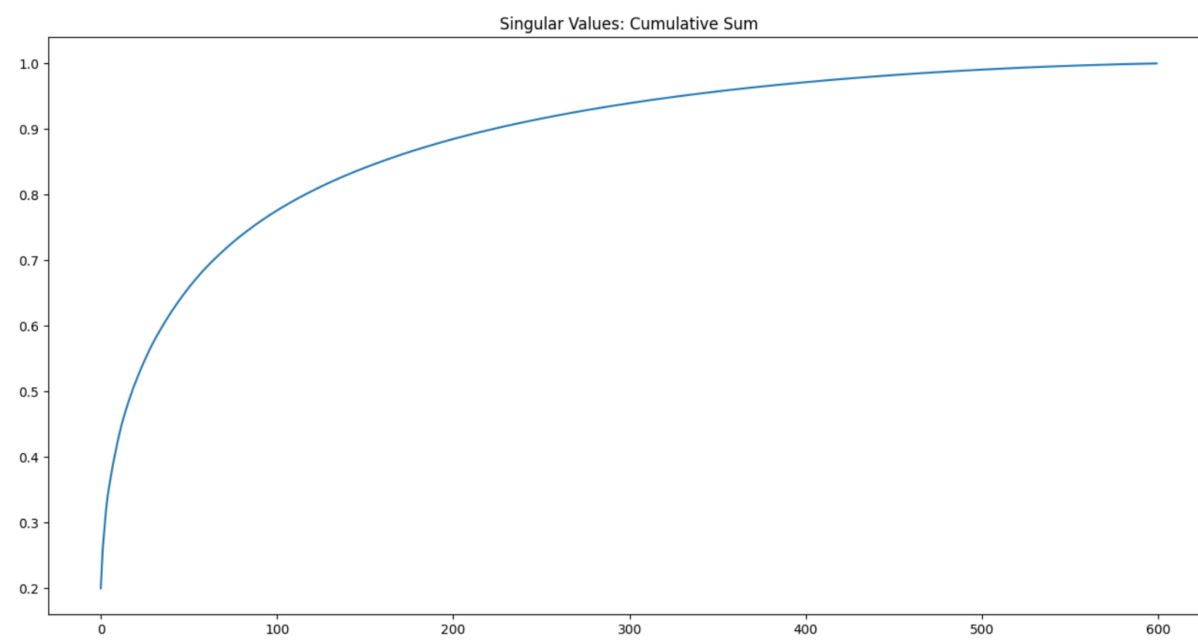
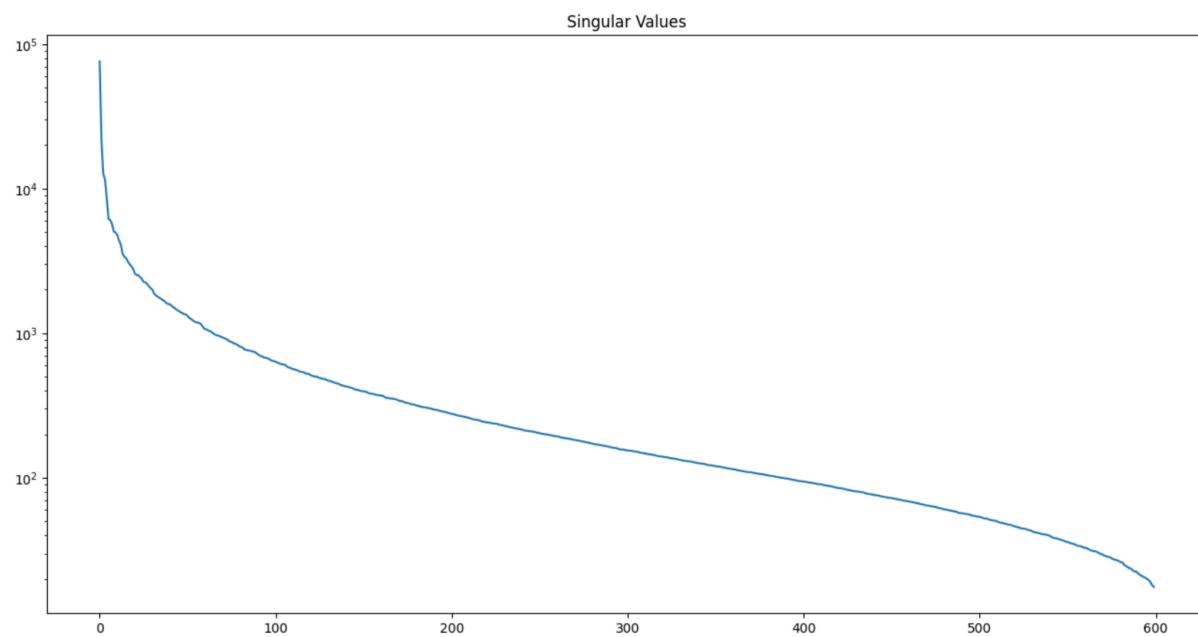
Aby to wyznaczyć programistycznie, przebieg zmienności zachowanej informacji w stosunku do liczby użytych wartości singularnych przedstawiono na wykresie skumulowanej sumy. W kodzie zrealizowano to za pomocą algorytmu

```

info_ratio = np.cumsum(np.diag(S)) / np.sum(np.diag(S))
r_90 = np.argmax(info_ratio >= 0.90) + 1

```

Z przeprowadzonych obliczeń numerycznych wynika, że aby zachować 90% sumy wartości singularnych badanego obrazu, należy zrekonstruować go za pomocą  $r = 224$  wartości singularnych.



### 3. Wnioski

Metoda SVD jest niezwykle skutecznym narzędziem do redukcji wymiarowości i kompresji danych obrazowych.

Jak dowiódł wygenerowany w programie wykres półlogarytmiczny spadku wartości singularnych, kluczowe cechy obrazu reprezentowane są przez bardzo małą część pierwszych, największych wartości singularnych. Większość z nich odpowiada za szum wizualny i subtelne detale.

SVD stanowi stabilną numerycznie i uniwersalną metodę poszukiwania pseudoodwrotności Moore'a-Penrose'a dla macierzy prostokątnych, co ma zastosowanie w rozwiązywaniu nadokreślonych układów równań

Pomimo wysokiej skuteczności, sam rozkład SVD jest dość kosztowny obliczeniowo dla macierzy o dużych wymiarach