

# SPRAWOZDANIE

Zajęcia: Matematyka Konkretna

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

Laboratorium Nr 10 Data 12.12.2023 Temat: Rozkład w trybie dynamicznym Wariant 6	Rafał Klinowski Informatyka II stopień, stacjonarne, 2 semestr, gr. a
---	--

## 1. Polecenie:

Ćwiczenie polegało na stworzeniu notatnika Jupyter w języku Python do zapoznania się z rozkładem dynamicznym (DMD) na podstawie podanych macierzy  $X$  i  $X'$ .

Wariant zadania: 6

## 2. Napisany program, uzyskane wyniki

Podczas realizacji laboratorium skorzystano z funkcji, obliczeń i parametrów zaproponowanych w instrukcji laboratoryjnej do laboratorium 10 – DMD.

Poniżej przedstawiona jest implementacja algorytmu DMD.

```
def DMD(X,Xprime,r):  
    U,Sigma,VTr = np.linalg.svd(X,full_matrices=0) # Step 1  
    Ur = U[:,r]  
    Sigmar = np.diag(Sigma[r])  
    VTr = VTr[:,r]  
    Atilde = np.linalg.solve(Sigmar.T,(Ur.T @ Xprime @ VTr.T).T).T # Step 2  
    Lambda, W = np.linalg.eig(Atilde) # Step 3  
    Lambda = np.diag(Lambda)  
  
    Phi = Xprime @ np.linalg.solve(Sigmar.T,VTr).T @ W # Step 4  
    alpha1 = Sigmar @ VTr[:,0]  
    b = np.linalg.solve(W @ Lambda,alpha1)  
    return Phi, Lambda, b
```

Następnie wczytano z pliku tekstowego CSV wartości  $X$  i  $X'$ , oraz przekonwertowano je na macierze. Konieczne było wczytanie danych w odpowiedni sposób, poprawnie oddzielając liczby po przecinku.

```
X = pd.read_table("War6_X.csv", delimiter=";", decimal=",")  
Xprime = pd.read_table("War6_Xprime.csv", delimiter=";", decimal=",")
```

```
X = np.matrix(X)
Xprime = np.matrix(Xprime)
```

Dla tak wczytanych danych możliwe było wywołanie DMD.

```
Phi, Lambda, b = DMD(X[:, :-1], Xprime[:, 1:], 5)
```

```
print("Phi:", Phi)
print("Lambda: ", Lambda)
print("b:", b)
```

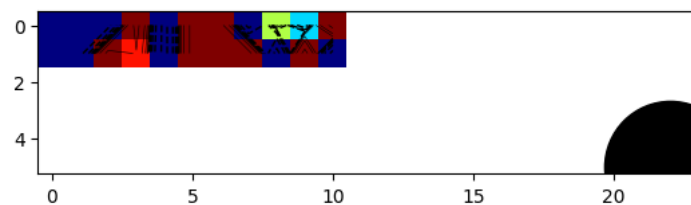
```
print("Phi shape:", Phi.shape)
```

```

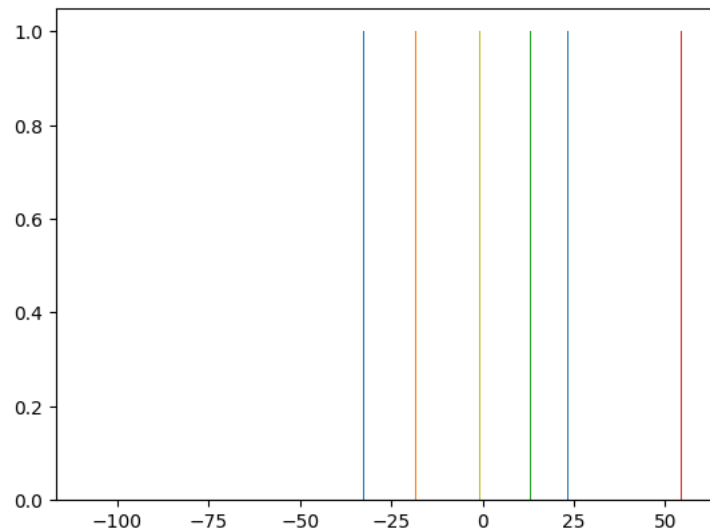
      0.      +0.j      ]
[  0.      +0.j      0.      +0.j
  0.      +0.j      67.9615023 +39.08470136j
  0.      +0.j      ]
[  0.      +0.j      0.      +0.j
  0.      +0.j      0.      +0.j
 67.9615023 -39.08470136j]]
b: [[-1.27239745e+21-1.90662979e+04j]
     [ 9.51610605e+18+2.92253423e+19j]
     [ 9.51610605e+18-2.92253423e+19j]
     [-1.47827910e+20-2.52738254e+20j]
     [-1.47827910e+20+2.52738254e+20j]]
Phi shape: (22, 5)
```

*Rysunek 1. Fragment uzyskanych wyników.*

Analogicznie do instrukcji laboratoryjnej utworzono wykresy.



*Rysunek 2. Wykres cylindryczny dla uzyskanego Phi.*



*Rysunek 3. Wykres słupkowy dla  $\Phi$ .*

Wnioski:

- DMD, o ile jest dość proste w implementacji, jest dość skomplikowanym algorytmem, trudnym do zrozumienia w pełni, który umożliwia uzyskanie rozkładu dla pewnych danych (szeregu czasowego)
- Wykorzystane dane, ze względu na ich rozmiar, ciężko było przedstawić na wykresie – ponieważ w wyniku uzyskano macierz o rozmiarze 22x1, trudno było ją reprezentować jako macierz o większym rozmiarze (np. 199x449)

Repozytorium zawierające uzyskane wyniki wraz z niezbędnymi plikami:

<https://github.com/Stukeley/MatematykaKonkretna/tree/master/Lab10>