# Sprawozdanie nr. 2 - Metody numeryczne i optymailzacja

Jakub Andryszczak 259519,
Jakub Żak 244255,
Maciej Cierpisz 249163

# Spis treści

# 1 Zadanie nr. 1

# 2 Zadanie nr. 2

Rozwiązać poniższy układ równań liniowych metodą eliminacji Gaussa z częściowym wyborem element podstawowego:

```
import numpy as np

A = np.array([ [4, 2, 0, 0],
               [1, 4, 1, 0],
               [0, 1, 4, 1],
               [0, 0, 2, 4],])

x = np.random.rand(4)
print(x)
print()

def normal_power_method(A,x):
    for i in range(200):
     x = np.dot(A,x)
     x  = x/np.linalg.norm(x)

    return np.dot(np.dot(A,x),x)/np.dot(x, x)

def shifted_power_method(A, x0, tol=1e-6):
  n = len(A)
  # Estimate a shift close to the smallest eigenvalue
  sigma = np.trace(A) / n
  x = x0.copy()
  for _ in range(200):
    y = np.dot(A, x) - sigma * x

  lambda_ = np.dot(x.T, np.dot(A, x))
  return lambda_


lambda1 = normal_power_method(A, x)
lambda2 = 1 / normal_power_method(A, x)
lambda3 = shifted_power_method(A, x)
lambda4 = 1 / shifted_power_method(A, x)

print("Largrest eigenvalue normal power",lambda1)
print("Smallest eigenvalue normal power", lambda2)
print("Largest eigenvalue shifted power", lambda3)
print("Smallest eigenvalue shifted power", lambda4)
```

Poniżej wynik działania programu:

```
Largrest eigenvalue normal power 5.68226505344719
Smallest eigenvalue normal power 0.17598615879302257
Largest eigenvalue shifted power 6.304928557128195
Smallest eigenvalue shifted power 0.15860607950417216
```

Rys. 2 Wyniki obliczeń

# 3 Zadanie nr. 3

# 4 Zadanie nr. 4

# 5 Zadanie nr. 5

# 6 Zadanie nr. 6

# 7 Zadanie nr. 7