Metody numeryczne

Mateusz Kwolek

6. Dana jest macierz

$$\begin{bmatrix} 2 & -1 & 0 & 0 & 1 \\ -1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 & -1 \\ 1 & 0 & 0 & -1 & 2 \end{bmatrix} \tag{5}$$

Znajdź jej (przybliżony) wektor własny do wartości własnej $\lambda \simeq 0.38197$.
Uwaga: w zadaniu **nie** chodzi o to, aby znaleźć **wszystkie** wartości własne powyższej macierzy, a następnie wskazać wektor własny odpowiadający podanej przybliżonej wartości własnej. Prawidłowe rozwiązanie nie obejmuje szukania żadnych wartości własnych, a **jedynie** konstrukcję (przybliżonego) wektora własnego odpowiadającego podanej (przybliżonej) wartości własnej.

1) Wybór algorytmu

Zaimplementowany przeze mnie algorytm jest jedną z metod iteracyjnych używanych do znajdowania wektorów własnych dla danej wartości własnej. Wykorzystuje on fakt, że dla macierzy A i wartości własnej λ wektor własny *v* spełnia równanie $\boldsymbol{(A-\lambda I)v=0}$. Na początku tworzony jest wektor odpowiadający rozmiarowi macierzy A, następnie powyższe równanie jest rozwiązywane w pętli, a otrzymywany wektor *v* jest normalizowany. Całość odbywa się do momentu aż nie uzyskamy wystarczająco zadowalającego wyniku. *(określonego poprzez ustaloną tolerancję tol=1e-10)*

2) Kod źródłowy

Do stworzenia programu użyłem języka Python ze względu na jego prostotę, bogactwo bibliotek oraz częste zastosowanie do wykonywania obliczeń matematycznych.

```python
import numpy as np
np.set_printoptions(suppress=True, linewidth=1000)

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

EIGEN_VALUE = 0.38197


# Algorytm iteracyjny do znajdowania wektora własnego
def find_eigen_vector(A, eigen_value, iterations=1000, tol=1e-10):
    size = len(A)

    vec = np.random.rand(size)
    vec /= np.linalg.norm(vec)
    for j in range(iterations):
        vec_z = np.linalg.solve(A - eigen_value * np.eye(size), vec)
        vec_new = vec_z / np.linalg.norm(vec_z)

        if np.linalg.norm(vec_new - vec) < tol:
            break

        vec = vec_new

    return vec_new

eigen_vector = find_eigen_vector(A, EIGEN_VALUE)

# Wypisanie wyniku
print("Eigen vector for eigen value of:", EIGEN_VALUE, "is:\n", eigen_vector)
```

3) Wynik

```
Eigen vector for eigen value of:  0.38197 is:
 [ 0.60150096  0.37174803  0.         -0.37174803 -0.60150096]
```