Zadanie NUM 5- Sprawozdanie

Karol Cichowski

1. Wstęp

(zadanie numeryczne NUM 6) Zadana jest macierz

$$\mathbf{M} = \begin{pmatrix} 9 & 2 & 0 & 0 \\ 2 & 4 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}.$$

(a) Stosując metodę potęgową znajdź największą co do modułu wartość własną macierzy $\mathbf{M}$ oraz odpowiadający jej wektor własny. Na wykresie w skali logarytmicznej zilustruj zbieżność metody w funkcji ilości wykonanych iteracji.

(b) Stosując algorytm QR bez przesunięć, opisany w zadaniu nr 6, znajdź wszystkie wartości własne macierzy $\mathbf{M}$. Sprawdź, czy macierze $\mathbf{A}_i$ upodabniają się do macierzy trójkątnej górnej w kolejnych iteracjach. Przeanalizuj i przedstaw na odpowiednim wykresie, jak elementy diagonalne macierzy $\mathbf{A}_i$ ewoluują w funkcji indeksu i.

(c) Zastanów się, czy zbieżność algorytmu z pkt. (a) i (b) jest zadowalająca. Jak można usprawnić te algorytmy?

Wyniki sprawdź używając wybranego pakietu algebry komputerowej lub biblioteki numerycznej.

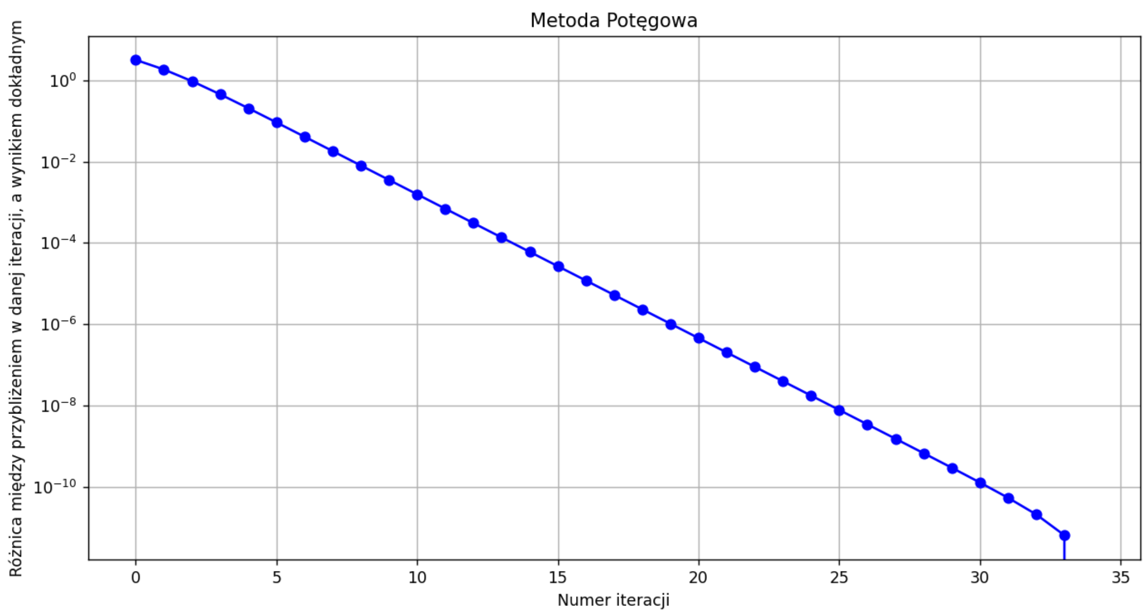
Przyjmujemy dokładność obliczeń równą 1e-12

2. Wyniki

Podpunkt (a):

największa wartość własna obliczona za pomocą metody potęgowej: 9.718548254114584

wektor własny do największej wartości własnej to: [0.93984758 0.33766292 0.05124653 0.0066394]



Podpunkt (b):

Wszystkie wartości własne obliczone za pomocą algorytmu QR:

- λ_1= 9.718548254119618
- λ_2= 4.301704905012378
- λ_3= 2.740194113151937
- λ_4= 1.2395527277160612

Macierz trójkątna górna która powstała przez algorytm QR:

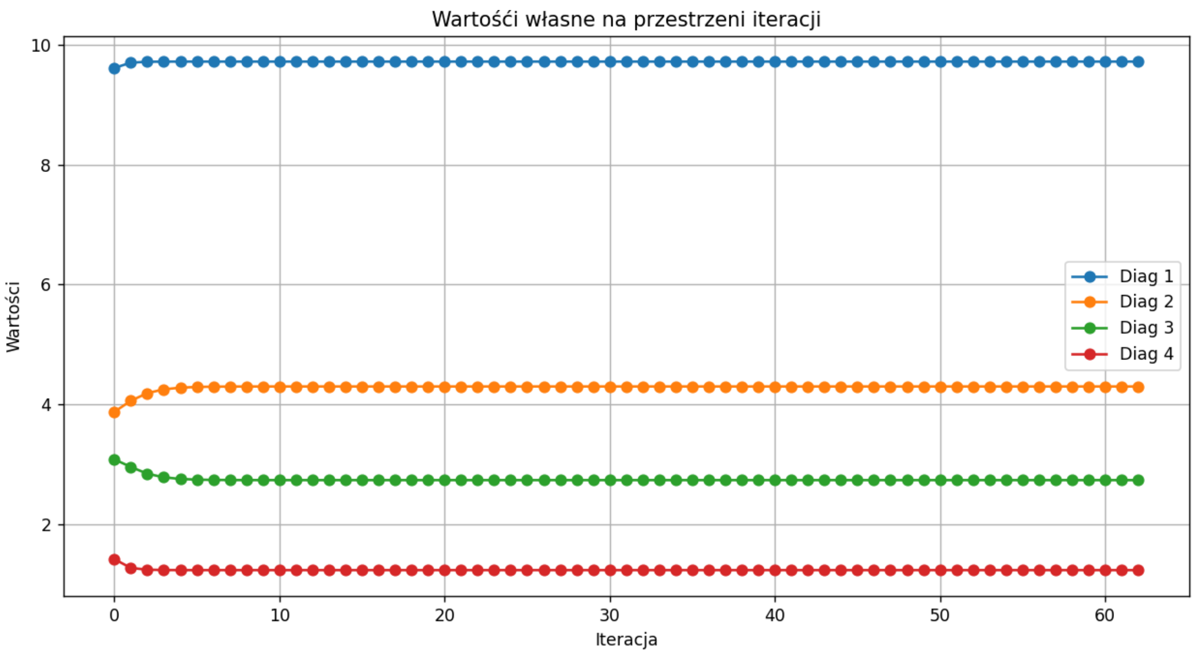
[[9.71854825e+00 1.10981252e-16 -4.42873937e-17 -6.35662595e-17]

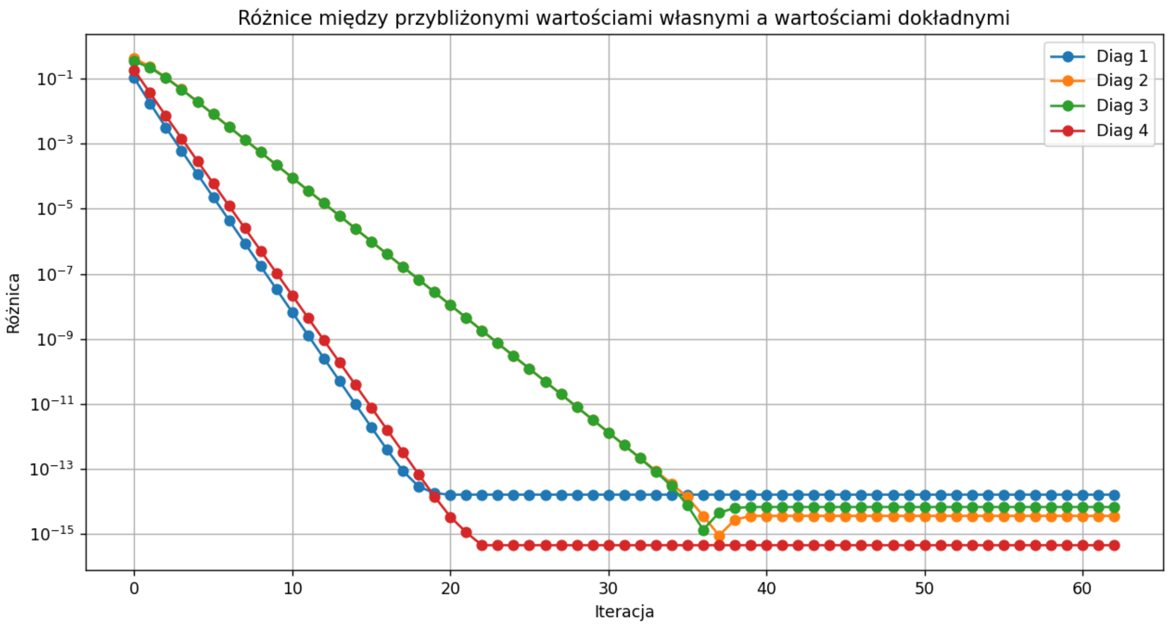
 [7.40723393e-23 4.30170491e+00 7.77015439e-13 2.04168533e-16]

 [0.00000000e+00 7.76383759e-13 2.74019411e+00 1.05768135e-16]

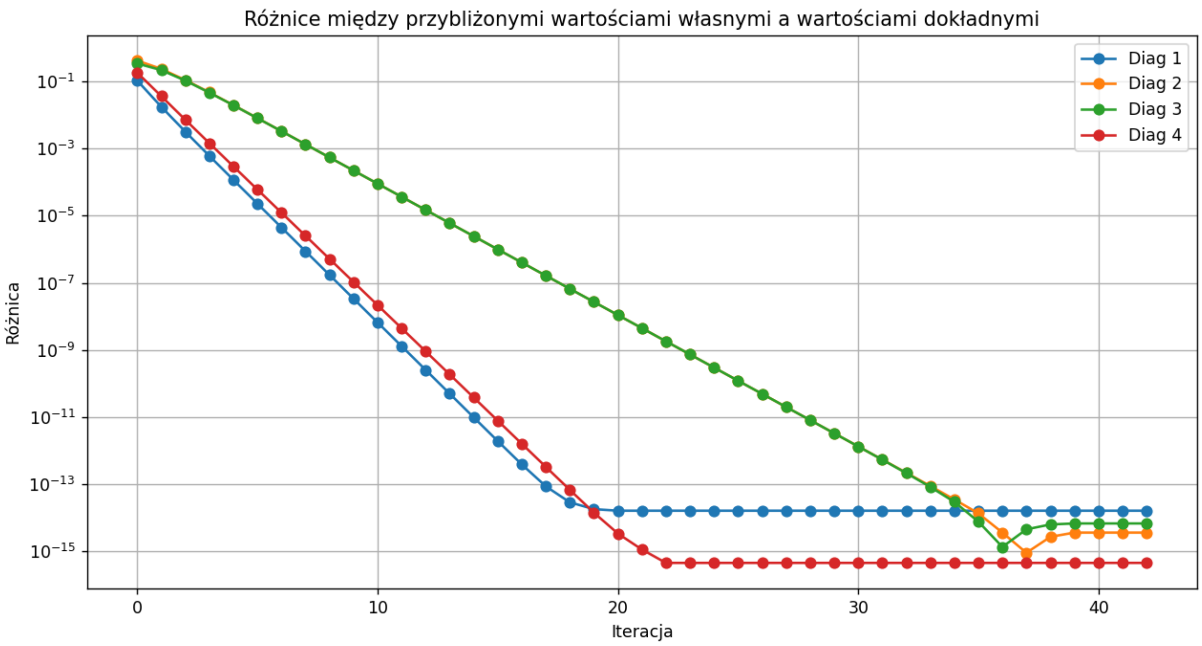
 [0.00000000e+00 0.00000000e+00 -2.18954204e-22 1.23955273e+00]]

(elementy pod diagonalą są na tyle małe że możemy je pominąć)





Jak widzimy nasze wartości własne są już dokładne po mniej niż 40 iteracjach, ale nasz algorytm dalej działa bo A_k nie jest macierzą trójkątną górną, nie interesuje nas dokładność macierzy górnej trójkątnej a tylko dokładność wartości własnych więc w tym wypadku możemy zmniejszyć dokładność sprawdzania postaci macierzy, a dokładność wyników się nie zmieni(zmieniamy dokładność na 1e-8)



Sprawdzenie z biblioteką numpy:

Wartości własne obliczone za pomocą funkcji bibliotecznych wynosi:

- λ_1= 9.71854825
- λ_2= 4.30170491
- λ_3= 2.74019411
- λ_4= 1.23955273

Wektory własne odpowiadające:

- u_1 = [0.93984758, 0.33766292, 0.05124653, 0.0066394]
- u_2 = [-0.2561512, 0.60173696, 0.69384939, 0.30145019]
- u_3 = [-0.21585202, 0.67559587, -0.41941561, -0.56662922]
- u_4 = [-0.06694031, 0.25974335, -0.58312722, 0.76682138]

3. Dyskusja

Jak widzimy użyte przez nas algorytmy dają poprawne wyniki, a ich różnica między winkami uzyskanymi za pomocą funkcji bibliotecznych jest niewielka i może zostać uznana za błąd zaokrąglenia, lub mniejszą dokładność obliczeń.

Czy nasze algorytmy są jednak optymalne?

Nasz algorytm z podpunktu (a) można uznać za optymalny ponieważ największa wartość własna jest dobrze odseparowana on innych więc metoda potęgowa zbiega stosunkowo szybko

Algorytm do podpunktu (b) możemy ulepszyć poprzez zoptymalizowanie faktoryzacji QR. Wiemy że jej złożoność obliczeniowa to $O(N^3)$, dla każdej iteracji naszego algorytmu. Natomiast dzięki temu że nasza macierz M jest macierzą trójdiagonalną możemy przyśpieszyć ten proces, poprzez dokonywanie faktoryzacji QR za pomocą metody obrotów Givens'a dzięki czemu złożoność obliczeniowa każdej iteracji wynosiła by O(N)