

Zadanie Numeryczne 6

Joanna Szewczyk

3.12.2024r

Treść zadania

Dane jest macierz:

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

- (a) Stosując metodę potęgową, znajdź największą (co do modułu) wartość własną macierzy **M** oraz odpowiadający jej wektor własny. Na wykresie w skali logarytmicznej zilustruj zbieżność metody w funkcji liczby wykonanych iteracji.
- (b) Stosując algorytm QR bez przesunięć (opisany w zadaniu nr 6), znajdź wszystkie wartości własne macierzy **M**. Sprawdź, czy macierze A_i w kolejnych iteracjach upodabniają się do macierzy trójkątnej górnej. Przeanalizuj i przedstaw na odpowiednim wykresie, jak elementy diagonalne macierzy A_i ewoluują w funkcji indeksu i .
- (c) Zastanów się, czy zbieżność algorytmów z pkt. (a) i (b) jest zadowalająca. Jak można usprawnić te algorytmy?

Wyniki sprawdź używając wybranego pakietu algebry komputerowej lub biblioteki numerycznej.

Wstęp

Algorytm QR - to numeryczna metoda służąca do znajdowania wartości własnych macierzy. Opiera się na dekompozycji macierzy A na iloczyn dwóch macierzy: ortogonalnej Q i górnotrójkątnej R . Algorytm iteracyjnie poprawia przybliżenie wartości własnych przez przemnażanie R i Q , aż do uzyskania satysfakcjonującej dokładności wartości własnych, będących elementami diagonalnymi macierzy A_k

Kroki algorytmu:

Dekompozycja QR: Dla macierzy A_1 znajdujemy macierze Q_1 i R_1 , takie że:

$$A_1 = Q_1 R_1$$

gdzie Q jest macierzą ortogonalną ($Q^T Q = I$), a R jest macierzą górnotrójkątną.

Tworzymy nową macierz $A_{(k+1)}$ jako iloczyn R_k i Q_k :

$$A_{(k+1)} = R_k Q_k$$

Powtarzamy kroki, aż macierz A_k stanie się prawie diagonalna. Elementy na przekątnej będą zbiegać do wartości własnych macierzy.

Metoda potęgowa - to iteracyjna technika znajdowania największej (co do modułu) wartości własnej macierzy oraz odpowiadającego jej wektora własnego. Wykorzystuje fakt, że wielokrotne mnożenie macierzy \mathbf{A} przez losowy wektor \mathbf{v} zbiega do wektora własnego odpowiadającego największej wartości własnej.

Kroki metody:

Wybieramy losowy wektor początkowy \mathbf{v}_0 i normalizujemy go:

$$\mathbf{v}_0 = \frac{\mathbf{v}_0}{\|\mathbf{v}_0\|}$$

Iteracyjnie mnożymy i normalizujemy \mathbf{v}_{k+1} , aby zapobiec wzrostowi wartości:

Wartość własną obliczamy jako:

$$\lambda_k = \frac{\mathbf{v}_k^T \mathbf{A} \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{v}_k}$$

Powtarzamy, aż różnica między kolejnymi wartościami własnymi będzie mniejsza niż zadana tolerancja ϵ :

$$|\lambda_{k+1} - \lambda_k| < \epsilon$$

Wyniki

Metoda potęgowa:

Największa wartość własna: 9.718548254100863

Odpowiadający wektor własny: [0.939847099686275, 0.33766403452546456, 0.0512478280189772, 0.006639964052657631]

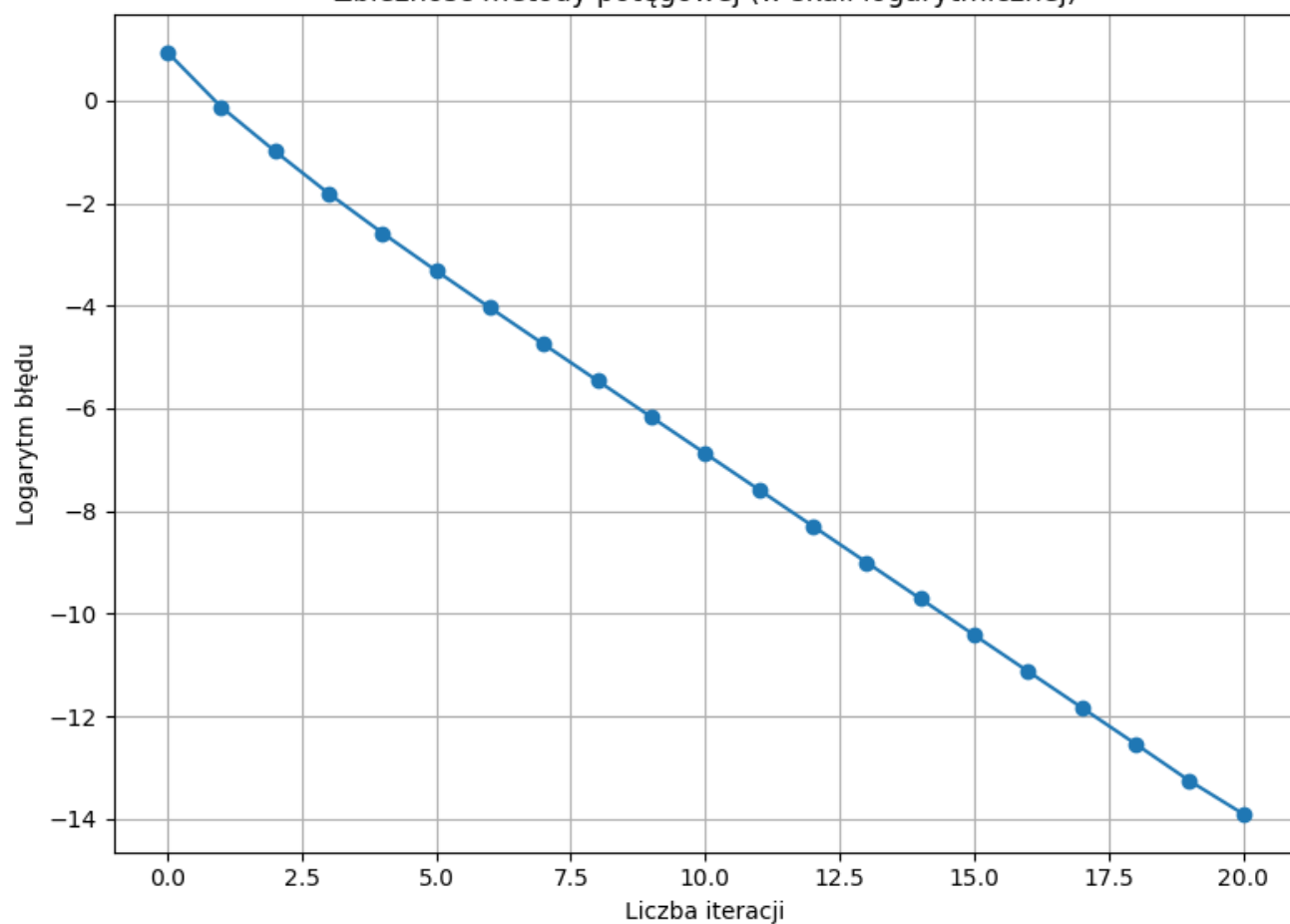
Algorytm QR:

Wartości własne: [9.718548254119625, 4.301704905012373, 2.740194113151942, 1.2395527277160612]

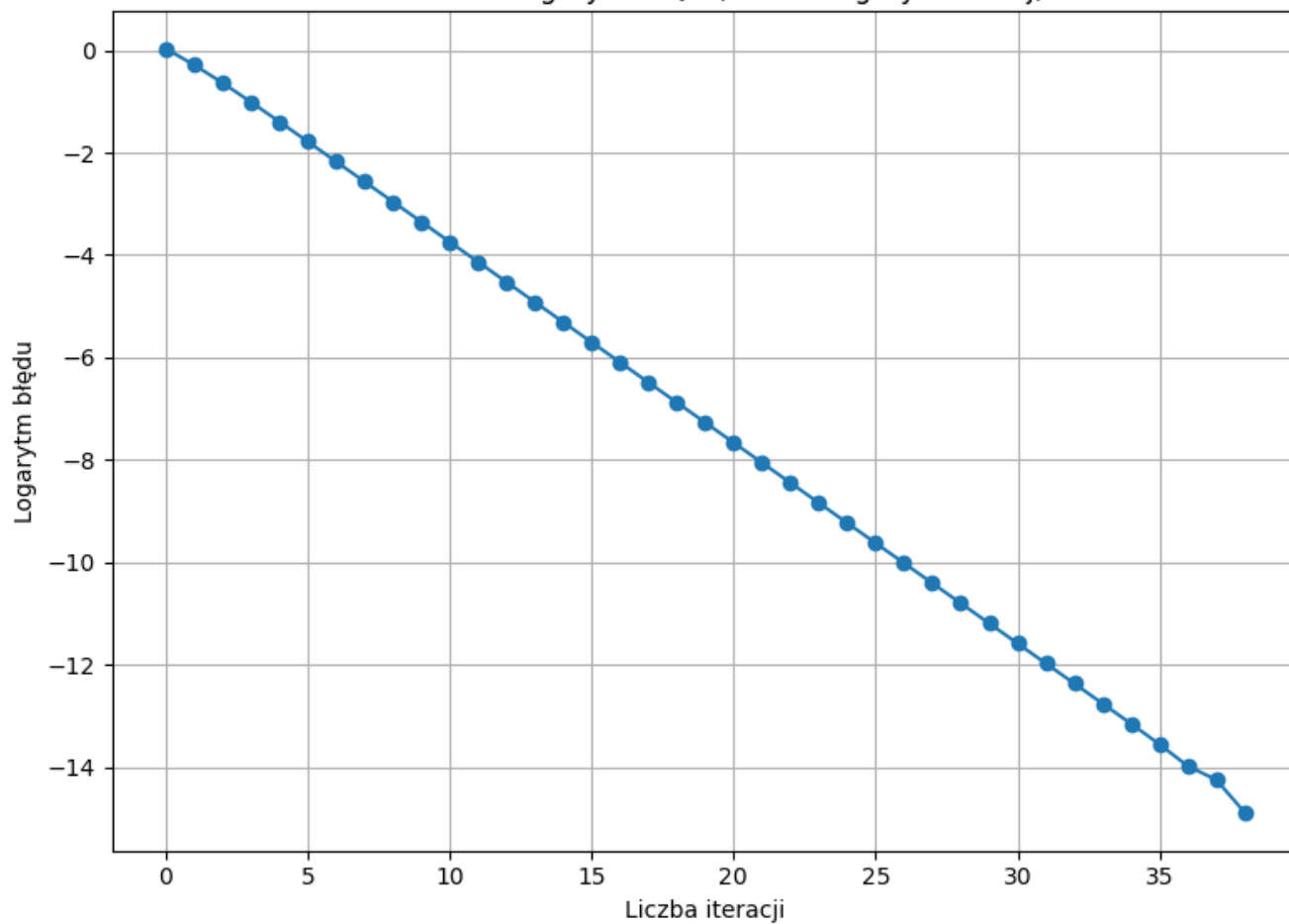
Porównanie wyników z funkcją wbudowaną (scipy.linalg.eig):

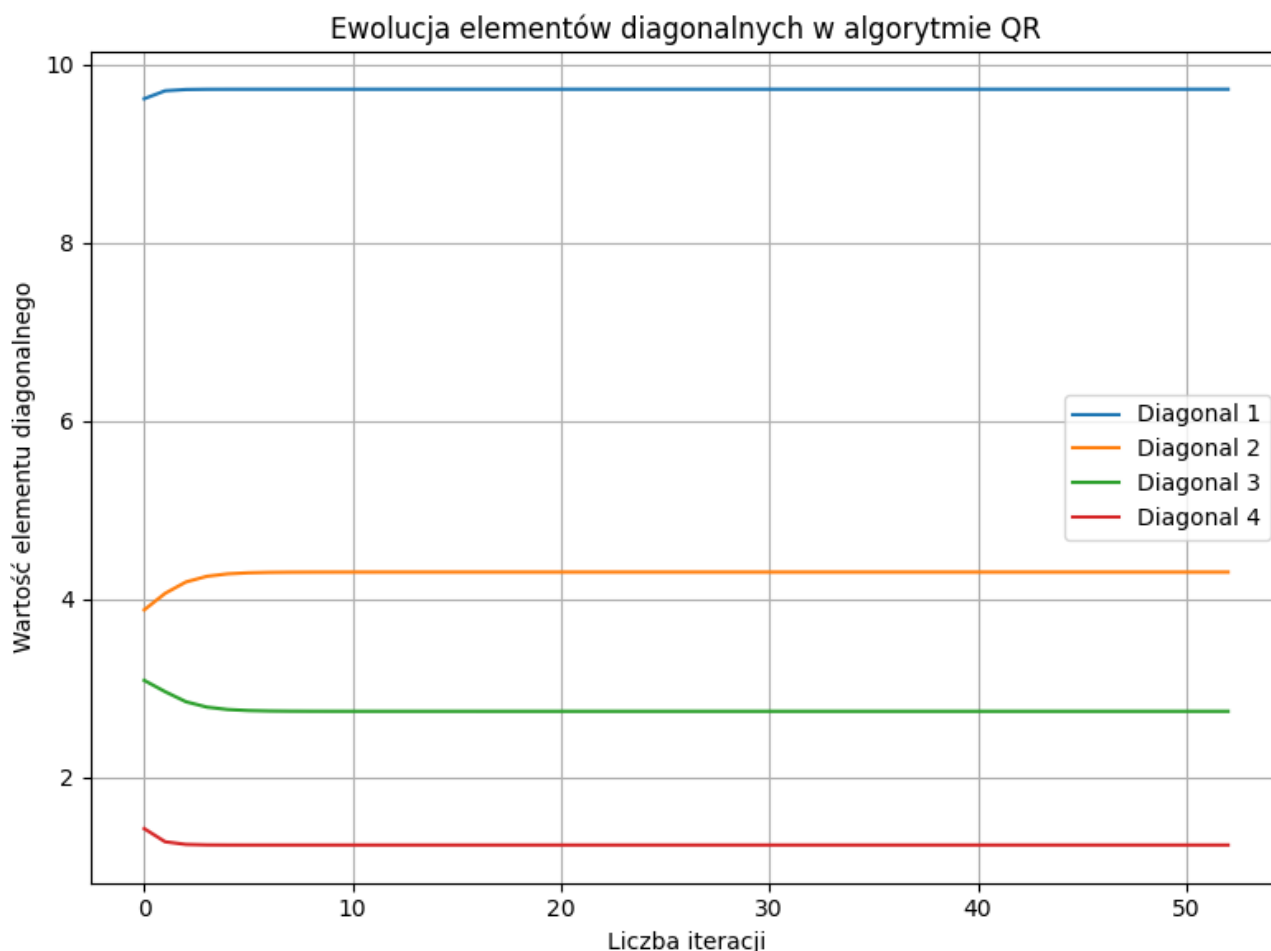
Wartości własne: [9.718548254119634, 4.301704905012374, 2.740194113151944, 1.239552727716061]

Zbieżność metody potęgowej (w skali logarytmicznej)



Zbieżność algorytmu QR (w skali logarytmicznej)





Podsumowanie

Metoda potęgowa zastosowana do naszej macierzy, jak możemy zauważyć na wykresach pozwala na uzyskanie dokładności na tym samym poziomie przy mniejszej ilości iteracji w porównaniu do algorytmu QR. Wadą metody potęgowej jest fakt iż działa ona tylko dla macierzy symetrycznych oraz fakt iż dla większych macierzy i konieczności znalezienia większej ilości wektorów własnych jej działanie stawałoby się bardziej kosztowne. Dla metody potęgowej istotnym usprawnieniem mogłoby być zastosowanie wersji odwrotnej lub przesuniętej, które poprawiają zbieżność w przypadku blisko położonych wartości własnych. W celu usprawnienia działania można zastosować znormalizowaną wersję metody potęgowej, by uniknąć problemów z precyzją.

Algorytm QR jest efektywną metodą znajdowania wartości własnych macierzy poprzez iteracyjne dekompozycje i przemnażanie macierzy, co prowadzi do uzyskania dokładnych wyników. Aby jeszcze bardziej usprawnić jego działanie możemy wykorzystać jedną z jego istotnych zalet, którą jest fakt, iż zachowuje on symetrię oraz strukturę trójdagonalną i postać Hessenberga macierzy. Można by także zastosować przesunięcia spektralne (np. metodą Rayleigha) by przyspieszyć zbieżność.

Porównując nasze rozwiązanie z rozwiązaniem z biblioteki numerycznej możemy zauważyć różnicę dopiero na 14 miejscu po przecinku. Jeśli chcielibyśmy aby nasze wyniki były dokładniejsze należałoby zwiększyć liczbę iteracji w algorytmie lub zastosować bardziej precyzyjne typy danych, takie jak liczby zmiennoprzecinkowe o podwójnej lub rozszerzonej precyzji. Jednak zwiększenie dokładności numerycznej wiąże się również ze wzrostem czasu obliczeń i większym zużyciem pamięci, co należy uwzględnić w praktycznych zastosowaniach.