

Autor: Marcin Sitko

# Sprawozdanie – NUM6

## Treść Zadania

$$M = \begin{pmatrix} 3 & 6 & 6 & 9 \\ 1 & 4 & 0 & 9 \\ 0 & 0.2 & 6 & 12 \\ 0 & 0 & 0.1 & 6 \end{pmatrix}$$

$$B = \begin{pmatrix} 3 & 4 & 2 & 4 \\ 4 & 7 & 1 & -3 \\ 2 & 1 & 3 & 2 \\ 4 & -3 & 2 & 2 \end{pmatrix}$$

- a) Stosując algorytm QR znajdź wszystkie wartości własne macierzy  $M$ .
- b) Stosując metodę potęgową znajdź największą co do modułu wartość własną macierzy  $B$

## Podpunkt A

Macierz  $M$  jest macierzą *Hessenberga*, Aby otrzymać wektor wartości własnych musimy sprowadzić tę macierz do postaci Trójkątnej górnej a następnie odczytać wartości na diagonalu są to składowe wektora wartości własnych aby otrzymać największą wartość własną wybieramy największą wartość spośród zapisanych wartości na diagonalu.

Algorytm  $QR$  który znajduje wartości własne polega na tym, że iterujemy rozkładając kolejne otrzymane macierze na iloczyn dwóch macierzy  $QR$ , można to zapisać:

$$M_n = R_{n-1}Q_{n-1} = Q_nR_n$$

Dzięki takiemu rozkładowi nasza macierz  $M$  zacznie przyjmować postać macierzy trójkątnej górnej gdzie wartości własne będą leżały na diagonalu tej macierzy  $M$ , Jesteśmy w stanie je w bardzo prosty sposób odczytać i wybrać największą z nich ( jeżeli szukamy największej wartości własnej )

Aby zobaczyć działanie tego algorytmu należy wybrać opcje numer 1 menu programu, następnie wpisać ilość iteracji, Aby nasza macierz „zbiegła” do postaci macierzy trójkątnej górnej należy wykonać około 180 iteracji, jest to dość duża ilość biorąc pod uwagę rozmiar naszej macierzy, można powiedzieć że metoda wyznaczania wartości własnych poprzez rozkład  $QR$  „wolno zbieżna” i potrzebuje znacznie więcej iteracji niż metoda wykorzystywana

na przykład w podpunkcie b czyli Metoda potęgowa otrzymywania wartości własnych macierzy.

Ale dlaczego Potrzeba tak dużo „iteracji” ?

Spowodowane jest to tym iż aby móc odczytać wartości własne musimy sprowadzić macierz  $M$  to postaci macierz trójkątnej górnej, tutaj pojawia się problem ponieważ przy mniejszej ilości iteracji macierz posiada pod diagonalą wartości bliskie zero natomiast nie są one zerowe, aby otrzymać wartości równe zero oczywiście dobierając odpowiednią precyzję potrzebujemy więcej iteracji, zadziwiające jest jednak to że w przypadku macierzy  $M$  po mniejszej ilości iteracji niezerowym elementem pod diagonalą zawsze pozostaje tam sama komórka macierzy dla przykładu

*90 iteracji zamiast 180 otrzymujemy następujący wynik*

```
Podaj Ilosc iteracji ( wystarczy ok. 180 ): 90
Macierz po przyblizonym rozkladzie QR
[[ 7.23099219e+00  4.65843629e+00 -1.23917527e+01  9.33319652e+00]
 [ 3.40141949e-08  5.90015738e+00 -2.84824768e+00  5.45978573e+00]
 [ 0.00000000e+00  0.00000000e+00  4.81580660e+00 -8.02266194e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.05304383e+00]]
Wartosci Wlasne Macierzy Hessenberga
[7.230992190356462, 5.900157381083713, 4.8158065969942845, 1.053043831565537]
```

Widać że druga wartość w pierwszej kolumnie przyjmuje wartość rzędu  $1e-8$

*Po Wpisaniu 160 iteracji zamiast 180 otrzymujemy ponownie:*

```
Podaj Ilosc iteracji ( wystarczy ok. 180 ): 160
Macierz po przyblizonym rozkladzie QR
[[ 7.23099231e+00  4.65843628e+00 -1.23917528e+01  9.33319666e+00]
 [ 2.22976169e-14  5.90015727e+00 -2.84824736e+00  5.45978551e+00]
 [ 0.00000000e+00  0.00000000e+00  4.81580659e+00 -8.02266192e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.05304383e+00]]
Wartosci Wlasne Macierzy Hessenberga
[7.2309923094192055, 5.900157268333202, 4.815806590682052, 1.053043831565537]
```

Widać że druga wartość w pierwszej kolumnie przyjmuje wartość rzędu  $1e-14$  Wszystkie inne komórki pod diagonalą zbiegają do zera po około 30 iteracjach natomiast ta jedna zawsze pozostaje większa od reszty

Po 180 Iteracjach wszystkie komórki zbiegają do 0 (oczywiście przy przyjęciu precyzji rzędu  $1e-15$ ) Wyniki Prezentują się w następujący sposób

```
Podaj Ilosc iteracji ( wystarczy ok. 180 ): 180

Macierz po przyblizonym rozkladzie QR

[[ 7.23099231  4.65843628 -12.39175275  9.33319666]
 [ 0.         5.90015727 -2.84824736  5.45978551]
 [ 0.         0.         4.81580659 -8.02266192]
 [ 0.         0.         0.         1.05304383]]

Wartosci Wlasne Macierzy Hessenberga

[7.230992309419283, 5.900157268333127, 4.815806590682049, 1.053043831565537]
```

Widać że macierz  $M$  jest postaci macierzy trójkątnej górnej, dzięki temu możemy odczytać wszystkie wartości własne i określić największą z nich

### Podpunkt B

Metoda potęgowa polega na wyznaczaniu największej wartości własnej i wektora własnego dla macierzy. W tym celu wykorzystuje się iteracyjny proces, w którym każdy nowy wektor jest mnożony przez macierz oraz wynik jest normowany aby wektor wynikowy nie „ucieł” do nieskończoności. Wartość własna jest określona jako stosunek skalarny między wektorem własnym i jego wielokrotnością po działaniu macierzy. Metoda potęgowa jest szybka i łatwa w implementacji – wymaga jedynie mnożenia macierzy z wektorem. Jednak ze względu na występujące w niej zbieżności i oscylacje, wyniki mogą być niedokładne. Co więcej, metoda nie zapewnia dokładnych wartości własnych, ale zazwyczaj zbliżone. Aby uzyskać dokładniejsze wyniki, można wykorzystać inne metody, takie jak metoda Jacobiego lub metoda  $QR$ .

Dodatkowo do normalizowania wektora, można użyć kilku norm dzięki temu można uzyskać „różne” wyniki natomiast stosunek wartości wektora własnego w porównaniu z każdym innym uzyskanym za pomocą innej normy będzie taki sam, postanowiłem to sprawdzić i uzyskałem następujące wyniki:

*Normalizowanie wynikowego wektora za każdym razem używając normy euklidesowej*

```
Podaj Ilość iteracji ( wystarczy ok. 25 ): 25
Największa wartość własna: 10.015982557921658
Wektor własny
[[0.55839647]
 [0.77608564]
 [0.28688842]
 [0.05982886]]
```

*Normalizowanie wynikowego wektora za każdym razem używając normy maximum*

```
Podaj Ilość iteracji ( wystarczy ok. 25 ): 25
Największa wartość własna: 10.016556317605096
Wektor własny
[[0.71950367]
 [1.         ]
 [0.36966078]
 [0.07709054]]
```

Widać że wektory własne różnią się natomiast stosunek każdej składowej w porównaniu tych dwóch wektorów będzie taki sam

$$x = (0.55839647, \quad 0.77608564, \quad 0.28688842, \quad 0.05982886)$$

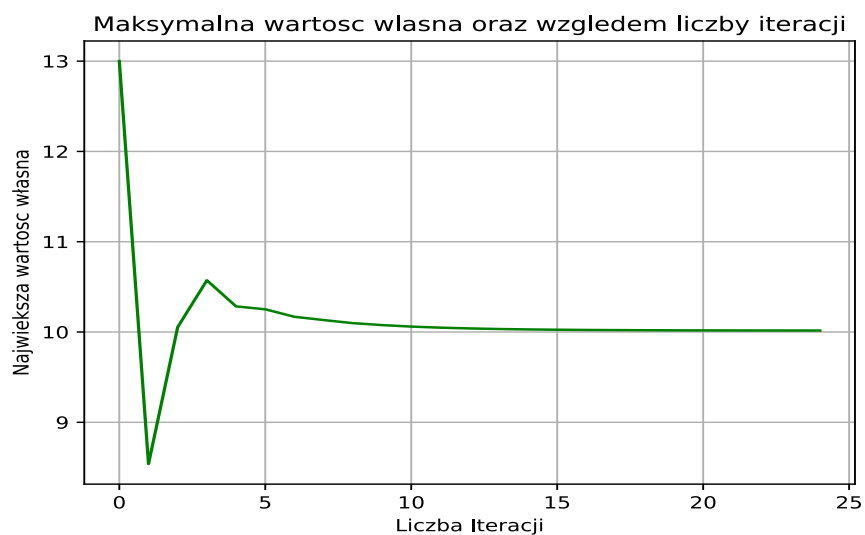
$$y = (0.71950367, \quad 1, \quad 0.36966078, \quad 0.07709054)$$

$$\frac{x_{11}}{y_{11}} = \frac{x_{22}}{y_{22}} = \dots = \frac{x_{nn}}{y_{nn}} = 0.77608564$$

Widać że jeżeli wybierzemy różne normy do normalizowania, wynikowego wektora w każdej iteracji i tak stosunek tych wektorów będzie wynosić tyle samo, tak więc nie ma to znaczenia jaka normę wybierzemy, i tak największa wartość własna zawsze będzie taka sama( zmieniać się będzie tylko wektor własny )

Dodatkowo możemy zauważyć że do otrzymania wyniku z satysfakcjonującą dokładności w metodzie potęgowej potrzebujemy znacznie mniej iteracji dla tej macierzy 4x4 po około 25 iteracjach otrzymujemy wynik z dokładnością do  $1e - 15$

Postanowiłem zrobić wykres który obrazuje zbieganie największej wartości własnej , Wyniki prezentują się w następujący sposób



Można zauważyć że tak naprawdę „porządny wynik” otrzymujemy już po około 10 iteracjach natomiast jeżeli chcemy otrzymać wynik z dokładnością rzędu  $1e-15$  musimy tych iteracji wykonać więcej