Sprawozdanie MNUM Projekt 02

Autor: **TOMASZ SACHANOWSKI**

Grupa: **czwartek 8-10**

Nr. Indexu: **276467**

Nr. Zadania**: 2.55**

Spis treści

[Treść zadań 2](#_Toc37434348)

[Zadanie 1 3](#_Toc37434349)

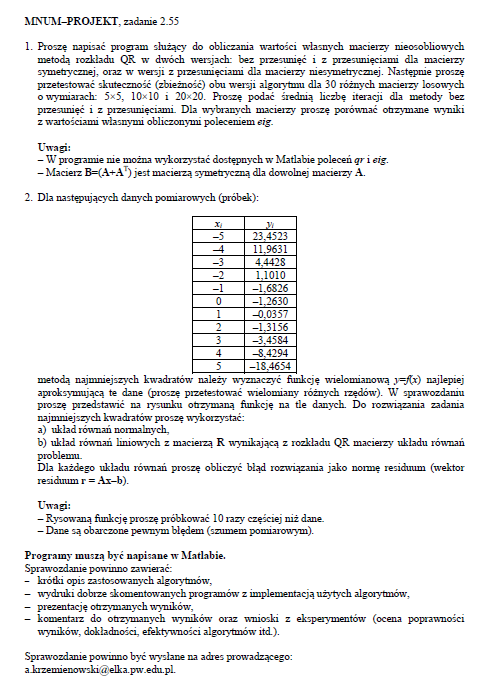
[Cel: 3](#_Toc37434350)

[Teoria: 3](#_Toc37434351)

[Rozwiązanie: 7](#_Toc37434352)

[Podsumowanie: 9](#_Toc37434353)

# Treść zadań



# Zadanie 1

## Cel:

Celem jest napisanie programu do obliczenia wartości własnych macierzy nieosobliwych metodą rozkładu QR w wersjach z przesunięciem i bez przesunięcia dla macierzy symetrycznych oraz w wersji z przesunięciem dla macierzy niesymetrycznej.

## Teoria:

Zdefiniowanie pojęć:

* macierz ortogonalna –jest to taka macierz Q, która: 𝐐∗= 𝐈 (jej kolumny są wektorami ortonormalnymi, I–macierz jednostkowa).
* macierz ortonormalne –macierz ortogonalna oraz długości jednostkowej.

Rozkład macierzy ***A*** do postaci iloczynu dwóch macierzy ***Q*** i ***R***, gdzie ***Q*** jest macierzą ortonormalną (lub ogólniej ortogonalną), a ***R*** jest macierzą trójkątną górną.

Możemy rozłożyć dowolną macierzy na iloczyn macierzy ***Q*** i ***R***.

Sposoby numeryczne rozkładu QR:

* metoda ortogonalizacji Grama-Schmidta (ew. zmodyfikowany algorytm Grama-Schmidta).
* metoda odbić Householdera.
* metoda obrotów Givensa (szczególnie macierze rzadkie tj.: macierz w której większość elementów ma wartość 0).

Wartości i wektory własne macierzy kwadratowej rzeczywistej 𝐴 są to takie liczby λ i odpowiadające im wektory 𝑣, że

***𝐴𝑣= 𝜆𝑣,***

gdzie λ – wartość własna, 𝑣 – odpowiadający jej wektor własny. Wartości i wektory własne spełniają więc równanie:

***(𝐴− 𝜆 𝐼) 𝑣 =0.***

Macierz kwadratowa ***𝑛***-wymiarowa ma dokładnie 𝑛 wartości własnych i odpowiadających im wartości własnych. Zbiór wszystkich wartości własnych macierzy nazywany jest widmem macierzy.

Wartości własne odgrywają ważną rolę w wielu dziedzinach nauki i techniki.

Z faktu, iż wektorem własnych jest każdy wektor pomnożony przez pewną stałą α, wprowadza się pojęcie wektorów unormowanych. Są to takie wektory własne, których długość jest równa 1.

Wybrane metody wyznaczania wartości własnych

* metoda QR (wykorzystana przeze mnie w tym zadaniu, opisana w następnym punkcie)
* metoda Jacobiego:

Służy ona do wyznaczania wartości własnych tylko macierzy symetrycznej ***A*** polegająca na przekształceniu macierzy do postaci diagonalnej ***P*** ciągiem obrotów Givensa. W macierzy diagonalnej na przekątnej głównej znajdą się wartości własne macierzy ***A***, natomiast wektory własne odpowiadające tym wartościom własnym będą zapisane w kolumnach macierzy P.

* metoda wyznacznikowa:

Metoda korzysta z faktu, iż wartości własne są zerami wielomianu charakterystycznego obliczając wartości własne wprost z definicji ***det(A− λI) = 0.***

* metoda QR – najbardziej ogólna, efektywna

Rozkład QR

Każdą macierz rzeczywistą (𝑚≥𝑛) o liniowo niezależnych kolumnach można przedstawić w postaci:

**=**

gdzie jest macierzą o kolumnach ortonormalnych, a jest macierzą trójkątną górną z dodatnimi elementami na diagonali.

algorytm podstawowy bez przesunięć:

Pojedynczy krok algorytmu ***QR*** (w k+i kroku):

***=*** (𝑓𝑎𝑘𝑡𝑜𝑟𝑦𝑧𝑎𝑐𝑗𝑎)

***=***

jest ortogonalna, więc:

***= =***

skąd mamy

**=**

Widzimy, że macierz jest przekształconą przez podobieństwo macierzą , więc ma te same wartości własne.

Czyli macierz jest przekształconą przez podobieństwo macierzą . Mają te same wartości własne.

Podsumowując dla macierzy symetrycznej ***A***, zbiega do macierzy diagonalnej.

Metoda QR bez przesunięć – algorytm podstawowy:

***=*** (𝑓𝑎𝑘𝑡𝑜𝑟𝑦𝑧𝑎𝑐𝑗𝑎)

***= =* =**

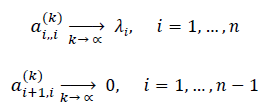
***= =* =**

***→ 𝐴𝑉=𝑑𝑖𝑎𝑔{}***

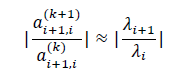
Dla macierzy ***𝐴*** symetrycznej macierz zbiega do macierzy diagonalnej ***𝑑𝑖𝑎𝑔{}.***Jeśli macierz 𝐴 ma 𝑛 wartości o własnych o różnych modułach,



to można pokazać, że:



i zbieżność elementu poddiagonalnego do zera jest liniowa z ilorazem zbieżności |𝜆𝑖+1𝜆𝑖|, czyli

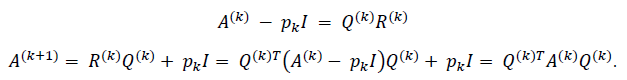


Stąd jeśli wartości własne leżą blisko siebie, to metoda jest wolno zbieżna. Aby poprawić szybkość zbieżności stosuje się algorytm metody QR z przesunięciami.

Algorytm z przesunięciami:

Gdy wartości własne λ leżą blisko siebie to metoda z algorytmem podstawowym jest wolno zbieżna. Wówczas stosuje się algorytm metody ***QR*** z przesunięciami (w celu poprawienia szybkości zbieżności).

Pojedynczy krok algorytmu QR z przesunięciami (w k+i kroku):



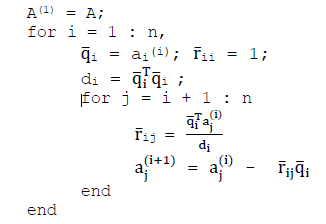
Zbieżność jest wtedy liniowa z ilorazem. Wynika z tego, że najlepszym przesunięciem byłoby aktualne przybliżenie.

Metoda ***QR*** wyznaczania wartości własnych dla macierzy niesymetrycznej

Podobnie jak w macierzy symetrycznej macierz wyjściową zaleca się przekształcić do postaci Hessenberga. Konstrukcja algorytmu QR dla macierzy dowolnych pozostaje w istocie taka sama jak dla macierzy symetrycznych, musimy wziąć pod uwagę możliwość wystąpienia wartości własnych zespolonych. Dlatego ważne jest wybieranie przesunięcia zawsze jako wartości własnej podmacierzy 2x2 z prawego dolnego rogu aktualnej macierzy. Macierz przekształcona zbiega na ogół do macierzy trójkątnej górnej. Metoda QR dla macierzy niesymetrycznych nie zawsze jest zbieżna.

## Rozwiązanie:

Program został podzielony na kilka funkcji. Kod znajdujący się w ***Generate\_symetric\_matrix*** i ***Generate\_symetric\_matrix*** generuje nam odpowiedniej wielkości macierze. Następnie funkcja ***Decompose\_QR*** rozkłada macierze na dwie Q i R. Został tu wykorzystany algorytm z książki. Jest to zmodyfikowany algorytm Grama-Schmidta, który działa następująco:



Zmodyfikowany algorytm w innej kolejności przeprowadza proces ortogonalizacji i ma znacznie lepsze własności numeryczne w porównaniu z algorytmem standardowym.

Standardowy algorytm ortogonalizuje kolumny macierzy po kolei, natomiast algorytm zmodyfikowany po wyznaczeniu kolejnej kolumny ortogonalnej od razu ortogonalizuje wobec niej wszystkie następne.

Kolejnymi krokami są funkcje wyznaczające wartości własne w macierzach są to odpowiednio: ***EigvalQRNoShift, EigvalQRShift.*** Na samym końcu działa funkcja ***compose\_task\_1*** pozwalająca przeprowadzić analizę dla kolejnych rozmiarów macierzy jak i ilości iteracji i błędów.

Wynik:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ŚREDNIA LICZBA ITERACJI** | | | | | | |
|  | *err 0.0001,*  *maxit = 1000* | | *err 0.00001,*  *maxit = 1000* | | *err 0.000001,*  *maxit = 1000* | |
| **ALG. Z PRZESUNIECIEM** | **NIE** | **TAK** | **NIE** | **TAK** | **NIE** | **TAK** |
| Macierz symetryczna 5x5 | 72.9000 | 7.2000 | 42.5172 | 7.5333 | 65.3333 | 8.0333 |
| Macierz symetryczna 10x10 | 187.3571 | 13.3667 | 183.0741 | 14.3333 | 225.5185 | 14.4667 |
| Macierz symetryczna 20x20 | 408.5000 | 26.7667 | 509.6500 | 28.1000 | 531.4444 | 29.7667 |
| Macierz asymetryczna 5x5 |  | 9.3000 |  | 9.9333 |  | 9.9333 |
| Macierz asymetryczna 10x10 |  | 19.2667 |  | 21.5000 |  | 22.6000 |
| Macierz asymetryczna 20x20 |  | 41.2333 |  | 45.8333 |  | 47.4667 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PROCENT UDANCYH PRÓB** | | | | | | |
|  | *err 0.0001,*  *maxit = 1000* | | *err 0.00001,*  *maxit = 1000* | | *err 0.000001,*  *maxit = 1000* | |
| **ALG. Z PRZESUNIECIEM** | **NIE** | **TAK** | **NIE** | **TAK** | **NIE** | **TAK** |
| Macierz symetryczna 5x5 | 1.0000 | 1.0000 | 0.9667 | 1.0000 | 1.0000 | 1.0000 |
| Macierz symetryczna 10x10 | 0.9333 | 1.0000 | 0.9000 | 1.0000 | 0.9000 | 1.0000 |
| Macierz symetryczna 20x20 | 0.6667 | 1.0000 | 0.6667 | 1.0000 | 0.6000 | 1.0000 |
| Macierz asymetryczna 5x5 |  | 1.0000 |  | 1.0000 |  | 1.0000 |
| Macierz asymetryczna 10x10 |  | 1.0000 |  | 1.0000 |  | 1.0000 |
| Macierz asymetryczna 20x20 |  | 1.0000 |  | 1.0000 |  | 1.0000 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ŚREDNI BŁĄD WZGLĘDEM EIG** | | | | | | |
|  | *err 0.0001,*  *maxit = 1000* | | *err 0.00001,*  *maxit = 1000* | | *err 0.000001,*  *maxit = 1000* | |
| **ALG. Z PRZESUNIECIEM** | **NIE** | **TAK** | **NIE** | **TAK** | **NIE** | **TAK** |
| Macierz symetryczna 5x5 | 0.1015 | 0.0000 | 0.1523 | 0.0000 | 0.0812 | 0.0000 |
| Macierz symetryczna 10x10 | 0.1199 | 0.0000 | 0.1026 | 0.0000 | 0.1309 | 0.0000 |
| Macierz symetryczna 20x20 | 0.0648 | 0.0000 | 0.0891 | 0.0000 | 0.1621 | 0.0000 |
| Macierz asymetryczna 5x5 |  | 0.5645 |  | 0.5029 |  | 0.5746 |
| Macierz asymetryczna 10x10 |  | 1.3914 |  | 1.3731 |  | 1.6333 |
| Macierz asymetryczna 20x20 |  | 3.8106 |  | 3.8004 |  | 3.5644 |

## 

## Podsumowanie:

Wniosek dotyczący wydajności danych algorytmów, średniej ilości iteracji:

Istotnym wnioskiem, że algorytm z przesunięciami jest znacznie wydajniejszy od algorytmu bez przesunięć.

Ponadto, działa on dla macierzy symetrycznych jak i niesymetrycznych. Dla obu macierzy symetrycznej jest zdecydowanie szybciej zbieżny –liczba iteracji jest parę razy mniejsza(jak nie paręnaście w niektórych przypadkach).

Warto zauważyć, że algorytm bez przesunięć przy danej liczbie iteracji maksymalnych nie radził sobie ze wszystkimi macierzami.

Wniosek podsumowujący algorytm z przesunięciami i wewnętrznie wbudowany w Matlaba:

Na podstawie porównania wyników uzyskanych za pomocą algorytmu z przesunięciami i funkcji wewnętrznej eig możemy zauważyć, że wartości własne macierzy są właściwie takie same. Ponadto algorytm z przesunięciem radzi sobie dużo lepej.