Raport Zadanie NUM6

Tomasz Dziób

17.12.2023

1 Wstęp techniczny

Poniższy raport dotyczy zadania numerycznego NUM6 z listy 5. Załączone programy o nazwie *num6_ a.cpp* i *num6_ b.cpp* zostały napisane w języku *C++*, oraz korzystają z bibiotek *GNU Plot* oraz *GSL*. Do sprawdzenia obliczeń zostały użyte pliki *check_ a.cpp* i *check_ b.cpp* korzystające z biblioteki *GSL*. Przed skompilowaniem programu należy zainstalować powyższe biblioteki.

1.1 Jak uruchomić program?

Razem z załączonymi plikami znajdziemy *Makefile* który służy do uruchomienia programów.

- *make runA* uruchamia program dotyczący podpunktu a wraz ze sprawdzeniem
- *make runB* uruchamia program dotyczący podpunktu b wraz ze sprawdzeniem

2 Nakreślenie problemu

a) Algorytm QR opiera się na rozkładzie QR macierzy oraz służy do znajdowania wartości własnych oraz wektorów własnych. Rozkład QR macierzy polega na sprowadzeniu macierzy A do postaci iloczynu dwóch macierzy gdzie **Q** jest macierzą ortogonalną a **R** jest macierzą trójkątną górną.

$$\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \ddots & \ddots & \ddots & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \ddots & \ddots & \ddots & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & \ddots & \ddots & \ddots & \bullet \\ 0 & 0 & 0 & \bullet & \bullet \\ 0 & 0 & 0 & 0 & \bullet \end{bmatrix}$$

$Q^T Q = \mathbb{1}$

$Q^{-1} = Q^T$

$detQ = \pm 1$

Po wykonaniu algorytmu otrzymujemy macierz z wartościami własnymi na diagonali. Rozwiązując równanie $(A - \lambda\mathbb{1})\vec{x} = 0$ dla każdej z wartości, otrzymujemy wektory własne.

b) W tym przypadku została użyta metoda potęgowa która służy do odnalezienia największej wartości własnej wraz z odpowiadającym jej wektorem. Jej zaletą jest to, że pozwala wyliczyć to bez marnowania mocy obliczeniowej na znalezienie pozostałych wartości oraz wektorów.

Plusem obu metod jest iteracyjna natura która pozwala na wyliczenie poszukiwanych wartości do zadanej dokładności co wiąże się z adekwatnie mniejszym czasem wykonania programu.

3 Użyta metoda

a) *Algorytm QR* polega dokonaniu rozkładu macierzy A_1 na kolejno Q_1 oraz R_1. Następnie w kolejnych iteracjach macierz A_2 konstruujemy z przemnożenia R_1 przez Q_1. Kontynuujemy to do uzyskania satysfakcjonującej dokładności wartości własnych, które znajdują się na diagonali A_n.

$$\begin{aligned} A_1 &= Q_1 R_1 \\ A_2 = R_1 Q_1 &= Q_2 R_2 \\ A_3 = R_2 Q_2 &= Q_3 R_3 \\ &\vdots \\ A_{n-1} = R_{n-2} Q_{n-2} &= Q_{n-1} R_{n-1} \\ A_n &= Q_{n-1} R_{n-1} \end{aligned}$$

b) W *metodzie potęgowej*, dla macierzy A, szukamy jej największej wartości własnej oraz odpowiadającego jej wektora. aby to uzyskać postępujemy tak:

$$A\vec{v_k} = \vec{x_k}$$
$$\vec{v}_{k+1} = \frac{\vec{x}_k}{||\vec{x}_k||}$$

1. Wybieramy losowy, niezerowy(!) wektor $\vec{v}$ oraz obliczamy $A\vec{v} = \vec{x}$.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Jest to nasze przybliżenie wartości własnej w danej iteracji.

2. Normalizujemy wektor $\vec{x}$ przez jego normę w celu uniknięcia mnożenia dużych liczb oraz wyjścia z zakresu używanego typu.

$$\begin{bmatrix} \frac{x_1}{||\vec{x}||} \\ \frac{x_2}{||\vec{x}||} \\ \frac{x_3}{||\vec{x}||} \end{bmatrix}$$

3. Za nasz wektor $\vec{v}$ przyjmujemy teraz znormalizowany wektor $\vec{x}$ i powtarzamy cały algorytm do uzyskania zadowalającego przybliżenia.

Naszą największą wartością własną jest norma wektora $||\vec{x}||$, a wektor własny jej odpowiadający to $\vec{v}$.

4 Uzyskany wynik

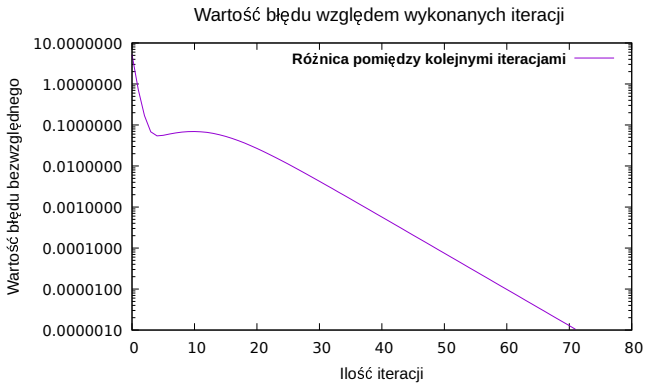
a) Wynik wykonania komendy *make runA*:

```
num6_a.cpp
Wartosci wlasne:
1.05304 4.81581 5.90016 7.23099

check_a.cpp
Wartosci wlasne:
1.05304 4.81581 5.90016 7.23099
```

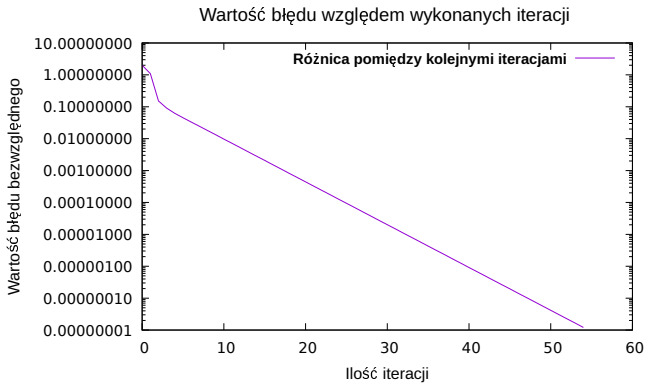
b) Wynik wykonania komendy *make runB*:

```
num6_b.cpp
Najwieksza wartosc wlasna:
10.016
Odpowiadajacy wektor wlasny:
0.558297 0.776208 0.286788 0.0596481

check_b.cpp
Wartosci wlasne:
1.44264 -3.81179 7.35317 10.016
Wektory wlasne:
-0.406417       -0.613835       0.382539        0.558297
-0.00966313     0.418661        -0.471307       0.776208
0.873226        -0.0764527      0.386502        0.286788
-0.268715       0.664894        0.69437         0.0596481
```

Przedstawiając ten wynik na wykresie wyglądałoby to tak jak poniżej. Wartośc błędu bezwględnego wyświetlana jest w obu przypadkach w skali logarytmicznej. Dla podpunktu a) jest to moduł różnicy pomiędzy pierwszą wartością własną uzyskaną w tej iteracji, a tej z poprzedniej. Dla podpunktu b) jest to norma wektora $\vec{v}_{k-1} - \vec{v}_k$.



Rysunek 1: Wykres dla podpunktu a)
dokładność = 10^{-6}



Rysunek 2: Wykres dla podpunktu b)
dokładność = 10^{-8}

5 Podsumowanie

Algorytm QR oraz *Metoda Potęgowa* przydają się najbardziej gdy nie potrzebujemy pełnej dokładności rozwiązania. Jeśli wystarcza nam przybliżone rozwiązanie, metody te mogą być bardziej efektywne, ponieważ mogą zatrzymać się po osiągnięciu pewnego akceptowalnego poziomu dokładności, co może przyspieszyć obliczenia. Jednak obywa rozwiązania mają swoje plusy i minusy.

Dodatkową zaletą algorytmu QR jest to, że zachowuje symetrię, postać trójdiagonalną symetryczną i postać Hessenberga macierzy.

Natomiast jeżeli przy pomocy metody potęgowej trzeba znaleźć więcej niż kilka wartości wektorów własnych, metoda, z uwagi na konieczność reortogonalizacji, staje się bardzo kosztowna. Jeżeli w widmie macierzy pojawiają się wartości własne bardzo zbliżone co do modułu — na przykład 2 oraz 1.99999999, ale też 1 oraz 0.99999999 — zbieżność jest bardzo wolna, a nawet można spodziewać się stagnacji. Nie działa też ona dla macierzy niesymetrycznych.