Zadanie domowe 3

Aleksander Profic

25. listopada 2018

Główne źródła w internecie oraz literatura, z których korzystałem:
- Wykład dr Pawła Góry z 2017 roku, ponieważ w tym roku jeszcze nie było Metody Potęgowej na wykładzie (ma być dopiero w styczniu)
- William H. Press Saul A. Teukolsky William T. Vetterling Brian P. Flannery "Numerical Recipes in C: The Art of Scientific Computing" Second Edition

Zadanie 1. Program obliczający wiodącą i pod wiodącą wartość własną oraz 2 odpowiadające im wektory własne macierzy hermitowskiej:

$$A = \frac{1}{12}\begin{bmatrix} 19 & 13 & 10 & 10 & 13 & -17 \\ 13 & 13 & 10 & 10 & -11 & 13 \\ 10 & 10 & 10 & -2 & 10 & 10 \\ 10 & 10 & -2 & 10 & 10 & 10 \\ 13 & -11 & 10 & 10 & 13 & 13 \\ -17 & 13 & 10 & 10 & 13 & 19 \end{bmatrix}$$

- Do rozwiązania zadania użyłem metody potęgowej, która pozwala obliczyć wartość własną o największym module i odpowiadający jej wektor własny.
- Metoda działa bez zakłóceń, jeśli macierz posiada następujące własności:
    1. Macierz jest symetryczna.
    2. Tylko jedna jej wartość własna (rzeczywista lub zespolona) ma największy moduł:

    $$|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \ldots \geq |\lambda_n| > 0$$

    3. Istnieje układ n wektorów własnych liniowo niezależnych, czyli macierz ma prostą strukturę.

Niech $A \in \mathbb{R}^{NxN}$ będzie macierzą symetryczną, $A = A^T$.
Wiadomo, że macierz taka jest diagonalizowalna, ma rzeczywiste wartości własne, a jej unormowane wektory własne tworzą bazę ortonormalną w $\mathbb{R}^N$. Niech bazą tą będzie $\{e_i\}_{i=1}^N$, przy czym $Ae_i = \lambda_i e_i$. Weźmy wektor $y \in \mathbb{R}^N$. Posiada on rozkład:

$$y = \sum_{i=1}^{N} \beta_i e_i$$

Wtedy:

$$Ay = A\sum_{i=1}^{N}\beta_i e_i = \sum_{i=1}^{N}\beta_i A e_i = \sum_{i=1}^{N}\beta_i\lambda_i e_i$$

$$A^k y = \sum_{i=1}^{N}\beta_i\lambda_i^k e_i$$

Widzimy, że dla dostatecznie dużych k wyraz zawierający λ_1^k będzie dominował w sumie po prawej stronie, pozostałe współczynniki będą zaniedbywalnie małe, a zatem prawa strona dla dostatecznie dużych k będzie dążyć do wektora proporcjonalnego do e_1, czyli do wektora własnego do największej wartości własnej.

A zatem iteracja ($\|y_1\| = 1$, poza tym może być dowolny)

$$Ay_k = z_k$$

$$y_{k+1} = \frac{z_k}{\|z_k\|}$$

zbiega się, przy przyjętych założeniach, do unormowanego wektora własnego macierzy A, odpowiadającego największej wartości własnej.
Gdy iteracja zbiegnie się do punktu stałego (kolejne wektory $y_k \simeq e_1$ przestaną się zauważalnie zmieniać), wartość własną obliczamy jako $\lambda_1 = \|z_k\|$.

Kiedy w sumie $A^k y = \sum_{i=1}^{N}\beta_i\lambda_i^k e_i$ wyraz z wektorem własnym do największej wartości własnej nie będzie dominować? Wtedy i tylko wtedy, gdy współczynnik odpowiadający temu wektorowi będzie znikać, $\beta_1 = 0$. To sugeruje sposób na szukanie wektora własnego odpowiadającego drugiej co do wielkości wartości własnej: należy upewnić się, że wektor y_1 i wszystkie jego iteraty są prostopadłe do uprzednio znalezionego e_1.

Iteracja ($\|y_1\| = 1, e_1^T y_1 = 0$, poza tym dowolny)

$$Ay_k = z_k$$

$$z_k = z_k - e_1(e_1^T z_k)$$

$$y_{k+1} = \frac{z_k}{\|z_k\|}$$

zbiega się, przy przyjętych założeniach, do unormowanego wektora własnego macierzy A, odpowiadającego drugiej co do wielkości wartości własnej. Drugi krok powyższego algorytmu oznacza ortogonalizację.

Wyniki z mojego programu:

```
1. fish  /Users/profitz/Documents/0STUDIA/3 semestr/Zestawy programowanie/Metody numeryczne/Z3 (fish)
profitz@Aleksanders-MacBook-Pro ~/D/0/3/Z/M/Z3> ./powerIteration.x
Macierz A:
19.0000 13.0000 10.0000 10.0000 13.0000 -17.000
13.0000 13.0000 10.0000 10.0000 -11.000 13.0000
10.0000 10.0000 10.0000 -2.0000 10.0000 10.0000
10.0000 10.0000 -2.0000 10.0000 10.0000 10.0000
13.0000 -11.000 10.0000 10.0000 13.0000 13.0000
-17.000 13.0000 10.0000 10.0000 13.0000 19.0000

Wiodąca wartość własna -> Lambda1:
4.0000000000000000000
Wiodący wektor własny -> v1:
[ 0.408248290463861019 0.408248290463863017 0.408248290463863017 0.408248290463863017 0.408248290463863017 0.408248290463865071 ]
Pod wiodąca wartość własna -> Lambda2:
2.9999999999999999556
Pod wiodący wektor własny -> v2:
[ 0.707106781186548905 0.000000000000001007 0.000000000000001155 0.000000000000001155 0.000000000000001081 -0.707106781186546240 ]
profitz@Aleksanders-MacBook-Pro ~/D/0/3/Z/M/Z3>
```

Wyniki z programu Mathematica:

```
In[4]:= Simplify[
          Eigenvalues[1 / 12 * {{19, 13, 10, 10, 13, -17}, {13, 13, 10, 10, -11, 13}, {10, 10, 10, -2, 10, 10}, {10, 10, -2, 10, 10, 10},
            {13, -11, 10, 10, 13, 13}, {-17, 13, 10, 10, 13, 19}}]]
Out[4]= {4, 3, -2, 2, -1, 1}

In[12]:= Eigenvectors[
          1 / 12 * {{19, 13, 10, 10, 13, -17}, {13, 13, 10, 10, -11, 13}, {10, 10, 10, -2, 10, 10}, {10, 10, -2, 10, 10, 10},
            {13, -11, 10, 10, 13, 13}, {-17, 13, 10, 10, 13, 19}}]
Out[12]= {{1, 1, 1, 1, 1, 1}, {-1, 0, 0, 0, 0, 1}, {1, -1, 0, 0, -1, 1}, {0, -1, 0, 0, 1, 0}, {1, 1, -2, -2, 1, 1}, {0, 0, -1, 1, 0, 0}}

In[19]:= Simplify[Normalize[{1, 1, 1, 1, 1, 1}]]
```

Out[19]= $\left\{\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right\}$

```
In[20]:= Simplify[Normalize[{-1, 0, 0, 0, 0, 1}]]
```

Out[20]= $\left\{-\frac{1}{\sqrt{2}}, 0, 0, 0, 0, \frac{1}{\sqrt{2}}\right\}$