

WPROWADZENIE

Rozkład QR jest uniwersalną metodą pozwalającą rozwiązywać zagadnienie własne dla danej macierzy. Przypomnijmy, że liczbę $\lambda \in \mathbb{C}$ nazywamy *wartością własną* macierzy $A \in \mathbb{C}^{n \times n}$ jeżeli istnieje niezerowy wektor v taki, że zachodzi równość

$$Av = \lambda v. \quad (1)$$

Wektor v nazywamy wtedy wektorem własnym. Zauważmy, że jeżeli v jest wektorem własnym, to jest nim również wektor cv dla dowolnej liczby zespolonej $c \neq 0$.

Definicja. Macierz kwadratową $A \in \mathbb{K}^{n \times n}$ (gdzie $\mathbb{K} = \mathbb{R}$ lub $\mathbb{C}$) nazywamy diagonalizowalną jeżeli istnieje macierz nieosobliwa $P \in \mathbb{K}^{n \times n}$ taka, że $P^{-1}AP$ jest macierzą diagonalną. Formalnie,

$$A \in \mathbb{K}^{n \times n} \text{ jest diagonalizowalna} \Leftrightarrow \exists P \in \mathbb{K}^{n \times n}, \det P \neq 0 \text{ i } P^{-1}AP \text{ jest diagonalna} \quad (2)$$

Fakt 1. Wektory własne odpowiadające różnym wartościom własnym tworzą układ liniowo niezależny.

Zatem jeżeli $v_1, v_2, \dots, v_k$ są wektorami własnymi macierzy A odpowiadającymi różnym wartościom własnym $\lambda_1, \lambda_2, \dots, \lambda_k$, to wtedy wektory $v_1, v_2, \dots, v_k$ są liniowo niezależne. Liniowa niezależność wektorów oznacza, że

$$\forall \alpha_1, \dots, \alpha_k \in \mathbb{C}: \sum_{j=1}^k \alpha_j v_j = 0 \Rightarrow \alpha_1 = \dots = \alpha_k = 0 \quad (3)$$

Dowód jest indukcyjny względem $k \in \mathbb{N}$.

(i) Dla $k=1$ jest oczywiste, bo jeżeli v_1 jest wektorem własnym, to z definicji jest niezerowy. A układ jednoelementowy $\{v_1\}$ jest liniowo niezależny $\Leftrightarrow v_1 \neq 0$.

(ii) Krok indukcyjny $k \rightarrow k+1$:

$$\begin{aligned} \alpha_1 v_1 + \dots + \alpha_{k+1} v_{k+1} &= 0 \Rightarrow A(\alpha_1 v_1 + \dots + \alpha_{k+1} v_{k+1}) = A(0) = 0 \\ \alpha_1 Av_1 + \dots + \alpha_{k+1} Av_{k+1} &= 0 \Rightarrow \alpha_1 \lambda_1 v_1 + \dots + \alpha_{k+1} \lambda_{k+1} v_{k+1} = 0 \end{aligned}$$

Mnożymy teraz pierwszą równość przez λ_{k+1} i wynik odejmujemy od ostatniej równości

$$\left. \begin{aligned} \lambda_{k+1} \alpha_1 v_1 + \dots + \lambda_{k+1} \alpha_{k+1} v_{k+1} &= 0 \\ \alpha_1 \lambda_1 v_1 + \dots + \alpha_{k+1} \lambda_{k+1} v_{k+1} &= 0 \end{aligned} \right\} \Rightarrow \alpha_1 (\lambda_1 - \lambda_{k+1}) v_1 + \dots + \alpha_k (\lambda_k - \lambda_{k+1}) v_k = 0$$

Z założenia indukcyjnego mamy teraz: $\alpha_1 (\lambda_1 - \lambda_{k+1}) = 0$, $\alpha_2 (\lambda_2 - \lambda_{k+1}) = 0$, ..., $\alpha_k (\lambda_k - \lambda_{k+1}) = 0$. Na mocy założenia wartości własne są parami różne, czyli $\lambda_1 - \lambda_{k+1} \neq 0, \dots, \lambda_k - \lambda_{k+1} \neq 0$ więc z poprzednich równości otrzymujemy $\alpha_1 = 0, \dots, \alpha_k = 0$. Tak więc $\alpha_{k+1} v_{k+1} = 0$. Ale wektor własny v_{k+1} jest z definicji różny od zera, zatem $\alpha_{k+1} v_{k+1} = 0 \Rightarrow \alpha_{k+1} = 0$. Pokazaliśmy, że wszystkie współczynniki $\alpha_1, \dots, \alpha_{k+1}$ muszą być równe zeru, co kończy dowód.

Fakt 2. Wartości własne są pierwiastkami wielomianu charakterystycznego, który można otrzymać przy pomocy wyznacznika:

$$\lambda \text{ jest wartością własną macierzy } A \Leftrightarrow \det[A - \lambda I] = 0. \quad (4)$$

Nawet jeżeli macierz A jest rzeczywista, to jej wartości własne mogą być zespolone. Wtedy też odpowiednie wektory własne będą miały składowe zespolone. Natomiast w jednym bardzo ważnym przypadku wartości własne będą zawsze rzeczywiste: gdy macierz A jest symetryczna.

Fakt 3. Jeżeli A jest macierzą symetryczną, tzn. $A = A^T$, to wszystkie jej wartości własne są rzeczywiste, a w konsekwencji także wektory własne są rzeczywiste.

Definicja. Mówimy, że macierze $A, B \in \mathbb{C}^{n \times n}$ są *podobne*, co zapisujemy $A \sim B$, jeżeli istnieje nieosobliwa macierz P , taka że

$$B = P^{-1}AP. \quad (5)$$

Zauważmy, że jest to relacja symetryczna, gdyż $B = P^{-1}AP \Rightarrow A = PBP^{-1}$.

Fakt 4. Macierze podobne mają dokładnie takie same wartości własne.

Wynika to z prostego rozumowania: jeżeli λ jest wartością własną macierzy B , to istnieje wektor v taki, że $Bv = \lambda v$, zatem $P^{-1}APv = \lambda v$, czyli $A(Pv) = \lambda Pv$. Oznacza to, że dla $v' = Pv$ zachodzi

$$Av' = \lambda v'$$

gdzie $v' = Pv$.

Fakt 5. Jeżeli macierz $A \in \mathbb{C}^{n \times n}$ posiada bazę wektorów własnych $v_1, \dots, v_n$, to jest diagonalizowalna przez macierz $P = [v_1 \ v_2 \ \dots \ v_n]$ (kolumnami macierzy P są wektory $v_1, \dots, v_n$). Zachodzi bowiem

$$P^{-1}AP = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n \end{bmatrix}. \quad (6)$$

Wniosek: jeżeli macierz $A \in \mathbb{C}^{n \times n}$ ma n różnych wartości własnych, to jest diagonalizowalna.

Wynika to stąd, że wektory własne odpowiadające różnym wartościom własnym są liniowo niezależne (**Fakt 1**). Zatem gdy macierz ma n różnych wartości własnych, to posiada n liniowo niezależnych wektorów własnych. Ponieważ wymiar przestrzeni $\mathbb{C}^n$ wynosi n , to wektory te są bazą. Tak więc wniosek wynika teraz z **Faktu 5**.

ROZKŁAD QR

Celem jest dokonanie rozkładu w formie faktoryzacji danej macierzy kwadratowej A na iloczyn dwóch macierzy

$$A = QR \quad (7)$$

gdzie Q jest macierzą unitarną, a R jest macierzą trójkątną górną. Z rozkładu (7) mamy możliwość szybkiego rozwiązania układu równań $Ax = b$, gdyż

$$Ax = b \Leftrightarrow QRx = b \Leftrightarrow Rx = Q^{-1}b \Leftrightarrow Rx = Q^*b$$

czyli problem sprowadzony został do układu równań z macierzą trójkątną. Prawą stronę też szybko obliczamy, gdyż Q^* oznacza tylko transponowanie macierzy Q (i ewentualnie wzięcie sprzężenia zespolonego dla macierzy o współczynnikach zespolonych).

Można wykazać następujące twierdzenie:

Dla dowolnej macierzy A istnieje taka macierz unitarna Q , że

$$R = Q^{-1}AQ \quad (8)$$

jest macierzą trójkątną górną.

Zadanie 1. Obliczyć wartości i wektory własne i zdiagonalizować podane macierze:

$$A = \begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 3 \\ 3 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 1 & 4 \\ 2 & 2 & 2 \\ 6 & -3 & 3 \end{bmatrix}.$$

Zadanie 2. Przeanalizować, uruchomić i przetestować program zapisany w pliku *QR_gsl_example.c*. W szczególności zmienić macierz $A \in \mathbb{R}^{n \times n}$ wybierając macierz symetryczną, tj. $A = A^T$.

Zadanie 3. Zmodyfikować wyjście tak, aby wartości własne i wektory były wypisywane w formacie:

wartość własna 1 -> wektor własny 1

.....

wartość własna n -> wektor własny n

Na przykład:

2.34 $\rightarrow$ [2.5, 3.7, -2.8, 0.2]

.....

0.12 $\rightarrow$ [0.2, 7.1, 3.8, -1.1]

Zadanie 4. Zmodyfikować program tak, aby sprawdzał czy otrzymane wektory są faktycznie wektorami własnymi dla odpowiednich wartości własnych. Oznacza to, że należy sprawdzić równości

$$Av_i = \lambda_i v_i \quad \text{dla } i = 1, \dots, n,$$

gdzie λ_i to wartość własna, a v_i odpowiadający jej wektor własny.

```
#include <stdio.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_eigen.h>
#define n 5

int main (void) {
    double A[n*n];
    int i, j;

    for (i=0; i < n; i++)
        for (j=0; j < n; j++) A[n*i + j] = 1 + (i+j) + 1.0*cos(i*j)/(1.5+sin(i+j));

    gsl_vector *wart_wlasne = gsl_vector_alloc(n);
    gsl_matrix *wekt_wlasne = gsl_matrix_alloc(n, n);
```

Lab12: Problem własny; Metoda potęgowa, Ilorazy Rayleigha

```
gsl_matrix_view A_mv = gsl_matrix_view_array(A, n, n);
gsl_eigen_symmv_workspace *wsp = gsl_eigen_symmv_alloc(n);

// Tutaj jest dokonywany rozkład QR
gsl_eigen_symmv(&A_mv.matrix, wart_wlasne, wekt_wlasne, wsp);

gsl_eigen_symmv_free(wsp);

gsl_eigen_symmv_sort(wart_wlasne, wekt_wlasne, GSL_EIGEN_SORT_ABS_ASC);

for (i = 0; i < n; i++) {
    double wart_i = gsl_vector_get (wart_wlasne, i);
    gsl_vector_view wekt_i = gsl_matrix_column(wekt_wlasne, i);

    printf("wartość własna = %lf\n", wart_i);
    printf("wektor własny = \n");
    gsl_vector_fprintf(stdout, &wekt_i.vector, "%lf");
}

gsl_vector_free(wart_wlasne);
gsl_matrix_free(wekt_wlasne);

return 0;
}
```

METODA POTĘGOWA ZADANIA WŁASNEGO

Znajdowanie wszystkich wektorów i wartości własnych dla dowolnej macierzy jest dość złożonym zagadnieniem numerycznym, ale w szczególnym przypadku gdy macierz kwadratowa $A \in \mathbb{R}^{n \times n}$ jest diagonalizowalna (posiada bazę wektorów własnych) oraz istnieje jedna największa co do modułu wartość własna (tzw. wartość własna *dominująca*), znajdowanie tej największej wartości własnej i odpowiadającego jej wektora własnego jest stosunkowo proste.

Zakładamy zatem, że wartości własne macierzy A można uporządkować następująco

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|. \quad (9)$$

Ogólny algorytm metody potęgowej opiera się na wybraniu dowolnego niezerowego funkcjonału liniowego

$$\varphi: \mathbb{R}^n \rightarrow \mathbb{R} \quad (10)$$

i iteracyjnym obliczaniu ilorazów $r_k = \varphi(x^{(k+1)}) / \varphi(x^{(k)})$ gdzie $x^{(k+1)} = Ax^{(k)}$. Wektor startowy $x^{(0)} \in \mathbb{R}^n$ wybieramy w zasadzie dowolny byle był różny od zera. Okazuje się, że wtedy zachodzi

$$\lim_{k \rightarrow \infty} r_k = \lambda_{\max}.$$

Może się okazać, że dla danego wektora początkowego $x^{(0)}$ iteracje nie są zbieżne, to wtedy próbujemy inny wektor.

W praktyce stosuje się jednak pewną modyfikację opisanej metody. W takiej „czystej” postaci, $x^{(k+1)} = Ax^{(k)}$, mimo że faktycznie ciąg wektorów $(x^{(k+1)})_{k=0}^{\infty}$ zbliża się do kierunku wektora własnego $v_{\max}$ odpowiadającego wartości własnej $\lambda_{\max}$, to ciąg ten może zmierzać do zera lub stawać się nieograniczony. Aby tego uniknąć stosuje się normalizację wektora $x^{(k)}$ po każdej iteracji: $x^{(k)} := x^{(k)} / \|x^{(k)}\|$, gdzie $\|\cdot\|$ jest jedną z kilku typowych norm w $\mathbb{R}^n$, na przykład

$$\begin{aligned}
\|x\|_{\infty} &= \max\{|x_i| : 1 \leq i \leq n\}, \\
\|x\|_1 &= \sum_{i=1}^n |x_i|, \\
\|x\|_2 &= \sqrt{\sum_{i=1}^n |x_i|^2}.
\end{aligned} \tag{11}$$

Algorytm metody potęgowej

wejście: $n, A \in \mathbb{R}^{n \times n}, x \in \mathbb{R}^n, \text{numIter} \in \mathbb{N}$

wyjście: $\lambda_{\max}, v_{\max}$

```

for k=1 to numIter {
    y = A·x
    r = φ(y) / φ(x)
    x = y / |y|
    k=k+1
}
print r, x

```

Wariant metody potęgowej – ilorazy Rayleigha

W tym przypadku ściśle rzecz biorąc nie posługujemy się wybranym i ustalonym funkcjonałem liniowym φ tylko obliczamy ilorazy $r^{(k)}$ w oparciu o iloczyn skalarny wektorów $x^{(k)}$ i $x^{(k+1)}$:

$$r_k = \frac{x^{(k+1)} \cdot x^{(k)}}{x^{(k)} \cdot x^{(k)}} = \frac{x^{(k+1)} \cdot x^{(k)}}{\|x^{(k)}\|_2^2}. \tag{12}$$

Oczywiście tak jak w przypadku standardowej metody potęgowej należy wykonywać normalizację wektorów w każdym kroku iteracji.

Wyznaczanie pozostałych wartości i wektorów własnych w metodzie potęgowej

Metoda potęgowa pozwala wyznaczyć wartość własną o największym module. Kolejną wartość własną i odpowiadający jej wektor możemy wyznaczyć poprzez redukcję (*deflację*) wyjściowej macierzy przy pomocy już uzyskanych wartości i wektora własnego. Jeżeli macierz A jest symetryczna, to najprostszym sposobem deflacji jest odjęcie od A macierzy $\lambda v v^T$, gdzie λ i v to wartość własna i jej wektor własny, który jest unormowany, $\|v\|=1$. Zatem definiujemy macierz

$$A_1 = A - \lambda v v^T. \tag{13}$$

We wzorze (13) wektor własny v jest wektorem kolumnowym, zatem

$$\begin{aligned}
 A_1 = A - \lambda v v^T &= \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} - \lambda \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 & \cdots & v_n \end{bmatrix} \\
 &= \begin{bmatrix} a_{11} - \lambda v_1 v_1 & a_{12} - \lambda v_1 v_2 & a_{13} - \lambda v_1 v_3 & \cdots & a_{1n} - \lambda v_1 v_n \\ a_{21} - \lambda v_2 v_1 & a_{22} - \lambda v_2 v_2 & a_{23} - \lambda v_2 v_3 & \cdots & a_{2n} - \lambda v_2 v_n \\ a_{31} - \lambda v_3 v_1 & a_{32} - \lambda v_3 v_2 & a_{33} - \lambda v_3 v_3 & \cdots & a_{3n} - \lambda v_3 v_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} - \lambda v_n v_1 & a_{n2} - \lambda v_n v_2 & a_{n3} - \lambda v_n v_3 & \cdots & a_{nn} - \lambda v_n v_n \end{bmatrix} = [a_{ij} - \lambda v_i v_j].
 \end{aligned}$$

Można pokazać, że wektory własne macierzy A_1 są takie same jak macierzy A , natomiast wartości własne macierzy A_1 to zero oraz wszystkie wartości własne macierzy A z wyjątkiem λ . Symbolicznie

$$\sigma(A_1) \setminus \{0\} = \sigma(A) \setminus \{\lambda\}.$$

Teraz stosujemy metodę potęgową do macierzy A_1 , co powinno dać drugą w kolejności wartość własną i wektor własny.

ZADANIA DO SAMODZIELNEGO WYKONANIA

- 1) Napisać program, który zaimplementuje metodę potęgową z następującymi funkcjonalnościami liniowymi

$$\varphi_i(x) = x_i \quad \text{gdzie } x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

oraz z normami (do normalizacji) $\|\cdot\|_\infty, \|\cdot\|_1, \|\cdot\|_2$. Program powinien wykonywać zadaną liczbę iteracji, a następnie wypisywać przybliżenie $\lambda_{\max}$ i $v_{\max}$.

- 2) Przetestować program na macierzach

$$A_1 = \begin{bmatrix} 79 & 27 & 95 & 28 & 74 \\ 27 & 55 & 39 & 0 & 10 \\ 95 & 39 & 118 & 31 & 93 \\ 28 & 0 & 31 & 42 & 27 \\ 74 & 10 & 93 & 27 & 95 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 8.0 & 2.6 & 9.6 & 2.8 & 7.2 & 6.5 \\ 2.6 & 13.6 & 1.8 & 6.6 & 5.4 & 3.1 \\ 9.6 & 1.8 & 16.7 & 2.7 & 7.0 & 14.2 \\ 2.8 & 6.6 & 2.7 & 4.2 & 2.7 & 2.4 \\ 7.2 & 5.4 & 7.0 & 2.7 & 9.9 & 4.3 \\ 6.5 & 3.1 & 14.2 & 2.4 & 4.3 & 16.0 \end{bmatrix}$$

- 3) Napisać program, który podobnie jak w p. 1) będzie obliczał największą co do modułu wartość własną i odpowiadający wektor własny, ale stosując tym razem ilorazy Rayleigha. Przetestować na przykładach z p. 2) oraz porównać wyniki z tymi uzyskanymi w z programu z p. 1).
- 4) Napisać program, który będzie wyznaczał dwie największe (co do modułu) wartości własne w macierzy symetrycznej $A \in \mathbb{R}^{n \times n}$ oraz odpowiadające im wektory własne metodą potęgową (dowolny wariant) z redukcją (deflacją). Przetestować na następującej macierzy:

Lab12: Problem własny; Metoda potęgowa, Ilorazy Rayleigha

$$A = \begin{bmatrix} 66 & 28 & 31 & 24 & 33 \\ 28 & 64 & 52 & 16 & 3 \\ 31 & 52 & 78 & 9 & -2 \\ 24 & 16 & 9 & 27 & 36 \\ 33 & 3 & -2 & 36 & 57 \end{bmatrix}.$$

Program powinien dawać następujące wartości własne: $\lambda_1 = 157,896$; $\lambda_2 = 90,142$.