Véc-tơ riêng và giá trị riêng của ma trận

Nhắc lại kiến thức từ học phần đại số:

Véc-tơ riêng, giá trị riêng:

Cho ma trận $A = (a_{ij})_{m\times n}$. Véc-tơ $x \neq 0$ gọi là véc-tơ riêng của A nếu $Ax = \lambda x$. Khi đó λ gọi là giá trị riêng của A.

Ví dụ:

```
clc
clear all
format short
A=[1 2 3 4;2 1 2 3;3 2 1 2;4 3 2 1]
```

```
A = 4×4
     1     2     3     4
     2     1     2     3
     3     2     1     2
     4     3     2     1
```

```
[X,y]=eig(A)
```

```
X = 4×4
    0.6533   -0.4483    0.2706    0.5468
    0.2706    0.5468   -0.6533    0.4483
   -0.2706    0.5468    0.6533    0.4483
   -0.6533   -0.4483   -0.2706    0.5468
y = 4×4
   -3.4142         0         0         0
         0   -1.0990         0         0
         0         0   -0.5858         0
         0         0         0    9.0990
```

Ma trận A có các véc-tơ riêng là các cột của ma trận X ứng với các giá trị riêng tương ứng nằm trên đường chéo của ma trận y. Ta kiểm tra lại theo định nghĩa

$$A\begin{pmatrix} 0.6533 \\ 0.2706 \\ -0.2706 \\ -0.6533 \end{pmatrix} - (-3.4142)\begin{pmatrix} 0.6533 \\ 0.2706 \\ -0.2706 \\ -0.6533 \end{pmatrix} = \begin{pmatrix} 0 \\ -0.2220 \\ 0.2220 \\ -0.4441 \end{pmatrix} 10^{-15}.$$

```
A*X(:,1)-y(1,1)*X(:,1)
```

```
ans = 4×1
10^-15 ×
         0
   -0.2220
    0.2220
   -0.4441
```

Đa thức đặc trưng:

Từ phương trình $Ax = \lambda x$ ta có $(A - \lambda E)x = 0$. Hệ phương trình này có nghiệm khác 0 khi và chỉ khi $\det(A - \lambda E) = 0$. Điều này có nghĩa λ là nghiệm của phương trình $P_A(t) = 0$ với

$$P_A(t) = \det(A - tE)$$

là đa thức đặc trưng của A.

Ví dụ: Tìm đa thức đặc trưng của ma trận A

```
clc
clear all
A=[1 2 3 4;2 1 2 3;3 2 1 2;4 3 2 1]
```

```
A = 4×4
     1     2     3     4
     2     1     2     3
     3     2     1     2
     4     3     2     1
```

```
syms t
P = det(A-t*eye(size(A,1)));
disp(P)
```

$t^4 - 4\,t^3 - 40\,t^2 - 56\,t - 20$

Ma trận đồng dạng:

Hai ma trận vuông cùng cấp A và B gọi là đồng dạng với nhau nếu tồn tại ma trận khả nghịch T sao cho $B = T^{-1}AT$. Ta có

$$\begin{aligned}&\det(B - tE) = \det(T^{-1}AT - tT^{-1}ET)\\&= \det(T^{-1}(A - tE)T) = \det(T^{-1})\det(A - tE)\det(T)\\&= \det(A - tE).\end{aligned}$$

Do đó hai ma trận đồng dạng với nhau có cùng đa thức đặc trưng. Nghĩa là chúng có cùng tập giá trị riêng.

Giả sử x là giá trị riêng của B. Theo định nghĩa $Bx = \lambda x$. Điều này tương đương với

$$T^{-1}ATx = \lambda x \Leftrightarrow A(Tx) = \lambda(Tx).$$

Do đó Tx là véc-tơ riêng của A ứng với giá trị riêng λ.

Khối Frobenious dạng 1:

$$C^{(r)} = \begin{bmatrix} p_1 & p_2 & \cdots & p_{r-1} & p_r \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}; \quad C^{(1)} = p_1.$$

Có thể chứng minh quy nạp theo r công thức sau:

$$\det(C^{(r)} - tE) = (-1)^r(t^r - p_1t^{r-1} - p_2t^{r-2} - \cdots - p_r). \qquad (1)$$

Chứng minh. Nếu $r = 1$ thì $\det(C^{(1)} - t * 1) = (p_1 - t) = (-1)^1(t - p_1)$.

Giả sử công thức (1) đúng với $r - 1$, nghĩa là

$$\det(C^{(r-1)} - tE) = (-1)^{r-1}(t^{r-1} - p_1t^{r-2} - p_2t^{r-3} - \cdots - p_{r-1}). \qquad (2)$$

Khai triển theo cột cuối cùng của khối $C^{(r)} - tE$ và sử dụng giả thiết quy nạp (2), ta có

$$\begin{aligned}
&\det(C^{(r)} - tE) = (-1)^{r+1}p_r + (-1)^{r+r}(-t)\det(C^{(r-1)} - tE)\\
&= (-1)^r(t^r - p_1t^{r-1} - p_2t^{r-2} - \cdots - p_{r-1}t) + (-1)^{r+1}p_r\\
&= (-1)^r(t^r - p_1t^{r-1} - p_2t^{r-2} - \cdots - p_{r-1}t - p_r).
\end{aligned}$$

Điều này chứng tỏ (1) đúng với mọi r.

Ví dụ:

```
clc
clear all
A=[2 3 -1 0;1 0 0 0;0 1 0 0;0 0 1 0]
```

```
A = 4×4
     2     3    -1     0
     1     0     0     0
     0     1     0     0
     0     0     1     0
```

```
syms t
p = det(A-t*eye(size(A,1)));
p = expand(p);
disp(p)
```

$t^4 - 2\,t^3 - 3\,t^2 + t$

Khối Frobenious dạng 2:

$$C^{(r)} = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & 1 & 0\\ 0 & 0 & \cdots & 0 & 1\\ -p_1 & -p_2 & \cdots & -p_{r-1} & -p_r \end{bmatrix}$$

$$\det(C^{(r)} - tE) = (-1)^r(t^r - p_rt^{r-1} - p_{r-1}t^{r-2} - \cdots - p_1).$$

Dạng chuẩn Frobenious:

$$F = \begin{bmatrix} F_1 & 0 & \cdots & 0\\ 0 & F_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & F_k \end{bmatrix}$$

trong đó $F_j,\ j = \overline{1,k}$ là các khối Frobenius. Khi đó

$$\det(F - tE) = \prod_{j=1}^{k} \det(F_j - tE).$$

Như ta đã thấy ở trên một ma trận dạng chuẩn Frobenious có thể dễ dàng tìm được đa thức đặc trưng.

Phương pháp Danielevsky

Ý tưởng: Dùng các ma trận chuyển cơ sở đưa ma trận A về ma trận đồng dạng F, với F là ma trận dạng chuẩn Frobenious.

Kí hiệu: Trong phần dưới đây ta sẽ dùng M_i để chỉ dòng thứ n của ma trận M và $\mathscr{M}_j$ để chỉ cột thứ j của ma trận M. Dùng A_k để kí hiệu ma trận con vuông cấp k ở góc trên bên trái của ma trận A.

Bước 1: Đưa dòng thứ n của A về dạng $(0, 0, \cdots, 1, 0)$ hoặc $(0, 0, \cdots, a_{nn})$.

Trường hợp 1.1. Nếu $a_{n,n-1} \neq 0$. Ta sẽ đưa dòng thứ n của A về dạng $(0, 0, \cdots, 1, 0)$. Đặt ma trận $M^{(1)}$ như sau:

$$M^{(1)} = \begin{pmatrix} E_1 \\ E_2 \\ \dots \\ A_n \\ E_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Khai triển theo cột thứ $n-1$ của $M^{(1)}$ dễ dàng thấy $\det(M^{(1)}) = a_{n,n-1} \neq 0$. Sử dụng phương pháp phần bù đại số có thể tính được

$$(M^{(1)})^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \dfrac{-a_{n1}}{a_{n,n-1}} & \dfrac{-a_{n2}}{a_{n,n-1}} & \cdots & \dfrac{1}{a_{n,n-1}} & \dfrac{-a_{nn}}{a_{n,n-1}} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Đặt $A_1 = M^{(1)} A (M^{(1)})^{-1}$ (vai trò ma trận biến đổi $T = (M^{(1)})^{-1}$). Xét dòng cuối cùng của ma trận A_1. Với $j \neq n-1$ ta có

$$a_{n,j}^{(1)} = M_n^{(1)} A(\mathscr{M}^{(1)})_j^{-1} = A_n \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ \dfrac{-a_{nj}}{a_{n,n-1}} \\ 0 \end{pmatrix},$$

$$= a_{nj} * 1 + a_{n,n-1} * \frac{-a_{nj}}{a_{n,n-1}} = 0.$$

ở đây số 1 của véc-tơ cột $(\mathscr{M}^{(1)})_j^{-1}$ nằm ở vị trí thứ j.

Với $j = n-1$ bằng cách tương tự có thể kiểm tra được $a_{n,n-1}^{(1)}$=1.

Do đó $A_n^{(1)} = (0, 0, \cdots, 1, 0)$.

Ví dụ: Ma trận

```
clc
clear all
A=[1 2 3 4;2 1 2 3;3 2 1 2;4 3 2 1]
```

```
A = 4×4
     1     2     3     4
     2     1     2     3
     3     2     1     2
     4     3     2     1
```

```
n=size(A,1);
```

có phần tử $a_{43} = 3 \neq 0$. Ta tạo ma trận M

```
M = eye(n);
M(n-1,:) = A(n,:)
```

```
M = 4×4
     1     0     0     0
     0     1     0     0
     4     3     2     1
     0     0     0     1
```

```
Minv = M;
Minv(n-1,:) = -A(n,:)/A(n,n-1);
Minv(n-1,n-1) = 1/A(n,n-1);
disp('Ma trận nghịch đảo của M là')
```

```
Ma trận nghịch đảo của M là
```

```
disp(Minv)
```

```
    1.0000         0         0         0
         0    1.0000         0         0
   -2.0000   -1.5000    0.5000   -0.5000
         0         0         0    1.0000
```

Ma trận $A^{(1)} = MAM^{-1}$ là

```
A1 = M*A*Minv;
disp(A1)
```

```
   -5.0000   -2.5000    1.5000    2.5000
   -2.0000   -2.0000    1.0000    2.0000
  -24.0000  -15.0000   11.0000   19.0000
         0         0    1.0000         0
```

Trường hợp 1.2: Nếu $a_{n,n-1} = 0$ và tồn tại $a_{nk} \neq 0$ với $k < n-1$ thì ta sẽ đổi vị trí của a_{nk} và $a_{n,n-1}$ cho nhau sử dụng ma trận C được tạo thành từ ma trận đơn vị bằng cách đổi cột $n-1$ và cột k cho nhau. Nghĩa là

$$C = [\mathscr{E}_1, \mathscr{E}_2, \cdots, \mathscr{E}_{n-1}, \cdots, \mathscr{E}_k, \mathscr{E}_n].$$

Dễ dàng kiểm tra được $C^{-1} = C$. Đặt $A^{(1)} = CAC^{-1}$. Xét phần tử $a^{(1)}_{n,n-1}$ của ma trận $A^{(1)}$:

$$a^{(1)}_{n,n-1} = C_n A \mathscr{C}^{-1}_{n-1} = E_n A \mathscr{E}_k = A_n \mathscr{E}_k = a_{nk} \neq 0.$$

Ta quay trở về trường hợp 1.1 với vai trò ma trận A chính là ma trận $A^{(1)}$. Lưu ý chúng ta cần tích lũy ma trận $T = (MC)^{-1}$.

Ví dụ: Xét ma trận A có $a_{43} = 0$ và $a_{42} = 3 \neq 0$.

```
clc
clear all
A=[1 2 3 4;2 1 2 3;3 2 1 2;4 3 0 1]
```

```
A = 4×4
     1     2     3     4
     2     1     2     3
     3     2     1     2
     4     3     0     1
```

```
n = size(A,1);
C= eye(n);
k = 2;
C(:,[k,n-1]) = C(:,[n-1,k]);
disp('Ma trận C là')
```

```
Ma trận C là
```

```
disp(C)
```

```
     1     0     0     0
     0     0     1     0
     0     1     0     0
     0     0     0     1
```

```
A1 = C*A*C;
```

```
disp('Ma trận A1 là:')
```

```
Ma trận A1 là:
```

```
disp(A1)
```

```
     1     3     2     4
     3     1     2     2
     2     2     1     3
     4     0     3     1
```

Trường hợp 1.3. Nếu $a_{nk} = 0$, $\forall k \leq n-1$. Khi đó ta có $A_n = (0, 0, \cdots, a_{nn})$. Bằng cách khai triển dòng cuối của ma trận $A - tE$, ta dễ dàng tính được

$$P_A(t) = P_{A_{n-1}}(t)(a_{nn} - t)$$

với A_{n-1} là ma trận cấp $n-1$ nằm ở góc trên bên trái của ma trận A.

Ta lặp lại quy trình trên với các dòng thứ $n-1$, $n-2$,...

Bước k **:** Lúc này ta đang muốn đưa dòng thứ $n-k+1$ về dạng $(0, 0, ..., 1, ..., 0)$ với số 1 đứng ở vị trí thứ $n-k$. Lúc này ở góc dưới bên tay phải của ma trận $A^{(k)}$ đang là một khối Frobenious dạng 1 cấp k như sau

$$F = \begin{bmatrix} a_{n-k+1,n-k+1}^{(k-1)} & a_{n-k+1,n-k+2}^{(k-1)} & \cdots & a_{n-k+1,n-1}^{(k-1)} & a_{n-k+1,n}^{(k-1)} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

Ma trận $A^{(k)}$ có dạng

$$A^{(k)} = \begin{pmatrix} A_{n-k} & B \\ D & F \end{pmatrix}, \qquad (3)$$

trong đó D có dạng

$$D = \begin{pmatrix} a_{n-k+1,1}^{(k-1)} & a_{n-k+1,2}^{(k-1)} & \cdots & a_{n-k+1,n-k}^{(k-1)} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Trường hợp $k.1$: Nếu $a_{n-k+1,n-k}^{(k-1)} \neq 0$ thì ta làm như trường hợp 1.1 để mở rộng tiếp khối Frobenious F sử dụng ma trận $M^{(k)}$ như sau

$$M^{(k)} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ A_{n-k+1} \\ E_{n-k+1} \\ \vdots \\ E_n \end{pmatrix}$$

trong đó A_{n-k+1} nằm ở dòng thứ $n-k$ của ma trận $M^{(k)}$. Khi đó ma trận nghịch đảo $(M^{(k)})^{-1}$ có dạng như sau

$$(M^{(k)})^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{-a^{(k-1)}_{n-k+1,1}}{a^{(k-1)}_{n-k+1,n-k}} & \frac{-a^{(k-1)}_{n-k+1,2}}{a^{(k-1)}_{n-k+1,n-k}} & \cdots & \frac{1}{a^{(k-1)}_{n-k+1,n-k}} & \frac{-a^{(k-1)}_{n-k+1,n}}{a^{(k-1)}_{n-k+1,n-k}} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Ma trận

$$A^{(k)} = M^{(k)}A^{(k-1)}(M^{(k)})^{-1}$$

có dòng thứ $n-k+1$ có dạng $(0, 0, ..., 1, ..., 0)$ với số 1 đứng ở vị trí thứ $n-k$.

Trường hợp $k.2$: Nếu $a^{(k-1)}_{n-k+1,n-k} = 0$ và tồn tại $a^{(k-1)}_{n-k+1,r} \neq 0$ với $r < n-k$ thì ta đổi vị trí $a^{(k-1)}_{n-k+1,n-k}$ và $a^{(k-1)}_{n-k+1,r}$ cho nhau sử dụng ma trận C như trường hợp 1.2. Ma trận C có dạng

$$C = \begin{pmatrix} \mathscr{E}_1 & \mathscr{E}_2 & \cdots & \mathscr{E}_{n-k} & \cdots & \mathscr{E}_r & \cdots & \mathscr{E}_n \end{pmatrix},$$

trong đó $\mathscr{E}_{n-k}$ nằm ở vị trí r và $\mathscr{E}_r$ nằm ở vị trí $n-k$. Ta quay trở về trường hợp $k.1$ với ma trận $CA^{(k-1)}C^{-1}$.

Trường hợp $k.3$: Nếu $a^{(k-1)}_{n-k+1,r} = 0$, $\forall r \leq n-k$ thì ta sẽ cố gắng đưa B trong (3) về dạng ma trận 0. **Bước** $k.3.1$: Đưa cột 1 của B về 0. Đặt ma trận

$$B^{(1)} = \begin{pmatrix} 0 & -\mathscr{B}_1 & 0 & \cdots & 0 \end{pmatrix},$$

$$S = \begin{pmatrix} E_{n-k} & B^{(1)} \\ 0 & E_k \end{pmatrix}.$$

Dễ dàng kiểm tra được

$$S^{-1} = \begin{pmatrix} E_{n-k} & -B^{(1)} \\ 0 & E_k \end{pmatrix}.$$

Xét ma trận $\mathbb{A} = SAS^{-1}$

$$\mathbb{A} = \begin{pmatrix} E_{n-k} & B^{(1)} \\ 0 & E_k \end{pmatrix} \begin{pmatrix} A_{n-k} & B \\ 0 & F \end{pmatrix} \begin{pmatrix} E_{n-k} & -B^{(1)} \\ 0 & E_k \end{pmatrix}$$
$$= \begin{pmatrix} A_{n_k} & -A_{n-k}B^{(1)} + B + B^{(1)}F \\ 0 & F \end{pmatrix}.$$

Dễ thấy cột đầu tiên của ma trận $-A_{n-k}B^{(1)} + B + B^{(1)}F$ bằng 0.

Bước $k.3.q$**:** Tiếp theo đưa cột thứ $q < k$ về 0 sử dụng ma trận

$$B^{(q)} = \begin{pmatrix} 0 & \cdots & -\mathscr{B}_q & 0 & \cdots & 0 \end{pmatrix},$$

trong đó cột $-\mathscr{B}_q$ đặt ở vị trí $q+1$ và ma trận

$$S = \begin{pmatrix} E_{n-k} & B^{(q)} \\ 0 & E_k \end{pmatrix}$$
$$S^{-1} = \begin{pmatrix} E_{n-k} & -B^{(q)} \\ 0 & E_k \end{pmatrix}.$$

Nếu cột thứ k tự động về 0 sau khi thực hiện các bước như trên với $q = 1, 2, 3, \cdots, k-1$ thì ta đưa được ma trận $A^{(k-1)}$ về dạng

$$A^{(k)} = \begin{pmatrix} A_{n-k} & 0 \\ 0 & F \end{pmatrix}.$$

Với ma trận A_{n-k} ta lại coi vai trò như ma trận A để đưa tiếp về dạng chuẩn Forbenious.

Nếu cột thứ k không tự động về 0 sau khi thực hiện các bước như trên với $q = 1, 2, 3, \cdots, k-1$ thì ta đổi cột thứ k của ma trận B về cột 1 của B và lặp lại các bước $k.3.q$ với $q = 1, 2, 3, ..., k-1$. Cụ thể như sau: Để đổi cột thứ k của ma trận B về cột 1 của B ta sử dụng ma trận

$$W_k = \begin{pmatrix} \mathscr{E}_k & \mathscr{E}_1 & \mathscr{E}_2 & \cdots & \mathscr{E}_{k-1} \end{pmatrix}$$

và

$$U = \begin{pmatrix} E_{n-k} & W_k \\ 0 & E_k \end{pmatrix},$$
$$U^{-1} = U^T.$$

Sau đó lặp lại các bước $k.3.q$ với $q = 1, 2, 3, \cdots, k-1$ với ma trận $\mathbb{A} = UAU^{-1}$.

Chú ý: *Có thể cột k của ma trận B sẽ không tự động về 0, khi đó phương pháp Danielevski thất bại và ta sẽ dừng thuật toán.*

Sau khi thực hiện các **bước** k với $k = 1, 2, \cdots, n$ ta đưa được ma trận A ban đầu về dạng đồng dạng F với F có dạng chuẩn Frobenious

$$F = \begin{bmatrix} F_1 & 0 & \cdots & 0 \\ 0 & F_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F_d \end{bmatrix}.$$

Véc-tơ riêng của khối Frobenious:

Giả sử λ là một giá trị riêng của khối Frobenious F. Ta có

$$Fx = \lambda x \Leftrightarrow (F - \lambda E)x = 0 \Leftrightarrow \begin{cases} p_1x_1 + p_2x_2 + \cdots + p_rx_r & = & \lambda x_1 \\ x_1 & = & \lambda x_2 \\ x_2 & = & \lambda x_3 \\ \cdots & \cdots & \cdots \\ x_{r-1} & = & \lambda x_r \end{cases}$$

Do đó có thể lấy

$$x = \begin{pmatrix} \lambda^{r-1} \\ \lambda^{r-2} \\ \vdots \\ \lambda \\ 1 \end{pmatrix}$$

là một véc-tơ riêng của F ứng với giá trị riêng λ.

Giả sử F_j có véc-tơ riêng x_j ứng với giá trị riêng λ_j với $j = \overline{1, d}$. Khi đó véc-tơ

$$\mathcal{X}_j = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ x_j \\ \vdots \\ 0 \end{pmatrix}$$

là một véc-tơ riêng của ma trận F ứng với giá trị riêng λ_j.

Do đó biết được các véc-tơ riêng và giá trị riêng của các khối Frobenious ta có thể dễ dàng tìm ra véc-tơ riêng và giá trị riêng của ma trận F.

Sau khi tìm được véc-tơ riêng $\mathcal{X}$ của F ta tìm được véc-tơ riêng của A bằng $T\mathcal{X}$ với T là ma trận biến đổi cơ sở tích lũy qua các phép biến đổi đồng dạng.

Dưới đây là code đề xuất cho phương pháp Danielevski

```
% ham so Danielevski nhan vao ma tran A va tra ra
%eigValues chua cac gia tri rieng cua A
%eigVectors co cac cot chua cac vector rieng tuong ung voi gia tri
%rieng o vector eigValues
```

```
    %P va Pinv la ma tran doi co so F = P*A*Pinv
    %F la dang frobenius cua ma tran A
    %tol la gia tri dieu chinh, neu phan tu nao nho hon tol thi cho bang 0.
    clc
    clear all
    tol = 1e-5;
    A=[1 2 3 4;2 1 2 3;3 2 1 2;4 3 2 1];
    count = size(A,1);
    n = size(A,1);
    eigValue = zeros(0,1);
    while count>0
         [eigValues,P, Pinv, A2] = Goi5(A);
         count = count - (n-size(A2));
         eigValues = [eigValue; eigValues];
    end
    F = P*A*Pinv;
    F(abs(F)<tol) = 0;
    m = length(eigValues);
    eigenVectors = zeros(n,m);
    for i = 1 : m
        eigenVectors(:,i) = Goi6(eigValues(i),Pinv);
    end
    disp(eigValues)
```

```
     9.0990
    -3.4142
    -1.0990
    -0.5858
```

```
    disp(eigenVectors)
```

```
     1.0000   -1.0000    1.0000   -1.0000
     0.8198   -0.4142   -1.2198    2.4142
     0.8198    0.4142   -1.2198   -2.4142
     1.0000    1.0000    1.0000    1.0000
```

```
function [eigValue,P, Pinv, A2] = Goi5(A)
% Goi5 nhan vao ma tran co khoi Frobenius o goc
% tra ra gia tri rieng cua khoi Frobenius, ma tran o goc
% phep biến đổi tương ứng.
n = size(A,1);
count = n-1;
check = false;
P = eye(n);
Pinv = eye(n);

while check ==  false
    while A(count+1, count) ~= 0
        [A, S, Sinv] = Goi2(A, count+1);
        P = S*P;
        Pinv = Pinv*Sinv;
        count = count-1;
```

```
        if count <1
            break
        end
    end
    if all(A(count+1,1:count)==0)==false
        [A, S, Sinv] = Goi3(A,count+1);
        P = S*P;
        Pinv = Pinv*Sinv;
    else
        check = true;
        [A, S, Sinv, eigValue] = Goi4(A);
        P = S*P;
        Pinv = Pinv*Sinv;
    end
end
A2 = A(1:count,1:count);
end
function [A1, S, Sinv] = Goi2(A, k)
%k la vi tri dong cần thuc hien dua ve dang chuan Frobenius
%ham nay dua dong thu k cua ma tran ve dang [0 0 .. 1 .. 0]
%so 1 o vi tri thu k
    n = size(A,1);
    S = eye(n);
    S(k-1,:) = A(k,:);
    Sinv = eye(n);
    Sinv(k-1,:) = -A(k,:)./A(k,k-1);
    Sinv(k-1,k-1) = 1/A(k,k-1);
    A1 = S*A*Sinv;

end
function [A1, S, Sinv, eigValue] = Goi4(A)
%dua vao ma tran co khoi Frobenius duoc tách riêng
%tra ra vecto chua gia tri rieng, ma tran chuyen co so
%dua ma tran goc tren ben phai ve 0
%chua lam duoc truong hop cot cuoi B khac 0
n = size(A,1);
k = n-1;

while (k>0) && (A(k+1,k) ~= 0)
    k = k-1;
end

F = A(k+1:end,k+1:end);
A2 = A(1:k,1:k);
B = A(1:k, k+1:end);
Z = zeros(size(B'));
S = eye(n); Sinv = S;
eigValue = Goi1(F);
eigValue = eigValue(:);
for col = 1:size(B,2)-1
```

```
    % dua dan cac cot cua B ve zeros
    Bplus = zeros(size(B));
    Bplus(:,col+1) = B(:,col);
    Bminus = - Bplus;
    S1 = [eye(size(A2,1)), Bminus; Z, eye(size(F,1))];
    S = S1*S;
    S1inv = [eye(size(A2,1)), Bplus; Z, eye(size(F,1))];
    Sinv = S*S1inv;
    B= A2*Bplus + B + Bminus*F;
    A1 = [A2, B; Z, F];
end

if all(B(:,end)==0) == false
    %chuyen cot B(:,end) len dau va thuc hien lai
    W = eye(size(F,1)); W = [W(:,end), W]; W(:,end)=[];
    U = [eye(size(A2,1)), Z'; Z, W];
    Uinv = U';
    S = U*S;
    Sinv = Sinv*Uinv;
    F = W*F*W';
    B = B*W';
    A1 = [A2, B; Z F];

end

end
function eigVector = Goi1(F)
%nhap vao khoi Frobenius dang 1, tra ra gia tri rieng tương ứng
n = size(F,1);
p = (-1)^(n+1)*[1, -1*F(1,:)];
eigVector = roots(p);
end
function eigVector = Goi6(eigValue, Pinv)
%Goi6 nhan vao gia tri rieng eigValue cua ma tran F(A)
% va ma tran bien doi dong dang (F(A) = P*A*Pinv)
%Tra ra vector rieng cua ma tran A tuong ung gia tri rieng EigValue
    n=size(Pinv,1);
    soMu = n-1:-1:0;
    v = eigValue.^soMu;
    eigVector = Pinv*v(:);
end
```