TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC

----------o0o----------

BÁO CÁO MÔN HỌC GIẢI TÍCH SỐ

PHƯƠNG PHÁP DANILEVSKI TÌM TRỊ RIÊNG VÀ VECTOR RIÊNG

Giảng viên hướng dẫn: **TS. Hà Thị Ngọc Yến**

Sinh viên thực hiện: Nguyễn Trường Giang - 20195868

Nguyễn Minh Đức - 20195858

Nguyễn Văn Tuấn - 20195939

Lưu Đức Trọng - 20195935

Nguyễn Quý Bách - 20195842

Hà Nội – 2021

Mục lục

LỜI NÓI ĐẦU

Giải tích số (Numerical Analysis) được biết đến là một khoa học nghiên cứu cách giải gần đúng, chủ yếu là giải số, các phương trình, các bài toán xấp xỉ hàm số và các bài toán tối ưu.

Thoạt đầu, toán học phát sinh do nhu cầu giải các bài toán thực tế: Tính diện tích đất đai, quỹ đạo sao chổi, đường đi của các tàu buôn trên biển v.v… Như vậy có thể nói lúc đầu toán học đồng nghĩa với toán học tính toán. Cùng với sự phát triển nội tại của toán học và các ngành khoa học khác, toán học chia thành toán lí thuyết và toán ứng dụng. Tuy nhiên, các nhà toán học vĩ đại như Newton, Lagrange, Euler,Lobasepski, Gauss, Chebysev, v.v… đều có các công trình nên móng trong Giải tích số.

Ngày nay với sự xuất hiện của siêu máy tính (Super Computer) khả năng song song hóa các quá trình tính toán được mở rộng. Nhiều thuật toán song song đã được đề xuất và áp dụng giải các bài toán thực tế.

Hiện nay, nhiều bài toán cơ học, vật lí được quy về việc tìm các giá trị riêng và vector riêng của ma trận.

Nhận biết được sự cần thiết việc đi tìm các giá trị riêng và vector riêng của ma trận, cùng sự hướng dẫn tận tình của giảng viên bộ môn, TS. Hà Thị Ngọc Yến, chúng em tiến hành tìm hiểu về đề tài “Phương pháp Danielevski tìm giá trị riêng”.

Có hai loại phương pháp thường được sử dụng để tìm trị riêng và vector riêng của ma trận- đó là các phương pháp trực tiếp và phương pháp lặp. Tư tưởng của các phương pháp trực tiếp là bằng một số hữu hạn các phép biến đổi tương đương đưa ma trận A về ma trận có cấu trúc đơn giản, từ đó dễ tìm đa thức đặc trưng và các vector riêng. Trong khi đó phương pháp lặp lại sử dụng ý tưởng khuếch đại sự khác biệt của các giá trị riêng bằng cách dùng các lũy thừa bậc cao. Và phương pháp Danielevski thuộc vào lớp phương pháp trực tiếp được nêu trên.

Do khả năng còn hạn chế và thời gian có hạn, nên báo cáo không tránh khỏi những sai sót. Rất mong giảng viên bộ môn cùng bạn đọc thông cảm và góp ý để bản báo cáo được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn giảng viên bộ môn - TS. Hà Thị Ngọc Yến đã giúp đỡ và hướng dẫn chúng em tận tình trong suốt thời gian viết bài báo cáo, nhờ đó mà chúng em hoàn thành bài báo cáo của mình được tốt hơn.

1. Nhắc lại một số kiến thức đại số.

1.1. Trị riêng và vector riêng.

Bài Toán. Cho ma trận A = [a_{ij}] vuông thực cấp n. Tìm số thực λ và vector $X \neq 0 \in R^n$ thỏa mãn:

$$AX = \lambda X \quad (1)$$

Trong đó số thực λ để tồn tại X thỏa mãn (1) được gọi là **trị riêng**, còn vector X ứng với λ thỏa mãn (1) được gọi là **vector riêng**. Từ (1), ta có:

$$(A - \lambda I)X = 0 \quad (2)$$

Để tồn tại X ≠ 0, ta phải có điều kiện:

$$det(A - \lambda I) = 0$$

Đa thức P(λ) = det(A − λI) được gọi là đa thức đặc trưng của ma trận A.

Một cách đi tìm trị riêng thông thường là chúng ta khai triển định thức trên tìm đa thức đặc trưng của ma trận A, và giải hệ phương trình (A − λI)X = 0 để tìm vector riêng sau khi thu được các vector riêng. Nhưng với cách làm như vậy, số lượng tính toán sẽ rất lớn với n! phép cộng và (n − 1)n! phép nhân.

1.2. Ma trận đồng dạng.

Cho hai ma trận A và B vuông cùng cấp n. A đồng dạng với ma trận B, A ~ B nếu tồn tại ma trận không suy biến T sao cho:

$$B = T A T^{-1}$$

Tính Chất

Nếu ma trận A đồng dạng với ma trận B thì:

1. B ~ A và nếu B ~ C thì A ~ C

2. X là vector riêng của B thì $T^{-1}X$ là vector riêng của A

3. A và B có cùng đa thức đặc trưng (hay A và B có cùng các trị riêng)

Từ các tính chất của hai ma trận đồng dạng như trình bày ở trên, phương pháp Danilevski rút ra từ ý tưởng là sẽ thực hiện các biến đổi đồng dạng lên ma trận A để đưa ma trận A về dạng thuận tiện rút ra được trị riêng và vector riêng.

2. Ma trận Frobenius.

2.1. Ma trận Frobenius.

Ma trận Frobenius cấp n có dạng

$$P = \begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \cdots & \boldsymbol{p_{n-1}} & \boldsymbol{p_n} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

Đa thức đặc trưng của P là:

$$det(P-\lambda\boldsymbol{I}) = \begin{vmatrix} \boldsymbol{p_1}-\lambda & \boldsymbol{p_2} & \boldsymbol{p_3} & \cdots & \boldsymbol{p_n} \\ \mathbf{1} & -\lambda & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & -\lambda & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -\lambda \end{vmatrix}$$

Ta chứng minh quy nạp

$$det(P - \lambda I) = (-1)^n (\lambda^n - p_1\lambda^{n-1} - p_2\lambda^{n-2} - \cdots - p_{n-1}\lambda - p_n)$$

Đặt P_i là ma trận P bỏ đi (n-i) hàng và cột cuối.
Với i = 2, ta có: $det(P_2 - \lambda I_2) = -\lambda(p_1 - \lambda) - p_2 = \lambda^2 - p_1\lambda - p_2$
Theo giả thiết quy nạp, giả sử rằng mệnh đề đúng với n = k, ta chứng minh mệnh đề đúng với ma trận cấp n = k+1: Khai triển Laplace theo cột k+1, ta có:

$$det(P_{k+1} - \lambda I) = (-1)^{k+2}p_{k+1} - \lambda det(P_k - \lambda I)$$
$$= (-1)^{k+2} p_{k+1} - \lambda[(-1)^k (\lambda\ k - p_1\lambda^{k-1} - \cdots - p_k\lambda)]$$
$$= (-1)^{k+1}(\lambda^{k+1} - p_1\lambda^k - p_2\lambda^{k-1} - \cdots - p_k\lambda^2 - p_{k+1})$$

Như vậy hệ số của đa thức đặc trưng của ma trận Frobenius có thể suy ra từ các phần tử nằm trên hàng đầu tiên của ma trận.

Vector riêng của Ma trận Frobenius

Giả sử λ_i là 1 trị riêng tìm được của một ma trận Frobenius. Vector riêng Y ≠ 0 thỏa mãn:

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \cdots & \boldsymbol{p_{n-1}} & \boldsymbol{p_n} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \\ \boldsymbol{y_3} \\ \vdots \\ \boldsymbol{y_n} \end{bmatrix} = \lambda_i \begin{bmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \\ \boldsymbol{y_3} \\ \vdots \\ \boldsymbol{y_n} \end{bmatrix}$$

Nếu $y_n = 0$, từ hệ phương trình ta có:

$$y_k = \lambda_i y_{k+1}\ \forall k = \overline{\mathbf{1},\ \boldsymbol{n-1}}$$

$$\Rightarrow y_k = 0\ \forall k = \overline{\mathbf{1},\ \boldsymbol{n}}$$

Mà Y ≠ 0 (mâu thuẫn). Vậy $y_n \neq 0$. Ta biết rằng, vector riêng sai khác nhau một hàng số nhân, nên để tìm ra 1 vector riêng, do $y_n \neq 0$ ta có thể giả sử rằng $y_n = 1$

Với $y_n = 1$, suy ra: $y_k = \lambda_i^{n-k}\ \forall k = \overline{\mathbf{1}, \boldsymbol{n-1}}$

Bộ các số y_k thỏa mãn phương trình đầu tiên của hệ vì:

$$p_1y_1 + p_2y_2 + \cdots + p_ny_n = \lambda_iy_1$$

$$\Leftrightarrow p_1\lambda_i^{n-1} + p_2\lambda_i^{n-2} + \cdots + p_n = \lambda_i^n$$

Đẳng thức trên đúng vì λ_i là nghiệm của đa thức đặc trưng. Từ cách chứng minh trên, ta thấy rằng, ứng với mỗi 1 trị riêng λ_i của ma trận Frobenius ứng với duy nhất 1 vector riêng độc lập tuyến tính. Hay không gian riêng ứng với các trị riêng phân biệt của ma trận Frobenius có số chiều bằng 1. Như vậy, ta có thể thu được tất cả các vector riêng của ma trận Frobenius từ các trị riêng.

2.2. Ma trận khối Frobenius.

Ma Trận khối Frobenious có dạng như sau:

$$P = \begin{bmatrix} \boldsymbol{P_1} & & & \\ & \boldsymbol{P_2} & & \\ & & \ddots & \\ & & & \boldsymbol{P_n} \end{bmatrix}$$

Trong đó các ma trận P_i là các ma trận Frobenius. Ta có đa thức đặc trưng của ma trận khối Frobenius:

$$det(P - \lambda I) = \prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} det(Pi - \lambda I_i)$$

(Trong đó I_i là các ma trận đơn vị có cấp tương ứng với các ma trận P_i)

Do đó, ta có thể dễ dàng thu được đa thức đặc trưng của ma trận khối Frobenius bằng tích của các đa thức Frobenius P_i

Giả sử λ là trị riêng của ma trận P, ta có:

$$P = \begin{bmatrix} \boldsymbol{P_1} & & & \\ & \boldsymbol{P_2} & & \\ & & \ddots & \\ & & & \boldsymbol{P_n} \end{bmatrix} \begin{bmatrix} \boldsymbol{v_1} \\ \boldsymbol{v_2} \\ \vdots \\ \boldsymbol{v_n} \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{v_1} \\ \boldsymbol{v_2} \\ \vdots \\ \boldsymbol{v_n} \end{bmatrix} \qquad (4)$$

(Với các v_i là các vector có cấp tương ứng với các ma trận P_i) Mặt khác, ta có:

$$P = \begin{bmatrix} \boldsymbol{P_1} & & & \\ & \boldsymbol{P_2} & & \\ & & \ddots & \\ & & & \boldsymbol{P_n} \end{bmatrix} \begin{bmatrix} \boldsymbol{v_1} \\ \boldsymbol{v_2} \\ \vdots \\ \boldsymbol{v_n} \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{P_1 v_1} \\ \boldsymbol{P_2 v_2} \\ \vdots \\ \boldsymbol{P_n v_n} \end{bmatrix} \quad (5)$$

Từ (4) và (5), ta có:

$$P_i v_i = \lambda v_i \quad \forall i = \overline{\boldsymbol{1, n}}$$

Nếu Pi không có trị riêng là λ, suy ra $v_i = 0$

Mặt khác, vì λ là trị riêng của ma trận P, nên λ phải là trị riêng của các ma trận ma trận Frobenius con nào đó trong P.

Giả sử λ là trị riêng của các ma trận P_{m1} , P_{m2} , . . . , P_{mk}
Theo chứng minh trên, không gian riêng ứng với một trị riêng phân biệt của ma trận Frobenius có số chiều bằng 1, ta gọi v_{mi} là vector riêng của ma trận P_{mi} ứng với trị riêng λ.
Như vậy:

$$P_{mi}\, v_{mi} = \lambda v_{mi} \;\; \forall i = \overline{\boldsymbol{1, n}}$$

Đặt (với v_{mi} nằm ở vị trí tương ứng): $y_{mi} = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \\ \boldsymbol{v_{mi}} \\ \vdots \\ \boldsymbol{0} \end{bmatrix}$

Dễ thấy tập hợp các vector y_{mi} với i = 1, n độc lập tuyến tính và thỏa mãn:

$$Py_{mi} = \lambda y_{mi} \text{ (do } P_{mi}\, v_{mi} = \lambda v_{mi} \; \forall\, i = \overline{\boldsymbol{1, n}} \text{)}$$

Vậy, từ ma trận khối Frobenius ta có thể dễ dàng thu được tất cả các vector riêng ứng với một trị riêng λ bất kì với cách xây dựng như trên.

3. Phương pháp Danilevski.

Ý tưởng phương pháp:

Thực hiện các phép biến đổi đồng dạng lên ma trận vuông thực A để đưa về dạng ma trận Frobenius hay ma trận khối Frobenius.

3.1. Phương pháp Danilevski tìm trị riêng.

Xét trong trường hợp tổng quát, giả sử tại bước thứ i + 1 ta luôn có

$$a_{n-i,n-i-1}^{(i+1)} \neq 0\ \forall i = \overline{1, n-1}$$

Như vậy, sau mỗi bước ta luôn xác định được $A^{(i+1)} \sim A^{i}$ bằng các bước thực hiện như sau

Bước 1: Đặt ma trận ban đầu $A = A^{(1)}$

$$A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,n-1}^{(1)} & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{21}^{(1)} & \cdots & a_{2,n-1}^{(1)} & a_{2,n}^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-1,1}^{(1)} & a_{n-1,2}^{(1)} & \cdots & a_{n-1,n-1}^{(1)} & a_{n-1,n}^{(1)} \\ a_{n,1}^{(1)} & a_{n,2}^{(1)} & \cdots & a_{n,n-1}^{(1)} & a_{n,n}^{(1)} \end{bmatrix}$$

Đưa ma trận $A^{(1)}$ về ma trận đồng dạng $A^{(2)}$ với ma trận $A^{(2)}$ có dạng như sau:

$$A^{(2)} = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1,n-1}^{(2)} & a_{1n}^{(2)} \\ a_{21}^{(2)} & a_{21}^{(2)} & \cdots & a_{2,n-1}^{(2)} & a_{2,n}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-1,1}^{(2)} & a_{n-1,2}^{(2)} & \cdots & a_{n-1,n-1}^{(2)} & a_{n-1,n}^{(2)} \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

Ta có biến đổi $A^{(2)} = M_1 A^{(1)} M_1^{-1}$, với các ma trận M_1, M_1^{-1} như sau:

$$M_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-1} & a_{n,n} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

và,

$$M_1^{-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\dfrac{a_{n,1}^{(1)}}{a_{n,n-1}^{(1)}} & -\dfrac{a_{n,2}^{(1)}}{a_{n,n-1}^{(1)}} & \cdots & \dfrac{1}{a_{n,n-1}^{(1)}} & -\dfrac{a_{n,n}^{(1)}}{a_{n,n-1}^{(1)}} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

Với điều kiện tổng quát $a_{n-i,n-i-1}^{(i+1)} \neq 0 \forall i = \overline{1, n-1}$ luôn xác định được M_{i+1}^{-1} vì $det(M^{(i+1)}) = a_{n-i,n-i-1}^{i+1} \neq 0$

Bước i: Biến đổi đồng dạng ma trận $A^{(i)}$ thành ma trận $A^{(i+1)}$ bằng biến đổi

$$A^{(i+1)} = M_i A^{(i)} M_i^{-1}$$

$$\boldsymbol{M_i} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\ \boldsymbol{a_{n-i+1,1}^{(i)}} & \boldsymbol{a_{n-i+1,2}^{(i)}} & \cdots & \boldsymbol{a_{n-i+1,n-i}^{(i)}} & \cdots & \boldsymbol{a_{n-i+1,n-1}^{(i)}} & \boldsymbol{a_{n-i+1,n}^{(i)}} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

$$\boldsymbol{M_i^{-1}} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ -\dfrac{\boldsymbol{a_{n-i+1,1}^{(i)}}}{\boldsymbol{a_{n-i+1,n-i}^{(i)}}} & -\dfrac{\boldsymbol{a_{n-i+1,2}^{(i)}}}{\boldsymbol{a_{n-i+1,n-i}^{(i)}}} & \cdots & -\dfrac{\mathbf{1}}{\boldsymbol{a_{n-i+1,n-i}^{(i)}}} & \cdots & -\dfrac{\boldsymbol{a_{n-i+1,n}^{(i)}}}{\boldsymbol{a_{n-i+1,n-i}^{(i)}}} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{1} \end{bmatrix}$$

$$\boldsymbol{A^{(i)}} = \begin{bmatrix} \boldsymbol{a_{11}^{(i)}} & \boldsymbol{a_{12}^{(i)}} & \cdots & \boldsymbol{a_{1,n-i}^{(i)}} & \cdots & \boldsymbol{a_{1,n-1}^{(i)}} & \boldsymbol{a_{1,n}^{(i)}} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{a_{n-i,1}^{(i)}} & \boldsymbol{a_{n-i,2}^{(i)}} & \cdots & \boldsymbol{a_{n-i,n-i}^{(i)}} & \cdots & \boldsymbol{a_{n-i,n-1}^{(i)}} & \boldsymbol{a_{n-i,n}^{(i)}} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} & \cdots & \mathbf{0} & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

Sau $\boldsymbol{n-1}$ bước biến đổi, ta thu được ma trận $\boldsymbol{A^{(n)}} \sim \boldsymbol{A}$ với

$$\boldsymbol{A^{(n)} = P = M_{n-1}A^{(n-1)}M_{n-1}^{-1}}$$

$$\boldsymbol{= M_{n-1}M_{n-2}A^{(n-2)}M_{n-2}^{-1}M_{n-1}^{-1}}$$

$$= \cdots =$$

$$\boldsymbol{= (M_{n-1}M_{n-2}\ldots M_2M_1)A_n^{(1)}(M_1^{-1}M_2^{-1}\ldots M_{n-2}^{-1}M_{n-1}^{-1})}$$

Đặt $\boldsymbol{M = M_{n-1}M_{n-2} \ldots M_2M_1}$ thì

$$\boldsymbol{M^{-1} = (M_{n-1}M_{n-2}\ldots M_2M_1)^{-1} = M_1^{-1}M_2^{-1}\ldots M_{n-2}^{-1}M_{n-1}^{-1}}$$

Khi đó $\boldsymbol{P = A^{(n)} = MA^{(1)}M^{-1} = MAM^{-1}}$

Tức là $\boldsymbol{P \sim A}$

- ⇨ Qua quá trình thực hiện trên, ta hoàn toàn xác định được trị riêng của *P*, từ đó xác định được trị riêng của *A*

Thực tế, trong quá trình tính toán không phải lúc nào cũng xảy ra trường hợp lý tưởng $\boldsymbol{a_{n-i+1,n-i}^{(i)} \neq 0}$ nên ta xét trường hợp $\boldsymbol{a_{n-i+1,n-i}^{(i)} = 0}$

TH1: $\exists \boldsymbol{j < n-i}: \boldsymbol{a_{n-i+1,j}^{(i)} \neq 0}$

Tức là: Tại hàng *n-i+1* của ma trận $A^{(i)}$ tồn tại phần tử $\boldsymbol{a_{n-i+1,j}^{(i)} \neq 0}$ với mọi $j < n\text{-}i$

$$\boldsymbol{A^{(i)}} = \begin{bmatrix} a_{11}^{(i)} & \cdots & a_{1,j}^{(i)} & a_{1,j+1}^{(i)} & \cdots & a_{1,n-i}^{(i)} & a_{1,n-i+1}^{(i)} & \cdots & a_{1,n}^{(i)} \\ a_{21}^{(i)} & \cdots & a_{2,j}^{(i)} & a_{2,j+1}^{(i)} & \cdots & a_{2,n-i}^{(i)} & a_{2,n-i+1}^{(i)} & \cdots & a_{2,n}^{(i)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n-i,1}^{(i)} & \cdots & a_{n-i,j}^{(i)} & a_{n-i,j+1}^{(i)} & \cdots & a_{n-i,n-i}^{(i)} & a_{n-i,n-i+1}^{(i)} & \cdots & a_{n-i,n}^{(i)} \\ 0 & \cdots & a_{n-i+1,j}^{(i)} & 0 & \cdots & 0 & a_{n-i+1,n-i+1}^{(i)} & \cdots & a_{n-i+1,n}^{(i)} \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Ta hoán vị cột *j* và cột *n-i*, hàng j và hàng *n-i* bằng cách thực hiện biến đổi $\boldsymbol{A^{(i)} = CA^{(i)}C^{-1}}$ sẽ thu được ma trận đồng dạng với ma trận $\boldsymbol{A^{(i)}}$, với ma trận C ($\boldsymbol{C = C^{-1}}$) có dạng

$$\boldsymbol{C} = \begin{array}{cc} & \begin{array}{ccccccc} & & j & & n-i & & \end{array} \\ \begin{array}{c} \\ \\ j \\ \\ n-i \\ \\ \\ \end{array} & \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \end{array}$$

Khi đó, phần tử $\boldsymbol{a_{n-i+1,n-i}^{(i)} \neq 0}$, ta tiếp tục quá trình biến đổi đồng dạng

TH2: $\forall \boldsymbol{j < n-i}: \boldsymbol{a_{n-i+1,j}^{(i)} = 0}$

Tức là: Tại hàng *n-i+1*, các phần tử trước $\boldsymbol{a_{n-i+1,n-i}^{(i)}}$ đều bằng 0 với mọi $j < n\text{-}i$

$$
A^{(i)} = \begin{bmatrix}
a_{11}^{(i)} & \cdots & a_{1,j}^{(i)} & a_{1,j+1}^{(i)} & \cdots & a_{1,n-i}^{(i)} & a_{1,n-i+1}^{(i)} & \cdots & a_{1,n}^{(i)} \\
a_{21}^{(i)} & \cdots & a_{2,j}^{(i)} & a_{2,j+1}^{(i)} & \cdots & a_{2,n-i}^{(i)} & a_{2,n-i+1}^{(i)} & \cdots & a_{2,n}^{(i)} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
a_{n-i,1}^{(i)} & \cdots & a_{n-i,j}^{(i)} & a_{n-i,j+1}^{(i)} & \cdots & a_{n-i,n-i}^{(i)} & a_{n-i,n-i+1}^{(i)} & \cdots & a_{n-i,n}^{(i)} \\
0 & \cdots & 0 & 0 & \cdots & 0 & a_{n-i+1,n-i+1}^{(i)} & \cdots & a_{n-i+1,n}^{(i)} \\
0 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0
\end{bmatrix}
$$

Ta biểu diễn ma trận $A^{(i)}$ thành ma trận khối như sau: $A^{(i)} = \begin{bmatrix} A' & B \\ 0 & P' \end{bmatrix}$

Ta thấy P' là ma trận Frobenius cấp i, và từ tính chất ma trận khối đã chứng minh trước, ta biến đổi ma trận vuông A' cấp n-i theo phương pháp Danilevski. Từ đó hoàn toán tính được trị riêng của A theo A' và P'.

$$\mathbf{det}(A^{(i)} - \lambda I) = \det[(A' - \lambda I)(P' - \lambda I)] = P'(\lambda)\,\mathbf{det}(A' - \lambda I)$$

3.2. Phương pháp Danilevski tìm vector riêng.

Tương ứng với phương pháp danilevski tìm trị riêng, nếu $a_{n-i+1,n-i}^{(i)} \neq 0$

thì tìm được vector riêng theo công thức đã chứng minh, ngược lại, $a_{n-i+1,n-i}^{(i)} = 0$ ta lại chia 2 trường hợp:

TH1: $\exists j < n - i : a_{n-i+1,j}^{(i)} \neq 0$

Tức là: Tại hàng n-i+1 của ma trận $A^{(i)}$ tồn tại phần tử $a_{n-i+1,j}^{(i)} \neq 0$ với mọi $j < n$-i

Dựa vào các tính chất về trị riêng và vector riêng của ma trận Frobenius đã chứng minh, ta tìm được tất cả các vector riêng của ma trận P đồng dạng với ma trận A. Gọi Y là vector riêng của ma trận Frobenius P, theo tính chất của 2 ma trận đồng dạng, ta có vector riêng của ma trận A:

$$X = M^{-1}Y$$

TH2: $\forall j < n - i : a_{n-i+1,j}^{(i)} = 0$

Tức là: Tại hàng n-i+1, các phần tử trước $a_{n-i+1,n-i}^{(i)}$ đều bằng 0 với mọi $j < n$-i

Ta đã biểu diễn dc A về dạng $A^{(i)} = \begin{bmatrix} A' & B \\ 0 & P' \end{bmatrix}$

Ta sẽ cố gắng triệt tiêu ma trận *B* với phép biến đổi đồng dạng. Giả sử ta muốn triệt tiêu cột *q* của ma trận $\boldsymbol{A^{(i)}}$ (nằm trong ma trận B). Thực hiện biến đối đồng dạng $\boldsymbol{A^{(i)} = S_q A^{(i)} S_q^{-1}}$

$$\boldsymbol{S_q} = \begin{array}{c} \begin{array}{cccccccc} & & & & \boldsymbol{q+1} & & & \end{array} \\ \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{-a_{1,q}} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \cdots & \boldsymbol{-a_{2,q}} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \cdots & \boldsymbol{-a_{3,q}} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{-a_{n-i-1,q}} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \cdots & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \end{array}$$

$$\boldsymbol{S_q^{-1}} = \begin{array}{c} \begin{array}{cccccccc} & & & & \boldsymbol{q+1} & & & \end{array} \\ \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{a_{1,q}} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \cdots & \boldsymbol{a_{2,q}} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \cdots & \boldsymbol{a_{3,q}} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{a_{n-i-1,q}} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \cdots & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \end{array}$$

Tiếp tục biến đổi đến cột *n-1*, ta thu được ma trận

$$\boldsymbol{A^{(i)}} = \begin{bmatrix} \boldsymbol{a_{11}^{(i)}} & \cdots & \boldsymbol{a_{1,n-i-1}^{(i)}} & \boldsymbol{a_{1,n-i}^{(i)}} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{b_{1,n}^{(i)}} \\ \boldsymbol{a_{21}^{(i)}} & \cdots & \boldsymbol{a_{2,n-i-1}^{(i)}} & \boldsymbol{a_{2,n-i}^{(i)}} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{b_{2,n}^{(i)}} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{a_{n-i,1}^{(i)}} & \cdots & \boldsymbol{a_{n-i,n-i-1}^{(i)}} & \boldsymbol{a_{n-i,n-i}^{(i)}} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{b_{n-i,n}^{(i)}} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{a_{n-i+1,n-i+1}^{(i)}} & \cdots & \boldsymbol{a_{n-i+1,n-1}^{(i)}} & \boldsymbol{a_{n-i+1,n}^{(i)}} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{1} & \boldsymbol{0} \end{bmatrix}$$

Tại đây, ta lại xét 2 trường hợp:

TH 2.1: $\boldsymbol{\forall j < n - i : a_{n-i+1,j}^{(i)} = 0}$ $\boldsymbol{v}$à $\boldsymbol{b_{j,n}^{(i)} = 0 \forall j = \overline{1, n-k}}$

Khi đó ma trận *A* có dạng $\boldsymbol{A^{(i)}} = \begin{bmatrix} \boldsymbol{A'} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P'} \end{bmatrix}$

Với *P'* là ma trận Frobenius, ta biến đổi *A'* về dạng Frobenius, cuối cùng sẽ thu được ma trận dạng khối Frobenius và tìm được vector riêng(chứng minh trên).

TH 2.2: $\forall \boldsymbol{j} < \boldsymbol{n} - \boldsymbol{i}: \boldsymbol{a}^{(i)}_{n-i+1,j} = \boldsymbol{0}$ $\boldsymbol{v}$à $\exists \boldsymbol{j} \in [\boldsymbol{1}, \boldsymbol{n} - \boldsymbol{k}]: \boldsymbol{b}^{(i)}_{j,n} \neq \boldsymbol{0}$ (với $\boldsymbol{b}^{(i)}_{j,n}$ là phần tử đầu tiên khác 0 từ dưới lên)

Ta dựng ma trận U_j

$$\boldsymbol{U_j} = \begin{bmatrix} \mathbf{1} & & & & & & & \\ & \ddots & & & & & & \\ & & \mathbf{1} & & & & & \\ & & & \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ & & & \mathbf{0} & \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \\ & & & \cdots & \cdots & \cdots & \ddots & \cdots \\ & & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \\ & & & \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

Phần khối ma trận chuyển hàng (cột) có cấp n − j + 1. Dễ thấy, ma trận nghịch đảo của ma trận Uj là ma trận chuyển vị của nó:

$$\boldsymbol{U_j^{-1} = U_j^T}$$

Thực hiện phép gán: $\boldsymbol{A^{(i)} = U_j A^{(k)} U_j^T}$, bắt đầu vòng lặp mới, thực hiện biến đổi sử dụng phương pháp Danilevski với ma trận thu được, ta sẽ thu được ma trận Frobenius hoặc ma trận khối Frobenius, từ đó tiếp tục thực hiện quá trình tìm vector riêng

4.Thuật toán và đánh giá

4.1. Thuật toán

Danilevski(A):

Argument:
A : Ma trận 2 chiều
Return:
Danh sách các vector riêng ứng với từng trị riêng
KHAI BÁO VÀ KHỞI TẠO

list_eigenvalues = []
list_eigenvectors = []
n = A.shape[0] : kích cỡ ma trận A
m = n : sử dụng trong vòng lặp
k = n : sử dụng trong vòng lặp
back = eye(n) : là ma trận đơn vị cỡ n.n, sử dụng để lưu tích các ma trận nghịch đảo M, dùng

để tính vecto riêng của A

CHẠY VÒNG LẶP ĐỂ ĐƯA A VỀ MA TRẬN FROBENIUS

```
while k > 1 :
    if A[k, k − 1]! = 0 :  # Trường hợp lí tưởng

#tìm ma trận đồng dạng
        A, inverseM = findSimpleA(A, k)
        back = back * inverseM   # cập nhật tích các ma trận nghịch đảo M
    else:
        case1 = False  # Trường hợp đặc biệt thứ 1
        # Tìm A[k, j] : tìm phần tử khác 0
        for j = 1, 2, ..., k − 2 :
            if A[k, j]! = 0 :
                + Hoán vị cột j và cột k-1 của ma trận A
                + Hoán vị hàng j và hàng k-1 của ma trận A
                + Hoán vị cột j và cột k-1 của ma trận back
                k = k + 1 #tiếp tục biến đổi theo trường hợp lí tưởng
                case1 = True
                break

if not case1: # Trường hợp đặc biệt thứ 2:không có phần tử nào khác 0
        # Khử cột từ k đến m-1 của ma trận A^(k2)
            #Triệt tiêu ma trận A(k2) với phép biến đổi đồng dạng
            for j = k, k + 1, ..., m − 1 :
                + Tìm S, S1 khử cột j của ma trận A
                + A = S * A * S1
                + back = back * S1
            #case2_1 = False
            #Trường hợp 2.1
            for j = k − 1, k − 2, ..., 1 :
            # Tìm phần tử b khác 0 đầu tiên của cột cuối
                if A[j, m]! = 0 :
                    + Tìm ma trận U va U nghịch đảo
                    Gán A = U * A * U' bắt đầu vòng lặp mới


            #Trường hợp 2.2
```

Với $b_{j,n}^{(k)} = 0 \ \forall \ j = \overline{1, n-k}$, khi đó, ma trận $A^{(k)}$ có dạng:

$$A^{(k)} = \begin{bmatrix} A^{(k1)} & 0 \\ 0 & A^{(k3)} \end{bmatrix}$$

Có $A^{(k3)}$ là ma trận frobenius , tìm trị riêng của $A^{(k3)}$ tìm vecto riêng của $A^{(k3)}$

```
if case2_1 == False:
    X = A[k : m, k : m]
     t = X.shape[0]
    lamda = findValue(X)
    list_eigenvalues.append(lamda)
    + Với mỗi trị riêng trong lamda:
        + Tìm vector riêng y của ma trận Frobenius X
        + Tìm vector riêng x của ma trận A ban đầu
        list_eigenvectors.append(x)
#Biến đổi A(k1) về frobenius
    m = k
    A = A[1:m, 1:m]


    back = eye(m)
k = k – 1
```

Sau khi hết vòng lặp , ta thực hiện tìm trị riêng và các vecto riêng tương ứng

```
lam = findValue(A)
list_eigenvalues.append(lam)
+ Với mỗi trị riêng trong lam:

    + Tìm vector riêng y của ma trận Frobenius A
    + Tìm vector riêng x của ma trận A ban đầu
    list_eigenvectors.append(x)
```

4.2 Đánh giá

Sai số

Phương pháp Danilevsky là phương pháp tìm đúng ma trận Frobenius P đồng dạng với ma trận A và ma trận M để đưa A về dạng Frobenius (tức là $MAM^{-1} = P$). Cho nên trong quá trình biến đồng dạng ma trận A chỉ có sai số tính toán. Ta có thể giảm đi sai số bằng cách như chọn ra trong tập $\{a^i_{(n-i)j}\}$ với $j = \overline{\mathbf{1}, \boldsymbol{n-\iota-1}}$ phần tử lớn nhất, và thực hiện phép biến đổi chỗ hai phần tư như trong trường hợp $a^i_{(n-i)(n-i-1)} = 0$. Đối với việc tìm nghiệm của đa thức đặc trưng thì như kết qua đã có, ta có thể tìm nghiệm thực của đa thức với sai số nhỏ hơn $\epsilon > 0$ tùy ý.

Độ phức tạp

- Đối với bài toàn đi tìm trị riêng, trong mỗi bước biến đổi đồng dạng ma trân A, ta thực hiện nhiều nhất 2 phép nhân ma trận cấp n. Do đó trong tình huống xấu nhất, khối lượng tính toán là:

$$O(n^4) .$$

- Đối với bài toán tìm vector riêng, trường hợp xấu nhất là ta liên tiếp gặp phải trường hợp $a^i_{(i-1)j} = 0 \ \forall j < n - i - 1$ và phải thực hiện lại quá trình từ đầu. Do đó khối lượng tính toán là:

$$O(n^5)$$

Trong các trường hợp còn lại thì khối lượng tính toán để tìm ra vector riêng chỉ tương đương với bài toán đi tìm trị riêng

$$(O(n^4))$$

Như vậy, so sánh với phương pháp đi tìm trị riêng và vector riêng thông thường, khối lượng tính toán của phương pháp Danilevsky được giảm xuống để chạy với thời gian đa thức.

5. Ví dụ

Ví dụ 1

Ma trận A:

1 2 3 4

2 1 2 3

3 2 1 2

4 3 2 1

Case lí tưởng, vị trí đang xét tại dòng thứ: 3

[-5. -2.5 1.5 2.5]

[-2. -2. 1. 2.]

[-24. -15. 11. 19.]

[0. 0. 1. 0.]

Case lí tưởng, vị trí đang xét tại dòng thứ: 2

[-1.00000000e+00 1.66666667e-01 -3.33333333e-01 -6.66666667e-01]

[6.00000000e+00 5.00000000e+00 3.40000000e+01 2.40000000e+01]

[1.33226763e-15 1.00000000e+00 0.00000000e+00 0.00000000e+00]

[0.00000000e+00 0.00000000e+00 1.00000000e+00 0.00000000e+00]

Case lí tưởng, vị trí đang xét tại dòng thứ: 1

[4.00000000e+00 4.00000000e+01 5.60000000e+01 2.00000000e+01]

[1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]

[2.22044605e-16 1.00000000e+00 -7.54951657e-15 -5.32907052e-15]

[0.00000000e+00 0.00000000e+00 1.00000000e+00 0.00000000e+00]

Dòng thứ: 0

[4.00000000e+00 4.00000000e+01 5.60000000e+01 2.00000000e+01]

[1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]

[2.22044605e-16 1.00000000e+00 -7.54951657e-15 -5.32907052e-15]

[0.00000000e+00 0.00000000e+00 1.00000000e+00 0.00000000e+00]

Vector riêng ứng với trị riêng: -3.414213562630226

[-1.   ]

[-0.41421356]

[0.41421356]

[1.   ]

Vector riêng ứng với trị riêng: -1.0990195132776432

[1.   ]

[-1.2198039]

[-1.2198039]

[1.   ]

Vector riêng ứng với trị riêng: -0.585786437626901

[-1.   ]

[2.41421356]

[-2.41421356]

[1.   ]

Vector riêng ứng với trị riêng: 9.099019513534778

[1.   ]

[0.8198039]

[0.8198039]

[1.   ]

Kiểm tra kết quả:

lamda = -3.414213562630226

A.X - lamda.X = 0:

 [-3.51158924e-09]

 [5.61855629e-09]

 [-1.40463241e-09]

 [6.21724894e-15]

lamda = -1.0990195132776432

A.X - lamda.X = 0:

 [-6.36461772e-10]

 [1.01834208e-09]

 [-2.54584354e-10]

 [1.33226763e-15]

lamda = -0.585786437626901

A.X - lamda.X = 0:

 [7.77156117e-15]

 [-1.55431223e-14]

 [2.88657986e-15]

 [-1.44328993e-15]

lamda = 9.099019513534778

A.X - lamda.X = 0:

 [1.19487549e-08]

 [-1.91179979e-08]

 [4.77939821e-09]

 [7.10542736e-14]

Ví dụ 2

A =

1 -2 1 2 3

-2 3 4 5 6

0 0 1 -2 1

0 0 1 0 0

0 0 0 1 0

Case lí tưởng, vị trí đang xét tại dòng thứ: 4

[1. -2. 1. 2. 3.]

[-2. 3. 4. 5. 6.]

[0. 0. 1. -2. 1.]

[0. 0. 1. 0. 0.]

[0. 0. 0. 1. 0.]

Case lí tưởng, vị trí đang xét tại dòng thứ: 3

[1. -2. 1. 2. 3.]

[-2. 3. 4. 5. 6.]

[0. 0. 1. -2. 1.]

[0. 0. 1. 0. 0.]

[0. 0. 0. 1. 0.]

Case 2, vị trí đang xét tại dòng thứ: 2

[1. -2. 0. -5. 3.]

[-2. 3. 0. 15. 6.]

[0. 0. 1. -2. 1.]

[0. 0. 1. 0. 0.]

[0. 0. 0. 1. 0.]

Case 2, vị trí đang xét tại dòng thứ: 2

[1. -2.   0.   0. -32.]

[-2.   3.   0.   0. 61.]

[0.   0.   1. -2.   1.]

[0.   0.   1.   0.   0.]

[0.   0.   0.   1.   0.]

Case 2.1, vị trí đang xét tại dòng thứ: 2

[1.   0.   0. -32. -2.]

[0.   1. -2.   1.   0.]

[0.   1.   0.   0.   0.]

[0.   0.   1.   0.   0.]

[-2.   0.   0. 61.   3.]

Case lí tưởng, vị trí đang xét tại dòng thứ: 4

[-4.91803279e-02 0.00000000e+00 0.00000000e+00

-5.24590164e-01 -4.26229508e-01]

 [3.27868852e-02 1.00000000e+00 -2.00000000e+00 1.63934426e-02 -4.91803279e-02]

 [0.00000000e+00 1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]

 [9.83606557e-02 0.00000000e+00 6.10000000e+01 4.04918033e+00 8.52459016e-01]

 [1.04083409e-16 0.00000000e+00 0.00000000e+00 1.00000000e+00 0.00000000e+00]]

Case lí tưởng, vị trí đang xét tại dòng thứ: 3

[-4.91803279e-02 0.00000000e+00 0.00000000e+00 -5.24590164e-01 -4.26229508e-01]

 [3.60118248e-02 1.00000000e+00 -3.27868852e-02 1.49153453e-01 -2.12308519e-02]

 [-4.83740930e-03 6.10000000e+01 4.04918033e+00 8.00859984e-01 -4.19242139e-02]

 [-6.50521303e-18 0.00000000e+00 1.00000000e+00 0.00000000e+00 0.00000000e+00]

 [1.04083409e-16 0.00000000e+00 0.00000000e+00 1.00000000e+00 0.00000000e+00]]

Case lí tưởng, vị trí đang xét tại dòng thứ: 2

[-4.91803279e-02 0.00000000e+00 0.00000000e+00 -5.24590164e-01 -4.26229508e-01]

 [2.20179663e+00 5.04918033e+00 -5.24832034e+00 8.25811412e+00 -1.25109591e+00]

 [1.21972744e-19 1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]

 [-6.50521303e-18 0.00000000e+00 1.00000000e+00 0.00000000e+00 0.00000000e+00]

 [1.04083409e-16 0.00000000e+00 0.00000000e+00 1.00000000e+00 0.00000000e+00]]

Case lí tưởng, vị trí đang xét tại dòng thứ: 1

[5.00000000e+00 -5.00000000e+00 8.00000000e+00 -2.00000000e+00 -1.00000000e+00]

[1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]

[5.53969168e-20 1.00000000e+00 2.90740765e-19 -4.57474060e-19 6.93068558e-20]

[-2.95450223e-18 1.49178145e-17 1.00000000e+00 2.43986165e-17 -3.69636564e-18]

[4.72720356e-17 -2.38685032e-16 2.48098786e-16 1.00000000e+00 5.91418503e-17]

Dòng thứ: 0

[5.00000000e+00 -5.00000000e+00 8.00000000e+00 -2.00000000e+00 -1.00000000e+00]

[1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]

[5.53969168e-20 1.00000000e+00 2.90740765e-19 -4.57474060e-19 6.93068558e-20]

[-2.95450223e-18 1.49178145e-17 1.00000000e+00 2.43986165e-17 -3.69636564e-18]

[4.72720356e-17 -2.38685032e-16 2.48098786e-16 1.00000000e+00 5.91418503e-17]

Vector riêng ứng với trị riêng: -0.23606797749982025

[1.61803399e+00]

[1.00000000e+00]

[-6.39787702e-16]

[1.99666672e-15]

[4.78436735e-15]]

Vector riêng ứng với trị riêng: 0.5698402910050216

[-0.82481821]

[-0.35206564]

[-0.02540835]

[-0.04458854]

[-0.07824743]

Vector riêng ứng với trị riêng: 4.2360679774990855

[-6.18033989e-01]

[1.00000000e+00]

[-2.18179316e-13]

[-1.58068003e-14]

[-7.26703420e-13]

Kiểm tra kết quả:

lamda = -0.23606797749982025

A.X - lamda.X = 0:

[8.01581024e-14]

[4.51028104e-14]

[2.12814225e-19]

[-1.68438627e-16]

[3.12610264e-15]

lamda = 0.5698402910050216

A.X - lamda.X = 0:

[1.43145495e-11]

[8.32642288e-12]

[1.38257461e-15]

[-2.76272061e-14]

[5.62425106e-13]

lamda = 4.2360679774990855

A.X - lamda.X = 0:

[7.81095189e-11]

[4.54365434e-11]

[1.09532780e-14]

[-1.51220635e-13]

[3.06255828e-12]

6. Tài liệu tham khảo.

Giáo trình "Giải tích số" - Lê Trọng Vinh -2007

Characteristic Polynomial - California State University, East Bay

Giáo trình “Đại số tuyến tính” – Bùi Xuân Diệu 2019

Eigenvalues and eigenvectors :Wikipedia