บทที่ 9
ค่าเจาะจง (Eigenvalues)

หน้า

บทที่ 9
ค่าเจาะจง (Eigenvalues)

ผู้อ่านต้องทบทวนเรื่องเกี่ยวกับเมตริกซ์ เช่น ดีเทอร์มิแนนท์ และ linear dependence ใน n-space มาก่อนที่จะศึกษาบทนี้

อาจสรุปเกี่ยวกับ linear independence ใน n-space อย่างย่อ ๆ ดังนี้

1. เวคเตอร์ $v_1, \ldots, v_k$ ถูกเรียกว่า linearly independent ถ้า

$$\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_k v_k = 0$$

นั้นเนื่องมาจาก $\alpha_1 = \alpha_2 = \ldots = \alpha_k = 0$ เท่านั้น

2. เมื่อกล่าวว่า n linearly independent vectors $v_1, \ldots, v_n$ เป็น a basis สำหรับ n-space นั้นหมายความว่า ทุก ๆ เวคเตอร์ x ใน n-space สามารถเขียนในรูปของผลบวกเชิงเส้น (linear combination) ในรูปต่อไปนี้ได้เพียงรูปเดียว คือ

$$x = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + a_n v_n$$

เราเรียกค่าคงที่ $\alpha_1, \ldots, \alpha_n$ ว่า component of x with respect to $\{v_1, \ldots, v_n\}$

3. เมตริกซ์ V ขนาด (nxn) เป็น nonsingular ก็ต่อเมื่อคอลัมน์ของ V คือ $v_1, \ldots, v_n$ นั้น linearly independent นั่นคือ $v_1, \ldots, v_n$ เป็น basis ของ n-space

9.1 คุณสมบัติเบื้องต้นของค่าเจาะจง (Eigenvalues หรือ Characteristic Values) และ เวคเตอร์เจาะจง (Eigenvectors หรือ Characteristic Vectors)

เราต้องการหาค่าคงที่ λ ซึ่งจะมี nonzero vector v ซึ่ง

$$\boxed{Av = \lambda v \text{ (A multiplies v into a mutiple of itself)}} \qquad \ldots(1)$$

เมื่อเป็นไปตาม (1) เราเรียก λ ว่า **ค่าเจาะจง** สำหรับ A

เรียก v ว่า **เวคเตอร์เจาะจง** สำหรับ A

และเรียก (λ, v) ว่า **eigenpairs** สำหรับ A

จาก (1) เราได้ว่า

ถ้า v เป็น เวคเตอร์เจาะจง สำหรับ λ แล้ว αv $(\alpha \neq 0)$ ก็เป็น เวคเตอร์เจาะจงสำหรับ λ ด้วย ดังนั้นเราจะไม่ถือว่า eigenpairs (λ, v) และ $(\lambda, \alpha v)$ ต่างกัน

เนื่องจากว่า $0x = 0x$ และ $I_n x = 1x$ สำหรับ x ใด ๆ

ดังนั้น สำหรับ nonzero x ใด ๆ $(0, x)$ คือ eigenpair ของ $0_{n \times n}$

และ $(1, x)$ คือ eigenpair ของ I_n

สำหรับเมตริกซ์ A,,, In q จะต้องหา eigenpairs ได้

พิสูจน์

$$Av = \lambda v, \; v \neq 0$$

$$Av - Xv = 0$$

$$(A - \lambda I_n)v = 0$$

นั่นคือ $Av = \lambda v \Longleftrightarrow (A - \lambda I_n) = 0 \qquad \ldots(2a)$

ดังนั้นเรายังคงหา eigenpairs ได้ ก็ต่อเมื่อ $(A - \lambda I_n)$ เป็น singular นั่นคือ

$$\boxed{p_A(\lambda) = |A - \lambda I_n| = 0} \qquad \ldots(2b)$$

จากคำจำกัดความของดีเทอร์มินแนนท์ แสดงว่า $p_A(\lambda)$ คือ polynomial in λ (ซึ่งมี leading term คือ $[(-1)^n\lambda^n]$) ได้มาจากผลคูณของสมาชิกบน diagonal ของเมตริกซ์ $(A - \lambda I_n)$ นั่นคือ $\prod_{k=1}^{n} (a_{kk} - \lambda)$

เราเรียก $p_A(\lambda)$ ว่า **Characteristic polynomial** ของ A เนื่องจาก $p_A(\lambda)$ เป็น polynomial มี degree n จาก (2b) และทฤษฎีของพีชคณิต สรุปได้ว่า

" สำหรับทุก (n x n) เมตริกซ์ A จะมี ค่าเจาะจง n ค่าเท่านั้น (อาจซ้ำกันหรืออาจเป็นจำนวนเชิงซ้อน) ซึ่งคือราก (roots) ของ $p_A(\lambda)$"

ให้ $\lambda_1, \lambda_2, \ldots, \lambda_n$ แทนรากของ $p_A(\lambda)$ ดังนั้น

$$p_A(\lambda) = (-1)^n(\lambda - \lambda_1)(\lambda - \lambda_2)\ldots(\lambda - \lambda_n) = \prod_{j=1}^{n} (\lambda_j - \lambda) \qquad \ldots(3a)$$

ถ้าให้ $\lambda = 0$ ใน (2a) และ (3a) เราจะเห็นว่า

$$\boxed{|A| = \lambda_1\lambda_2\ldots\lambda_n} \qquad \ldots(3b)$$

(3b) คือ เทอมค่าคงที่ของ $p_A(\lambda)$

ดังนั้นจะหา A^{-1} ได้ ก็ต่อเมื่อ ค่าเจาะจงทุกตัวไม่เป็นศูนย์ (นั่นคือ $|A| \neq 0$)

จาก (1) $\quad Av = \lambda v$

$$\lambda^{-1}A^{-1}A = \lambda^{-1}A^{-1}\lambda v$$

$$\lambda^{-1}v = A^{-1}v$$

ดังนั้น $\quad A^{-1}v = (1/\lambda)v$

$$\boxed{(\lambda,v) \text{ เป็น eigenpair สำหรับ } A \;\longleftrightarrow\; (1/\lambda,v) \text{ eigenpair สำหรับ } A^{-1}}$$

...(4)

ตามทฤษฎี eigenpairs ทั้งหมดสำหรับ A นั้นอาจหาได้โดยเริ่มหาผลเฉลยของ $p_A(\lambda) = 0$ ก่อน แล้วหา nontrivial solutions ของ $(A - \lambda I_n)v = 0$

ตัวอย่าง จงหา eigenpairs ทั้งหมดสำหรับ

$$(a)\ A = \begin{bmatrix} 3 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 3 \end{bmatrix} \qquad (b)\ A^{-1} = \begin{bmatrix} 3/8 & 0 & -1/8 \\ 0 & -1/3 & 0 \\ -1/8 & 0 & 3/8 \end{bmatrix}$$

...(5)

Solution

(a) ใช้ arrow rule เพื่อหา

$$p_A(\lambda) = |A - \lambda I| = \begin{vmatrix} 3-x & 0 & 1 \\ 0 & 3-x & 0 \\ 1 & 0 & 3-x \end{vmatrix}$$

$$= (3-\lambda)^2(-3-\lambda)+(3+\lambda)$$

$$= -(3+\lambda)[(3-\lambda)^2-1]$$

$$= -(3+\lambda)(\lambda^2-6\lambda+8)$$

$$= (-1)^3(\lambda+3)(\lambda-2)(\lambda-4)$$

ดังนั้น ค่าเจาะจงของ A (ตามลำดับจากขนาด (magnitude) ใหญ่ไปหาเล็ก) คือ

$\lambda_1 = 4,\ \lambda_2 = -3\ ,\ \lambda_3 = 2$ ...(6a)

เพื่อหา เวคเตอร์เจาะจงสำหรับ $\lambda_1 = 4$ เราต้องหาผลเฉลยของ $(A - 4I)v = 0$ เพื่อหา nonzero $v' = [v_1, v_2, v_3]$

ใช้ การกำจัดของเกาส์ (Gaussian elimination) เราได้

$$[A - 4I : 0] = \left[\begin{array}{ccc|c} -1 & 0 & 1 & 0 \\ 0 & -7 & 0 & 0 \\ 1 & 0 & -1 & 0 \end{array}\right] \xrightarrow{r_1 + r_3} \left[\begin{array}{ccc|c} -1 & 0 & 1 & 0 \\ 0 & -7 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

$-v_1 + v_3 = 0$

$-7v_2 = 0$

ดังนั้น $v_2 = 0$ และ $v_1 = v_3$

นั่นคือผลเฉลยของ $Av = 4v$ อยู่ในรูป $[\alpha \;\; 0 \;\; \alpha]' = \alpha[1 \;\; 0 \;\; 1]'$ โดยที่ $\alpha \neq 0$

ในทำนองเดียวกัน เวคเตอร์เจาะจงสำหรับ $\lambda_2 = -3$ และ $\lambda_3 = 2$ นั่นคือการคูณ $v_2 = [0 \;\; 1 \;\; 0]'$ และ $v_3 = [1 \;\; 0 \;\; -1]'$ ด้วยค่าคงที่ใด ๆ

ชุดหนึ่งของเวคเตอร์เจาะจงสำหรับ ค่าเจาะจงใน (6a) คือ

$$v_1 = [1 \;\; 0 \;\; 1]', \; v_2 = [0 \;\; 1 \;\; 0]', \; v_3 = [1 \;\; 0 \;\; -1]' \qquad \ldots(6b)$$

(b) แทนที่จะเริ่มต้นทำเช่นเดียวกับ (a) เราจะใช้ (4) และ (6) เพื่อหาค่าตอบ เราได้ว่า $(1/4, v_1)$, $(-1/3, v_2)$, และ $(1/2, v_3)$ เป็น eigenpairs ของ A^{-1}

<u>หมายเหตุ</u> ให้ผู้อ่านตรวจสอบว่า $A^{-1}v = (1/\lambda_i)v_i$ สำหรับ $i = 1,2,3$

วิธีหา ค่าเจาะจงจากรากของ $p_A(\lambda)$ นั้นทำได้ยากเมื่อ n มีขนาดใหญ่ วิธีอื่นในการหาจะได้กล่าวต่อไป

9.1B <u>Similar Matrices และ Diagonalizability</u>

Diagonal matrix: D

$$D = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & \lambda_n \end{bmatrix} \qquad \ldots(7a)$$

$De_j = \lambda_j e_j$ โดยที่ $e_j = col_j I_n = [0 \ldots 0 \underset{\uparrow}{1} 0 \ldots 0]' \quad \ldots(7b)$

$j = 1,\ldots, n$ สมาชิกตัวที่ j

ดังนั้น eigenpairs ของ D คือ (λ_1,e_1) $(\lambda_2,e_2),\ldots(\lambda_n,e_n)$ เราจะใช้ผลนี้ในการหาวิธีที่จะแปลงรูป (nxn) เมตริกซ์ A ให้อยู่ในรูป diagonal matrix ซึ่งมี ค่าเจาะจงเดียวกัน แต่เราจะใช้ elementary operations เพื่อ reduce A --> D นั้นไม่เหมาะสมเพราะ SCALE และ SUBTRACT operations นั้นจะทำให้ค่าของ ค่าเจาะจงเปลี่ยนไป สิ่งที่เราต้องการคือ **similarity transformations** $A \to V^{-1}AV$ ซึ่งจะอธิบายต่อไป

เมตริกซ์ (n x n) 2 เมตริกซ์คือ **A** และ **B** ถูกเรียกว่า **similar** ถ้ามี nonsingular matrix **V** ซึ่งทำให้ $B = V^{-1}AV$ (หรือ $A = VBV^{-1}$)

ในกรณีนี้ $Av = \lambda v \iff VBV^{-1}v = Xv \iff B(V^{-1}v) = \lambda(V^{-1}v)$ ดังนั้น similar matrices มีค่าเจาะจงที่เหมือนกัน (identical) จริง ๆ แล้ว

(λ,v) คือ an eigenpair for A <-->
$(\lambda,V^{-1}v)$ คือ an eigenpair for $V^{-1}AV$

$\ldots(8)$

(n x n) เมตริกซ์ A ถูกเรียกว่า **diagonalizable** ถ้าเมตริกซ์ **A similar** กับ a diagonal matrix

Diagonalization Theorem

A เป็นเมตริกซ์ขนาด (n x n)

$V^{-1}AV = D = diag(\lambda_1,\ldots, \lambda_n)$ โดยที่ $V = [v_1 : v_2 : \ldots : v_n]$

$\ldots(9a)$

ก็ต่อเมื่อคอลัมน์ $v_1,\ldots, v_n$ ของ V เป็น เวคเตอร์เจาะจง (สำหรับ $\lambda_1,\ldots, \lambda_n$ ตามลำดับ) ซึ่งเป็น **basis** ของ n-space

สิ่งนี้หมายความว่าทุก ๆ เวคเตอร์ x อาจถูกแสดงในรูปเดียวคือรูปผลบวกเชิงเส้นดังนี้

$x = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$ โดยที่ $Av_j = \lambda_j v_j$ สำหรับ $j = 1,\ldots,n$

$\ldots(9b)$

ผลจาก Diagonalization Theorem คือ ทุก ๆ เมตริกซ์ซึ่งมี ค่าเจาะจง เป็นเลขจำนวนจริงที่ต่างกันหมดนั้นเป็น diagonalizable แต่อย่างไรก็ดียังมีเมตริกซ์ซึ่งมี ค่าเจาะจงซ้ำกันบ้าง(ไม่ต่างกันหมด) ที่เป็น diagonalizable

<u>ตัวอย่าง</u> จากเวคเตอร์เจาะจง ใน (6b)

$$
\mathrm{v} = [\mathbf{v}_1 : \mathbf{v}_2 : \mathbf{v}_3] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \quad \mathbf{V}^{-1} = \begin{bmatrix} 1 & 0 & 1/2 \\ \acute{0}2 & 1 & 0 \\ 1/2 & 0 & -1/2 \end{bmatrix}
$$

...(10a)

ให้ผู้อ่านตรวจสอบว่า $\mathbf{VV}^{-1} = \mathrm{I}_3$ และ

$$
\mathbf{V}^{-1}\mathbf{AV} = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \\ 1/2 & 0 & -1/2 \end{bmatrix} \begin{bmatrix} 3 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & \mathrm{o} & -3 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 2 \end{bmatrix}
$$

$$
= \mathrm{diag}(4, -3, \quad) = \mathrm{D}
$$

...(10b)

ถ้าเราทำจนได้ (10b) โดยเราไม่ทราบ eigenpairs ของ A มาก่อน เราอาจใช้ Diagonalization Theorem เพื่อสรุปว่า

$$
\mathbf{Av}_1 = 4\mathbf{v}_1, \ \mathbf{Av}_2 = -3\mathbf{v}_2 \text{ และ } \mathbf{Av}_3 = 2\mathbf{v}_3
$$

...(10c)

<u>ตัวอย่าง</u> จงแสดงว่าเมตริกซ์ $\mathbf{A} = \begin{bmatrix} 5 & 4 \\ -1 & 1 \end{bmatrix}$ ไม่เป็น diagonalizable

<u>Solution</u> เริ่มด้วยการหา $p_A(\lambda) = \left| A - \lambda I_2 \right|$ ก่อนดังนี้

$$p_A(\lambda) = \begin{vmatrix} 5-x & 4 \\ -1 & 1-A \end{vmatrix} = (5-\lambda)(1-\lambda)-(-1)(4)$$

$$= \lambda^2 - 6\lambda \ t \ 9$$

$$= (\lambda-3)^2$$

ดังนั้น $v = [v_1 \quad v_2]' \neq 0$ เป็น เวคเตอร์เจาะจงของ A ถ้า $(A - 3I_2)v = 0$ นั่นคือ

$$\begin{bmatrix} 2 & 4 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \;\langle\text{---}\rangle\; v_1 = -2v_2 \;\langle\text{---}\rangle\; v = \alpha[2 \quad -1]'$$

ดังนั้น เวคเตอร์เจาะจงทั้งหมดของ A คือ ค่าคงที่ที่ไม่ใช่ศูนย์คูณกับ [2 -1]' เพราะว่า **two** linearly independent vectors ต้องถูกใช้สำหรับสร้าง **basis** หนึ่ง แต่ เวคเตอร์เจาะจงของ A ไม่เป็น **basis** ดังนั้นโดย Diagonalization Theorem **A** ไม่เป็น diagonalizable

9.2 Power Method

ถ้าชุดของ ค่าเจาะจงของ **A** คือ **x** = [2 -4 4 -1/2]'
4 และ -4 เป็น **dominant** (largest magnitude) component ของเวคเตอร์ **x** และ -1/2 เป็น **least dominant**
เรียง ค่าเจาะจงตามลำดับตามขนาดจากใหญ่ไปหาเล็ก นั่นคือ

$$|\lambda_1| >= |\lambda_2| >= |\lambda_3| >= \ldots >= |\lambda_n| \qquad \ldots(1)$$

ดังนั้น λ_1 จะเป็น **dominant**

(λ_1, v_1) จะเป็น **dominant eigenpair**

เมื่อข้อสมมุติ 2 ข้อต่อไปนี้เป็นจริง จะได้แสดงวิธีการหา dominant eigenpair ต่อไป

ข้อสมมุติ 1 $|\lambda_1|$ ต้องใหญ่กว่า $|\lambda_i|$, $i = 2,\ldots,n$

ข้อสมมุติ 2 A มี เวคเตอร์เจาะจง $v_1,\ldots, v_n$ (โดยที่ $Av_i = \lambda_i v_i$ สำหรับทุกค่าของ i) ซึ่งเป็น basis ของ n-space

9.2A Power Method สำหรับการหาค่าเจาะจงที่มีขนาดใหญ่ที่สุด (Dominant Eigenvalues)

Power Method คือขั้นตอนวิธีของการทำซ้ำที่ทำการคูณค่าเดาเริ่มต้น x_0 ด้วย successively higher powers ของ A:

$$\boxed{\text{เลือก } x_0 \text{ แล้วหา } x_{k+1} = Ax_k \text{ สำหรับ } k = 0,1,2,\ldots} \qquad \ldots(2a)$$

ดังนั้น $x_1 = Ax_0,\ x_2 = Ax_1 = A^2x_0,\ x_3 = Ax_2 = A^3x_0,\ldots$

$$\boxed{x_k = A^k x_0 \text{ สำหรับ } k = 1,2,3,\ldots} \qquad \ldots(2b)$$

จากข้อสมมุติ 1 x_0 อาจถูกเขียนแทนได้ด้วย

$$x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$$

แต่ $Av_i = \lambda_i v_i$

ดังนั้น

$$\boxed{A^k v_i = \lambda_i^k v_i \text{ สำหรับ } k >= 1}$$

จากข้อสมมุติ 1

$$A^k x_0 = \alpha_1 \lambda_1^k v_1 + \alpha_2 \lambda_2^k v_2 + \ldots + \alpha_n \lambda_n^k v_n \qquad \ldots(3a)$$

$$= \lambda_1^k [\alpha_1 v_1 + \alpha_2 (\lambda_2/\lambda_1)^k v_2 + \ldots + \alpha_n (\lambda_n/\lambda_1)^k v_n] \qquad \ldots(3b)$$

$$\approx \lambda_1^k \alpha_1 v_1 \text{ สำหรับ } k \text{ ใหญ่ ๆ และ } \alpha_1 \neq 0 \qquad \ldots(3c)$$

เพราะว่า $\lambda_1^k \alpha_1 v_1$ เป็น scalar multiple ของ v_1, $x_k = A^k x_0$ จะมีค่าเข้าใกล้ เวคเตอร์เจาะจงสำหรับค่าเจาะจงที่มีขนาดใหญ่ที่สุด λ_1 (นั่นคือ $Ax_k \approx \lambda_1 x_k$) ดังนั้นถ้าทำให้ dominant component ของ x_k เป็น 1 แล้ว

(dominant component ของ Ax_k) $\approx$ [λ_1 คูณกับ (dominant component ของ x_k)] $= \lambda_1$ ...(3d)

Scaled Power **Method Algorithm** นั้นเป็นไปตามสูตร (3) ขั้น {scale} ทำให้ dominant component ของ current x เป็น 1 ดังนั้นเมื่อการทดสอบเพื่อหยุดการทำซ้ำเป็นจริง จาก (3d) เราได้ว่า current BigXi จะประมาณค่าค่าเจาะจงที่มีขนาดใหญ่ที่สุด λ_1 และจาก (3c) current x จะประมาณเวคเตอร์เจาะจงสำหรับ λ_1

Algorithm: Scaled Power Method

Purpose: **To** find the dominant eigenpair $(\lambda_1, \mathbf{v}_1)$ of a given $n \times n$ matrix A *to* *NumSig* significant digits.

{initialize}
GET n, A, $\mathbf{x}_0$, {initial nonzero guess of $\mathbf{v}_1$}
NumSig, MaxIt (termination parameters)
$Tol \leftarrow 10^{-NumSig}$; $\mathbf{x} \leftarrow \mathbf{x}_0$ {$\mathbf{x} = [x_1\ x_2\ \cdots\ x_n]^T$ is the current $\mathbf{x}_k$}

{iterate}
DO FOR $k = 1$ TO *MaxIt* UNTIL **termination test** is **satisfied**
BEGIN
$\mathbf{x}_{new} \leftarrow Ax$; $BigXi \leftarrow \max_{1 \le i \le n} (|i\text{th component of } \mathbf{x}_{new}|)$
{scale} $\mathbf{x}_{new} \leftarrow \dfrac{1}{BigXi}\mathbf{x}_{new}$ {dominant component of $\mathbf{x}_{new}$ is now 1}
$\mathbf{dx} \leftarrow \mathbf{x}_{new} - \mathbf{x}$
{update} $\mathbf{x} \leftarrow \mathbf{x}_{new}$
{termination test: $|dx_i| \le Tol * \max(1, |x_i|)$, $i = 1, 2, \ldots, n$}
END

IF **termination test succeeded**
THEN OUTPUT (Dominant eigenpair, to *NumSig* digits, is $(BigXi, \mathbf{x})$)
ELSE **OUTPUT** (Convergence did not *occur* in *MaxIt* iterations)

รูป 9.2-1 รหัสเทียมสำหรับขั้นตอนวิธี Scaled Power

ตัวอย่าง จงทำ 4 iterations ของ Scaled Power Method สำหรับเมตริกซ์ A ใน (5) ของหัวข้อ 9.1A โดยเริ่มต้นด้วย x_o = [0 1 21”

Solution โดยการใช้สัญลักษณ์ Ax_k = BigXi x_{k+1} เราได้

$$k = 0:\ Ax_o = \begin{bmatrix} 3 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 3 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ 6 \end{bmatrix} = 6\begin{bmatrix} 1/3 \\ -1/2 \\ 1 \end{bmatrix}\ \text{C=BigXi } x_1]$$

$$k = 1:\ Ax_1 = \begin{bmatrix} 3 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 3 \end{bmatrix}\begin{bmatrix} 1/3 \\ -1/2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3/2 \\ 10/3 \end{bmatrix} = (10/3)\begin{bmatrix} 3/5 \\ 9/20 \\ 1 \end{bmatrix}$$

C=BigXi x_2]

ทำซ้ำต่ออีก 2 ครั้ง คือ Ax_2 = BigXi x_3 และ Ax_3 = BigXi x_4:

$$Ax_2 = \begin{bmatrix} 14/5 \\ -27/20 \\ 18/5 \end{bmatrix} = (18/5)\begin{bmatrix} 7/9 \\ -3/8 \\ 1 \end{bmatrix}$$

และ

$$Ax_3 = \begin{bmatrix} 10/3 \\ 9/8 \\ 34/9 \end{bmatrix} = (34/9)\begin{bmatrix} 15/17 \\ 81/272 \\ 1 \end{bmatrix}$$

จะเห็นว่า BigXi --> 4 = λ_1 และ x_k --> [1 0 1]' = v_1

(ดู (6)ของหัวข้อ 9.1)

โปรดสังเกตว่าสมาชิกตัวที่ 2 ของ x_k นั้นมีเครื่องหมายสลับไปมา ขณะที่ขนาดลดลงเรื่อย ๆ แสดงว่ามีการลู่เข้าหาศูนย์ สมาชิกที่มีเครื่องหมายสลับไปมาแต่ขนาดคงที่จะแสดงว่า (1) fails (นั่นคือ $|\lambda_1|=|\lambda_2|$) ในกรณีนี้วิธีการยังคงให้ค่า λ_1 แต่จะไม่ให้ v_1

9.2B การพิจารณาการลู่เข้า

จาก (3b)จะเห็นว่าการลู่เข้าของ x_k เข้าสู่ a scalar multiple ของ v_1 จะเป็นไปอย่างรวดเร็ว เมื่อ $|\lambda_1| >> |\lambda_i|$ (นั่นคือ $\lambda_i/\lambda_1 \approx 0$) สำหรับ $i > 1$

การลู่เข้าช้าของสมาชิกตัวที่ 2 ในตัวอย่างข้างต้นเป็นผลมาจากความจริงที่ว่า $\lambda_2/\lambda_1 = (-3)/4 = -0.75$

ถ้าเราไม่ทราบค่าโดยประมาณของ v_1 เราจะใช้ x_0' = [1 1 ... 1] แต่การกระทำดังกล่าวอาจทำให้ได้ $\alpha_1 = 0$ ถ้าเหตุการณ์ดังกล่าวเกิดขึ้น Power Method จะให้ค่า λ_2 และ v_2

ถ้า $|\lambda_1|=|\lambda_2|$ (นั่นคือ $\lambda_1 = \pm\lambda_2$ หรือ λ_1 และ λ_2 เป็นค่าสังยุค)

Power Method จะล้มเหลวไม่ว่าจะ Scaled หรือไม่ สังเกตได้จากความผิดปกติของ component บางตัวของ x_k ถ้าเป็นดังกล่าวต้องมีการปรับปรุงวิธีการ หรือใช้วิธีการที่ยุ่งยากซับซ้อนขึ้นเช่นใช้ QR algorithm (9.3E)

9.2C Inverse Power Method สำหรับการหาค่าเจาะจงที่มีขนาดเล็กที่สุด (Smallest Eigenvalues)

จาก (4) ใน 9.1A ถ้า A เป็น nonsingular matrix

(λ, v) **คือ eigenpair สำหรับ A <---> $(1/\lambda, v)$ คือ eigenpair สำหรับ A^{-1}**

...(4)

ดังนั้น ค่าเจาะจงที่มีขนาดใหญ่ที่สุดของ A^{-1} คือ λ_n^{-1}

ดังนั้นถ้า $\lambda_n \neq 0$ และ $|\lambda_n| < |\lambda_i|$ สำหรับ $i \neq n$ เราสามารถหา $1/\lambda_n$ และเวคเตอร์เจาะจง v_n โดยการใช้ Scaled Power Method กับ A^{-1} นั่นคือโดยการทำซ้ำขั้น

$$\boxed{x_{k+1} = (1/BigXi)A^{-1}x_k}$$ (BigXi = **dominant** component **ของ $A^{-1}x_k$**)

...(5)

จนกระทั่ง $x_{k+1} \approx v_n$ และ BigXi $\approx 1/\lambda_n$

ถ้า A **เป็น** singular **เราจะหา A^{-1} ไม่ได้ แสดงว่า** $\lambda_n = 0$

การใช้ Scaled Power Method **กับ A^{-1} เราเรียกว่า** Inverse Power **Method สำหรับหา** least dominant eigenpair (λ_n, v_n) **สำหรับ** A

<u>ตัวอย่าง</u> จงหา least dominant eigenpair **สำหรับ** A **ถ้า**

$$A = \begin{bmatrix} 3 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 3 \end{bmatrix} \quad \text{ซึ่งมี } A^{-1} = \begin{bmatrix} 3/8 & 0 & -1/8 \\ 0 & -1/3 & 0 \\ -1/8 & 0 & 3/8 \end{bmatrix} \qquad ...(6)$$

โดยใช้ Inverse Power Method **เริ่มต้นด้วย x_0** = co 1 21'

<u>Solution</u>

โดยใช้สัญลักษณ์ $A^{-1}x_k$ = BigXi x_{k+1} **สำหรับ** k = **0,1,2,.. เราได้**

$$A^{-1}x_0 = \begin{bmatrix} 3/8 & 0 & 1/8 \\ 0 & -1/3 & 0 \\ -1/8 & 3/8 & 3/8 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} -114 \\ -1/3 \\ 3/4 \end{bmatrix} = (3/4) \begin{bmatrix} -113 \\ -4/9 \\ 1 \end{bmatrix}$$

$$= \text{BigXi } x_1$$

$$
A^{-1}x_1 = \begin{bmatrix} 3/a & 0 & 1/8 \\ 0 & -1/3 & 0 \\ -1/8 & 3/8 & 3/8 \end{bmatrix} \begin{bmatrix} -1/3 \\ -4/9 \\ 1 \end{bmatrix} = \begin{bmatrix} -1/4 \\ 4/27 \\ 5/12 \end{bmatrix} = (5/12) \begin{bmatrix} -3/5 \\ 16/45 \\ 1 \end{bmatrix}
$$

$$
= \text{BigXi } x_2
$$

$$
A^{-1}x_2 = \begin{bmatrix} 3/8 & 0 & 1/8 \\ 0 & -1/3 & 0 \\ -1/8 & 3/8 & 3/8 \end{bmatrix} \begin{bmatrix} -3/5 \\ 16/45 \\ 1 \end{bmatrix} = \begin{bmatrix} -7/20 \\ -16/135 \\ 9/20 \end{bmatrix} = (9/20) \begin{bmatrix} -7/9 \\ -64/243 \\ 1 \end{bmatrix}
$$

$$
= \text{BigXi } x_3
$$

BigXi 's: 3/4, 5/12, 9/20 เข้าสู่ 1/2 (ส่วนกลับของ $\lambda_3 = 2$)

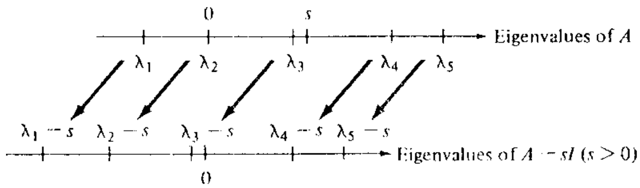
และ x_k เข้าสู่ a scalar multiple ของ เวคเตอร์เจาะจง v_3 = Cl 0 -11'

9.2D Shifting Eigenvalues

สำหรับค่าคงที่ใด ๆ s

$$
Av = \lambda v \ \langle --- \rangle \ (A - sI)v = (\lambda - s)v \qquad \ldots (7a)
$$

ดังนั้นจากรูป 9.2-12



รูป 9.2-Z ค่าเจาะจงของ A และ A - sI (s > 0)

(λ, v) คือ eigenpair สำหรับ A $\langle --- \rangle$ $(\lambda - s, v)$ คือ eigenpair สำหรับ A - sI

...(7b)

ให้ λ_s เป็น ค่าเจาะจงของ **A** ที่อยู่ใกล้ s และให้ v_s เป็น เวคเตอร์เจาะจงที่สมนัยกับ λ_s

แล้ว $(\lambda_s - s, v_s)$ จะเป็น least dominant eigenpair สำหรับ A - sI ดังนั้น $((\lambda_s - s)^{-1}, v_s)$ จะเป็น dominant eigenpair ของ $(A - sI)^{-1}$ solve หา λ_s เราได้

$$\boxed{\lambda_s = 1/[\text{ค่าเจาะจงที่มีขนาดใหญ่ที่สุดของ } (A - sI)^{-1}] + s} \quad \ldots(8)$$

วิธีการหา (λ_s, v_s) จะเรียกว่า Shifted Inverse Pover **Method**

ตัวอย่าง จงใช้ Shifted Inverse Power Method โดยเริ่มด้วย x_o =[1 1 1]' เพื่อหา ค่าเจาะจงที่มีค่าใกล้ s = -5/2 ของ

$$A = \begin{bmatrix} 3 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 3 \end{bmatrix} \text{ ซึ่งมี } A - sI = A + (5/2)I = \begin{bmatrix} 11/2 & 0 & 1 \\ 0 & -1/2 & 0 \\ 0 & 0 & 11/2 \end{bmatrix}$$

Solution

โดยการใช้ Scaled Inverse Power Method กับ A - sI จะได้

$(A + (5/2)I)^{-1}x_o$ = (-2)[-7.69231E-2 1 -7.69231E-2]' = BigXi x_1

$(A + (5/2)I)^{-1}x_1$ = (-2)[-5.91716E-3 1 -5.91716E-3]' = BigXi x_2

$(A + (5/2)I)^{-1}x_2$ = (-2)[-4.55166E-4 1 -4.55166E-4]' = BigXi x_3

จากการพิจารณาว่าสมาชิกตัวแรก และตัวสุดท้ายของ x_k เปลี่ยนเครื่องหมายสลับไปมา และลดค่าลง เราเห็นว่า v_s = [0 1 0]' และ (-2) เป็นค่าเจาะจงที่มีขนาดใหญ่ที่สุดของ $(A + (5/2)I)^{-1}$

ดังนั้นจาก (8) ค่าเจาะจงของ A ใกล้ s = -5/2 คือ

$$\lambda_s = [1/(-2)]+(-5/2) = -3$$

9.3 วิธีการหา Eigenpairs ทั้งหมดของเมตริกซ์หนึ่ง ๆ

ถ้าเมตริกซ์ A ใด ๆ ซึ่งมีขนาด (m X n) แล้ว

A' จะมีขนาด (n x m) โดยที่ ith row คือ $(col_i A)'$, $i=1,\ldots,n$

และ สำหรับ A และ B

$(A')' = A$

$(AB)' = B'A'$ ...(1)

9.3A เมตริกซ์สมมาตร (Symmetric Matrices)

เมตริกซ์ A เป็นเมตริกซ์สมมาตร ถ้า $a_{ij} = a_{ji}$ สำหรับทุก ๆ $i \neq j$ หรือเขียนได้ว่า

เมตริกซ์ A เป็นเมตริกซ์สมมาตร $\longleftrightarrow$ $A' = A$ ...(2)

ตัวอย่างของเมตริกซ์สมมาตร เช่น diagonal matrices, I_n, 0_{nxn}

จาก (1) $(A'A)' = (A')(A')' = A'A$

และทำนองเดียวกัน

$(AA')' = AA'$

ดังนั้น A'A และ AA' ต่างก็เป็นเมตริกซ์สมมาตร สำหรับเมตริกซ์ A ใด ๆ ซึ่งมีขนาด (m x n) ...(3)

วิธีวิเคราะห์เชิงตัวเลขในการหาค่าเจาะจงของเมตริกซ์สมมาตร A ส่วนมากขึ้นอยู่กับความสามารถในการแปลง A ให้อยู่ในรูป diagonal matrix $U^{-1}AU$ โดยที่ U^{-1} นั้นหาได้ง่าย ดังจะได้อธิบายต่อไป

9.3B เมตริกซ์ออร์ธอโกนอล (Orthogonal Matrices)

เมตริกซ์ U ขนาด (n x n) ถูกเรียกว่า ออร์ธอโกนอล ถ้าเมตริกซ์สลับเปลี่ยนของ U (U') คือเมตริกซ์ส่วนกลับของ U (U^{-1}) นั่นคือ

$UU' = U'U = I_n$ ...(4a)

จากค่าจำกัดความของการคูณเมตริกซ์ (4a)หมายความว่า

$$\text{row}_i U\ (\text{row}_j U)' = (\text{col}_i U)'\ \text{col}_j U = \begin{cases} 0 & \text{ถ้า } i \neq j \\ 1 & \text{ถ้า } i = j \end{cases} \qquad \ldots(4b)$$

ถ้า U และ V เป็นออร์ธอโกนอล จาก (1) และ (4a)

$$(UV)'(UV) = (V'U')(UV) = V'(U'U)V = V'V = I"$$

แสดงว่า **ผลคูณของเมตริกซ์ออร์ธอโกนอล เป็น ออร์ธอโกนอล ด้วย**

เมตริกซ์ออร์ธอโกนอลมีคุณสมบัติทางเรขาคณิตที่น่าสนใจคือ

ถ้า $\mathbf{x} = [x_1 \ \ x_2 \ \ldots \ x_n]'$ และ $\mathbf{y} = [y_1 \ \ y_2 \ldots \ y_n]'$ เป็นจุดใน n-space เราให้ค่าจำกัดความต่อไปนี้

$$\mathbf{x}.\mathbf{y} = x'y = x_1y_1 + x_2y_2 + \ldots + x_ny_n$$ Cdot product ของ x และ y] ...(5a)

$$||\mathbf{x}|| = \sqrt{\mathbf{x}.\mathbf{x}} = \sqrt{x_1^2 + \ldots + x_n^2}$$ [Euclidean length ของ xl ...(5b)

$$\angle(\mathbf{x},\mathbf{y}) = \cos^{-1}[\mathbf{x}.\mathbf{y}/||\mathbf{x}||\,||\mathbf{y}||]$$ [มุมระหว่าง x และ y (ถ้า $\mathbf{x},\mathbf{y} \neq 0$)] ...(5c)

แล้วสำหรับเมตริกซ์ออร์ธอโกนอล U กับ (5) คือ

$U\mathbf{x}.U\mathbf{y} = \mathbf{x}.\mathbf{y}$ สำหรับทุก x, y ...(6a)

$||U\mathbf{x}|| = ||\mathbf{x}||$ สำหรับทุก x ...(6b)

$\angle(U\mathbf{x},U\mathbf{y}) = \angle(\mathbf{x},\mathbf{y})$ สำหรับ x และ y ที่ไม่ใช่ 0 ...(6c)

จริง ๆ แล้วเมื่อใช้ (5a), (1) และ (4a) เราพบว่า สำหรับ x และ y ใด ๆ

$$U\mathbf{x}.U\mathbf{y} = (U\mathbf{x})'U\mathbf{y} = \mathbf{x}'U'U\mathbf{y} = x'y = \mathbf{x}.\mathbf{y}$$

การพิสูจน์ (6a), (6b) และ (6c) ทำได้โดยตรงจาก (5b) และ (5c)

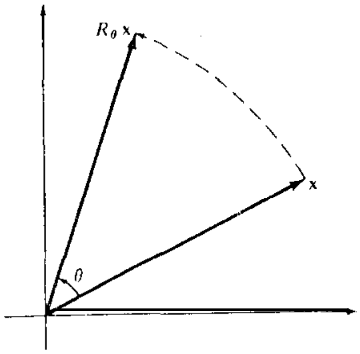
จากค่าจำกัดความ (5) (4b) หมายความว่า rows (หรือ columns) ของเมตริกซ์ออร์ธอโกนอล เป็น unit vectors (นั่นคือมีความยาวเป็น 1) นั้นตั้งฉากซึ่งกันและกัน (mutually perpendicular)

<u>ตัวอย่าง</u> Rotation **Matrices**

Rotation matrix R_θ ซึ่งถูกกำหนดโดย

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad \ldots(7)$$

นั่นสอดคล้องกับ $R_{(-\theta)} = R_\theta' = R_\theta^{-1}$ ดังนั้น R จึงเป็นออร์ธอโกนอล สำหรับมุม θ ใด ๆ



รูป 9.3-1 การคูณ x โดย R_θ ใน 2-space

การคูณ x ด้วย R_θ นั้นผลที่ได้คือ การหมุนทวนเข็มนาฬิกาไปเป็นมุม θ เรเดียน วัดที่จุดกำเนิด ในรูปข้างบนเป็นการหมุนใน 2-space

โดยทั่ว ๆ ไป i,j-rotation matrix ซึ่งกำหนดให้สำหรับ $i < j$ คือเมตริกซ์ขนาด (n x n)

$$R_{ij}(\theta) = \begin{array}{c} \left[\begin{array}{ccccccc} 1 & 0 & \ldots & 0 & \ldots & 0 & \ldots\ 0 \\ \vdots & \vdots & & & & \vdots & \\ 0 & 0 & \ldots & \cos\theta & \ldots & -\sin\theta & \ldots\ 0 \\ \vdots & \vdots & & \vdots & & \vdots & \ldots \\ 0 & 0 & \ldots & \sin\theta & \ldots & \cos\theta & \ldots\ 0 \\ \vdots & \vdots & & \vdots & & \vdots & 0 \\ 0 & 0 & \ldots & \underbrace{0}_{\text{col } i} & \ldots & \underbrace{0}_{\text{col } j} & \ldots\ 1 \end{array}\right] \end{array} \begin{array}{l} \\ \\ \leftarrow \text{row } i \\ \\ \leftarrow \text{row } j \\ \\ \\ \end{array} \qquad \ldots(8)$$

(สมาชิกที่ไม่ได้แสดงไว้นั้นเหมือนกับของ I_n) เป็นออร์ธอโกนอลสำหรับ θ ใด ๆ และ $i < j$

<u>**ตัวอย่าง**</u> เมื่อ $n = 3$

$$R_{13}(\pi/6) = \begin{bmatrix} \cos(\pi/6) & 0 & -\sin(\pi/6) \\ 0 & 1 & 0 \\ \sin(\pi/6) & 0 & \cos(\pi/6) \end{bmatrix} = \begin{bmatrix} \sqrt{3}/2 & 0 & -1/2 \\ 0 & 1 & 0 \\ 1/2 & 0 & \sqrt{3}/2 \end{bmatrix} \qquad \ldots(9)$$

คูณ $x = [x\ y\ z]'$ ด้วย $R_{13}(\pi/6)$ จะทำให้หมุน x ทวนเข็มนาฬิกาโดยทำมุม $\pi/6$ เรเดียนกับแกน-y ใน 3-space

$U^{-1} = U'$ (ถ้า U เป็นออร์ธอโกนอล) ซึ่งหาได้ง่ายสำหรับ n ใด ๆ

ถ้าเมตริกซ์ดังกล่าวปรากฏใน similarity transformation $B = U^{-1}AU$ จะกลายเป็น

$$B = U'AU \qquad \ldots(10)$$

เราเรียก B ว่า orthogonally similar กับ A

<u>Symmetric Diagoaalization Theorem</u>

ถ้า A เป็นเมตริกซ์สมมาตร แล้วจะมีเมตริกซ์ออร์ธอโกนอล U ซึ่งทำให้ $U'AU = \mathrm{diag}(\lambda_1,\ldots,\lambda_n)$ ดังนั้นค่าเจาะจงของ A ทั้งหมดเป็นเลขจำนวนจริง และคอลัมน์ $u_1,\ldots,u_n$ ของ U จะเป็น basis ของ mutually perpendicular unit eigenvectors (หรือเรียกว่า Orthonormal basis of eigenvectors)

<u>**ตัวอย่าง**</u> เมตริกซ์สมมาตร $A = \begin{bmatrix} 3 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 3 \end{bmatrix}$

และ $U = R_{13}(\pi/4)$ เป็นออร์ธอโกนอล

$B = U'AU = \mathrm{diag}(4, -3, 2)$

$$\begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 3 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 0 & 1 & 0 \\ 1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$U' = U^{-1} \qquad A = A' \qquad U = R_{13}(\pi/4) \qquad \mathrm{diag}(4,-3,2)$$

ถ้า A เป็นเมตริกซ์สมมาตร U เป็นเมตริกซ์ออร์ธอโกนอล และ B = U'AU แล้ว

$$B' = U'A'(U')' = U'AU = B$$

ดังนั้นเมตริกซ์ใด ๆ ซึ่ง orthogonally similar กับ เมตริกซ์สมมาตรหนึ่งจะสมมาตรด้วย นั่นคือถ้า $A = (a_{ij})$ เป็นเมตริกซ์สมมาตรและ

$$B = R_{ij}'(\theta)A\,R_{ij}(\theta) \qquad \text{...(11a)}$$

แล้ว B จะสมมาตรด้วย และจากการคูณแสดงว่า

$$b_{ij} = b_{ji} = a_{ij}[\cos^2\theta - \sin^2\theta] + (a_{jj}-a_{ii})\sin\theta\cos\theta \qquad \text{...(11b)}$$

$$b_{ik} = b_{ki} = a_{ik}\cos\theta + a_{jk}\sin\theta$$

สำหรับ $k \neq i, j\ (\mathrm{row}_i B, \mathrm{col}_i B)$ ...(11c)

$$b_{jk} = b_{kj} = -a_{ik}\sin\theta + a_{jk}\cos\theta$$

สำหรับ $k \neq i, j\ (\mathrm{row}_j B, \mathrm{col}_j B)$ ...(11d)

$$b_{ii} = a_{ii}\cos^2\theta + a_{jj}\sin^2\theta + 2a_{ij}\sin\theta\cos\theta \qquad \text{...(11e)}$$

$$b_{jj} = a_{ii}\sin^2\theta + a_{jj}\cos^2\theta - 2a_{ij}\sin\theta\cos\theta \qquad \text{...(11f)}$$

9.3C วิธีของจาโคบี (Jacobi's method) สำหรับเมตริกซ์สมมาตร

ถ้า θ ถูกเลือกเพื่อทำให้

$$a_{ij}\cos(2\theta) + (1/2)(a_{jj}-a_{ii})\sin(2\theta) = 0 \qquad \text{...(12a)}$$

แล้ว (จากการพิจารณา 11b) $B = R_{ij}'(\theta)A\,R_{ij}(\theta)$ จะสอดคล้องกับ

$$b_{ij} = b_{ji} = 0 \qquad \text{...(12b)}$$

θ (ในเรเดียน) ซึ่งสอดคล้องกับ (12a) อาจถูกเขียนในรูป

$$\boxed{\theta = \begin{cases} (1/2)\ \tan^{-1}[2a_{ij}/(a_{ii}-a_{jj})] & \text{ถ้า } a_{ii} \neq a_{jj} \\ \pi/4 & \text{ถ้า } a_{ii} = a_{jj} \end{cases}} \quad \begin{matrix} \ldots(13a) \\ \ldots(13b) \end{matrix}$$

ถ้า **A** เป็นเมตริกซ์สมมาตร และ i และ j เป็นดรรชนีของสมาชิกซึ่งเป็น dominant superdiagonal ของ **A** แล้วจากการพิจารณา (12b) เมตริกซ์ **B** ควรจะเกือบเป็น **diagonal** มากกว่าเมตริกซ์ **A** วิธีการดังกล่าวถูกนำไปใช้ในขั้นตอนวิธี ซึ่งเราเรียกว่า **วิธีของยาโคบิ (Jacobi's method)** ถ้าเราเริ่มต้นด้วย $\mathbf{A}_o = \mathbf{A}$ วิธีการจะให้ **rotation matrices**

$$\boxed{\mathbf{R}_k = \mathbf{R}_{ij}(\theta_k)\ ,\ k = 1,2,\ldots} \qquad \ldots(14)$$

ดังนั้น $\mathbf{A}_k = \mathbf{R}_k'\mathbf{A}_{k-1}\mathbf{R}_k \dashrightarrow \mathbf{D} = \mathrm{diag}(\lambda_1,\ldots,\lambda_n)$ เมื่อ $k \dashrightarrow \infty$

...(15)

เนื่องจากว่า $\mathbf{A}_o = \mathbf{A}$ เราอาจเขียน $\mathbf{A}_k$ ในเทอมของ **A** ดังนี้

$$\boxed{\mathbf{A}_k = \mathbf{U}_k'\mathbf{A}\mathbf{U}_k \text{ โดยที่ } \mathbf{U}_k = \mathbf{R}_1\mathbf{R}_2\ldots\mathbf{R}_k = \mathbf{U}_{k-1}\mathbf{R}_k} \qquad \ldots(16)$$

เมตริกซ์ $\mathbf{U}_k$ ใน (16) คือผลคูณของเมตริกซ์ออร์ธอโกนอล ดังนั้น $\mathbf{U}_k$ จึงเป็นเมตริกซ์ออร์ธอโกนอลด้วย เมื่อ $k \to \infty$ $\mathbf{U}_k$ จะมีค่าเข้าใกล้ **U** ทั้งนี้รับประกันโดย Symmetric Diagonalization Theorem ข้างต้น

โปรดสังเกตว่าวิธีของยาโคบิ จะให้ค่าประมาณของค่าเจาะจงทั้งหมด (สมาชิกบน diagonal ของ $\mathbf{A}_k$) และค่าประมาณของเวคเตอร์เจาะจงทั้งหมด (คอลัมน์ของ $\mathbf{U}_k$)

รูป 9.3-2 แสดงขั้นตอนวิธีของวิธีของยาโคบิ

Algorithm: Jacobi's Method*

Purpose: To find all eigenpairs of a given $n \times n$ matrix A to a prescribed accuracy. The method uses the rotation matrix $R_{ij}(\theta)$ defined in (8) iteratively to form an orthogonal matrix U such that $U^T A U$ = diag (λ_1, , λ_n) to the desired accuracy.

```
{initialize}
GET n, A,                 [matrix parameters}
    MaxIt, NumDec (termination parameters)
U ← I , ;  AbsTol ← 10^(-NumDec)

{iterate}
DO FOR k = 1 TO MaxIt UNTIL termination test is satisfied
  BEGIN
  Get i, j of the dominant superdiagonal entry, a_ij {i < j}
  MaxOffDiag ← a_ij|
  {rotate i, j)
    BEGIN
    {get θ} IF a_ii = a_jj THEN Theta ← tan^-1(1) {= π/4}
                      ELSE Theta ← ½tan^-1[2a_ij/(a_ii − a_jj)]
    R ← R_ij(Theta)    {R is R_k}
    A ← R^T A R        {A is A_k;  a_ij is now zero}
    U ← UR             {U is U_k}
    END
  /termination test: MaxOffDiag < AbsTol}
  END

IF termination test succeeded
  THEN OUTPUT (The eigenpairs of A arc (a_jj, col_j U), j = 1, , n)
  ELSE OUTPUT (Convergence did not occur in MaxIt iterations)
```

รูป 9.3-2 รหัสเทียมสำหรับขั้นตอนวิธีของวิธีของยาโคบิ

ถ้า **A** มีขนาด (2x2) แล้วจะใช้การ rotate เพียง 1 ครั้ง เพื่อทำให้เมตริกซ์ **A** เป็น diagonal เมตริกซ์ อย่างไรก็ดี โดยทั่ว ๆ ไป A_k ใน (15) จะไม่เท่ากับ D ทีเดียว

ในการกล่าวถึง $R_k = R_{ij}(\theta_k)$ จะเป็นการสะดวกที่จะใช้ตัวย่อต่อไปนี้

$$S_k = \sin\theta_k \text{ และ } C_k = \cos\theta_k \text{ , } k = 1,2,\ldots \qquad \ldots(17)$$

<u>**ตัวอย่าง**</u>* จงทำ 2 iterations ของวิธีของยาโคบิ สำหรับเมตริกซ์สมมาตร A

$$A = \begin{bmatrix} 3 & 0.01 & 0.02 \\ 0.031 & 2 & 0.1 \\ 0.02 & 0.1 & 1 \end{bmatrix} \qquad \ldots(18)$$

Solution สมาชิกใน superdiagonal ตัวใหญ่ที่สุดของ A คือ 0.1 = a_{23}

โดย (13a)

$$\theta_1 = (1/2)\tan^{-1}[2a_{23}/(a_{22}-a_{33})]$$

$$= (1/2)\tan^{-1}[2(0.1)/(2-1)$$

$$= 0.0986978 \text{ (เรเดียน)}$$

$$C_1 = \cos\theta_1 = 0.995133$$

และ $S_1 = \sin\theta_1 = 0.0985376$

แทนค่าเหล่านี้ใน

$R_1 = R_{23}(\theta_1)$ เราได้ A_1 ดังนี้

$$A_1 = R_1{}'A_0R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_1 & S_1 \\ 0 & -S_1 & C_1 \end{bmatrix} \begin{bmatrix} 3 & 0.01 & 0.02 \\ 0.01 & 2 & 0.1 \\ 0.02 & 0.1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & C_1 & -S_1 \\ 0 & S_1 & C_1 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 0.011922 & 0.018917 \\ 0.011922 & 2.009902 & 0 \\ 0.018917 & 0 & 0.990098 \end{bmatrix} \qquad \ldots(19)$$

สำหรับการ rotate ครั้งแรก U_1 ก็คือ R_1 นั่นเอง

อย่างไรก็ดี dominant superdiagonal entry ของ A_1 (คือ a_{13} = 0.018917)

นั้นเล็กกว่า dominant superdiagonal entry ของ A_0 (คือ a_{23} = 0.1)

ทำต่อไปเราจะได้

$$\theta_2 = (1/2)\tan^{-1}[2a_{13}/(a_{11}-a_{33})]$$

$$= (1/2)\tan^{-1}[2(0.018917)/(3-0.990098)]$$

$$= 0.00941079$$

$$C_2 = \cos\theta_2 = 0.939956$$

และ $S_2 = \sin\theta_2 = 0.00941065$

แทนค่าเหล่านี้ใน $R_2 = R_{13}(\theta_2)$ เราได้ $A_2 = R_2'A_1R_2$ ดังนี้

$$A_2 = \begin{bmatrix} C_2 & 0 & S_2 \\ 0 & 1 & 0 \\ -S_2 & 0 & C_2 \end{bmatrix} \begin{bmatrix} 3 & 0.011922 & 0.018917 \\ 0.011922 & 2.009902 & 0 \\ 0.018917 & 0 & 0.990098 \end{bmatrix} \begin{bmatrix} C_2 & 0 & -S_2 \\ 0 & 1 & 0 \\ S_2 & 0 & C_2 \end{bmatrix}$$

$$= \begin{bmatrix} 3.000178 & 0.011922 & 0 \\ 0.011922 & 2.009902 & -0.000112 \\ 0 & -0.000112 & -0.989920 \end{bmatrix} \qquad \ldots(20a)$$

และเราได้ $U_2 = U_1R_2 = R_{23}(\theta_1)R_{13}(\theta_2)$ ดังนี้

$$U_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.995133 & -0.098538 \\ 0 & 0.098538 & 0.995133 \end{bmatrix} \begin{bmatrix} 0.999956 & 0 & -0.009411 \\ 0 & 1 & 0 \\ 0.009411 & 0 & 0.999956 \end{bmatrix}$$

$$= \begin{bmatrix} 0.999956 & 0 & -0.009411 \\ -0.000927 & 0.995133 & -0.098533 \\ 0.009365 & 0.098538 & 0.995089 \end{bmatrix} \qquad \ldots(20b)$$

การทำซ้ำครั้งที่ 3 โดยใช้ $R_3 = R_{12}(\theta_3)$ โดยที่ $\theta_3 = 0.012035$ (เรเดียน)
จะให้ค่าของ $A_3 = R_3'A_2R_3$ และ $U_3 = U_2R_3$ ดังนี้

$$A_3 = \begin{bmatrix} 3.00032 & 0 & \underline{-1.35E-6} \\ 0 & 2.00976 & \underline{-1.122E-4} \\ -1.35E-6 & \underline{-1.122E-4} & 0.989920 \end{bmatrix} \qquad \ldots(21a)$$

$$
U_3 = \begin{bmatrix} 0.999883 & -0.0120\underline{355} & -0.009410\underline{8} \\ 0.0110502 & 0.9950\underline{72} & -0.098\underline{5332} \\ 0.01055\underline{03} & 0.0984\underline{177} & 0.9950\underline{89} \end{bmatrix} \qquad \ldots(21b)
$$

ตัวเลขที่ถูกขีดเส้นใต้คือตัวเลขที่จะผิดหลังจากการปัดเศษ ถึงแม้เราไม่ทราบค่าที่แท้จริงของค่าเจาะจง เราทราบจาก Gerschgorin's disk theorem และการตรวจสอบ A_3 ว่า

$$
|\lambda_1 - 3.00032| < 1.4E\text{-}6; \quad |\lambda_2 - 2.00976 < 1.13E\text{-}4; \quad |\lambda_3 - 0.98992| < 1.14E\text{-}4
$$

โปรแกรมย่อยชื่อ ROTATE ในภาษาฟอร์แทรนในรูป 9.3-3 จะทำการ rotate 1 ครั้ง ในวิธีของยาโคบิหลังจากเลือกค่า THETA ในบรรทัดที่ 900 ถึง 1100 [ดู (13)] แล้วใช้ (11) เพื่อแทน A ด้วย R'AR และแทน U ด้วย UR [ทั้งนี้โดยที่ไม่ต้องหา R = R_{ij}(THETA)] ในบรรทัดที่ 1700 ถึง 3100 และ NROTAT ซึ่งใช้นับจำนวนครั้งของการ rotate นั้นจะถูกเพิ่มค่าในบรรทัดที่ 3300

```
00100          SUBROUTINE  ROTATE(I,J)
00200          DIMENSION  A(6,6),  U(6,6)
00300          COMMON  N,  NROTAT,  A,  U
00400    C - - - - - - - - - - - - - - - - - - - - - - - - - - - - C
00500    C  THIS  SUBROUTINE  PERFORMS  THE  ROTATION  REPLACEMENTS  C
00600    C        A  <--  RTRANSPOSE*A*R    AND    U <-- U*R         C
00700    C WHERE  R  =  R[I,J](THETA) MAKES  A(I,J) = A(J,I) = 0     C
00800    C - - - - - - - - - - - - - - - - - VERSION 1:  5/1/81 C
00900          THETA  =  ATAN(1.0)
01000          IF  (ABS(A(I,J)-A(J,J)) .GT. 1.E-6*ABS(A(I,I)))
01100       &     THETA  =  .5*ATAN(2.*A(I,J)/(A(I,I)-A(J,J)))
01200          SIN  =  SIN(THETA)
01300          COS  =  COS(THETA)
01400    C
01500    C  ROTATE:   A  <-- RTRANS*A*R  AND  U  <-- U*R
01600    C
01700          DO  10  K=1,N
01800             UKI = U(K,I)
01900             U(K,I)  =  UKI*COS  +  U(K,J)*SIN
02000             U(K,J) =  -UKI*SIN  + U(K,J)*COS
02100             IF  (K.EQ.I .OR. K.EQ.J)  GOTO  10
02200                A(K,I)  =  A(I,K)*COS  +  A(J,K)*SIN
02300                A(K,J)  =  -A(I,K)*SIN  +  A(J,K)*COS
02400                A(J,K)  =  A(K,J)
02500                A(I,K)  =  A(K,I)
02600       10  CONTINUE
```

รูป 9.3-3 ROTATE: ซับรูทีนสำหรับการหมุนแกนตามวิธีของยาโคบิหนึ่งครั้ง

```
      AII = A(I,I)
      A(I,I) = AII*COS**2 + A(J,J)*SIN**2 + 2.*A(I,J)*SIN*COS
      A(J,J) = AII*SIN**2 + A(J,J)*COS**2 - 2.*A(I,J)*SIN*COS
      A(I,J) = 0.
      A(J,I) = 0.
C
      NROTAT = NROTAT + 1
C
      RETURN
      END
```

9.3D ข้อควรพิจารณาในทางปฏิบัติของวิธีของจาโคบิ

ในภาษาต่าง ๆ ที่ใช้เขียนโปรแกรมนั้น เราอาจใช้ไลบรารีฟังก์ชันเพื่อหาฟังก์ชัน arc tan ($\tan^{-1}$) เช่น ภาษาฟอร์แทรน: ATAN

ภาษาปาสกาล : ARCTAN

ภาษาเบสิค : ATN

ดังนั้นเราอาจใช้ (13) เพื่อหา θ แล้วหา $\sin\theta$ และ $\cos\theta$

นอกจากนั้นเราอาจใช้วิธีต่อไปนี้

IF $a_{ii} = a_{jj}$

THEN $\sin\theta = \cos\theta = 1/\ 2$ $\{\theta = \pi/4\}$ ...(22a)

ELSE BEGIN

$\tan 2\theta = [2a_{ij}/(a_{ii}-a_{jj})]$ $\{-\pi/2 < 2\theta < \pi/2\}$ ...(22b)

$\cos 2\theta = 1/[\ 1+\tan^2 2\theta\ ]$ $\{\cos 2\theta > 0\}$ ...(22c)

$\cos\theta = 1/[\ (1/2)(1+\cos 2\theta)]$ $\{\cos\theta > 0\}$ ...(22d)

$\sin\theta = (1/2)(\tan 2\theta \cos 2\theta)/\cos\theta$

{เครื่องหมายเหมือนกับ $\tan 2\theta$}...(22e)

END

สำหรับ n ใหญ่ ๆ จะเสียเวลามาก(นั่นคือเสียค่าใช้จ่ายมาก)ในการใช้คอมพิวเตอร์เพื่อกวาดตรวจ (scan) สมาชิก $n(n-1)/2$ ตัวบน superdiagonal (หรือ subdiagonal) ของ current A เพื่อหา maximum off-diagonal $|a_{ij}|$ สิ่งที่มักจะทำกันในทางปฏิบัติคือ กำหนดขนาด threshold คือ *Thresh* สำหรับการกวาดตรวจ เราจะทำการ rotate ทุกครั้งเมื่อ $|a_{ij}| >=$ *Thresh* ดังแสดงใน

ขั้นตอนวิธีในรูป 9.3-4 ค่าของ *Thresh* สำหรับการกวาดตรวจใด ๆ จะถูกกำหนดให้มีค่า

$$Fraction * MaxOffdiag$$

โดยที่ *Fraction* คือเลขจำนวนที่มีค่าระหว่าง 0 และ 1 และ *MaxOffdiag* คือขนาดของเทอมใน Off-diagonal ที่ใหญ่ที่สุด ซึ่งไม่ได้ถูกทำให้มีค่าเป็นศูนย์ระหว่างการกวาดตรวจก่อนหน้านั้น ขั้น rotate i,j จะถูกทำในซับรูทีนดังเช่นแสดงในรูป 9.3-3

ถ้าใช้มาตรการนี้กับ $A_{3\times3}$ ของตัวอย่างในหัวข้อ 9.3C โดยให้ *Fraction*=0.3 เราจะได้ผลลัพธ์ดังแสดงในรูป 9.3-5 eigenpairs ทั้งหมดมีความแม่นยำถึง 6s

Algorithm: Jacobi's Method with Thresholds

Purpose: To perform Jacobi's method more efficiently by using fewer superdiagonal scans and more rotations per scan. The parameter Fraction (between 0 and I) controls the frequency of rotations, with the number of rotations per scan increasing as *Fraction* $\to$ *0.*

```
{initialize}
GET n, A, NumDec, MaxIt,        (as in Jacobi's method algorithm)
    Fraction                    (0 < Fraction < 1}
U ← I,,;   AbsTol ← 10^-NumDec
MaxOffDiag ← (largest superdiagonal |a_ij|)   (i < j}

{iterate}
DO FOR k = 1 TO MaxIt UNTIL termination test is satisfied
  BEGIN
   Thresh ← Fraction * MaxOffDiag;   MaxOffDiag ← 0
  DO FOR i = 1 TO n − 1        (scan ith row of A)
    DO FOR j = i + 1 TO n
      BEGIN
       If |a_ij| > Thresh THEN rotate i, j (as in Jacobi's method)
      IF |a_ij| > MaxOffDiag THEN MaxOffDiag ← |a_ij|
      END
  (Now MaxOffDiag = largest unrotated a_ij | scanned)
  (termination test: MaxOffDiag < AbsTol}
  END

IF termination test succeeded
  THEN OUTPUT (The eigenpairs of A are (a_jj, col_j U), j = I, , n)
  ELSE OUTPUT (Convergence did not occur in MaxIt iterations)
```

รูป 9.3-4 รหัสเทียมสำหรับขั้นตอนวิธีของวิธีของยาโคบิโดยมี thresholds

JACOBI'S METHOD WITH THRESHOLDS (FRACT= 0.300) FOR THE MATRIX

3.00000	0.01000	0.02000
0.01000	2.00000	0.10000
0.02000	0.10000	1.00000

SCAN # 1 (THRESH = 0.030000): 1 ROTATIONS. [A : U] IS:

3.0000000	0.0119221	0.0189173	1.0000000	0.0000000	0.0000000
0.0119221	2.0099019	0.0000000	0.0000000	0.9951333	-0.0985376
0.0189173	0.0000000	0.9900980	0.0000000	0.0985376	0.9951333

SCAN # 2 (THRESH = 0.006000): 2 ROTATIONS. [A : U] IS:

3.0003215	-0.0000021	0.0000000	0.9998833	-0.0120387	-0.0094088
-0.0000021	2.0097583	-0.0002277	0.0110524	0.9950612	-0.0986460
0.0000000	-0.0002277	0.9899200	0.0105499	0.0985305	0.9950781

SCAN # 3 (THRESH = 0.000068~: 1 ROTATIONS. [A : u] IS:

3.0003215	-0.0000021	-0.0000000	0.9998833	-0.0120366	-0.0094114
-0.0000021	2.0097583	0.0000000	0.0110524	0.9950832	-0.0984238
-0.0000000	0.0000000	0.9899200	0.0105499	0.0983083	0.9951001

รูป 9.3-5 การกวาดตรวจ 3 ครั้งตามวิธีของยาโคบิโดยมี thresholds

9.3E Factorization Methods

ถ้าต้องการหา eigenpairs ทั้งหมดของเมตริกซ์ที่ไม่สมมาตร A เราจะใช้วิธีของยาโคบิไม่ได้ วิธีที่จะใช้คือวิธี factorization ซึ่งคล้ายกับวิธี LU-factorization ของหัวข้อ 3.3C

วิธีหนึ่งซึ่งมีประสิทธิภาพดีกว่าวิธีอื่นสำหรับการหา eigenpairs ทั้งหมดของเมตริกซ์ A ใด ๆคือ QR-method ซึ่งจะทำใน 2 ขั้นตอนดังนี้

ขั้นที่ 1 แทนที่ A ด้วย similar matrix $P^{-1}AP$ ซึ่งมีสมาชิกมากกว่า 1 row ใต้ main diagonal เป็นศูนย์ทั้งหมด เมตริกซ์ดังกล่าวถูกเรียกว่าอยู่ในรูป upper-Hessenberg nonsingular เมตริกซ์ P ซึ่งจะทำให้ได้สิ่งที่กล่าวมาข้างต้นนั้นอาจถูกหาได้โดยวิธีหนึ่งใน 2 วิธีต่อไปนี้ คือ

(a) Householder's method

วิธีนี้ใช้ (n-2) Householder transformations

$$A_k = P_k' A_{k-1} P_k \text{ โดยที่ } P_k = I - 2u_k u_k' \quad \ldots(23)$$

สำหรับบาง unit vector u_k , $k = 1, \ldots, n-2$ ดังนั้น P_k's เป็นเมตริกซ์สมมาตร และ orthogonal ถ้า $A = A_o$ เป็นเมตริกซ์สมมาตร A_{n-2} จะเป็น tridiagonal และสมมาตรด้วย

(b) Elementary Transformation Method

วิธีนี้ใช้การกำจัดของเกาส์ในการหา (n-2) transformations

$$A_k = E_k A_{k-1} E_k^{-1} \text{ โดยที่ } E_k \text{ คือ elementary matrix} \quad \ldots(24)$$

นั่นคือ E_k คือเมตริกซ์ซึ่ง $E_k A$ นั้นทำให้เกิด elementary row operation หนึ่ง

Elementary Transformation Method นั้นจะไม่ให้ tridiagonal matrix ถ้า A เป็นเมตริกซ์สมมาตร และไม่คล้ายกับ Householder transformations (ซึ่ง$P = P_1 P_2 \ldots P_{n-2}$ เป็นออร์ธอโกนอล) ใน(b) จะใช้ $P = (E_{n-1} \ldots E_2 E_1)^{-1}$ ซึ่งอาจทำให้ $P^{-1}AP$ มีรูปที่แย่ไปกว่า A (คือห่างจาก diagonal เมตริกซ์มากกว่า) อย่างไรก็ดีวิธีนี้จะใช้การคำนวณทางคณิตศาสตร์เพียงประมาณครึ่งหนึ่งของ Householder's method

<u>**ขั้นที่ 2**</u> เริ่มต้นด้วย $A_o = A$ แล้วหา

$$A_k - s_k I = Q_k R_k, \; A_{k+1} = R_k Q_k + s_k I \quad \ldots(25)$$

โดยที่ Q_k เป็นออร์ธอโกนอล R_k คือ upper(right)-triangular matrix และ s_k คือ shifted scalar[ดู (7) ของหัวข้อ 9.2C]ซึ่งถูกเลือกเพื่อทำให้

$$A_{k+1} \longrightarrow \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \text{ อย่างรวดเร็วเมื่อ } k \longrightarrow \infty \quad \ldots(26)$$

รายละเอียดของ QR-method นั้นเกี่ยวข้องกับ factorization method ซึ่งมีชื่อ เรียกว่า LR-method

แบบฝึกหัดบทที่ 9

9.1 ใน (a)-(b) จงหา characteristic polynomial และ eigenpairs ทั้งหมดของ A แล้วจงหา eigenpairs ของ A^{-1} ด้วย

(a) $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ (b) $A = \begin{bmatrix} 4 & 5 \\ 3 & 2 \end{bmatrix}$

9.2 สำหรับ A ใน (a)-(b) ของข้อ 9.1 จงหาเมตริกซ์ V ซึ่งทำให้ $V^{-1}AV = \mathrm{diag}(\lambda_1, \lambda_2)$ ถ้า v นั้นหาได้ และจงตรวจสอบว่า $V^{-1}AV = \mathrm{diag}(\lambda_1, \lambda_2)$

สำหรับเมตริกซ์ A และค่าเดาเริ่มต้น x_0 ในข้อ 9.3 และข้อ 9.4 จงทำ (a)-(c) ต่อไปนี้

(a) จงหา x_1, x_2, x_3, x_4 โดยใช้ Scaled Pouer method

(b) จงหา x_1, x_2, x_3, x_4 โดยใช้ Inverse Power Method (ใช้สูตรสำหรับ A^{-1})

(c) จงหา x_1, x_2, x_3, x_4 โดยใช้ Shifted Inverse method และใช้ shifted scalar s (ใช้สูตรสำหรับ $(A - sI)^{-1}$)

9.3

$A = \begin{bmatrix} 4 & -2 \\ -3 & 5 \end{bmatrix}$, $x_0 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $s = 3$

Eigenpairs: $[\,2, \begin{bmatrix} 1 \\ 1 \end{bmatrix}], [\,7, \begin{bmatrix} -2 \\ 3 \end{bmatrix}]$

9.4 $A = \begin{bmatrix} 3 & 2 \\ 4 & 5 \end{bmatrix}$, $x_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, s = 3

Eigenpairs: [-1, $\begin{bmatrix} -1 & 1 \end{bmatrix}$],[7, $\begin{bmatrix} 5 \\ 3 \end{bmatrix}$]

Answers:

9.1 (a) $p_A(\lambda) = \lambda^2 - 1$; eigenpairs คือ

(1,[1 1]') และ (-1,[1 - 1 1)

(b) $p_A(\lambda) = \lambda^2 - 6\lambda - 7$; eigenpairs คือ

(7,[5 3]') และ (-1,[1 -1]')

9.2 (a) $V = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $V^{-1}AV$ = diag(1, -1)

(b) $V = \begin{bmatrix} 5 & 1 \\ 3 & -1 \end{bmatrix}$, $V^{-1}AV$ = diag(7, -1)

9.3

	x_1'	x_2'	x_3'	x_4'	$\lim_k x_k'$
(a)	C-O.7500 11	C-O.6897 11	C-O.6732 11	C-O.6665 11	C-213 11
(b)	cl -0.33331	Cl 0.38461	Cl 0.78671	Cl 0.93511	Cl 11
(c)	co 11	C l 1/2]	CO.8572 11	Cl 0.96151	Cl 11

9.4	x_1'	x_2'	x_3'	x_4'	$\lim_k x_k$'
(a)	Cl 0.61541	Cl 0.59761	Cl 0.60031	Cl 0.60001	C0 0.61
(b)	[1/2 11	Cl -0.62501	C-0.9316' 1]	[1 -0.99011	+[1 -11
(c)	Cl 0.714311	Cl 0.51	Cl 0.71431	Cl 0.51	None

9.5 จงแสดงว่า (AB)' = B'A' เมื่อ A = $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ และ B = $\begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 0 \end{bmatrix}$

9.6 สำหรับ x = [1 -1 1]' และ y = [2 3 -1]' จงหา cos $\angle(x,y)$

9.7 ใน (a)-(b) จงใช้วิธีของจาโคบิ เพื่อหา rotation matrix U และ diagonal matrix D = diag(λ_1, λ_2) ซึ่ง U'AU = D

(a) A = $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ (b) B = $\begin{bmatrix} -23 & 36 \\ 36 & -2 \end{bmatrix}$

9.6 สำหรับเมตริกซ์ A = $\begin{bmatrix} 1 & -1 & 2 \\ -1 & 1 & 0 \\ 2 & 0 & 2 \end{bmatrix}$ จงทำ 2 iterations ของวิธีของจาโคบิ

Answers:

9.6 cos $\angle(x,y) = x'y/||x||\,||y|| = -2/\sqrt{42}$

9.7 (a) $U = 1/\sqrt{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, $D = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$

(b) $U = 1/5\begin{bmatrix} 4 & 3 \\ -3 & 4 \end{bmatrix}$, $D = \begin{bmatrix} -50 & 0 \\ 0 & 25 \end{bmatrix}$

9.8 เริ่มใช้ $R_1 = R_{13}(-0.662909)$ แล้วใช้ $R_2 = R_{12}(0.395664)$

$$A_1 = \begin{vmatrix} -0.561553 & -0.788205 & 0 \\ -0.788205 & 1 & -0.615412 \\ 0 & -0.615412 & 3.561553 \end{vmatrix}$$

$$A_1 = \begin{bmatrix} -0.890226 & 0 & -0.236853 \\ -0.236853 & -0.568008 & 3.561553 \\ 0 & 1.328674 & -0.568008 \end{bmatrix}$$