

6. EIGENWERTE UND EIGENVEKTOREN

Die Problemstellung in diesem Kapitel lautet, zu einer gegebenen Matrix $A \in \mathbb{C}^{n,n}$ eine komplexe Zahl λ (einen Eigenwert von A) und einen von Null verschiedenen Vektor $x \in \mathbb{C}^n$ (einen Eigenvektor zum Eigenwert λ) zu finden, so dass gilt

$$Ax = \lambda x.$$

Ein Beispiel aus der Mechanik ist die Bestimmung der Eigenfrequenzen einer Membran (Trommel). Die Auslenkung $u : \Omega \rightarrow \mathbb{R}$ genügt der partiellen Differentialgleichung

$$-\Delta u = \lambda u \quad \text{in } \Omega, \quad u \neq 0.$$

Nach Diskretisierung erhält man ein Eigenwertproblem mit reeller, symmetrischer Matrix A .

6.1 Grundlagen

Ist $Ax = \lambda x$ für ein $x \neq 0$, dann ist $x \in \mathbb{C}^n$ ein Eigenvektor und $\lambda \in \mathbb{C}$ ist ein Eigenwert von A . Die Menge

$$\lambda(A) = \{\lambda \mid \lambda \text{ ist Eigenwert von } A\}$$

heißt **Spektrum** von A . Dann gilt

$$\begin{aligned} \lambda \in \lambda(A) &\iff \exists x \neq 0, \text{ so dass } (A - \lambda I)x = 0 \\ &\iff A - \lambda I \text{ ist singulär} \\ &\iff p(\lambda) := \det(A - \lambda I) = 0. \end{aligned}$$

$p(\lambda)$, $p \in \mathcal{P}_n$ wird **charakteristisches Polynom** von A genannt, und $p(\lambda) = 0$ ist die charakteristische Gleichung. Die Vielfachheit der Nullstelle λ heißt **algebraische Vielfachheit** des Eigenwertes λ . Nach dem Fundamentalsatz der Algebra ist

$$\lambda(A) = \{\lambda_1, \dots, \lambda_n\}$$

eine diskrete Menge von n komplexen Zahlen $\lambda_j \in \mathbb{C}$.

Für $X \in \mathbb{C}^{n,n}$ nicht singulär heißt die Abbildung $B = X^{-1}AX$ **Ähnlichkeitstransformation** und wir sagen, dass A ähnlich zu B ist. Aus

$$Bv = \lambda v \iff X^{-1}AXv = \lambda v \iff A(Xv) = \lambda(Xv)$$

folgt, dass A und B dieselben Eigenwerte haben und dass v genau dann Eigenvektor von B ist, wenn Xv Eigenvektor von A ist. Ebenso leicht sieht man, dass ähnliche Matrizen dasselbe charakteristische Polynom haben.

Eine naheliegende Idee zur Berechnung von Eigenwerten wäre, zunächst das charakteristische Polynom zu berechnen und dann die Nullstellen dieses Polynoms numerisch zu bestimmen. Die Berechnung der Nullstellen eines Polynoms aus seinen Koeffizienten ist jedoch ein schlecht konditioniertes Problem. Betrachten wir

$$\begin{aligned} p(\lambda) &= \sum_{j=0}^n a_j \lambda^j, \\ \tilde{p}(\lambda, \varepsilon) &= \sum_{j=0}^n \left(a_j + \varepsilon a_j \frac{\varepsilon_j}{\varepsilon} \right) \lambda^j, & |\varepsilon_j| \leq \varepsilon, \\ &= p(\lambda) + \varepsilon q(\lambda), & q(\lambda) = \sum_{j=0}^n b_j \lambda^j, \end{aligned}$$

so ist $|b_j| \leq |a_j|$. Wir untersuchen jetzt die Nullstellen $\lambda(\varepsilon)$ des gestörten Polynoms $\tilde{p}(\lambda, \varepsilon)$ in Abhängigkeit der Störung ε . Es sei $\lambda(0) = \lambda^*$ Nullstelle von p . Aus $\tilde{p}(\lambda(\varepsilon), \varepsilon) = 0$ folgt durch Differentiation nach ε

$$\begin{aligned} 0 &= \frac{\partial \tilde{p}}{\partial \lambda}(\lambda(\varepsilon), \varepsilon) \lambda'(\varepsilon) + \frac{\partial \tilde{p}}{\partial \varepsilon}(\lambda(\varepsilon), \varepsilon) \\ &= (p'(\lambda(\varepsilon)) + \varepsilon q'(\lambda(\varepsilon))) \lambda'(\varepsilon) + q(\lambda(\varepsilon)) \end{aligned}$$

unter Vernachlässigung von Termen der Ordnung $O(\varepsilon)$

$$\lambda'(0) \approx -\frac{q(\lambda^*)}{p'(\lambda^*)}.$$

Der relative Fehler ist dann durch

$$\frac{|\lambda(\varepsilon) - \lambda^*|}{|\lambda^*|} \approx \frac{|\lambda'(0)\varepsilon|}{|\lambda^*|} \approx \frac{|q(\lambda^*)|}{|\lambda^* p'(\lambda^*)|} |\varepsilon|$$

gegeben. Für $|\lambda^* p'(\lambda^*)|$ klein oder $|q(\lambda^*)|$ groß tritt also eine große Fehlerverstärkung auf.

Beispiel. Für das einfache Beispiel

$$A = \text{diag}(10, 11, \dots, 16) \in \mathbb{R}^{7,7}$$

ist

$$p(\lambda) = -\lambda^7 + 91\lambda^6 - 3535\lambda^5 + \dots - 31\,813\,200\lambda + 57\,657\,600.$$

Für $\lambda^* = 10$ ist damit

$$\begin{aligned} |q(\lambda^*)| &\leq 10^7 + 91 \cdot 10^6 + \dots + 57\,657\,600 \approx 3 \cdot 10^9 \\ |p'(\lambda^*)| &= 720. \end{aligned}$$

Der Verstärkungsfaktor kann also im ungünstigsten Fall $\varepsilon_j = \text{sign}(a_j)\varepsilon$ von der Größenordnung 10^5 sein. \diamond

Fazit: Man sollte *nie* die Koeffizienten des charakteristischen Polynoms zur Berechnung der Eigenwerte einer Matrix verwenden.

Nehmen wir an, dass es für eine Matrix $X \in \mathbb{C}^{n,k}$ mit $1 \leq \text{Rang}(X) = k \leq n$ eine Matrix $B \in \mathbb{C}^{k,k}$ gibt, so dass

$$AX = XB.$$

Mit $\mathcal{R}(X) = \{Xy \mid y \in \mathbb{C}^k\}$ bezeichnen wir das Bild von X . $\mathcal{R}(X)$ ist dann ein **rechts A -invarianter Unterraum**, d. h. $\mathcal{R}(AX) \subseteq \mathcal{R}(X)$, denn für jedes $x \in \mathcal{R}(AX)$ gibt es ein $y \in \mathbb{C}^k$, so dass $x = AXy = XBy \in \mathcal{R}(X)$. Außerdem gibt es zu $\lambda \in \lambda(B)$ ein $y \neq 0$ mit $By = \lambda y$ und daraus folgt

$$AXy = XBy = \lambda Xy.$$

Damit ist $\lambda \in \lambda(A)$ und $Xy \neq 0$ ist ein Eigenvektor von A zum Eigenwert λ . Es gilt also $\lambda(B) \subseteq \lambda(A)$.

Ebenso folgt aus $Y^H A = BY^H$ für eine Matrix $Y \in \mathbb{C}^{n,k}$ mit $1 \leq \text{Rang}(Y) = k \leq n$, dass $\mathcal{R}(Y)$ ein **links A -invarianter Unterraum** ist.

Definition 6.1. Eine Matrix $A \in \mathbb{C}^{n,n}$ heißt **reduzibel**, wenn es eine Permutationsmatrix P gibt, so dass

$$P^T A P = B = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},$$

mit quadratischen Matrizen $A_{11} \in \mathbb{C}^{k,k}$ und $A_{22} \in \mathbb{C}^{n-k,n-k}$ gilt. Anderenfalls heißt A **irreduzibel**.

Wegen

$$\begin{aligned} B \begin{bmatrix} I_k \\ 0 \end{bmatrix} &= \begin{bmatrix} I_k \\ 0 \end{bmatrix} A_{11} \\ \begin{bmatrix} 0 & I_{n-k} \end{bmatrix} B &= A_{22} \begin{bmatrix} 0 & I_{n-k} \end{bmatrix}, \end{aligned}$$

ist $\mathcal{R}\left(\begin{bmatrix} I_k \\ 0 \end{bmatrix}\right)$ ein rechts und $\mathcal{R}\left(\begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix}\right)$ ein links B -invarianter Unterraum und es gilt $\lambda(A) = \lambda(A_{11}) \cup \lambda(A_{22})$. ($\mathcal{R}(P \begin{bmatrix} I_k \\ 0 \end{bmatrix})$ ist rechts und $\mathcal{R}(P \begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix})$ links A -invarianter Unterraum.)

Lemma 6.2. Es seien $A \in \mathbb{K}^{n,n}$, $B \in \mathbb{K}^{k,k}$ und $X \in \mathbb{K}^{n,k}$ mit $\mathbb{K} = \mathbb{C}$ oder $\mathbb{K} = \mathbb{R}$ und es gelte

$$AX = XB, \quad \text{Rang } X = k. \quad (6.1)$$

Dann gibt es eine unitäre Matrix $Q \in \mathbb{K}^{n,n}$ mit

$$Q^H A Q = T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}, \quad (6.2)$$

wobei $\lambda(T_{11}) = \lambda(A) \cap \lambda(B)$.

Beweis. Es sei

$$X = Q \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad Q \in \mathbb{K}^{n,n}, \quad R \in \mathbb{K}^{k,k}$$

eine QR-Zerlegung von X . Setzen wir dies in (6.2) ein, so ergibt sich

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix} B,$$

mit

$$Q^H A Q = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}.$$

R ist nach Voraussetzung nicht singulär. Aus $T_{21}R = 0$ und $T_{11}R = RB$ können wir $T_{21} = 0$ und $\lambda(T_{11}) = \lambda(B)$ schließen. Die übrige Behauptung haben wir bereits oben gezeigt. \square

Diese obigen Resultate lassen sich in offensichtlicher Weise auf beliebige Blockdreiecksma-
trizen

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ 0 & A_{22} & \cdots & A_{2k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_{kk} \end{bmatrix}$$

verallgemeinern. Hier gilt

$$\lambda(A) = \bigcup_{j=1}^k \lambda(A_{jj}).$$

Viele Algorithmen basieren darauf, eine Folge von Ähnlichkeitstransformationen durch-
zuführen

$$A_0 = A, \quad A_k = S_k^{-1} A_{k-1} S_k, \quad k = 1, 2, \dots, \quad S_k \text{ nicht singulär.}$$

A_k ist ähnlich zu A und falls y Eigenvektor von A_k ist, dann erhalten wir durch $x = S_1 \cdots S_k y$ einen Eigenvektor von A . Das Ziel ist jetzt, Ähnlichkeitstransformationen derart zu finden, dass $\lambda(A_k)$ einfach zu berechnen ist. Ein Beispiel wäre eine Dreiecksform für A_k .

6.2 Normalformen

Ähnlichkeitstransformationen erlauben es, eine Matrix in verschiedene Normalformen zu transformieren. Dabei bleiben die Eigenwerte unverändert und Eigenvektoren können mit Hilfe der Transformationsmatrix berechnet werden. Wir stellen die wichtigsten Transformationen zusammen.

Satz 6.3. (Schur-Normalform)

Zu jedem $A \in \mathbb{C}^{n,n}$ gibt es eine unitäre Matrix $U \in \mathbb{C}^{n,n}$ so, dass

$$U^H A U = R = D + N,$$

wobei R obere Dreiecksform, N strikte obere Dreiecksform (obere Dreiecksform mit Nullen auf der Diagonalen) hat und $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ mit den Eigenwerten $\lambda_1, \dots, \lambda_n$ von A ist. Die Matrix U kann so gewählt werden, dass die Eigenwerte in beliebiger Reihenfolge in D auftreten.

Beweis. Der Beweis erfolgt durch Induktion nach n . Für $n = 1$ ist nichts zu zeigen.

Nehmen wir an, dass die Aussage für alle Matrizen der Ordnung $n - 1$ gilt und wählen wir einen beliebigen Eigenwert λ von A . Dann gilt $Ax = \lambda x$ für einen Vektor $x \neq 0$ und wir können $u_1 = x/\|x\|$ setzen. Zu u_1 konstruieren wir $U_2 \in \mathbb{C}^{n,n-1}$, so dass $V = [u_1 \ U_2]$ eine unitäre $n \times n$ Matrix ist. Wegen

$$AV = A [u_1 \ U_2] = [\lambda u_1 \ AU_2]$$

haben wir

$$V^H AV = \begin{bmatrix} u_1^H \\ U_2^H \end{bmatrix} AV = \begin{bmatrix} \lambda u_1^H u_1 & u_1^H AU_2 \\ \lambda U_2^H u_1 & U_2^H AU_2 \end{bmatrix} = \begin{bmatrix} \lambda & w^H \\ 0 & B \end{bmatrix}.$$

Angewandt auf $B \in \mathbb{C}^{n-1,n-1}$ liefert die Induktionsannahme die Existenz einer unitären Matrix $\tilde{U} \in \mathbb{C}^{n-1,n-1}$, so dass $\tilde{U}^H B \tilde{U} = \tilde{R}$ obere Dreiecksform hat. Daher ist

$$U^H A U = R = \begin{bmatrix} \lambda & w^H \tilde{U} \\ 0 & \tilde{R} \end{bmatrix}, \quad U = V \begin{bmatrix} 1 & 0 \\ 0 & \tilde{U} \end{bmatrix}.$$

Offensichtlich ist U unitär.

Die obige Herleitung macht klar, dass wir U so wählen können, dass die Eigenwerte von A in beliebiger Reihenfolge auf der Diagonalen von R auftreten. \square

Ist A reell, dann ist es häufig von Vorteil, sich auf reelle Ähnlichkeitstransformationen zu beschränken.

Satz 6.4. (*Reelle Schur-Normalform*)

Zu jedem $A \in \mathbb{R}^{n,n}$ gibt es eine reelle Orthogonalmatrix $Q \in \mathbb{R}^{n,n}$, so dass

$$Q^T A Q = R = D + N,$$

wobei R eine reelle obere Blockdreiecksmatrix ist, deren Blockdiagonalmatrix D aus 1×1 und 2×2 Blöcken besteht, bei denen alle 2×2 Blöcke konjugiert komplexe Eigenwerte haben. N ist eine strikte obere Dreiecksmatrix.

Beweis. Man geht analog zum Beweis von Satz 6.3 vor, wobei man Eigenvektoren zu konjugiert komplexen Eigenwerten in Real- und Imaginärteil zerlegt. Die Details überlassen wir als Übung. \square

Definition 6.5. Eine Matrix $A \in \mathbb{C}^{n,n}$ ist **normal** genau dann, wenn $AA^H = A^H A$.

Beispiel. Hermitesche Matrizen ($A = A^H$), schief-Hermitesche Matrizen ($A = -A^H$) und unitäre Matrizen sind normal. \diamond

Satz 6.6. Eine Matrix $A \in \mathbb{C}^{n,n}$ ist normal genau dann, wenn sie unitär diagonalisierbar ist, d. h. es gibt eine unitäre Matrix $U \in \mathbb{C}^{n,n}$, so dass

$$U^H A U = D = \text{diag}(\lambda_1, \dots, \lambda_n).$$

Beweis. Angenommen, A ist normal. $U^H A U = R = D + N$ sei die Schur-Normalform von A . Wegen

$$(U^H A U)^H (U^H A U) = U^H (A^H A) U = U^H (A A^H) U = (U^H A U) (U^H A U)^H$$

folgt, dass A genau dann normal ist, wenn R normal ist.

Es ist leicht zu sehen, dass eine obere Dreiecksmatrix genau dann normal ist, wenn sie diagonal ist, also $R = D$ (Übung).

Ist umgekehrt A unitär diagonalisierbar, dann erhalten wir sofort

$$A^H A = U D^H D U^H = U D D^H U^H = A A^H,$$

also ist A normal. \square

Obwohl die unitäre Matrix U in der Schur-Normalform nicht eindeutig ist, ist

$$\|N\|_F^2 = \|A\|_F^2 - \|D\|_F^2 = \|A\|_F^2 - \sum_{j=1}^n |\lambda_j|^2$$

unabhängig von der Wahl von U . $\|N\|_F$ wird **Abstand zur Normalität** von A genannt. (Ist A normal, dann ist $N = 0$.)

Ist A nicht normal, dann kann A nicht unitär auf Diagonalform transformiert werden. Eine beliebige Matrix A kann im Allgemeinen überhaupt nicht mit Ähnlichkeitstransformationen auf Diagonalform transformiert werden. Eine – insbesondere für theoretische Zwecke – nützliche Normalform ist die Jordan-Normalform.

Satz 6.7. (Jordan Normalform)

Zu jedem $A \in \mathbb{C}^{n,n}$ gibt es eine nicht singuläre Matrix $X \in \mathbb{C}^{n,n}$, so dass

$$X^{-1}AX = J = \text{diag}(J_{m_1}(\lambda_1), \dots, J_{m_r}(\lambda_r)),$$

wobei

$$J_{m_i}(\lambda_i) = \begin{bmatrix} \lambda_i & 1 & & 0 \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{bmatrix} = \lambda_i I + N \in \mathbb{C}^{m_i, m_i}, \quad m_i \geq 1.$$

Die Zahlen m_i sind eindeutig und es gilt $\sum_{i=1}^r m_i = n$. Zu jedem Jordan-Kästchen $J_{m_i}(\lambda_i)$ gehört ein eindimensionaler Eigenraum. Die Zahl der Jordan-Kästchen, die zu einem mehrfachen Eigenwert λ gehören, ist gleich der geometrischen Vielfachheit von λ . \square

Stimmen geometrische und algebraische Vielfachheit (Vielfachheit der Nullstelle λ im charakteristischen Polynom) von λ nicht überein (es gibt dann ein Jordan-Kästchen der Dimension größer eins), dann nennt man λ einen Eigenwert mit **Defekt** und sagt, dass A Defekt hat. A hat keinen Defekt genau dann, wenn A diagonalisierbar ist.

Da die Jordan-Normalform im Allgemeinen nicht stetig von den Einträgen in der Matrix A abhängt, ist es schwierig, sie numerisch zu bestimmen, wenn A nicht diagonalisierbar ist.

Beispiel 6.1. Es sei

$$J_m(\lambda, \epsilon) = \begin{bmatrix} \lambda & 1 & & 0 \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ \epsilon & & & \lambda \end{bmatrix} \in \mathbb{C}^{m,m}$$

ein gestörtes Jordan-Kästchen. Die Matrix $J_m(\lambda, 0)$ hat einen Eigenwert λ der Vielfachheit m und ist bereits in Jordan-Normalform.

Für jedes $\epsilon > 0$ hat die Matrix $J_m(\lambda, \epsilon)$ jedoch m paarweise verschiedene Eigenwerte μ_j , die gerade die Nullstellen von

$$(\lambda - \mu)^m - (-1)^m \epsilon = 0,$$

wie man durch Entwicklung von $\det(J_m(\lambda, \epsilon) - \mu I)$ nach der ersten Spalte leicht sieht. Daher ist $J_m(\lambda, \epsilon)$ diagonalisierbar für jedes $\epsilon \neq 0$ und seine Eigenwerte μ_j genügen $|\mu_j - \lambda| = |\epsilon|^{1/m}$.

Für $m = 6$ und die kleine Störung von $\epsilon = 10^{-6}$ ergibt sich eine erhebliche Abweichung der Eigenwerte von 0.1. \diamond

Satz 6.8. Es sei eine Matrix $A \in \mathbb{C}^{n,n}$ mit Spektralradius $\rho = \rho(A) = \max_{\lambda \in \lambda(A)} |\lambda|$ gegeben. Bezeichnen wir mit $\|\cdot\|$ eine beliebige p -Norm, $1 \leq p \leq \infty$, so ist für $T \in \mathbb{C}^{n,n}$ nicht singulär

$$\|A\|_T = \|T^{-1}AT\|$$

die zugehörige T -Norm definiert und es gilt

$$(a) \quad \rho(A) \leq \|A\|.$$

(b) Ist A diagonalisierbar, dann gibt es eine nicht singuläre Matrix T , so dass

$$\|A\|_T = \rho.$$

(c) Zu jedem $\epsilon > 0$ gibt es eine nicht singuläre Matrix $T(\epsilon)$, so dass

$$\|A\|_{T(\epsilon)} \leq \rho + \epsilon.$$

Beweis. Man überprüft leicht, dass $\|A\|_T$ eine Matrixnorm ist (Definition 3.6).

(a) Sei $\lambda \in \lambda(A)$ und sei x ein zugehöriger Eigenvektor normiert auf $\|x\| = 1$. Dann gilt

$$\|A\| = \max_{\|y\|=1} \|Ay\| \geq \|Ax\| = |\lambda| \|x\| = |\lambda|.$$

(b) Ist A diagonalisierbar, dann gibt es eine nicht singuläre Matrix T , so dass

$$T^{-1}AT = D = \text{diag}(\lambda_1, \dots, \lambda_n), \quad \lambda_j \in \lambda(A).$$

Offensichtlich gilt $\|A\|_T = \|D\| = \rho$.

(c) Es sei

$$X^{-1}AX = J = \text{diag}(J_{m_1}(\lambda_1), \dots, J_{m_r}(\lambda_r)),$$

mit

$$J_i := J_{m_i}(\lambda_i) = \lambda_i I + N_i \in \mathbb{C}^{m_i, m_i}$$

die Jordan-Normalform von A . Definieren wir $D = \text{diag}(D_1, \dots, D_r)$

$$D_i = \text{diag}(1, \epsilon, \dots, \epsilon^{m_i-1}),$$

so gilt $D^{-1}JD = \text{diag}(D_1^{-1}J_1D_1, \dots, D_r^{-1}J_rD_r)$, mit

$$D_i^{-1}J_iD_i = \lambda_i I + \epsilon N_i.$$

Folglich ist

$$\|D^{-1}JD\| = \max_{i=1, \dots, r} \|D_i^{-1}J_iD_i\| \leq \max_{i=1, \dots, r} (|\lambda_i| + \epsilon \|N_i\|) \leq \rho + \epsilon \max_{i=1, \dots, r} \|N_i\|.$$

Für $y = [y_1 \ \dots \ y_{m_i}]^T$ ist $N_i y = [y_2 \ \dots \ y_{m_i} \ 0]^T$. Aus $\|N_i y\| \leq \|y\|$ folgt dann $\|N_i\| \leq 1$, woraus $\|D^{-1}JD\| \leq \rho + \epsilon$ folgt. Setzen wir $T = XD$, so erhalten wir schließlich

$$\|A\|_T = \|T^{-1}AT\| = \|D^{-1}X^{-1}AXD\| = \|D^{-1}JD\| \leq \rho + \epsilon$$

und damit die Behauptung. \square

6.3 Störungstheorie, Einschließungssätze

Numerische Verfahren werden von Rundungsfehlern beeinflusst. Das Beste, was wir von einem Eigenwertalgorithmus verlangen können, ist, dass er approximative Eigenwerte von A liefert, die exakte Eigenwerte einer leicht gestörten Matrix $A + E$ sind, also stabil im Sinne der Rückwärtsanalyse ist. Um solche Eigenschaften zu zeigen, müssen wir zunächst wissen, wie sich eine Störung E auf Eigenwerte und Eigenvektoren auswirken kann.

Satz 6.9. (Bauer-Fike)

Sei $A \in \mathbb{C}^{n,n}$ diagonalisierbar, $X^{-1}AX = D = \text{diag}(\lambda_1, \dots, \lambda_n)$ und sei $\mu \in \lambda(A + E)$, $E \in \mathbb{C}^{n,n}$. Dann gilt in jeder p -Norm

$$\min_{1 \leq j \leq n} |\mu - \lambda_j| \leq \kappa(X) \|E\|.$$

Beweis. Für $\mu \in \lambda(A)$ ist die Aussage trivial. Nehmen wir also $\mu \notin \lambda(A)$ an. Da $\mu \in \lambda(A+E)$ ist $A+E-\mu I$ singulär und dasselbe gilt für

$$X^{-1}(A+E-\mu I)X = D-\mu I+X^{-1}EX.$$

Es gibt daher ein $y \neq 0$, so dass

$$(D-\mu I)y = -X^{-1}EXy$$

gilt. Wegen $\mu \notin \lambda(A)$ ist dies äquivalent zu

$$y = -(D-\mu I)^{-1}X^{-1}EXy.$$

Nehmen wir auf beiden Seiten die Norm, so folgt

$$\|y\| \leq \|(D-\mu I)^{-1}\| \kappa(X) \|E\| \cdot \|y\|.$$

Die Behauptung folgt dann aus $\|(D-\mu I)^{-1}\| = 1/\min_{1 \leq j \leq n} |\mu - \lambda_j|$. \square

Bemerkung. Nach Satz 6.6 ist A normal genau dann, wenn X unitär gewählt werden kann. In diesem Fall ist $\kappa(X) = 1$ und die Eigenwerte sind perfekt konditioniert, sogar wenn sie nicht einfach sind.

Beispiel. Die Matrix

$$A = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix}, \quad 0 < \epsilon \ll 1$$

wird durch

$$X = \begin{bmatrix} 1 & 1 \\ \sqrt{\epsilon} & -\sqrt{\epsilon} \end{bmatrix}, \quad X^{-1} = \frac{1}{2\sqrt{\epsilon}} \begin{bmatrix} \sqrt{\epsilon} & 1 \\ \sqrt{\epsilon} & -1 \end{bmatrix}.$$

diagonalisiert. Hier ist $\kappa_\infty(X) = \|X^{-1}\|_\infty \|X\|_\infty = \frac{\sqrt{\epsilon}+1}{2\sqrt{\epsilon}} 2 = \frac{1}{\sqrt{\epsilon}} + 1 \gg 1$. Im Grenzfall $\epsilon \rightarrow 0$ ist A nicht diagonalisierbar. \diamond

Im Allgemeinen wird A sowohl gut als auch schlecht konditionierte Eigenwerte haben. Daher ist es nützlich, Störungstheorie für einzelne Eigenwerte zu betreiben. Wir beginnen mit Einschließungssätzen.

Satz 6.10. (*Gershgorin*)

Es gilt

$$\lambda(A) \subseteq \bigcup_{j=1}^n \mathcal{D}_j, \quad \mathcal{D}_j = \{z \in \mathbb{C} : |z - a_{jj}| \leq r_j\}, \quad r_j = \sum_{l=1, l \neq j}^n |a_{jl}|.$$

Beweis. Sei $\lambda \in \lambda(A)$ und $x \neq 0$ ein zugehöriger Eigenvektor. Wir betrachten $Ax = \lambda x$ komponentenweise:

$$(\lambda - a_{jj})x_j = \sum_{l=1, l \neq j}^n a_{jl}x_l, \quad j = 1, \dots, n.$$

In dieser Gleichung wählen wir j so, dass $|x_j| = \|x\|_\infty > 0$. Dann gilt

$$|\lambda - a_{jj}| \leq \sum_{l=1, l \neq j}^n |a_{jl}| \frac{|x_l|}{|x_j|} \leq r_j,$$

also $\lambda \in \mathcal{D}_j$. \square

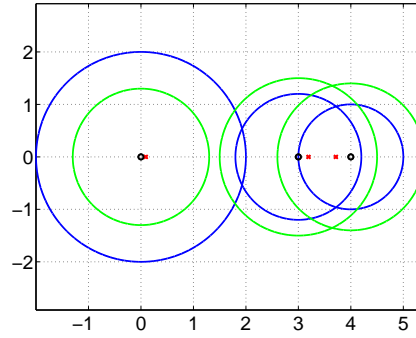


Abb. 6.1: Gershgorin-Kreise der Matrix A aus Beispiel 6.1 (blau) und von A^T (grün). Die Mittelpunkte der Gershgorin-Kreise sind als schwarze Kreise, die Eigenwerte von A durch rote Kreuze dargestellt.

Die Kreisscheiben \mathcal{D}_j werden auch Gershgorin-Kreise genannt.

Beispiel 6.1. Wir betrachten die Matrix

$$A = \begin{bmatrix} 0 & 1 & -1 \\ 0.5 & 4 & -0.5 \\ 0.8 & 0.4 & 3 \end{bmatrix}.$$

Die Gershgorin-Kreise von A sind in Abbildung 6.1 eingezeichnet. Wegen $\lambda(A) = \lambda(A^T)$ erhält man eine bessere Einschließung, wenn man zusätzlich die Gershgorin-Kreise von A^T berechnet und dann beide Mengen schneidet. \diamond

Eine Verschärfung dieses Satzes ist:

Satz 6.11. *Ist die Vereinigung \mathcal{M} von k Gershgorinkreisen disjunkt zu allen übrigen Kreisen, dann enthält \mathcal{M} genau k Eigenwerte.*

Beweis. Für $t \in [0, 1]$ betrachten wir

$$A(t) = tA + (1-t)D_A, \quad D_A = \text{diag}(A) = \text{diag}(a_{11}, \dots, a_{nn}).$$

Da die Koeffizienten des charakteristischen Polynoms von $A(t)$ stetige Funktionen von t sind, gilt dies auch für die Eigenwerte $\lambda(t)$ von $A(t)$. Nach Definition ist $A(0) = D_A$ und $A(1) = A$, woraus $\lambda_j(0) = a_{jj}$, $\lambda_j(1) = \lambda_j$ folgt. Für $t = 0$ sind nach Voraussetzung genau k Eigenwerte in \mathcal{M} ($\lambda_j(0)$ sind ja gerade die Mittelpunkte der Kreise). Die j -ten Gershgorin-Kreise $\mathcal{D}_j(t)$ von $A(t)$ haben Mittelpunkt a_{jj} und Radius tr_j , denn $a_{jj}(t) = a_{jj}$ und

$$r_j(t) = \sum_{l=1, l \neq j}^n |a_{jl}(t)| = \sum_{l=1, l \neq j}^n |ta_{jl}| = tr_j.$$

Daraus folgt die Inklusion $\mathcal{D}_j(t_0) \subset \mathcal{D}_j(t_1)$ für $t_1 > t_0$ und insbesondere

$$\lambda(A(t)) \subset \bigcup_{j=1}^n \mathcal{D}_j \quad \text{für alle } t \in [0, 1].$$

Daher kann aus Stetigkeitsgründen ein Eigenwert $\lambda_j(t)$ nicht in eine Teilmenge der Vereinigung aller Kreisscheiben springen, die keine stetige Verbindung zu a_{jj} für $t = 1$ hat. Damit liegen genau k Eigenwerte von $A = A(1)$ in \mathcal{M} . \square

Mit dieser Beweistechnik können wir jetzt folgenden Satz beweisen:

Satz 6.12. Sei $\lambda \in \lambda(A)$ ein einfacher Eigenwert von A und seien x und y zugehörige rechte und linke Eigenvektoren:

$$Ax = \lambda x, \quad y^H A = \lambda y^H.$$

Dann hat die Matrix $A + \epsilon E$ für ϵ hinreichend klein einen einfachen Eigenwert $\lambda(\epsilon)$, so dass

$$\lambda(\epsilon) = \lambda + \epsilon \frac{y^H E x}{y^H x} + O(\epsilon^2).$$

Beweis. Wir betrachten eine Kreisscheibe $\mathcal{D} = \{\mu \in \mathbb{C} : |\mu - \lambda| < \delta\}$ um λ , die so gewählt ist, dass $\mathcal{D} \cap \lambda(A) = \{\lambda\}$. Wie im Beweis von Satz 6.11 kann man zeigen, dass für ϵ hinreichend klein die Matrix $A + \epsilon E$ einen einfachen Eigenwert $\lambda(\epsilon)$ in \mathcal{D} hat. Ist $x(\epsilon)$ ein zugehöriger Eigenvektor, so gilt

$$(A + \epsilon E)x(\epsilon) = \lambda(\epsilon)x(\epsilon), \quad x(0) = x, \quad \lambda(0) = \lambda.$$

Aus dem Satz über implizite Funktionen kann man folgern, dass $x(\epsilon)$, $\lambda(\epsilon)$ analytische Funktionen von ϵ sind, solange $\epsilon < \epsilon_0$. Daher gilt

$$(A + \epsilon E)x'(\epsilon) + Ex(\epsilon) = \lambda(\epsilon)x'(\epsilon) + \lambda'(\epsilon)x(\epsilon).$$

Werten wir diese Gleichung an $\epsilon = 0$ aus, so erhalten wir

$$(A - \lambda I)x'(0) + Ex = \lambda'(0)x.$$

Wegen $y^H(A - \lambda I) = 0$ folgt daraus $\lambda'(0) = \frac{y^H E x}{y^H x}$. □

Bemerkung. Für $\|E\| = 1$, $\|x\| = \|y\| = 1$ ist $|\lambda'(0)| \leq \frac{1}{|y^H x|}$. Man nennt $\frac{1}{|y^H x|}$ die **Konditionszahl** des Eigenwerts λ , denn nach obigem Satz werden einfache Eigenwerte in erster Näherung durch $\frac{\epsilon}{|y^H x|}$ gestört.

Ist λ ein Eigenwert mit Defekt, dann muss man Störungen der Größenordnung $\epsilon^{1/m}$ erwarten, wobei m die Dimension des größten Jordan-Kästchens ist (vgl. Beispiel 6.1). Man beachte, dass für ein Jordan-Kästchen $x = e_1$, $y = e_m$, also $y^H x = 0$ gilt.

Beispiel. Es sei

$$A + \epsilon E = \begin{bmatrix} 1 & \epsilon & 2\epsilon \\ \epsilon & 2 & \epsilon \\ \epsilon & 2\epsilon & 2 \end{bmatrix}, \quad A = \text{diag}(1, 2, 2).$$

Eigenvektoren von A sind $x_j = y_j = e_j$, $j = 1, 2, 3$. Wegen

$$y_j^H E x_j = (E)_{jj} = 0$$

verschwinden die Terme erster Ordnung in Satz 6.12 und wir erhalten eine $O(\epsilon^2)$ -Schranke für die Störung des einfachen Eigenwerts 1. Der Satz von Bauer-Fike liefert hingegen eine $O(\epsilon)$ -Schranke für alle Eigenwerte. \diamond

Definition 6.13. Zu gegebener Matrix A und $x \neq 0$ heißt

$$\varrho_A(x) = \frac{x^H A x}{x^H x}$$

Rayleigh-Quotient von x .

Falls die Matrix aus dem Zusammenhang klar ist, lassen wir den Index A weg.

Definition 6.14. Die Menge

$$\mathcal{F}(A) = \{\varrho_A(x), x \in \mathbb{C}^n, x \neq 0\}$$

aller Rayleigh-Quotienten von A heißt **Wertebereich** von A .

Wichtig ist, dass auch für reelle Matrizen A der Wertebereich von allen Rayleigh-Quotienten von Vektoren in \mathbb{C}^n gebildet wird.

Lemma 6.15. Es gilt

- (a) $\varrho(\gamma x) = \varrho(x)$ für alle $\gamma \neq 0, \gamma \in \mathbb{C}$.
- (b) $\lambda(A) \subset \mathcal{F}(A)$, d. h. alle Eigenwerte liegen im Wertebereich.
- (c) Für normale Matrizen (d. h. $A^H A = A A^H$) gilt $\mathcal{F}(A) = \text{conv}(\lambda(A))$.

Beweis. (a) folgt sofort aus der Definition.

(b) Wegen $Ax = \lambda x, x \neq 0$ folgt $\varrho(x) = \lambda$, also ist $\lambda(A) \subset \mathcal{F}(A)$.

(c) Ist A normal, dann gibt es eine unitäre Matrix U , so dass $U^H A U = D = \text{diag}(\lambda_j)$. Zu $\|x\| = 1$ definieren wir $y = U^H x$. Dann ist

$$\varrho(x) = x^H A x = x^H U D U^H x = y^H D y = \sum_{j=1}^n \lambda_j |y_j|^2.$$

Wegen $\|y\| = 1$ und $|y_j| \geq 0$ ist dies eine Konvexkombination der Eigenwerte von A . \square

Ist A nicht normal, dann kann der Wertebereich deutlich größer sein, als die Konvexkombination der Eigenwerte. Hausdorff (1919) konnte jedoch zeigen, dass der Wertebereich immer eine kompakte und konvexe Menge ist.

Für Hermitesche Matrizen sind die folgenden Schranken und Charakterisierungen für Rayleigh-Quotienten hilfreich:

Satz 6.16. (Rayleigh-Ritz)

Ist $A \in \mathbb{C}^{n,n}$ Hermitesch, dann gilt

- (a) $\lambda_{\min} \leq \varrho(x) \leq \lambda_{\max} \quad \forall x \neq 0$
- (b) $\lambda_{\max} = \max_{x \neq 0} \varrho(x)$
- (c) $\lambda_{\min} = \min_{x \neq 0} \varrho(x)$

Beweis. (a) folgt sofort aus Lemma 6.15(c), (b) und (c) folgen aus Lemma 6.15(b) durch Einsetzen der zum größten und kleinsten Eigenwert gehörenden Eigenvektoren. \square

Ein weiterer Einschließungssatz basiert auf dem Wertebereich Hermitescher bzw. schief-Hermitescher Matrizen. Wir haben gerade gezeigt, dass der Wertebereich einer Hermiteschen Matrix das Intervall $\mathcal{F}(A) = [\lambda_{\min}, \lambda_{\max}]$ ist. Hieraus können wir für den Wertebereich einer beliebigen Matrix die folgende Einschließung angeben:

Satz 6.17. (*Bendixson-Hirsch*)

Für $A \in \mathbb{C}^{n,n}$ gelte für $\mu_n \leq \mu_1$ und $\tau_n \leq \tau_1$

$$\lambda\left(\frac{1}{2}(A + A^H)\right) \subseteq [\mu_n, \mu_1] \quad \text{und} \quad \lambda\left(\frac{1}{2i}(A - A^H)\right) \subseteq [\tau_n, \tau_1].$$

Dann ist $\mathcal{F}(A) \subseteq [\mu_n, \mu_1] \times i[\tau_n, \tau_1]$.

Beweis. Es sei $\|x\| = 1$. Dann gilt

$$\begin{aligned} \varrho_A(x) &= x^H A x = x^H \left(\frac{1}{2}(A + A^H) \right) x + i x^H \left(\frac{1}{2i}(A - A^H) \right) x \\ &= \varrho_{\frac{1}{2}(A+A^H)}(x) + i \varrho_{\frac{1}{2i}(A-A^H)}(x). \end{aligned}$$

Da $(A + A^H)/2$ und $(A - A^H)/(2i)$ Hermitesch sind, haben wir damit ϱ_A in Real- und Imaginärteil zerlegt und die Behauptung folgt direkt aus Lemma 6.15(c). \square

6.4 Potenzenmethode

Zur Berechnung von einigen wenigen Eigenwerten und Eigenvektoren einer Matrix A eignen sich Varianten des einfachsten und ältesten Verfahrens, der Potenzenmethode:

$$y_0 \in \mathbb{C}^n \text{ beliebig,} \quad y_{k+1} = A y_k, \quad k = 0, 1, 2, \dots$$

Offensichtlich gilt $y_k = A^k y_0$.

Wir sortieren im Folgenden die Eigenwerte von A nach ihrem Betrag:

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|.$$

Satz 6.18. Es sei $A \in \mathbb{C}^{n,n}$ diagonalisierbar mit $X^{-1}AX = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, $X = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}$, $\|x_i\| = 1$ und es gelte

$$\eta := \frac{|\lambda_2|}{|\lambda_1|} < 1.$$

Ist für $a := X^{-1}y_0$, $a = [\alpha_1 \ \dots \ \alpha_n]^T$ die erste Komponente $\alpha_1 \neq 0$, dann gilt für $y_{k+1} = A y_k$

- (a) $y_k = \lambda_1^k [\alpha_1 x_1 + O(\eta^k)]$ (y_k / λ_1^k konvergiert gegen einen Eigenvektor von A).
- (b) Für die Rayleigh-Quotienten gilt $\varrho_A(y_k) = \lambda_1 + O(\eta^k)$.
- (c) Falls A normal ist, gilt $\varrho_A(y_k) = \lambda_1 + O(\eta^{2k})$.

Beweis. Nach Voraussetzung ist $y_0 = Xa$, also

$$y_1 = A y_0 = A X a = X \Lambda a, \quad y_k = A^k y_0 = A^k X a = X \Lambda^k a.$$

(a) Folgt aus

$$y_k = \lambda_1^k \left[\alpha_1 x_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k x_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k x_n \right].$$

(b) Wir berechnen Zähler und Nenner des Rayleigh-Quotienten getrennt:

$$\begin{aligned}
 y_k^H y_k &= a^H \bar{\Lambda}^k X^H X \Lambda^k a \\
 &= \sum_{i,j=1}^n \bar{\alpha}_i \alpha_j \bar{\lambda}_i^k \lambda_j^k x_i^H x_j \\
 &= |\lambda_1|^{2k} \left[\sum_{i=1}^n |\alpha_i|^2 \left| \frac{\lambda_i}{\lambda_1} \right|^{2k} + \sum_{\substack{i,j=1 \\ i \neq j}}^n \bar{\alpha}_i \alpha_j \left(\frac{\bar{\lambda}_i}{\lambda_1} \right)^k \left(\frac{\lambda_j}{\lambda_1} \right)^k x_i^H x_j \right], \\
 y_k^H A y_k &= a^H \bar{\Lambda}^k X^H X \Lambda^{k+1} a \\
 &= |\lambda_1|^{2k} \lambda_1 \left[\sum_{i=1}^n |\alpha_i|^2 \left| \frac{\lambda_i}{\lambda_1} \right|^{2k} \frac{\lambda_i}{\lambda_1} + \sum_{\substack{i,j=1 \\ i \neq j}}^n \bar{\alpha}_i \alpha_j \left(\frac{\bar{\lambda}_i}{\lambda_1} \right)^k \left(\frac{\lambda_j}{\lambda_1} \right)^{k+1} x_i^H x_j \right].
 \end{aligned}$$

Für den Rayleigh-Quotient ergibt sich daraus

$$\varrho_A(y_k) = \frac{y_k^H A y_k}{y_k^H y_k} = \frac{\lambda_1(1 + O(\eta^k))}{1 + O(\eta^k)} = \lambda_1(1 + O(\eta^k)).$$

(c) Ist A normal, dann gilt $X^H X = I$. In obigen Formeln fallen die Summen für $i \neq j$ also weg und es gilt

$$\varrho_A(y_k) = \frac{y_k^H A y_k}{y_k^H y_k} = \frac{\lambda_1(1 + O(\eta^{2k+1}))}{1 + O(\eta^{2k})} = \lambda_1(1 + O(\eta^{2k})). \quad \square$$

Beispiel. Die Matrix $A = \text{tridiag}(1, 2, 1) \in \mathbb{R}^{3,3}$ hat maximalen Eigenwert

$$\lambda_1 = 2\left(1 + \cos \frac{\pi}{4}\right) = 3.414213562 \dots$$

Der Konvergenzfaktor für die Potenzenmethode ist $\eta \approx 0.586$. Die Potenzenmethode (ohne Normierung) generiert folgende Vektoren:

$$y_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad y_1 = \begin{bmatrix} 3 \\ 4 \\ 3 \end{bmatrix}, \quad y_2 = \begin{bmatrix} 10 \\ 14 \\ 10 \end{bmatrix}.$$

Als Näherung für λ_1 erhalten wir

k	$\varrho(y_k)$	$ \varrho(y_k) - \lambda_1 $
1	3.333333333333	$8.08 \cdot 10^{-2}$
2	3.411764705882	$2.44 \cdot 10^{-3}$
3	3.414141414141	$7.21 \cdot 10^{-5}$
4	3.414211438475	$2.12 \cdot 10^{-6}$
5	3.414213499851	$6.25 \cdot 10^{-8}$

Die schnelle Konvergenz resultiert aus der Normalität von A sowie dem kleinen Konvergenzfaktor η . \diamond

Algorithmus 6.1 Potenzenmethode

```

 $y_0 \neq 0$  gegebener Startvektor,  $y_0 = y_0 / \|y_0\|$ 
for  $k = 0, 1, \dots$  do
     $z_{k+1} = Ay_k$ 
     $\varrho_k = y_k^H z_{k+1}$ 
     $y_{k+1} = \frac{1}{\|z_{k+1}\|} z_{k+1}$ 
end for

```

Um Overflow oder Underflow zu verhindern, normiert man in der Praxis die Folge y_k , vgl. Algorithmus 6.1.

Die Konvergenz der Potenzenmethode ist langsam, falls η nahe bei eins ist. Ein weiterer Nachteil der Potenzenmethode ist, dass man nur den betragsgrößten Eigenwert und den zugehörigen Eigenvektor berechnen kann. Weitere Eigenwerte und schnellere Konvergenz erreicht man mit der inversen Potenzenmethode mit Shift. Dazu sei μ eine Approximation an einen gesuchten Eigenwert $\lambda_j \in \lambda(A)$ so, dass gilt

$$|\mu - \lambda_j| \ll |\mu - \lambda_k|, \quad k \neq j.$$

Da $1/(\mu - \lambda_k)$ die Eigenwerte von $(\mu I - A)^{-1}$ sind, ist dann nämlich $1/(\mu - \lambda_j)$ der betragsgrößte Eigenwert von $(\mu I - A)^{-1}$. Anwendung der Potenzenmethode auf $(\mu I - A)^{-1}$ liefert

$$y_0 \in \mathbb{C}^n \text{ beliebig,} \quad (\mu I - A)y_{k+1} = y_k,$$

bzw. die normierte Version in Algorithmus 6.2.

Algorithmus 6.2 Inverse Potenzenmethode mit Shift

```

 $\mu \in \mathbb{C}$  gegebener Shift,  $y_0 \neq 0$  gegebener Startvektor
 $y_0 = y_0 / \|y_0\|$ 
Berechne die LU-Zerlegung von  $\mu I - A$ 
for  $k = 0, 1, \dots$  do
    Löse  $(\mu I - A)z_{k+1} = y_k$  mit der LU-Zerlegung
     $y_{k+1} = \frac{1}{\|z_{k+1}\|} z_{k+1}$ 
     $\varrho_{k+1} = \varrho_A(y_{k+1}) = y_{k+1}^H A y_{k+1}$ 
end for

```

Die linearen Gleichungssysteme lassen sich mit nur einer einzigen LU-Zerlegung der Koeffizientenmatrix lösen. Satz 6.18 ist direkt anwendbar und liefert Konvergenz mit Konvergenzfaktor

$$\eta = \max_{k \neq j} \frac{|\mu - \lambda_k|^{-1}}{|\mu - \lambda_j|^{-1}} = \max_{k \neq j} \frac{|\mu - \lambda_j|}{|\mu - \lambda_k|} \ll 1.$$

Beispiel. Wählen wir im obigen Beispiel $\mu = 3.4117647\dots$ (also die Näherung aus zwei Schritten der Potenzenmethode) und den zugehörigen Vektor y_2 als Startvektor, so erhalten wir

k	$\varrho(y_k)$	$ \varrho(y_k) - \lambda_1 $
1	3.414213562319	$5.42 \cdot 10^{-11}$
2	3.414213562373	$4.44 \cdot 10^{-16}$
3	3.414213562373	0

Der Konvergenzfaktor ist hier $\eta \approx 1.734 \cdot 10^{-3}$. Gegenüber der Potenzenmethode erreicht man also eine wesentliche Konvergenzbeschleunigung. \diamond

Eine weitere Konvergenzbeschleunigung erzielt man, wenn man den Shift in jedem Iterationsschritt der inversen Potenzenmethode durch die aktuelle Näherung für den gesuchten Eigenwert ersetzt. Da jetzt in jedem Iterationsschritt eine neue LU-Zerlegung der geschifteten

Algorithmus 6.3 Rayleigh-Quotienten Iteration

$\mu_0 \in \mathbb{C}$ gegebener Shift, $y_0 \neq 0$ gegebener Startvektor, $y_0 = y_0 / \|y_0\|$
for $k = 0, 1, \dots$ **do**
 Löse $(\mu_k I - A)z_{k+1} = y_k$
 $y_{k+1} = \frac{1}{\|z_{k+1}\|} z_{k+1}$
 $\mu_{k+1} = \varrho_A(y_{k+1}) = y_{k+1}^H A y_{k+1}$
end for

Matrix berechnet werden muss, ist der Aufwand wesentlich größer als der der inversen Iteration. Für diesen Mehraufwand wird man zumindest im Fall normaler Matrizen mit lokal kubischer Konvergenz belohnt:

Satz 6.19. *Ist A normal, dann konvergiert die Folge $\{\mu_k\}_k$ der Rayleigh-Quotienten Iteration lokal kubisch gegen einen Eigenwert von A , d. h. wenn die Folge y_k gegen einen Eigenvektor konvergiert, dann konvergiert μ_k gegen den zugehörigen Eigenwert.*

Beweis. (Demmel 1997, Theorem 5.9) Wir zeigen zunächst, dass es genügt, die Aussage für Diagonalmatrizen zu beweisen. Da A normal ist, existiert eine unitäre Matrix U , so dass $U^H A U = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$. Zu y_k, z_k aus Algorithmus 6.3 definieren wir

$$\hat{y}_k = U^H y_k, \quad \hat{z}_k = U^H z_k, \quad k = 0, 1, \dots$$

Es ist $\|\hat{y}_k\| = \|y_k\|$ und $\|\hat{z}_k\| = \|z_k\|$. Nach einem Schritt gilt

$$\hat{z}_{k+1} = U^H (\mu_k I - A)^{-1} U U^H y_k = (\mu_k I - \Lambda)^{-1} \hat{y}_k$$

und

$$\hat{y}_{k+1}^H \Lambda \hat{y}_{k+1} = (U^H y_{k+1})^H \Lambda (U^H y_{k+1}) = y_{k+1}^H U \Lambda U^H y_{k+1} = y_{k+1}^H A y_{k+1}.$$

Damit ist es äquivalent, Algorithmus 6.3 mit Startwert y_0 auf A oder mit Startwert \hat{y}_0 auf Λ anzuwenden. Ohne Einschränkung können wir daher annehmen, dass A bereits in Diagonalf orm ist.

Nehmen wir weiter ohne Einschränkung an, dass y_k gegen e_1 konvergiert, d. h.

$$y_k = e_1 + d_k \quad \text{mit} \quad \|d_k\| = \epsilon \ll 1. \quad (6.3)$$

Um kubische Konvergenz zu beweisen, müssen wir $\|d_{k+1}\| = O(\epsilon^3)$ zeigen.

Es gilt

$$1 = \|y_k\|^2 = \|e_1 + d_k\|^2 = 1 + \epsilon^2 + 2\text{Re } d_{k,1},$$

also $\text{Re } d_{k,1} = -\epsilon^2/2$. Der zugehörige Rayleigh-Quotient ist durch

$$\mu_k = \varrho(y_k) = (e_1 + d_k)^H \Lambda (e_1 + d_k) = \lambda_1 + \eta_k, \quad (6.4)$$

mit

$$\eta_k = 2\lambda_1 \text{Re } d_{k,1} + d_k^H \Lambda d_k = -\epsilon^2 \lambda_1 + d_k^H \Lambda d_k$$

gegeben. Hieraus folgt

$$|\eta_k| \leq \epsilon^2 |\lambda_1| + \epsilon^2 \|\Lambda\| \leq 2\epsilon^2 \|\Lambda\|,$$

also ist $\mu_k = \lambda_1 + O(\epsilon^2)$.

Nun betrachten wir $z_{k+1} = (\mu_k I - \Lambda)^{-1} y_k$. Wegen (6.3) und (6.4) ist

$$z_{k+1,1} = \frac{y_{k,1}}{\eta_k}, \quad z_{k+1,j} = \frac{d_{k,j}}{\mu_k - \lambda_j} = \frac{y_{k,1}}{\eta_k} \frac{d_{k,j} \eta_k}{y_{k,1}(\lambda_1 - \lambda_j + \eta_k)}.$$

Daher können wir

$$z_{k+1} = \frac{y_{k,1}}{\eta_k} (e_1 + \hat{d}_{k+1})$$

schreiben. Für $y_{k,1}$ gelten wegen $\|y_k\| = 1$ und (6.3) die Abschätzungen

$$|y_{k,1}| \leq 1 \quad \text{und} \quad |y_{k,1}| \geq 1 - |d_{k,1}| \geq 1 - \epsilon.$$

Definieren wir $\text{gap} = \min_{j \neq 1} |\lambda_1 - \lambda_j|$ und wählen ϵ so klein, dass $|\eta_k| \leq \text{gap}/2$ gilt, so können wir wegen $\hat{d}_{k+1,1} = 0$ die Abschätzung

$$\|\hat{d}_{k+1}\| \leq \frac{\|d_k\| \cdot |\eta_k|}{(1 - \epsilon)(\text{gap} - |\eta_k|)} \leq \frac{2\epsilon^3 \|\Lambda\|}{(1 - \epsilon)(\text{gap} - |\eta_k|)} = O(\epsilon^3)$$

schließen. Da nach Voraussetzung $y_k \rightarrow e_1$ folgt aus

$$y_{k+1} = e_1 + d_{k+1} = (e_1 + \hat{d}_{k+1}) / \|e_1 + \hat{d}_{k+1}\|$$

auch $\|d_{k+1}\| = O(\epsilon^3)$. □

6.5 QR-Algorithmus

Wir nehmen weiterhin an, dass die Eigenwerte nach absteigendem Betrag sortiert sind. Mit Hilfe der Potenzenmethode haben wir Näherungen an den betragsgrößten Eigenwert λ_1 und einen zugehörigen normierten Eigenvektor u_1 mit Hilfe der Iteration

$$Ay_k = \lambda_1^{(k+1)} y_{k+1}, \quad \lambda_1^{(k+1)} \text{ so, dass } \|y_{k+1}\| = 1$$

berechnet. In diesem Abschnitt wollen wir nun ausgehend von dieser einfachen Iteration die gesamte Schurform der Matrix A konstruieren. Sind aus der Potenzenmethode λ_1 und u_1 bekannt, so betrachten wir nun die Menge

$$\mathcal{S}_1 = u_1^\perp = \{u \in \mathbb{C}^n \mid u_1^H u = 0\}$$

der Dimension $n - 1$. Da alle anderen Vektoren in der Schurform orthogonal zu u_1 sind, genügt es, die Iteration in \mathcal{S}_1 fortzuführen. Dazu definieren wir auf \mathcal{S}_1 die Abbildung

$$\mathcal{A}_1 : \mathcal{S}_1 \xrightarrow{A|_{\mathcal{S}_1}} \mathbb{C}^n \xrightarrow{P_1 \text{ orth. Proj.}} \mathcal{S}_1.$$

Der Orthogonalprojektor P_1 auf \mathcal{S}_1 ist durch

$$P_1 = I - u_1 u_1^H$$

gegeben, denn $P_1^2 = P_1$, $P_1 u_1 = 0$ und $P_1 x = x$ für alle $x \in \mathcal{S}_1$.

Lemma 6.20. Ist $U^H A U = R = D + N$ die Schurform von A und $A_1 := P_1 A$, dann ist $U^H A_1 U = \begin{bmatrix} 0 & \\ & I \end{bmatrix} R$ die Schurform von A_1 . Insbesondere hat A_1 die Eigenwerte $0, \lambda_2, \dots, \lambda_n$.

Beweis. Nach Voraussetzung und Definition von A_1 ist

$$\begin{aligned} A_1 U &= (I - u_1 u_1^H) U R U^H U \\ &= (U - u_1 e_1^T) R \\ &= \begin{bmatrix} 0 & u_2 & \cdots & u_n \end{bmatrix} R \\ &= U \begin{bmatrix} 0 & \\ & I \end{bmatrix} R \end{aligned}$$

die Schurform von A_1 . □

Nach diesem Lemma ist λ_2 der betragsgrößte Eigenwert von A_1 und kann damit mit Hilfe der Potenzenmethode bestimmt werden. Wichtig ist, dass man dazu nicht A_1 explizit berechnen muss, sondern nur Matrix-Vektorprodukte mit A_1 benötigt werden. Selbst wenn A dünn besetzt ist, ist A_1 nämlich im Allgemeinen voll besetzt, was dieses Verfahren für große Dimensionen unbrauchbar machen würde. $A_1 x$ kann man durch

$$A_1 x = (I - u_1 u_1^H) A x = A x - \beta u_1, \quad \beta = u_1^H A x$$

mit einer Matrix-Vektormultiplikation und einem SAXPY effizient implementieren.

Diese Konstruktion führt man nun fort, indem man zunächst mit $\mathcal{S}_2 = \text{span}\{u_1, u_2\}^\perp$ die Abbildung \mathcal{A}_2 definiert, dann \mathcal{S}_3 und \mathcal{A}_3 , usw.. Effizienter ist es jedoch, all diese Schritte simultan durchzuführen. Dazu startet man mit einer Orthonormalbasis, die durch eine unitäre Matrix U_0 gegeben ist, und berechnet damit $A U_0$. Da die so erhaltene Matrix im Allgemeinen nicht mehr unitär ist, muss man die Orthogonalität mit Hilfe einer QR-Zerlegung von $A U_0$ wiederherstellen:

Algorithmus 6.4 Simultane Iteration

$U_0 \in \mathbb{C}^{n,m}$ unitär, $U_0^H U_0 = I_m$, $A_0 = A$
for $k = 0, 1, 2, \dots$ **do**
 Setze $Y_{k+1} = A U_k$
 Berechne die QR-Zerlegung $Y_{k+1} = U_{k+1} R_{k+1}$
 $A_{k+1} = U_{k+1}^H A U_{k+1}$
end for

Ohne die Orthogonalisierung der Spalten von U_k in jedem Schritt würden alle Spalten von Y_{k+1} gegen einen Eigenvektor zum betragsgrößten Eigenwert konvergieren, denn dann entspräche der Algorithmus der Potenzenmethode simultan angewandt auf m Startvektoren. Die QR-Zerlegung zur Orthogonalisierung ist also wesentlich.

Satz 6.21. Sei $A = X \Lambda X^{-1}$ diagonalisierbar, alle Eigenwerte von A paarweise betragsmäßig verschieden und alle Hauptuntermatrizen von X ($X_{1:j,1:j}$) nicht singulär. Dann konvergiert die Folge $\{A_k\}_k = U_k^H A U_k$ mit U_k aus Algorithmus 6.4, $U_0 = I$ gegen die Schurform von A ,

$$U_k^H A U_k \rightarrow R, \quad \text{diag}(R) = \text{diag}(\lambda_1, \dots, \lambda_n).$$

Beweisskizze. Die Voraussetzung an X entspricht der Voraussetzung $\alpha_1 \neq 0$ in Satz 6.18 und zwar in der Form, dass für alle $m = 1, \dots, n$ für die Startvektoren $\{e_1, \dots, e_m\}$ der simultanen Iteration gesichert ist, dass sie Beiträge im von den Eigenvektoren x_1, \dots, x_m aufgespannten Raum haben. Dies ist klar, wenn

$$\begin{bmatrix} e_1 & \cdots & e_m \end{bmatrix}^T \begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix} = \begin{bmatrix} I_m & 0 \end{bmatrix} X \begin{bmatrix} I_m \\ 0 \end{bmatrix}$$

vollen Rang hat, also die Voraussetzung an die Hauptuntermatrizen von X erfüllt ist.

Nach Konstruktion ist U_k unitär, also A_k ähnlich zu A . Für $1 \leq m \leq n$ beliebig spalten wir U_k gemäß

$$U_k = \begin{bmatrix} V_k & W_k \end{bmatrix}, \quad V_k \in \mathbb{C}^{n,m}, W_k \in \mathbb{C}^{n,n-m}$$

auf. Dann ist

$$A_k = \begin{bmatrix} V_k^H A V_k & V_k^H A W_k \\ W_k^H A V_k & W_k^H A W_k \end{bmatrix}$$

Durch Verallgemeinerung des Satzes über die Konvergenz der Potenzenmethode kann man zeigen, dass $\mathcal{R}(V_k)$ gegen den Unterraum $\text{span}\{u_1, \dots, u_m\}$ konvergiert. Da dieser Raum A -invariant ist, konvergiert $A V_k$ ebenfalls dagegen. Wegen $W_k^H V_k = 0$ konvergiert dann $W_k^H A V_k \rightarrow 0$. Da dies für alle m richtig ist, konvergiert A_k gegen R . \square

Eine äquivalente Formulierung der simultanen Iteration für $m = n$, die sich zur Konstruktion von Varianten mit besseren Eigenschaften als günstiger erweisen wird, ist der QR-Algorithmus 6.5. Die Äquivalenz ist so gemeint, dass die Matrizen A_k bis auf unitäre Ähnlichkeitstransformation mit Diagonalmatrizen übereinstimmen. Sie basiert darauf, dass die QR-Zerlegung einer Matrix im Wesentlichen eindeutig ist (Übung).

Algorithmus 6.5 QR-Algorithmus

```

 $A_0 = A$ 
for  $k = 0, 1, 2, \dots$  do
    Berechne die QR-Zerlegung  $A_k = Q_k R_k$ 
    Setze  $A_{k+1} = R_k Q_k$ 
end for
```

In dieser Form sollte der QR-Algorithmus jedoch nicht angewendet werden, denn wir werden sehen, dass untere Nebendiagonalelemente der Schurform mit Konvergenzfaktor $\eta_m = |\lambda_{m+1}/\lambda_m|$ gegen Null konvergieren. Wie bei der Potenzenmethode ist die Konvergenz sehr langsam, wenn $\eta_m \approx 1$. Die dort zur Beschleunigung verwendete inverse Iteration mit Shifts kann auch beim QR-Algorithmus eingesetzt werden. Es ergibt sich Algorithmus 6.6.

Algorithmus 6.6 QR-Algorithmus mit Shifts

```

 $A_0 = A$ 
for  $k = 0, 1, 2, \dots$  do
    Wähle einen Shift  $\mu_k$ 
    Berechne die QR-Zerlegung  $A_k - \mu_k I = Q_k R_k$ 
    Setze  $A_{k+1} = R_k Q_k + \mu_k I$ 
end for
```

Noch allgemeiner ist der QR-Algorithmus 6.7 mit mehrfachen Shifts, die durch Shiftpolynome

$$f_k(\lambda) = \prod_{i=1}^r (\lambda - \mu_i^{(k)})$$

definiert sind. Ein Mehrfachshift vom Grad r ist äquivalent zu einer Folge von r Mehrfachshifts vom Grad 1 (Übung).

Algorithmus 6.7 QR-Algorithmus mit mehrfachen Shifts

```

 $A_0 = A$ 
for  $k = 0, 1, 2, \dots$  do
  Wähle ein Shiftpolynom  $f_k$ 
  Berechne die QR-Zerlegung  $f_k(A_k) = Q_k R_k$ 
  Setze  $A_{k+1} = Q_k^H A_k Q_k$ 
end for
  
```

Lemma 6.22. Für den QR-Algorithmus 6.7 gilt

$$\begin{aligned} A_k &= U_k^H A U_k, & U_k &= Q_0 \cdots Q_{k-1} \\ p_k(A) &= \prod_{i=0}^k f_i(A) = U_{k+1} T_{k+1}, & T_{k+1} &= R_k \cdots R_0 \end{aligned}$$

Beweis. Die erste Gleichung folgt direkt aus der Iterationsvorschrift, die zweite beweisen wir mit Induktion nach k . Für $k = 0$ ist sie offensichtlich richtig, denn

$$f_0(A) = f_0(A_0) = Q_0 R_0 = U_1 T_1.$$

Nehmen wir also an, wir hätten die Behauptung für $k - 1$ bereits gezeigt. Dann ergibt sich aus $A = U_k A_k U_k^H$, der Induktionsannahme und der Iterationsvorschrift

$$f_k(A) f_{k-1}(A) \cdots f_0(A) = f_k(A) U_k T_k = U_k f_k(A_k) T_k = U_k Q_k R_k T_k = U_{k+1} T_{k+1}$$

und damit die Behauptung für alle $k = 0, 1, \dots$. □

Definition 6.23. Mit $G(m, n, X) \subset \mathbb{C}^{n,n}$, $1 \leq m < n$ bezeichnen wir die Teilmenge aller Matrizen $A \in \mathbb{C}^{n,n}$ mit folgenden Eigenschaften

- (a) $A = X \Lambda X^{-1}$ mit Blockdiagonalmatrix $\Lambda = \text{diag}(\Lambda_1, \Lambda_2)$, wobei $\Lambda_1 \in \mathbb{C}^{m,m}$ und $\Lambda_2 \in \mathbb{C}^{n-m,n-m}$.
- (b) Die Hauptuntermatrix der Dimension m von X^{-1} (Zeilen und Spalten 1 bis m von X^{-1}) ist nicht singulär.

Lemma 6.24. (Tyrtysnikov 1997, Lemma 10.4.1) Es sei $A \in G(m, n, X)$ gegeben und A_1 aus einem Schritt des allgemeinen QR-Algorithmus 6.7 mit dem Shiftpolynom $f = f_0$ berechnet. Zerlegen wir A und A_1 gemäß

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = A_1 = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

mit $A_{11}, B_{11} \in \mathbb{C}^{m,m}$, $A_{22}, B_{22} \in \mathbb{C}^{n-m,n-m}$, und setzen wir voraus, dass $F_1 = f(\Lambda_1)$ und $F_2 = f(\Lambda_2)$ nicht singulär sind, dann gilt

$$\|B_{21}\| \leq c_1(1 + c_2\phi)^2\phi\|A_{21}\|, \quad \phi = \|F_2\| \|F_1^{-1}\|$$

mit Konstanten $c_1, c_2 > 0$, die nur von m und X abhängen.

Beweis. Ein Schritt des QR-Algorithmus 6.7 liefert

$$f(A) = QR, \quad B = A_1 = Q^H A Q. \quad (6.5)$$

Die Voraussetzung $A \in G(m, n, X)$ erlaubt eine Block-LU-Zerlegung von X^{-1} der Form

$$X^{-1} = LU = \begin{bmatrix} I_m & 0 \\ L_{21} & I_{n-m} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}, \quad U_{11} \in \mathbb{C}^{m,m}.$$

Die Matrizen L und U sind invertierbar und man rechnet leicht nach, dass

$$L^{-1} = \begin{bmatrix} I_m & 0 \\ -L_{21} & I_{n-m} \end{bmatrix}, \quad U^{-1} = \begin{bmatrix} U_{11}^{-1} & -U_{11}^{-1}U_{12}U_{22}^{-1} \\ 0 & U_{22}^{-1} \end{bmatrix}.$$

Da wiederum nach Voraussetzung die Blockdiagonalmatrix $F = f(\Lambda)$ nicht singulär ist, folgt aus (6.5)

$$\begin{aligned} B &= Q^H A Q = R f(A)^{-1} A f(A) R^{-1} \\ &= R A R^{-1} \\ &= R U^{-1} L^{-1} \Lambda L U R^{-1} \\ &= (R U^{-1} F^{-1}) (F L^{-1} \Lambda L F^{-1}) (F U R^{-1}) \end{aligned}$$

Uns interessiert die Norm von B_{21} . Da $F U R^{-1}$ und natürlich ebenso die Inverse $R U^{-1} F^{-1}$ obere Blockdreiecksmatrizen sind und $F L^{-1} \Lambda L F^{-1}$ eine untere Blockdreiecksmatrix ist, gilt

$$\begin{aligned} B_{21} &= (R U^{-1} F^{-1})_{22} (F L^{-1} \Lambda L F^{-1})_{21} (F U R^{-1})_{11} \\ &= (R U^{-1} F^{-1})_{22} (F_2 (L^{-1} \Lambda L)_{21} F_1^{-1}) (F U R^{-1})_{11} \end{aligned} \quad (6.6)$$

$$\begin{aligned} &\begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \quad (6.7) \end{aligned}$$

Wir schätzen jetzt die Normen dieser drei Faktoren einzeln ab. Da

$$Q = f(A) R^{-1} = X F X^{-1} R^{-1} = X (F L F^{-1}) (F U R^{-1})$$

unitär ist, folgt

$$\|F U R^{-1}\| = \|F L^{-1} F^{-1} X^{-1} Q\| \leq \|F L^{-1} F^{-1}\| \|X^{-1}\|$$

und

$$\|Q R U^{-1} F^{-1}\| = \|R U^{-1} F^{-1}\| \leq \|X\| \|F L F^{-1}\|.$$

Wegen

$$F L F^{-1} = \begin{bmatrix} I & 0 \\ F_2 L_{21} F_1^{-1} & I \end{bmatrix}$$

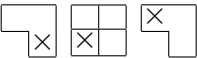
gilt

$$\|FLF^{-1}\| \leq (1 + \|L_{21}\|\phi), \quad \|FL^{-1}F^{-1}\| \leq (1 + \|L_{21}\|\phi).$$

Für den mittleren Term in (6.6) verwenden wir, dass wegen

$$A = X\Lambda X^{-1} = U^{-1}L^{-1}\Lambda LU$$

und der oberen Blockdreiecksform von U

$$(L^{-1}\Lambda L)_{21} = (UAU^{-1})_{21} = U_{22}A_{21}U_{11}^{-1}$$


gilt. Fassen wir alle Ungleichungen zusammen, so ergibt sich aus

$$\|B_{21}\| \leq (1 + \|L_{21}\|\phi)^2 \phi \kappa(X) \|U_{22}\| \cdot \|A_{21}\| \cdot \|U_{11}^{-1}\|$$

die Behauptung. □

Als weiteres Hilfsmittel benötigen wir

Lemma 6.25. *Zu jeder Matrix $A \in \mathbb{C}^{n,n}$ mit Spektralradius $\rho = \rho(A)$ und zu jedem $\epsilon > 0$ gibt es ein $N = N(A, \epsilon)$, so dass $\|A^k\| \leq (\rho + \epsilon)^k$ für alle $k \geq N$. Hierbei ist $\|\cdot\|$ eine beliebige p -Norm.*

Beweis. Die Matrix $\tilde{A} = (\rho + \epsilon)^{-1}A$ hat Spektralradius < 1 . Daher gibt es nach Satz 6.8 eine Matrixnorm $\|\cdot\|_T$ mit $\|\tilde{A}\|_T < 1$ und es gilt

$$\|\tilde{A}^k\|_T \leq \|\tilde{A}\|_T^k \rightarrow 0, \quad k \rightarrow \infty.$$

Da in $\mathbb{C}^{n,n}$ alle Normen äquivalent sind, gibt es ein $N = N(A, \epsilon)$, so dass $\|\tilde{A}^k\| \leq 1$ für alle $k \geq N$ oder äquivalent $\|A^k\| \leq (\rho + \epsilon)^k$ für alle $k \geq N$ gilt. □

Mit Hilfe dieser Lemmas können wir Konvergenz für den einfachsten Fall des QR-Algorithmus ohne Shift beweisen.

Satz 6.26. (Tyrtysnikov 1997, Lemma 10.5.1) *Es sei $A \in G(m, n, X)$ nicht singulär gegeben mit $\lambda(\Lambda_1) = \{\lambda_1, \dots, \lambda_m\}$ und $\lambda(\Lambda_2) = \{\lambda_{m+1}, \dots, \lambda_n\}$. Ferner sei $\eta = |\lambda_{m+1}|/|\lambda_m| < 1$. Dann generiert der QR-Algorithmus 6.5 ohne Shifts Matrizen*

$$A_k = \begin{bmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ A_{21}^{(k)} & A_{22}^{(k)} \end{bmatrix},$$

derart, dass für jedes $\rho \in (\eta, 1)$ ein $N = N(\rho, \Lambda_1, \Lambda_2) \in \mathbb{N}$ existiert mit

$$\|A_{21}^{(k)}\| \leq c(\rho)\rho^k \quad \text{für alle } k \geq N.$$

Insbesondere geht $A_{21}^{(k)} \rightarrow 0$ für $k \rightarrow \infty$.

Beweis. k Schritte von Algorithmus 6.5 sind äquivalent zu einem Schritt des allgemeinen QR-Algorithmus 6.7 mit $f(\lambda) = \lambda^k$. Nach Lemma 6.22 gilt in der Notation von Algorithmus 6.5 also

$$A^k = A_k = U_{k+1}T_{k+1}, \quad A_k = U_k^H A U_k.$$

Die Voraussetzungen von Lemma 6.24 sind wegen der Invertierbarkeit von A erfüllt, also gibt es eine Konstante c mit

$$\|A_{21}^{(k)}\| \leq c\|\Lambda_2^k\|\|\Lambda_1^{-k}\|.$$

Aus Lemma 6.25 wissen wir, dass es zu jedem $\delta > 0$ ein $N = N(\delta, \Lambda_1, \Lambda_2)$ gibt, so dass

$$\|\Lambda_2^k\| \leq (|\lambda_{m+1}| + \delta)^k, \quad \|\Lambda_1^{-k}\| \leq \left(\frac{1}{|\lambda_m|} + \delta\right)^k$$

für $k \geq N$. □

Korollar 6.27. *Ist unter den Annahmen von Satz 6.26 die Matrix Λ normal, dann gilt sogar*

$$\|A_{21}^{(k)}\| \leq c\eta^k \quad \text{für alle } k = 1, 2, \dots$$

Beweis. Gegenüber dem Beweis von Satz 6.26 vereinfacht sich die letzte Abschätzung zu $\|\Lambda_2^k\| = |\lambda_{m+1}|^k$ und $\|\Lambda_1^{-k}\| = \frac{1}{|\lambda_m|^k}$. □

Bemerkung. Insbesondere gilt Korollar 6.27 also für Λ diagonal oder Hermitesch.

Korollar 6.28. *Sind die Annahmen von Satz 6.26 für jedes $1 \leq m < n$ erfüllt, also insbesondere alle Eigenwerte von A betragsmäßig verschieden, und existiert die LU-Zerlegung von X^{-1} , dann gilt*

$$\lim_{k \rightarrow \infty} (A_k)_{ij} = 0, \quad i > j \qquad \lim_{k \rightarrow \infty} \text{diag}(A_k) = \Lambda.$$

Beweis. Folgt aus Korollar 6.27. □

Bemerkungen.

- (a) Sind die Voraussetzungen von Satz 6.26 nicht erfüllt, dann konvergieren im ungünstigsten Fall gewisse untere Diagonalblöcke nicht gegen Null.
- (b) Die Voraussetzungen können durch beliebig kleine Störungen von A erfüllt werden.
- (c) Die Voraussetzung der sogenannten strengen Regularität von X^{-1} ist nicht wesentlich. Dazu verwendet man, dass für jede nicht singuläre Matrix A die **modifizierte Bruhat-Zerlegung**

$$A = LPU$$

mit einer Permutationsmatrix P und nicht singulären unteren und oberen Dreiecksmatrizen L und U existiert. Sind außer der strengen Regularität von X^{-1} alle Voraussetzungen von Korollar 6.28 erfüllt, dann konvergiert A_k gegen eine untere Dreiecksmatrix mit $\text{diag}(A_k) \rightarrow \text{diag}(P^T \Lambda P)$. In der Praxis wird man dieses Konvergenzverhalten wegen (b) fast nie beobachten.

Von wesentlich größerer Bedeutung ist die Konvergenz des QR-Algorithmus mit Shifts, bei der die Wahl der Shifts natürlich eine entscheidende Rolle spielt. Nach Lemma 6.24 können wir schnelle Konvergenz erwarten, wenn wir das Shiftpolynom so wählen, dass $\phi = \|f(\Lambda_2)\| \cdot \|f(\Lambda_1)^{-1}\| \ll 1$ ist. Das Polynom f muss also klein auf $r = n - m$ Eigenwerten $\lambda_{m+1}, \dots, \lambda_n$ und groß auf m Eigenwerten $\lambda_1, \dots, \lambda_m$ sein (die Eigenwerte sind jetzt nicht mehr sortiert). Wir diskutieren hier die Konvergenz für verallgemeinerte Rayleigh-Shifts, die wie folgt definiert sind.

Definition 6.29. Ein r -facher Mehrfachshift wird ein (r -facher) **Rayleigh-Shift** genannt, wenn $f_k(\lambda) = \det(A_{22}^{(k)} - \lambda I)$ das charakteristische Polynom des $(2, 2)$ -Blocks der Dimension r von A_k ist.

Im Spezialfall $r = 1$ wählt man also das Element $a_{n,n}^{(k)}$ als Shift. Dies ist sinnvoll, da in diesem Schritt das Element $a_{n,n}^{(k)}$ die beste verfügbare Approximation an λ_n ist, in diesem Sinne also $|f(\lambda)| = |\lambda_n - \lambda|$ minimiert.

Satz 6.30. Sei $A \in G(m, n, X)$ mit Λ diagonal und sei der allgemeine QR-Algorithmus 6.7 mit Rayleigh-Mehrfachshifts vom Grad $r = n - m$ konvergent, d. h.

$$\epsilon_k := \|A_{21}^{(k)}\| \rightarrow 0.$$

Dann ist die Konvergenz quadratisch, d. h. es gibt eine Konstante $c > 0$ mit $\epsilon_{k+1} \leq c\epsilon_k^2$.

Beweis. Für einen Schritt von Algorithmus 6.7 gilt nach Lemma 6.24

$$\epsilon_{k+1} \leq c\epsilon_k \|f_k(\Lambda_2)\| \cdot \|f_k(\Lambda_1)^{-1}\|.$$

Es seien $\mu_1^{(k)}, \dots, \mu_r^{(k)}$ die Eigenwerte des Blocks $A_{22}^{(k)}$. Die Shifts $\mu_j^{(k)}$ fassen wir als Eigenwerte der gestörten Matrix

$$A_k + E = \begin{bmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 0 \\ -A_{21}^{(k)} & 0 \end{bmatrix}$$

auf. Nach Satz 6.9 von Bauer und Fike gibt es r Eigenwerte von A , sagen wir $\lambda_{m+1}, \dots, \lambda_n$, so dass

$$|\lambda_{m+i} - \mu_i^{(k)}| \leq \kappa(X)\epsilon_k, \quad i = 1, \dots, r \quad (6.8)$$

gilt, denn aus

$$A_k = U_k^H A U_k = U_k^H X \Lambda X^{-1} U_k$$

mit einer unitären Matrix U_k folgt $\kappa(U_k^H X) = \kappa(X)$ für die Kondition der Eigenvektormatrix von A_k . Für ϵ_k hinreichend klein kann man auch garantieren, dass nicht zwei Shifts $\mu_j^{(k)}$ Abstand $\leq \kappa(X)\epsilon_k$ von λ_{m+j} haben. Die Zuordnung in (6.8) ist damit eindeutig. Nehmen wir weiter $\kappa(X)\epsilon_k \leq 1$ an, dann gibt es Konstanten c_1, c_2 unabhängig von k , so dass

$$\begin{aligned} |f_k(\lambda_{m+j})| &= \left| \prod_{i=1}^r (\lambda_{m+j} - \mu_i^{(k)}) \right| \\ &= \prod_{i=1}^r |\lambda_{m+j} - \lambda_{m+i} + \lambda_{m+i} - \mu_i^{(k)}| \\ &\leq (2\|A\| + \kappa(X)\epsilon_k)^{r-1} \kappa(X)\epsilon_k \\ &\leq c_1 \epsilon_k, \end{aligned} \quad j = 1, \dots, r,$$

wobei $c_1 = (2\|A\| + 1)^{r-1}$. Ist ϵ_k so klein gewählt, dass

$$\kappa(X)\epsilon_k \leq \gamma/2, \quad \gamma = \min_{\substack{1 \leq i \leq m \\ 1 \leq j \leq r}} |\lambda_i - \lambda_{m+j}| > 0,$$

dann gilt sogar $c_1 = (2\|A\| + \frac{\gamma}{2})^{r-1}$ und analog folgt

$$\begin{aligned} |f_k(\lambda_j)| &= \left| \prod_{i=1}^r (\lambda_j - \mu_i^{(k)}) \right| \\ &\geq \prod_{i=1}^r (|\lambda_j - \lambda_{m+i}| - |\lambda_{m+i} - \mu_i^{(k)}|) \\ &\geq \prod_{i=1}^r (|\lambda_j - \lambda_{m+i}| - \kappa(X)\epsilon_k) \\ &\geq \left(\frac{\gamma}{2}\right)^r =: c_2 \end{aligned} \quad j = 1, \dots, m.$$

Hieraus ergibt sich $\|f_k(\Lambda_2)\| \leq c_1\epsilon_k$ und $\|f_k(\Lambda_1)^{-1}\| \leq 1/c_2$, also $\epsilon_{k+1} \leq c\epsilon_k^2$. \square

Wie bei der Rayleigh-Quotienten-Iteration kann man unter weiteren Voraussetzungen an A sogar kubische Konvergenz zeigen. Dazu benötigen wir zunächst eine Variante des Satzes von Bauer und Fike:

Lemma 6.31. *Es sei $A = \text{diag}(A_1, A_2) \in \mathbb{C}^{n,n}$ eine diagonalisierbare Blockdiagonalmatrix,*

$$C_1^{-1}A_1C_1 = \Lambda_1 = \text{diag}(\lambda_1, \dots, \lambda_m), \quad C_2^{-1}A_2C_2 = \Lambda_2 = \text{diag}(\lambda_{m+1}, \dots, \lambda_n).$$

Ist μ ein Eigenwert der gestörten Matrix

$$A + E \quad \text{mit} \quad E = \begin{bmatrix} 0 & E_1 \\ E_2 & 0 \end{bmatrix}, \quad E_1 \in \mathbb{C}^{m,n-m},$$

dann gilt

$$\min_{1 \leq j \leq m} |\lambda_j - \mu| \min_{m+1 \leq j \leq n} |\lambda_j - \mu| \leq \kappa(C_1)\kappa(C_2) \|E_1\| \cdot \|E_2\|.$$

Beweis. Da für $\mu \in \lambda(A)$ nichts zu zeigen ist, können wir im Folgenden $\mu \notin \lambda(A)$ annehmen. Wegen $\mu \in \lambda(A + E)$ ist die Matrix $A + E - \mu I$ singulär und gleiches gilt für

$$C^{-1}(A + E - \mu I)C = \begin{bmatrix} \Lambda_1 - \mu I & C_1^{-1}E_1C_2 \\ C_2^{-1}E_2C_1 & \Lambda_2 - \mu I \end{bmatrix}, \quad C = \text{diag}(C_1, C_2).$$

Es existiert daher ein $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \neq 0$ mit

$$(\Lambda_1 - \mu I)y_1 = -C_1^{-1}E_1C_2y_2, \quad (\Lambda_2 - \mu I)y_2 = -C_2^{-1}E_2C_1y_1.$$

Da $\mu \notin \lambda(A)$ folgt $y_1 \neq 0$ und $y_2 \neq 0$, denn sonst wäre $y = 0$. Durch Elimination von y_2 erhält man

$$y_1 = (\Lambda_1 - \mu I)^{-1}C_1^{-1}E_1C_2(\Lambda_2 - \mu I)^{-1}C_2^{-1}E_2C_1y_1.$$

Nimmt man in diesen Gleichungen auf beiden Seiten die Norm und teilt durch $\|y_1\|$, dann ergibt sich die Behauptung. \square

Satz 6.32. *Gilt zusätzlich zu den Voraussetzungen von Satz 6.30 $\|A_{21}^{(k)}\| = \|A_{12}^{(k)}\|$ für jedes $k = 0, 1, 2, \dots$ (insbesondere also für A Hermitesch), so ist die Konvergenz sogar kubisch.*

Beweis. Der Beweis geht analog zu dem von Satz 6.30. Die Shifts sind Eigenwerte der Blockdiagonalmatrix $\text{diag}(A_{11}^{(k)}, A_{22}^{(k)})$. Fasst man A_k als Störung dieser Blockdiagonalmatrix auf, dann liefert obige Variante des Satzes von Bauer und Fike mit $E_1 = A_{21}^{(k)}$ und $E_2 = A_{12}^{(k)}$ in (6.8) auf der rechten Seite den Faktor $\kappa(C_1^{(k)})\kappa(C_2^{(k)})\epsilon_k^2$. Jetzt muss man noch zeigen, dass $\kappa(C_1^{(k)})\kappa(C_2^{(k)})$ unabhängig von k beschränkt bleibt. Dies ist im wichtigen Spezialfall, dass A Hermitesch ist, klar. Für den allgemeinen Fall für ϵ_k hinreichend klein folgt dies aus Theorem 3 und 5 in Smith, The condition numbers of the matrix eigenvalue problem, Numer. Math. 10:232–240 (1967). \square

6.6 Effiziente Implementierung des QR-Algorithmus

Im letzten Abschnitt haben wir gesehen, dass der QR-Algorithmus mit Rayleigh-Shifts sehr gute Konvergenzeigenschaften hat. In der bisherigen Form ist er jedoch sehr aufwendig, denn in jedem Schritt muss die QR-Zerlegung einer verschobenen Matrix $A - \mu_k I$ berechnet werden. Dies kostet $\frac{2}{3}n^3$ komplexe Operationen für eine beliebige Matrix A . Der Aufwand zur Berechnung aller Eigenwerte ist damit mindestens $O(n^4)$. Bei der allgemeinen Variante mit Mehrfachshift hat man zusätzlich das Problem, dass man das Matrixpolynom $f_k(A)$ benötigt. Wir werden jetzt zeigen, dass man durch eine Vorbehandlung der Matrix A , nämlich einer Transformation auf obere Hessenberg-Form, den Aufwand deutlich reduzieren und die explizite Berechnung von $f_k(A)$ vermeiden kann.

Eine wesentliche Rolle bei der Reduktion spielen Householder-Matrizen (vgl. Abschnitt 3.7), daher zunächst eine kurze Erinnerung: Für $u \in \mathbb{C}^n$ mit $\|u\| = 1$ heißt die Hermitesche und unitäre Matrix $P = I - 2uu^H \in \mathbb{C}^{n,n}$ Householder-Matrix. u kann so gewählt werden, dass für ein gegebenes $x \in \mathbb{C}^n$, $x \neq 0$

$$Px = \alpha e_1, \quad \alpha \in \mathbb{C}$$

gilt, wobei

$$|\alpha| = \|x\|, \quad u = \frac{x - \alpha e_1}{\|x - \alpha e_1\|}. \quad (6.9)$$

Satz 6.33. *Jede Matrix $A \in \mathbb{C}^{n,n}$ kann durch $(n-2)$ Householder-Transformationen auf Hessenberg-Form transformiert werden:*

$$U^H A U = H = \begin{bmatrix} h_{11} & \cdots & h_{1,n-1} & h_{1,n} \\ h_{21} & \ddots & \vdots & \vdots \\ & \ddots & h_{n-1,n-1} & h_{n-1,n} \\ 0 & & h_{n,n-1} & h_{n,n} \end{bmatrix}, \quad U = U_1 \cdots U_{n-2}$$

mit Householder-Matrizen U_1, \dots, U_{n-2} .

Beweis. Wir wählen die $(n-1) \times (n-1)$ Householder-Matrix $\tilde{U}_1 = I - 2u_1 u_1^H$, $\|u_1\| = 1$ so, dass

$$\tilde{U}_1 \begin{bmatrix} a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} \star \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

und setzen $U_1 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{U}_1 \end{bmatrix}$. Dann gilt

$$A_1 = U_1 A U_1^H = \left[\begin{array}{c|c} \star & \\ \star & \\ 0 & \star \\ \vdots & \\ 0 & \end{array} \right] \begin{bmatrix} 1 & 0 \\ 0 & \tilde{U}_1 \end{bmatrix} = \left[\begin{array}{c|c} \star & \\ \star & \\ 0 & \star \\ \vdots & \\ 0 & \end{array} \right].$$

Im nächsten Schritt wählen wir die $(n-2) \times (n-2)$ Householder-Matrix $\tilde{U}_2 = I - 2u_2u_2^H$, $\|u_2\| = 1$ so, dass

$$\tilde{U}_2 \begin{bmatrix} a_{32}^{(1)} \\ a_{42}^{(1)} \\ \vdots \\ a_{n2}^{(1)} \end{bmatrix} = \begin{bmatrix} \star \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

und setzen $U_2 = \begin{bmatrix} I_2 & 0 \\ 0 & \tilde{U}_2 \end{bmatrix}$. Dann gilt

$$A_2 = U_2 A_1 U_2^H = \left[\begin{array}{cc|c} \star & \star & \\ \star & \star & \\ 0 & \star & \star \\ 0 & 0 & \\ \vdots & \vdots & \\ 0 & 0 & \end{array} \right] \begin{bmatrix} I_2 & 0 \\ 0 & \tilde{U}_2 \end{bmatrix} = \left[\begin{array}{cc|c} \star & \star & \\ \star & \star & \\ 0 & \star & \star \\ 0 & 0 & \\ \vdots & \vdots & \\ 0 & 0 & \end{array} \right].$$

So fährt man nun fort, bis $U^H A U = \tilde{H}$ obere Hessenberg-Form hat. Für U erhält man nach dieser Konstruktion $U^H = U_{n-2} \cdots U_1$. \square

Ist A Hermitesch, dann gilt dasselbe auch für H . Hermitesche Matrizen werden also auf Tridiagonalform transformiert. Der Aufwand für die Transformation auf Hessenberg-Form beträgt bei einer allgemeinen Matrix $\frac{10}{3}n^3$ und bei einer Hermiteschen Matrix $\frac{4}{3}n^3$ komplexe Operationen (Übung).

Die Bedeutung dieser Transformation für den QR-Algorithmus liegt darin, dass die obere Hessenberg-Form während der gesamten Iteration erhalten bleibt.

Lemma 6.34. *Es sei H eine obere Hessenberg-Matrix und*

$$H = QR, \quad \tilde{H} = RQ.$$

Dann ist \tilde{H} ebenfalls eine obere Hessenberg-Matrix.

Beweis. Wir betrachten die Konstruktion der QR-Zerlegung von H mit Hilfe von Householder-Transformationen. Im ersten Schritt wählt man $U_1 = I - 2u_1u_1^H$ so, dass $U_1 H e_1 = \alpha_1 e_1$. Wegen (6.9) hat u_1 nur in den ersten beiden Komponenten von Null verschiedene Einträge und U_1 hat die Form

$$U_1 = \begin{bmatrix} \star & \star & & \\ \star & \star & & \\ & & I_{n-2} & \end{bmatrix}.$$

Die im nächsten Schritt konstruierte Householder-Matrix U_2 hat die Form

$$U_2 = \begin{bmatrix} 1 & & & \\ & \star & \star & \\ & \star & \star & \\ & & & I_{n-3} \end{bmatrix}.$$

Man kann sich dann leicht überlegen, dass $Q = U_1 \dots U_{n-2}$ obere Hessenberg-Form hat und dass dies auch für $\tilde{H} = RQ$ gilt. \square

Verschwindet während des Algorithmus ein Nebendiagonalelement von H_k , dann hat H_k die Form

$$H_k = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}, \quad H_{11} \in \mathbb{C}^{p,p}, \quad 1 \leq p < n.$$

Das Problem zerfällt in zwei kleinere Teilprobleme. Man spricht dann auch von **Deflation**. In der Praxis wird es allerdings selten vorkommen, dass ein Nebendiagonalelement exakt Null wird. Man betrachtet es als Null und spaltet kleinere Teilprobleme ab, wenn

$$|h_{p+1,p}| \leq c \, \mathbf{eps} \, (|h_{p,p}| + |h_{p+1,p+1}|)$$

für eine kleine Konstante $c > 0$ gilt. Dieses Vorgehen ist dadurch gerechtfertigt, dass Rundungsfehler in der Größenordnung $\mathbf{eps}\|H\|$ ohnehin schon bei der Berechnung von H_k entstanden sind.

Die optimale Wahl der Shifts beim QR-Algorithmus für Hessenberg-Matrizen wären die gesuchten Eigenwerte von H_k , denn aus Lemma 6.24 folgt, dass bei Wahl des Shiftpolynoms so, dass $f(\lambda_{m+j}) = 0$ für $j = 1, \dots, n - m$ gilt, die Matrix B_{21} , also der untere Diagonalblock des nächsten Schritts, verschwindet. Man spricht dann auch von **exakten Shifts**. Bei exakten Shifts tritt also in exakter Arithmetik Deflation nach einem Schritt auf. Neben den Rayleigh-Shifts wird vor allem der **Wilkinson-Shift** in der Praxis eingesetzt. Als Shift μ wird derjenige Eigenwert von

$$\begin{bmatrix} h_{n-1,n-1} & h_{n-1,n} \\ h_{n,n-1} & h_{n,n} \end{bmatrix}$$

gewählt, der näher an $h_{n,n}$ liegt.

Ist die Matrix A und damit auch die auf Hessenberg-Form transformierte Matrix H reell und nicht symmetrisch, dann kann A Paare konjugiert komplexer Eigenwerte haben. Es ist dann nicht mehr sinnvoll, sich auf reelle Shifts zu beschränken. Andererseits möchte man bei komplexen Shifts die komplexe Arithmetik vermeiden. Hier bieten sich Mehrfachshifts an, zum Beispiel Doppelshifts, bei denen ein konjugiert komplexes Paar von Shifts verwendet wird. Wir zeigen jetzt, dass es bei oberen Hessenberg-Matrizen möglich ist, mehrere Shifts auf einmal anzuwenden, ohne die Matrix $f_k(H_k)$ explizit berechnen zu müssen. Basis für diese Aussage ist

Satz 6.35. (*Implizites Q-Theorem*)

Es seien $Q = [q_1 \ \dots \ q_n]$ und $U = [u_1 \ \dots \ u_n]$ unitäre Matrizen für die

$$Q^H A Q = H \quad \text{und} \quad U^H A U = G$$

obere Hessenberg-Form haben. Ist $u_1 = q_1$ und H unreduziert, dann gilt $Q = U D$ mit einer unitären Diagonalmatrix D und $|h_{j,j-1}| = |g_{j,j-1}|$ für $j = 2, \dots, n$.

Beweis. Definieren wir $V = U^H Q$, dann gilt nach Voraussetzung

$$GV = GU^H Q = U^H A Q = U^H Q Q^H A Q = V H.$$

Für $V = [v_1 \ \dots \ v_n]$ folgt daraus

$$Gv_j = \sum_{i=1}^{j+1} h_{i,j} v_i \quad \text{oder} \quad h_{j+1,j} v_{j+1} = Gv_j - \sum_{i=1}^j h_{i,j} v_i.$$

Wegen $v_1 = e_1$ und der oberen Hessenberg-Form von G müssen die Einträge $j+1$ bis n in v_j verschwinden, d. h. V hat obere Dreiecksform. Da V als Produkt unitärer Matrizen auch unitär ist, ist $V = D$ diagonal. \square

Die Bedeutung des impliziten Q-Theorems liegt darin, dass man, um $A_{k+1} = Q_k^H A_k Q_k$ aus A_k im QR-Algorithmus zu berechnen, lediglich

- (a) die erste Spalte von Q_k (die der normierten ersten Spalte von $A_k - \mu_k I$ entspricht) berechnen muss und
- (b) die übrigen Spalten von Q_k so wählt, dass Q_k unitär ist und A_{k+1} eine unreduzierte obere Hessenberg-Matrix ist.

Wir illustrieren das implizite Q-Theorem an einem kleinen Beispiel.

Beispiel. Sei $A_k \in \mathbb{C}^{5,5}$ die obere Hessenberg-Matrix, die durch k Schritte des QR-Algorithmus berechnet wurde.

- (a) Wähle $P_1^H = \begin{bmatrix} \tilde{P}_1 & \\ & I_3 \end{bmatrix}$ mit einer Householder-Matrix $\tilde{P}_1 \in \mathbb{C}^{2,2}$, für die

$$P_1 e_1 = \frac{(A_k - \mu_k I) e_1}{\|(A_k - \mu_k I) e_1\|} \quad \text{und} \quad B_1 = P_1^H A_k P_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \otimes & \times & \times & \times & \times \\ \oplus & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

Es entsteht an der mit \oplus gekennzeichneten Position eine Ausbuchtung, die eliminiert werden muss, um die obere Hessenberg-Form wiederherzustellen. Die Idee ist, die Ausbuchtung (*bulge*) solange nach unten zu "jagen" bis sie "herausfällt". Dieser Prozess wird in der Literatur als *bulge chasing* bezeichnet.

- (b) Wähle $P_2^H = \begin{bmatrix} 1 & & \\ & \tilde{P}_2 & \\ & & I_2 \end{bmatrix}$ mit einer unitären Matrix $\tilde{P}_2 \in \mathbb{C}^{2,2}$, die den durch \oplus und \otimes gekennzeichneten Vektor auf ein Vielfaches des ersten Einheitsvektors abbildet, also

$$P_2^H B_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \quad \text{und} \quad B_2 = P_2^H B_1 P_2 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \otimes & \times & \times & \times \\ 0 & \oplus & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

gilt. Die Ausbuchtung hat sich von Position (3, 1) auf Position (4, 2) verlagert.

(c) Wähle $P_3^H = \begin{bmatrix} I_2 & & \\ & \tilde{P}_3 & \\ & & 1 \end{bmatrix}$ mit einer unitären Matrix $\tilde{P}_3 \in \mathbb{C}^{2,2}$, für die

$$P_3^H B_2 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \quad \text{und} \quad B_3 = P_3^H B_2 P_3 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \otimes & \times & \times \\ 0 & 0 & \oplus & \times & \times \end{bmatrix}$$

gilt. Die Ausbuchtung wurde hier von Position (4, 2) auf (5, 3) gejagt.

(d) Wähle $P_4^H = \begin{bmatrix} I_3 & & \\ & \tilde{P}_4 & \\ & & 1 \end{bmatrix}$ mit einer unitären Matrix $\tilde{P}_4 \in \mathbb{C}^{2,2}$, so dass

$$P_4^H B_3 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \quad \text{und} \quad B_4 = P_4^H B_3 P_4 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

gilt. Die Matrix B_4 hat dann wieder obere Hessenberg-Form.

Es hat also $P^H A_k P$ obere Hessenberg-Form, wobei

$$P = P_1 P_2 P_3 P_4, \quad P e_1 = P_1 e_1 = \frac{(A_k - \mu_k I) e_1}{\|(A_k - \mu_k I) e_1\|}.$$

Nach dem impliziten Q-Theorem ist P bis auf eine unitäre Diagonalmatrix D dieselbe Matrix, wie die der QR-Zerlegung von $A_k - \mu_k I$, d. h. $Q_k = PD$ und $A_{k+1} = Q_k^H A_k Q_k$. \diamond

Als nächstes betrachten wir die implizite Anwendung eines Doppelshifts. Der QR-Algorithmus 6.7 berechnet

$$f_k(A_k) = Q_k R_k, \quad A_{k+1} = Q_k^H A_k Q_k,$$

wobei $f_k(\lambda) = (\lambda - \mu_1^{(k)})(\lambda - \mu_2^{(k)})$.

Wir möchten nun Q_k berechnen, ohne vorher explizit $f_k(A_k)$ berechnet zu haben. Nach dem impliziten Q-Theorem müssen wir mit der ersten Spalte beginnen. Nach Definition der QR-Zerlegung gilt

$$Q_k e_1 = \frac{1}{\|f_k(A_k) e_1\|} f_k(A_k) e_1 = \begin{bmatrix} \star \\ \star \\ \star \\ 0_{n-3} \end{bmatrix}.$$

Diesen Vektor transformiert man nun mit einer Householder-Matrix $P_1 = \begin{bmatrix} \tilde{P}_1 & \\ & I_{n-3} \end{bmatrix}$ mit $\tilde{P}_1 \in \mathbb{C}^{3,3}$ auf ein Vielfaches des ersten Einheitsvektors. In der Matrix $P_1 A_k P_1^H$ entstehen dadurch Nichtnullelemente an den Positionen (3, 1), (4, 1) und (4, 2), eine Ausbuchtung der Dimension 2×2 . Diese Ausbuchtung jagt man ähnlich wie bei einfachen Shifts solange, bis sie unten aus der Matrix “fällt”. Hierfür benötigt man in jedem Schritt unitäre Matrizen \tilde{P}_j

der Dimension 3×3 , denn $f_k(A_k)e_1$ hat drei Nichtnullelemente. Die Details besprechen wir in den Übungen.

Zur Berechnung der Eigenvektoren aus dem QR-Algorithmus gibt es zum Beispiel folgende Möglichkeiten:

- (a) Akkumulation aller unitärer Transformationen: $Q = Q_0 Q_1 Q_2 \dots$. Im Grenzfall ist dann $R = Q^H A Q$ eine obere Dreiecksmatrix. Die Eigenvektoren von A berechnen wir dann aus den Eigenvektoren von R . Ist x ein Eigenvektor von R zum Eigenwert $\lambda = r_{ii}$, dann kann man x durch

$$x_i = 1, \quad x_j = 0 \quad \text{für} \quad j > i$$

und x_1, \dots, x_{i-1} als Lösung des Dreieckssystems

$$(R_{1:i-1,1:i-1} - r_{ii}I_{i-1})x_{1:i-1} = -R_{1:i-1,i}$$

effizient berechnen. Diese Vorgehensweise ist jedoch teuer, da man die Matrix Q explizit benötigt.

- (b) Inverse Potenzenmethode mit Shift (inverse Iteration) mit der auf Hessenberg-Form transformierten Matrix $A_0 = H = U^H A U$. Die in jedem Schritt der inversen Iteration auftretenden linearen Gleichungssysteme $H - \lambda I$ löst man mit Hilfe der QR-Zerlegung unter Ausnutzung der Struktur mit einem Aufwand von $O(n^2)$. Die Eigenvektoren von A kann man wieder aus denen von H mit Hilfe der Transformationsmatrix U berechnen: Ist x ein Eigenvektor von H , dann ist Ux ein Eigenvektor von A . Der Vorteil ist hierbei, dass man U nicht explizit benötigt, sondern $Ux = U_1 \dots U_{n-2}x$ durch sukzessive Multiplikation eines Vektors mit den $(n-2)$ Householder-Matrizen U_j berechnen kann. Dies kostet nur jeweils 1 SAXPY und ein Skalarprodukt.

Man kann zeigen, dass der QR-Algorithmus stabil im Sinne der Rückwärtsanalyse ist: Ist $R = \hat{Q}^H A \hat{Q}$ die berechnete Schurform, dann ist $\hat{R} = Q^H \hat{A} Q$ mit $Q^H Q = I$ die Schurform einer gestörten Matrix mit $\|A - \hat{A}\| \leq c \text{eps} \|A\|$. Ferner ist die berechnete Matrix \hat{Q} fast unitär, $\hat{Q}^H \hat{Q} = I + E$ mit $\|E\| \leq c \text{eps}$.

6.7 QR-Algorithmus zur Berechnung der Singulärwertzerlegung

Wir wollen jetzt zeigen, wie man die Singulärwertzerlegung

$$U^H A V = \Sigma = \begin{bmatrix} \tilde{\Sigma} \\ 0 \end{bmatrix} \in \mathbb{R}^{m,n}, \quad \tilde{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_n) \in \mathbb{R}^{n,n}$$

einer Matrix $A \in \mathbb{C}^{m,n}$ mit einer Variante des QR-Algorithmus berechnen kann. Nach Satz 3.33 erscheint das folgende Vorgehen naheliegend:

- (a) Berechne $C = A^H A$.
- (b) Transformiere C auf Tridiagonalform und wende darauf den symmetrischen QR-Algorithmus an, um $V^H C V = \tilde{\Sigma}^2 = \text{diag}(\sigma_j^2)$ zu erhalten.
- (c) Berechne die QR-Zerlegung von AV : $U^H(AV) = R$. Da AV wegen (b) orthogonale Spalten hat, muss R diagonal sein, also $R = \Sigma$.

Die explizite Berechnung von C ist nicht nur teuer, sondern sie kann auch zu Stabilitätsproblemen führen. Algorithmen, die auf C basieren, werden Fehler mit $\kappa(A)^2$ statt mit $\kappa(A)$ verstärken. Die Idee von Golub und Kahan (1965) war, U und V simultan durch implizite Anwendung des QR-Algorithmus auf $A^H A$ zu berechnen (Golub & van Loan 1996). Hierbei nutzt man aus, dass für beliebige unitäre Matrizen P und Q , A und PAQ dieselben Singulärwerte haben, denn

$$A = U\Sigma V^H \iff PAQ = (PU)\Sigma(V^H Q).$$

Der erste Schritt ist wieder eine Vorbehandlung, in diesem Fall eine Transformation von A auf obere Bidiagonalform:

Lemma 6.36. Zu $A \in \mathbb{C}^{m,n}$ mit $m \geq n$ existieren unitäre Matrizen P, Q mit

$$PAQ = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad B = \begin{bmatrix} d_1 & f_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & f_{n-1} \\ 0 & \cdots & \cdots & 0 & d_n \end{bmatrix}.$$

Beweis. Die Transformation erfolgt mittels geeigneter Householdertransformationen:

$$A \xrightarrow{P_1} \left[\begin{array}{c|c} \begin{bmatrix} \star \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \star \end{array} \right] \xrightarrow{\cdot Q_1} \left[\begin{array}{c|c} \begin{bmatrix} \star & \star & 0 & \cdots & 0 \\ 0 & \star & \star & 0 & \cdots & 0 \\ \vdots & 0 & \star & \star & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} & \star \end{array} \right] \xrightarrow{P_2 \cdot Q_2} \left[\begin{array}{c|c} \begin{bmatrix} \star & \star & 0 & \cdots & \cdots & 0 \\ 0 & \star & \star & 0 & \cdots & 0 \\ \vdots & 0 & \star & \star & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} & \star \end{array} \right]$$

usw. bis schließlich obere Bidiagonalform erreicht ist. \square

Algorithmus 6.8 Bidiagonalisierung von $A \in \mathbb{C}^{m,n}$

```
{ function u = house(x) liefere u mit  $(I - 2uu^H)x = \alpha e_1$  gemäß (6.9)}
for j = 1 : n do
    u = house(A(j : m, j))
    A(j : m, j : n) = (Im-j+1 - 2uuH)A(j : m, j : n)
    if j ≤ n - 2 then
        u = house(A(j, j + 1 : n)T)
        A(j : m, j + 1 : n) = A(j : m, j + 1 : n)(In-j - 2uuH)
    end if
end for
```

Damit haben wir das Problem auf die Berechnung der Singulärwerte von B bzw. der Eigenwerte von $B^H B =: T$ tridiagonal reduziert.

Tatsächlich kann man den QR-Algorithmus durchführen, ohne T explizit zu berechnen, denn nach dem impliziten Q-Theorem genügt es, die erste Spalte von $T - \mu I$ (zum Beispiel mit Rayleigh-Shift μ) zu kennen:

$$t = (T - \mu I)e_1 = (B^H B - \mu I)e_1 = \begin{bmatrix} |d_1|^2 - \mu \\ d_1 \overline{f_1} \\ 0_{n-2} \end{bmatrix}.$$

Da T tridiagonal ist, kann t mit einer Householdermatrix der Form

$$Q_1 = \begin{bmatrix} \times & \times & & \\ \times & \times & & \\ & & I_{n-2} \end{bmatrix}$$

auf ein Vielfaches von e_1 abgebildet werden. Anschließend müssen wir T auf Tridiagonalform $Q^H T Q$, $Q = Q_1 \tilde{Q}$ transformieren. Nun ist aber $Q^H T Q = Q^H B^H P^H P B Q$ für jedes unitäre P . Die Matrix $Q^H T Q$ ist tridiagonal, wenn $P B Q$ bidiagonal ist, also transformieren wir nicht $Q_1^H T Q_1$ sondern $B Q_1$:

$$\begin{array}{ccc} B & \xrightarrow{\cdot Q_1} & \begin{bmatrix} \otimes & \times & & & \\ \oplus & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times & \times \\ & & & & & \times \end{bmatrix} & \xrightarrow{P_1} & \begin{bmatrix} \times & \otimes & \oplus & & \\ & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times & \times \\ & & & & & \times \end{bmatrix} \\ & \xrightarrow{\cdot Q_2} & \begin{bmatrix} \times & \times & & & \\ & \otimes & \times & & \\ & \oplus & \times & \times & \\ & & & \times & \times \\ & & & & \times & \times \\ & & & & & \times \end{bmatrix} & \xrightarrow{P_2} & \begin{bmatrix} \times & \times & & & \\ & \times & \otimes & \oplus & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times & \times \\ & & & & & \times \end{bmatrix} \end{array}$$

Diese Zickzackjagd nach den Nichtnullelementen führen wir fort, bis die Matrix wieder Bidiagonalform hat. Details sind in Algorithmus 6.9 angegeben.

Algorithmus 6.9 Golub-Kahan Schritt zur Singulärwertzerlegung

Gegeben sei eine Bidiagonalmatrix $A \in \mathbb{C}^{m,n}$

Zu gegebenem Shift μ berechne erste Spalte von $A^H A - \mu I$: $t = \begin{bmatrix} |a_{11}|^2 - \mu \\ a_{11} \overline{a_{12}} \end{bmatrix}$.

for $j = 1 : n - 1$ **do**

$u = \text{house}(t)$

$\tilde{j} = \max\{1, j - 1\}$;

$A(\tilde{j} : j + 1, j : j + 1) = A(\tilde{j} : j + 1, j : j + 1)(I - 2uu^H)$

$u = \text{house}(A(j : j + 1, j))$;

$\tilde{j} = \min\{j + 2, n\}$;

$A(j : j + 1, j : \tilde{j}) = (I - 2uu^H)A(j : j + 1, j : \tilde{j})$;

if $j < n - 1$ **then**

$t = A(j, j + 1 : j + 2)^T$;

end if

end for

Da ein Schritt dieses Verfahrens äquivalent zu einem Schritt des QR-Verfahrens angewandt auf die Hermitesche Matrix $B^H B$ ist, ist die Konvergenz lokal kubisch. Typischerweise wird also nach wenigen QR-Schritten ein neues Nebendiagonalelement f_p klein. Falls

$$|f_p| \leq \text{eps} \|B\|_F \quad \text{oder} \quad |f_p| \leq \text{eps} (|d_p| + |d_{p+1}|),$$

$$\begin{aligned}
B &= \begin{bmatrix} \times & \times & & & & \\ & \times & \times & & & \\ & & 0 & \otimes & & \\ & & & \otimes & \times & \\ & & & & \times & \times \\ & & & & & \times \end{bmatrix}, & P_1 &= \begin{bmatrix} I_2 & & & \\ & \times & \times & \\ & \times & \times & \\ & & & I_2 \end{bmatrix}, \\
&\xrightarrow{P_1} \begin{bmatrix} \times & \times & & & & \\ & \times & \times & & & \\ & & 0 & 0 & \oplus & \\ & & & \times & \times & \\ & & & & \otimes & \times \\ & & & & & \times \end{bmatrix}, & P_2 &= \begin{bmatrix} I_2 & & & \\ & \times & & \times \\ & & 1 & \\ & \times & & \times \\ & & & 1 \end{bmatrix}, \\
&\xrightarrow{P_2} \begin{bmatrix} \times & \times & & & & \\ & \times & \times & & & \\ & & 0 & 0 & 0 & \oplus \\ & & & \times & \times & \\ & & & & \times & \times \\ & & & & & \otimes \end{bmatrix}, & P_3 &= \begin{bmatrix} I_2 & & & \\ & \times & & \times \\ & & I_2 & \\ & \times & & \times \end{bmatrix}, \\
&\xrightarrow{P_3} \begin{bmatrix} \times & \times & & & & \\ & \times & \times & & & \\ & & 0 & 0 & 0 & 0 \\ & & & \times & \times & \\ & & & & \times & \times \\ & & & & & \times \end{bmatrix}
\end{aligned}$$

Abb. 6.2: Elimination eines Nebendiagonalelements, falls ein Diagonalelement null ist

vernachlässigt man $|f_p|$ und arbeitet mit den beiden unabhängigen Teilproblemen weiter. Des Weiteren verschwindet ein Nebendiagonalelement von $B^H B$, wenn $d_p = 0$ ist für ein $p < n$. Hier kann das zugehörige Nebendiagonalelement f_p durch eine Folge von $(n - p)$ Householdertransformationen (von links) ebenfalls eliminiert werden. Für $p = n$ kann man die n -te Spalte durch eine Folge von $n - 1$ Householdertransformationen (von rechts) eliminieren. Wir illustrieren die Vorgehensweise anhand von Beispielen der Dimension 6. Das Verfahren bei beliebiger Dimension leitet man daraus leicht ab.

Beispiel. Die Elimination eines Nebendiagonalelements, falls ein Diagonalelement null ist, erfolgt wie in Abbildung 6.2 dargestellt. Die durch \oplus gekennzeichneten Elemente sind durch die Transformation neu entstanden und müssen eliminiert werden. Hierfür verwendet man die angegebenen Householder-Transformationen, die die durch \oplus und \otimes gekennzeichneten Vektoren der Länge zwei auf ein Vielfaches des zweiten Einheitsvektors e_2 abbilden. Durch Anwendung dieser speziellen Householder-Matrizen P_j ändern sich nur die Einträge in zwei Zeilen der Matrix. \diamond

Beispiel. Die Elimination der letzten Spalte von B wenn die letzte Zeile null ist, ist in Abbildung 6.3 dargestellt. Die Notation ist wie im Beispiel oben, wobei hier jedoch die Householdertransformationen die durch \otimes und \oplus gekennzeichneten Vektoren der Länge zwei auf ein Vielfaches des ersten Einheitsvektors abbilden. Auch hier wirken sich die Householder-

Transformationen nur auf die Einträge zweier Spalten aus. \diamond

Der vollständige Algorithmus zur Berechnung der Singulärwerte einer Matrix ist in Algorithmus 6.10 zusammengefasst.

Algorithmus 6.10 Singulärwertzerlegung von $A \in \mathbb{C}^{m,n}$

Transformiere A mit Algorithmus 6.8 auf Bidiagonalform $\begin{bmatrix} B \\ 0 \end{bmatrix}$, setze $q = 0$.

while $q < n$ **do**

Setze $b_{i,i+1} = 0$, falls $|b_{i,i+1}| \leq \text{eps} (|b_{ii}| + |b_{i+1,i+1}|)$, $i = 1, \dots, n-1$.

Bestimme das kleinste p und das größte q , so dass für

$$B = \begin{bmatrix} B_{11} & 0 & 0 \\ 0 & B_{22} & 0 \\ 0 & 0 & B_{33} \end{bmatrix}, \quad B_{11} \in \mathbb{C}^{p,p}, B_{22} \in \mathbb{C}^{n-p-q, n-p-q}, B_{33} \in \mathbb{C}^{q,q}$$

B_{33} diagonal ist und B_{22} nicht reduziert ist.

if $q < n$ **then**

if ein Diagonalelement in B_{22} verschwindet **then**

if das Diagonalelement ist nicht in der letzten Zeile von B_{22} **then**

Eliminiere mit Householdertransformationen die gesamte Zeile von B_{22}

else

Eliminiere mit Householdertransformationen die letzte Spalte von B_{22}

end if

else

Wende Algorithmus 6.9 auf B_{22} an

end if

end if

end while

6.8 Trägheit einer Matrix, Bisektionsverfahren

Definition 6.37. Für $A \in \mathbb{C}^{n,n}$ und eine nicht singuläre Matrix $T \in \mathbb{C}^{n,n}$ heißt die Abbildung $A \mapsto T^H A T$ eine **Kongruenztransformation**. A und $T^H A T$ werden kongruent genannt.

Bemerkung. Außer für T unitär erhalten Kongruenztransformationen im Allgemeinen nicht die Eigenwerte. Sylvesters berühmter Trägheitssatz besagt jedoch, dass die Vorzeichen der Eigenwerte einer Hermiteschen Matrix erhalten werden.

Definition 6.38. Für $A \in \mathbb{C}^{n,n}$ Hermitesch heißt das Triple $\text{in}(A) = (\pi, \nu, \delta)$ der Anzahl der positiven, negativen und Eigenwerten gleich Null von A die **Trägheit** (inertia) von A .

Satz 6.39. (Sylvesters Trägheitssatz)

Es sei A Hermitesch und T nicht singulär. Dann haben A und $T^H A T$ dieselbe Trägheit.

Beweis. Wie definieren $\hat{A} = T^H A T$. Dann existieren U, \hat{U} unitär, so dass

$$U^H A U = D, \quad \hat{U}^H \hat{A} \hat{U} = \hat{D},$$

$$\begin{aligned}
 B &= \begin{bmatrix} \times & \times & & & & \\ & \times & \times & & & \\ & & \times & \times & & \\ & & & \times & \times & \\ & & & & \times & \otimes \\ & & & & & \otimes \\ & & & & & 0 \end{bmatrix}, & Q_1 &= \begin{bmatrix} I_4 & & \\ & \times & \times \\ & \times & \times \end{bmatrix}, \\
 \xrightarrow{\cdot Q_1} & \begin{bmatrix} \times & \times & & & & \\ & \times & \times & & & \\ & & \times & \times & & \\ & & & \otimes & \times & \oplus \\ & & & & \times & 0 \\ & & & & & 0 \end{bmatrix}, & Q_2 &= \begin{bmatrix} I_3 & & \\ & \times & \times \\ & & 1 \\ & \times & \times \end{bmatrix}, \\
 \xrightarrow{\cdot Q_2} & \begin{bmatrix} \times & \times & & & & \\ & \times & \times & & & \\ & & \otimes & \times & & \oplus \\ & & & \times & \times & 0 \\ & & & & \times & 0 \\ & & & & & 0 \end{bmatrix}, & Q_3 &= \begin{bmatrix} I_2 & & \\ & \times & \times \\ & & I_2 \\ & \times & \times \end{bmatrix}, \\
 \xrightarrow{\cdot Q_3} & \begin{bmatrix} \times & \times & & & & \\ & \otimes & \times & & & \oplus \\ & & \times & \times & & 0 \\ & & & \times & \times & 0 \\ & & & & \times & 0 \\ & & & & & 0 \end{bmatrix}, & Q_4 &= \begin{bmatrix} 1 & & \\ & \times & \times \\ & & I_3 \\ & \times & \times \end{bmatrix}, \\
 \xrightarrow{\cdot Q_4} & \begin{bmatrix} \otimes & \times & & & & \oplus \\ & \times & \times & & & 0 \\ & & \times & \times & & 0 \\ & & & \times & \times & 0 \\ & & & & \times & 0 \\ & & & & & 0 \end{bmatrix}, & Q_5 &= \begin{bmatrix} \times & & \times \\ & I_4 & \\ \times & & \times \end{bmatrix}, \\
 \xrightarrow{\cdot Q_5} & \begin{bmatrix} \times & \times & & & & 0 \\ & \times & \times & & & 0 \\ & & \times & \times & & 0 \\ & & & \times & \times & 0 \\ & & & & \times & 0 \\ & & & & & 0 \end{bmatrix}
 \end{aligned}$$

Abb. 6.3: Elimination der letzten Spalte von B , wenn die letzte Zeile Null ist.

wobei $D = \text{diag}(\lambda_1, \dots, \lambda_n)$, $\lambda_j \in \lambda(A)$, $\hat{D} = \text{diag}(\hat{\lambda}_1, \dots, \hat{\lambda}_n)$, $\hat{\lambda}_j \in \lambda(\hat{A})$. Nach Definition ist $\text{in}(A) = \text{in}(D)$, $\text{in}(\hat{A}) = \text{in}(\hat{D})$; wir müssen also zeigen, dass $\text{in}(D) = \text{in}(\hat{D})$ wobei

$$\hat{D} = \hat{U}^H \hat{A} \hat{U} = \hat{U}^H T^H A T \hat{U} = S^H D S$$

mit $S = U^H T \hat{U}$.

Der Beweis erfolgt durch Widerspruch. Nehmen wir an, dass $\pi \neq \hat{\pi}$, sagen wir $\pi > \hat{\pi}$. Für $x = S\hat{x}$ betrachten wir die quadratische Form $\psi(x) = x^H D x = \hat{x}^H \hat{D} \hat{x}$ oder

$$\psi(x) = \sum_{j=1}^n \lambda_j |x_j|^2 = \sum_{j=1}^n \hat{\lambda}_j |\hat{x}_j|^2,$$

wobei $\lambda_j > 0$ für $j \leq \pi$ und $\hat{\lambda}_j > 0$ für $j \leq \hat{\pi}$. Sei $y \neq 0$ eine Lösung der $n - \pi + \hat{\pi}$ ($< n$) homogenen linearen Gleichungen

$$y_j = 0, \quad j > \pi, \quad \hat{y}_j = (S^{-1}y)_j = 0, \quad j \leq \hat{\pi}.$$

Es gilt dann

$$\psi(y) = \sum_{j=1}^{\pi} \lambda_j |y_j|^2 > 0, \quad \psi(y) = \sum_{j=\hat{\pi}+1}^n \hat{\lambda}_j |\hat{y}_j|^2 \leq 0$$

im Widerspruch zur Annahme. Die Annahme $\pi \neq \hat{\pi}$ ist daher falsch, A und \hat{A} haben also dieselbe Anzahl positiver Eigenwerte. Mit demselben Argument für $-A$ zeigt man, dass $\nu = \hat{\nu}$ und dann muss auch $\delta = \hat{\delta}$ gelten. \square

Bemerkung. Sei $A \in \mathbb{C}^{n,n}$ Hermitesch und $\sigma \in \mathbb{R}$. Angenommen, dass die symmetrische Gauß-Elimination ohne Pivotsuche für $A - \sigma I$ durchgeführt werden kann, vgl. Algorithmus 6.11 für $\gamma_j = \overline{\beta_{j+1}}$ und

$$A - \sigma I = LDL^H, \quad D = \text{diag}(d_1, \dots, d_n) \quad (6.10)$$

liefert, wobei L eine untere Diagonalmatrix mit $\text{diag}(L) = I$ ist. Dann besagt der Trägheitssatz von Sylvester, dass $\text{in}(D) = \text{in}(A - \sigma I)$. Die Anzahl der Eigenwerte von A größer als σ ist gerade die Anzahl der positiven Elemente $\pi(D)$ in der Folge d_1, \dots, d_n .

Beispiel. Die LDL^T -Zerlegung

$$A - 1 \cdot I = \begin{bmatrix} 1 & 2 & \\ 2 & 2 & -4 \\ & -4 & -6 \end{bmatrix} = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & -2 & \\ & & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & \\ & 1 & 2 \\ & & 1 \end{bmatrix}$$

zeigt, dass A genau zwei Eigenwerte größer als eins hat. \diamond

Ist $A - \sigma I$ indefinit, dann ist es möglich, dass die LDL^H -Zerlegung nicht existiert (z. B. für den Shift $\sigma = 2$ in vorangegangenen Beispiel). In diesem Fall, kann man eine symmetrische Pivotsuche durchführen, die zu einer Blockdiagonalmatrix D mit 1×1 und 2×2 Blöcken führt. Jeder 2×2 Block gehört zu einem positiven und einem negativen Eigenwert. Die Trägheit von D kann daher auch in diesem Fall einfach bestimmt werden.

Die Zerlegung (6.10) kostet $n^3/6$ Operationen für eine voll besetzte Matrix. Im Spezialfall, dass A eine Hermitesche Tridiagonalmatrix ist, ist das Verfahren besonders effizient ($O(n)$ Operationen) und stabil.

Für eine allgemeine Tridiagonalmatrix

$$T = \begin{bmatrix} \alpha_1 & \gamma_1 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{n-1} \\ & & \beta_n & \alpha_n \end{bmatrix} \in \mathbb{C}^{n,n} \quad (6.11)$$

liefert Algorithmus 6.11 eine LDU-Zerlegung

$$T = LDU = \begin{bmatrix} 1 & & & \\ l_2 & 1 & & \\ & \ddots & \ddots & \\ & & l_n & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{bmatrix} 1 & u_1 & & \\ & 1 & \ddots & \\ & & \ddots & u_{n-1} \\ & & & 1 \end{bmatrix}. \quad (6.12)$$

Für Hermitesche Matrizen T gilt dann $U = L^H$. Die Anzahl $\pi = \pi(T - \sigma I)$ von Eigenwerten von T größer als ein gegebenes $\sigma > 0$ kann unter Verwendung von Algorithmus 6.11 und wegen $\gamma_j = \overline{\beta_{j+1}}$ mit Algorithmus 6.12 berechnet werden.

Algorithmus 6.11 LDU-Zerlegung einer Tridiagonalmatrix (6.11)

```

 $d_1 = \alpha_1$ 
for  $j = 2 : n$  do
     $u_{j-1} = \gamma_{j-1} / d_{j-1}$ 
     $l_j = \beta_j / d_{j-1}$ 
     $d_j = \alpha_j - l_j \gamma_{j-1}$ 
end for
    
```

Algorithmus 6.12 Tridiagonales Spektrum Slicing

```

 $d_1 = \alpha_1 - \sigma$ 
if  $|d_1| < \sqrt{\text{eps}}$  then  $d_1 = \sqrt{\text{eps}}$  /* vermeide Underflow */
if  $d_1 > 0$  then  $\pi = 1$  else  $\pi = 0$ 
for  $j = 2 : n$ 
     $d_j = \alpha_j - \frac{|\beta_j|^2}{d_{j-1}} - \sigma$ 
    if  $|d_j| < \sqrt{\text{eps}}$  then  $d_j = \sqrt{\text{eps}}$  /* vermeide Underflow */
    if  $d_j > 0$  then  $\pi = \pi + 1$ 
end for
    
```

Lemma 6.40. Die Werte d_j , die in Gleitkommaarithmetik mit Algorithmus 6.12 berechnet werden, haben dieselben Vorzeichen wie die Werte \hat{d}_j , die in exakter Arithmetik ausgehend von einer Tridiagonalmatrix \hat{A} berechnet werden, wobei

$$\hat{\alpha}_j = \alpha_j, \quad |\hat{\beta}_j| = |\beta_j|(1 + \epsilon_j), \quad |\epsilon_j| \leq 2.5\text{eps} + O(\text{eps}^2)$$

Beweis. Die berechneten Werte \tilde{d}_j erfüllen

$$\tilde{d}_j = \left[(\alpha_j - \sigma)(1 + \epsilon_{1,j}) - \frac{|\beta_j|^2(1 + \epsilon_{2,j})}{\tilde{d}_{j-1}}(1 + \epsilon_{3,j}) \right] (1 + \epsilon_{4,j}), \quad (6.13)$$

wobei $|\epsilon_{i,j}| \leq \mathbf{eps}$. Wir definieren neue Variablen

$$\hat{d}_j = \frac{\tilde{d}_j}{(1 + \epsilon_{1,j})(1 + \epsilon_{4,j})} \quad (6.14)$$

$$|\hat{\beta}_j| = |\beta_j| \left(\frac{(1 + \epsilon_{2,j})(1 + \epsilon_{3,j})}{(1 + \epsilon_{1,j})(1 + \epsilon_{1,j-1})(1 + \epsilon_{4,j-1})} \right)^{1/2} = |\beta_j|(1 + \epsilon_j). \quad (6.15)$$

Es gilt $\text{sign}(\hat{d}_j) = \text{sign}(\tilde{d}_j)$ und $|\epsilon_j| \leq 2.5\mathbf{eps} + O(\mathbf{eps}^2)$. Setzt man nun (6.14) und (6.15) in (6.13) ein, so ergibt sich

$$\hat{d}_j = \alpha_j - \sigma - \frac{|\hat{\beta}_j|^2(1 + \epsilon_{1,j-1})(1 + \epsilon_{4,j-1})}{\tilde{d}_{j-1}} = \alpha_j - \sigma - \frac{|\hat{\beta}_j|^2}{\hat{d}_{j-1}}.$$

Dies beweist die Aussage. \square

Algorithmus 6.12 also stabil im Sinne der Rückwärtsanalysis. Der Aufwand beträgt $2n$ Operationen.

Die sogenannte Spektrum Slicing Technik kann angewandt werden, um einzelne Eigenwerte λ_k von T zu bestimmen. Angenommen, wir kennen $\alpha < \beta$ so, dass mindestens k Eigenwerte rechts von α und höchstens $k - 1$ Eigenwerte echt rechts von β liegen, d. h.

$$\pi(T - \alpha I) \geq k, \quad \pi(T - \beta I) < k$$

gilt. Dann gibt es einen Eigenwert im Intervall $[\alpha, \beta]$. Eine Startnäherung für α, β kann man zum Beispiel aus dem Satz von Gershgorin berechnen. Anschließend können alle Eigenwerte von T im Intervall $[\alpha, \beta]$ bis auf eine gegebene Toleranz `tol` mit Hilfe der Algorithmen 6.12 und 6.13 bestimmt werden.

Eine Berechnung von $\pi(A - \sigma I)$ mit Algorithmus 6.12 kostet $2n$ Operationen, die Gesamtkosten für k Eigenwerte betragen $O(kn)$ Operationen.

Algorithmus 6.13 Bisektionsverfahren: Alle Eigenwerte in $[\alpha, \beta)$, $\alpha < \beta$

```

 $\pi_\alpha = \pi(T - \alpha I)$           /* Anzahl der Eigenwerte  $> \alpha$  */
 $\pi_\beta = \pi(T - \beta I)$        /* Anzahl der Eigenwerte  $> \beta$  */
if  $\pi_\alpha = \pi_\beta$ , stop      /* kein Eigenwert in  $[\alpha, \beta)$  */
schreibe  $[\alpha, \pi_\alpha, \beta, \pi_\beta]$  in die Arbeitsliste
while Arbeitsliste ist nicht leer
    nehme  $[low, \pi_{low}, up, \pi_{up}]$  aus der Arbeitsliste und entferne es daraus
    if  $up - low < \text{tol}$  then
        Ausgabe:  $\pi_{low} - \pi_{up}$  Eigenwerte in  $[low, up)$ 
    else
         $mid = (low + up)/2$ 
         $\pi_{mid} = \pi(T - mid \cdot I)$ 
        if  $\pi_{mid} < \pi_{low}$  then      /* es gibt Eigenwerte in  $[low, mid)$  */
            schreibe  $[low, \pi_{low}, mid, \pi_{mid}]$  in die Arbeitsliste
        end if
        if  $\pi_{up} < \pi_{mid}$  then      /* es gibt Eigenwerte in  $[mid, up)$  */
            schreibe  $[mid, \pi_{mid}, up, \pi_{up}]$  in die Arbeitsliste
        end if
    end if
end while

```
