

Numerik des Matrizen-Eigenwertproblems
WS 07/08

Prof. Dr. P. Spellucci

Revision 29.1.2008

Dieses Skriptum stellt den Inhalt der Vorlesung in einer sehr knappen, sicher nicht buchreifen Form dar. Es soll nicht das Studium der einschlägigen Lehrbücher ersetzen. Für Hinweise auf Fehler, unklare Formulierungen, wünschenswerte Ergänzungen etc. bin ich jederzeit dankbar. Man bedenke jedoch den Zeitrahmen der Veranstaltung, der lediglich 14 Doppelstunden umfasst, weshalb der eine oder andere Punkt wohl etwas zu kurz kommt oder auch einmal ganz wegfallen muss. Abschnitte, die im Kleindruck erscheinen, insbesondere eher technische Beweise, werden in der Vorlesung nicht vorgetragen. Sie sind aber für einen interessierten Leser zur Arbeitsvereinfachung hier aufgenommen worden. Diese Abschnitte sind durch eine Sequenz aus << und >> eingeklammert, um die Orientierung zu erleichtern. Viele der in diesem Skript beschriebenen Verfahren können mit unserem interaktiven System NUMAWWW

<http://numawww.mathematik.tu-darmstadt.de:8081>

erprobt werden, ohne dabei selbst Programme erstellen zu müssen. Ebenso steht den Studierenden auf dem CIP-Pool MATLAB in der Version R12.1 zur Verfügung, das einige dieser Verfahren als fest implementierte Funktionen zur Verfügung stellt.

Weiterführende Literatur:

1. Bai, Z.; Demmel, J.; Dongarra, J.; Ruhe, A.; van der Vorst, H.: *Templates for the solution of algebraic eigenvalue problems* SIAM 2000 .
2. Golub, G.H.; van Loan, Ch.: *Matrix Computations*. 3rd ed. Johns Hopkins Univ. Press 1996
3. Householder, A.S.: *The Theory of Matrices in Numerical Analysis*. 2nd ed. Dover Books on Elementary and Intermediate Mathematics. New York: Dover Publications, Inc. XI, 257 p. (1975).
4. Parlett, N.P.: *The Symmetric Eigenvalue Problem*. Unabridged, corrected republication of 1980. Classics in Applied Mathematics. 20. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. xxiv, 398 p. (1998).
5. Stewart, G.W.: *Matrix Algorithms. Vol II: Eigensystems* Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. (2001)
6. Wilkinson, J.H.: *The Algebraic Eigenvalue Problem* Clarendon Press Oxford 1965.
7. Wilkinson, J.H.; Reinsch, Ch.: *Handbook for Automatic Computation. Vol II Linear Algebra*. Springer Yellow Series 186 (1971).
8. Zurmühl, Rudolf; Falk, Sigurd: *Matrizen und ihre Anwendungen*. Teil 1: 6. Aufl. 1992, Teil 2: 5. Aufl. 1986 Berlin: Springer.

Inhaltsverzeichnis

1	Das Matrizen–Eigenwertproblem	5
1.1	Lokalisierung von Eigenwerten. Die Sensitivität des Eigenwertproblems	7
1.2	Unitäre Ähnlichkeitstransformation auf obere Hessenberg-Form bzw. Tridiagonalform	18
1.3	Eigenwerte einer hermiteschen Tridiagonalmatrix	22
1.4	Bestimmung der Eigenvektoren einer hermiteschen Dreibandmatrix	27
1.5	Direkte Iteration nach v. Mises und Verfahren von Wielandt	30
1.5.1	Die simultane Vektoriteration	37
1.6	Das Jacobi–Verfahren	41
1.7	Das QR–Verfahren	48
1.8	Das Lanczos–Verfahren	61
1.9	Allgemeine Eigenwertprobleme	68
1.10	Die Singulärwertzerlegung (svd)	73
1.11	Zusammenfassung	78
2	Zugang zu numerischer Software und anderer Information	79
2.1	Softwarebibliotheken	79
2.2	Suchen nach software	80
2.3	Hilfe bei Fragen	80

Kapitel 1

Das Matrizen–Eigenwertproblem

In diesem Kapitel beschäftigen wir uns mit der Lokalisierung und numerischen Berechnung der reellen und komplexen Eigenwerte einer Matrix $A \in \mathbb{C}^{n \times n}$ (oder $\mathbb{R}^{n \times n}$) und der zugehörigen Eigenvektoren. Von naiven Standpunkt aus könnte man vermuten, daß es mit der Bestimmung der Nullstellen λ_i des charakteristischen Polynoms

$$p_n(\lambda; A) \stackrel{def}{=} \det(A - \lambda I)$$

und der Lösung der homogenen Gleichungssysteme

$$(A - \lambda_i I)x_i = 0$$

getan sei, also der Kombination eines skalaren Nullstellenproblems mit linearer Algebra. Der so formal beschriebene Bestimmungsweg:

- Berechnung der Koeffizienten von $p_n(\lambda; A)$
- Nullstellenbestimmung
- Lösung homogener Gleichungssysteme

erweist sich in der Praxis jedoch als völlig unbrauchbar, sowohl unter dem Gesichtspunkt des Rechenaufwandes als auch unter dem Gesichtspunkt der numerischen Stabilität. Zur Erläuterung des letzteren diene das folgende kleine Beispiel:

Die Matrix

$$A = \begin{pmatrix} 1000 & 1 \\ 1 & 1000 \end{pmatrix}$$

hat die Eigenwerte $\lambda_1 = 1001$ und $\lambda_2 = 999$. Ändert man A ab zu

$$\tilde{A} = \begin{pmatrix} 1000.001 & 1 \\ 1 & 1000 \end{pmatrix}$$

dann erhält man $\tilde{\lambda}_1 = 1001.00050\dots$, $\tilde{\lambda}_2 = 999.00050\dots$

Es ist $p_2(\lambda; A) = \lambda^2 - 2000\lambda + 999999$ und

Satz 1.1.1 Sei $A \in \mathbb{C}^{n \times n}$ und $\|\cdot\|$ eine einer Vektornorm zugeordnete Matrixnorm auf $\mathbb{C}^{n \times n}$. Dann gilt für jeden Eigenwert $\lambda(A)$

$$|\lambda(A)| \leq \rho(A) \leq \|A\|$$

□

Eine bereits wesentlich genauere Lokalisierung liefert häufig

Satz 1.1.2 Kreisesatz von Gerschgorin Sei $A \in \mathbb{C}^{n \times n}$ und

$$\mathcal{K}_i := \left\{ \lambda \in \mathbb{C} : |\lambda - \alpha_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |\alpha_{ij}| \right\}$$

$$\tilde{\mathcal{K}}_i := \left\{ \lambda \in \mathbb{C} : |\lambda - \alpha_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |\alpha_{ji}| \right\}$$

Ist dann $\lambda(A)$ ein Eigenwert von A , dann gilt:

$$\lambda(A) \in \left(\bigcup_{i=1}^n \mathcal{K}_i \right) \cap \left(\bigcup_{i=1}^n \tilde{\mathcal{K}}_i \right).$$

Beweis als einfache Übung. Hinweis: $Ax = \lambda x$, $x \neq 0$. Betrachte Zeile i mit $|x_i| = \|x\|_\infty$. □

Da die Matrizen A und $D^{-1}AD$ die gleichen Eigenwerte besitzen, kann man manchmal durch die Wahl geeigneter Transformationsmatrizen D (gewöhnlich beschränkt man sich auf Diagonalmatrizen) die Aussage von Satz 1.1.2 bedeutend verschärfen.

Beispiel 1.1.1 Sei

$$A = \begin{pmatrix} 1 & 10^{-3} & 10^{-4} \\ 10^{-3} & 2 & 10^{-3} \\ 10^{-4} & 10^{-3} & 3 \end{pmatrix}$$

Dann gilt nach Satz 1.1.2, weil A symmetrisch, d.h. $\lambda = \lambda(A)$ reell, für jeden Eigenwert von A

$$\lambda \in [1 - 0.0011, 1 + 0.0011] \cup [2 - 0.002, 2 + 0.002] \cup [3 + 0.0011, 3 + 0.0011].$$

Mit $D_1 := \text{diag}(1, 100, 10)$, $D_2 := \text{diag}(100, 1, 100)$, $D_3 := \text{diag}(10, 100, 1)$ erhält man nacheinander auch die folgenden Einschließungsmengen:

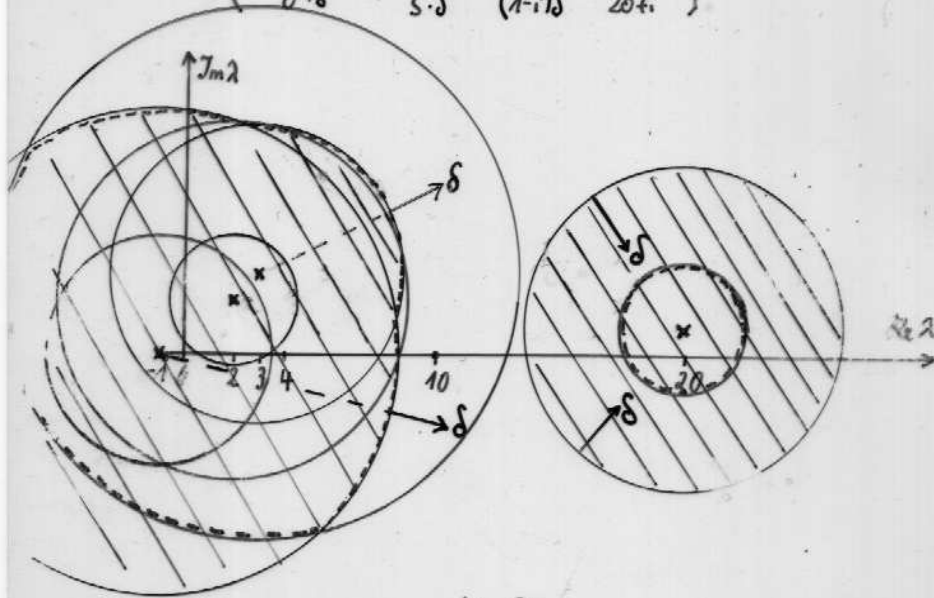
$$\begin{aligned} & [1 - 2 \cdot 10^{-5}, 1 + 2 \cdot 10^{-5}] \cup [2 - 0.11, 2 + 0.11] \cup [3 - 0.0011, 3 + 0.0011] \\ & [1 - 0.1001, 1 + 0.1001] \cup [2 - 2 \cdot 10^{-5}, 2 + 2 \cdot 10^{-5}] \cup [3 - 0.1001, 3 + 0.1001] \\ & [1 - 0.0011, 1 + 0.0011] \cup [2 - 2 \cdot 10^{-2}, 2 + 2 \cdot 10^{-2}] \cup [3 - 2 \cdot 10^{-5}, 3 + 2 \cdot 10^{-5}] \end{aligned}$$

und der Durchschnitt aller dieser Bereiche ist

$$[1 - 2 \cdot 10^{-5}, 1 + 2 \cdot 10^{-5}] \cup [2 - 2 \cdot 10^{-5}, 2 + 2 \cdot 10^{-5}] \cup [3 - 2 \cdot 10^{-5}, 3 + 2 \cdot 10^{-5}] \quad \square$$

Kreisesatz von Gerschgorin :

$$A = \begin{bmatrix} 2+2i & 1+i & i & 0 \\ 2+2i & 3+3i & 2i & 1/\delta \\ 3+3i & 4 & -1 & (1-i)/\delta \\ 0 \cdot \delta & 5 \cdot \delta & (1-i)\delta & 20+i \end{bmatrix}$$



- Kreise für A
- Kreise für A^T
- Schnitt \cup Kreise (A) und \cup Kreise (A^T) enthält alle EW
- Wirkung Ähnlichkeitsskalierung mit $\delta < 1$ auf Kreisradius

Es gilt sogar die folgende Verschärfung von Satz 1.1.2:

Zusatz zu Satz 1.1.2

Ist $\{i_1, \dots, i_m\}$ Permutation von $\{1, \dots, n\}$ und

$$\left(\bigcup_{j=1}^m \mathcal{K}_{i_j}\right) \cap \mathcal{K}_{i_s} = \emptyset \quad s = m+1, \dots, n,$$

dann enthält $\bigcup_{j=1}^m \mathcal{K}_{i_j}$ genau m Eigenwerte von A (mit ihrer Vielfachheit gezählt),

d.h. jede Wegzusammenhangskomponente von $\bigcup_{i=1}^n \mathcal{K}_i$ enthält genauso viele Eigenwerte wie Kreise. □

Beweis: Man setze

$$\begin{aligned} D &:= \text{diag}(\alpha_{11}, \dots, \alpha_{nn}) \\ B(\tau) &:= D + \tau(A - D) \quad 0 \leq \tau \leq 1, \end{aligned}$$

d.h. $B(0) = D$ und $B(1) = A$. Alle Eigenwerte von $B(\tau)$ liegen nach Satz 1.1.2 in

$$\bigcup_{i=1}^n \mathcal{K}_i(\tau); \quad \mathcal{K}_i(\tau) = \{z \in \mathbb{C} : |z - \alpha_{ii}| \leq \tau \sum_{\substack{j \neq i \\ j=1}}^n |\alpha_{ij}|\}$$

und die Aussage vom Zusatz zu Satz 1.1.2 gilt trivialerweise für $B(0)$. Die Eigenwerte von $B(\tau)$ hängen nach Satz 1.1.3 stetig von τ ab. Da aber $\bigcup_{j=1}^m \mathcal{K}_{i_j}(0)$ genau m Eigenwerte von $B(0)$ enthält und

$$\forall \tau \in [0, 1]: \quad \left(\bigcup_{j=1}^m \mathcal{K}_{i_j}(\tau)\right) \cap \mathcal{K}_{i_s}(1) \subset \left(\bigcup_{j=1}^m \mathcal{K}_{i_j}(1)\right) \cap \mathcal{K}_{i_s}(1) = \emptyset$$

enthält auch $\bigcup_{j=1}^m \mathcal{K}_{i_j}(\tau)$ genau m Eigenwerte für $0 \leq \tau \leq 1$ □

Beispiel 1.1.1 Fortsetzung

Somit gilt für die drei Eigenwerte $\lambda_1, \lambda_2, \lambda_3$ von A bei geeigneter Numerierung:

$$i - 2 \cdot 10^{-5} \leq \lambda_i \leq i + 2 \cdot 10^{-5}, \quad i = 1, 2, 3$$

□

Für eine beliebige Matrix kennt man nur die folgende allgemeine Störungsaussage für die Eigenwerte, die keine Annahmen über das Eigenvektorsystem voraussetzt:

Satz 1.1.3 Seien $A, B \in \mathbb{C}^{n \times n}$, λ_i die Eigenwerte von A und λ'_i die Eigenwerte von B , $i = 1, \dots, n$ (jeweils mit ihrer Vielfachheit gezählt.)

Sei

$$\rho := \max\{|\alpha_{ij}|, |\beta_{ij}| : 1 \leq i, j \leq n\}$$

$$\delta := \frac{1}{n\rho} \sum_{i=1}^n \sum_{j=1}^n |\alpha_{ij} - \beta_{ij}| \quad .$$

Dann gibt es eine Numerierung der λ_i und λ'_i , so daß zu jedem λ_i ein λ'_i gehört mit

$$|\lambda_i - \lambda'_i| \leq 2(n+1)^2 \rho \sqrt[n]{\delta}$$

(d.h. die Eigenwerte einer Matrix sind Hölderstetige Funktionen der Matrixkoeffizienten vom Index $\frac{1}{n}$)

Beweis: siehe bei Ostrowski, A.M.: Solution of Equations in Euclidean and Banach Spaces, 3.ed., Acad. Press 1973, p.334-335 und 276-279. \square

Die Abschätzung dieses Satzes kann noch verfeinert werden. Elsner (*An optimal bound for the spectral variation of two matrices*, Lin Alg Applics 71 (1985), 77-80) kommt in obiger Notation zu der Abschätzung

$$|\lambda_i - \lambda'_i| \leq 4(\|A\|_2 + \|B\|_2)^{1-\frac{1}{n}} \|B - A\|_2^{\frac{1}{n}}$$

Die Aussage dieses Satzes (Hölderstetigkeit der Eigenwerte mit Hölderindex $1/n$) kann nicht verbessert werden, wie folgendes Beispiel zeigt:

Beispiel 1.1.2

$$A(\varepsilon) = \begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 \\ \varepsilon & 0 & \dots & 0 & 1 \end{pmatrix}$$

hat für $\varepsilon = 0$ den n -fachen Eigenwert 1 und für $\varepsilon > 0$ die n paarweise verschiedenen Eigenwerte

$$\lambda_j = 1 - \varepsilon^{(1/n)} \exp(2\pi i(j-1)/n), \quad j = 1, \dots, n$$

wobei mit $\varepsilon^{(1/n)}$ der (reelle) Hauptwert gemeint ist.

Ist x ein Eigenvektor von A , dann kann man mit Hilfe von

$$Ax = \lambda x \Rightarrow \lambda = \frac{x^H Ax}{x^H x}$$

den zugehörigen Eigenwert berechnen. Hat man eine Eigenvektornäherung x (im folgenden Satz kann jeder beliebige Vektor $x \neq 0$, für den auch $Ax \neq 0$ ist, als eine solche Näherung dienen), dann kann man mit Hilfe dieses **Rayleighquotienten**

$$R(x; A) := \frac{x^H Ax}{x^H x}$$

eine zugehörige Eigenwertnäherung definieren, für die man ebenfalls eine Einschließungsaussage herleiten kann. Weil für $x \neq 0$ und $Ax = 0$ x ein Eigenvektor zum Eigenwert null von A ist, schliessen wir diesen Fall jetzt aus:

Satz 1.1.4 $A \in \mathbb{C}^{n \times n}$ sei diagonalähnlich mit Eigenwerten $\lambda_1, \dots, \lambda_n$. Sei $x \in \mathbb{C}^n, x \neq 0$ und $Ax \neq 0$. Definiere

$$\lambda := R(x; A)$$

Dann gilt

(i) $\|Ax - \lambda x\|_2^2 \leq \|Ax - cx\|_2^2 \quad \forall c \in \mathbb{C}$

(ii) $\exists \lambda_j \neq 0, \quad \lambda_j$ Eigenwert von A und

$$\left| \frac{\lambda_j - \lambda}{\lambda_j} \right| \leq \frac{\|Ax - \lambda x\|_2}{\|Ax\|_2} \text{cond}_{\|\cdot\|_2}(U)$$

wobei $U = (u_1, \dots, u_n)$ ein vollständiges Eigenvektorsystem von A ist

(iii) Ist A normal, (d.h. $AA^H = A^H A$), dann $\exists \lambda_j \neq 0$ Eigenwert von A mit

$$\left| \frac{\lambda_j - \lambda}{\lambda_j} \right| \leq \frac{\|Ax - \lambda x\|_2}{\|Ax\|_2}$$

□

Beweis:

(i) o.B.d.A. $\|x\|_2 = 1$

$$\begin{aligned} \|Ax - cx\|_2^2 &= (x^H A^H - \bar{c}x^H)(Ax - cx) = x^H A^H Ax - \bar{c}x^H Ax - cx^H A^H x + |c|^2 \\ &= \|Ax\|_2^2 + |c - x^H Ax|^2 - |x^H Ax|^2 \geq \|Ax\|_2^2 - |x^H Ax|^2 \geq 0 \quad \text{mit "=" für } c = \lambda \end{aligned}$$

(Man beachte daß $|x^H Ax|^2 \leq \|Ax\|_2^2 \|x\|_2^2$ nach der Cauchy-Schwarzschen Ungleichung und $\|x\|_2^2 = 1$ nach Setzung.)

(ii) Sei $U^{-1}AU = \text{diag}(\lambda_1, \dots, \lambda_n) \stackrel{\text{def}}{=} \Lambda$; $y \stackrel{\text{def}}{=} U^{-1}x$

$$\begin{aligned}
\frac{\|Ax - \lambda x\|_2}{\|Ax\|_2} &= \frac{\|U(\Lambda - \lambda I)U^{-1}x\|_2}{\|U\Lambda U^{-1}x\|_2} \\
&\geq \frac{\|(\Lambda - \lambda I)y\|_2}{\|U^{-1}\|_2 \|U\|_2 \|\Lambda y\|_2} \\
&= \frac{1}{\|U\|_2 \|U^{-1}\|_2} \cdot \left(\frac{\sum_{i=1}^n |\lambda_i - \lambda|^2 |\eta_i|^2}{\sum_{i=1}^n |\lambda_i|^2 |\eta_i|^2} \right)^{1/2} \\
&= \frac{1}{\text{cond}_{\|\cdot\|_2}(U)} \left(\frac{|\lambda|^2 \sum_{\lambda_i=0} |\eta_i|^2 + \sum_{\lambda_i \neq 0} \left| \frac{\lambda_i - \lambda}{\lambda_i} \right|^2 |\eta_i \lambda_i|^2}{\sum_{\lambda_i \neq 0} |\lambda_i \eta_i|^2} \right)^{1/2} \\
&\geq \frac{1}{\text{cond}_{\|\cdot\|_2}(U)} \cdot \min_{i: \lambda_i \neq 0} \left| \frac{\lambda_i - \lambda}{\lambda_i} \right|
\end{aligned}$$

Man beachte, daß

$$\forall z : \|Uz\| \geq \frac{1}{\|U^{-1}\|} \|z\|.$$

(iii) Für normales A existiert ein unitäres vollständiges Eigenvektorsystem, d.h. es wird $\text{cond}_{\|\cdot\|_2}(U) = 1$

□

Der oben eingeführte Rayleighquotient ist also (im Sinne einer Einsetzprobe für ein Eigenwert–Eigenvektorpaar) eine optimale Schätzung für einen Eigenwert zu einer gegebenen Eigenvektornäherung.

Für hermitesche Matrizen, die in den Anwendungen eine besonders wichtige Rolle spielen, hat der Rayleighquotient viele schöne Eigenschaften, deren wichtigste hier angeführt seien.

Satz 1.1.5 Sei $A \in \mathbb{C}^{n \times n}$ hermitisch mit vollständigem unitärem Eigenvektorsystem $X = (x_1, \dots, x_n)$, $Ax_j = \lambda_j x_j$ $j = 1, \dots, n$. $\tilde{x} \in \mathbb{C}^n$ sei gegeben als Näherung für x_j mit

$$\tilde{x}^H \tilde{x} = 1, \quad \tilde{x} = x_j + \sum_{k=1}^n \epsilon_k x_k, \quad |\epsilon_k| \leq \epsilon \quad (\forall k)$$

Dann gilt

$$|R(\tilde{x}; A) - \lambda_j| \leq \sum_{\substack{i=1 \\ i \neq j}}^n |\lambda_i - \lambda_j| |\epsilon_i|^2 \leq 2\|A\|(n-1)\epsilon^2$$

d.h. der Fehler im Rayleighquotienten ist quadratisch klein in den Fehlern der Eigenvektornäherung. □

Beweis:

$$\begin{aligned}
R(\tilde{x}; A) &= (x_j + \sum_{k=1}^n \epsilon_k x_k)^H \underbrace{\sum_{i=1}^n \lambda_i x_i x_i^H}_A (x_j + \sum_{k=1}^n \epsilon_k x_k) \\
&= \sum_{i=1}^n \lambda_i \underbrace{((x_j + \sum_{k=1}^n \epsilon_k x_k)^H x_i)(x_i^H (x_j + \sum_{k=1}^n \epsilon_k x_k))}_{\delta_{ij} + \sum_{k=1}^n \bar{\epsilon}_k \delta_{ki}} \\
&\quad \underbrace{\hspace{10em}}_{|\delta_{ij} + \sum_{k=1}^n \epsilon_k \delta_{ki}|^2} \\
&= \sum_{\substack{i=1 \\ i \neq j}}^n \lambda_i |\epsilon_i|^2 + \lambda_j |1 + \epsilon_j|^2 \\
&= \sum_{\substack{i=1 \\ i \neq j}}^n (\lambda_i - \lambda_j) |\epsilon_i|^2 + \lambda_j \underbrace{(|1 + \epsilon_j|^2 + \sum_{\substack{i=1 \\ i \neq j}}^n |\epsilon_i|^2)}_{=\tilde{x}^H \tilde{x} = 1}
\end{aligned}$$

Betragsabschätzung, Anwendung der Dreiecksungleichung und der trivialen Schranke

$$|\lambda_i - \lambda_j| \leq 2\rho(A) \leq 2\|A\|$$

□

Eine vollständige Charakterisierung aller Eigenwerte einer hermiteschen Matrix durch eine Maximierungsaufgabe mit Nebenbedingungen liefert der

Satz 1.1.6 Courant'sches Minimax-Prinzip Sei $A \in \mathbb{C}^{n \times n}$ hermitisch. Die Eigenwerte von A seien (mit ihrer Vielfachheit gezählt) geordnet nach

$$\lambda_1 \geq \dots \geq \lambda_n .$$

\mathcal{V}_j bezeichne das System aller j -dimensionalen Teilräume von \mathbb{C}^n , $\mathcal{V}_0 = \{0\}$. Es gilt

$$\lambda_k = \min_{V \in \mathcal{V}_{k-1}} \max\{R(x; A) : x \neq 0, x^H v = 0 \forall v \in V\} = \min_{V: \dim(V)=k-1} \left\{ \max_{\substack{x: x \perp V \\ x \neq 0}} R(x; A) \right\}$$

$$\lambda_k = \max_{V \in \mathcal{V}_{n-k}} \min\{R(x; A) : x \neq 0, x^H v = 0 \forall v \in V\} = \max_{V: \dim(V)=n-k} \left\{ \min_{\substack{x: x \perp V \\ x \neq 0}} R(x; A) \right\}$$

□

<<

Beweis: Sei $U = (u_1, \dots, u_n)$ ein unitäres vollständiges Eigenvektorsystem von A , d.h. $Au_i = \lambda_i u_i$, $U^H U = I$. Ist $x \in \mathbb{C}^n$ beliebig, dann

$$x = \sum_{i=1}^n \gamma_i u_i \quad \text{mit } \gamma_i = u_i^H x$$

Somit

$$\begin{aligned}
 R(x; A) &= \frac{x^H U \Lambda U^H x}{x^H U^H U x} \\
 &= \frac{\sum_{i=1}^n |\gamma_i|^2}{\sum_{j=1}^n |\gamma_j|^2} \lambda_i \quad \begin{cases} \geq \lambda_k, & \text{falls } \gamma_{k+1} = \dots = \gamma_n = 0 \\ \leq \lambda_k, & \text{falls } \gamma_1 = \dots = \gamma_{k-1} = 0 \end{cases}
 \end{aligned}$$

Ist nun $V \in \mathcal{V}_{k-1}$, dann existiert $\tilde{x} \in \mathbb{C}^n$ mit folgenden Eigenschaften:

$$\tilde{x} \neq 0, \quad \tilde{x} = \sum_{i=1}^k \gamma_i u_i, \quad \tilde{x}^H v = 0 \quad \forall v \in V$$

Setze zum Beweis mit einer orthonormierten Basis v_1, \dots, v_{k-1} von V

$$g = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_k \end{pmatrix}, \quad g \neq 0 \quad \text{Lösung von} \quad \begin{pmatrix} v_1^H \\ \vdots \\ v_{k-1}^H \end{pmatrix} (u_1, \dots, u_k) g = 0$$

Also ist in diesem Falle

$$\max\{R(x; A) : x \neq 0, x^H v = 0 \quad \forall v \in V\} \geq \lambda_k$$

Gleichheit tritt hier ein für den Fall $V = \text{span}\{u_1, \dots, u_{k-1}\}$, d.h.

$$g = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \gamma_k \end{pmatrix} \quad \gamma_k \neq 0. \text{ Ist } V \in \mathcal{V}_{n-k}, \text{ dann existiert } \tilde{x} \in \mathbb{C}^n \text{ mit}$$

$$\tilde{x} \neq 0, \quad \tilde{x} = \sum_{j=k}^n \gamma_j u_j, \quad \tilde{x}^H v = 0 \quad \forall v \in V$$

Sei dazu $g = (\gamma_k, \dots, \gamma_n)^T \neq 0$ eine Lösung von $\begin{pmatrix} v_1^H \\ \vdots \\ v_{n-k}^H \end{pmatrix} (u_k, \dots, u_n) g = 0$ mit einer orthonormierten Basis v_1, \dots, v_{n-k} von \mathcal{V}_{n-k} , also

$$\min\{R(x; A) : x \neq 0, x^H v = 0 \quad \forall v \in V\} \leq \lambda_k$$

mit Gleichheit für $V = \text{span}\{u_{k+1}, \dots, u_n\}$ (dann ist $g = (\gamma_k, 0, \dots, 0)^T$) □

>>

Eine Folgerung ist:

Satz 1.1.7 Seien $A, B \in \mathbb{C}^{n \times n}$ beide hermitisch. $\lambda_i(A), \lambda_i(B)$ bezeichne die Eigenwerte von A und B und die Numerierung sei so vorgenommen, daß $\lambda_1(A) \geq \dots \geq \lambda_n(A), \quad \lambda_1(B) \geq \dots \geq \lambda_n(B)$. Dann gilt

$$\forall k \in \{1, \dots, n\} : \quad |\lambda_k(A) - \lambda_k(B)| \leq \rho(B - A) \quad (1.1)$$

□

Beweis: Setze $B := A + (B - A), \quad C := B - A$, verwende Satz 1.1.6 (Übung!) □

(1.1) stellt natürlich ein viel günstigeres Resultat dar als der Satz 1.1.3. Man beachte, daß (1.1) auch bei mehrfachen Eigenwerten gilt! Natürlich ist die Voraussetzung, daß beide Matrizen hermitisch sein sollen, sehr einschränkend. Die Eigenwerte einer diagonalähnlichen Matrix hängen wenigstens noch lipschitzstetig von den Matrixkoeffizienten ab. Dies besagt

Satz 1.1.8 Bauer-Fike-Theorem *Ist $A \in \mathbb{C}^{n \times n}$ diagonalähnlich, $U = (u_1, \dots, u_n)$ ein vollständiges Eigenvektorsystem von A , und ist $B \in \mathbb{C}^{n \times n}$ beliebig, dann gibt es zu jedem Eigenwert $\lambda_j(B)$ einen Eigenwert $\lambda_{i(j)}(A)$ von A mit*

$$|\lambda_{i(j)}(A) - \lambda_j(B)| \leq \text{cond}_{\|\cdot\|_\infty}(U) \|B - A\|_\infty \quad (1.2)$$

□

(Bem.: Diese Aussage gilt sogar für jede absolute Norm, d.i. eine Norm mit $\| |x| \| = \|x\| \quad \forall x \in \mathbb{C}^n$.)

Beweisskizze: Nichttrivialer Fall:

$\lambda(B) \neq \lambda_i(A) \quad i = 1, \dots, n, \quad \lambda(B)$ ein Eigenwert von B mit Eigenvektor $x \neq 0 \quad \Rightarrow$

$Bx - Ax = \lambda(B)x - Ax \quad \Rightarrow$

$x = (\lambda(B)I - A)^{-1}(B - A)x$, Normabschätzung, $I = UU^{-1}$,

$A = U\Lambda_A U^{-1}$ einsetzen.

$$\|(\lambda(B)I - \Lambda_A)^{-1}\|_\infty = \frac{1}{\min_i |\lambda(B) - \lambda_i(A)|}$$

□

Für nichtdiagonalähnliche Matrizen kann eine zu (1.1) bzw. (1.2) analoge Aussage nicht erwartet werden, siehe obiges Beispiel zur Hölderstetigkeit.

Neben den bis jetzt abgeleiteten Eigenwertabschätzungen interessieren natürlich auch asymptotische Fehleraussagen für die Eigenwerte und Eigenvektoren für kleine Störungen.

Satz 1.1.9 Wilkinson Sei $A \in \mathbb{C}^{n \times n}$ diagonalähnlich, $X = (x_1, \dots, x_n)$ ein vollständiges Eigenvektorsystem von A , $Ax_i = \lambda_i x_i$ sowie $\|x_i\|_2 = 1 \quad i = 1, \dots, n$.

Ferner sei

$$X^{-1} =: Y = \begin{pmatrix} y_1^H \\ \vdots \\ y_n^H \end{pmatrix} \quad (\text{d.h. } y_i^H A = y_i^H \lambda_i \quad y_i \text{ sogenannter Linkseigenvektor zu } \lambda_i).$$

λ_j sei ein einfacher Eigenwert von A . Dann gilt: zu $F \in \mathbb{C}^{n \times n}$ mit $\|F\|_2$ hinreichend klein existiert ein einfacher Eigenwert μ_j von $A + F$ mit Eigenvektor z_j , $\|z_j\| = 1$ so daß

$$\begin{aligned} \mu_j &= \lambda_j + \frac{y_j^H F x_j}{\|y_j\|_2 \|x_j\|_2} \cdot \frac{\|y_j\|_2 \|x_j\|_2}{y_j^H x_j} + \mathcal{O}(\|F\|_2^2) \\ z_j &= x_j + \left(\sum_{\substack{i=1 \\ i \neq j}}^n \frac{y_i^H F x_j}{\|y_i\|_2 \|x_j\|_2} \cdot \frac{1}{\lambda_j - \lambda_i} \frac{\|y_i\|_2 \|x_i\|_2}{y_i^H x_i} x_i \right) + \mathcal{O}(\|F\|_2^2) \end{aligned}$$

□

Beweisskizze: Man benutze den Hauptsatz über implizite Funktionen für das Problem $G(x, y, \lambda, \mu, \varepsilon) = 0$ mit $G(x_j, y_j, \lambda_j, \lambda_j, 0) = 0$,

$$G(x, y, \lambda, \varepsilon) = \begin{pmatrix} (A^H + \varepsilon F_0^H)y - \mu y \\ (A + \varepsilon F_0)x - \lambda x \\ y^H x - 1 \\ x^H x - 1 \end{pmatrix}$$

$$F = \varepsilon F_0$$

mit den Unbekannten x, y (Rechts- und Linkseigenvektor) und λ, μ als Eigenwerten und ε als Parameter. Diese System wird nun in der Umgebung des Punktes $x_j, y_j, \lambda_j, \mu_j = \lambda_j$, der eine offensichtliche Lösung ist, betrachtet mit ε in einer geeigneten Nullumgebung. Damit stellt man die Lösung x, y, λ als Funktionen von ε dar. Bei komplexen Eigenwerten und (oder) komplexer Matrix schreibt man dieses System zunächst in ein reelles System doppelter Dimension um. Man kann dabei A schon in der Jordan-Normalform (also hier diagonal) annehmen. Der wesentliche Beweisschritt dabei ist der Nachweis der Invertierbarkeit der Jacobimatrix des Systems bezüglich x, y und λ (bzw. bzgl. der Real- und Imaginärteile), was an die Einfachheit des Eigenwertes gebunden ist. □

Statt der asymptotischen Aussage dieses Satzes gibt es auch strenge quantitative Abschätzungen der gleichen Art, siehe dazu bei G.W.Stewart.

Entscheidender Fehlerverstärkungsfaktor für einen Eigenwert ist also $\|y_j\|_2 \|x_j\|_2 / |y_j^H x_j| (\gg 1 \text{ möglich bei nichtnormalen Matrizen})$, während bei einem

Eigenvektor zusätzlich die **Separation** der Eigenwerte und alle Terme

$$\|y_i\|_2 \|x_i\|_2 / |y_i^H x_i| = \frac{1}{\cos(\angle(x_i, y_i))}$$

eine Rolle spielen. Nach den Setzungen des Satzes ist natürlich $|y_i^H x_i| = 1$, um aber die Tatsache hervorzuheben, daß im allgemeinen Rechts- und Linkseigenvektoren nicht orthogonal sind, bevorzugen wir diese Schreibweise, bei der der erste Term stets $\leq \|F\|$ ist, während die übrigen die Fehlerverstärkungs- bzw. Dämpfungsfaktoren darstellen.

1.2 Unitäre Ähnlichkeitstransformation auf obere Hessenberg-Form bzw. Tridiagonalform

Die Lösung des vollständigen Matrizen-Eigenwertproblems für eine vollbesetzte Matrix beginnt stets mit einer Ähnlichkeitstransformation auf "kondensierte" Form, mit dem Ziel, die Matrix möglichst "schmal" besetzt zu erhalten. Diese Transformation schleppt beim praktischen Rechnen unvermeidbare Rundungsfehler ein. Um noch brauchbar zu sein, dürfen jedoch die Eigenwerte dadurch nicht wesentlich stärker verfälscht werden als wenn die Ausgangsmatrix selbst im Rahmen der Rundungsgenauigkeit abgeändert würde. Aus diesem Grund kommt bei einer allgemeinen Matrix nur die Transformation auf Hessenberggestalt in Frage (was wir aber nicht beweisen wollen). Eine Transformation einer allgemeinen Matrix auf Tridiagonalgestalt ist bekannt, aber leider numerisch instabil. Wesentlich ist, daß diese Transformation keinerlei Information über das Eigensystem der Matrix benötigt.

In diesem Abschnitt geben wir eine **unitäre** Ähnlichkeitstransformation auf Hessenberggestalt an. Eine Ähnlichkeitstransformation mit Dreiecksmatrizen, wie sie bei der Gauss-Elimination benutzt werden, ist auch möglich, wir beschränken uns hier aber auf die Verwendung von Householdermatrizen, die numerisch besonders unempfindlich ist.

$$\square = A \rightarrow U_1 A U_1 \rightarrow \dots U_{n-2} \dots U_1 A U_1 U_2 \dots U_{n-2} = \triangle$$

Dabei sind die einzelnen U_i hermitisch und unitär. Ist A selbst hermitisch, dann wird

$$(U_{n-2} \dots U_1 A U_1 U_2 \dots U_{n-2})^H = U_{n-2} \dots U_1 A U_1 U_2 \dots U_{n-2}, \quad \text{hermitisch}$$

d.h. die **transformierte Matrix** erhält automatisch **Dreibandform!** Die Transfor-

mation verläuft in $n - 2$ Schritten. Sei nach $j - 1$ Schritten

$$A_j = U_{j-1} \dots U_1 A U_1 U_2 \dots U_{j-1} = \begin{pmatrix} \alpha_{11}^{(j)} & \alpha_{12}^{(j)} & \cdots & \alpha_{1,j-1}^{(j)} & \alpha_{1j}^{(j)} & \cdots & \alpha_{1n}^{(j)} \\ \alpha_{21}^{(j)} & \alpha_{22}^{(j)} & \cdots & \alpha_{2,j-1}^{(j)} & \alpha_{2j}^{(j)} & \cdots & \alpha_{2n}^{(j)} \\ 0 & \alpha_{32}^{(j)} & & \vdots & \vdots & & \vdots \\ \vdots & 0 & \ddots & \alpha_{j,j-1}^{(j)} & \alpha_{jj}^{(j)} & & \alpha_{j,n}^{(j)} \\ \vdots & \vdots & \ddots & 0 & \alpha_{j+1,j}^{(j)} & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & \alpha_{n,j}^{(j)} & \cdots & \alpha_{n,n}^{(j)} \end{pmatrix}$$

Dann wird

$$A_{j+1} = U_j A_j U_j$$

mit

$$U_j = \left(\begin{array}{c|c} I & O \\ \hline O & \hat{U}_j \end{array} \right) \quad \hat{U}_j = I - \beta_j \hat{w}_j \hat{w}_j^H$$

wobei wiederum \hat{U}_j so konstruiert ist, daß

$$\hat{U}_j \begin{pmatrix} \alpha_{j+1,j}^{(j)} \\ \vdots \\ \vdots \\ \alpha_{n,j}^{(j)} \end{pmatrix} = -\exp(i\varphi_j) \sigma_j \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Es ergeben sich die bekannten Formeln

$$\hat{w}_j = \begin{pmatrix} \exp(i\varphi_j) (|\alpha_{j+1,j}^{(j)}| + \sigma_j) \\ \alpha_{j+2,j}^{(j)} \\ \vdots \\ \alpha_{n,j}^{(j)} \end{pmatrix} \quad \sigma_j = \left(\sum_{k=j+1}^n |\alpha_{k,j}^{(j)}|^2 \right)^{1/2}$$

$$\beta_j = \frac{2}{\hat{w}_j^H \hat{w}_j} \quad \alpha_{j+1,j}^{(j)} = \exp(i\varphi_j) |\alpha_{j+1,j}^{(j)}| \quad (\text{def } \varphi_j)$$

Da die ersten j Spalten von U_j Einheitsspalten sind, ändert die Multiplikation von $U_j A_j$ mit U_j von rechts die eben neu erzeugten Nullen in Spalte j nicht. Die Multiplikation von A_j mit U_j von links ändert die ersten j Zeilen nicht. Damit ist die Transformation bereits vollständig hergeleitet.

Bei der praktischen Durchführung der Transformation nutzt man die spezielle Struktur von U_j aus. Sei

$$A_j = \left(\begin{array}{c|c} A_{11}^{(j)} & A_{12}^{(j)} \\ \hline A_{21}^{(j)} & A_{22}^{(j)} \end{array} \right)$$

Dann wird

$$A_{j+1} = \left(\begin{array}{c|c} A_{11}^{(j)} & A_{12}^{(j)} \hat{U}_j \\ \hline \hat{U}_j A_{21}^{(j)} & \hat{U}_j A_{22}^{(j)} \hat{U}_j \end{array} \right)$$

$$\begin{aligned} \hat{U}_j A_{21}^{(j)} &= A_{21}^{(j)} - \beta_j \hat{w}_j \hat{w}_j^H A_{21}^{(j)} = A_{21}^{(j)} - \hat{u}_j \hat{z}_j^H \\ A_{12}^{(j)} \hat{U}_j &= A_{12}^{(j)} - \beta_j A_{12}^{(j)} \hat{w}_j \hat{w}_j^H = A_{12}^{(j)} - \hat{y}_j \hat{u}_j^H \\ \hat{U}_j A_{22}^{(j)} \hat{U}_j &= (I - \beta_j \hat{w}_j \hat{w}_j^H) (A_{22}^{(j)} - \beta_j A_{22}^{(j)} \hat{w}_j \hat{w}_j^H) \\ &= A_{22}^{(j)} - \hat{u}_j (\hat{w}_j^H A_{22}^{(j)}) - (A_{22}^{(j)} \hat{w}_j) \hat{u}_j^H + (\hat{w}_j^H A_{22}^{(j)} \hat{w}_j) \hat{u}_j \hat{u}_j^H \\ &= A_{22}^{(j)} - \hat{u}_j (\hat{s}_j^H - \frac{\gamma_j}{2} \hat{u}_j^H) - (\hat{t}_j - \frac{\gamma_j}{2} \hat{u}_j) \hat{u}_j^H \end{aligned}$$

mit

$$\begin{aligned} \hat{u}_j &= \beta_j \hat{w}_j \\ \hat{t}_j &= A_{22}^{(j)} \hat{w}_j \\ \gamma_j &= \hat{w}_j^H \hat{t}_j \\ \hat{s}_j^H &= \hat{w}_j^H A_{22}^{(j)} \end{aligned}$$

und $\hat{z}_j^H = \hat{w}_j^H A_{21}^{(j)}$, $\hat{y}_j = A_{12}^{(j)} \hat{w}_j$.

Nach Voraussetzung ist jedoch

$$A_{21}^{(j)} = \begin{pmatrix} 0 & \cdots & 0 & \alpha_{j+1,j}^{(j)} \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & \alpha_{n,j}^{(j)} \end{pmatrix}$$

so daß die explizite Berechnung von $\hat{U}_j A_{21}^{(j)}$ entfällt:

$$\hat{U}_j A_{21}^{(j)} = \begin{pmatrix} 0 & \cdots & 0 & -\exp(i\varphi_j) \sigma_j \\ \vdots & & \vdots & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

Im hermiteschen Fall beachte man ferner

$$\begin{aligned} A \text{ hermitisch: } \quad A_{12}^{(j)} \hat{U}_j &= (\hat{U}_j A_{21}^{(j)})^H \quad (\text{keine explizite Berechnung}) \\ \hat{t}_j &= \hat{s}_j \end{aligned}$$

wodurch sich der Gesamtrechenaufwand mehr als halbiert.

(A allgemein: $\frac{5}{3}n^3 + \mathcal{O}(n^2)$, A hermitisch: $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ wesentliche Operationen) Wir haben somit konstruktiv gezeigt:

Satz 1.2.1 Jede komplexe $n \times n$ -Matrix kann durch eine unitäre Ähnlichkeitstransformation aus $n - 2$ Householdermatrizen auf obere Hessenberggestalt transformiert werden. Ist die Ausgangsmatrix hermitisch, dann ist die resultierende Matrix hermitisch tridiagonal. \square

Beispiel 1.2.1 Wir transformieren die Matrix

$$A = \begin{pmatrix} 1 & 4 & 3 \\ 4 & -3 & 9 \\ 3 & 9 & 3 \end{pmatrix}$$

auf Tridiagonalgestalt. Es genügt ein Schritt zur Transformation auf HESSENBERGgestalt, die wegen der Symmetrie von A mit der gewünschten Tridiagonalgestalt übereinstimmt. Dabei ist $j = 1$ und $n = 3$, außerdem $A_1 = A$.

Für U_1 werden β_1 und \hat{w}_1 benötigt, um $\begin{pmatrix} \alpha_{21} \\ \alpha_{31} \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$ zu transformieren. Mit

$$\sigma_1 = \sqrt{\sum_{k=j+1}^n |\alpha_{kj}^{(j)}|^2} = \sqrt{4^2 + 3^2} = \sqrt{25} = 5$$

ist

$$\beta_1 = \frac{1}{\sigma_1(\sigma_1 + |\alpha_{21}^{(1)}|)} = \frac{1}{5(5 + 4)} = \frac{1}{45}$$

und

$$\hat{w}_1 = \begin{pmatrix} \alpha_{21} \\ \alpha_{31} \end{pmatrix} + \sigma_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \end{pmatrix} + \begin{pmatrix} 5 \\ 0 \end{pmatrix} = \begin{pmatrix} 9 \\ 3 \end{pmatrix}.$$

Von der Transformation $A_2 = U_1 A U_1$ mit

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

$$A_2 = \begin{pmatrix} I & 0 \\ 0 & \hat{U}_1 \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & \hat{U}_1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \hat{U}_1 \\ \hat{U}_1 A_{21} & \hat{U}_1 A_{22} \hat{U}_1 \end{pmatrix}$$

sind bereits $A_{11} = 1$ und $\hat{U}_1 A_{21} = -\sigma_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -5 \\ 0 \end{pmatrix}$ bekannt. Die Symmetrie von A liefert

$A_{12}\hat{U}_1 = (-5, 0)$, so daß nur $\hat{U}_1 A_{22} \hat{U}_1$ zu bestimmen ist:

$$\begin{aligned}
\hat{U}_1 A_{22} \hat{U}_1 &= (I - \beta_1 \hat{w}_1 \hat{w}_1^H) A_{22} (I - \beta_1 \hat{w}_1 \hat{w}_1^H) \\
&= A_{22} - \beta_1 \hat{w}_1 \hat{w}_1^H A_{22} - A_{22} \beta_1 \hat{w}_1 \hat{w}_1^H + (\beta_1 \hat{w}_1 \hat{w}_1^H) A_{22} (\beta_1 \hat{w}_1 \hat{w}_1^H) \\
&= \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} - \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} \\
&\quad - \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \\
&\quad + \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \\
&= \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} - \frac{1}{45} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (0, 90) - \frac{1}{45} \begin{pmatrix} 0 \\ 90 \end{pmatrix} (9, 3) \\
&\quad + \frac{270}{(45)^2} \begin{pmatrix} 9 \\ 3 \end{pmatrix} (9, 3) \\
&= \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} - \frac{1}{45} \begin{pmatrix} 0 & 810 \\ 0 & 270 \end{pmatrix} - \frac{1}{45} \begin{pmatrix} 0 & 0 \\ 810 & 270 \end{pmatrix} + \frac{270}{(45)^2} \begin{pmatrix} 81 & 27 \\ 27 & 9 \end{pmatrix} \\
&= \begin{pmatrix} -3 & 9 \\ 9 & 3 \end{pmatrix} - \begin{pmatrix} 0 & 18 \\ 0 & 6 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 18 & 6 \end{pmatrix} + \frac{2}{5} \begin{pmatrix} 27 & 9 \\ 9 & 3 \end{pmatrix} \\
&= \begin{pmatrix} -3 & -9 \\ -9 & -9 \end{pmatrix} + \frac{6}{5} \begin{pmatrix} 9 & 3 \\ 3 & 1 \end{pmatrix} \\
&= \frac{1}{5} \begin{pmatrix} 39 & -27 \\ -27 & -39 \end{pmatrix}
\end{aligned}$$

Insgesamt ist somit

$$A_2 = \frac{1}{5} \begin{pmatrix} 5 & -25 & 0 \\ -25 & 39 & -27 \\ 0 & -27 & -39 \end{pmatrix}.$$

□

Eine entsprechende Transformation auf untere Hessenbergform ist genauso möglich. Man arbeitet von rechts die Zeilen ab.

1.3 Eigenwerte einer hermiteschen Tridiagonalmatrix

Sei

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \gamma_{n-1} & \alpha_n \end{pmatrix} \quad \begin{array}{l} \gamma_i = \bar{\beta}_i \\ \alpha_i \in \mathbb{R} \end{array} \quad (1.3)$$

Wir setzen voraus: $\gamma_i \neq 0, \quad i = 1, \dots, n-1$.

Andernfalls zerfällt die Tridiagonalmatrix in kleinere Tridiagonaluntermatrizen, deren Eigenwertproblem gesondert betrachtet werden kann.

Satz 1.3.1 Ist $T \in \mathbb{C}^{n \times n}$ eine hermitesche Tridiagonalmatrix der Form (1.3) und $\gamma_i \neq 0$ für $i = 1, \dots, n-1$, dann hat T nur einfache reelle Eigenwerte.

Beweis: Eine hermitesche Matrix hat ein vollständiges Eigenvektorsystem. Es ist also für jeden Eigenwert geometrische Vielfachheit = algebraische Vielfachheit. Nach Voraussetzung enthält jedoch die Matrix $T - \lambda I$ für jedes λ eine invertierbare Untermatrix der Dimension $n-1$ (mit der Nebendiagonalen als Diagonale), also ist die geometrische Vielfachheit stets 1. \square

Bemerkung 1.3.1 Ist T eine reelle unsymmetrische Tridiagonalmatrix mit $\beta_i \gamma_i > 0, \quad i = 1, \dots, n-1$, dann kann T durch eine Ähnlichkeitstransformation mit der Diagonalmatrix $D = \text{diag}(\delta_i), \quad \delta_1 := 1, \quad \delta_{i+1} := \delta_i \sqrt{\beta_i / \gamma_i}$

$\hat{T} := DTD^{-1}$ in eine symmetrische Matrix \hat{T} überführen. Auch solche Matrizen haben also nur reelle einfache Eigenwerte. \square

Zur Berechnung der Eigenwerte von T benutzen wir den

Satz 1.3.2 Trägheitssatz von Sylvester Sei $A \in \mathbb{C}^{n \times n}$ hermitisch und $X \in \mathbb{C}^{n \times n}$ regulär. Dann haben $X^H A X$ und A gleichviele Eigenwerte $> 0, = 0, < 0$.

Beweis: siehe z.B. bei Falk-Zurmühl, Matrizen. \square

Übertragen auf $A - \mu I$ heißt das:

$A - \mu I$ und $X^H(A - \mu I)X$ haben gleichviele Eigenwerte $> 0, = 0, < 0$,

d.h. A hat entsprechend viele Eigenwerte

$> \mu, = \mu, < \mu \quad (\mu \in \mathbb{R})$. Dies Resultat soll auf T (1.3) angewendet werden mit einer Matrix

$$X \in \mathbb{C}^{n \times n} \quad \text{wo} \quad X^{-1} = \begin{pmatrix} 1 & \xi_1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & \xi_{n-1} \\ & & & & 1 \end{pmatrix}$$

Dabei soll gelten

$$X^H(T - \mu I)X = Q = \text{diag}(q_1, \dots, q_n) \quad q_i \in \mathbb{R}$$

d.h.

$$T - \mu I = (X^H)^{-1} Q X^{-1} = \begin{pmatrix} 1 & & & & \\ \bar{\xi}_1 & 1 & & & \\ & \bar{\xi}_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & \bar{\xi}_{n-1} & 1 \end{pmatrix} \begin{pmatrix} q_1 & & & & \\ & q_2 & & & \\ & & q_3 & & \\ & & & \ddots & \\ & & & & q_n \end{pmatrix} \begin{pmatrix} 1 & \xi_1 & & & \\ & 1 & \xi_2 & & \\ & & 1 & \ddots & \\ & & & \ddots & \xi_{n-1} \\ & & & & 1 \end{pmatrix}$$

Die q_i sind also die Quotienten aufeinanderfolgender Hauptabschnitts - Unterdeterminanten von $T - \mu I$. Hieraus ergeben sich die Gleichungen

$$\begin{aligned} q_1 &= \alpha_1 - \mu, & q_1 \xi_1 &= \beta_1 \quad (\Rightarrow q_1 \bar{\xi}_1 = \gamma_1 = \bar{\beta}_1) \\ q_2 + q_1 |\xi_1|^2 &= \alpha_2 - \mu, & q_2 \xi_2 &= \beta_2 \end{aligned}$$

allgemein

$$\begin{aligned} q_k + q_{k-1} |\xi_{k-1}|^2 &= \alpha_k - \mu, \\ q_k \xi_k &= \beta_k, \quad k = 1, \dots, n \end{aligned}$$

mit der Initialisierung

$$\xi_0 = 0 \quad \text{und} \quad q_0 = 1, \quad \text{sowie} \quad \beta_n = 0.$$

Weil $\beta_k \neq 0$ für $k = 1, \dots, n-1$, existiert ξ_k für $q_k \neq 0$ $k = 1, \dots, n-1$, d.h.

$$\xi_k = \beta_k / q_k \quad k = 1, \dots, n-1$$

Wird ein $q_k = 0$, so ersetzt man q_k durch $\epsilon \ll 1$ (d.h. α_k durch $\alpha_k + \epsilon$). Wegen (1.1) ändert dies die Eigenwerte nur um ϵ . Man rechnet also stets gemäß

$$\boxed{q_k = \alpha_k - \mu - |\beta_{k-1}|^2 / q_{k-1} \quad k = 1, \dots, n, \quad q_0 := 1, \quad \beta_0 := 0} \quad (1.4)$$

Dies ist nichts Anderes als der Gauss'sche Algorithmus ohne Zeilentausch, angewandt auf $T - \mu I$, mit den Pivots q_k , $k = 1, \dots, n$. Nach Satz 1.3.2 gilt für die Anzahlen

$$\#\{k : q_k < 0, \quad 1 \leq k \leq n\} = \#\{\lambda : \lambda \text{ Eigenwert von } T \text{ und } \lambda < \mu\}$$

Dieses Ergebnis kann man unmittelbar zu einer Intervallschachtelungsmethode zur Bestimmung jedes beliebigen Eigenwerts λ_j von T ausnutzen. Ausgehend von der Nummerierung $\lambda_1 \leq \dots \leq \lambda_n$ und z.B. der trivialen Einschließung

$$[a_0, b_0] := [-\|T\|_\infty, \|T\|_\infty],$$

die alle Eigenwerte von T enthält, (Satz 1.1.1) setzt man für $s = 0, 1, \dots$

$$\begin{aligned} \mu_s &:= (a_s + b_s) / 2 \\ m &:= \#\{q_k : q_k < 0, \text{ berechnet aus (1.4) mit } \mu = \mu_s\} \\ a_{s+1} &:= \begin{cases} a_s & \text{falls } m \geq j \\ \mu_s & \text{sonst} \end{cases} \\ b_{s+1} &:= \begin{cases} \mu_s & \text{falls } m \geq j \\ b_s & \text{sonst} \end{cases} \end{aligned}$$

Dann gilt $\lim_{s \rightarrow \infty} \mu_s = \lambda_j$

Dieses Verfahren, in der Literatur als Bisektionsverfahren bekannt, ist außerordentlich robust und sehr effizient.

Beispiel 1.3.1

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 3 & 2 & 0 \\ 0 & 2 & 5 & 3 \\ 0 & 0 & 3 & 7 \end{pmatrix}$$

$$\lambda_2 \in [1, 2] =: [a_0, b_0]$$

s	μ	q_1	q_2	q_3	q_4	m	
0	1.5	-0.500000	3.500000	2.357143	1.681818	1	$\Rightarrow \lambda_2 > 1.5$
1	1.75	-0.750000	2.583333	1.701613	-0.039100	2	$\Rightarrow \lambda_2 < 1.75$
2	1.625	-0.625000	2.975000	2.030462	0.942511	1	
3	1.6875	-0.687500	2.767045	1.866915	0.491712	1	
4	1.71875	-0.718750	2.672554	1.784555	0.237975	1	
5	1.734375	-0.734375	2.627327	1.743165	0.102603	1	
6	1.7421875	-0.742187	2.605181	1.722410	0.032577	1	
7	1.74609375	-0.746094	2.594220	1.712017	-0.003050	2	
8	1.744140625	-0.744141	2.599691	1.717215	0.014816	1	

□

Ist A allgemein nur hermitisch und besitzt A eine Zerlegung

$$A - \mu I = LDL^H \quad D = \text{diag}(\delta_i)$$

mit invertierbarem L , dann ist die Anzahl der negativen δ_i gleich der Anzahl der Eigenwerte von A , die kleiner sind als μ . Die Schwierigkeit in der Anwendung dieser Tatsache besteht bei allgemeinem A in der Tatsache, daß man die Elemente von L nicht aus der Rekursion für die δ eliminieren kann und diese Zerlegung numerisch instabil wird, wenn μ in der Nähe eines Eigenwertes einer Hauptuntermatrix von A liegt. Es gibt allerdings eine numerisch stabile Alternative dazu, die Bunch-Parlett-Zerlegung. Dies ist eine Zerlegung mit gleichnamigen Zeilen- und Spaltenvertauschungen

$$P^T A P = LDL^H$$

worin D nun eine Blockdiagonalmatrix mit Blöcken der Dimension 1 oder 2 ist. In der (besonders aufwendigen) Variante mit totaler Pivotsuche geht man so vor, daß man einen "normalen" 1×1 -Pivot wählt, wenn das maximale Element der Restmatrix auf der Diagonale zu finden ist und einen 2×2 -Pivot sonst. Im letzteren Fall führt man dann eine Blockelimination mit zwei Unbekannten aus gemäss dem Schema

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} I & O \\ A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & O \\ O & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix} \begin{pmatrix} I & A_{11}^{-1}A_{12} \\ O & I \end{pmatrix}$$

mit $A_{21} = A_{12}^T$. Es gibt auch numerisch stabile Varianten mit eingeschränkter Pivotwahl (von Bunch-Kaufman und Fletcher). Diese Zerlegung wird manchmal in der Optimierung benutzt, spielt aber in der Eigenwertberechnung keine Rolle.

Der obige Algorithmus hat bei einem Abbruch mit der Intervallbreite ε eine Aufwandskomplexität von $\mathcal{O}(n^2 \ln(\varepsilon))$ bei Bestimmung aller Eigenwerte und Eigenvektoren. Bei

sehr grossem n kann dies schon sehr aufwendig sein und es gibt eine Alternative, bekannt als “divide and conquer“. Die Idee besteht darin, die Eigenwertbestimmung rekursiv auf die von jeweils zwei kleineren Matrizen zurückzuführen. Es wird hier nur die wesentliche Idee dargestellt, die algorithmischen Details sind nicht trivial. Das Verfahren ist in der LAPACK-Bibliothek implementiert. Im Folgenden sind O eine Nullmatrix und o ein Nullvektor passender Dimension. Wir schreiben die Tridiagonalmatrix T als

$$T = \begin{pmatrix} T_1 & \begin{pmatrix} o & O \\ \beta & o^T \end{pmatrix} \\ \begin{pmatrix} o^T & \beta \\ o & O \end{pmatrix} & T_2 \end{pmatrix}$$

Dabei sind also T_1 und T_2 zwei Untermatrizen von T und β ein Ausserdiagonalelement. $\beta \neq 0$, sonst ist auf dieser Stufe nichts zu tun. (Das Eigenwertproblem zerfällt.) Zur Vereinfachung nehmen wir auch noch an, T sei reell. Wir bilden nun

$$\hat{T}_1 = T_1 - \begin{pmatrix} o & O \\ o^T & \beta \end{pmatrix}, \quad \hat{T}_2 = T_2 - \begin{pmatrix} \beta & o^T \\ o & O \end{pmatrix}.$$

Dann wird

$$T = \begin{pmatrix} \hat{T}_1 & O \\ O & \hat{T}_2 \end{pmatrix} + \begin{pmatrix} O & o & o & O \\ o^T & \beta & \beta & o^T \\ o^T & \beta & \beta & o^T \\ O & o & o & O \end{pmatrix}$$

d.h. die Summe einer Blockdiagonalmatrix und einer Matrix vom Rang 1. Ist das Eigenwert/Eigenvektorproblem von \hat{T}_1 und \hat{T}_2 gelöst, also

$$\hat{T}_i = U_i \Lambda_i U_i^T, \quad i = 1, 2$$

mit unitären U_i und diagonalen Λ_i , dann gilt

$$T = \text{blockdiag}(U_1, U_2) (\text{blockdiag}(\Lambda_1, \Lambda_2) + \beta z z^T) \text{blockdiag}(U_1^T, U_2^T)$$

mit

$$z^T = (u_1^T, u_2^T)$$

wo u_1^T die letzte Zeile von U_1 und u_2^T die erste Zeile von U_2 ist. Dies ist eine Ähnlichkeitstransformation von T und es genügt deshalb, das Eigenwertproblem der inneren Matrix noch zu lösen, um das Gesamtproblem zu lösen. Das ist aber das Eigenwert/Eigenvektorproblem einer Diagonalmatrix nach einer Rang-1-Änderung. Wir schreiben dieses Problem nun als

$$(D + \beta z z^T)w = \mu w \quad \text{mit } w \neq 0 \tag{1.5}$$

Multiplikation mit w^T ergibt

$$w^T D w + \beta (z^T w)^2 = \mu \|w\|^2$$

1.4. BESTIMMUNG DER EIGENVEKTOREN EINER HERMITISCHEN DREIBANDMATRIX 27

Für $z^T w = 0$ ist also μ ein Eigenwert und w ein Eigenvektor von D (diagonal), also ein Koordinateneinheitsvektor und dann muss auch z selbst die entsprechende Komponente null haben. Zu allen Komponenten von z , die null sind, ist also μ gleich dem entsprechenden Diagonalelement von D und w der entsprechende Koordinateneinheitsvektor. Deshalb können wir für das Folgende annehmen, daß alle Komponenten von z ungleich null sind und deshalb μ keinem Diagonalelement von D gleich ist. Wir schreiben (1.5) um zu

$$(D - \mu I)w + \beta(z^T w)z = 0$$

also

$$w + (D - \mu I)^{-1}\beta(z^T w)z = 0$$

oder

$$z^T w + z^T(D - \mu I)^{-1}z\beta(z^T w) = 0$$

und wegen $z^T w \neq 0$

$$1 + \beta \sum_i \frac{z_i^2}{d_{ii} - \mu} = 0 .$$

Dies ist ein reelles Nullstellenproblem für eine rationale Funktion mit den Polstellen d_{ii} , das mit einer geeigneten Modifikation des gedämpften Newtonverfahrens gelöst werden kann. Es sei hier daran erinnert, daß die d_{ii} paarweise verschieden sein müssen, weil wir von einer hermiteschen Tridiagonalmatrix mit nichtverschwindenden Nebendiagonalelementen ausgegangen sind. Nach der Bestimmung der μ findet man die Eigenvektoren wie im Folgenden beschrieben und hat damit die Problemlösung auf einer Rekursionsstufe vollständig vorliegen. Kritische Punkte bei der Implementierung sind die Entscheidung, ob ein z_i als null anzusehen ist, was als "vernachlässigbar kleines Ausserdiagonalelement" zu gelten hat und wie das Nullstellenproblem zuverlässig und genau zu lösen ist.

1.4 Bestimmung der Eigenvektoren einer hermiteschen Dreibandmatrix

Wir behandeln hier die Frage, wie man das fast singuläre (oder im Glücksfall sogar exakt singuläre) System

$$(A - \lambda I)x = 0, \quad x \neq 0$$

mit einer "guten" Eigenwertnäherung λ behandeln soll. Wir setzen im Folgenden voraus, T sei eine nichtzerfallende hermitesche Dreibandmatrix (d.h. $\gamma_i \neq 0 \quad i = 1, \dots, n-1$ (vgl. 1.3)) und μ eine bis auf Maschinengenauigkeit bestimmte Eigenwertnäherung für einen Eigenwert λ von T (z.B. mit dem Bisektionsverfahren bestimmt bis $\mu_s \leq a_s$ oder $\mu_s \geq b_s$ aufgrund von Rundungseffekten). Wir wissen, daß für beliebiges λ $\text{Rang}(T - \lambda I) \geq n - 1$ und daß keines der Subdiagonalelemente von T verschwindet, so daß die Dreieckszerlegung von $T - \mu I$ mit Zeilenvertauschung vollständig durchführbar ist. Bei Spaltenpivotsuche (diese ist hier unerlässlich) gilt für die Dreieckszerlegung mit der Permutationsmatrix P deshalb

$$P(T - \mu I) = L \cdot R$$

$|\rho_{jj}| \geq |\beta_j|$, $j = 1, \dots, n-1$. Ist $\mu = \lambda_j$, dann wird bei rundungsfehlerfreier Rechnung $\rho_{nn} = 0$ und eine Lösung x von $Rx = 0$ mit $\xi_n = 1$ wird Eigenvektor von T . In der Praxis kann man aber keineswegs immer ein "kleines" ρ_{nn} beobachten, auch wenn μ eine sehr gute Eigenwertnäherung ist. Folgender Satz gibt Auskunft, wie man dennoch eine gute Eigenvektornäherung finden kann, wenn nur die Eigenwertnäherung brauchbar ist:

Satz 1.4.1 Sei T eine hermitesche $n \times n$ Dreibandmatrix, $P(T - \mu I) = LR$ eine mit Spaltenpivotsuche durchgeführte Dreieckszerlegung, μ eine Eigenwertnäherung für den Eigenwert λ_j von T mit

$$\mu = \lambda_j + \vartheta\delta, \quad \text{wo } |\vartheta| \leq 1, \quad \delta \text{ hinreichend klein.}$$

Alle Subdiagonalelemente von T seien von null verschieden. Dann existiert (mindestens) ein $i \in \{1, \dots, n\}$, so daß die Lösung x_i von

$$Rx_i = e_i \rho_{ii}, \quad x_i = (\xi_{i1}, \dots, \xi_{in})^T$$

(mit $\xi_{nn} := 1$ falls $i = n$ und $\rho_{nn} = 0$)

$$\frac{x_i}{\|x_i\|_2} = \alpha u_j + d \quad \text{mit} \quad \begin{array}{l} \|d\|_2 \leq \frac{\sqrt{2}n^{5/2}\delta}{\min\{|\lambda_i - \lambda_j|, i \neq j\}} + \mathcal{O}(\delta^2) \\ |\alpha| = 1 \end{array} \quad (1.6)$$

erfüllt, wobei (u_1, \dots, u_n) ein orthonormiertes Eigenvektorsystem von T bezeichnet und $Tu_i = \lambda_i u_i$, $i = 1, \dots, n$

□

<<

Beweis: Setze $y_i := \rho_{ii} P L^T e_i$, $U = (u_1, \dots, u_n)$.

Dann wird mit $T = U \Lambda U^H$, $\Lambda = \text{diag}(\lambda_i)$

$$(T - \mu I)x_i = P^T L R x_i = \rho_{ii} P^T L e_i = y_i$$

Im Falle $\rho_{nn} = 0$ ist nichts mehr zu zeigen. Sei $\rho_{nn} \neq 0$ und

$$x'_i := \frac{1}{\rho_{ii}} x_i, \quad y'_i := \frac{1}{\rho_{ii}} y_i, \quad i = 1, \dots, n$$

und x'_i, y'_i nach den u_1, \dots, u_n entwickelt:

$$x'_i = \sum_{k=1}^n \tilde{\xi}_{ik} u_k, \quad y'_i = \sum_{k=1}^n \tilde{\eta}_{ik} u_k.$$

Dann besteht der Zusammenhang

$$(\lambda_k - \mu) \tilde{\xi}_{ik} = \tilde{\eta}_{ik} \quad i, k = 1, \dots, n.$$

Wir benutzen die Cauchy-Schwarzsche Ungleichung:

$$\|U^H P^T L e_i\|_\infty = \max\{|e_k^T U^H P^T L e_i|\} \leq \|e_k^T U^H P^T\|_2 \|L e_i\|_2$$

Es gilt (da die Elemente von $L e_i$ betragsmäßig kleinergleich 1 sind)

$$\begin{pmatrix} \tilde{\eta}_{i1} \\ \vdots \\ \tilde{\eta}_{in} \end{pmatrix} = U^H P^T L e_i \Rightarrow |\tilde{\eta}_{ik}| \leq \sqrt{2} \quad i, k = 1, \dots, n$$

weil L pro Spalte höchstens zwei Elemente ungleich null besitzt und weil $\|Ue_j\|_\infty \geq 1/\sqrt{n}$ und $\|L^{-T}\|_\infty \leq n$

$$\begin{aligned} \max_{i \in \{1, \dots, n\}} |\tilde{\eta}_{ij}| &= \max_{i \in \{1, \dots, n\}} |e_j^T U^H P^T L e_i| \\ &= \|L^T P U e_j\|_\infty \geq \|P U e_j\|_\infty / \|L^{-T}\|_\infty \geq 1/n^{3/2} \end{aligned}$$

(P ist eine Permutation und das grösste Element eines Vektors der Länge 1 hat den Betrag mindestens $1/\sqrt{n}$.) Nun ist aber mit

$$\begin{aligned} \sigma &:= \min\{|\lambda_i - \lambda_j| : i \neq j\} > 0 \\ |\tilde{\xi}_{ik}| &= \frac{|\tilde{\eta}_{ik}|}{|\lambda_k - \lambda_j - \delta\vartheta|} \leq \frac{\sqrt{2}}{\sigma - \delta} \quad \text{für } k \neq j, \quad \forall i \end{aligned}$$

(für δ hinreichend klein ist $\sigma - \delta > 0$), während

$$|\tilde{\xi}_{ij}| = \frac{|\tilde{\eta}_{ij}|}{|\delta\vartheta|} \geq \frac{1}{n^{3/2}\delta} \quad \text{für } i \text{ geeignet.}$$

Für dieses i wird

$$\begin{aligned} \|x'_i\|_2 &= \left(\sum_{k=1}^n |\tilde{\xi}_{ik}|^2 \right)^{1/2} = |\tilde{\xi}_{ij}| \left(1 + \sum_{\substack{k=1 \\ k \neq j}}^n \frac{|\tilde{\xi}_{ik}|^2}{|\tilde{\xi}_{ij}|^2} \right)^{1/2} \\ &= |\tilde{\xi}_{ij}| (1 + \vartheta_{ij}) \end{aligned}$$

mit

$$0 \leq \vartheta_{ij} \leq 2n^3\delta^2/(\sigma - \delta)^2 .$$

(Man beachte dazu $1 \leq \sqrt{1 + \xi} \leq 1 + \xi$ für $0 \leq \xi \leq 1$.)

Somit

$$\begin{aligned} \frac{x_i}{\|x_i\|_2} = \frac{x'_i}{\|x'_i\|_2} &= \frac{\tilde{\xi}_{ij}}{|\tilde{\xi}_{ij}|(1 + \vartheta_{ij})} u_j + \sum_{\substack{k=1 \\ k \neq j}}^n \frac{\tilde{\xi}_{ik}}{|\tilde{\xi}_{ij}|(1 + \vartheta_{ij})} u_k \\ &= \underbrace{\frac{\tilde{\xi}_{ij}}{|\tilde{\xi}_{ij}|}}_{\alpha} u_j + d \end{aligned}$$

mit

$$\begin{aligned} \|d\|_2 &\leq 1 - \frac{1}{1 + \frac{2n^3\delta^2}{(\sigma - \delta)^2}} + \frac{(n-1)\sqrt{2}n^{3/2}\delta}{\sigma - \delta} \\ &\leq \frac{\sqrt{2}n^{5/2}\delta}{\sigma} + \mathcal{O}(\delta^2) \end{aligned}$$

□

>>

Bemerkung 1.4.1 Aufgrund der Herleitung ist klar, daß der Faktor $n^{5/2}$ in der Abschätzung (1.6) vergleichsweise pessimistisch ist. Viel eher kann man hierfür den Faktor 1 annehmen. Man kann zeigen, daß $i = n$ geeignet ist, wenn mit der Partitionierung

$$R = \left(\begin{array}{c|c} R_{11} & r \\ \hline 0 & \rho_{nn} \end{array} \right)$$

$\|R_{11}^{-1}r\|_2$ “klein” (z.B. $\leq n$) ist. In der Praxis geht man so vor, daß man

$$Rx = \rho_{n,n}e_n$$

löst, dies als Startwert für die gebrochene Iteration (s.h.) benutzt und die Rechnung nach einem weiteren Schritt abbricht, nachdem zum erstenmal

$$\|x_i\|_\infty \geq \frac{1}{100n\varepsilon} |\rho_{n,n}| \quad (1.7)$$

galt. Der Faktor $\frac{1}{100n}$ ist dabei ein (etwas willkürlich) gewählter Sicherheitsfaktor (der wahre Fehler in μ ist ja unbekannt!). Falls der Test (1.7) nach drei Schritten nicht erfüllt ist, betrachtet man auch die Eigenwertnäherung μ als “zu schlecht”. Man kann ferner zeigen, daß Abänderungen der Elemente von T in der Größenordnung $\varepsilon\|T\|_2$ Fehler in den Eigenvektoren von der Größenordnung

$$\frac{n\varepsilon\|T\|_2}{\min_{i \neq j} |\lambda_i - \lambda_j|}$$

zur Folge haben können (Satz 1.1.8). Satz 1.3.2 stellt also ein ganz ausgezeichnetes Resultat dar. Man kann weiter zeigen, daß auch die Rundungsfehler bei der Dreieckszerlegung von $P(T - \mu I)$ und bei der Lösung von (1.6) die Aussage des Satzes nicht wesentlich ändern. \square

Bemerkung 1.4.2 Es besteht kein quantitativer Zusammenhang zwischen der Größe der Außen-diagonalelemente (d.h. $\min_i |\beta_i|$) und $\sigma = \min_{i \neq j} |\lambda_i - \lambda_j|$.

Wilkinson hat ein Beispiel einer 21×21 -Matrix angegeben mit $\beta_i = 1 \quad \forall i$ und σ in der Größenordnung von 10^{-9} . Bei Matrizen mit solch “fast zusammenfallenden” Eigenwerten ist die Bestimmung eines einzelnen Eigenvektors ganz außerordentlich schwierig. \square

1.5 Direkte Iteration nach v. Mises und Verfahren von Wielandt

Ziel der in diesem Abschnitt beschriebenen Verfahren ist es, zunächst eine Eigenvektornäherung zu finden. Wie man dazu dann eine Eigenwertnäherung bekommen kann, haben wir bereits in Satz 1.1.4 gesehen. Die (für die Praxis allerdings unbrauchbare) Grundversion der **direkten Iteration von v. Mises** lautet:

1. Wähle $x_0 \in \mathbb{C}^n$, $x_0 \neq 0$ geeignet

2. Für $k = 0, 1, 2, \dots$ setze

$$x_{k+1} = Ax_k$$

Satz 1.5.1 *Es sei $A \in \mathbb{C}^{n \times n}$ diagonalähnlich und $Au_i = \lambda_i u_i$, $U = (u_1, \dots, u_n)$ ein vollständiges Eigenvektorsystem von A . Ferner gelte*

$$x_0 = \sum_{i=1}^n \xi_i u_i \quad \text{mit } \xi_1 \neq 0$$

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$$

Dann gilt für die oben konstruierte Vektorfolge $\{x_k\}$

(i) $x_k = \xi_1 \lambda_1^k (u_1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))$

(ii) $R(x_k; A) = \lambda_1 (1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))$

□

Beweis: Man beachte, daß

$$\xi_1 = v_1^H x_0$$

gilt, wo v_1^H der biorthonormierte Linkseigenvektor zu λ_1 ist, also die erste Zeile von $(u_1, \dots, u_n)^{-1}$.

$$x_k = A^k x_0 = (u_1, \dots, u_n) \operatorname{diag}(\lambda_1^k, \dots, \lambda_n^k) (u_1, \dots, u_n)^{-1} (u_1, \dots, u_n) \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix}$$

$$= \sum_{i=1}^n \lambda_i^k \xi_i u_i = \xi_1 \lambda_1^k (u_1 + \sum_{i=2}^n \underbrace{\frac{\xi_i}{\xi_1} (\frac{\lambda_i}{\lambda_1})^k}_{|\cdot| \leq |\frac{\lambda_2}{\lambda_1}|} u_i)$$

$$R(x_k; A) = \frac{x_k^H x_{k+1}}{x_k^H x_k}$$

$$= \frac{\bar{\xi}_1 \bar{\lambda}_1^k (u_1^H + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k)) \xi_1 \lambda_1^{k+1} (u_1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^{k+1}))}{\bar{\xi}_1 \bar{\lambda}_1^k (u_1^H + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k)) \xi_1 \lambda_1^k (u_1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))}$$

$$= \lambda_1 (1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))$$

□

Bemerkung 1.5.1

- (a) Für $|\lambda_1| \neq 1$ führt die praktische Durchführung der obigen “Grundversion” schnell zu Exponentenüber- oder -unterlauf. Man rechnet stattdessen mit Normierung nach

$$\begin{aligned}\tilde{x}_{k+1} &:= Ax_k \\ \varrho_k &= x_k^H \tilde{x}_{k+1} \\ x_{k+1} &= \tilde{x}_{k+1} / \|\tilde{x}_{k+1}\|\end{aligned}$$

mit $x_0 : \|x_0\| = 1$ geeignet gewählt. D.h. ϱ_k ist der Rayleighquotient zu x_k . Es gilt dann

$$x_k = \vartheta_k(u_1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k)) \quad |\vartheta_k| = 1$$

und

$$\varrho_k = \lambda_1(1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k))$$

und im hermiteschen Fall sogar

$$\varrho_k = \lambda_1(1 + \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^{2k})) .$$

Normiert man x_k auf $(x_k)_j := 1$ für eine Komponente j mit $(u_1)_j \neq 0$, dann ist $\{x_k\}$ konvergent gegen ein Vielfaches von u_1 .

- (b) Satz 1.5.1 gilt entsprechend für $\lambda_1 = \lambda_2 = \dots = \lambda_r$,
 $|\lambda_1| > |\lambda_{r+1}| \geq \dots \geq |\lambda_n|$, falls $x_0 = \sum_{i=1}^n \xi_i u_i$ und $\sum_{i=1}^r |\xi_i| \neq 0$.
 Man erhält aber nur ein Element aus dem Eigenraum!

- (c) Die Voraussetzung $\xi_1 \neq 0$ bzw. $\sum_{i=1}^r |\xi_i| \neq 0$ wird in der Praxis durch eingeschleppte Rundungsfehler stets erfüllt, auch wenn x_0 ungeeignet war.

- (d) Man kann auf die Voraussetzung “A diagonalähnlich” verzichten. Dann tritt aber an die Stelle des Fehlerterms $\mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k)$, die ja wenigstens noch Konvergenz mit der Konvergenzgeschwindigkeit der geometrischen Reihe sicherstellt, ein Term $\mathcal{O}(\frac{1}{k})$, die Konvergenzgeschwindigkeit ist dann also sublinear.

- (e) Bei verschiedenen betragsgleichen und betragsdominanten Eigenwerten von A (z.B. betragsdominanter komplexer Eigenwert einer reellen Matrix!) tritt keine Konvergenz ein. Man kennt aber Verallgemeinerungen des Verfahrens auch auf diesen Fall. (simultane Vektoriteration, vgl. Literaturhinweis am Ende dieses Kapitels)

□

Beispiel 1.5.1 Wir betrachten die direkte Iteration für eine symmetrische 4×4 -Matrix mit den Eigenvektoren

$$\sqrt{\frac{2}{n+1}}(\sin(\frac{ij\pi}{n+1}))_{i=1,\dots,n}$$

mit den folgenden Eigenwerten:

1. 10,5,1,10
2. 10,5,1,-10
3. 10,5,1,-9.99
4. 10,5,1,-9.9
5. 10,5,1,-9

Wir versuchen, den jeweils dominanten Eigenwert mit einer Genauigkeitsforderung 10^{-6} mit dem Startvektor $(1, 0, 0, 0)^T$ zu finden. Es ergibt sich folgendes:

Fall	λ_2/λ_1	Schritte	Bemerkung
1.	$5/10 = 0.5$	11	Doppelter Eigenwert
2.			Keine Konv.: zwei betragsgrößte EW
3.	$9.99/10 = 0.999$	7252	
4.	$9.9/10 = 0.99$	723	Faktor 10 zum Fall 3)
5.	$9/10 = 0.9$	70	Faktor 10 zum Fall 4)

Sowohl das Konvergenzverhalten ($\mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^{2k})$) als auch die geforderten Voraussetzungen für das Verfahren lassen sich durch die numerischen Resultate voll bestätigen:

$$0.5^{22} = 2.3 \cdot 10^{-7}, \quad 0.999^{14504} = 4.99 \cdot 10^{-7}, \quad 0.99^{1446} = 4.88 \cdot 10^{-7}, \quad 0.9^{140} = 3.9 \cdot 10^{-7}.$$

□

Die Konvergenzgeschwindigkeit der direkten Iteration hängt entscheidend von dem Quotienten $|\lambda_2/\lambda_1| < 1$ ab. Seien $\lambda_1, \dots, \lambda_n$ die Eigenwerte von A . μ sei eine Eigenwertnäherung für λ_i und es gelte

$$0 < |\lambda_i - \mu| < |\lambda_j - \mu| \quad \forall j \in \{1, \dots, n\} \setminus \{i\}$$

Dann ist $A - \mu I$ regulär und für die Eigenwerte $\tau_k = \frac{1}{\lambda_k - \mu}$ von $(A - \mu I)^{-1}$ gilt

$$|\tau_i| > |\tau_j| \quad \forall j \in \{1, \dots, n\} \setminus \{i\}$$

Ferner wird $\max_{j \neq i} |\tau_j|/|\tau_i|$ um so kleiner, je besser die Eigenwertnäherung war. Die direkte Iteration für $(A - \mu I)^{-1}$ führt dann also zu schneller Konvergenz. Dies ist die dem Verfahren von Wielandt, der sogenannten “gebrochenen” oder “inversen” Iteration, zugrundeliegende Idee. Entscheidend für die praktische Brauchbarkeit des Verfahrens ist

es, daß die Inverse $(A - \mu I)^{-1}$ nicht explizit gebildet zu werden braucht. Vielmehr löst man pro Schritt das lineare Gleichungssystem

$$\begin{aligned}(A - \mu I)\tilde{x}_{k+1} &= x_k \\ x_{k+1} &:= \tilde{x}_{k+1}/\|\tilde{x}_{k+1}\|\end{aligned}$$

was ohne großen Aufwand möglich ist, wenn man (ein für allemal) eine Dreieckszerlegung von $(A - \mu I)$ (zumindest mit Spaltenpivotstrategie!) berechnet hat: Mit

$$P(A - \mu I)Q = L \cdot R$$

rechnet man dann gemäß

$$\begin{aligned}z_k &:= Px_k \\ Lv_k &= z_k \\ R w_k &= v_k \\ Q^T \tilde{x}_{k+1} &= w_k\end{aligned}$$

Man könnte mit der neu errechneten Eigenvektornäherung \tilde{x}_{k+1} auch eine neue Eigenwertnäherung definieren, eine neue Dreieckszerlegung berechnen usw. Wegen des hohen Rechenaufwandes lohnt sich dies aber in der Regel nicht.

Beispiel 1.5.2 *Gesucht ist der kleinste Eigenwert λ_3 der Matrix*

$$\begin{pmatrix} 30 & 2 & 0 \\ 2 & 20 & 1 \\ 0 & 1 & 10 \end{pmatrix}.$$

Nach dem Kreisesatz von Gerschgorin liegt der kleinste Eigenwert in einem Kreis vom Radius 1 um 10 und ist isoliert. Wir benutzen deshalb den Shift $\mu = 10$. Als Startvektor nehmen wir $x_0 = (0, 0, 1)^T$. Mit dem Shift $\mu = 10$ erhält man die Matrix

$$\tilde{A} = A - \mu I = A - 10I = \begin{pmatrix} 20 & 2 & 0 \\ 2 & 10 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Nun muß das Gleichungssystem $(A - \mu I)x_1 = \tilde{A}x_1 = x_0$ gelöst werden:

$$\begin{array}{ccc|ccc|ccc|ccc|ccc} 20 & 2 & 0 & 0 & 1 & 5 & 1/2 & 0 & 1 & 5 & 1/2 & +0 & 1 & 0 & 0 & -0.1 \\ 2 & 10 & 1 & 0 & \mapsto & 0 & 1 & 0 & 1 & \mapsto & 0 & 1 & 0 & +1 & \mapsto & 0 & 1 & 0 & +1 \\ 0 & 1 & 0 & 1 & 0 & -98 & -10 & 0 & 0 & 0 & 1 & -9.8 & 0 & 0 & 1 & -9.8 \end{array}$$

Damit lautet $x_1 = (-0.1, 1, -9.8)^T$. Mit dem RAYLEIGH-Quotienten erhält man zuerst eine Näherung für den Eigenwert $\sigma_i = \frac{1}{\lambda_i - \mu}$ von $\tilde{A}^{-1} = (A - \mu I)^{-1}$:

$$\sigma_3 \approx R(x_0; \tilde{A}^{-1}) = \frac{x_0^T \tilde{A}^{-1} x_0}{x_0^T x_0} = \frac{x_0^T x_1}{x_0^T x_0} = \frac{-9.8}{1} = -9.8$$

Wegen $\lambda_3 = \mu + \frac{1}{\sigma_3}$ liefert dies als neue Näherung für λ_3 :

$$\lambda_3 \approx \mu + \frac{1}{R(x_0; \tilde{A}^{-1})} = \mu + \frac{x_0^T x_0}{x_0^T x_1} = 10 + \frac{1}{-9.8} \approx 9.89796$$

<<

Bezüglich der Beschaffung eines geeigneten Startvektors gibt der folgende Satz Auskunft:

Satz 1.5.2 Sei $A \in \mathbb{C}^{n \times n}$ diagonalähnlich, $Au_i = \lambda_i u_i \quad i = 1, \dots, n$,
 $\|u_i\|_2 = 1 \quad \forall i$ worin $U = (u_1, \dots, u_n)$ ein vollständiges Eigenvektorsystem von A bezeichnet
und

$$P(A - \mu I)Q = LR$$

P, Q Permutationsmatrizen,

L untere Dreiecksmatrix mit Diagonale 1 und Elementen von Betrag ≤ 1

R obere Dreiecksmatrix

Ferner sei s definiert durch $|\rho_{ss}| = \min_i |\rho_{ii}|$ und $\hat{R} := \text{diag}(\rho_{11}^{-1}, \dots, \rho_{nn}^{-1})R$. Dann gilt, falls μ kein Eigenwert ist

$$\begin{aligned} \min_j |\lambda_j - \mu| &\leq \sqrt{n} |\rho_{ss}| \text{cond}_{\|\cdot\|_2}(U) \\ &\leq \min_j |\lambda_j - \mu| \|L^{-1}\|_2 \|\hat{R}^{-1}\|_2 \text{cond}_{\|\cdot\|_2}(U) \sqrt{n} \end{aligned}$$

Definiert man x_1 durch

$$RQ^T x_1 = \rho_{ss} e_s \quad (\hat{=} x_0 := \rho_{ss} P^T L e_s) \quad (1.8)$$

und den Index k durch $|\xi_k| = \max_i |\xi_i|$, wo

$$x_1 = \sum_{i=1}^n \xi_i u_i$$

dann gilt für den zugehörigen Eigenwert λ_k die Abschätzung

$$|\lambda_k - \mu| \leq n^{3/2} |\rho_{ss}| \text{cond}_{\|\cdot\|_2}(U)$$

(sind also die verschiedenen Eigenwerte von A hinreichend separiert im Vergleich zu $\min_j |\lambda_j - \mu|$, dann wird $|\lambda_k - \mu| = \min_j |\lambda_j - \mu|$ und somit x_1 eine zur inversen Iteration sehr gut geeignete Startnäherung.)

□

Beweis: Ändert man den Wert ρ_{ss} in der Dreieckszerlegung ab in 0, dann ist dies äquivalent zur Abänderung von

$$A - \mu I \quad \text{in} \quad B := A - \mu I - \rho_{ss} P^T L e_s e_s^T Q^T$$

und B wird singulär. Satz 1.1.8 liefert die Abschätzung

$$\begin{aligned} \left| \underbrace{0}_{\lambda(B)} - \underbrace{(\lambda_i - \mu)}_{\lambda(A - \mu I)} \right| &\leq \text{cond}_{\|\cdot\|_2}(U) |\rho_{ss}| \|P^T L e_s e_s^T Q\|_2 \\ &\leq \sqrt{n} |\rho_{ss}| \text{cond}_{\|\cdot\|_2}(U) \end{aligned}$$

für ein geeignetes i . Sei nun j_0 definiert durch $|\lambda_{j_0} - \mu| = \min_i |\lambda_i - \mu|$.

Es gilt wegen $\|u_{j_0}\|_2 = 1$

$$(A - \mu I)^{-1} u_{j_0} = \frac{1}{\lambda_{j_0} - \mu} u_{j_0}$$

$$\Rightarrow \frac{1}{|\lambda_{j_0} - \mu|} \leq \|(A - \mu I)^{-1}\|_2 = \|(P^T L R Q^T)^{-1}\|_2 \leq \|L^{-1}\|_2 \|\hat{R}^{-1}\|_2 \frac{1}{|\rho_{ss}|}$$

Somit

$$|\rho_{ss}| \leq \min_j |\lambda_j - \mu| \|L^{-1}\|_2 \|\hat{R}^{-1}\|_2$$

Nach Definition von x_1 und des Index k ist

$$1 \leq \|x_1\|_2 \leq n |\xi_k|$$

Weiterhin gilt

$$x_0 = \sum_{i=1}^n \xi_i (\lambda_i - \mu) u_i$$

$$|\xi_k| |\lambda_k - \mu| \leq \|U^{-1} x_0\|_2 = \|U^{-1} \rho_{ss} P^T L e_s\|_2 \leq |\rho_{ss}| \|U^{-1}\|_2 \sqrt{n}$$

$$|\lambda_k - \mu| \leq n^{3/2} |\rho_{ss}| \|U^{-1}\|_2 \leq n^{3/2} |\rho_{ss}| \text{cond}_{\|\cdot\|_2}(U)$$

□

Bemerkung 1.5.2 Wurde die Dreieckszerlegung mit vollständiger Pivotwahl durchgeführt, dann gilt für die Elemente von \hat{R} : $\hat{\rho}_{ii} = 1, |\hat{\rho}_{ij}| \leq 1$.

In diesem Fall kann $\|L^{-1}\|_2 \|\hat{R}^{-1}\|_2$ als eine Funktion von n allein abgeschätzt werden (Üb.). Ist $\mu = \lambda_j$ für ein j , dann wird $\rho_{ss} = 0$ und x_1 selbst wird zugehöriger Eigenvektor. Man erkennt, daß ρ_{ss} linear mit $\min_j |\lambda_j - \mu|$ gegen null geht. Dennoch treten bei obiger Bestimmung von x_1 keine numerischen Probleme auf. Man kann auch hier einen zu Satz 1.4.1 analogen Satz formulieren, d.h. ist $\mu - \lambda_j \approx f(n)\vartheta\varepsilon$, dann ist bereits $x_1 \approx u_j$ bis auf einen Fehler der Ordnung ε , bei dem als Fehlerverstärkungsfaktor allerdings (u.U. große) Terme analog Satz 1.1.8 auftreten. □

>>

Ist A eine nichtzerfallende obere Hessenbergmatrix, dann ist der erste Koordinateneinheitsvektor stets ein geeigneter Startvektor für das von Mises bzw. Wielandtverfahren, weil kein Linkseigenvektor von A die erste Komponente =0 haben kann, wie man leicht durch Widerspruchsbeweis verifiziert.

Das Wielandt-Verfahren wird in der Praxis benutzt, um zu bereits mit hoher Genauigkeit gefundenen Eigenwerten die entsprechenden Eigenvektoren zu bestimmen, allerdings meist in der rudimentären Form, nur x_1 gemäß Satz 1.5.2 zu bestimmen, und nur 1 bis 2 Iterationsschritte folgen zu lassen, und dies auch nur für Hessenbergmatrizen.

1.5.1 Die simultane Vektoriteration

Beim Verfahren von v. Mises bzw. Wielandt wird stets ein einzelner Eigenvektor bzw. Eigenwert bestimmt. In den Anwendungen tritt das Problem der Eigenwertbestimmung aber in der Regel in der Form auf, daß die p kleinsten Eigenwerte (typisch $1 \leq p \leq 30$) mit den Eigenvektoren zu bestimmen sind. Im Prinzip könnte man nun das Wielandt-Verfahren p -mal mit p verschiedenen geeigneten Shifts μ anwenden. Zusätzlich zu dem damit verbundenen Aufwand kommt aber erschwerend hinzu, daß häufig, insbesondere bei Problemen wie der Balken- oder Plattenbiegung, extrem dicht beieinanderliegende Eigenwerte auftreten, so daß man nicht sicher sein kann, ohne aufwendige Zusatzmaßnahmen p linear unabhängige Eigenvektoren zu finden. Es ist deshalb besser, p Eigenvektoren simultan anzunähern und deren Unabhängigkeit algorithmisch zu erzwingen. Im Folgenden beschränken wir uns auf den für die Praxis wichtigsten Fall eines allgemeinen Eigenwertproblems

$$A x = \lambda B x$$

mit den reell-symmetrischen und positiv definiten Matrizen A , B . Mit Hilfe der Cholesky-Zerlegung von B :

$$B = L L^T$$

könnte man dieses Problem im Prinzip auf ein gewöhnliches Eigenwertproblem transformieren:

$$L^{-1} A (L^T)^{-1} L^T x = \lambda L^T x$$

oder

$$C y = \lambda y$$

mit

$$\begin{aligned} C &= L^{-1} A (L^T)^{-1}, \\ y &= L^T x. \end{aligned}$$

Wegen $(L^T)^{-1} = (L^{-1})^T$ ist dabei C wieder reell, symmetrisch und positiv definit. Also ist C reell-orthogonal diagonalisierbar und die Eigenvektoren y_k können paarweise orthogonal gewählt werden:

$$y_k^T y_i = 0 \quad \text{für } k \neq i.$$

Die Eigenvektoren $x_k = (L^T)^{-1} y_k$ des Ausgangsproblems sind deshalb orthogonal bezüglich des Skalarprodukts $\langle u, v \rangle = u^T B v$ wegen

$$\begin{aligned} x_k^T B x_i &= y_k^T ((L^T)^{-1})^T L L^T (L^T)^{-1} y_i \\ &= y_k^T L^{-1} L L^T (L^T)^{-1} y_i = y_k^T y_i. \end{aligned}$$

Bei der Berechnung der y_j stellt man aber C nicht explizit auf, außer wenn n recht klein ist, sondern benutzt die Darstellung von C nur implizit.

Es sollen nun die zu den p kleinsten Eigenwerten von C gehörenden Eigenvektoren y_1, \dots, y_p angenähert werden. Dazu wird im Prinzip die inverse Iteration nach

Wielandt angewandt, ausgehend von p normierten paarweise orthogonalen Vektoren $y_1^{(0)}, \dots, y_p^{(0)}$:

$$C(\hat{y}_1^{(1)}, \dots, \hat{y}_p^{(1)}) = (y_1^{(0)}, \dots, y_p^{(0)}).$$

Die "Eigenvektornäherung" $\hat{y}_j^{(1)}$ erhält man durch die Rechenschritte

$$\begin{aligned} \hat{x}_j^{(1)} &= L y_j^{(1)}, \\ A \hat{z}_j^{(1)} &= \hat{x}_j^{(1)}, \quad (\text{Gleichungssystem lösen}), \\ \hat{y}_j^{(1)} &= L^T \hat{z}_j^{(1)}. \end{aligned}$$

Die $\hat{y}_j^{(1)}$ sind nicht mehr paarweise orthogonal. Aus diesen p Vektoren soll nun eine Orthonormalbasis $y_1^{(1)}, \dots, y_p^{(1)}$ konstruiert werden. Ist $Q_1 \in \mathbb{R}^{n \times p}$ eine spezielle Orthonormalbasis des von $\hat{y}_1^{(1)}, \dots, \hat{y}_p^{(1)}$ aufgespannten Raumes, dann gilt für jede andere Orthonormalbasis die Darstellung $Q_1 V_1$ mit einer orthonormalen $p \times p$ -Matrix V_1 . Sei also

$$\hat{Y}^{(1)} = (\hat{y}_1^{(1)}, \dots, \hat{y}_p^{(1)}) = Q_1 R_1$$

eine QR -Zerlegung mit

$$Q_1^T Q_1 = I_p, \quad R_1 = p \times p \text{ obere Dreiecksmatrix.}$$

Q_1 kann z.B. aus den ersten p Zeilen bei der Householder-Orthogonalisierung von $\hat{Y}^{(1)}$ entstehenden Matrix Q gebildet werden. Dann machen wir den Ansatz

$$Y^{(1)} = (y_1^{(1)}, \dots, y_p^{(1)}) = Q_1 V_1$$

mit einer noch zu bestimmenden Matrix V_1 .

V_1 soll nun so bestimmt werden, daß der Einsetzfehler

$$C^{-1} Y^{(1)} - Y^{(1)} \Lambda$$

für eine geeignet gewählte Diagonalmatrix Λ eine minimale euklidische Norm erhält. Wir ergänzen die $n \times p$ -Matrix Q_1 durch eine $n \times (n - p)$ -Matrix \tilde{Q}_1 zu einer orthonormalen $n \times n$ -Matrix und errechnen

$$\begin{aligned} \|C^{-1} Y^{(1)} - Y^{(1)} \Lambda\|_F^2 &= \left\| \begin{pmatrix} Q_1^T \\ \tilde{Q}_1^T \end{pmatrix} (C^{-1} Y^{(1)} - Y^{(1)} \Lambda) \right\|_F^2 \\ &= \left\| \begin{pmatrix} Q_1^T C^{-1} Q_1 V_1 - V_1 \Lambda \\ \tilde{Q}_1^T C^{-1} Q_1 V_1 \end{pmatrix} \right\|_F^2 \\ &= \sum \lambda_i (\Omega_1^T \Omega_1 + \Omega_2^T \Omega_2) \end{aligned}$$

mit

$$\begin{aligned} \Omega_1 &= Q_1^T C^{-1} Q_1 V_1 - V_1 \Lambda, \\ \Omega_2 &= \tilde{Q}_1^T C^{-1} Q_1 V_1. \end{aligned}$$

Dann ist aber

$$\lambda_{\max}^{1/2}(\Omega_1^T \Omega_1 + \Omega_2^T \Omega_2) \geq \lambda_{\max}^{1/2}(\Omega_2^T \Omega_2)$$

mit Gleichheit für $\Omega_1 = \mathbf{0}$, d.h.

$$\begin{aligned} \Lambda &= \text{Diagonalmatrix der Eigenwerte von } Q_1^T C^{-1} Q_1, \\ V_1 &= \text{zugehöriger Eigenvektormatrix.} \end{aligned}$$

Wegen

$$\begin{aligned} \Omega_2^T \Omega_2 &= V_1^T Q_1^T C^{-1} \tilde{Q}_1 \tilde{Q}_1^T C^{-1} Q_1 V_1, \\ \Omega_2 \Omega_2^T &= \tilde{Q}_1^T C^{-1} Q_1 Q_1^T C^{-1} \tilde{Q}_1 \end{aligned}$$

und (von $n - p$ Eigenwerten 0 abgesehen)

$$\lambda(\Omega_2^T \Omega_2) = \lambda(\Omega_2 \Omega_2^T)$$

ist $\lambda_{\max}^{1/2}(\Omega_2^T \Omega_2)$ von der speziellen Wahl von V_1, \tilde{Q}_1 unabhängig, d.h. die im Sinne einer minimalen euklidischen Norm bei der Einsetzprobe optimale Konstruktion von $Y^{(1)}$ ist

$$Y^{(1)} = Q_1 V_1, \quad V_1 = \text{orthonormierte Eigenvektormatrix von } Q_1^T C^{-1} Q_1.$$

Hiermit könnte man nun die Vorgehensweise wiederholen. Bei großem n wäre aber die Eigenvektorbestimmung von $Q_1^T C^{-1} Q_1$ ganz unpraktikabel, weil man dazu C^{-1} benötigte. Wir stellen uns nun vor, das Verfahren sei schon für eine große Zahl k von Schritten durchgeführt worden, gemäß

$$\begin{aligned} C \hat{Y}^{(k+1)} &= Y^{(k)}, \\ \hat{Y}^{(k+1)} &= Q_{k+1} R_{k+1}, \\ Q_{k+1}^T C^{-1} Q_{k+1} &= V_{k+1} \Lambda_{k+1}^{-1} V_{k+1}^T, \\ &\quad (\text{Eigenwertproblem lösen}), \\ Y^{(k+1)} &= Q_{k+1} V_{k+1}. \end{aligned}$$

Dann gilt

$$\begin{aligned} R_{k+1} R_{k+1}^T &= Q_{k+1}^T \hat{Y}^{(k+1)} \hat{Y}^{(k+1)T} Q_{k+1} \\ &= Q_{k+1}^T C^{-1} Y^{(k)} Y^{(k)T} C^{-1} Q_{k+1} \\ &= Q_{k+1}^T C^{-1} Q_k V_k V_k^T Q_k^T C^{-1} Q_{k+1} \\ &= Q_{k+1}^T C^{-1} Q_k Q_k^T C^{-1} Q_{k+1}, \end{aligned}$$

und bei geeigneter Normierung kann dann angenommen werden, daß

$$Q_k Q_k^T \approx Q_{k+1} Q_{k+1}^T,$$

also

$$R_{k+1} R_{k+1}^T \approx (Q_{k+1}^T C^{-1} Q_{k+1})^2.$$

$(Q_{k+1}^T C^{-1} Q_{k+1})^2$ hat die gleichen Eigenvektoren wie $Q_{k+1}^T C^{-1} Q_{k+1}$. Also wird man anstelle der unzugänglichen Eigenvektormatrix V_{k+1} die Eigenvektormatrix von $R_{k+1} R_{k+1}^T$ zur Konstruktion von $Y^{(k+1)}$ benutzen.

Dies führt auf folgenden Gesamtalgorithmus (RITZIT von Rutishauser):

Gegeben sei die $n \times p$ -Matrix $Y^{(0)}$ mit $(Y^{(0)})^T Y^{(0)} = I$.

Für $k = 0, 1, 2, \dots$ lauten die Rechenvorschriften

$$\begin{array}{ll} \hat{X}^{(k+1)} &= L Y^{(k)} && \text{Matrix-Multiplikation} \\ A \hat{Z}^{(k+1)} &= \hat{X}^{(k+1)} && \text{Gleichungssystem lösen} \\ \hat{Y}^{(k+1)} &= L^T \hat{Z}^{(k+1)} && \text{Matrix-Multiplikation} \\ \hat{Y}^{(k+1)} &= Q_{k+1} R_{k+1} && \text{QR-Zerlegung} \\ R_{k+1} R_{k+1}^T &= V_{k+1} \Lambda_{k+1}^{-2} V_{k+1}^T && \text{vollständiges Eigenwertproblem lösen} \\ Y^{(k+1)} &= Q_{k+1} V_{k+1} && \text{Matrix-Multiplikation.} \end{array}$$

Wir haben hier die Bezeichnung Λ^{-2} gewählt, weil die Eigenwerte von $R_{k+1} R_{k+1}^T$ für die Quadrate der Reziprokwerte der kleinsten Eigenwerte sind.

Da $R_{k+1} R_{k+1}^T$ eine symmetrische $p \times p$ -Matrix und p nie sehr groß ist, ist die Bestimmung aller Eigenwerte und Eigenvektoren dieser Matrix unproblematisch, wie wir noch sehen werden. Wenn die Bandbreiten von A und B nicht allzu groß sind, ist die Berechnung der Cholesky-Zerlegung von B und die Gleichungsauflösung mit der Matrix A , etwa ebenfalls mit der Cholesky-Zerlegung von A , noch vertretbar.

Für das hergeleitete Verfahren, die simultane Vektoriteration nach Rutishauser, gilt der folgende Konvergenzsatz

Satz 1.5.3 *Seien*

$$0 < \lambda_1 \leq \dots \leq \lambda_p < \lambda_{p+1} < \dots \leq \lambda_n$$

die Eigenwerte des allgemeinen Eigenwertproblems

$$A x = \lambda B x, \quad A = A^T, \quad B = B^T \quad \text{pos. def.}, \quad B = L L^T,$$

mit den zugehörigen Eigenvektoren x_i . Ferner gelte: $Y^{(0)T} L^T(x_1, \dots, x_p)$ ist regulär. Dann gibt es Konstanten γ_i , so daß

$$|\sin \angle(y_i^{(k)}, L^T x_i)| \leq \gamma_i (\lambda_i / \lambda_{p+1})^k.$$

Dabei ist $y_i^{(k)}$ die i -te Spalte von $Y^{(k)}$. Zum Beweis vgl. z.B. bei Golub und van Loan. □

Die hier nicht diskutierte wesentliche zusätzliche Aussage von Satz 1.5.3 ist, daß die Konvergenzgeschwindigkeit für die niedrigen Eigenwerte besser ist als etwa für λ_p . In der Praxis wird man p deshalb etwas größer wählen als die Anzahl der eigentlich gewünschten Eigenwerte bzw. Eigenvektoren. Die Voraussetzung “ $(Y^{(0)})^T L^T(x_1, \dots, x_p)$ regulär“ entspricht der Voraussetzung “ $\xi_1 \neq 0$ “ beim v. Mises Verfahren.

Wenn $Y^{(0)}$ nicht gut gewählt ist, können Eigenwerte “übersprungen“ werden, d.h. man

findet statt der gewünschten $\lambda_1, \dots, \lambda_5$ etwa $\lambda_1, \lambda_4, \lambda_5, \lambda_6, \lambda_7$! Sicherheit hierüber kann man sich nur durch einen zusätzlichen Test an der Matrix $A - \mu B$ verschaffen, mit μ gleich der letzten Näherung für λ_p . Liegen mehrfache Eigenwerte vor und wird p falsch gewählt, etwa

$$\lambda_1 < \lambda_2 = \lambda_3 = \lambda_4 < \lambda_5 < \dots, \quad p = 3,$$

dann tritt keine Konvergenz mehr ein. Gute Programme für dieses Verfahren (z.B. RITZIT im Buch von Wilkinson und Reinsch) steuern deshalb den Parameter p in Abhängigkeit vom beobachteten Konvergenzverhalten selbst. Das gleiche gilt für die Wahl von $Y^{(0)}$.

Bemerkung 1.5.3 Für den nichthermitischen Fall gibt es eine Verallgemeinerung dieses Verfahrens von Stewart: *Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices. Numer. Math. 25, 123-136 (1976).*

1.6 Das Jacobi-Verfahren

Während bei den Unterraummethoden ein Ausschnitt der Gesamtheit der Eigenwerte bzw. Eigenvektoren bestimmt wird (etwa die p kleinsten Eigenwerte mit Eigenvektoren), wird bei den Transformationsmethoden, ausgehend von der Matrix

$$A^{(0)} = A$$

eine Folge von durch Ähnlichkeitstransformationen verknüpften Matrizen

$$A^{(k+1)} = (T^{(k)})^{-1} A^{(k)} T^{(k)} \quad k = 0, 1, 2, \dots$$

konstruiert mit dem Ziel, im Grenzwert eine Diagonalmatrix oder zumindest eine obere Dreiecksmatrix zu erhalten.

Wegen des Satzes von Schur (s.h.) ist letzteres für jede quadratische Matrix sogar mit unitären Transformationen zu erreichen ($(T^{(k)})^{-1} = (T^{(k)})^H$).

Es wird dann

$$\lim_{k \rightarrow \infty} a_{i,i}^{(k)} = \lambda_{\pi_i}, \quad (\pi_1, \dots, \pi_n) \text{ Permutation von } (1, \dots, n).$$

Jeder Häufungswert von

$$\prod_{k=0}^{\infty} T^{(k)}$$

ist Eigenvektormatrix von $A^{(0)}$. Es sind dann alle Eigenwerte und Eigenvektoren von A gefunden.

Die Transformationsmethoden werden deshalb stets dann angewendet, wenn alle Eigenwerte und Eigenvektoren von $A = A^{(0)}$ zu bestimmen sind. Dies ist z.B. bei der simultanen Vektoriteration der Fall, wo in jedem Iterationsschritt alle Eigenwerte und Eigenvektoren von $R^{(k)}(R^{(k)})^T$ zu bestimmen sind. Ebenso tritt diese Aufgabe im

3. Man setze

$$A^{(k+1)} = (T^{(k+1)})^T A^{(k)} T^{(k+1)} = (A^{(k+1)})^T.$$

Beim Übergang von $A^{(k)}$ zu $A^{(k+1)}$ brauchen nur die Elemente in der r -ten und s -ten Zeile und Spalte von $A^{(k+1)}$ neu berechnet zu werden, für die übrigen gilt

$$a_{\mu\nu}^{(k+1)} = a_{\mu\nu}^{(k)}, \quad \mu \neq r, s; \quad \nu \neq r, s. \quad (1.9)$$

Die neu zu berechnenden Elemente von $A^{(k+1)}$ dagegen sind

$$\begin{aligned} a_{\nu r}^{(k+1)} &= a_{r\nu}^{(k+1)} = a_{\nu r}^{(k)} \cos \varphi_{k+1} - a_{\nu s}^{(k)} \sin \varphi_{k+1}, & \nu = 1, \dots, n; \quad \nu \neq r, s, \\ a_{\nu s}^{(k+1)} &= a_{s\nu}^{(k+1)} = a_{\nu r}^{(k)} \sin \varphi_{k+1} + a_{\nu s}^{(k)} \cos \varphi_{k+1}, & \nu = 1, \dots, n; \quad \nu \neq r, s, \\ a_{rr}^{(k+1)} &= a_{rr}^{(k)} \cos^2 \varphi_{k+1} - a_{rs}^{(k)} \sin(2\varphi_{k+1}) + a_{ss}^{(k)} \sin^2 \varphi_{k+1}, & (1.10) \\ a_{ss}^{(k+1)} &= a_{rr}^{(k)} \sin^2 \varphi_{k+1} + a_{rs}^{(k)} \sin(2\varphi_{k+1}) + a_{ss}^{(k)} \cos^2 \varphi_{k+1}, \\ a_{rs}^{(k+1)} &= a_{sr}^{(k+1)} = 0. \end{aligned}$$

Über die Konvergenz des Jacobi-Verfahrens gilt der

Satz 1.6.1 *Ist A eine reell-symmetrische Matrix, so gilt*

$$\lim_{k \rightarrow \infty} A^{(k)} = D. \quad \square$$

Beweis: Aus (1.10) folgt zunächst

$$\left(a_{rr}^{(k)}\right)^2 + \left(a_{ss}^{(k)}\right)^2 + 2\left(a_{rs}^{(k)}\right)^2 = \left(a_{rr}^{(k+1)}\right)^2 + \left(a_{ss}^{(k+1)}\right)^2. \quad (1.11)$$

Es verschwinden ferner die Außerdiagonalelemente $a_{rs}^{(k+1)}$ und $a_{sr}^{(k+1)}$ von $A^{(k+1)}$. Hieraus folgt weiter

$$\sum_{i=1}^n \left(a_{ii}^{(k+1)}\right)^2 = \sum_{i=1}^n \left(a_{ii}^{(k)}\right)^2 + 2\left(a_{rs}^{(k)}\right)^2. \quad (1.12)$$

Die Größe

$$\text{Sp } B = \sum_{i=1}^n b_{ii}$$

wird als Spur von B bezeichnet. Ist T eine orthogonale $n \times n$ -Matrix und B reell, so gilt

$$\text{Sp}(B^T B) = \text{Sp}(T^T B^T B T). \quad (1.13)$$

Denn es ist nach dem Vietaschen Wurzelsatz, wenn λ_i^2 die Eigenwerte der symmetrischen Matrix $B^T B$ sind,

$$\text{Sp}(B^T B) = \sum_{i=1}^n \lambda_i^2.$$

Andererseits besitzen $B^T B$ und $T^T B^T B T$ wegen $T^T = T^{-1}$ und $\det (T^T B^T B T - \lambda I) = \det T^T (B^T B - \lambda I) T = \det (B^T B - \lambda I)$ die gleichen Eigenwerte, und es folgt (1.13). Es gilt dann

$$\begin{aligned} \operatorname{Sp}((A^{(k+1)})^T A^{(k+1)}) &= \operatorname{Sp}((T^{(k+1)})^T (A^{(k)})^T T^{(k+1)} (T^{(k+1)})^T A^{(k)} T^{(k+1)}) \\ &= \operatorname{Sp}((T^{(k+1)})^T (A^{(k)})^T A^{(k)} T^{(k+1)}) = \operatorname{Sp}((A^{(k)})^T A^{(k)}), \\ &k = 0, 1, \dots \end{aligned}$$

Wegen

$$\operatorname{Sp}(B^T B) = \sum_{i,j=1}^n b_{ij}^2,$$

folgt hieraus

$$\sum_{i,j=1}^n (a_{ij}^{(k+1)})^2 = \sum_{i,j=1}^n (a_{ij}^{(k)})^2. \quad (1.14)$$

Setzen wir für $l = 0, 1, \dots$

$$p_l = \sum_{\substack{i,j=1 \\ i \neq j}}^n (a_{ij}^{(l)})^2,$$

so gilt mit (1.12), (1.14) für $k = 0, 1, \dots$

$$\begin{aligned} p_{k+1} &= \sum_{i,j=1}^n (a_{ij}^{(k+1)})^2 - \sum_{i=1}^n (a_{ii}^{(k+1)})^2 = \sum_{i,j=1}^n (a_{ij}^{(k)})^2 - \sum_{i=1}^n (a_{ii}^{(k)})^2 - 2(a_{rs}^{(k)})^2 \\ &= p_k - 2(a_{rs}^{(k)})^2. \end{aligned} \quad (1.15)$$

Da $a_{rs}^{(k)}$ betragsmäßig maximales Element außerhalb der Hauptdiagonalen ist und es genau $n^2 - n$ Nichtdiagonalelemente gibt, folgt ferner

$$p_k \leq (n^2 - n) (a_{rs}^{(k)})^2, \quad (1.16)$$

d.h.

$$-2(a_{rs}^{(k)})^2 \leq -\frac{2p_k}{n^2 - n}.$$

(Man beachte, daß hier selbstverständlich $n \geq 2$ gilt). Daher hat man mit (1.15)

$$p_{k+1} \leq p_k - \frac{2}{n^2 - 2} p_k = \frac{n^2 - n - 2}{n^2 - n} p_k = \alpha(n) p_k$$

mit $\alpha(n) < 1$. Hieraus folgt schließlich

$$p_{k+1} \leq [\alpha(n)]^{k+1} p_0$$

oder ausführlich

$$\sum_{\substack{i,j=1 \\ i \neq j}}^n (a_{ij}^{(k+1)})^2 \leq [\alpha(n)]^{k+1} \sum_{\substack{i,j=1 \\ i \neq j}}^n a_{ij}^2$$

und somit

$$\lim_{k \rightarrow \infty} \sum_{\substack{i,j=1 \\ i \neq j}}^n \left(a_{ij}^{(k+1)} \right)^2 = 0.$$

Daher konvergiert die Folge $\{ A^{(k)} \}$ gegen eine Diagonalmatrix, welche, wie man aufgrund der Konstruktionsvorschrift sieht, in der Hauptdiagonalen genau die Eigenwerte von A enthält, d.h. mit der Matrix D übereinstimmt. Damit ist der Satz bewiesen. \square

Nach (1.10) gilt $a_{rs}^{(k+1)} = a_{sr}^{(k+1)} = 0$. Dies bedeutet im allgemeinen jedoch nicht, daß auch

$$a_{rs}^{(l)} = a_{sr}^{(l)} = 0, \quad l > k + 1$$

gilt, denn sonst würde in endlich vielen Schritten die Diagonalisierung durchgeführt sein. Die Folge der Zahlen p_k , deren Größe ja ein Maß für die Abweichung der Matrix $A^{(k)}$ von der Diagonalform ist, nimmt jedoch stets monoton ab, und zwar in der Regel um so langsamer, je größer $\alpha(n)$, d.h. je größer n ist. Zumindest bei großen Matrizen A ist das Verfahren daher oft recht aufwendig. Infolge der sehr groben Abschätzung (1.16) ist die Konvergenzgeschwindigkeit in der Praxis jedoch höher als nach diesen theoretischen Überlegungen erwartet werden kann.

Beispiel 1.6.1

a) Wir erläutern das Jacobi-Verfahren zunächst am Beispiel der Matrix

$$A^{(0)} = A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

mit den Eigenwerten 1 und 3. Trivialerweise liefert hier der Algorithmus uns schon nach einem Schritt die Diagonalmatrix.

Wir wählen $a_{rs} = a_{21} = -1$, d.h. $r > s$, und erhalten wegen $a_{ss} = a_{rr} = 2$

$$\varphi_1 = \text{sign}(a_{rs}) \frac{\pi}{4} = \text{sign}(-1) \frac{\pi}{4} = -\frac{\pi}{4}$$

und -es sind ja die Indizes r und s zu vertauschen-

$$\begin{aligned} t_{22}^{(1)} &= t_{11}^{(1)} = \cos \varphi_1 = \cos\left(-\frac{\pi}{4}\right) = \cos \frac{\pi}{4}, \\ t_{21}^{(1)} &= -t_{12}^{(1)} = \sin \varphi_1 = \sin\left(-\frac{\pi}{4}\right) = -\sin \frac{\pi}{4}, \end{aligned}$$

somit

$$T^{(1)} = \begin{bmatrix} \cos \frac{\pi}{4} & \sin \frac{\pi}{4} \\ -\sin \frac{\pi}{4} & \cos \frac{\pi}{4} \end{bmatrix}.$$

Dann ist

$$A^{(1)} = \begin{bmatrix} \cos \frac{\pi}{4} & -\sin \frac{\pi}{4} \\ \sin \frac{\pi}{4} & \cos \frac{\pi}{4} \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} \cos \frac{\pi}{4} & \sin \frac{\pi}{4} \\ -\sin \frac{\pi}{4} & \cos \frac{\pi}{4} \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix},$$

d.h. $A^{(1)}$ hat Diagonalform und A die Eigenwerte 1 und 3.

b) Wir betrachten die Matrix

$$A^{(0)} = A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

mit den Eigenwerten 2 , $2 - \sqrt{2}$, $2 + \sqrt{2}$. Wir wollen die erste Iterierte $A^{(1)}$ berechnen und wählen etwa $a_{rs} = a_{12} = -1$. Dann ist

$$\varphi_1 = \text{sign}(a_{12}) \frac{\pi}{4} = \text{sign}(-1) \frac{\pi}{4} = -\frac{\pi}{4}$$

und

$$\begin{aligned} t_{11}^{(1)} &= t_{22}^{(1)} = \cos\left(-\frac{\pi}{4}\right) = \cos \frac{\pi}{4}, \\ t_{12}^{(1)} &= -t_{21}^{(1)} = \sin\left(-\frac{\pi}{4}\right) = -\sin \frac{\pi}{4}, \end{aligned}$$

so daß

$$T^{(1)} = \begin{bmatrix} \cos \frac{\pi}{4} & -\sin \frac{\pi}{4} & 0 \\ \sin \frac{\pi}{4} & \cos \frac{\pi}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

und

$$A^{(1)} = (T^{(1)})^T A^{(0)} T^{(1)} = \begin{bmatrix} 1 & 0 & -\sin \frac{\pi}{4} \\ 0 & 3 & -\cos \frac{\pi}{4} \\ -\sin \frac{\pi}{4} & -\cos \frac{\pi}{4} & 2 \end{bmatrix}$$

gilt. Um $A^{(2)}$ zu berechnen, kann man etwa

$$a_{rs}^{(1)} = a_{13}^{(1)} = -\sin \frac{\pi}{4}$$

wählen. □

Abschließend sei über das Jacobi-Verfahren noch bemerkt:

- a) Es läßt sich auf hermitesche Matrizen direkt übertragen.
- b) Für nichtsymmetrische (bzw. nichthermitesche) Matrizen gibt es ein ähnliches Verfahren. Man vgl. hierzu etwa bei Wilkinson und Reinsch.

Bei den hier betrachteten reell symmetrischen Matrizen kann natürlich stets $r < s$ gewählt werden. Wendet man das Jacobi-Verfahren aber auf nichtsymmetrische Matrizen an, so muß das Vorzeichen von $s - r$ berücksichtigt werden. Man vgl. z.B. bei Wilkinson und Reinsch. Es ist nicht notwendig, in jedem Schritt das betragsgrösste Ausserdiagonalelement in null zu überführen. Ebenfalls beweisbar konvergent ist die Methode in einem sogenannten "sweep" (das ist ein sequentieller Durchlauf über alle Ausserdiagonalelemente) alle Elemente zu null zu machen, deren Betrag über einem Schwellenwert, etwa

$$\tau = \left(\sum \sum_{p < q} |a_{pq}|^2 / (n(n-1)) \right)^{1/2}$$

liegt, und dabei jeweils τ neu zu berechnen. Das kostet nur eine Multiplikation, eine Division und eine Quadratwurzel pro Rotation. Man kann sogar beweisen, daß das Verfahren asymptotisch quadratisch konvergiert über einen vollständigen sweep, im Sinne von

$$\tau_{k+1} \leq C\tau_k^2$$

mit einer Konstanten C die von der Separation der Eigenwerte abhängt, wobei k die sweeps zählt. Details dazu findet man nur in Originalarbeiten von Schönhage (Num. Math. 3, (1961), 374–380) und Wilkinson (Num. Math. 4, (1962), 296–300).

1.7 Das QR–Verfahren

Das im Folgenden beschriebene QR–Verfahren wird vor allem dann eingesetzt, wenn es gilt, alle Eigenwerte (und evtl. auch Eigenvektoren) einer Matrix zu bestimmen. Aus Aufwandsgründen führt man dieses Verfahren nur an Hessenberg– bzw. Tridiagonalmatrizen durch, d.h. eine allgemeine Matrix wird zunächst auf die entsprechende Gestalt transformiert. (Diese Matrixformen sind invariant gegenüber der im Algorithmus definierten Transformation; Übg.). In der folgenden Darstellung betrachten wir jedoch den allgemeinen Fall und gehen auf rechentechnische Besonderheiten in Bemerkungen ein. Wesentliche Hilfsmittel zum Verständnis des Verfahrens sind der Satz von Schur und das Verfahren von Wielandt, jetzt mit variablen Spektralverschiebungen μ_k . Zunächst zeigen wir

Satz 1.7.1 Satz von Schur Sei $A \in \mathbb{C}^{n \times n}$ beliebig. Dann existiert ein unitäres U , so daß

$$U^H A U = R = \text{obere Dreiecksmatrix}$$

(mit den Eigenwerten von A auf der Diagonalen von R) □

Beweis: Sei x ein Eigenvektor von A zum Eigenwert λ . Wähle Q unitär mit

$$Q^H x = \|x\| e_1$$

z.B. als eine Householdermatrix. Dann ist

$$Q^H A Q = \left(\begin{array}{c|c} \lambda_1 & a^H \\ \hline 0 & \tilde{A} \end{array} \right)$$

denn

$$Q^H A Q e_1 = Q^H A \frac{x}{\|x\|} = Q^H \lambda \frac{x}{\|x\|} = \lambda e_1.$$

\tilde{A} hat die übrigen Eigenwerte von A zu Eigenwerten. Wir wiederholen den Vorgang mit \tilde{A} und definieren die entsprechende Ähnlichkeitstransformation für A durch

$$\left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{Q} \end{array} \right)$$

usw. U ist dann Produkt aller dieser Matrizen. □

Beispiel 1.7.1 Wir berechnen die Schurnormalform der Matrix

$$A = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 3 & -1 \\ 2 & 3 & -2 \end{pmatrix}.$$

Ein Eigenvektor von A ist $\frac{1}{3}(1, 2, 2)^T$. Mit dem gegebenen Eigenvektor wird die erste HOUSEHOLDER-Matrix bestimmt, als

$$U_1 = I - \beta_1 u_1 u_1^T$$

wobei

$$u_1 = \begin{pmatrix} \text{sign}(y_1)(|y_1| + \|y\|_2) \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \frac{4}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{pmatrix}$$

und folglich $\beta_1 = \frac{2}{u_1^T u_1} = \frac{3}{4}$ ist. Es wird natürlich vermieden, die Housholdermatrix explizit aufzustellen, vielmehr berechnet man die Matrix $A_1 = U_1 A U_1$ spaltenweise, bzw. zeilenweise.

Die i -te Spalte der Matrix $\tilde{A}_1 = U_1 A$ ist dann gegeben durch

$$(\tilde{A}_1)_{\cdot, i} = (U_1 A)_{\cdot, i} = (I - \beta_1 u_1 u_1^T) A_{\cdot, i} = A_{\cdot, i} - \beta_1 (u_1^T A_{\cdot, i}) u_1.$$

Es ergibt sich

$$\tilde{A}_1 = \frac{1}{3} \begin{pmatrix} -6 & -13 & 7 \\ -6 & 1 & 2 \\ 0 & -1 & 1 \end{pmatrix}$$

Die j -te Zeile der Matrix $A_1 = \tilde{A}_1 U_1$ lautet

$$(A_1)_{j, \cdot} = (\tilde{A}_1 U_1)_{j, \cdot} = (\tilde{A}_1)_{j, \cdot} (I - \beta_1 u_1 u_1^T) = (\tilde{A}_1)_{j, \cdot} - \beta_1 ((\tilde{A}_1)_{j, \cdot} u_1) u_1^T.$$

Das ergibt

$$A_1 = \frac{1}{3} \begin{pmatrix} 6 & -7 & 13 \\ 0 & 4 & 5 \\ 0 & 1 & -1 \end{pmatrix}.$$

Im zweiten Schritt muß nun ein Eigenvektor der Restmatrix

$$\bar{A}_1 = \frac{1}{3} \begin{pmatrix} 4 & 5 \\ 1 & -1 \end{pmatrix}$$

bestimmt werden. Das charakteristische Polynom lautet

$$\left(\frac{4}{3} - \lambda\right)\left(-\frac{1}{3} - \lambda\right) - \frac{5}{9} = 0$$

und liefert die Eigenwerte

$$\lambda_2 = \frac{1}{2}(1 + \sqrt{5}) \quad \text{und} \quad \lambda_3 = \frac{1}{2}(1 - \sqrt{5}).$$

Der Eigenvektor zum Eigenwert λ_2 ist

$$v_2 = \begin{pmatrix} 1 \\ -\frac{5}{10} + \frac{3}{10}\sqrt{5} \end{pmatrix}.$$

Die zweite Householdermatrix U_2 wird dann gebildet durch

$$U_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \hat{U}_{11} & \hat{U}_{12} \\ 0 & \hat{U}_{21} & \hat{U}_{22} \end{pmatrix} \quad \text{mit} \quad \hat{U}_2 = \begin{pmatrix} \hat{U}_{11} & \hat{U}_{12} \\ \hat{U}_{21} & \hat{U}_{22} \end{pmatrix} = I - \beta_2 u_2 u_2^T,$$

wobei

$$u_2 = \begin{pmatrix} 2.014485 \\ 0.170820 \end{pmatrix} \quad \text{und} \quad \beta_2 = \frac{2}{u_2^T u_2} = 0.489317.$$

Nach dem selben Schema wie im ersten Schritt kann nun $A_2 = U_2 A_1 U_2$ berechnen ohne U_2 explizit aufzustellen. Es folgt

$$A_2 = \begin{pmatrix} 2 & 1.570365 & 4.664352 \\ 0 & 1.618034 & -1.333333 \\ 0 & 0 & -.618034 \end{pmatrix}.$$

Eine Umformulierung dieses Satzes ist

Satz 1.7.2 Sei $A \in \mathbb{C}^{n \times n}$ bel.; $A^H x = \bar{\lambda} x$, mit $\|x\|_2 = 1$ und $Q \in \mathbb{C}^{n \times n}$ unitär mit $Q e_n = \alpha x$. Dann gilt

$$Q^H A^H Q e_n = \bar{\lambda} e_n$$

d.h.

$$Q^H A Q = \left(\begin{array}{ccc|c} & & & * \\ & \tilde{A} & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & \lambda \end{array} \right) \quad \text{mit } \tilde{A} \in \mathbb{C}^{(n-1) \times (n-1)}$$

□

Wir betrachten nun die Anwendung dieses Satzes im Zusammenhang mit ungenauen Eigenvektornäherungen

Wir nehmen nun zunächst einmal an, μ_0 sei eine "gute" Näherung für einen Eigenwert λ von A und daß die Anwendung des Wielandtverfahrens sinnvoll ist.

Sei eine QR -Zerlegung von $A - \mu_0 I$ gegeben.

$$A - \mu_0 I = Q_0 R_0$$

Mit $\mu_0 \rightarrow \lambda$ gilt $\rho_{ii}^{(0)} \rightarrow 0$ für mindestens ein i . Ist A eine obere Hessenbergmatrix mit nicht verschwindenden Subdiagonalelementen (in der Praxis allein interessierender Fall!), dann ist notwendig $i = n$ und deshalb wollen wir im Folgenden diesen Fall betrachten. Sei also $\rho_{nn}^{(0)}$ "sehr klein". Es wird

$$(A^H - \bar{\mu}_0 I) Q_0 e_n = R_0^H e_n = \bar{\rho}_{nn}^{(0)} e_n \approx 0, \quad (|\bar{\rho}_{nn}^{(0)}| = |\rho_{nn}^{(0)}| \approx 0)$$

d.h. $x_1 := Q_0 e_n$ ist offenbar eine gute Eigenvektornäherung von $(A^H - \bar{\mu}_0 I)$ (d.h. Linkseigenvektornäherung von $A - \mu_0 I$) und geht aus dem Wielandtverfahren für A^H hervor mit der Verschiebung $\bar{\mu}_0$ und $x_0 := \bar{\rho}_{nn}^{(0)} e_n$.

Ferner wird

$$Q_0^H (A^H - \bar{\mu}_0 I) Q_0 e_n \approx 0 \quad \text{d.h.} \quad e_n^T Q_0^H A Q_0 \approx \mu_0 e_n^T$$

d.h. die letzte Zeile von

$$Q_0^H A Q_0 = Q_0^H (Q_0 R_0 + \mu_0 I) Q_0 = R_0 Q_0 + \mu_0 I$$

enthält also außerhalb der Diagonalen nur "kleine" Elemente, vgl. Satz 1.7.2. Da $x_1 = Q_0 e_n$ offenbar eine gute Eigenvektornäherung ist, ist

$$\bar{\mu}_1 := R(x_1; A^H) = e_n^T \underbrace{Q_0^H A^H Q_0}_{=: A_1^H} e_n = \bar{\alpha}_{nn}^{(1)}$$

d.h.

$$\mu_1 := \alpha_{nn}^{(1)} \quad \text{mit} \quad A_1 := Q_0^H A Q_0 = R_0 Q_0 + \mu_0 I$$

eine eventuell bessere Eigenwertnäherung für λ . Wir wiederholen darum den Vorgang für

$$\begin{aligned} A_1 - \mu_1 I & \quad (= R_0 Q_0 + (\mu_0 - \mu_1) I) : \\ A_1 - \mu_1 I & = Q_1 R_1 \end{aligned}$$

Dann

$$\begin{aligned} (A_1^H - \bar{\mu}_1 I) Q_1 e_n & = \bar{\rho}_{nn}^{(1)} e_n \\ (Q_0^H A^H Q_0 - \bar{\mu}_1 I) Q_1 e_n & = \bar{\rho}_{nn}^{(1)} e_n \\ (A^H - \bar{\mu}_1 I) \underbrace{Q_0 Q_1 e_n}_{=: x_2} & = \bar{\rho}_{nn}^{(1)} x_1 \end{aligned}$$

d.h. $x_2 := Q_0 Q_1 e_n$ ist das Ergebnis eines weiteren Wielandtschrittes, jetzt mit einer anderen, eventuell besseren Spektralverschiebung. Dies legt folgende Grundversion des QR-Algorithmus nahe:

Sei $A_0 := (\alpha_{ij}^{(0)}) := A$. Wähle $0 < \delta < 1$ (z.B. $\delta = 10^{-1}$ ist sinnvoll)

Für $k = 0, 1, \dots$

1. Wähle μ_k geeignet, z.B.

$$\mu_k = \begin{cases} 0 & \text{falls } k = 0 \text{ oder } \sum_{j=1}^{n-1} |\alpha_{n,j}^{(k)}|^2 > \delta \sum_{j=1}^{n-1} |\alpha_{n,j}^{(0)}|^2 \\ \alpha_{n,n}^{(k)} & \text{sonst} \end{cases}$$

2. Berechne die QR-Zerlegung

$$A_k - \mu_k I = Q_k R_k$$

3. $A_{k+1} := R_k Q_k + \mu_k I$

Beispiel 1.7.2 Sei

$$A = \begin{pmatrix} 30 & 2 & 0 \\ 2 & 20 & 1 \\ 0 & 1 & 10 \end{pmatrix}$$

Wir schätzen zunächst den kleinsten Eigenwert von A zu 10 unter Benutzung des Kreisesatzes. Wir betrachten zunächst die QR-Zerlegung einer oberen Hessenbergmatrix. In jedem Schritt erfolgt eine Multiplikation der Art:

$$P_i A = \begin{pmatrix} I & 0 & 0 \\ 0 & \tilde{P}_i & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ 0 & \tilde{P}_i A_{22} & \tilde{P}_i A_{23} \\ 0 & A_{32} & A_{33} \end{pmatrix}$$

wobei \tilde{P}_i eine 2×2 Matrix ist, die in der Diagonalen ab der i -ten Position steht. Wir sehen, daß bei der Produktbildung nur die Zeilen i und $i+1$ von links mit \tilde{P}_i multipliziert werden. Analog erkennt man, daß bei dem Produkt RQ die Spalten i und $i+1$ von rechts mit \tilde{P}_i multipliziert werden.

Im Folgenden beschreiben wir nur die wesentlichen Teile \tilde{P}_i der P_i

$$A - 10I = \begin{pmatrix} 20 & 2 & 0 \\ 2 & 10 & 1 \\ 0 & 1 & 0 \end{pmatrix} \Rightarrow \tilde{P}_1 = \nu \begin{pmatrix} 20 & 2 \\ 2 & -20 \end{pmatrix} \text{ mit } \nu = \frac{1}{\sqrt{404}}$$

Es folgt für $A_1 = P_1(A - 10I)$:

$$A_1 = \begin{pmatrix} 1/\nu & 60\nu & 2\nu \\ 0 & -196\nu & -20\nu \\ 0 & 1 & 0 \end{pmatrix} \Rightarrow \tilde{P}_2 = \mu \begin{pmatrix} -196\nu & 1 \\ 1 & 196\nu \end{pmatrix} \text{ mit } \mu = \frac{1}{\sqrt{1 + 196^2\nu^2}}$$

$$R = P_2 A_1 = \begin{pmatrix} 1/\nu & 60\nu & 2\nu \\ 0 & 1/\mu & 3920\nu^2\mu \\ 0 & 0 & -20\nu\mu \end{pmatrix}$$

$$RP_1 = \begin{pmatrix} 20 + 120\nu^2 & 2 - 1200\nu^2 & 2\nu \\ 2\nu/\mu & -20\nu/\mu & 3920\nu^2\mu \\ 0 & 0 & -20\nu\mu \end{pmatrix} = \begin{pmatrix} 20.29703 & * & * \\ 0.97539 & -9.75386 & 0.98985 \\ 0 & 0 & -0.10151 \end{pmatrix}$$

Aus Symmetriegründen brauchen wir die Multiplikation nicht ganz auszuführen.

$$RP_1 P_2 = \begin{pmatrix} 20.29703 & * & * \\ 0.97539 & 9.80395 & * \\ 0 & -0.01036 & -0.10098 \end{pmatrix}$$

$\Rightarrow \lambda_1 \in [9.88865, 9.90938]$. Dies ist nun bereits eine erhebliche Verbesserung

Zu diesem Verfahren gilt

Satz 1.7.3 Seien die Folgen $\{A_k\}$, $\{Q_k\}$, $\{R_k\}$ durch obigen Algorithmus definiert und μ_k kein Eigenwert von A ($\forall k$). Setze

$$\tilde{Q}_k := Q_0 \cdots Q_k, \quad \tilde{R}_k := R_k \cdots R_0$$

Dann gilt

$$(A - \mu_k I) \cdots (A - \mu_0 I) = \tilde{Q}_k \tilde{R}_k$$

$$(i) \quad \tilde{Q}_k e_n = (((A - \mu_k I) \cdots (A - \mu_0 I))^{-1})^H \underbrace{e_n}_{x_0} / \tau_k$$

$$|\tau_k| = \|(((A - \mu_k I) \cdots (A - \mu_0 I))^{-1})^H e_n\|$$

$$(ii) \quad \alpha_{nn}^{(k)} = R(\tilde{Q}_{k-1} e_n; A) = \lambda_0 + e_n^T \tilde{Q}_{k-1}^H (A - \lambda_0 I) \tilde{Q}_{k-1} e_n$$

wenn man $\tilde{Q}_k e_n$ als Eigenvektornäherung zum Eigenwert $\bar{\lambda}_0$ von A^H ansieht. □

Beweis:

$$\begin{aligned} A_k &= R_{k-1} Q_{k-1} + \mu_{k-1} I = Q_{k-1}^H (A_{k-1} - \mu_{k-1} I) Q_{k-1} + \mu_{k-1} I \\ &= Q_{k-1}^H A_{k-1} Q_{k-1} = \cdots = Q_{k-1}^H \cdots Q_0^H A Q_0 \cdots Q_{k-1} \\ &= \tilde{Q}_{k-1}^H A \tilde{Q}_{k-1}. \end{aligned}$$

Daher

$$\begin{aligned} A - \mu_j I &= \tilde{Q}_{j-1} A_j \tilde{Q}_{j-1}^H - \mu_j I = \tilde{Q}_{j-1} (A_j - \mu_j I) \tilde{Q}_{j-1}^H \\ &= \tilde{Q}_{j-1} Q_j R_j \tilde{Q}_{j-1}^H = \tilde{Q}_j R_j \tilde{Q}_j^H \\ (A - \mu_k I) \cdots (A - \mu_0 I) &= \tilde{Q}_k R_k \tilde{Q}_{k-1}^H \tilde{Q}_{k-1} R_{k-1} \cdots \tilde{Q}_0 \tilde{R}_0 \\ &= \tilde{Q}_k R_k \cdots R_0 = \tilde{Q}_k \tilde{R}_k \end{aligned}$$

also ist

$$((A - \mu_0 I)^{-1} \cdots (A - \mu_k I)^{-1})^H e_n = (\tilde{R}_k^{-1} \tilde{Q}_k^H)^H e_n = \tilde{Q}_k e_n / \bar{\rho}_{nn}^{(k)}$$

und wegen $\|\tilde{Q}_k e_n\| = 1$ folgt (i).

Sei λ_0 ein Eigenwert von A und

$$v_k := (A^H - \bar{\lambda}_0 I) \tilde{Q}_{k-1} e_n = (\tilde{Q}_{k-1} A_k^H \tilde{Q}_{k-1}^H - \bar{\lambda}_0 I) \tilde{Q}_{k-1} e_n$$

also

$$\begin{aligned} e_n^T A_k &= \lambda_0 e_n^T + v_k^H \tilde{Q}_{k-1} \\ \alpha_{nn}^{(k)} &= e_n^T A_k e_n = \lambda_0 + v_k^H \tilde{Q}_{k-1} e_n \\ &= e_n^T \tilde{Q}_{k-1}^H (A - \lambda_0 I) \tilde{Q}_{k-1} e_n + \lambda_0 \\ &= R(\tilde{Q}_{k-1} e_n; A) \end{aligned}$$

□

Es ist also jeweils $\tilde{Q}_k e_n$ Resultat des Wielandt–Verfahrens mit den variablen Verschiebungen $\bar{\mu}_0, \dots, \bar{\mu}_k$ (für A^H) und $\alpha_{nn}^{(k)}$ der zur letzten Eigenvektornäherung gehörende Rayleighquotient. Die Resultate von Satz 1.1.4 sowie Abschnitt 5 zeigen, daß dementsprechend die letzte Zeile von A_k außerhalb der Diagonalen außerordentlich schnell gegen null konvergieren wird, wenn nur μ_0 bereits hinreichend gute Eigenwertnäherung war. Gestartet wird das Verfahren mit Verschiebung 0, bis sich in der letzten Zeile die Konvergenz zu manifestieren beginnt (vgl. obige Steuerung). Dazu gilt folgender Konvergenzsatz:

Satz 1.7.4 *Es sei $A \in \mathbb{C}^{n \times n}$ und für die Eigenwerte λ_i von A gelte*

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$$

Ferner sei

$$YA = \Lambda Y, \quad \Lambda := \text{diag}(\lambda_1, \dots, \lambda_n)$$

(d.h. $Y = X^{-1}$, wo X ein vollständiges Eigenvektorsystem von A ist) und Y besitze eine Dreieckszerlegung

$$Y = L_Y R_Y, \quad L_Y = \begin{array}{|c} \diagdown \\ \hline \end{array}, \quad R_Y = \begin{array}{|c} \hline \diagup \\ \end{array}, \quad \text{diag}(L_Y) = (1, \dots, 1)$$

Dann gilt für den QR–Algorithmus mit $\mu_k \equiv 0$

$$\alpha_{ij}^{(k)} \xrightarrow{k \rightarrow \infty} 0 \quad \forall i, j \text{ mit } i > j, \quad \alpha_{ii}^{(k)} \xrightarrow{k \rightarrow \infty} \lambda_i$$

Mit $q := \max_{i>j} \left| \frac{\lambda_i}{\lambda_j} \right|$ gilt genauer $\alpha_{ij}^{(k)} = \mathcal{O}(q^k)$ $i < j$, $\alpha_{ii}^{(k)} = \lambda_i + \mathcal{O}(q^k)$

□

<<

Beweis:

1. Sei $X := Y^{-1}$, also $A = X\Lambda Y$ und daher

$$A^k = X\Lambda^k Y = X\Lambda^k L_Y \Lambda^{-k} \Lambda^k R_Y$$

Sei

$$X\Lambda^k L\Lambda^{-k} = U_k \hat{R}_k$$

eine QR–Zerlegung mit $\hat{\rho}_{ii}^{(k)} > 0 \quad i = 1, \dots, n$ ²

Dann wird

$$A^k = U_k \underbrace{\hat{R}_k \Lambda^k R_Y}_{=: R_k^*} = U_k R_k^*$$

eine QR–Zerlegung und andererseits ist

$$A^k = \tilde{Q}_{k-1} \tilde{R}_{k-1}$$

ebenfalls eine QR–Zerlegung. Also (die QR–Zerlegung ist eindeutig bis auf unitäre Diagonaltransformation)

$$\exists \Theta_k \in \mathbb{C}^{n \times n} : \quad |\Theta_k| = I, \quad \tilde{Q}_{k-1} = U_k \Theta_k, \quad \tilde{R}_{k-1} = \bar{\Theta}_k R_k^*$$

²Diese Zerlegung ist eindeutig bestimmt und \hat{R}_k hängt reell differenzierbar von den Real- und Imaginärteilen der Matrix ab (Cholesky–Faktor)

2. Wir analysieren das Grenzverhalten von $\Lambda^k L_Y \Lambda^{-k}$.

Es ist mit $L_Y = (l_{ij})$

$$(\Lambda^k L_Y \Lambda^{-k})_{ij} = \begin{cases} 0 & i < j \\ 1 & i = j \\ \mathcal{O}(|\lambda_i/\lambda_j|^k) & i > j \end{cases} \quad (1.17)$$

also

$$\Lambda^k L_Y \Lambda^{-k} = I + \mathcal{O}(q^k)$$

Mit $X = Q_X R_X$ QR-Zerlegung mit $\rho_{ii}^{(x)} > 0$ folgt

$$X \Lambda^k L_Y \Lambda^{-k} = Q_X R_X (I + \mathcal{O}(q^k)) = U_k \hat{R}_k$$

und daher $\hat{R}_k = R_X + \mathcal{O}(q^k)$, $U_k = Q_X + \mathcal{O}(q^k)$

(denn der Cholesky-Faktor einer positiv definiten Matrix hängt reell differenzierbar von den $2n^2$ Real- und Imaginärteilen der Matrix ab),

3. Grenzverhalten von $\Theta_k A_k \bar{\Theta}_k$

$$\begin{aligned} \Theta_k A_k \bar{\Theta}_k &= \Theta_k \tilde{Q}_{k-1}^H A \tilde{Q}_{k-1} \bar{\Theta}_k = U_k^H A U_k \\ &= (Q_X + \mathcal{O}(q^k))^H A (Q_X + \mathcal{O}(q^k)) \\ &= (R_X X^{-1} + \mathcal{O}(q^k)) X \Lambda X^{-1} (X R_X^{-1} + \mathcal{O}(q^k)) \\ &\quad (\text{wegen } Q_X^H = Q_X^{-1}) \\ &= R_X \Lambda R_X^{-1} + \mathcal{O}(q^k) \end{aligned}$$

Die Diagonalelemente von $R_X \Lambda R_X^{-1}$ sind aber gerade $\lambda_i \Rightarrow$ Beh.

□

Bemerkung 1.7.1

- a) Man kann auf die Voraussetzung der Existenz einer Dreieckszerlegung von Y verzichten. Es existiert stets eine Dreieckszerlegung

$$PY = L_Y R_Y, \quad P = \begin{pmatrix} e_{\pi_1}^T \\ \vdots \\ e_{\pi_n}^T \end{pmatrix}$$

mit $l_{ij} = 0$ falls $i > j$ und $\pi_i < \pi_j$

(zugehörige Pivotstrategie: erstes Element $\neq 0$ in der jeweiligen Spalte wird Pivot)

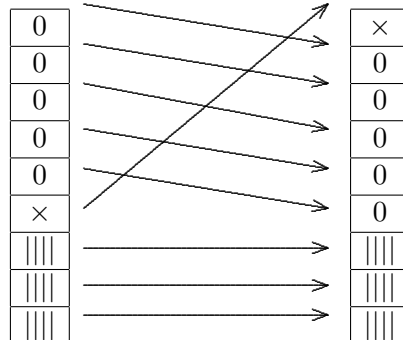


Abbildung 1.7.1

Setzt man dies entsprechend im Beweis von Satz 1.7.4 ein, dann ergibt sich $\alpha_{ii}^{(k)} \rightarrow \lambda_{\pi_i}$, die übrigen Aussagen bleiben unverändert.

- b) Durch eingeschleppte Rundungsfehler werden mehrfache Eigenwerte in der Praxis aufgelöst zu clustern dicht benachbarter Eigenwerte. Solche cluster werden durch Spektralschiebung aufgelöst in Eigenwerte von unterschiedlichem Betrag, abgesehen den Fall konjugiert komplexer Eigenwerte und reeller Verschiebungen:

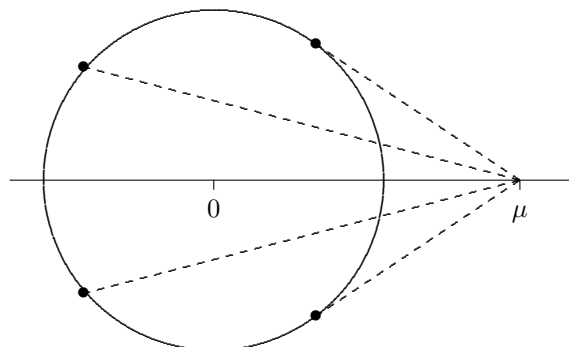


Abbildung 5.6.2

Bei einem solchen Eigenwertcluster nahe bei null sind die Quotienten $|\lambda_i/\lambda_j|$ stets deutlich von 1 verschieden.

- c) Sieht man den obigen Beweis noch einmal durch, so erkennt man, daß im Falle $|\lambda_1| \geq \dots \geq |\lambda_{n-2}| > |\lambda_{n-1}| = |\lambda_n|$ jedenfalls $\alpha_{n,j}^{(k)} \rightarrow 0$, $\alpha_{n-1,j} \rightarrow 0$ für $j = 1, \dots, n-2$. Man kann dann λ_n und λ_{n-1} aus der rechten unteren 2×2 Untermatrix approximieren. Für den Fall konjugiert komplexer Eigenwerte einer reellen Matrix gibt es eine in rein reeller Rechnung verlaufende “Doppelshift–Technik”, die auch dann für Konvergenzbeschleunigung sorgt.

□

>>

Aufgrund der Bemerkungen 1.5.1 a) – c) folgt, daß in der Praxis der QR–Algorithmus bei geeigneten Zusatzmaßnahmen für jede Matrix konvergiert. Wenn die Elemente der letzten Zeile außerhalb der Diagonale hinreichend klein geworden sind, beginnt man mit der Shift–Technik, wodurch sich die Konvergenz enorm beschleunigt.³ Sind die Elemente der letzten Zeile praktisch zu null geworden, kann man die QR–Transformation der linken oberen $(n-1) \times (n-1)$ Untermatrix zur Konstruktion der QR–Transformation der vollen Matrix benutzen gemäß

$$\hat{Q} \in \mathbb{C}^{(n-1) \times (n-1)} \longmapsto \left(\begin{array}{c|c} \hat{Q} & 0 \\ \hline 0 & 1 \end{array} \right) \in \mathbb{C}^{n \times n}$$

Diese Technik benötigt man aber nur, wenn man die Eigenvektoren der Grenzdreiecksmatrix (und daraus alle Eigenvektoren von A) bestimmen will. Sonst kann man einfach die Dimension des Problems verkleinern.

³Die Größenordnung der Außerdiagonalelemente quadriert sich pro Schritt. Bei hermiteschen Matrizen ist die Konvergenz sogar normalerweise von dritter Ordnung.

Beispiel 1.7.3 Hier werden die Eigenwerte einer symmetrischen Matrix mit einem Eigenwertcluster bei 1000 berechnet. Das benutzte Verfahren ist eine Variante, das QL-Verfahren, bei dem eine untere Dreiecksmatrix aus einer unteren Hessenberg- bzw. Tridiagonalmatrix erzeugt wird. Das maßgebliche Element, das zu Null gemacht wird, ist also $a_{1,2}$. Die Eingabematrix wird also zunächst wie in Abschnitt 5.2 beschrieben auf Tridiagonalgestalt gebracht. Dies ist hier nicht wiedergegeben.

Matrix A :

```

row/column  1          2          3          4          5
  1  .6110000D+03  .1960000D+03  -.1920000D+03  .4070000D+03  -.8000000D+01
  2  .1960000D+03  .8990000D+03  .1130000D+03  -.1920000D+03  -.7100000D+02
  3  -.1920000D+03  .1130000D+03  .8990000D+03  .1960000D+03  .6100000D+02
  4  .4070000D+03  -.1920000D+03  .1960000D+03  .6110000D+03  .8000000D+01
  5  -.8000000D+01  -.7100000D+02  .6100000D+02  .8000000D+01  .4110000D+03
  6  -.5200000D+02  -.4300000D+02  .4900000D+02  .4400000D+02  -.5990000D+03
  7  -.4900000D+02  -.8000000D+01  .8000000D+01  .5900000D+02  .2080000D+03
  8  .2900000D+02  -.4400000D+02  .5200000D+02  -.2300000D+02  .2080000D+03

```

```

row/column  6          7          8
  1  -.5200000D+02  -.4900000D+02  .2900000D+02
  2  -.4300000D+02  -.8000000D+01  -.4400000D+02
  3  .4900000D+02  .8000000D+01  .5200000D+02
  4  .4400000D+02  .5900000D+02  -.2300000D+02
  5  -.5990000D+03  .2080000D+03  .2080000D+03
  6  .4110000D+03  .2080000D+03  .2080000D+03
  7  .2080000D+03  .9900000D+02  -.9110000D+03
  8  .2080000D+03  -.9110000D+03  .9900000D+02

```

Berechnete Eigenwerte und Fehler

```

lam[ 1] = -.10200490184300D+04  lam_exakt[ 1]-lam[ 1]  .9095D-12
lam[ 2] = -.14085954624932D-12  lam_exakt[ 2]-lam[ 2]  .1409D-12
lam[ 3] = .98048640721745D-01  lam_exakt[ 3]-lam[ 3]  -.1731D-12
lam[ 4] = .100000000000000D+04  lam_exakt[ 4]-lam[ 4]  .1137D-12
lam[ 5] = .100000000000000D+04  lam_exakt[ 5]-lam[ 5]  -.3411D-12
lam[ 6] = .10199019513593D+04  lam_exakt[ 6]-lam[ 6]  .0000D+00
lam[ 7] = .102000000000000D+04  lam_exakt[ 7]-lam[ 7]  -.5684D-12
lam[ 8] = .10200490184300D+04  lam_exakt[ 8]-lam[ 8]  -.2274D-12

```

Die Iteration ist im Folgenden dargestellt. k zählt die Iterationen pro Eigenwert und $n(k)$ gibt die laufende Dimension des Problems an, die sich mit jedem akzeptierten Eigenwert verringert. Man erkennt die durch die Shifts erzeugte ausserordentliche Konvergenzgeschwindigkeit.

Iterationsprotokoll :

```

  k  a(1,2)(k)  n(k)
  0  .5843D-06  8
  1  .3034D-13  8
Eigenwertnaeherung akzeptiert !
  0  -.1304D-05  7
  1  -.1621D-28  7
Eigenwertnaeherung akzeptiert !
  0  .1088D-02  6
  1  -.1030D-06  6
  2  .8160D-20  6
Eigenwertnaeherung akzeptiert !
  0  .3903D-02  5
  1  .2011D-12  5
  2  .2529D-51  5
Eigenwertnaeherung akzeptiert !
  0  -.1663D-10  4
  1  .1497D-47  4
Eigenwertnaeherung akzeptiert !
  0  .1877D-01  3

```


Satz 1.7.6 Sind Q und W zwei unitäre Matrizen, die in ihrer ersten Spalte übereinstimmen und sind sowohl $Q^H A Q$ als auch $W^H A W$ obere Hessenbergmatrizen, dann stimmen $Q^H A Q$ und $W^H A W$ bis auf eine diagonale Ähnlichkeitstransformation mit einer Diagonalmatrix D , $|D| = I$ überein. (Francis) Beweis: Übg. \square

Man bestimmt nun eine Givensmatrix Ω_1 so, daß sie die erste Spalte von $T - \mu I$ in ein Vielfaches des ersten Einheitsvektors überführt. Diese wendet man nun auf T als Ähnlichkeitstransformation an. (nicht $T - \mu I$) Dabei entsteht ein Element ungleich null in den Positionen (1,3) und (3,1). Dieses treibt man durch eine weitere Ähnlichkeitstransformation mit einer Givensrotation für Zeilen/Spaltenpaar (2,3) auf die Position (2,4) bzw. (4,2) und so weiter, bis es verschwindet und die Tridiagonal- bzw. Hessenbergstruktur wiederhergestellt ist. Dann hat man genau den Zusammenhang wie in vorstehendem Satz, d.h. bis auf eine triviale Ähnlichkeitstransformation den Übergang von einer Iterierten zur nächsten im QR-Verfahren mit Shift. Der Rundungsfehlereinfluss der Shifts auf die Genauigkeit der Eigenwerte wird so vermieden.

1.8 Das Lanczos-Verfahren

Beim v. Mises-Verfahren, dem Wielandt-Verfahren und bei der simultanen Vektoriteration wird der k -te Iterationsschritt nur mit Hilfe der Information aus dem $(k-1)$ -ten Iterationsschritt ausgeführt. Die Grundidee des Lanczos-Verfahrens ist es, die mit der Folge $x^{(0)}, x^{(1)}, \dots, x^{(k)}$ im v. Mises-Verfahren bzw. Wielandt-Verfahren gewonnene Information möglichst gut auszunutzen. Wir betrachten wieder den Fall eines symmetrischen Eigenwertproblems

$$Ax = \lambda x, \quad A = A^T \in \mathbb{R}^{n \times n}.$$

Die Eigenwerte von $Q_j^T A Q_j$ sollen als Näherungen für die Eigenwerte von A dienen. Dabei ist Q_j eine Orthonormalbasis des von $x^{(0)}, Ax^{(0)}, \dots, A^{j-1}x^{(0)}$ aufgespannten Raumes. Die erste Spalte von Q_j wird gleich $x^{(0)} / \|x^{(0)}\|_2$ gesetzt und allgemein gilt mit $X_j = (x^{(0)}, \dots, x^{(j-1)})$:

$$X_j = Q_j R_j \text{ mit einer oberen Dreiecksmatrix } R_j \text{ und } Q_j^T Q_j = I.$$

Die hohe Effizienz des Lanczos-Verfahrens ist dadurch bedingt, daß zum einen die größten bzw. kleinsten Eigenwerte von $Q_j^T A Q_j$ sehr schnell gegen die größten bzw. kleinsten Eigenwerte von A konvergieren (falls $x^{(0)}$ geeignet gewählt ist), zum anderen die Spalten von Q_j sukzessiv durch eine dreigliedrige Rekursion berechnet werden können und $Q_j^T A Q_j = T_j$ Tridiagonalgestalt erhält. Dies bedeutet, daß das Eigenwertproblem für T_j sehr effizient gelöst werden kann. Dabei treten nur Matrix-Vektorprodukte mit der Matrix A auf, sodaß auch Dünnbesetztheit dieser Matrix voll ausgenutzt werden kann. Das Verfahren wird ausschliesslich für hochdimensionale Probleme (Eigenwertprobleme diskretisierter Differentialgleichungen) mit Dimensionen bis 100000 und mehr angewendet. Man muss allerdings beachten, daß die Matrizen Q_j voll besetzt sind. Insoweit stellt der verfügbare Hauptspeicher eine gewisse Grenze für die behandelbaren

Probleme dar. Es gilt dazu

Satz 1.8.1 Sei A eine reelle symmetrische $n \times n$ -Matrix, $x^{(0)} \neq 0 \in \mathbb{R}^n$ beliebig und $x^{(i)} = A^i x^{(0)}$, $X_j = (x^{(0)}, \dots, x^{(j-1)})$, $Q_j = (q^{(1)}, \dots, q^{(j)})$, sowie für $i = 1, 2, \dots$

$$r^{(i+1)} = Aq^{(i)} - \alpha_i q^{(i)} - \beta_{i-1} q^{(i-1)}$$

mit

$$\begin{aligned}\alpha_i &= (q^{(i)})^T Aq^{(i)}, \\ \beta_i &= \|r^{(i+1)}\|_2,\end{aligned}$$

wo $q^{(1)} = x^{(0)} / \|x^{(0)}\|_2$, $q^{(0)} = 0$, $\beta_0 = 1$,

$$q^{(i+1)} = r^{(i+1)} / \beta_i \quad \text{für } \beta_i \neq 0.$$

Dann gilt: Der Algorithmus ist durchführbar, solange $\beta_i \neq 0$. In diesem Falle ist

$$Q_i^T A Q_i = \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{i-1} \\ 0 & & & \beta_{i-1} & \alpha_i \end{bmatrix} = T_i,$$

$Q_i^T Q_i = I$, $X_i = Q_i R_i$, R_i obere Dreiecksmatrix,
d.h. die $q^{(j)}$ bilden eine Orthonormalbasis des von den $x^{(j)}$ aufgespannten Raumes. \square

<<

Beweis: Mit dem künstlich eingeführten Vektor $q^{(0)}$ gilt

$$\beta_{j-1} q^{(j-1)} + \alpha_j q^{(j)} + \beta_j q^{(j+1)} = Aq^{(j)}, \quad j = 1, \dots, i$$

und deshalb

$$Q_i T_i = A Q_i + \beta_i q^{(i+1)} e_i^T.$$

Wir zeigen nun zuerst, daß die $q^{(k)}$ paarweise orthogonal sind. Dann folgt bereits

$$Q_i^T A Q_i = T_i.$$

Die Orthogonalität der $q^{(k)}$ wird induktiv gezeigt. Wegen

$$(q^{(1)})^T q^{(2)} = (q^{(1)})^T A q^{(1)} - \alpha_1 (q^{(1)})^T q^{(1)} = 0$$

haben wir eine Induktionsverankerung. Seien nun $q^{(1)}, \dots, q^{(k)}$ paarweise orthogonal und nor-

miert. Dann wird für $r^{(k+1)} \neq 0$

$$\begin{aligned} (q^{(j)})^T q^{(k+1)} &= \frac{1}{\|r^{(k+1)}\|} (q^{(j)})^T (Aq^{(k)} - \alpha_k q^{(k)} - \beta_{k-1} q^{(k-1)}) \\ &\quad (\text{drücke } (Aq^{(j)})^T \text{ mittels der Rekursion durch } r^{(j+1)}, q^{(j)}, q^{(j-1)} \text{ aus}) \\ &= \frac{1}{\|r^{(k+1)}\|} \left((r^{(j+1)} + \alpha_j q^{(j)} + \beta_j q^{(j-1)})^T q^{(k)} - \alpha_k (q^{(j)})^T q^{(k)} - \beta_k (q^{(j)})^T q^{(k-1)} \right) \\ &= 0 \quad \text{für } j+1 < k, \text{ also für } j < k-1. \end{aligned}$$

Es bleiben die Fälle $j = k$ und $j = k-1$. Nun ist wegen der Normierung der $q^{(j)}$ und der Definition von β_{k-1}

$$\begin{aligned} (q^{(k)})^T q^{(k+1)} &= \frac{1}{\|r^{(k+1)}\|} \left((q^{(k)})^T Aq^{(k)} - \alpha_k (q^{(k)})^T q^{(k)} - \beta_k (q^{(k)})^T q^{(k-1)} \right) \\ &= 0 \text{ nach Definition von } \alpha_k \text{ und Induktionsvoraussetzung und} \\ (q^{(k-1)})^T q^{(k+1)} &= \frac{1}{\|r^{(k+1)}\|} \left((q^{(k-1)})^T Aq^{(k)} - \alpha_k (q^{(k-1)})^T q^{(k)} - \beta_{k-1} \right) \\ &= \frac{1}{\|r^{(k+1)}\|} (\|r^{(k)}\| (q^{(k)})^T q^{(k)} - \beta_{k-1}) \\ &= 0. \end{aligned}$$

Man beachte daß

$$\|r^{(k)}\| = \beta_{k-1} \text{ und } (q^{(k)})^T q^{(k)} = 1$$

Ferner ist

$$q^{(1)} = x^{(0)} / \|x^{(0)}\|.$$

Sei nun als Induktionsvoraussetzung

$$q^{(k)} \in \text{span}(x^{(0)}, Ax^{(0)}, \dots, A^{k-1}x^{(0)}).$$

Dann ist nach dem Bildungsgesetz für $q^{(k+1)}$

$$q^{(k+1)} \in \text{span}(x^{(0)}, Ax^{(0)}, \dots, A^{k-1}x^{(0)}) \cup \text{span}(Ax^{(0)}, A^2x^{(0)}, \dots, A^kx^{(0)})$$

und somit

$$Q_i = X_i \tilde{R}_i \text{ mit einer invertierbaren oberen Dreiecksmatrix } \tilde{R}_i$$

solange $\|r^{(i+1)}\| \neq 0$. □

>>

Spätestens für $i = n$ bricht das Verfahren (theoretisch) ab mit $r^{(n+1)} = 0$, d.h. $\beta_n = 0$. In diesem Fall wäre A durch eine orthonormale Ähnlichkeitstransformation auf Tridiagonalgestalt transformiert. Zu diesem Zweck ist das Verfahren aber ganz ungeeignet, weil aufgrund der Rundungsfehlereinflüsse in der Praxis die Matrix Q_j sehr schnell ihre Orthonormalität verliert. Dennoch bleibt die Tatsache gültig, daß für maßvoll kleines j die größten bzw. kleinsten Eigenwerte der tatsächlich berechneten Tridiagonalmatrix $\hat{T}_i = \text{tridiag}(\hat{\beta}_{i-1}, \hat{\alpha}_i, \hat{\beta}_i)$, wo $\hat{\alpha}_i, \hat{\beta}_i$ die berechneten Größen bezeichnen, die größten bzw. kleinsten Eigenwerte von A sehr gut approximieren, wenn $x^{(0)}$ geeignet gewählt

ist. Selbstverständlich kann auch bei exakter Rechnung der Algorithmus in Abhängigkeit von $x^{(0)}$ vorzeitig abbrechen, z.B. wenn $x^{(0)}$ ein Eigenvektor von A ist, schon im ersten Schritt. Durch eine geeignete Umspeicherung während der Berechnung kann man den Algorithmus mit nur zwei Hilfsvektoren der Länge n durchführen, d.h. er ist auch nur sehr wenig speicheraufwendig.

Algorithmus:

$$v := \frac{x^{(0)}}{\|x^{(0)}\|} = q^{(1)},$$

$$u := 0,$$

$$\beta_0 := 1,$$

$$j := 0.$$

Solange $\beta_j \neq 0$:

$$\text{Für } i = 1, \dots, n : \left\{ \begin{array}{l} \gamma := u_i; \quad u_i := v_i/\beta_j; \quad v_i := -\gamma\beta_j. \end{array} \right\}$$

$$\text{wenn } j \geq 1 \quad q^{(j+1)} := u; v := Au + v,$$

$$j := j + 1,$$

$$\alpha_j := u^T v,$$

$$v := v - \alpha_j u,$$

$$\beta_j := \|v\|_2.$$

Die Matrix A wird dabei niemals geändert. Man benötigt lediglich eine Routine für die Ausführung der Matrix-Vektormultiplikation Ax , wobei man die Besetzungsstruktur von A voll ausnutzen kann. Im Zusammenhang mit der Methode der Finiten Elemente genügt es z.B., die einzelnen Elementsteifigkeitsmatrizen vorliegen zu haben, anstelle der um Größenordnungen aufwendiger zu speichernden Gesamtsteifigkeitsmatrix, um diese Operation auszuführen.

Wenn man die Operation Au ersetzt durch die Gleichungslösung $Aw = u$, hat man das Lanczos-Verfahren in Verbindung mit der inversen Iteration.

Wie bereits erwähnt, dienen die Eigenwerte der aus den berechneten Werten α_i, β_i gebildeten Tridiagonalmatrix

$$T_j = \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{j-1} \\ 0 & & & \beta_{j-1} & \alpha_j \end{bmatrix}$$

für jeden Wert von j (d.h. für jeden weiteren Lanczos-Schritt) als Näherungen für einige Eigenwerte von A .

Für das Verfahren ist wesentlich, daß man Schätzungen für die Genauigkeit dieser Näherungen aus dem Eigenwertproblem von T_j selbst erhält, samt Näherungen für die dazugehörigen Eigenvektoren von A . Dies ist der Inhalt des folgenden Satzes.

Satz 1.8.2 Sei V_j eine orthonormierte Eigenvektor-Matrix von T_j :

$$V_j^T T_j V_j = \text{diag} [\Theta_1, \dots, \Theta_j],$$

und Y_j sei definiert durch

$$Y_j = Q_j V_j = [y_1, \dots, y_j].$$

Dann gilt

$$\|Ay_i - \Theta_i y_i\|_2 = |\beta_j| |v_{ji}|.$$

Bew.: Übung □

Ist also $v_{ji}\beta_j$ „klein“, dann bedeutet dies, daß Θ_i eine gute Eigenwertschätzung für A ist mit zugehöriger Eigenvektorschätzung y_i . Dies ist natürlich insbesondere dann der Fall, wenn β_j selbst sehr klein ist. Letzteres tritt allerdings in der Praxis selten auf. Dagegen wird $|v_{ji}|$ oft sehr schnell klein. Einen Hinweis auf die Konvergenzgeschwindigkeit der Eigenwertschätzungen liefert

Satz 1.8.3 Die reell-symmetrische Matrix A besitze die Eigenwerte $\lambda_1 > \dots > \lambda_n$ mit den zugehörigen orthonormierten Eigenvektoren z_1, \dots, z_n . $\Theta_{1,j} > \dots > \Theta_{j,j}$ seien die Eigenwerte von T_j . Ferner seien φ_1, ϱ_1 sowie φ_n, ϱ_n definiert durch

$$\begin{aligned} |\cos \varphi_1| &= |(q^{(1)})^T z_1|, & \varrho_1 &= (\lambda_1 - \lambda_2)/(\lambda_2 - \lambda_n), \\ |\cos \varphi_n| &= |(q^{(1)})^T z_n|, & \varrho_n &= (\lambda_{n-1} - \lambda_n)/(\lambda_1 - \lambda_{n-1}). \end{aligned}$$

und es gelte

$$x^{(0)} = \sum_{i=1}^n \gamma_i z_i \text{ mit } \gamma_1, \gamma_n \neq 0.$$

Dann gilt für $1 \leq j \leq n$

$$\begin{aligned} \lambda_1 &\geq \Theta_{1,j} \geq \lambda_1 - (\lambda_1 - \lambda_n)(\tan \varphi_1/p_{j-1}(1 + 2\varrho_1))^2, \\ \lambda_n &\leq \Theta_{j,j} \leq \lambda_n + (\lambda_1 - \lambda_n)(\tan \varphi_n/p_{j-1}(1 + 2\varrho_n))^2. \end{aligned}$$

Dabei ist p_j das Tschebyscheffpolynom erster Art von genauem Grad j mit der Normierung $p_j(1) = 1$. □

Man erkennt, daß im Fall gut separierter Eigenwerte und $|\tan \varphi_1|, |\tan \varphi_n|$ „klein“ (d.h. $x^{(0)}$ hat einen genügend großen Anteil in der Richtung von z_1 bzw. z_n), die Fehlerschranken sehr schnell klein werden, weil die Tschebyscheffpolynome außerhalb des Intervalls $[-1, 1]$ sehr schnell anwachsen.

Auswertungen dieser Schranken zeigen, daß das Lanczos-Verfahren bezüglich seiner Näherungsgüte der direkten Vektoriteration hoch überlegen ist.

Leider werden die theoretisch so günstigen Eigenschaften des Lanczos-Verfahrens durch die extreme Rundungsfehlerempfindlichkeit des Verfahrens stark nivelliert. Diese Rundungsfehlerempfindlichkeit zeigt sich darin, daß die tatsächlich berechneten Vektoren

$r^{(i)}$ sehr schnell ihre Orthogonalität verlieren. Die Orthogonalität ist aber für alle Aussagen über das Verfahren entscheidend.

Beispiel 1.8.1 *Es sollen die kleinsten Eigenwerte der 50×50 5-Bandmatrix*

$$A = \begin{bmatrix} 1 & -2 & 1 & \cdots & \cdots & \cdots & 0 \\ -2 & 5 & -4 & 1 & & & \vdots \\ 1 & -4 & 6 & -4 & 1 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 1 & -4 & 6 & -4 & 1 \\ \vdots & & & 1 & -4 & 5 & -2 \\ 0 & \cdots & \cdots & \cdots & 1 & -2 & 1 \end{bmatrix}$$

berechnet werden. Dieses Problem entsteht aus der Diskretisierung des Differentialgleichungseigenwertproblems

$$\begin{aligned} y^{(4)} &= \lambda y, \\ y''(0) &= y'''(0) = y''(1) = y'''(1) = 0. \end{aligned}$$

(Balkenbiegung eines beidseitig elastisch gelagerten Balkens).
Die 10 kleinsten Eigenwerte von A sind

$$\begin{aligned} \lambda_{50} &= 1.43894475 \cdot 10^{-5}, \\ \lambda_{49} &= 2.29794694 \cdot 10^{-4}, \\ \lambda_{48} &= 1.15966134 \cdot 10^{-3}, \\ \lambda_{47} &= 3.6489003 \cdot 10^{-3}, \\ \lambda_{46} &= 8.85782128 \cdot 10^{-3}, \\ \lambda_{45} &= 0.0182399992, \\ \lambda_{44} &= 0.0335144259, \\ \lambda_{43} &= 0.0566323917, \\ \lambda_{42} &= 0.0897396256, \\ \lambda_{41} &= 0.1351343. \end{aligned}$$

Es wurde das Lanczos-Verfahren in Verbindung mit der inversen Iteration verwendet. Als Startvektor diente dabei $x^{(0)} = [1, 2, 3, \dots]^T$. Schon im dritten Lanczosschritt ergeben sich die Näherungen

$$\begin{aligned} \theta_3 &= \underline{1.43899796} \cdot 10^{-5} && \text{für } \lambda_{50}, \\ \theta_2 &= \underline{2.40714104} \cdot 10^{-4} && \text{für } \lambda_{49}, \\ \theta_1 &= \underline{0.0154333215} && (\text{für } \lambda_{45}). \end{aligned}$$

Die korrekten Stellen sind unterstrichen.

Ferner ist

$$\|I - \hat{Q}_3^T \hat{Q}_3\|_F = 1.75 \cdot 10^{-8},^5$$

d.h. die ersten drei $\hat{q}^{(i)}$ erfüllen die Forderung der Orthonormalität im Rahmen der Rechengenauigkeit von 10 Stellen recht gut ($\hat{q}^{(i)}$ bezeichnet die tatsächlich berechneten Werte).

Im 5. Lanczos-Schritt haben wir

$$\begin{aligned}\Theta_5 &= \underline{1.43899751} \cdot 10^{-5} && \text{für } \lambda_{50}, \\ \Theta_4 &= \underline{2.29795209} \cdot 10^{-4} && \text{für } \lambda_{49}, \\ \Theta_3 &= \underline{1.16123194} \cdot 10^{-3} && \text{für } \lambda_{48}, \\ \Theta_2 &= 4.51726268 \cdot 10^{-3}, \\ \Theta_1 &= 0.125580791, \\ \|I - \hat{Q}_5^T \hat{Q}_5\|_F &= 3.75 \cdot 10^{-4}.\end{aligned}$$

Im 7. Lanczos-Schritt schließlich wird

$$\begin{aligned}\Theta_7 &= \underline{1.43899751} \cdot 10^{-5} && \text{für } \lambda_{50}, \\ \Theta_6 &= \underline{1.43901098} \cdot 10^{-5} && \text{für } \lambda_{50} \quad (!), \\ \Theta_5 &= \underline{2.29795195} \cdot 10^{-4} && \text{für } \lambda_{49}, \\ \Theta_4 &= \underline{1.15966716} \cdot 10^{-3} && \text{für } \lambda_{48}, \\ \Theta_3 &= \underline{3.68169396} \cdot 10^{-3} && \text{für } \lambda_{47}, \\ \Theta_2 &= \underline{0.0116532546} && \text{für } \lambda_{45}, \\ \Theta_1 &= 0.257072226, \\ \|I - \hat{Q}_7^T \hat{Q}_7\|_F &= 1.414 \quad (!),\end{aligned}$$

d.h. die Matrix \hat{Q}_7 ist nun auch nicht annäherungsweise orthonormal. Gleichzeitig tritt eine Näherung für λ_{50} als Doublette auf, obwohl λ_{50} ein einfacher, gut separierter Eigenwert von A ist.

Dies ist typisch für das Lanczos-Verfahren unter Rundungsfehlereinfluß. Das Erkennen solcher falschen Doubletten stellt eine besondere Schwierigkeit für die Anwendung des Verfahrens dar. Man erkennt auch, daß die Eigenwerte nicht alle systematisch angenähert werden, z.B. fehlt eine Näherung für λ_{46} , während eine für λ_{45} vorliegt. Dies liegt am Startvektor. \square

Den Verlust der Orthogonalität bei den $\hat{q}^{(i)}$ könnte man dadurch ausgleichen, daß man jedes berechnete $\hat{q}^{(i)}$ sofort bezüglich aller zuvor berechneten Vektoren $\hat{q}^{(1)}, \dots, \hat{q}^{(i-1)}$ orthogonalisiert. Dies würde aber den Rechen- und auch den Speicherzugriffsaufwand für das Verfahren (die $\hat{q}^{(i)}$ wird man bei großem n gewöhnlich auf einem Hintergrundspeicher halten) enorm erhöhen. Um das Entstehen falscher Doubletten zu vermeiden genügt es, $\hat{q}^{(i)}$ bzgl. der bereits mit hinreichender Genauigkeit gefundenen Eigenvektoren von A zu orthogonalisieren (sogenannte selektive Orthogonalisierung). Als „hinreichend“ genau definiert man dabei einen Defekt in der Einsetzprobe von der Größenordnung der halben Rechengenauigkeit, d.h.

$$\|Ay_i - \Theta_i y_i\|_2 \leq \sqrt{\varepsilon} \|A\|_F,$$

wobei nur die y_i getestet werden, für die

$$|\beta_j| |v_{ji}| \leq \sqrt{\varepsilon} \|A\|_F$$

im j -ten Lanczos-Schritt gilt. Natürlich muß man dazu in jedem Schritt das vollständige Eigenwert / Eigenvektor-Problem der Matrizen T_j lösen, was aber nur wenig aufwendig ist.

⁵Ist A eine beliebige Matrix, so ist $\|A\|_F := (\text{Sp}(AA^H))^{1/2}$

Beispiel 1.8.2 Wir betrachten die Aufgabenstellung von Beispiel 1.8.1, jetzt mit selektiver Orthogonalisierungs-Testgröße

$$\|Ay_i - \Theta_i y_i\| \leq 16 \cdot 10^{-5}.$$

Im 10-ten Lanczos-Schritt ist

$$\begin{aligned} \|I - \hat{Q}_{10}^T \hat{Q}_{10}\|_F &= 4.631 \cdot 10^{-4}, \\ \Theta_{10} &= \underline{1.43899752} \cdot 10^{-5}, \\ \Theta_9 &= \underline{2.29795196} \cdot 10^{-4}, \\ \Theta_8 &= \underline{1.15966203} \cdot 10^{-3}, \\ \Theta_7 &= \underline{3.64890097} \cdot 10^{-3}, \\ \Theta_6 &= \underline{8.85782268} \cdot 10^{-3}, \\ \Theta_5 &= \underline{0.018240746}, \\ \Theta_4 &= \underline{0.0337289387}, \\ \Theta_3 &= \underline{0.0651717673}, \\ \Theta_2 &= \underline{0.185803438}, \\ \Theta_1 &= \underline{1.74281859}. \end{aligned}$$

Mit einem Aufwand, der 10 Schritten der einfachen inversen Iteration im wesentlichen entspricht, sind bereits die sieben kleinsten Eigenwerte von A mit guter Genauigkeit gefunden. \square

1.9 Allgemeine Eigenwertprobleme

Neben dem speziellen Eigenwertproblem tritt häufig auch das allgemeine Eigenwertproblem

$$A x = \lambda B x, \quad x \neq 0, \quad (1.18)$$

auf, allerdings meistens für $A = A^H$, $B = B^H$, B positiv definit. In den Anwendungen treten auch noch allgemeinere Aufgaben auf, etwa

$$(K + \lambda C - \lambda^2 M) x = 0, \quad x \neq 0, \quad (1.19)$$

oder sogar

$$(A - M(\lambda)) x = 0, \quad x \neq 0,$$

mit einer von λ abhängenden Matrix $M(\lambda)$.

Wir wollen zunächst (1.18) im Falle allgemeiner komplexer $n \times n$ Matrizen A , B betrachten. Solange B invertierbar bleibt, kann (1.18) unmittelbar auf das spezielle Eigenwertproblem zurückgeführt werden:

$$A x = \lambda B x \Leftrightarrow B^{-1} A x = \lambda x.$$

Die explizite Durchführung dieser Transformation ist nur dann zu empfehlen, wenn die Dimension des Problems nicht groß und B nicht zu schlecht konditioniert ist. Wenn A

und B hermitisch sind und B positiv definit, wird man die Transformation mit Hilfe der Cholesky-Zerlegung von B bevorzugen, die die hermitische Struktur des Problems erhält:

$$\begin{aligned} A x &= \lambda B x, \quad B = L L^H \quad \Leftrightarrow \quad L^{-1} A (L^{-1})^H L^H x = \lambda L^H x \\ \Leftrightarrow \quad C y &= \lambda y \quad \text{mit} \quad y = L^H x, \quad C = C^H = L^{-1} A (L^{-1})^H. \end{aligned}$$

Zunächst wollen wir uns kurz mit dem allgemeinen Problem (1.18) befassen. Eine hinreichende und notwendige Bedingung zur Existenz von $x \neq 0$ in (1.18) ist ersichtlich

$$\det (A - \lambda B) = 0, \tag{1.20}$$

und dies ist wiederum ein Polynom vom Höchstgrad n in λ . Somit ist jede komplexe Zahl λ Lösung von (1.20) oder es gibt höchstens n solcher Werte. Der erste Fall kann durchaus eintreten, wie das Beispiel

$$A = \begin{bmatrix} 2 & 4 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix}$$

zeigt.

Wie beim speziellen Eigenwertproblem ist die Lösungsstruktur von (1.18) unmittelbar überschaubar, wenn A und B obere (oder untere) Dreiecksmatrizen sind. Gilt nämlich

$$a_{ij} = 0, \quad j < i \quad \text{und} \quad b_{ij} = 0, \quad j < i,$$

dann ist ersichtlich

$$\begin{aligned} \lambda \in \mathbb{C}, \quad \lambda \text{ beliebig, Lösung von (1.20), wenn } a_{ii} = b_{ii} = 0 \text{ für ein } i, \\ \lambda \in \{a_{ii}/b_{ii} : b_{ii} \neq 0\} \text{ sonst.} \end{aligned}$$

Nun ist mit unitärem Q und Z

$$\det (A - \lambda B) = 0 \quad \Leftrightarrow \quad \det (Q^H A Z - \lambda Q^H B Z) = 0.$$

Ferner gilt folgende Verallgemeinerung des Satzes von Schur:

Satz 1.9.1 Zu beliebigen $A, B \in \mathbb{C}^{n \times n}$ existieren unitäre Matrizen Q und Z , so daß

$$T = Q^H A Z \quad \text{und} \quad S = Q^H B Z \tag{1.21}$$

obere Dreiecksgestalt besitzen.

(Zum Beweis vergleiche etwa G.H. Golub, Ch. van Loan: Matrix Computations, J. Hopkins Press.) □

Der aus dem QR-Algorithmus hergeleitete *QZ-Algorithmus von Stewart und Moler* bestimmt iterativ eine Folge von unitären Transformationen, die die Transformation (1.21) annähert.

Im Folgenden beschränken wir uns auf den Fall reeller Matrizen A , B . Der QZ-Algorithmus beginnt mit einer vorbereitenden Transformation

$$A \mapsto \tilde{A} = Q^T A Z, \quad B \mapsto \tilde{B} = Q^T B Z,$$

so daß \tilde{A} eine obere Hessenbergmatrix und \tilde{B} eine obere Dreiecksmatrix wird. Diese vorbereitende Transformation besteht aus zwei Teilen: Zuerst wird B auf obere Dreiecksgestalt gebracht, etwa durch Householder-Transformationen, und die gleiche Transformation wird auf A angewendet. Dann werden in der Reihenfolge

$$(n, 1), (n-1, 1), \dots, (3, 1), (n, 2), \dots, (4, 2), \dots, (n, n-2)$$

jeweils durch eine Givenstransformation von links ein Element in A in null überführt und durch weitere Givenstransformationen von rechts ein dadurch in der Subdiagonale von B eingeführtes Element ungleich Null annulliert, ohne die Nullstruktur in A wieder zu zerstören, z.B.

$$A = \begin{bmatrix} 1 & -1 & 10 & -10 \\ 2 & 1 & 20 & 30 \\ 0 & 3 & 5 & 5 \\ 0 & 4 & -5 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 2 & -3 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 10 & -\frac{15}{4} \\ 0 & 0 & 0 & 5 \end{bmatrix},$$

$$\Omega_1 = \left[\begin{array}{cc|cc} 1 & 0 & & 0 \\ 0 & 1 & & \\ \hline & & 3/5 & 4/5 \\ 0 & & 4/5 & -3/5 \end{array} \right],$$

$$\Omega_1 A = \begin{bmatrix} 1 & -1 & 10 & -10 \\ 2 & 1 & 20 & 30 \\ 0 & 5 & -1 & 7 \\ 0 & 0 & 7 & 1 \end{bmatrix}, \quad \Omega_1 B = \begin{bmatrix} 1 & 2 & 2 & -3 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 6 & \frac{7}{4} \\ 0 & 0 & 8 & -6 \end{bmatrix},$$

$$\Omega_2 = \left[\begin{array}{cc|cc} 1 & 0 & & 0 \\ 0 & 1 & & \\ \hline & & 0.6 & 0.8 \\ 0 & & 0.8 & -0.6 \end{array} \right],$$

$$\Omega_1 A \Omega_2 = \begin{bmatrix} 1 & -1 & -2 & 14 \\ 2 & 1 & 36 & -2 \\ 0 & 5 & 5 & -5 \\ 0 & 0 & 5 & 5 \end{bmatrix}, \quad \Omega_1 B \Omega_2 = \begin{bmatrix} 1 & 2 & -1.2 & 3.4 \\ 0 & 1 & 3.2 & 2.6 \\ 0 & 0 & 5 & 3.75 \\ 0 & 0 & 0 & 10 \end{bmatrix}.$$

Der weitere Algorithmus geht von der Normalform

- A nichtzerfallende obere Hessenbergmatrix,
- B invertierbare Dreiecksmatrix

aus. Ist eines der Subdiagonalelemente von A null, zerfällt das allgemeine Eigenwertproblem in zwei solche kleinerer Dimension. Ist ein Diagonalelement von B null, kann man das Problem durch weitere Äquivalenztransformationen in eines mit einer zerfallenden Hessenbergmatrix A überführen. Der weitere Algorithmus ist im Prinzip der QR-Algorithmus für die Matrix AB^{-1} , d.h. für eine obere nichtzerfallende Hessenbergmatrix, da A obere nichtzerfallende Hessenbergmatrix und B^{-1} eine obere Dreiecksmatrix ist. Die Matrix AB^{-1} wird jedoch dabei nicht explizit gebildet, vielmehr arbeitet man stets mit orthonormalen Transformationen an A und B . Dies beruht auf den folgenden Zusammenhängen: Ist A Hessenbergmatrix und B invertierbare obere Dreiecksmatrix, so ist

$$Ax = \lambda Bx \Leftrightarrow AB^{-1}y = \lambda y \text{ mit } y = Bx .$$

AB^{-1} ist wieder eine obere Hessenbergmatrix. Nun gilt

Satz 1.9.2 *Ist A eine beliebige $n \times n$ Matrix und B eine nichtzerfallende obere Hessenbergmatrix sowie Q unitär, und*

$$B = Q^H A Q .$$

Dann ist B bis auf eine unitäre diagonale Ähnlichkeitstransformation und Q durch den entsprechenden diagonalen Faktor von rechts eindeutig bestimmt durch die erste Spalte von Q .

Im QZ-Algorithmus arbeitet man mit Matrizen A_k und B_k wie oben beschrieben. Man bestimmt nun den ersten Givenstransformationsschritt für die QR-Zerlegung der Matrix

$$C_k = A_k B_k^{-1} - \mu_k I$$

Dies ist aber zugleich auch der erste QR-Zerlegungsschritt für

$$A_k - \mu_k B_k$$

(ist also auch möglich sogar für singuläres B_k). Dies legt die erste Spalte einer unitären Matrix \tilde{Q}_k fest. Man wendet diese Transformation nun auf A_k und B_k an und sodann weitere Givenstransformationen von links und rechts, bis mit so definierten unitären Matrizen \tilde{Q}_k und Z_k wieder gilt

$$A_{k+1} \text{ obere Hessenbergmatrix und } B_{k+1} \text{ obere Dreiecksmatrix}$$

wo

$$A_{k+1} = \tilde{Q}_k A_k Z_k \quad B_{k+1} = \tilde{Q}_k B_k Z_k .$$

Dann ist nach obigem Satz \tilde{Q}_k bis auf unitäre Diagonaltransformation identisch mit dem Q_k aus einem QR-Schritt für C_k und somit gilt mit den obigen A_{k+1} und B_{k+1}

$$C_{k+1} = A_{k+1} B_{k+1}^{-1}$$

bis auf eine unitäre diagonale Ähnlichkeitstransformation, wo C_{k+1} aus C_k mit einem QR-Schritt hervorgeht. In der Praxis verwendet man im reellen Fall in der Regel Doppelschritte mit zwei Shifts aus der rechten unteren 2×2 Matrix von $A_k - \mu_k B_k$. Da der Algorithmus nur orthonormale Transformationen benutzt, ist er sehr rundungsfehlerstabil. Weil er in Verbindung mit der Doppelshifttechnik benutzt werden kann, ist er auch vergleichsweise effizient. Numerische Erfahrungen zeigen, daß man etwa $30n^3$ Operationen braucht, um alle Eigenwerte sowie die angenäherten Matrizen Q und Z aus (1.21) zu bestimmen.

Für ein hermitisches allgemeines Eigenwertproblem ist der QZ-Algorithmus nicht zu empfehlen, da er diese wichtige Eigenschaft des Ausgangsproblems zerstört. Ist die Dimension des Problems n klein, B positiv definit und nicht zu schlecht konditioniert, kann man die Transformation auf ein spezielles hermitisches Problem mittels der Cholesky-Zerlegung von B anwenden. Bei Problemen hoher Dimension mit schwach besetzten Matrizen A und B bietet sich die simultane Vektoriteration oder eine angepaßte Variante des Lanczos-Verfahrens an.

Für allgemeinere nichtlineare Eigenwertprobleme, etwa (1.19), benutzt man gerne folgende Verallgemeinerung der Wielandtiteration mit variablem Shift:

Gegeben sei ein nichtlineares Eigenwertproblem

$$(A - M(\lambda))x = 0, \quad x \neq 0, \quad (1.22)$$

mit $A = A^H$, $M(\lambda) = M^H(\lambda)$, $\frac{d}{d\lambda} M(\lambda) = M'(\lambda)$ positiv definit in einer Umgebung eines Eigenwertes λ_1 von (1.22). (λ_1 heißt Eigenwert der Aufgabe (1.22), falls $\det(A - M(\lambda_1)) = 0$ gilt).

In (1.18) etwa ist

$$M(\lambda) = \lambda B, \quad M'(\lambda) = B,$$

in (1.19)

$$M(\lambda) = -\lambda C + \lambda^2 K, \quad M'(\lambda) = -C + 2\lambda K,$$

d.h. die Voraussetzungen an $M(\lambda)$ bedeuten hier

$$\begin{aligned} B &= B^H && \text{positiv definit, } \lambda \text{ beliebig,} \\ C &= C^H && - C \text{ positiv semidefinit,} \\ K &= K^H && \text{positiv definit, } \lambda > 0. \end{aligned}$$

Mit $\lambda^{(0)} \neq \lambda_1$ und $u^{(0)}$ mit $(u^{(0)})^H M'(\lambda_1) u_1 \neq 0$, worin u_1 ein zu λ_1 gehörender Eigenvektor von (1.22) ist, iteriere man dann gemäß

$$\begin{aligned} (A - M(\lambda^{(\nu)})) w^{(\nu)} &= M'(\lambda^{(\nu)}) u^{(\nu)} \\ &\quad \text{(Gleichungssystem für } w^{(\nu)}) \\ \lambda^{(\nu+1)} &= \lambda^{(\nu)} + \frac{(u^{(\nu)})^H M'(\lambda^{(\nu)}) w^{(\nu)}}{(u^{(\nu)})^H M'(\lambda^{(\nu)}) u^{(\nu)}} \\ u^{(\nu+1)} &= w^{(\nu)} / \|w^{(\nu)}\| \end{aligned}$$

für $\nu = 0, 1, 2, \dots$

Angewandt auf ein spezielles hermitesches Eigenwertproblem ($M'(\lambda) = I$) ist dies gerade das Wielandtverfahren mit dem Rayleighquotienten als Shift. Man kann zeigen, daß dieses Verfahren lokal von zweiter Ordnung konvergiert, wenn λ_1 ein einfacher isolierter Eigenwert von (1.22) ist, $M(\lambda)$ in einer Umgebung von λ_1 dreimal stetig differenzierbar ist und $\lambda^{(0)}$, $u^{(0)}$ hinreichend gute Näherungen für λ_1 und u_1 sind (siehe z.B. bei Höhn, Habilitationsschrift).

Das Kernproblem bei diesem Algorithmus ist die Auflösung des Systems für $w^{(\nu)}$ mit der in der Regel indefiniten und sehr großen schwach besetzten Matrix $A - M(\lambda^{(\nu)})$. Wegen der Indefinitheit versagen hier die Standardmethoden zur iterativen Lösung dieses Systems.

1.10 Die Singulärwertzerlegung (svd)

Insbesondere im Zusammenhang mit der numerischen Rangbestimmung und der Lösung schlecht konditionierter Ausgleichsprobleme ist folgende Matrixfaktorisierung von grossem Nutzen:

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^H \quad (1.23)$$

mit A als komplexer $m \times n$ -Matrix $m \geq n$, U als unitärer $m \times m$ -Matrix, V als unitärer $n \times n$ -Matrix, Σ als Diagonalmatrix mit nichtnegativen Diagonalelementen. 0 ist eine $(m - n) \times n$ -Nullmatrix. Anschaulich läßt sie die folgende Deutung zu:

Die durch die Matrix A beschriebene lineare Transformation des \mathbb{C}^n in den \mathbb{C}^m entsteht durch die Hintereinanderschaltung einer Drehspiegelung in \mathbb{C}^n (V^H), einer Achsenmaßstabsänderung (Σ), der kanonischen Einbettung in den größeren Raum \mathbb{C}^m (Anhängen von $m - n$ Nullen) und einer weiteren Drehspiegelung (U) im \mathbb{C}^m .

Beispiel 1.10.1 Für

$$A = \frac{1}{\sqrt{15}} \begin{bmatrix} 1 & 3 \\ 5 & 0 \\ 1 & 3 \end{bmatrix}$$

errechnet man

$$A A^H = \frac{1}{3} \begin{bmatrix} 2 & 1 & 2 \\ 1 & 5 & 1 \\ 2 & 1 & 2 \end{bmatrix}$$

mit den Eigenwerten

$$\lambda_1 = \sigma_1^2 = 2, \quad \lambda_2 = \sigma_2^2 = 1, \quad \lambda_3 = \sigma_3^2 = 0.$$

Die zugehörigen orthonormierten Eigenvektoren sind der Reihe nach

$$u_1 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \quad u_2 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \quad u_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}.$$

Es ergibt sich weiter

$$\begin{aligned} A^H U &= \frac{1}{\sqrt{10}} \begin{bmatrix} 4 & -\sqrt{2} & 0 \\ 2 & 2\sqrt{2} & 0 \end{bmatrix} \\ &= (V\Sigma, \begin{pmatrix} 0 \\ 0 \end{pmatrix}) \end{aligned}$$

mit

$$\Sigma = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 1 \end{bmatrix}, \quad V = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}.$$

□

Bei der praktischen Durchführung der Zerlegung (1.23) ist es jedoch nicht sinnvoll, den Weg über das Eigenwertproblem von $A^H A$ oder $A A^H$ zu gehen, weil bei der Aufstellung dieser Matrizen durch eingeschleppte Rundungsfehler Information über die kleinsten Singulärwerte verloren geht. Man berechnet vielmehr die Zerlegung (1.23) auf iterativem Weg. Hierzu gibt es verschiedene Zugänge. Hier beschreiben wir die Lösung nach Golub, Kahan und Reinsch (realisiert in LAPACK und MATLAB). Es gibt zwei Varianten, hier die zum QR-Verfahren passende: Zuerst wird durch geeignete Householder-Transformationen A auf obere Bidiagonalgestalt gebracht:

$$J = U A W$$

mit

$$J = \underbrace{\left[\begin{array}{cccc|c} * & * & 0 & 0 & \ddots \\ 0 & * & * & 0 & \ddots \\ & \ddots & \ddots & & \\ 0 & & * & * & \\ 0 & & 0 & * & \\ \hline & & & 0 & \end{array} \right]}_n \left. \begin{array}{l} \vphantom{\left[\right]} \\ \vphantom{\left[\right]} \\ \vphantom{\left[\right]} \\ \vphantom{\left[\right]} \\ \vphantom{\left[\right]} \\ \vphantom{\left[\right]} \end{array} \right\} \begin{array}{l} n \\ m-n \end{array}$$

U und W entstehen dabei durch n bzw. $n-2$ Householder-Transformationen, und zwar wird zuerst die erste Spalte von A durch eine Transformation von links in ein Vielfaches des ersten Koordinateneinheitsvektors überführt, danach die Elemente $(1,3)$ bis $(1,n)$ der ersten Zeile in null durch eine Householder-Transformation von rechts, die die erste Spalte unberührt lässt. Daraufhin werden die Elemente der zweiten Spalte unterhalb des Diagonalelements in null überführt, und so fort gemäß dem Beispiel mit $m=5, n=3$:

$$A = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \longrightarrow P_1 A = \begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \end{bmatrix} \longrightarrow P_1 A \tilde{P}_1 = \begin{bmatrix} * & * & 0 \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \end{bmatrix} \longrightarrow$$

$$P_2 P_1 A \tilde{P}_1 = \begin{bmatrix} * & * & 0 \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & * \\ 0 & 0 & * \end{bmatrix} \longrightarrow P_3 P_2 P_1 A \tilde{P}_1 = \begin{bmatrix} * & * & 0 \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

U und W haben also die Form

$$\begin{aligned} U &= P_n \cdots P_1, \\ W &= \tilde{P}_1 \cdots \tilde{P}_{n-2}. \end{aligned}$$

Die Matrix $J^H J$ ist eine Tridiagonalmatrix und der QR-Algorithmus mit Wilkinson-Shift ist global konvergent. Er erhält die Tridiagonalstruktur. Man will aber die Tridiagonalmatrix nicht explizit bilden und transformieren, sondern immer weiter nur an J arbeiten. Dies ist tatsächlich möglich. Grundlage dafür ist folgende Beobachtung von Francis (Francis, J.: The QR transformation. A unitary analogue to the LR transformation. Comput. J. 4, 265–271 (1961,1962)): Sei T eine Tridiagonalmatrix und

$$T - \mu I = \Omega_1 \cdots \Omega_{n-1} R$$

R ist obere Dreiecksmatrix und Ω_i sind die benutzten Givensrotationen zur Annullierung der Subdiagonale von T . (Man beachte, daß Ω_1 der erste Faktor ist, der von links auf $T - \mu I$ angewendet wird.) Die nächste Matrix in der Folge ist dann bekanntlich

$$R \Omega_1 \cdots \Omega_{n-1} + \mu I \stackrel{def}{=} T_+$$

und Ω_1 ist so bestimmt, daß es von links auf T angewandt das Element (2,1) annulliert und von rechts angewandt das Element (1,2). Die erste Spalte des Produktes

$$\Omega_1 \cdots \Omega_{n-1}$$

stimmt mit der ersten Spalte von Ω_1 überein, da die übrigen Matrizen nur Spalten 2 bis n tangieren. Ist nun W eine beliebige unitäre Matrix, deren erste Spalte mit der von Ω_1 übereinstimmt, hat T kein Subdiagonalelement gleich null (wie wir immer voraussetzen) und ist Folgendes erfüllt:

$$\tilde{T} \stackrel{def}{=} W^H T W \text{ ist tridiagonal}$$

dann ist

$$\tilde{T} = D^H T_+ D$$

mit einer Diagonalmatrix aus Elementen vom Betrag eins, also identisch bis auf eine triviale Ähnlichkeitstransformation. Wir konstruieren nun eine zweiseitige unitäre Transformation von $J \hat{Q}_k$ (mit $\hat{Q}_j = Q_0 \cdots Q_j$ aus den vorausgegangenen Schritten), die mit Ω_1 von rechts beginnt und die Bidiagonalstruktur erhält. Diese besteht aus einer Wechselfolge von Givensrotationen von rechts und links. Die links auftretenden Operationen heben sich bei der Multiplikation mit der konjugiert komplexen wieder heraus. Es entsteht dann eine neue Tridiagonalmatrix, die genau der Bildung des obigen \tilde{T} entspricht. W ist dabei das Produkt der von rechts arbeitenden Givensrotationen und weil

wir mit Ω_1 beginnen, erfüllt es die Bedingungen von Francis. Somit erhalten wir durch das direkte Arbeiten an $J\hat{Q}_k$ ein Äquivalent zu einem QR-Schritt an der zugehörigen Tridiagonalmatrix und der bekannte Konvergenzsatz für das QR-Verfahren mit Wilkinsonshift ist anwendbar. Dies bedeutet hier, daß das Element $(n, n-1)$ der zugehörigen Tridiagonalmatrix entsprechend dem Element $(n-1, n)$ von J schnell gegen null konvergiert und ein erster Singulärwert gefunden wird. Dann erfolgt die Erniedrigung der Dimension usw. Schematisch sieht diese Transformationsfolge an $J\hat{Q}_k$ so aus (für den Fall $n=5$) (Hier steht wieder J für $J\hat{Q}_k$)

$$J \Omega_1 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ + & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_1 J \Omega_1 = \begin{bmatrix} * & * & + & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_1 J \Omega_1 \Omega_2 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & + & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & + & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 \Omega_3 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & + & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_3 \tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 \Omega_3 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & + \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_3 \tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 \Omega_3 \Omega_4 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & + & * \end{bmatrix} \longrightarrow$$

$$\tilde{\Omega}_4 \tilde{\Omega}_3 \tilde{\Omega}_2 \tilde{\Omega}_1 J \Omega_1 \Omega_2 \Omega_3 \Omega_4 = \begin{bmatrix} * & * & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix}.$$

In diesem Schema bedeuten „*“ Elemente der Bidiagonalstruktur und „+“ Elemente ungleich 0, die diese Struktur stören und durch die zusätzlichen Givens-Transformationen wieder in null überführt werden. Wegen $\tilde{\Omega}_i^H \tilde{\Omega}_i = I$ ist die so implizit erhaltene Transformation von $J^H J$ (bzw. $\hat{Q}_k^H J^H J \hat{Q}_k$) äquivalent mit der Transformation, die ein Schritt des QR-Algorithmus an dieser Tridiagonalmatrix bewirken würde. Der Konvergenzsatz 1.7.5 überträgt sich entsprechend, d.h. im Grenzwert nähert dann J die Matrix $\begin{pmatrix} \Sigma \\ 0 \end{pmatrix}$ aus (1.23) an. Die Akkumulation aller Givens- bzw. Householder-Transformationen von links bzw. rechts entspricht dann der Matrix U bzw. V^H .

Als Konvergenzbedingung erhalten wir lediglich, daß die Bidiagonalmatrix nicht zerfällt, d.h. kein Superdiagonalelement ist null. Ist dies aber der Fall, kann man das Problem wieder in mehrere kleinere Probleme zerlegen.

Eine der wichtigsten Anwendungen der Singulärwertzerlegung liegt in der Lösung singulärer bzw. sehr schlecht konditionierter Ausgleichsaufgaben.

Ist die Zerlegung (1.23) gegeben und Σ invertierbar, dann lautet die Lösung der Aufgabe

$$\begin{aligned} \|A \tilde{x} - b\|_2 &= \min_x \|A x - b\|_2, \\ \tilde{x} &= V(\Sigma^{-1}, 0) U^H b. \end{aligned} \quad (1.24)$$

Ist jedoch Σ nicht invertierbar, d.h. hat A den Rang $r < n$, dann ist die Lösung von (1.24) nicht eindeutig bestimmt. Erst durch geeignete Zusatzbedingungen an x wird die Lösung der Minimaufgabe wieder eindeutig. Üblich ist in diesem Zusammenhang die Forderung minimaler Länge auch für x , d.h.

$$\|\tilde{x}\|_2 = \min\{ \|y\|_2 : \|A y - b\|_2 \leq \|A x - b\|_2 \text{ für alle } x\}.$$

Die Lösung lautet dann

$$\tilde{x} = V(\Sigma^+, 0) U^H b$$

mit

$$\Sigma^+ = \text{diag}(\sigma_i^+) \quad \text{und} \quad \sigma_i^+ = \begin{cases} 1/\sigma_i & \text{falls } \sigma_i \neq 0 \\ 0 & \text{sonst.} \end{cases}$$

Bemerkung 1.10.1 *Die Matrix*

$$A^I = V(\Sigma^+, 0) U^H$$

heißt die Moore-Penrose-Pseudoinverse von A und stellt eine Verallgemeinerung des Begriffs „inverse Matrix“ auf nichtinvertierbare und nichtquadratische Matrizen dar. Man kann zeigen, daß A^I folgende vier Bedingungen erfüllt, durch die sie auch eindeutig bestimmt ist:

$$\begin{aligned} (A^I A) &= (A^I A)^H, \\ (A A^I) &= (A A^I)^H, \\ A^I A A^I &= A^I, \\ A A^I A &= A. \end{aligned}$$

□

Für eine Matrix A von vollem Rang ist der kleinste Singulärwert der Abstand zur nächstgelegenen Matrix mit Rangabfall, gemessen in der euklidischen Matrixnorm. Kennt man also die Ungenauigkeit in A bzw. die Größe des Rundungsfehlereinflusses in den berechneten Singulärwerten, dann kann man wenigstens entscheiden, ob die Matrix „sicher von vollem Rang ist“. Dies versteht man unter „numerischer Rangbestimmung“.

Bemerkung 1.10.2 *Es gibt auch eine zum Jacobi-Verfahren analoge Vorgehensweise zur Bestimmung der SVD, bei der keine Vorbehandlung der Matrix erforderlich ist. Durch Rotationen von links und von rechts werden alle Ausserdiagonalelemente im Grenzwert auf Null gebracht. Die Singulärvektoren erhält man einfach aus den Produkten aller jeweiligen Rotationen.*

1.11 Zusammenfassung

Die Problemstellung eines Matrixeigenwertproblems tritt in der Praxis in zwei Varianten auf: als vollständiges Eigenwert/Eigenvektorproblem, wo es gilt, alle Eigenwerte und Eigenvektoren zu finden, und als partielles Problem, wo nur einige Eigenwerte mit Eigenvektoren gesucht sind, in den technischen Anwendungen in der Regel die kleinsten und in den Anwendungen in der Stochastik die grössten Eigenwerte. Das vollständige Eigenwertproblem tritt in der Regel nur bei kleineren Dimensionen auf (n maximal im Bereich von einigen hundert). Hier bietet sich mit dem QR-Verfahren ein universell einsetzbares Instrument an, daß bei geeigneter Implementierung quasi als „black box“ nutzbar ist. Die Methoden zur Bestimmung einzelner Eigenwerte sehr grosser Matrizen, also das Lanczos-Verfahren und seine Verallgemeinerungen sowie die (hier nicht besprochene) simultane Vektoriteration erfordern dagegen eine sinnvolle Wahl der Startvektoren, was nur mit Detailkenntnissen der spezifischen Problemstellung gelingt.

Kapitel 2

Zugang zu numerischer Software und anderer Information

Don't reinvent the wheel! Für die Standardaufgaben der Numerischen Mathematik gibt es inzwischen public domain Programme sehr guter Qualität, sodass es oft nur notwendig ist, mehrere solcher Module zusammenzufügen, um ein spezifisches Problem zu lösen. Hier wird eine Liste der wichtigsten Informationsquellen angegeben.

2.1 Softwarebibliotheken

In der Regel findet man im Netz bereits vorgefertigte Softwarelösungen, die meisten davon für akademischen Gebrauch kostenfrei: Die bei weitem grösste und wichtigste Quelle ist die

NETLIB

Dies ist eine Sammlung von Programmbibliotheken in f77, f90 , c, c++ für alle numerischen Anwendungen:

<http://www.netlib.org/>

Man kann nach Stichworten suchen ("search") und bekommt auch Informationen aus dem NaNet (Numerical Analysis Net)

Die Bibliotheken findet man unter "browse repository".

Die wichtigsten Bibliotheken im Zusammenhang mit dem Matrizeigenwertproblem sind:

1. lapack, clapack, lapack90
die gesamte numerische Lineare Algebra (voll besetzte und Band-Matrizen) inklusive Eigenwertprobleme und lineare Ausgleichsrechnung in sehr guter Qualität. Die lineare Algebra in MATLAB ab Version 6 beruht auf der lapack-Bibliothek.

2. `linpack`, `eispack` : die Vorläufer von `lapack`. Einige der Verfahren aus diesen Bibliotheken wurden jedoch nicht in `lapack` übernommen.
3. `lanz`, `lanzoz` : Eigenwerte/Eigenvektoren grosser dünn besetzter symmetrischer Matrizen
4. `toms`: Transactions on mathematical software. Sammlung von Algorithmen für verschiedene Aufgaben, sehr gute Qualität. Enthält mehrere spezielle Eigensystems codes, u.a. Bo Kagstroms Code für die Jordan-Normalform.
5. `linalg`: Iterative Verfahren für lineare Systeme, sonstige lineare Algebra
6. `svdpack` Approximative svd-Löser für grosse Systeme
7. `c/meschach` eigenständige Bibliothek mit vielen wichtigen LA-Routinen in C.

2.2 Suchen nach software

Man beginnt sinnvollerweise zuerst mit dem Dienst

<http://math.nist.gov/HotGAMS/>

Dort öffnet sich ein Suchmenü, wo man nach Problemklassen geordnet durch einen Entscheidungsbaum geführt wird bis zu einer Liste verfügbarer Software (auch in den kommerziellen Bibliotheken IMSL und NAG). Falls der code als public domain vorliegt, wird er bei "Anklicken" sofort geliefert.

Software für C++ findet man unter

<http://oonumerics.org/oon/>

2.3 Hilfe bei Fragen

Hat man Fragen, z.B. nach Software, Literatur oder auch zu spezifischen mathematischen Fragestellungen, kann man in einer der Newsgroups eine Anfrage plazieren. Häufig bekommt man sehr schnell qualifizierte Hinweise. Zugang zu Newsgroups z.B. über

`xrn`

. mit "subscribe" . Die wichtigste News-Group ist hier

`sci.math.num-analysis`

Im `xrn`-Menu kann man mit "post" ein Anfrage abschicken und dabei die Zielgruppe frei wählen.

Index

- Ähnlichkeitstransformation, 18
- Courant'sches Minimax-Prinzip, 14
- allgemeines Eigenwertproblem, 37, 68
- Bauer-Fike, 16
- Eigenwerte, tridiagonal, 22
- Einzelsensitivität von Eigenwerten, 17
- Givens-Rotation, 59
- Golub und Kahan, 76
- Hessenberg-Form, 18
- Iteration v. Mises, 30
- Jacobi-Verfahren, 42
- Kreisesatz von Gerschgorin, 8
- Lanczos-Verfahrens, 61
- Lokalisierung von Eigenwerten, 7
- Moore-Penrose-Pseudoinverse, 78
- Norm, absolute, 16
- QR-Verfahren, 48
- QZ-Algorithmus , 69
- Rayleighquotienten, 12
- RITZIT , 40
- Schur, Transforamtion, 48
- Sensitivität des Eigenwertproblems, 7
- simultane Vektoriteration, 40
- Sylvester, 23
- Trägheitssatz, 23
- Tridiagonalform, 18
- Tridiagonalmatrix, 22
- Verallgemeinerung der Wielandtiteration, 72
- Verfahren von Wielandt, 30
- Wielandt-Verfahren , 37
- Wielandtverfahren, 33
- Wilkinson'schen Shift-Technik, 60