Beiträge zum Lanczosalgorithmus in endlicher Arithmetik

Dissertation
zur Erlangung des Doktorgrades
der Fakultät für Mathematik der Universität Bielefeld

vorgelegt von
Wolfgang Wülling
März 2004

1. Gutachter: Professor Dr. L. Elsner
2. Gutachter: Professor Dr. R. Nabben (TU Berlin)
Datum der mündlichen Prüfung: 19. July 2004

Gedruckt auf alterungsbeständigem Papier nach ∞ ISO 9706

Danksagung

Für die Möglichkeit, diese Arbeit anzufertigen, und für die geduldige Betreuung möchte ich mich bei Herrn Prof. Dr. L. Elsner bedanken. Seine Kommentare und Anregungen haben auch zur Verbesserung der Darstellung beigetragen.

Bei Herrn Prof. Dr. Z. Strakoš bedanke ich mich für die Einladung nach Prag und die dort geführten, hilfreichen Diskussionen.

Zudem gilt Herrn Dr. J. Liesen Dank für das dieser Arbeit entgegengebrachte Interesse und den Hinweis auf [38].

Inhaltsverzeichnis

Kapitel 1

Einleitung

Neben der Lösung linearer Gleichungssysteme ist die zweite zentrale Aufgabenstellung der numerischen linearen Algebra die Lösung des linearen Eigenwertproblems

$$Ax = \lambda x, \tag{1.1}$$

dabei ist $A \in \mathbb{C}^{N\times N}$ eine vorgegebene quadratische Matrix, zu der eine Zahl $\lambda \in \mathbb{C}$ und ein nicht trivialer Vektor $x \in \mathbb{C}^N$ gesucht wird. Da sich die Eigenwerte λ von (1.1) im Allgemeinen nicht direkt, d.h. nicht in endlich vielen Schritten, berechnen lassen, denn sie sind Nullstellen des charakteristischen Polynoms $\psi(\lambda) = \det(\lambda I - A)$ von A, müssen Eigenwertmethoden einen iterativen Charakter haben.

Wir unterscheiden zwischen *großen* und *kleinen* Matrizen. Dabei wird eine Matrix als *klein* betrachtet, falls es möglich ist, den ganzen zweidimensionalen Array in den Arbeitsspeicher eines Computer zu schreiben, ohne dass besondere Speichertechniken verwendet werden müssen. Andernfalls heißt eine Matrix *groß*. Zur Berechnung der Eigenpaare von kleinen Matrizen sind beispielsweise das QR-Verfahren oder das Diagonalisierungsverfahren von Jacobi geeignet. Für *große*, dünnbesetzte Matrizen können aufgrund der Restriktion bezüglich des Arbeitsspeichers diese Verfahren in der Regel nicht verwendet werden. Stattdessen werden Reduktionsalgorithmen bzw. Unterraummethoden benutzt. Das soll heißen, dass die Matrix A auf eine *kleinere* Matrix reduziert wird, für die das Eigenwertproblem dann mit den zuvor erwähnten Methoden bearbeitet werden kann. Bei der Reduktion von A auf eine kleinere Matrix wird (in der Regel vermöge der Potenzmethode) eine Basis eines Unterraumes erzeugt, aus dem dann die Eigenvektorapproximationen stammen. Und ein Unterraumverfahren ist iterativ in dem Sinne, dass man die Basis des Unterraums sukzessive vergrößert.

Für symmetrische Matrizen hat der Lanczosalgorithmus, der also zu den Unterraumverfahren zu zählen ist, sehr schöne theoretische Eigenschaften: er erzeugt sofort eine orthonormale Basis eines Krylowunterraumes, der Algorithmus hat eine kurze, aus drei Termen bestehende Rekursion, es wird

eine tridiagonale Matrix erzeugt und - sofern man nur an Eigenwertapproximationen, nicht aber an Eigenvektoren, interessiert ist - es müssen in jedem Schritt nur zwei schon berechnete Lanczosvektoren gespeichert werden. Die Approximationen an die Eigenpaare von A heißen Ritzwerte und Ritzvektoren und darunter verstehen wir folgendes: Ist $\mathcal{K}$ ein Untervektorraum von $\mathbb{R}^N$ (bzw. $\mathbb{C}^N$), dann heißt $\theta \in \mathbb{C}$ ein Ritzwert von A und $z \in \mathcal{K}$, $z \neq 0$, ein Ritzvektor, falls

$$Az - \theta z \perp \mathcal{K} \tag{1.2}$$

gilt. Es wird dann (θ, z) auch Ritzpaar genannt.

In dieser Arbeit beschäftigen wir uns mit Problemen, die entstehen, wenn der einfache Lanczosalgorithmus auf einem Computer implementiert, also in endlicher Arithmetik ausgeführt wird. Insbesondere geht die Orthogonalität unter den Lanczosvektoren aufgrund von Rundungsfehlern sehr schnell verloren.

In Kapitel 2 beschreiben wir kurz den einfachen Lanczosalgorithmus sowie seine Eigenschaften in exakter und endlicher Arithmetik. C. Paige leitete in der Arbeit [23] die mittlerweile bekannte Formel (2.13) her, die erstmals einen Zusammenhang zwischen dem Orthogonalitätsverlust und der Konvergenz von Ritzpaaren herstellte; wir zitieren in Abschnitt 2.3 die für unsere Arbeit relevanten Ergebnisse der Fehleranalyse von Paige. In endlicher Arithmetik kann - und wird es meistens - vorkommen, dass der Lanczosalgorithmus wegen des erwähnten Orthogonalitätsverlustes nicht nach $d \leq N$ Schritten stoppt. Er erzeugt eine Folge von Tridiagonalmatrizen, deren Eigenwerte die Ritzwerte sind. Und obwohl die Orthogonalitätsbedingung aus der Definition (1.2) der Ritzpaare bei Rechnungen in endlicher Arithmetik nicht mehr erfüllt sein muß, werden wir trotzdem immer den Begriff Ritzwert bzw. Ritzwert für die entsprechenden Größen benutzen. Numerisch ist es so, dass sich Ritzwerte um Eigenwerte von A *clustern*. Wir analysieren in Abschnitt 2.4 das Konvergenzverhalten von Ritzpaaren in einem Cluster anhand einer Kennzahl für das Residuum. Wir können die in [38] geäußerte Vermutung beweisen, dass alle Ritzpaare, die zu einem Cluster gehören, der einen Eigenwert approximiert, gute Näherungen an die Eigenpaare von A sind. Die einzige Ausnahme tritt auf, falls die Clustergröße, das ist die Anzahl der zu einem Cluster gehörenden Ritzwerte, alternierend ist. Hier gilt es dann die geraden und ungeraden Iterationsschritte des Lanczosalgorithmus zu unterscheiden. Für diesen Ausnahmefall wird ein Gegenbeispiel konstruiert.

Aus den Ergebnissen folgt dann auch, dass jeder Cluster von Ritzwerten stabilisiert ist, d.h. er kann sich nur in der Nähe eines Eigenwertes bilden: je mehr Ritzwerte in dem Cluster sind, desto präziser muß die beste Approximation an den Eigenwert sein.

Aus den Eigenpaaren von A und dem Startvektor des Lanczosalgorithmus definiert sich eine Gewichtsfunktion $m(\lambda)$ bzw. ein Riemann-Stieltjes Integral

$$\int f(\lambda)dm(\lambda) := \sum_{j=1}^{N} m_j f(\lambda_j), \tag{1.3}$$

wobei λ_j, $1 \leq j \leq N$, die Eigenwerte von A sind, und die Gewichte m_j ergeben sich aus dem Startvektor des Lanczosalgorithmus bzw. des *cg*-Verfahrens und den zugehörigen Eigenvektoren u_j. Der Lanczosalgorithmus bzw. das *cg*-Verfahren berechnet implizit eine Folge von Gaußquadraturapproximationen an das Integral (1.3), d.h. Gewichtsfunktionen $m^{(k)}(\lambda)$. Es stellt sich daher folgende Frage: Wenn es im k-ten Schritt des Lanczosalgorithmus einen einfachen Ritzwert $\theta_j^{(k)}$ gibt, der einen einfachen Eigenwert λ_r hinreichend gut approximiert, nehmen dann die beiden Gewichtsfunktionen $m(\lambda)$ und $m^{(k)}(\lambda)$ dort ungefähr den gleichen Wert an? Ist das zum Ritzwert $\theta_j^{(k)}$ gehörende Gewicht $m_j^{(k)}$ ungefähr gleich m_r? In endlicher Arithmetik bilden sich eventuell Cluster von Ritzwerten. Wird also λ_r von einem Cluster approximiert, so beobachtet man numerisch, dass dann die Summe der Gewichte, die zu den Ritzwerten im Cluster gehören, ebenfalls ungefähr gleich m_r ist.

Wir können im dritten Kapitel diese Stabilisierung der Gewichte sowohl in exakter als auch in endlicher Arithmetik nachweisen. Das ist auch deshalb interessant, da sich die Fehler im *cg*-Verfahren durch Gaußquadraturapproximationen an (1.3) darstellen lassen (siehe [40]) und sich eventuell so die Konvergenzverzögerung des *cg*-Verfahrens beim Rechnen mit Rundungsfehlern erklären läßt.

Nachdem wir in den Kapiteln 2 und 3 die in [38] formulierten, offenen Fragen zur Konvergenzanalyse des Lanczosalgorithmus gelöst haben, beschäftigen wir uns in Kapitel 4 mit dem von Elsner und Bunse-Gerstner entwickelten unitären Lanczos-ähnlichen Algorithmus. Denn durch die Cayleytransformation besteht ein enger Zusammenhang zwischen den hermiteschen und unitären Matrizen: Das Bild einer hermiteschen (symmetrischen) Matrix unter der Cayleytransformation ist eine unitäre Matrix. Implementiert man diesen Algorithmus, indem die auftretenden Unterraumbasen mit dem Gram-Schmidt Verfahren orthogonalisiert werden, so zeigt der unitäre Lanczos-ähnliche Algorithmus in endlicher Arithmetik ein ähnliches Verhalten wie der symmetrische Lanczosalgorithmus, d.h. sehr rascher Orthogonalitätsverlust der Vektoren der berechneten Unterraumbasis und damit einhergehend das Clustern von Ritzwerten um Eigenwerte. Zu den Analogien zählt insbesondere auch - wie wir zeigen werden - die Berechenbarkeit der Residuen nur aus Größen des kleinen Eigenwertproblems. Wir führen zudem eine detaillierte Rundungsfehleranalyse durch und weisen analog zu

(2.13) nach, dass der Orthogonalitätsverlust auch hier die Konvergenz von Ritzpaaren nach sich zieht.

Notationen und arithmetisches Modell

Mit $\mathbb{R}$ wird die Menge der reellen Zahlen und mit $\mathbb{C}$ die der komplexen Zahlen bezeichnet. Für reelle Zahlen x wird mit $[x]$ die größte ganze Zahl, die kleiner oder gleich x ist, bezeichnet. Zu einer komplexen Zahl $w \in \mathbb{C}$ stellt $\bar{w} \in \mathbb{C}$ die konjugiert komplexe Zahl dar, $|w|$ den Betrag von w und mit $\Re(w)$ sowie $\Im(w)$ werden Real- und Imaginärteil von w bezeichnet. Mit I_k wird die $k \times k$ Einheitsmatrix bezeichnet, wobei der Index k weggelassen wird, falls die Dimension aus dem Kontext heraus klar sein sollte. Mit dem $*$ werden transponierte Matrizen bzw. Vektoren mit konjugiert komplexen Einträgen gekennzeichnet, d.h. zu $B \in \mathbb{C}^{N\times M}$ ist $B^* \in \mathbb{C}^{M\times N}$ und es gilt $(B^*)_{i,j} = (\bar{B})_{j,i}$. Als Symbol für die Menge der Eigenwerte einer quadratischen Matrix A wird $\Lambda(A)$ verwendet. Ferner stellt $||\cdot||$ für Vektoren die euklidische Norm und für Matrizen die Spektralnorm dar. Ist A eine symmetrische und positiv definite Matrix, so wird mit $||\cdot||_A$ die Energie- bzw. A-Norm für Vektoren x bezeichnet, es gilt $||x||_A = \sqrt{x^*Ax}$. Die $j$te Spalte der Einheitsmatrix wird mit e_j bezeichnet und wir benutzen den Vektor

$$e = \sum_{j=1}^{N} e_j = (1, 1, \ldots, 1)^* \in \mathbb{R}^N.$$

Als Symbol für den Grad eines Polynoms p benutzen wird $\deg p$. Die übrigen Bezeichnungen werden im Zusammenhang gegeben sobald sie benötigt werden.

Wir legen das Standardmodell der Gleitkommaarithmetik (vgl. [18] S. 39 ff) zu Grunde: Mit ϵ sei die Maschinengenauigkeit bezeichnet und mit $fl(\cdot)$ der Gleitkommaoperator. Für die Grundrechenarten in Gleitkommaarithmetik[1] gilt

$$fl(x \text{ op } y) = (x \text{ op } y)(1 + \delta_1) = \frac{(x \text{ op } y)}{1 + \delta_2} \tag{1.4}$$

mit $|\delta_1|, |\delta_2| \leq \epsilon$ und op $\in \{+, -, /, *, \sqrt{}\}$.

[1] Für weitere Details siehe Anhang A.

Kapitel 2

Symmetrischer Lanczosalgorithmus

In diesem Kapitel untersuchen wir das Konvergenzverhalten von Ritzpaaren beim Lanczosalgorithmus, wenn sich Ritzwerte clustern. Dabei werden so genannte Eigenvektor-Eigenwert Identitäten für symmetrische Tridiagonalmatrizen ausgenutzt. Derartige Relationen sind nicht nur für symmetrische Tridiagonalmatrizen bekannt. Beispielsweise werden in [9] zum einen Eigenvektor-Eigenwert Beziehungen für das allgemeine Eigenwertproblem hergeleitet und zum anderen werden auch stetige Analoga dieser Relationen im Kontext von Sturm-Liouvill'schen Differentialgleichungen aufgezeigt. Solche Formeln finden sich für (obere) Hessenbergmatrizen auch in [11] und [46], wo das Rundungsfehlerverhalten von Krylowunterraumverfahren mit Hilfe der Eigenvektor-Eigenwert Relationen betrachtet wird.

Bei unserer hier vorgenommenen Analyse untersuchen wir eine Kennzahl $\delta_{k,j}$ (die genaue Definition findet sich in (2.7)). In verschiedenen Arbeiten, in denen man sich mit Clustern von Eigenwerten bei (symmetrischen) Tridiagonalmatrizen beschäftigt hat, hat dieser Ausdruck große Aufmerksamkeit erfahren. In [45] bestimmt Ye mit Hilfe dieser Größe (und einer dazu analogen) Untermatrizen einer Tridiagonalmatrix, die einen Eigenwertcluster hat, so, dass diese Untermatrizen nach Möglichkeit einen Eigenwert besitzen, der in bzw. sehr dicht bei dem Cluster liegt: dazu müssen die erwähnten Kennzahlen *klein* sein.

Ähnlich vorgehend entwickelt Parlett in [29] eine Methode zur Berechnung der zu einem Cluster von zwei (oder mehreren) Eigenwerten einer symmetrischen Tridiagonalmatrix gehörenden orthogonalen Eigenvektoren: die Eigenvektoren werden aus denen geeigneter Untermatrizen zusammengesetzt, wobei geeignet bedeutet, dass die entsprechenden Kennzahlen $\delta_{k,j}$ hinreichend klein sind.

Überdies bezeichnet Parlett in [27] beim Lanczosalgorithmus einen Index k, für den $\delta_{k,j}$ ein lokales Minimum besitzt, als *Point of Discovery*. Dies im

Zusammenhang mit der so genannten *Misconvergence*, worunter folgendes verstanden wird: Hat die Matrix A einen Cluster von Eigenwerten (nicht zu verwechseln mit einem Cluster von Ritzwerten), so kann es passieren, dass ein Ritzwert für mehrere Iterationsschritte in der Mitte der beiden Eigenwerte stagniert, bevor ein weiterer Ritzwert auftaucht, so dass beide Eigenwerte approximiert werden. Nimmt $\delta_{k,j}$ für den stagnierenden Ritzwert ein lokales Minimum an, so nennt Parlett diesen Index k eben *Point of Disvocery*. Aber hier ist Vorsicht geboten, denn anders als in [27] behauptet, muß das Konvergenzverhalten anschließend nicht wieder routinemäßig verlaufen, wie wir anhand von Beispiel (2.22) sehen werden (vgl. dazu auch Tabelle 1).

Schließlich benutzt Greenbaum in [16] in ihrer Rückwärtsanalyse des Lanczosalgorithmus die Größe $\delta_{k,j}$, um konvergente und nicht konvergierte Cluster von Ritzwerten zu unterscheiden. Wir werden im folgenden auch sehen, dass diese Unterscheidung nicht notwendig ist.

2.1 Tridiagonalisierung

Es sei $A \in \mathbb{R}^{N\times N}$ eine symmetrische Matrix mit einfachen Eigenwerten λ_j und Eigenvektoren u_j: $Au_j = \lambda_j u_j$, $1 \le j \le N$, mit $u_j^* u_j = 1$ und die Eigenwerte seien in aufsteigender Reihenfolge sortiert:

$$A = U\Lambda U^*$$

$$\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_N), \quad \lambda_1 < \lambda_2 < \ldots < \lambda_N.$$

Angenommen, wir haben eine Jacobimatrix $T \in \mathbb{R}^{N\times N}$, d.h. T ist tridiagonal, symmetrisch und besitzt nur strikt positive Subdiagonalelemente:

$$T = T_N = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_N \\ & & & \beta_N & \alpha_N \end{bmatrix}, \quad \beta_j > 0, \tag{2.1}$$

sowie eine orthonormale Matrix Q, so dass

$$A = QTQ^* \tag{2.2}$$

gilt. Zur Frage der Eindeutigkeit von (2.2) haben wir

Satz 2.1 ([26] S.113) *Es gelte (2.2) mit orthonormaler Matrix Q und einer Jacobimatrix T gemäß (2.1). Dann werden T und Q eindeutig durch A und $q_1 = Qe_1$ bestimmt.*

Der Beweis dieses Satzes liefert, indem man spaltenweise (2.2) bzw. $AQ = QT$ vergleicht, gerade den einfachen Lanczosalgorithmus:

Algorithmus 2.2 (Einfacher Lanczosalgorithmus) *Gegeben* $q_1 \in \mathbb{R}^N$ *mit* $q_1^* q_1 = 1$, $q_0 = 0 \in \mathbb{R}^N$ *und* $\beta_1 = 0 \in \mathbb{R}$, *berechnet man nun rekursiv*

$$\begin{aligned} \textit{Für} \quad k = 1, 2, \ldots & \\ \alpha_k &= q_k^*(Aq_k - \beta_k q_{k-1}) \\ \tilde{q}_{k+1} &= Aq_k - \alpha_k q_k - \beta_k q_{k-1} \\ \beta_{k+1} &= ||\tilde{q}_{k+1}|| \\ q_{k+1} &= \frac{\tilde{q}_{k+1}}{\beta_{k+1}} \end{aligned} \tag{2.3}$$

Dieser Algorithmus würde in (2.3) zusammenbrechen, wenn $\beta_{k+1} = 0$ ist; das bedeutet man stoppt das Verfahren in diesem Fall. Zudem haben wir den Lanczosalgorithmus hier mit dem Adjektiv *einfach* versehen, da bei dem Algorithmus 2.2 weder eine Reorthogonalisierung der Vektoren q_j noch eine explizite oder implizite Restarttechnik benutzt wird (siehe z.B. [36], [44] und [42] S.391 ff).

2.2 Rayleigh-Ritz Approximationen

Mit der bloßen Tridiagonalisierung einer symmetrischen Matrix hätte man aber noch nichts gewonnen. Das Ziel ist die Berechnung, d.h. die Approximation von Eigenwerten und Eigenvektoren. Der Lanczosalgorithmus stellt das Rayleigh-Ritz Verfahren angewandt auf den Krylowunterraum

$$\mathcal{K}_k(A, q) = \text{span}(q, Aq, A^2q, \ldots, A^{k-1}q)$$

dar. Krylowunterräume kommen gewissermaßen natürlich ins Spiel, da sie durch einfache Vektoriteration gebildet werden: Bei großen und dünn besetzten Matrizen hat man in der Regel nicht die Matrix selbst zur Verfügung sondern ein Unterprogramm, so dass man lediglich das Ergebnis einer Multiplikation der Matrix mit einem Vektor kennt. Dabei besitzen Krylowunterräume außerdem den Vorteil, eine aufsteigende Fahne von Unterräumen zu bilden. Der Lanczosalgorithmus bestimmt sukzessive orthonormale Basen der Krylowunterräume $\mathcal{K}_k(A, q)$, nämlich $q_1, q_2, \ldots, q_k$, die so genannte Lanczosbasis (siehe [26] S. 252 ff.). Mit

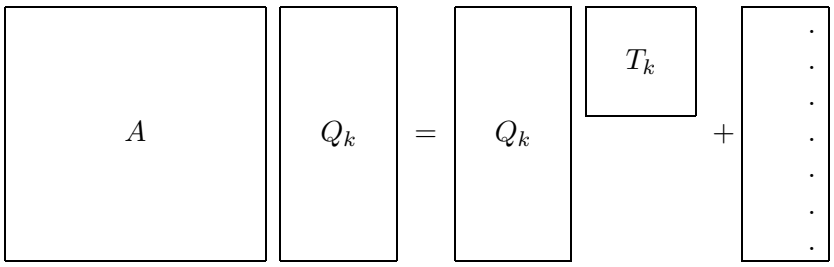
$$Q_k = [q_1, q_2, \ldots, q_k] \ \in \mathbb{R}^{N \times k}$$

werden die Rekursionen des Lanczosalgorithmus durch zwei Matrixgleichungen charakterisiert:

$$AQ_k = Q_k T_k + \beta_{k+1} q_{k+1} e_k^* \tag{2.4}$$

$$Q_k^* Q_k = I_k. \tag{2.5}$$

Ein Bild von (2.4):



Mit der Rayleigh-Ritz Prozedur werden Eigenpaare aus einem Unterraum approximiert, das ist hier also $\mathcal{K}_k(A, q_1)$. Dazu wird zunächst zu diesem Unterraum eine orthonormale Basis, beim Lanczosverfahren also Q_k, bestimmt. Bezüglich dieser orthonormalen Basis des Unterraumes berechnet man dann den Rayleighquotienten der Matrix A, das bedeutet hier $T_k = Q_k^* A Q_k$, von dem anschließend die Eigenpaare exakt bestimmt werden. Für symmetrische Matrizen hat dieses Verfahren die folgenden Eigenschaften (vgl. [26] S. 215 ff.):

- Die Ritzwerte sind Rayleighquotienten von A.
- Die Residualmatrix hat minimale Norm, d.h. es gilt $\beta_{k+1} = ||AQ_k - Q_k T_k|| \leq ||AQ_k - Q_k B||$ für alle $B \in \mathbb{R}^{k \times k}$.
- Die Ritzpaare sind Eigenpaare der Projektion von A auf den Unterraum $\mathcal{K}_k(A, q_1)$.

Wir bezeichnen mit

$$T_k = S_k \Theta_k S_k^*, \quad S_k^* S_k = S_k S_k^* = I_k, \tag{2.6}$$

die Spektralzerlegung von T_k; $S_k = [s_1^{(k)}, \ldots, s_k^{(k)}]$ und $\Theta_k = \mathrm{diag}(\theta_1^{(k)}, \ldots, \theta_k^{(k)})$. Dabei numerieren wir auch die Eigenwerte von T_k in aufsteigender Reihenfolge

$$\theta_1^{(k)} \leq \theta_2^{(k)} \leq \ldots \leq \theta_k^{(k)};$$

sind die Nebendiagonalelemente von T_k alle ungleich Null - wie in (2.1) - so sind diese Ungleichungen strikt. Die Eigenwerte von T_k sind die Rayleigh-Ritz Näherungen an die Eigenwerte von A und zu jedem Ritzwert $\theta_i^{(k)}$ gehört ein Ritzvektor $z_i^{(k)}$ aus dem Krylowraum $\mathcal{K}_k(A, q_1)$:

$$z_i^{(k)} = Q_k s_i^{(k)} = \sum_{j=1}^{k} q_j s_{j,i}^{(k)}.$$

Tritt im einfachen Lanczosalgorithmus der Fall ein, dass $\beta_{k+1} = 0$ gilt, so stellt $\mathcal{K}_k(A, q_1)$ einen unter A invarianten Unterraum dar, und das bedeutet, dass die Ritzwerte $\theta_1^{(k)}, \ldots, \theta_k^{(k)}$ Eigenwerte von A sind.

Eine weitere wichtige und hervorzuhebende Eigenschaft des Lanczosalgorithmus besteht in der Möglichkeit, das Residuum der Rayleigh-Ritz Approximationen ohne großen Aufwand zu messen, denn aus $T_k s_j^{(k)} = \theta_j^{(k)} s_j^{(k)}$ und (2.4) folgt

$$||Az_j^{(k)} - \theta_j^{(k)} z_j^{(k)}|| = \beta_{k+1}|s_{k,j}^{(k)}|.$$

Das bedeutet, dass der letzten Zeile der Eigenvektormatrix von T_k, also den Zahlen $s_{k,j}^{(k)} = e_k^* S_k e_j$ eine Schlüsselrolle zukommt, denn man kann allein aus Größen des kleinen Eigenwertproblems für T_k die Güte der Ritzapproximationen beurteilen, ohne die *langen* Ritzvektoren $z_j^{(k)}$ ausrechnen zu müssen. Wir definieren für jedes Ritzpaar im $k$ten Schritt des Lanczosalgorithmus nun

$$\delta_{k,j} = \beta_{k+1}|s_{k,j}^{(k)}|. \tag{2.7}$$

Für ein kleines $\delta_{k,j}$ liegt der Ritzwert $\theta_j^{(k)}$ also in der Nähe eines Eigenwertes von A. Außerdem gilt:

Satz 2.3 ([26] S. 222) *Sei $A \in \mathbb{R}^{N\times N}$ eine symmetrische Matrix mit einfachen Eigenwerten und $y \in \mathbb{R}^N$ ein Vektor der Länge 1, $y^*y = 1$, und mit Rayleighquotienten $\theta = y^*Ay$. Es sei α der Eigenwert von A, dessen Abstand zu θ am kleinsten ist, und der auf euklidische Länge eins normierte Eigenvektor zu α sei z. Außerdem bezeichnen wir mit $\gamma = \min_{\lambda\in\Lambda(A)\setminus\{\alpha\}} |\lambda - \theta|$ und $\zeta = \angle(y, z)$. Dann gilt:*

$$|\sin(\zeta)| \le \frac{||Ay - \theta y||}{\gamma} \quad , \quad |\alpha - \theta| \le \frac{||Ay - \theta y||^2}{\gamma}.$$

2.3 Endliche Arithmetik

Werden die Berechnungen von Algorithmus 2.2 in exakter Arithmetik ausgeführt, so bricht dieser mit einem $\beta_d = 0$, $d \le N + 1$ in (2.3) ab. Ein anderes Bild zeichnet sich ab, wenn man den Lanczosalgorithmus auf einem Computer implementiert und die Rechnungen daher in endlicher Arithmetik ausgeführt werden. Als erstes beobachtet man, dass die Orthogonalität unter den Lanczosvektoren verloren geht, und zwar in der Regel schon nach wenigen Iterationen ($k \ll N$). Daher ist die Beziehung zwischen der berechneten Matrix T_k und der gegebenen Matrix A zunächst nicht klar. Die Orthogonalität unter den Lanczosvektoren geht dabei nicht rein zufällig verloren, das soll heißen, dass hinter dem Orthogonalitätsverlust eine gewisse Struktur steckt, die von C. Paige (siehe [23], [24] und [25]) entdeckt wurde: Die Konvergenz von Ritzpaaren kann - muß aber nicht - den Orthogonalitätsverlust unter den Lanczosvektoren nach sich ziehen. Wir werden dies in Satz 2.4 präzisieren und geben nun die Analyse von C.Paige wieder. An

die Stelle der Rekursionen (2.4) und (2.5) treten nun

$$\begin{aligned} AQ_k &= Q_k T_k + \beta_{k+1} q_{k+1} e_k^* + F_k \quad (2.8) \\ Q_k^* Q_k &= C_k^* + \Delta_k + C_k, \quad (2.9) \end{aligned}$$

wobei C_k eine strikte obere Dreiecksmatrix, d.h. mit Nullen auf der Diagonale, ist und $\Delta_k = \mathrm{diag}(q_1^* q_1, \ldots, q_k^* q_k)$. Wir haben die Notation hier ein wenig geändert: $Q_k, q_{k+1}, \beta_{k+1}$ und T_k sind das Ergebnis von Rechnungen in Gleitkommaarithmetik. Um das zu verdeutlichen und die Resultate von denen exakter Rechnungen zu unterscheiden, müsste man sie beispielsweise mit einem Dach versehen und es würden sich (2.8) und (2.9) als

$$\begin{aligned} A\hat{Q}_k &= \hat{Q}_k \hat{T}_k + \hat{\beta}_{k+1} \hat{q}_{k+1} e_k^* + F_k \\ \hat{Q}_k^* \hat{Q}_k &= C_k^* + \Delta_k + C_k, \end{aligned}$$

lesen. Darauf haben wir verzichtet, weisen aber darauf hin, dass $Q_k, q_{k+1}, \beta_{k+1}$ und T_k berechnete Größen sind. Weiterhin wird im Nachfolgenden - wie auch schon bei Paige [23] - angenommen, dass das *kleine* Eigenwertproblem für T_k (bzw. $\hat{T}_k$) exakt gelöst wird, dass also (2.6) auch für die berechnete Jacobimatrix gilt. Die Maschinengenauigkeit wird mit ϵ und die maximale Anzahl von Nichtnullelementen in den Zeilen von A mit m bezeichnet. Paige definiert die folgenden Vielfachen der Maschinengenauigkeit:

$$\epsilon_0 = 2(N+4)\epsilon, \quad \epsilon_1 = 2\left(7 + m\frac{\| \mid A \mid \|}{\|A\|}\right)\epsilon$$

$$\epsilon_2 = \sqrt{2}\max\{6\epsilon_0, \epsilon_1\}, \quad \eta_k = k^{2.5}\|A\|\epsilon_2.$$

Unter der Annahme, dass

$$\epsilon_0 < \frac{1}{12}, \qquad k(3\epsilon_0 + \epsilon_1) < 1 \quad (2.10)$$

ist und Terme der Ordnung von $\mathcal{O}(\epsilon^2)$ ignorierend, bewies Paige, dass für die Fehlermatrix $F_k \in \mathbb{R}^{N\times k}$ die Abschätzung

$$\|F_k\| \leq \sqrt{k}\|A\|\epsilon_1.$$

richtig ist, und dass die Ritzwerte im Wesentlichen von den Eigenwerten von A eingeschlossen werden:

$$\lambda_1 - \eta_k \leq \theta_j^{(k)} \leq \lambda_N + \eta_k, \quad j = 1, \ldots, k.$$

Die in (2.7) definierten Zahlen $\delta_{k,j}$ spielen auch beim Rechnen in endlicher Arithmetik eine wichtige Rolle, denn aus der einfachen Fehlerschranke ([26] S.69) folgt

$$\min_{1\leq l\leq N} |\lambda_l - \theta_j^{(k)}| \leq \frac{||Az_j^{(k)} - \theta_j^{(k)} z_j^{(k)}||}{||z_j^{(k)}||} \leq \frac{1}{||z_j^{(k)}||}(\delta_{k,j} + \sqrt{k}||A||\epsilon_1). \quad (2.11)$$

Für Ritzvektoren $z_j^{(k)}$, deren Norm ungefähr gleich eins ist, folgt aus einem kleinen $\delta_{k,j}$, dass $\theta_j^{(k)}$ in der Nähe eines Eigenwertes von A liegt. Ist die Norm eines berechneten Ritzvektors aber sehr viel kleiner als eins, $||z_j^{(k)}|| \ll 1$, so kann man anhand eines kleinen $\delta_{k,j}$ aus (2.11) diese Schlußfolgerung nicht ziehen. Es gilt aber

Satz 2.4 (Paige, [25]) *Für jeden im kten Schritt des einfachen Lanczosalgorithmus berechneten Ritzwert $\theta_j^{(k)}$ gilt:*

$$\min_{1\le l\le N} |\lambda_l - \theta_j^{(k)}| \le \max\{2.5(\delta_{k,j} + \sqrt{k}||A||\epsilon_1), (k+1)^3||A||\epsilon_2\}, \qquad (2.12)$$

$$|q_{k+1}^* z_j^{(k)}| = \frac{|\epsilon_{jj}^{(k)}|}{\delta_{k,j}}, \quad |\epsilon_{jj}^{(k)}| \le k||A||\epsilon_2, \qquad (2.13)$$

mit $\left(\epsilon_{lj}^{(k)}\right)_{1\le l,j\le k} = S_k^*(K_k + N_k)S_k$. *Dabei ist K_k das strikte obere Dreieck der schiefsymmetrischen Matrix $F_k^*Q_k - Q_k^*F_k$ und N_k ist das strikte obere Dreieck von $\Delta_k T_k - T_k\Delta_k$.*

Damit folgt für $\delta_{k,j} \ll 1$ aus (2.12) die Konvergenz eines Ritzwertes $\theta_j^{(k)}$ gegen einen Eigenwert λ von A unabhängig von der Norm des Ritzvektors. Gleichung (2.13) wird wie folgt interpretiert: Geht die Orthogonalität zwischen q_{k+1} und $z_j^{(k)}$ verloren, so muß $\delta_{k,j}$ klein sein und somit approximiert $\theta_j^{(k)}$ einen Eigenwert. Die Umkehrung gilt nicht. Wir wollen noch darauf hinweisen, dass Paige diesen Zusammenhang durch einen Basiswechsel herstellt. Im $k$ten Schritt des einfachen Lanczosalgorithmus in exakter Arithmetik hat man zwei orthogonale Basen des Krylowraumes $\mathcal{K}_k(A, q_1)$, nämlich die Lanczosbasis Q_k und die Ritzbasis $Z_k = Q_k S_k$. Paige mißt mit (2.13) den Orthogonalitätsverlust zwischen dem zuletzt berechneten Lanczosvektor und den Ritzvektoren.

Das Lanczosphänomen

Auf einem Computer implementiert wird, wie bereits erwähnt, Algorithmus 2.2 fast nie abbrechen. Man beobachtet, dass, obwohl die Orthogonalität unter den Lanczosvektoren verloren geht, die Ritzvektoren in der Regel sehr gute Approximationen an die Eigenwerte sind. Weiter findet man eventuell sehr viele Ritzwerte in einem Radius von $\mathcal{O}(\epsilon||A||)$ um den größten Eigenwert von A, bevor man eine einzige Ritzapproximation an einen Eigenwert in der Mitte des Spektrums von A findet. Trotzdem werden meistens alle Eigenwerte unter den Ritzwerten auftauchen. Die einzige (bekannte) Ausnahme ist der Fall, wo A eine Diagonalmatrix ist und der Startvektor q_1 eine Komponente gleich Null hat. Diese Eigenschaft, dass auch in endlicher Arithmetik alle Eigenwerte gefunden werden, wenn man den Algorithmus

2.2 nur lange genug laufen läßt, wird als *Lanczosphänomen* bezeichnet (vgl. [6] S.102).

Eine Abschätzung für den Fehler der berechneten Ritzwerte, die auch geeignet ist, das Lanczosphänomen zu erklären, wurde das erste Mal in [8] entwickelt:

Satz 2.5 (Theorem 2 in [8]) *Es sei* $||A|| \leq 0.9$, $\lambda_r = 0$, $u_r^* q_1 \neq 0$ *und* $k \geq 3$. *Es gelte außerdem* $9k^{2.5}\epsilon_2 \leq 1$, $k^2\epsilon_1 \leq 0.25\sqrt{0.095}|u_r^* q_1|$ *sowie (2.10), dann gibt es einen Index* $1 \leq j \leq k$, *so dass*

$$\left|\theta_j^{(k)} - \lambda_r\right| \leq \frac{1}{2}\left(\left(\frac{10\sqrt{k}}{\sqrt{6}|u_r^* q_1|}\right)^{2/(k-2)} - 1\right) \tag{2.14}$$

gilt.

Die wesentliche Voraussetzung ist dabei, dass der Startvektor q_1 Komponenten hat, die in Richtung des Eigenvektors u_r zeigen. Dann findet der Lanczosalgorithmus den dazu gehörenden Eigenwert λ_r. Die übrigen Voraussetzungen in Satz 2.5 sind technischer Natur und entspringen der Tatsache, dass im Beweis von den Optimalitätseigenschaften der Tschebyscheffpolynome (siehe z.B. [31] S.73 ff oder [34] S.143) Gebrauch gemacht wird. Da $k^{1/k}$ für $k \to \infty$ gegen 1 konvergiert, besagt (2.14), dass es mindestens einen Ritzwert geben wird, der den Eigenwert λ_r approximiert, sofern nur k groß genug ist, d.h. wenn man lange genug iteriert.

Die Aussage läßt sich nun dahingehend interpretieren, dass der einfache Lanczosalgorithmus alle Eigenwerte findet, denn mit einem geeigneten Shift κ und einer passenden Skalierung σ erreicht man, dass für $\tilde{A} = \frac{1}{\sigma}(A + \kappa I)$ die Norm kleiner oder gleich 0.9 ist, $||\tilde{A}|| \leq 0.9$ und dass $\tilde{\lambda}_r = \lambda_r + \kappa = 0$ ist. Mit $\tilde{T}_k = \frac{1}{\sigma}(T_k + \kappa I_k)$ geht die Lanczosrekursion (2.8) in

$$\tilde{A}Q_k = Q_k\tilde{T}_k + \frac{1}{\sigma}(\beta_{k+1}q_{k+1}e_k^* + F_k)$$

über, so dass dann $\tilde{A}$ die Voraussetzungen von Satz 2.5 erfüllt.

2.4 Ritzvektoren und Cluster von Ritzwerten

Die erste Frage, die sich A. Greenbaum und Z. Strakoš in [38] gestellt haben ist die, ob es möglich ist, die Größe von $\delta_{k,j}$ wie in (2.7) definiert nur aus Informationen über die Ritzwerte abzuschätzen. Das hätte den Vorteil, dass man selbst bei der kleinen Eigenwertaufgabe für T_k Eigenvektoren erst dann bestimmen müßte, wenn man weiß, dass die Approximationen gut genug sind. Diese Frage läßt sich zwar positiv beantworten (siehe Lemma 5.1, Lemma 5.2 und Theorem 5.3 in [38]), aber den dort angegebenen Schranken

haftet der Nachteil an, dass sie im Allgemeinen nicht angewendet werden können, um festzustellen, ob $\delta_{k,j}$ auch für alle Ritzwerte in einem Cluster klein ist oder nicht. Für Ritzvektoren finden wir (vgl. [23] S.138)

Lemma 2.6 ([38]) *Für jeden im kten Schritt von Algorithmus 2.2 berechneten Ritzvektor $z_j^{(k)}$ gilt*

$$||z_j^{(k)} - \xi_{i,j}^{(k)} u_i|| \leq \frac{\delta_{k,j} + \sqrt{k}||A||\epsilon_1}{\min\limits_{r \neq i} |\lambda_r - \theta_j^{(k)}|}, \quad mit \quad \xi_{i,j}^{(k)} = u_i^* z_j^{(k)}, \ \ i = 1, \ldots N. \tag{2.15}$$

Vermutung 2.7 ([38], Strakoš, Greenbaum) *Für jeden Ritzwert $\theta_j^{(k)}$, der zu einem Cluster von Ritzwerten gehört, wobei der Cluster eine* hinreichend große Lücke[1] *zu den übrigen Ritzwerten aufweist, ist $\delta_{k,j}$ klein. Das bedeutet nach Lemma 2.6, dass der zugehörige Ritzvektor eine gute Approximation an einen Eigenvektor von A darstellt.*

Außerdem wird in der Arbeit [38] eine Idee für eine ober Schranke gegeben: Ist δ der Durchmesser des Clusters und γ der minimale Abstand des Clusters zu den übrigen Ritzwerten, so vermuten A. Greenbaum und Z. Strakoš, dass $\delta_{k,j}$ für jeden Ritzwert im Cluster von der Größenordnung

$$\sqrt{\frac{\delta}{\gamma}} \tag{2.16}$$

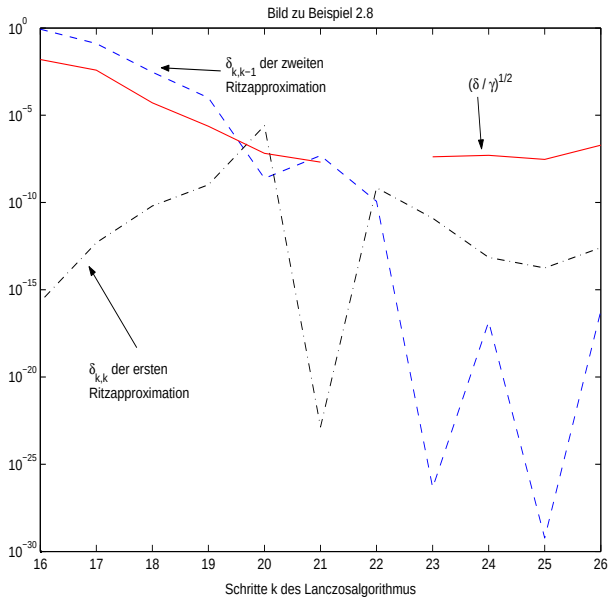
ist. Diese Vermutung wird gestützt durch Experimente mit Matrizen, die die folgende Eigenwertverteilung haben:

$$\begin{array}{c} \lambda_1 = 0.1, \ \ \lambda_N = 100, \\ \lambda_\nu = \lambda_1 + \frac{\nu-1}{N-1}(\lambda_N - \lambda_1)\rho^{N-\nu}, \ \ \nu = 2, \ldots, N-1 \end{array} . \tag{2.17}$$

Es wurde beobachtet, dass bei Anwendung des Lanczosalgorithmus auf derartige Matrizen für einige Parameterwerte, $0.6 \leq \rho \leq 0.8$ (vgl. [38] und [39]), die Orthogonalität unter den Lanczosvektoren sehr schnell verloren geht. Wir machen ein Experiment aus [38] nach:

Beispiel 2.8 *Zu den Parametern $N = 24$ und $\rho = 0.7$ in (2.17) wenden wir Algorithmus 2.2 mit zufällig gewähltem Startvektor q_1 an und beobachten den den Eigenwert $\lambda_{24} = 100$ approximierenden Cluster aus zwei Ritzwerten zwischen $k = 16$ und $k = 26$:*

[1] Wir werden im folgenden öfter die Formulierung *hinreichend gut isolierter Cluster* benutzen, worunter verstanden werden soll, dass die Ritzwerte in einem kleinen Intervall der Länge δ liegen und der Abstand zu den übrigen Ritz- und Eigenwerten γ ist, so dass $\delta \ll \gamma$ gilt.

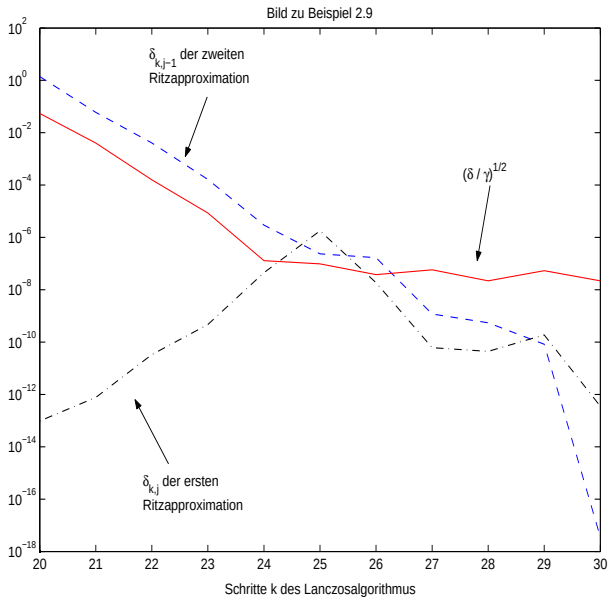


In dem Bild wird mit der punktiert gestrichelten Linie (schwarz) $\delta_{k,k}$ für die erste Ritzapproximation und mit der gestrichelten Linie (blau) $\delta_{k,k-1}$ für die zweite Ritzapproximation dargestellt. Im Vergleich dazu haben wir mit der durchgezogenen Linie (rot) (2.16) aufgetragen, also die Wurzel aus dem Abstand der beiden Ritzapproximationen zueinander dividiert durch den Abstand zu den übrigen Ritzwerten im $k$ten Schritt. Für $k = 22$ waren die beiden Ritzapproximationen allerdings rechnerisch im Computer identisch (und daher die Lücke in der roten Linie). Taucht das erste mal eine zweite Ritzapproximation in der Nähe von λ_{24} auf, so ist für diese die Zahl $\delta_{k,k-1}$ größer als (2.16). Erst nach ein paar weiteren Iterationsschritten fallen beide Ritznäherungen mit (2.7) unter (2.16).

Beispiel 2.9 *Wir betrachten die gleiche Matrix wie im vorherigen Beispiel und beobachten diesmal den Cluster, der* $\lambda_{22} \approx 44.7944$ *approximiert.*

Wir haben wieder mit der punktiert gestrichelten Linie die Größe (2.7) für die erste Ritzapproximation und mit der gestrichelten Linie (2.7) für die zweite Ritzapproximation aufgetragen; die durchgezogenen Linie stellt (2.16) dar. Es ergibt sich ein ähnliches Bild: Es dauert eine gewisse Zeit, bis nach dem Eintreten eines zusätzlichen Ritzwertes in einen Cluster für jede Ritzapproximation schließlich (2.7) kleiner als (2.16) ist.

Ein Gegenbeispiel
Wir benutzen nun eine spezielle Matrix, mit der wir demonstrieren, dass Vermutung 2.7 im Allgemeinen nicht richtig ist. Auch zeigt sich, dass die



Größen $\delta_{k,j}$ - anders als in den vorangegangenen beiden Beispielen - nicht zwangsläufig nach einer gewissen Zeit unter die in (2.16) angegebene Schranke fallen. Angenommen, wir haben eine Diagonalmatrix

$$A = \operatorname{diag}(\lambda_1, \ldots, \lambda_l, 0, -\lambda_1, \ldots, -\lambda_l) \in \mathbb{R}^{2l+1,2l+1} \tag{2.18}$$

und Vektoren mit der folgenden Struktur

$$y = (a_1, a_2, \ldots, a_l, a_{l+1}, -a_1, -a_2, \ldots, -a_l)^* \in \mathbb{R}^{2l+1}, \tag{2.19}$$

$$z = (b_1, b_2, \ldots, b_l, b_{l+1}, b_1, b_2, \ldots, b_l)^* \in \mathbb{R}^{2l+1}. \tag{2.20}$$

Dann rechnet man sofort nach, dass

$$y^*Ay = z^*Az = 0$$

und

$$\left\{ \begin{array}{c} Ay - z \\ Az - y \end{array} \right\} = \left\{ \begin{array}{c} (c_1, \ldots, c_l, c_{l+1}, c_1, \ldots, c_l)^* \\ (d_1, \ldots, d_l, d_{l+1}, -d_1, \ldots, -d_l)^* \end{array} \right\}$$

gelten. Wenden wir also Algorithmus 2.2 auf eine Matrix der Form (2.18) an mit einem Startvektor, der die Struktur (2.19) oder (2.20) hat, so erzeugt dieser Jacobimatrizen T_k, deren Diagonaleinträge gleich Null sind:

$$\alpha_1 = \alpha_2 = \ldots = \alpha_k = 0. \tag{2.21}$$

Und für Jacobimatrizen ist (2.21) äquivalent zu

$$\theta \in \Lambda(T_k) \quad \Rightarrow \quad -\theta \in \Lambda(T_k),$$

was insbesondere

$$0 \in \left\{ \begin{array}{lll} \in \Lambda(T_k) & , k & \text{ungerade} \\ \notin \Lambda(T_k) & , k & \text{gerade} \end{array} \right\}$$

impliziert. Für unser numerisches Beispiel wählen wir $A \in \mathbb{R}^{23\times 23}$ mit Eigenwerten

$$\begin{array}{c} \lambda_1 = 100,\ \lambda_{i+1} = \lambda_i - 0.1 \text{ für } i = 1, \dots 10 \\ \lambda_{12} = 0 \text{ und } \lambda_{12+i} = -\lambda_i \text{ für } i = 1, \dots, 11, \end{array} \tag{2.22}$$

Neben den symmetrisch zum Ursprung liegenden Eigenwerten haben wir auch die Spektrallücke zu $\lambda_{12} = 0$ groß gewählt, denn man weiß, dass Ritzwerte um so schneller gegen einen Eigenwert konvergieren, je größer dessen Lücke im Spektrum ist (siehe z.B. [26] S.242 ff). Als Startvektor nehmen wir $q_1 = \frac{1}{\sqrt{23}}(1, 1, \dots, 1)^* \in \mathbb{R}^{23}$ und beobachten den Cluster von Ritzwerten um $\lambda_{12} = 0$.

k	β_{k+1}	$\lvert s_{k,j}^{(k)}\rvert$	$\lvert s_{k,j+1}^{(k)}\rvert$	$\sqrt{\frac{\delta_k}{\gamma_k}}$
9	0.5164	9.0268×10^{-07}	-	-
10	99.4996	7.0710×10^{-01}	7.0710×10^{-01}	9.7023×10^{-05}
11	0.4924	4.6849×10^{-09}	-	-
12	99.4995	7.0710×10^{-01}	7.0710×10^{-01}	6.8258×10^{-06}
13	0.4626	2.1383×10^{-11}	-	-
14	99.4993	7.0710×10^{-01}	7.0709×10^{-01}	4.6544×10^{-07}
15	0.4254	1.0778×10^{-13}	-	-
16	99.4991	4.8387×10^{-01}	8.7513×10^{-01}	3.0700×10^{-08}
17	0.3782	4.5013×10^{-16}	-	-
18	99.4989	6.8904×10^{-01}	7.2471×10^{-01}	1.8764×10^{-09}
19	0.3185	2.4808×10^{-17}	-	-
20	98.9435	9.9589×10^{-01}	9.0499×10^{-02}	2.0156×10^{-10}

Tabelle 1

In der zweiten Spalte von Tabelle 1 steht die Norm des jeweils neuen Lanczosvektors, d.h. $\beta_{k+1} = ||q_{k+1}||$. In der fünften Spalte findet sich die in (2.16) erwähnte Schranke. Für k gerade finden sich zwei Ritzwerte, die $\lambda_{12} = 0$ approximieren, und zu denen sich die Beträge der letzten Elemente der Eigenvektoren von T_k in den Spalten 3 und 4 finden. Der Strich signalisiert, dass für k ungerade nur eine Ritznäherung zu λ_{12} vorlag. Man stellt fest, dass für die beiden Ritzapproximationen (wenn also k eine gerade Zahl ist) $\delta_{k,j}$ und $\delta_{k,j+1}$ sehr viel größer als $\sqrt{\frac{\delta_k}{\gamma_k}}$ sind. Offensichtlich kann $\delta_{k,j}$ nicht für jeden hinreichend gut isolierten Cluster von Ritzwerten proportional zu (2.16) sein.

Wir untersuchen nun, wie die Anzahl von Ritzwerten in einem hinreichend gut isolierten Cluster in zwei aufeinanderfolgenden Schritten des Lanczosalgorithmus Aussagen über die Qualität der Ritzapproximationen zulassen. Dazu führen wir als weitere Bezeichnungen

$$\psi_k(\lambda) = \det(\lambda I - T_k) \quad \text{und} \quad \psi_{2,k}(\lambda) = \det(\lambda I - T_{2,k}),$$

ein, wobei

$$T_{2,k} = \begin{bmatrix} \alpha_2 & \beta_3 & & & \\ \beta_3 & \alpha_3 & \beta_4 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \end{bmatrix}$$

ist. Für die normierten Eigenvektoren von T_k gelten die wohl bekannten Formeln (siehe z.B. [26], S.129), und das sind die einleitend zu diesem Kapitel erwähnten Eigenvektor-Eigenwert Identitäten.

Lemma 2.10 ([23] S.103 ff) *Für die normierten Eigenvektoren in (2.6) gilt*

$$s_{1,j}^{(k)} s_{k,j}^{(k)} \psi_k'(\theta_j^{(k)}) = \beta_2\beta_3\cdots\beta_k \tag{2.23}$$

$$(s_{1,j}^{(k)})^2 \psi_k'(\theta_j^{(k)}) = \psi_{2,k}(\theta_j^{(k)}) \tag{2.24}$$

$$(s_{k,j}^{(k)})^2 \psi_k'(\theta_j^{(k)}) = \psi_{k-1}(\theta_j^{(k)}) \tag{2.25}$$

für $1 \le j \le k$.

Sind wie bei Jacobimatrizen die Nebendiagonalelemente von T_k alle ungleich Null, $\beta_2\beta_3\cdots\beta_k \neq 0$, so besitzen die Eigenwerte von T_{k-1} und von T_k die so genannte Interlacing Eigenschaft:

$$\theta_1^{(k)} < \theta_1^{(k-1)} < \theta_2^{(k)} < \theta_2^{(k-1)} < \ldots < \theta_{k-1}^{(k-1)} < \theta_k^{(k)}.$$

Das gilt auch für die Eigenwerte von $T_{2,k}$ und von T_k.

Die in (2.7) definierten Zahlen $\delta_{k,j}$ sind das Produkt aus einem Element der Nebendiagonale einer Jacobimatrix, β_{k+1}, und Beträgen der Elemente der letzten Zeile der Eigenvektormatrix S_k. Da die Nebendiagonalelemente von Jacobimatrizen aber durch

$$0 < 2\beta_l < \theta_k^{(k)} - \theta_1^{(k)} \le \lambda_N - \lambda_1 + 2\eta_k, \quad l = 2, 3, \ldots, k.$$

beschränkt sind, konzentrieren wir uns zunächst auf die letzte Zeile der Eigenvektormatrix T_k. Analog zu Lemma 5.1 in [38] erhalten wir allein durch Ausnutzung der Interlacing Eigenschaft der Eigenwerte von T_k und T_{k+1}:

Lemma 2.11 *Für $i \geq 1$ und $i + c \leq k + 1$, $c \geq 1$, sei $\delta = \theta_{i+c}^{(k+1)} - \theta_i^{(k)}$ und*

$$\gamma = \min \left\{ \theta_i^{(k)} - \theta_i^{(k+1)}, \theta_{i+c}^{(k)} - \theta_{i+c}^{(k+1)} \right\},$$

wobei wir $\theta_{k+1}^{(k)} = +\infty$ setzen. Dann gilt

$$(s_{k+1,j}^{(k+1)})^2 < \frac{\delta}{\gamma}, \tag{2.26}$$

für $j = i + 1, \ldots, i + c$.

Beweis: Ist $j \in \{i + 1, \ldots, i + c\}$, so folgt aus (2.25), dass

$$\begin{aligned}
(s_{k+1,j}^{(k+1)})^2 &= \frac{\psi_k(\theta_j^{(k+1)})}{\psi'_{k+1}(\theta_j^{(k+1)})} = \frac{\prod_{l=1}^{k} (\theta_j^{(k+1)} - \theta_l^{(k)})}{\prod_{\substack{l=1 \\ l \neq j}}^{k+1} (\theta_j^{(k+1)} - \theta_l^{(k+1)})} \\
&\leq \frac{\delta}{(\theta_j^{(k+1)} - \theta_i^{(k+1)})} \frac{\prod_{\substack{l=1 \\ l \neq i}}^{k} (\theta_j^{(k+1)} - \theta_l^{(k)})}{\prod_{\substack{l=1 \\ l \neq j,i}}^{k+1} (\theta_j^{(k+1)} - \theta_l^{(k+1)})} \\
&< \frac{\delta}{\theta_j^{(k+1)} - \theta_i^{(k)}} \leq \frac{\delta}{\gamma}.
\end{aligned}$$

gilt. Die beiden letzten Ungleichungen folgen dabei aus der Interlacing Eigenschaft der Eigenwerte von Jacobimatrizen, denn es ist $0 < \theta_j^{(k+1)} - \theta_l^{(k)} < \theta_j^{(k+1)} - \theta_l^{(k+1)}$ falls l kleiner als j ist, und $\theta_{l+1}^{(k+1)} - \theta_j^{(k+1)} > \theta_l^{(k)} - \theta_j^{(k+1)}$ für $l \geq j$. □

Jetzt muß man auf zwei Punkte hinweisen. Zum einen, im Gegensatz zu den Betrachtungen der vorangegangenen Beispiele, wurde in Lemma 2.11 der Clusterdurchmesser δ und die Spektrallücke γ anhand der Ritzwerte in **zwei** aufeinander folgenden Schritten des Lanczosalgorithmus definiert. Und zum anderen haben wir gerade ausschließlich die Interlacing Eigenschaft der Nullstellen von ψ_k und ψ_{k+1} ausgenutzt. Wenn aber, wie im vorangegangenen Lemma, die Anzahl der Ritzwerte in einem Cluster in zwei aufeinander folgenden Schritten konstant ist, können wir eine bessere Schranke als (2.26) entwickeln, die in Satz 2.14 angegeben werden wird. Zuvor benötigen wir noch ein weiteres Lemma, das Auskunft über die Lage von Extremstellen einer gewissen rationalen Funktion gibt.

Lemma 2.12 *Seien* $p(z) = \prod_{j=1}^{l}(z - x_j)$ *und* $q(z) = \prod_{j=1}^{l}(z - y_j)$ *Polynome mit Nullstellen* $x_j, y_j \in (-1, 1)$ *und es gelte*

$$\sum_{j=1}^{l} \frac{(x_j - y_j)(1 - x_j y_j)}{|z - x_j|^2 |z - y_j|^2} > 0 \quad (\textit{oder} \quad < 0) \quad \textit{für alle} \quad |z| = 1, \tag{2.27}$$

dann finden sich die Extremstellen von $\left|\frac{p(z)}{q(z)}\right|$ *auf dem Einheitskreis bei* $z = 1$ *und* $z = -1$.

Beweis: Das Bestimmen der Extremwerte von $\left|\frac{p(z)}{q(z)}\right|$ auf dem Einheitskreis ist gleichbedeutend damit, das Maximum und Minimum der Funktion

$$f(t) = \left|\frac{p(e^{it})}{q(e^{it})}\right|^2$$

für $0 \leq t \leq \pi$ zu finden. Der Zähler der Ableitung von f ist aber gerade

$$2\sin(t) \sum_{j=1}^{l} \frac{(x_j - y_j)(1 - x_j y_j)}{|z - x_j|^2 |z - y_j|^2} \prod_{\nu=1}^{m} |z - x_\nu|^2 |z - y_\nu|^2, \tag{2.28}$$

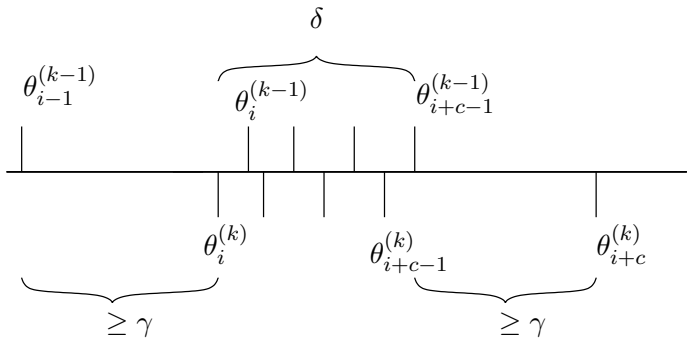
mit $z = e^{it}$. Da (2.28) nach (2.27) im Intervall von 0 bis π nur dann den Wert Null annimmt, falls $t = 0$ oder $t = \pi$ ist, ist der Beweis erbracht. □

Bemerkung 2.13 *Es ist offensichtlich, dass (2.27) erfüllt ist, wenn die Nullstellen wie folgt angeordnet sind:* $x_j \leq y_j$ *(bzw.* $y_j \leq x_j$*) für* $1 \leq j \leq l$ *und es gibt mindestens einen Index* ι*, so dass* $x_\iota < y_\iota$ *(bzw.* $y_\iota < x_\iota$*) gilt, womit auch der triviale Fall einer konstanten Funktion ausgeschlossen ist.*

Wir können nun eine obere Schranke für die Summe der Quadrate der Einträge der letzten Zeile der normierten Eigenvektoren von T_k entwickeln, deren Ritzvektoren zu einem hinreichend gut isolierten Cluster gehören. Interessant ist, dass die Anzahl der Ritzwerte im Cluster nicht in die Schranke mit einfließt. Wir setzen für den folgenden Satz noch $\theta_0^{(k-1)} = -\infty$ und $\theta_k^{(k-1)} = +\infty$.

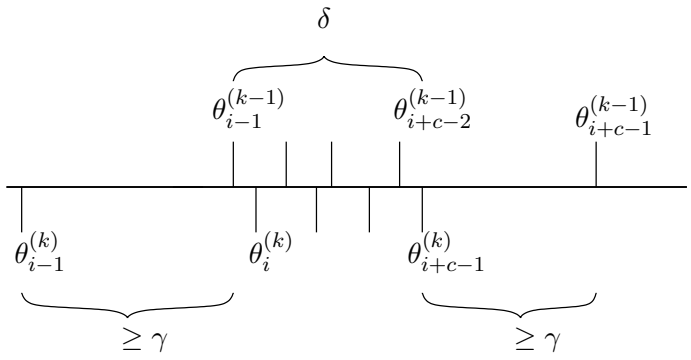
Satz 2.14 *Angenommen,* $1 \leq c \leq k-1$ *Nullstellen von* ψ_{k-1} *und* ψ_k *bilden einen Cluster mit Durchmesser* δ *und Spektrallücke* γ*, d.h. dass entweder*

$$\delta = \theta_{i+c-1}^{(k-1)} - \theta_i^{(k)}, \quad \gamma = \min\{\theta_{i+c}^{(k)} - \theta_{i+c-1}^{(k-1)}, \theta_i^{(k)} - \theta_{i-1}^{(k-1)}\} \tag{2.29}$$



mit $i \geq 1$ *und* $i + c \leq k$ *oder*

$$\delta = \theta_{i+c-1}^{(k)} - \theta_{i-1}^{(k-1)}, \quad \gamma = \min\{\theta_{i+c-1}^{(k-1)} - \theta_{i+c-1}^{(k)}, \theta_{i-1}^{(k-1)} - \theta_{i-1}^{(k)}\} \tag{2.30}$$



mit $i \geq 2$ *und* $i + c - 1 \leq k$ *gilt. Ist dann* $\frac{\delta}{2} < \gamma$*, so folgt*

$$\sum_{l=1}^{c} \left(s_{k,i+l-1}^{(k)}\right)^2 < 3\frac{\delta}{\gamma - \frac{\delta}{2}} \tag{2.31}$$

Beweis: Wir nehmen an, dass die Ritzwerte nach (2.29) verteilt sind. Der Beweis ändert sich kaum für (2.30). Wir bezeichnen den Mittelpunkt des Clusters mit $a = \frac{1}{2}\left(\theta_i^{(k)} + \theta_{i+c-1}^{(k-1)}\right)$ und rechnen nach, dass

$$\frac{a - \delta - \theta_{i+c-1}^{(k-1)}}{a - \delta - \theta_i^{(k)}} = 1 + \frac{\theta_i^{(k)} - \theta_{i+c-1}^{(k-1)}}{a - \delta - \theta_i^{(k)}} = 1 + \frac{-\delta}{-\frac{1}{2}\delta} = 3.$$

richtig ist. Und wir bezeichnen mit $B = B_\delta(a)$ die offene Kreisscheibe vom Radius δ mit Mittelpunkt a. Da die Nullstellen von ψ_{k-1} und ψ_k die Interlacing Eigenschaft besitzen, können wir Lemma 2.12 anwenden:

$$\begin{aligned} \max_{z\in\partial B}\left|\prod_{l=1}^{c}\frac{z-\theta_{i+l-1}^{(k-1)}}{z-\theta_{i+l-1}^{(k)}}\right| &= \max_{z\in\{a+\delta,a-\delta\}}\left|\prod_{l=1}^{c}\frac{z-\theta_{i+l-1}^{(k-1)}}{z-\theta_{i+l-1}^{(k)}}\right| \\ &< \frac{a-\delta-\theta_{i+c-1}^{(k-1)}}{a-\delta-\theta_i^{(k)}} = 3. \end{aligned}$$

Ferner überprüft man leicht, dass die folgenden beiden Ungleichungen für alle $z \in \partial B$ gelten:

$$\left| \frac{z - \theta_j^{(k-1)}}{z - \theta_j^{(k)}} \right| < 1, \quad \text{falls} \;\; 1 \leq j \leq i-1$$

und

$$\left| \frac{z - \theta_j^{(k-1)}}{z - \theta_{j+1}^{(k)}} \right| < 1, \quad \text{falls} \;\; i+c \leq j \leq k-1.$$

Damit schätzen wir nun das Kurvenintegral von $\frac{\psi_{k-1}}{\psi_k}$ entlang ∂B ab:

$$\begin{aligned} \left| \frac{1}{2\pi i} \int\limits_{\partial B} \frac{\psi_{k-1}(z)}{\psi_k(z)}\, dz \right| &\leq \delta \max_{z \in \partial B} \left| \frac{\psi_{k-1}(z)}{\psi_k(z)} \right| \\ &< \frac{\delta}{\min\limits_{z \in \partial B} |z - \theta_{i+c}^{(k)}|} \max_{z \in \partial B} \left| \prod_{l=1}^{c} \left(\frac{z - \theta_{k-l}^{(k-1)}}{z - \theta_{k-l+1}^{(k)}} \right) \right| \\ &< 3 \frac{\delta}{\gamma - \frac{\delta}{2}}. \end{aligned}$$

Man beachte, dass $\theta_j^{(k)}$ eine einfache Nullstelle von $\frac{\psi_k(\lambda)}{\psi_{k-1}(\lambda)}$ ist und aus diesem Grund

$$\frac{d}{d\lambda} \frac{\psi_k(\lambda)}{\psi_{k-1}(\lambda)} \bigg|_{\lambda = \theta_j^{(k)}} = \frac{\psi_k'(\theta_j^{(k)})}{\psi_{k-1}(\theta_j^{(k)})} \neq 0$$

gilt. Wir wenden (2.25) zusammen mit dem Residuensatz (siehe z.B. [2], S. 150) an und erhalten schließlich

$$\sum_{l=1}^{c} \left(s_{k,i+l-1}^{(k)} \right)^2 = \sum_{l=1}^{c} \frac{\psi_{k-1}(\theta_{i+l-1}^{(k)})}{\psi_k'(\theta_{i+l-1}^{(k)})} = \frac{1}{2\pi i} \int\limits_{\partial B} \frac{\psi_{k-1}(z)}{\psi_k(z)}\, dz. \tag{2.32}$$

□

Sind also im $(k-1)$ten und $k$ten Schritt des Lanczosalgorithmus gleich viele Ritzwerte in einem hinreichend gut isolierten Cluster, so ist nach (2.31) $\delta_{k,j}$ von der Größenordnung $\mathcal{O}\left(\sqrt{\frac{\delta}{\gamma}}\right)$, denn β_{k+1} ist beschränkt. Da nun Algorithmus 2.2 aufgrund von Rundungsfehlern fast nie abbricht, wird auch die Clustergröße zunehmen, was wir in den nächsten beiden Sätzen untersuchen werden. Nachfolgend nehmen wir an, dass die Anzahl der Ritzwerte in einem Cluster in den Schritten k und $k+1$ gleich ist. Der Satz ist also anwendbar, falls vom $(k-1)$ten zum $k$ten Schritt ein Ritzwert zum Cluster hinzugekommen ist.

Satz 2.15 *Es mögen $1 \leq c \geq k$ Nullstellen von ψ_k und ψ_{k+1} einen Cluster mit Durchmesser δ und Spektrallücke γ bilden, d.h. dass entweder*

$$\delta = \theta_{i+c}^{(k+1)} - \theta_i^{(k)}, \quad \gamma = \min\{\theta_{i+c}^{(k)} - \theta_{i+c}^{(k+1)}, \theta_i^{(k)} - \theta_i^{(k+1)}\} \tag{2.33}$$

mit $i \geq 1$ und $i + c \leq k$ oder

$$\delta = \theta_{i+c-1}^{(k)} - \theta_i^{(k+1)}, \quad \gamma = \min\{\theta_{i+c}^{(k+1)} - \theta_{i+c-1}^{(k)}, \theta_i^{(k+1)} - \theta_{i-1}^{(k)}\}$$

mit $i \geq 2$ und $i + c \leq k$ gilt. Ist dann $\frac{\delta}{2} < \gamma$, so folgt

$$\sum_{l=1}^{c} \left(s_{k,i-l+1}^{(k)}\right)^2 < 3\frac{(\sigma_{k+1} - \gamma + \frac{\delta}{2})^2}{\beta_{k+1}^2} \frac{\delta}{\gamma - \frac{\delta}{2}}, \tag{2.34}$$

wobei $\sigma_{k+1} = \theta_{k+1}^{(k+1)} - \theta_1^{(k+1)}$ ist. Liegt der Cluster am rechten Ende des Spektrums, d.h. ist

$$\delta = \theta_{k+1}^{(k+1)} - \theta_{k-c+1}^{(k)}, \quad \gamma = \theta_{k-c+1}^{(k+1)} - \theta_{k-c+1}^{(k)}$$

und ist ebenfalls $\frac{\delta}{2} < \gamma$, so folgt

$$\sum_{l=1}^{c} \left(s_{k,k-l+1}^{(k)}\right)^2 < 3\frac{(\sigma_{k+1} + \frac{\delta}{2})}{\beta_{k+1}^2} \delta. \tag{2.35}$$

(Natürlich gilt (2.35) auch, wenn sich der Cluster am linken Ende des Spektrums befindet.)

Beweis: Der Beweis unterscheidet sich nicht wesentlich von dem des Satzes 2.14. Wir beweisen (2.34) unter der Voraussetzung von (2.33). Dank der aus drei Termen bestehenden Rekursion der charakteristischen Polynome von Tridiagonalmatrizen gilt

$$\psi_{k+1}(\theta_j^{(k)}) = (\theta_j^{(k)} - \alpha_{k+1})\psi_k(\theta_j^{(k)}) - \beta_{k+1}^2\psi_{k-1}(\theta_j^{(k)}) = -\beta_{k+1}^2\psi_{k-1}(\theta_j^{(k)}).$$

Wir bezeichnen wieder mit $a = \frac{1}{2}\left(\theta_i^{(k)} + \theta_{i+c}^{(k+1)}\right)$ den Mittelpunkt des Clusters und mit $B = B_\delta(a)$ die offene Kreisscheibe vom Radius δ mit Mittelpunkt a. Anstelle von (2.32) gilt nun

$$\sum_{l=1}^{c} \left(s_{k,i+l-1}^{(k)}\right)^2 = -\frac{1}{\beta_{k+1}^2} \sum_{l=1}^{c} \frac{\psi_{k+1}(\theta_{i+l-1}^{(k)})}{\psi_k'(\theta_{i+l-1}^{(k)})} = -\frac{1}{\beta_{k+1}^2} \frac{1}{2\pi i} \int_{\partial B} \frac{\psi_{k+1}(z)}{\psi_k(z)}\, dz.$$

Wie im vorherigen Beweis führt die Standardabschätzung für Integrale, Lemma 2.12 und die Interlacing Eigenschaft der Nullstellen von ψ_k und ψ_{k+1} schließlich zu

$$\left| \frac{1}{2\pi i} \int_{\partial B} \frac{\psi_{k+1}(z)}{\psi_k(z)}\, dz \right| < 3\left(\sigma_{k+1} - \gamma + \frac{\delta}{2}\right)^2 \frac{\delta}{\gamma - \frac{\delta}{2}}.$$

□

Bei dem Gegenbeispiel mit der Matrix (2.22) ist die Anzahl der Ritzwerte in dem die Null approximierenden Cluster alternierend. Und wie Tabelle 1 zeigt, können wir nicht schlußfolgern, dass $\delta_{k,j}$ für jeden Ritzwert im Cluster klein ist, falls nur $\delta \ll \gamma$. Das ist allerdings nicht die ganze Wahrheit, denn Tabelle 1 zeigt auch, dass die Ritzapproximation für den Fall, dass k eine ungerade Zahl ist, sehr gut ist.

Lemma 2.16 *Angenommen, wir haben zwei Polynome* $p(z) = \prod_{j=1}^{l+1}(z - x_j)$ *und* $q(z) = \prod_{j=1}^{l}(z - y_j)$ *mit* $\delta > 0$ *und*

$$-\frac{\delta}{2} = x_1 < y_1 < x_2 < \ldots < y_l < x_{l+1} = \frac{\delta}{2}. \tag{2.36}$$

Dann gilt:

$$\max_{|z|=\delta} \left|\frac{p(z)}{q(z)}\right| \leq \frac{4}{3}\sqrt{3}\,\delta.$$

Beweis: a) Ist $|\Re(z)| \geq \frac{\delta}{2}$, so folgt für alle $|z| = \delta$ aufgrund der Interlacing Eigenschaft (2.36) der Polynome p und q:

$$\left|\frac{p(z)}{q(z)}\right| \leq \max\{|z - x_1|,\, |z - x_{l+1}|\}.$$

Und da $\delta^2 = (\Re(z))^2 + (\Im(z))^2$ gilt, erkennt man $(\Im(z))^2 \leq \frac{3}{4}\delta^2$. Also gilt

$$\begin{aligned}\left|\frac{p(z)}{q(z)}\right| &\leq \sqrt{\max\{(\Re(z) - x_1)^2, (\Re(z) - x_{l+1})^2\} + (\Im(z))^2} \\ &\leq \sqrt{\frac{9}{4}\delta^2 + \frac{3}{4}\delta^2} = \sqrt{3}\,\delta\end{aligned}$$

für $|z| = \delta$ mit $|\Re(z)| \geq \frac{\delta}{2}$.

b) Andererseits, ist $|\Re(z)| < \frac{\delta}{2}$, so muß auch $|\Im(z)| > \sqrt{\frac{3}{4}}\delta$ für $|z| = \delta$ gelten. Wiederum dank der Interlacing Eigenschaft (2.36) existiert ein Index $\iota \in \{1, \ldots, l\}$, so dass

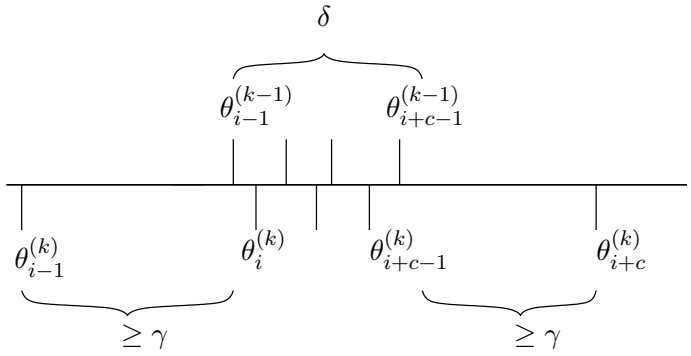
$$\begin{aligned}\left|\frac{p(z)}{q(z)}\right| &< \frac{|z - x_1|\,|z - x_{l+1}|}{|z - y_\iota|} \leq \frac{|z - x_1|\,|z - x_{l+1}|}{|\Im(z)|} \\ &< \sqrt{\frac{4}{3}}\frac{1}{\delta}\sqrt{((\Re(z) - x_1)^2 + (\Im(z))^2)\,((\Re(z) - x_{l+1})^2 + (\Im(z))^2)} \\ &< \sqrt{\frac{4}{3}}\frac{1}{\delta}\sqrt{(\delta^2 + \delta^2)(\delta^2 + \delta^2)} = \frac{4}{3}\sqrt{3}\,\delta\end{aligned}$$

richtig ist. □

Somit bekommen wir

Satz 2.17 *Es mögen $2 \le c+1 \le k-1$ Nullstellen von ψ_{k-1} und c Nullstellen von ψ_k einen Cluster mit Durchmesser δ und Spektrallücke γ bilden, d.h.*

$$\delta = \theta_{i+c-1}^{(k-1)} - \theta_{i-1}^{(k-1)}, \quad \gamma = \min\{\theta_{i-1}^{(k-1)} - \theta_{i-1}^{(k)}, \theta_{i+c}^{(k)} - \theta_{i+c-1}^{(k-1)}\}$$



für ein $i \ge 2$ mit $i + c \le k$. Ist dann $\frac{\delta}{2} < \gamma$, so folgt:

$$\sum_{l=1}^{c} \left(s_{k,i+l-1}^{(k)}\right)^2 < \frac{4}{3}\sqrt{3}\frac{\delta^2}{\left(\gamma - \frac{\delta}{2}\right)^2}$$

Ist die Anzahl der Nullstellen von ψ_k in einem Cluster mit Durchmesser δ und Spektrallücke γ gleich c und die Anzahl der Nullstellen von ψ_{k+1} gleich $c+1$, d.h.

$$\delta = \theta_{i+c}^{(k+1)} - \theta_{i}^{(k+1)}, \quad \gamma = \min\{\theta_{i}^{(k+1)} - \theta_{i-1}^{(k)}, \theta_{i+c}^{(k)} - \theta_{i+c}^{(k+1)}\}$$

und es ist entweder $i \ge 1$ und $i + c \le k$ oder $i \ge 2$ und $i + c \le k + 1$. Gilt dann $\frac{\delta}{2} < \gamma$, so folgt

$$\sum_{l=1}^{c} \left(s_{k,i+l-1}^{(k)}\right)^2 < \frac{4}{3}\sqrt{3}\frac{\left(\sigma_{k+1} - \gamma + \frac{\delta}{2}\right)}{\beta_{k+1}^2}\frac{\delta^2}{\left(\gamma - \frac{\delta}{2}\right)^2}. \tag{2.37}$$

Haben wir also ein Intervall der Länge δ, in dem sich in mehreren aufeinanderfolgenden Schritten ein Cluster befindet, und hat dieses Intervall einen Abstand von $\gamma > \frac{\delta}{2}$ zu den übrigen Ritzwerten, so folgt:

Korollar 2.18 *Es sei $\theta_\bullet^{(k)}$ der Ritzwert in einem Cluster, so dass $|s_{k,\bullet}^{(k)}|$ minimal ist. Wir bezeichnen mit c_k die Anzahl von Ritzwerten im $k$ten Schritt im Cluster. Dann gilt:*

$$\delta_{k,\bullet} \le \begin{cases} \sqrt{3} & \beta_{k+1} & \frac{1}{\sqrt{c_k}} & \sqrt{\frac{\delta}{\gamma - \frac{\delta}{2}}} & , \quad c_{k-1} = c_k \\ \sqrt{3} & \left(\sigma_{k+1} - \gamma + \frac{\delta}{2}\right) & \frac{1}{\sqrt{c_k}} & \sqrt{\frac{\delta}{\gamma - \frac{\delta}{2}}} & , \quad c_{k+1} = c_k \\ \frac{4}{\sqrt{3}} & \beta_{k+1} & \frac{1}{\sqrt{c_k}} & \frac{\delta}{\gamma - \frac{\delta}{2}} & , c_{k-1} = c_k + 1 \\ \frac{4}{\sqrt{3}} & \left(\sigma_{k+1} - \gamma + \frac{\delta}{2}\right) & \frac{1}{\sqrt{c_k}} & \frac{\delta}{\gamma - \frac{\delta}{2}} & , c_{k+1} = c_k + 1 \end{cases} \tag{2.38}$$

Damit kann man auch schlußfolgern, dass ein *Durchgangscluster*, worunter wir einen Cluster verstehen, der nicht in der Nähe eines Eigenwertes liegt, nur aus einem Ritzwert bestehen kann. Knizhnerman hat in [19] ein ähnliches Resultat generiert, doch erscheint uns die dortige Beweisführung wesentlich komplizierter.

Angenommen, wir haben zwei benachbarte Eigenwerte λ_j und λ_{j+1}, die einen hinreichend großen Abstand von 2γ zueinander haben. Befänden sich in zwei aufeinanderfolgenden Schritten des Lanczosalgorithmus jeweils ein Ritzwert ungefähr bei $\frac{1}{2}(\lambda_j + \lambda_{j+1})$ und deren Abstand zueinander wäre $\delta \ll \gamma$, so würde aus Satz 2.14 und Satz 2.4 folgen, dass sich dort auch ein Eigenwert befände, was aber von vorneherein ausgeschlossen ist.

Nach dem so genannten *Persistance Theorem* (siehe [25], Formel (3.9) dort) gilt für $t > k$:

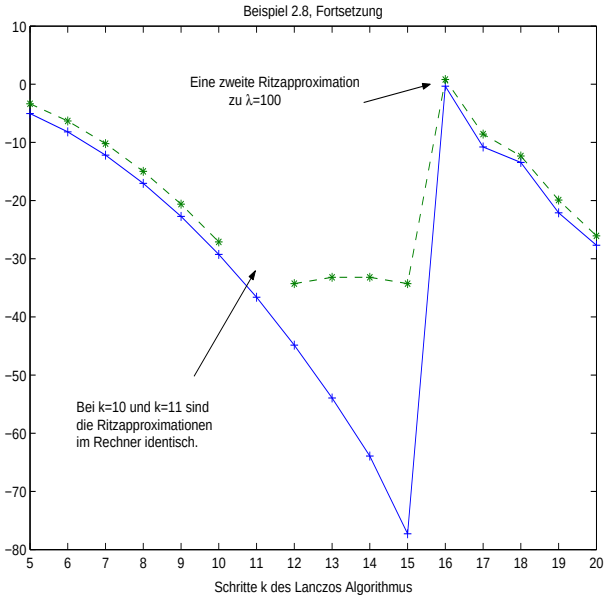
$$\min_{1 \le l \le t} |\theta_\bullet^{(k)} - \theta_l^{(t)}| \le \delta_{k,\bullet},$$

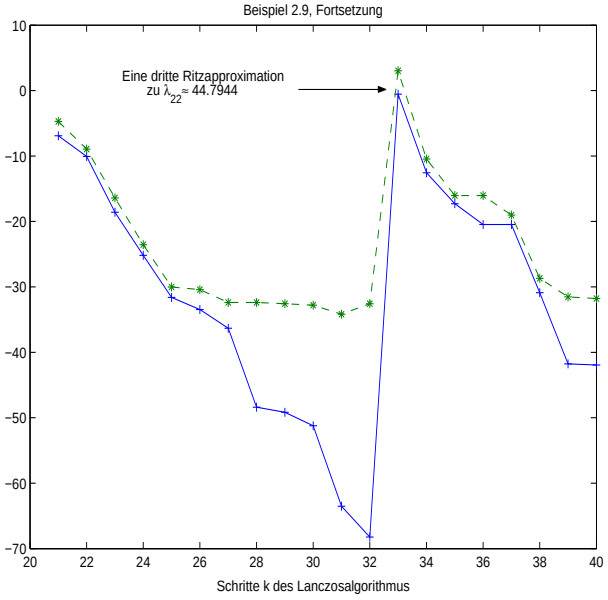
was zusammen mit (2.38) bedeutet: Hat sich einmal ein Cluster gebildet, so bleibt demzufolge immer mindestens ein Ritzwert in der Nähe eines Eigenwertes.

Fortsetzung von Beispiel 2.8 und Beispiel 2.9

Wir verwenden die Daten, d.h. Matrix und Startvektor aus Beispiel 2.8. In dem Diagramm wird für die Ritzapproximationen an λ_{24} nun $\sum_{\text{Cluster}} \left(s_{k,\bullet}^{(k)}\right)^2$ und die obere Schranke aus den Sätzen 2.14, 2.15 und 2.17 aufgetragen. Genauer benutzen wir bis $k = 14$ die Schranke (2.31), bei $k = 15$ das Minimum von (2.31) und (2.37), bei $k = 16$ wurde (2.35) und ab $k = 17$ wieder (2.31) verwendet.

Fortsetzung von Beispiel 2.9, d.h. Betrachtung des Cluster zu $\lambda_{22} \approx 44.7944$. Das Diagramm zeigt wieder $\sum_{\text{Cluster}} \left(s_{k,\bullet}^{(k)}\right)^2$ und die Schranken aus den Sätzen 2.14, 2.15 und 2.17: Für $k = 32$ ist das Minimum aus (2.31) und (2.37), für $k = 33$ die obere Schranke (2.34) und sonst (2.31) aufgetragen.


Beispiel 2.8, Fortsetzung
Eine zweite Ritzapproximation
zu λ=100
Bei k=10 und k=11 sind
die Ritzapproximationen
im Rechner identisch.
Schritte k des Lanczos Algorithmus


Beispiel 2.9, Fortsetzung
Eine dritte Ritzapproximation
zu λ22 ≈ 44.7944
Schritte k des Lanczosalgorithmus

Kapitel 3

Stabilisierung der Gewichte

Ist in diesem Kapitel vom *cg*-Verfahren die Rede, so setzen wir implizit, d.h. ohne es jedesmal zu erwähnen, noch zusätzlich voraus, dass die symmetrische Matrix $A \in \mathbb{R}^{N\times N}$ auch positiv definit ist. Das ließe sich durch einen positiven Shift $\tilde{A} = A + \kappa I$, $\kappa > 0$, immer erreichen und bedeutet für das Spektrum lediglich eine Verschiebung auf der reellen Zahlengeraden. Da die Krylowunterräume $\mathcal{K}_k(A, q)$ zudem shiftinvariant sind, d.h. es gilt

$$\mathcal{K}_k(A, q) = \mathcal{K}_k(A + \kappa I, q), \quad \kappa \in \mathbb{R},$$

wie man sich leicht überlegen kann, bedeutet die Annahme der positiven Definitheit von A für den Lanczosalgorithmus keine Einschränkung, sie ist aber für das *cg*-Verfahren eine notwendige Voraussetzung. Als erstes zeigen wir nun den Zusammenhang zwischen Lanczosalgorithmus und *cg*-Verfahren auf.

Betrachtet man das lineare Gleichungssystem

$$Ax = b, \quad b \in \mathbb{R}^N, \tag{3.1}$$

dessen eindeutige Lösung $\tilde{x}$ ist, so lauten die Rekursionen des *cg*-Verfahrens bei gegebenem Startvektor $x_0 \in \mathbb{R}^N$ wie folgt:

Algorithmus 3.1 (*cg*-Verfahren) $x_0 \in \mathbb{R}^N$, $r_0 = b - Ax_0$, $p_0 = r_0$. *Für* $k = 1, 2, \ldots$

$$\begin{aligned}
\varrho_{k-1} &= \frac{r_{k-1}^* r_{k-1}}{p_{k-1}^* A p_{k-1}} \\
x_k &= x_{k-1} + \varrho_{k-1} p_{k-1} \\
r_k &= r_{k-1} - \varrho_{k-1} A p_{k-1} \\
\varsigma_k &= \frac{r_k^* r_k}{r_{k-1}^* r_{k-1}} \\
p_k &= r_k + \varsigma_k p_{k-1}
\end{aligned}$$

Die Vektoren $\{r_0, r_1, \ldots, r_{k-1}\}$ bilden eine orthogonale Basis des Krylowraumes $\mathcal{K}_k(A, r_0)$. Gilt für den Startvektor q_1 des Lanczosalgorithmus, dass $q_1 = \frac{r_0}{||r_0||}$ ist, so erhält man die Koeffizienten α_j, β_j des Lanczosalgorithmus aus denen des *cg*-Verfahrens und die Lanczosvektoren sind die normierten Residuen $r_k = b - Ax_k$ der Approximationen x_k des *cg*-Verfahrens an die Lösung $\tilde{x}$ von (3.1).

$$\begin{aligned} q_{k+1} &= (-1)^k \frac{r_k}{||r_k||} \\ \alpha_k &= \frac{1}{\varrho_{k-1}} + \frac{\varsigma_{k-1}}{\varrho_{k-2}} \\ \beta_{k+1} &= \frac{\sqrt{\varsigma_k}}{\varrho_{k-1}} \end{aligned}$$

wobei $\varsigma_0 = 0$ und $\varrho_{-1} = 1$ gesetzt wurden. Das bedeutet, dass beide Verfahren ihre Näherungslösungen über bzw. aus dem gleichen Krylowraum gewinnen, denn für die *cg*-Iterierten x_k gilt

$$x_k \in x_o + \mathcal{K}_k(A, r_0), \tag{3.2}$$

d.h. es gibt einen Vektor $y_k \in \mathbb{R}^k$, so dass

$$x_k = x_0 + Q_k y_k \tag{3.3}$$

gilt. Nutzt man aus, dass r_k orthogonal zur Lanczosbasis $\{q_1, \ldots, q_k\}$ ist und dass $q_1 = \frac{r_0}{||r_0||}$ gilt, so folgt mit (3.3) (siehe [40])

$$\begin{aligned} 0 &= Q_k^* r_k = Q_k^* (b - Ax_k) = Q_k(b - Ax_0 - AQ_k y_k) \\ &= Q_k^* r_0 - Q_k^* A Q_k y_k = ||r_0|| e_1 - T_k y_k. \end{aligned}$$

Also kann man die Näherungslösungen des *cg*-Verfahrens gewinnen, indem man ein lineares Gleichungssystem mit der vom Lanczosalgorithmus erzeugten Jacobimatrix T_k löst.Umgekehrt wird in [15] S. 494 ff beschrieben, wie man die Koeffizienten ϱ_j, ς_j aus denen des Lanczosverfahrens erhält.

Der Lanczosalgorithmus zur Eigenwertapproximation und das *cg*-Verfahren zur Gleichungssystemlösung erzeugen sukzessive eine (und zwar die gleiche) Folge orthogonaler Polynome, wobei sich das innere Produkt als ein Riemann-Stieltjes Integral zu einer bestimmten Gewichtsfunktion $m(\lambda)$ darstellt. Bei exakter Rechnung erhalten wir nach N Schritten - falls $\mathcal{K}_N(A, q_1) = \mathbb{R}^N$ ist:

$$AQ_N = Q_N T_N, \quad T_N = S_N \Lambda S_N^*.$$

Die charakteristischen Polynome

$$1, \psi_1, \ldots, \psi_N$$

der führenden Hauptabschnitte von T_N sind orthogonal bezüglich des diskreten Skalarproduktes

$$(\phi, \varphi) = \sum_{j=1}^{N} m_j \phi(\lambda_j)\varphi(\lambda_j),$$

dabei sind die positiven Gewichte m_j gegeben durch das Quadrat des Kosinus des Winkels zwischen dem $j$ten Eigenvektor u_j von A und dem Startvektor q_1 des Lanczosalgorithmus

$$m_j = (q_1^* u_j)^2 = (s_{1,j}^{(N)})^2, \quad \sum_{j=1}^{N} m_j = 1. \tag{3.4}$$

Damit bekommen wir eine Gewichtsfunktion

$$m(\lambda) = \begin{cases} 0 & \text{falls} \quad \lambda < \lambda_1 \\ \sum\limits_{l=1}^{j} m_l & \text{falls} \quad \lambda_j \le \lambda < \lambda_{j+1} \\ 1 & \text{falls} \quad \lambda_N \le \lambda \end{cases} \tag{3.5}$$

und das Riemann-Stieltjes Integral ($\zeta \le \lambda_1$, $\lambda_N \le \xi$)

$$\int_{\zeta}^{\xi} f(\lambda) dm(\lambda) = \sum_{l=1}^{N} m_l f(\lambda_l). \tag{3.6}$$

Bezeichnet $\mathcal{M}_j$ die Menge aller Polynome vom Grad kleiner gleich j, deren Koeffizient der höchsten Potenz gleich eins ist, so erfüllen die Lanczospolynome die folgende Minimierungseigenschaft (siehe z.B. [13] S.25 ff.)

$$\psi_j = \arg \min_{\psi \in \mathcal{M}_j} \int_{\zeta}^{\xi} \psi^2(\lambda) dm(\lambda), \quad j = 0, \ldots, N. \tag{3.7}$$

Nach k Schritten des Lanczosalgorithmus angewandt auf A, q_1 erhalten wir eine Jacobimatrix $T_k = S_k \Theta_k S_k^*$ und wir können demzufolge das k-dimensionale Problem formulieren

$$(\phi, \varphi)_k = \sum_{j=1}^{k} m_j \phi(\theta_j^{(k)})\varphi(\theta_j^{(k)}), \quad m_j^{(k)} = (s_{1,j}^{(k)})^2, \quad \sum_{l=1}^{k} m_j^{(k)} = 1. \tag{3.8}$$

Das heißt, wir haben im k-ten Schritt die Gewichtsfunktion $m^{(k)}(\lambda)$ als

$$m^{(k)}(\lambda) = \begin{cases} 0 & \text{falls} \quad \lambda < \theta_1^{(k)} \\ \sum\limits_{l=1}^{j} m_l^{(k)} & \text{falls} \quad \theta_j^{(k)} \le \lambda < \theta_{j+1}^{(k)} \\ 1 & \text{falls} \quad \theta_k^{(k)} \le \lambda \end{cases} \tag{3.9}$$

Und die ersten k Polynome von $1, \psi_1, \psi_2, \ldots, \psi_N$, die durch (3.7) bzw. von der Gewichtsfunktion $m(\lambda)$ bestimmt werden, erhält man ebenfalls aus

$$\psi_j = \arg \min_{\psi \in \mathcal{M}_j} \int_{\zeta}^{\xi} \psi^2(\lambda) dm^{(k)}(\lambda), \quad j = 0, \ldots, k, \tag{3.10}$$

d.h. aus der Gewichtsfunktion $m^{(k)}(\lambda)$. Im nächsten Schritt des Lanczosalgorithmus werden dann die die Gewichtsfunktion $m^{(k)}(\lambda)$ bestimmenden Paare $\{\theta_j^{(k)}, m_j^{(k)}\}_{1 \le j \le k}$ durch die Paare $\{\theta_j^{(k+1)}, m_j^{(k+1)}\}_{1 \le j \le k+1}$ ersetzt, welche die Gewichtsfunktion $m^{(k+1)}(\lambda)$ bestimmen, und die Polynome $1, \psi_1, \ldots, \psi_k$ sind orthogonal bezüglich des durch $m^{(k+1)}(\lambda)$ definierten Skalarproduktes (3.8) bzw. (3.10).

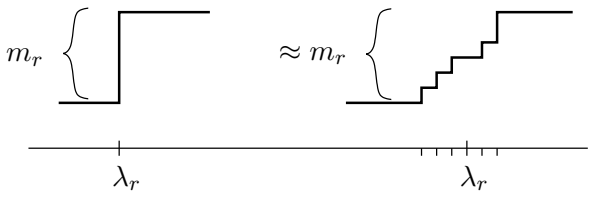
Während sich also die Gewichtsfunktionen in jedem Schritt ändern, bleibt die Orthogonalität unter den zuvor bestimmten Polynomen (3.9) auch bezüglich der jeweils neuen Skalarprodukte erhalten. Die Erwartung und Beobachtung ist, dass sich die Skalarprodukte (3.8) dem ursprünglichen (3.4) in einer bestimmten Weise *annähern*: Stabilisierung der Gewichte, und darunter verstehen wir folgendes:

a) In exakter Arithmetik: Nach k Schritten des Lanczosalgorithmus möge $\theta_j^{(k)}$ ein einfacher Ritzwert sein, der den Eigenwert λ_r approximiert, so dass $\delta_{k,j}$ klein ist und der Abstand von λ_r und $\theta_j^{(k)}$ zu den übrigen Ritz- und Eigenwerten möge hinreichend groß sein . Ist es richtig, dass dann
$$m_r \approx m_j^{(k)} \tag{3.11}$$
gilt? Diese Frage können wir mit Satz 3.4 bereits positiv beantworten.

b) Endliche Arithmetik: In der Arbeit [16] hat A. Greenbaum gezeigt, dass es nach k Schritten des Lanczosalgorithmus bzw. des *cg*-Verfahrens eine Matrix A_K (sowie beim *cg*-Verfahren eine rechte Seite b_K) gibt, so dass das in endlicher Arithmetik berechnete Ergebnis, also die Jacobimatrix T_k, das Gleiche ist als hätte man den Lanczosalgorithmus bzw. das *cg*-Verfahren mit der Matrix A_K (und dem Vektor b_K) exakt ausgeführt. Typischerweise ist die Dimension von A_K sehr viel größer als die von A: $\dim(A_K) \gg \dim(A)$. Ritzwerte bilden einen Cluster nur in der Nähe eines Eigenwertes. Die Matrix A_K, die nicht explizit berechnet wird, hat den Nachteil von der Anzahl der Iterationen k abzuhängen, es gibt in jedem Schritt des Lanczosalgorithmus eine andere Matrix $A_K = A_K(k)$. Daher stellt sich die Frage: Wenn man ein kleines Intervall vorgibt, in dem sich Ritzwerte (ohne die Anzahl genau zu spezifizieren) um einen Eigenwert λ_r clustern, ist es dann richtig, dass auch in diesem Fall die Gewichtsfunktion $m(\lambda)$ aus (3.5)

reproduziert wird, dass also

$$m_r \approx \sum_{\text{Cluster}} m_\bullet^{(k+l)}, \quad l \geq 1, \tag{3.12}$$



gilt? Die numerisch beobachtete Stabilisierung der Gewichte für einen beliebigen Cluster von Ritzwerten werden wir mit Hilfe der Sätze 3.7 und 3.8 nachweisen.

Wir beschreiben das an einem

Beispiel 3.1 *Wir wenden den Lanczosalgorithmus an auf eine Diagonalmatrix mit Eigenwerten (2.17),* $\rho = 0.7$ *und* $N = 24$*, und benutzen als Startvektor* $q_1 = \frac{1}{\sqrt{24}}e$*. Wir beobachten die Summe der Gewichte, die zu einem Cluster von Ritzwerten gehört, die alle den Eigenwert* $\lambda_{23} \approx 66.9896$ *approximieren. Dabei ist dann* $m_{23} = (q_1^* e_{23})^2 \approx 0.041666666666667$*. In der Tabelle wurden für einige Schritte k des Lanczosalgorithmus die Summe der Gewichte zu diesem Cluster in der zweiten Spalte sowie die Größe des Clusters* C_k *in der dritten Spalte aufgelistet. Man erkennt, daß nach einer kurzen anfänglichen Phase, sich die Gewichte* stabilisieren*. Und auch, wenn man den Lanczosalgorithmus weit über die Dimension* $N = 24$ *von A hinauslaufen läßt (*$k \geq 50$ *in Tabelle 2), bleibt die Summe der Gewichte zum Cluster konstant und gibt das Gewicht* m_{23} *wieder.*

k	$\sum_{\text{Cluster}} m_\bullet^{(k)}$	C_k
1	-	0
2	-	0
3	-	0
4	-	0
5	-	0
6	0.04214440362557	1
7	0.04168101932042	1
8	0.04166682193259	1
9	0.04166666731948	1
10	0.04166666666777	1
11	0.04166666666667	1
18	0.04166666666667	2
50	0.04166666666667	4
51	0.04166666666667	4
52	0.04166666666667	5
53	0.04166666666667	5

Tabelle 2

Die Beweisführung für (3.11) und (3.12) erfolgt, indem man die normierten Eigenvektoren von Jacobimatrizen betrachtet. Die Umformulierung von a) und b) mit Jacobimatrizen lautet daher: Angenommen, man hat eine Jacobimatrix T_K und betrachtet einen führenden Hauptabschnitt T_l, $l < K$, so dass jeweils ein Teil des Spektrums von T_K und T_l in dem gleichen Intervall der Länge δ liegt und so dass sich in nach links und rechts angrenzenden Intervallen, die mindestens die Länge γ haben, keine Eigenwerte von T_K bzw. T_l befinden: Sind dann die zu diesen beiden Clustern gehörenden Summen der Gewichte annähernd gleich, falls nur δ sehr viel kleiner als γ ist? In [38] findet sich:

Vermutung 3.2 (Strakoš, Greenbaum) *Es sei T_K eine Jacobimatrix und $C_{\nu,\eta}^{(K)} = \{\theta_\nu^{(K)}, \ldots, \theta_{\nu+\eta}^{(K)}\}$ eine nichtleere Teilmenge des Spektrums von T_K, $J = \{\nu, \ldots, \nu+\eta\}$. Ebenso sei für den führenden Hauptabschnitt T_l $C_{\bar\nu,\bar\eta}^{(l)} = \{\theta_{\bar\nu}^{(l)}, \ldots, \theta_{\bar\nu+\bar\eta}^{(l)}\}$ eine nichtleere Teilmenge der Eigenwerte von T_l, die die Punkte in $C_{\nu,\eta}^{(k)}$ approximieren, $\bar J = \{\bar\nu, \ldots, \bar\nu+\bar\eta\}$. Es sei*

$$\begin{aligned}
\gamma_{\nu,\eta}^{(K)} &= \min\left\{\theta_\nu^{(K)} - \theta_{\nu-1}^{(K)}, \theta_{\nu+\eta+1}^{(K)} - \theta_{\nu+\eta}^{(K)}\right\} \\
\gamma_{\bar\nu,\bar\eta}^{(l)} &= \min\left\{\theta_{\bar\nu}^{(l)} - \theta_{\bar\nu-1}^{(l)}, \theta_{\bar\nu+\bar\eta+1}^{(l)} - \theta_{\bar\nu+\bar\eta}^{(l)}\right\} \\
\gamma &= \min\left\{\gamma_{\nu,\eta}^{(K)}, \gamma_{\bar\nu,\bar\eta}^{(l)}\right\} \\
\delta &= \max\left\{|\theta_j^{(K)} - \theta_r^{(l)}|,\ j \in J,\ r \in \bar J\right\},
\end{aligned}$$

wobei wir $\theta_0^{(l)} = \theta_0^{(K)}$ *und* $\theta_{l+1}^{(l)} = \theta_{K+1}^{(K)}$ *formal auf* $-\infty$ *bzw.* $+\infty$ *gesetzt haben. Wird die Funktion* $h_{t,k}(\gamma,\delta)$ *definiert durch*

$$\sum_{j=\nu}^{\nu+\eta} m_j^{(k)} = \sum_{r=\bar{\nu}}^{\bar{\nu}+\bar{\eta}} m_r^{(l)} + h_{t,k}(\gamma,\delta),$$

so folgt aus $\delta \ll \gamma$*, dass*

$$|h_{t,k}(\gamma,\delta)| \ll 1 \tag{3.13}$$

ist.

Bei einer 3×3 Matrix läßt sich diese Vermutung noch leicht nachprüfen, indem man im Wesentlichen lediglich die Interlacing Eigenschaft der Nullstellen der charakteristischen Polynome von Jacobimatrizen ausnutzt.

Beispiel 3.3 *Es sei* $\delta = \theta_3^{(3)} - \theta_2^{(2)}$ *und* $\gamma = \theta_2^{(2)} - \theta_2^{(3)}$*, dann gilt:*

$$\left| m_2^{(2)} - m_3^{(3)} \right| < 3\frac{\delta}{\gamma}. \tag{3.14}$$

Beweis: Als erstes erhält man mit Hilfe von Lemma 2.10, dass

$$\Delta_2 = m_2^{(2)} - m_3^{(3)} = \frac{\beta_2^2}{\psi_1(\theta_2^{(2)})\psi_2'(\theta_2^{(2)})} - \frac{\beta_2^2\beta_3^3}{\psi_2(\theta_3^{(3)})\psi_3'(\theta_3^{(3)})}$$

gilt, und man kann leicht nachprüfen, dass

$$\beta_2^2 = (\theta_2^{(2)} - \theta_1^{(1)})(\theta_1^{(1)} - \theta_1^{(2)})$$

richtig ist. Als nächstes benutzen wir die drei Terme Rekursion der charakteristischen Polynome, d.h. wir brauchen $-\beta_3^2\psi_1(\theta_2^{(2)}) = \psi_3(\theta_2^{(2)})$. Einsetzen der Polynome liefert dann

$$\begin{aligned}
\Delta_2 &= \beta_2^2 \frac{\psi_2(\theta_3^{(3)})\psi_3'(\theta_3^{(3)}) + \psi_3(\theta_2^{(2)})\psi_2'(\theta_2^{(2)})}{\psi_1(\theta_2^{(2)})\psi_2'(\theta_2^{(2)})\psi_2(\theta_3^{(3)})\psi_3'(\theta_3^{(3)})} \\
&= (\theta_1^{(1)} - \theta_1^{(2)}) \left\{ \frac{(\theta_3^{(3)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_2^{(3)})(\theta_3^{(3)} - \theta_1^{(3)})}{(\theta_2^{(2)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_2^{(3)})(\theta_3^{(3)} - \theta_1^{(3)})} \right. \\
&\quad \left. - \frac{(\theta_2^{(2)} - \theta_2^{(3)})(\theta_2^{(2)} - \theta_1^{(3)})(\theta_2^{(2)} - \theta_1^{(2)})}{(\theta_2^{(2)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_2^{(3)})(\theta_3^{(3)} - \theta_1^{(3)})} \right\} \\
&= (\theta_1^{(1)} - \theta_1^{(2)}) \left\{ \frac{(\theta_3^{(3)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_2^{(3)}) - (\theta_2^{(2)} - \theta_2^{(3)})(\theta_2^{(2)} - \theta_1^{(2)})}{(\theta_2^{(2)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_2^{(3)})} \right\} \\
&\quad + \frac{\delta(\theta_1^{(1)} - \theta_1^{(2)})(\theta_2^{(2)} - \theta_2^{(3)})}{(\theta_3^{(3)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_2^{(3)})(\theta_3^{(3)} - \theta_1^{(3)})},
\end{aligned}$$

wobei wir auch $\theta_2^{(2)} - \theta_1^{(3)} = \theta_3^{(3)} - \theta_1^{(3)} - \delta$ benutzt haben. Der Rest unserer Rechnung besteht nun aus dem Ausnutzen der Interlacing Eigenschaft: $(\theta_1^{(1)} - \theta_1^{(2)}) < (\theta_3^{(3)} - \theta_1^{(3)})$, $(\theta_2^{(2)} - \theta_2^{(3)}) < (\theta_3^{(3)} - \theta_2^{(3)})$, $(\theta_1^{(1)} - \theta_1^{(2)}) < (\theta_2^{(2)} - \theta_1^{(2)})$ und $(\theta_3^{(3)} - \theta_1^{(2)}) > \gamma$. Somit gilt

$$\begin{aligned}\Delta_2 \;<\; & (\theta_1^{(1)} - \theta_1^{(2)}) \left\{ \frac{(\theta_3^{(3)} - \theta_1^{(2)}) - (\theta_2^{(2)} - \theta_1^{(2)})}{(\theta_2^{(2)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_1^{(2)})} \right\} \\ & + \frac{\delta(\theta_1^{(1)} - \theta_1^{(2)})}{(\theta_3^{(3)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_2^{(3)})} + \frac{\delta}{\gamma} < 3\frac{\delta}{\gamma}.\end{aligned}$$

Aus

$$(\theta_3^{(3)} - \theta_1^{(2)})(\theta_3^{(3)} - \theta_2^{(3)})(\theta_3^{(3)} - \theta_1^{(3)}) \;>\; (\theta_2^{(2)} - \theta_1^{(2)})(\theta_2^{(2)} - \theta_2^{(3)})(\theta_2^{(2)} - \theta_1^{(3)})$$

folgt außerdem noch, dass $\psi_2(\theta_3^{(3)})\psi_3'(\theta_3^{(3)}) + \psi_3(\theta_2^{(2)})\psi_2'(\theta_2^{(2)}) > 0$, d.h. $\Delta_2 > 0$ und damit ist (3.14) bewiesen. □

Dieses kleine Beispiel legt also nahe, dass die Vermutung 3.2 richtig ist. Hat sich noch kein *richtiger* Cluster gebildet, d.h. besteht der Cluster aus jeweils nur einem Ritzwert, so erhalten wir

Satz 3.4 *Für $l \geq 1$ sei*

$$\gamma = \min_{\substack{1 \leq i \leq k+l \\ i \neq \bar{j}}} \left| \theta_{\bar{j}}^{(k+l)} - \theta_i^{(k+l)} \right|,$$

dann gilt

$$\left| m_{\bar{j}}^{(k+l)} - m_j^{(k)} \right| \;<\; 2\frac{\delta_{k,j}}{\gamma}. \tag{3.15}$$

Beweis: Wir definieren $\tilde{s}_j^{(k)} = \left((s_j^{(k)})^*, 0^* \right)^* \in \mathbb{R}^{k+l}$ und setzen $\delta = \theta_{\bar{j}}^{(k+l)} - \theta_j^{(k)}$. Außerdem können wir annehmen, dass $\zeta = \angle \left(s_{j+1}^{(k+1)}, \tilde{s}_j^{(k)} \right) \in [0, \frac{\pi}{2}]$ (andernfalls, falls $\zeta \notin [0, \frac{\pi}{2}]$ wäre, würden wir $\tilde{s}_j^{(k)}$ als $-((s_j^{(k)})^*, 0^*)^*$ definieren). Nun gilt

$$\theta_j^{(k)} s_{1,j}^{(k)} = e_1^* T_{k+l} \tilde{s}_j^{(k)} \tag{3.16}$$

und

$$\theta_{\bar{j}}^{(k+l)} s_{1,\bar{j}}^{(k+l)} = e_1^* T_{k+l} s_{\bar{j}}^{(k+l)}. \tag{3.17}$$

Subtrahiert man (3.16) von (3.17), so erhält man

$$\theta_j^{(k)} \left(s_{1,\bar{j}}^{(k+l)} - s_{1,j}^{(k)} \right) + \delta s_{1,\bar{j}}^{(k+l)} = e_1^* T_{k+l} \left(s_{\bar{j}}^{(k+l)} - \tilde{s}_j^{(k)} \right),$$

und nimmt man davon den Betrag und macht Gebrauch von der Cauchy-Schwarz'schen Ungleichung, so ergibt sich

$$\left| \theta_j^{(k)} \right| \left| s_{1,\bar{j}}^{(k+l)} - s_{1,j}^{(k)} \right| \leq \sqrt{(\alpha_1^2 + \beta_2^2)}\, || s_{\bar{j}}^{(k+l)} - \tilde{s}_j^{(k)} || + \delta.$$

Da $s_{\bar{j}}^{(k+l)}$ und $\tilde{s}_j^{(k)}$ euklidische Länge gleich eins haben und $\zeta \in [0, \frac{\pi}{2}]$ liegt, folgt:

$$\|s_{\bar{j}}^{(k+l)} - \tilde{s}_j^{(k)}\| = \sqrt{2}\sin\left(\frac{\zeta}{2}\right) \leq \sin(\zeta) = |\sin(\zeta)|.$$

Mit $||T_{k+l}\tilde{s}_j^{(k)} - \theta_j^{(k)}\tilde{s}_j^{(k)}|| = \beta_{k+1}|s_{k,j}^{(k)}| = \delta_{k,j}$ liefert eine Anwendung von Satz 2.3 nun

$$\begin{aligned}\left|\theta_j^{(k)}\right| \left|s_{1,j+1}^{(k+1)} - s_{1,j}^{(k)}\right| &\leq \sqrt{(\alpha_1^2 + \beta_2^2)}\,|\sin(\zeta)| + \delta \\ &\leq \sqrt{(\alpha_1^2 + \beta_2^2)}\,\frac{\delta_{k,j}}{\gamma} + \delta.\end{aligned}$$

Verschiebt man das Spektrum von T_{k+l} um $\sigma \in \mathbb{R}$, so haben T_{k+l} und $T_{k+l} + \sigma I$ die gleichen Eigenvektoren. Zudem bleiben die Abstände unter den Eigenwerten gleich. Es gilt

$$\lim_{\sigma\to\infty} \frac{\|(T_{k+1} + \sigma I)e_1\|}{|\theta_j^{(k)} + \sigma|} = \lim_{\sigma\to\infty} \frac{\sqrt{(\alpha_1 + \sigma)^2 + \beta_2^2}}{|\theta_j^{(k)} + \sigma|} = 1.$$

Da zudem $\left|s_{1,\bar{j}}^{(k+l)} + s_{1,j}^{(k)}\right| < 2$ ist, folgt schließlich (3.15) aus

$$\left|m_{\bar{j}}^{(k+l)} - m_j^{(k)}\right| = \left|s_{1,\bar{j}}^{(k+l)} + s_{1,j}^{(k)}\right| \left|s_{1,\bar{j}}^{(k+l)} - s_{1,j}^{(k)}\right|.$$

□

Ist ein Ritzpaar $(\theta_j^{(k)}, z_j^{(k)})$ hinreichend gut konvergiert, so dass $\delta_{k,j}$ klein ist und besteht der Cluster aus Ritzwerten nur aus einem Element, so ist der Abstand der Gewichte für $\delta \ll \gamma$ klein, denn für konstante Clustergröße wissen wir mit Hilfe von Satz 2.14, daß $\delta_{k,j}$ in der Größenordnung von $\mathcal{O}\left(\sqrt{\frac{\delta}{\gamma}}\right)$ liegt. Anhand der Beispielmatrix (2.22) in Kombination mit Satz 2.17 haben wir gesehen, dass $\delta_{k,j}$ bei alternierender Clustergröße von der Ordnung $\mathcal{O}\left(\frac{\delta}{\gamma}\right)$ ist, wenn man im $k$ten Schritt einen Ritzwert weniger im Cluster hat als im $(k-1)$ten. Jedoch kann Satz 3.4 eigentlich nur herangezogen werden, um (3.11) zu erklären, nicht aber, für wachsende Clustergröße und auch nicht für den Fall, dass - wie in Tabelle 1 gesehen - $\delta_{k,j}$ groß ist.

Beispiel 3.5 *Lanczosalgorithmus angewandt auf eine Diagonalmatrix (2.22) mit Startvektor* $q_1 = \frac{1}{\sqrt{23}}e$. *Wir beobachten wieder den Cluster um* $\lambda_{12} = 0$*;* $m_{12} \approx 0.04347826086957$.

k	$\sum_{Cluster} m_{\bullet}^{(k)}$	C_k
1	1.00000000000000	1
2	-	0
3	0.04351690497075	1
4	-	0
5	0.04347826208743	1
6	0.04347826695856	2
7	0.04347826086960	1
8	0.04347826086981	2
9	0.04347826086957	1
10	0.04347826086957	2
11	0.04347826086957	1

Tabelle 3

Lemma 3.6 *Es sei* $p(t) = \prod_{j=1}^{M}(t - x_j)$ *mit reellen Nullstellen* x_j *und es sei* $a \in \mathbb{R}$ *ein Punkt mit* $a \geq x_j$ *(oder* $a \leq x_j$*) für* $j = 1, \ldots, M$*. Es sei* $r > 0$ *und* $B = B_r(a)$*, dann gilt*

$$\min_{z \in \partial B} |p(z)| = |p(a - r)| \quad (oder \ = |p(a + r)|).$$

Beweis: Das Bestimmen des Minimums des Betrages von p auf dem Kreis vom Radius r um den Mittelpunkt a bedeutet, dass man die folgende Funktion

$$f(t) = p(a + re^{it})p(a + re^{-it})$$

für $0 \leq t \leq \pi$ minimieren muß. Die Ableitung von f nach t ist

$$\begin{aligned} f'(t) &= ir\{e^{it}p'(a + re^{it})p(a + re^{-it}) - e^{-it}p(a + re^{it})p'(a + re^{-it})\} \\ &= 2r\sin(t)\sum_{j=1}^{M}(x_j - a)\prod_{\substack{\nu=1 \\ \nu \neq j}}^{M}\left|a + re^{it} - x_\nu\right|^2. \end{aligned}$$

Aufgrund der Lage der Nullstellen von p zum Mittelpunkt des Kreises ist f monoton in $[0, \pi]$. □

Damit können wir nun den Betrag der Differenz der Summe der Gewichte für einen hinreichend gut isolierten Cluster in zwei aufeinanderfolgenden Schritten des Lanczosalgorithmus nach oben abschätzen, auch im Falle einer alternierenden Clustergröße. Es gilt allerdings noch eine Bemerkung vorwegzunehmen. In den oberen Schranken (3.18) und (3.24) taucht ein Term der Form

$$\frac{(2k-2)^{2k-2}}{\left((2c-1)^{(2c-1)}\right)\left((2k-2c-1)^{(2k-2c-1)}\right)}$$

bzw.

$$\frac{(2k-2)^{2k-2}}{\left((2c-2)^{(2c-2)}\right)\left((2k-2c)^{(2k-2c)}\right)}$$

auf. Hierin ist k die Dimension der Jacobimatrix und c die Anzahl von Ritzwerten in dem betrachteten Cluster. Typischerweise ist in Anwendungen c sehr viel kleiner als k; ist z.B. $c = 1$, so wachsen die Terme nur linear mit k. Ungünstig, aber in Anwendungen kaum vorkommend, wäre der Fall, wenn $c = \frac{k}{2}$ ist, denn das würde ein exponentielles Wachstum dieser Terme in k bedeuten.

Satz 3.7 *Es mögen jeweils $c \geq 1$ Nullstellen von ψ_{k-1} und von ψ_k in einem Cluster mit Durchmesser δ und Spektrallücke γ liegen. Wir nehmen zusätzlich an, dass (2.30) gilt. Ist dann $2\delta < \gamma$, so folgt*

$$\left|\sum_{l=1}^{c} m_{i+l-1}^{(k)} - \sum_{l=1}^{c} m_{i+l-2}^{(k-1)}\right| < \left(\frac{\delta}{\gamma}\right)^{2c-1} \frac{1}{\left(1-\frac{2\delta}{\gamma}\right)^{2k-2}} \frac{(2k-2)^{2k-2}}{\left((2c-1)^{(2c-1)}\right)\left((2k-2c-1)^{(2k-2c-1)}\right)} \tag{3.18}$$

Beweis: Aufgrund der aus drei Termen bestehenden Rekursion der charakteristischen Polynome sieht man, dass

$$\psi_k(\theta_l^{(k-1)}) = -\beta_k^2 \psi_{k-2}(\theta_l^{(k-1)}) \quad \text{für} \quad l = 1, 2, \ldots, k-1$$

richtig ist. Damit, und mit Hilfe der Formeln (2.23), (2.24) und (2.25) für die normierten Eigenvektoren von Jacobimatrizen erhalten wir für das Produkt der Nebendiagonalelemente von T_k:

$$\begin{aligned}
\prod_{\nu=2}^{k} \beta_\nu^2 &= \left|\beta_k s_{1,l}^{(k-1)} s_{k-1,l}^{(k-1)} \psi'_{k-1}(\theta_l^{(k-1)}) s_{1,j}^{(k)} s_{k,j}^{(k)} \psi'_k(\theta_j^{(k)})\right| \\
&= \beta_k \sqrt{|\psi_{2,k-1}(\theta_l^{(k-1)})\psi_{k-2}(\theta_l^{(k-1)})|}\sqrt{|\psi_{2,k}(\theta_j^{(k)})\psi_{k-1}(\theta_j^{(k)})|} \\
&= \sqrt{|\psi_k(\theta_l^{(k-1)})\psi_{k-1}(\theta_j^{(k)})\psi_{2,k}(\theta_j^{(k)})\psi_{2,k-1}(\theta_l^{(k-1)})|}
\end{aligned}$$

für $1 \leq l \leq k-1$ und $1 \leq j \leq k$. Außerdem erfüllen auch die Nullstellen von ψ_k und von $\psi_{2,k}$ die strikte Interlacing Eigenschaft, somit folgt

$$\begin{aligned}
\left|\psi_{2,k}(\theta_{i+c-1}^{(k)})\right| &\leq \delta^{c-1} \prod_{j=i+c-1}^{k-1} |\theta_{i+c-1}^{(k)} - \mu_j^{(k)}| \prod_{j=1}^{i-1} (\theta_{i+c-1}^{(k)} - \mu_j^{(k)}) \\
&< \delta^{c-1} \prod_{j=i+c}^{k} |\theta_{i+c-2}^{(k-1)} - \theta_j^{(k)}| \prod_{j=1}^{i-1} (\theta_{i+c-1}^{(k)} - \theta_j^{(k)}),
\end{aligned}$$

wobei wir die Nullstellen von $\psi_{2,k}$ mit $\mu_j^{(k)}$ bezeichnet haben. Analog sieht man, dass

$$\left|\psi_{2,k-1}(\theta_{i+c-2}^{(k-1)})\right| < \delta^{c-1} \prod_{j=i+c-1}^{k-1} |\theta_{i+c-2}^{(k-1)} - \theta_j^{(k-1)}| \prod_{j=1}^{i-2} (\theta_{i+c-1}^{(k)} - \theta_j^{(k-1)}) \quad (3.19)$$

richtig ist. Damit folgt nun

$$\begin{aligned} \prod_{\nu=2}^{k} \beta_\nu^2 &< \delta^{2c-1} \prod_{j=1}^{i-1} (\theta_{i+c-1}^{(k)} - \theta_j^{(k)}) \prod_{j=1}^{i-2} (\theta_{i+c-1}^{(k)} - \theta_j^{(k-1)}) \\ &\quad \left| \prod_{j=i+c}^{k} \theta_{i+c-2}^{(k-1)} - \theta_j^{(k)} \right| \left| \prod_{j=i+c-1}^{k-1} \theta_{i+c-2}^{(k-1)} - \theta_j^{(k-1)} \right|. \end{aligned} \quad (3.20)$$

Wir wählen den größten Ritzwert im Cluster, also $\theta_{i+c-1}^{(k)}$, als Mittelpunkt eines Kreises mit Radius $\delta < r < \gamma - \delta$: $B = B_r(\theta_{i+c-1}^{(k)})$. Man beachte, dass für alle reellen Zahlen θ mit $\theta \geq \theta_{i+c-1}^{(k-1)}$ und für alle $z \in \partial B$

$$\frac{|\theta_{i+c-2}^{(k-1)} - \theta|}{|z - \theta|} \leq \frac{\theta - \theta_{i+c-2}^{(k-1)}}{\theta - \Re(z)} < \frac{1}{1 - \left(\frac{r+\delta}{\gamma}\right)} \quad (3.21)$$

gilt. Benutzt man wieder die Formeln für die normierten Eigenvektoren von T_k, so stellt sich die Differenz der Summe der Gewichte zu diesem Cluster als

$$\begin{aligned} \Delta_{k-1} &= \sum_{l=1}^{c} m_{i+l-1}^{(k)} - m_{i+l-2}^{(k-1)} = \sum_{l=1}^{c} \left(s_{1,i+l-1}^{(k)}\right)^2 - \left(s_{1,i+l-2}^{(k-1)}\right)^2 \\ &= \prod_{\nu=2}^{k} \beta_\nu^2 \left(\sum_{l=1}^{c} \frac{1}{\psi_{k-1}(\theta_{i+l-1}^{(k)}) \psi_k'(\theta_{i+l-1}^{(k)})} - \frac{1}{\beta_k^2 \psi_{k-2}(\theta_{i+l-2}^{(k-1)}) \psi_{k-1}'(\theta_{i+l-2}^{(k-1)})} \right) \\ &= \prod_{\nu=2}^{k} \beta_\nu^2 \left(\sum_{l=1}^{c} \frac{1}{\psi_{k-1}(\theta_{i+l-1}^{(k)}) \psi_k'(\theta_{i+l-1}^{(k)})} + \frac{1}{\psi_k(\theta_{i+l-2}^{(k-1)}) \psi_{k-1}'(\theta_{i+l-2}^{(k-1)})} \right) \end{aligned}$$

dar. Es ist $\theta_j^{(k)}$ eine Nullstelle von $\psi_k(\lambda)$ und daher gilt

$$(\psi_{k-1}\psi_k)'(\theta_j^{(k)}) = (\psi_{k-1}\psi_k')(\theta_j^{(k)}).$$

Ebenso gilt auch $(\psi_{k-1}\psi_k)'(\theta_j^{(k-1)}) = (\psi_{k-1}'\psi_k)(\theta_j^{(k-1)})$. Außerdem sind $\theta_j^{(k)}$ und $\theta_j^{(k-1)}$ einfache Nullstellen von $(\psi_{k-1}\psi_k)(\lambda)$ und somit folgt schließlich aus dem Residuensatz:

$$\Delta_{k-1} = \frac{1}{2\pi i} \prod_{\nu=2}^{k} \beta_\nu^2 \int_{\partial B} \frac{1}{\psi_{k-1}(z)\psi_k(z)}\, dz. \quad (3.22)$$

Wir definieren

$$g_1(z) = \frac{\prod_{j=1}^{i-1}(\theta_{i+c-1}^{(k)} - \theta_j^{(k)}) \prod_{j=1}^{i-2}(\theta_{i+c-1}^{(k)} - \theta_j^{(k-1)})}{\prod_{j=1}^{i+c-1}|z-\theta_j^{(k)}| \prod_{j=1}^{i+c-2}|z-\theta_j^{(k-1)}|}$$

und

$$g_2(z) = \prod_{j=i+c}^{k} \frac{|\theta_{i+c-2}^{(k-1)} - \theta_j^{(k)}|}{|z-\theta_j^{(k)}|} \prod_{j=i+c-1}^{k-1} \frac{|\theta_{i+c-2}^{(k-1)} - \theta_j^{(k-1)}|}{|z-\theta_j^{(k-1)}|},$$

dann liefern die Standardabschätzung für Kurvenintegrale, Lemma 3.6 sowie die Ungleichungen (3.20) und (3.21)

$$\begin{aligned}
\left| \frac{1}{2\pi i} \prod_{\nu=2}^{k} \beta_\nu^2 \int_{\partial B} \frac{dz}{\psi_{k-1}(z)\psi_k(z)} \right| &\leq r \frac{\prod_{\nu=2}^{k} \beta_\nu^2}{\min_{z\in\partial B} |\psi_k(z)\psi_{k-1}(z)|} \\
&< r\delta^{2c-1} \min_{z\in\partial B} g_1(z) \min_{z\in\partial B} g_2(z) \\
&< r\delta^{2c-1} g_1(\theta_{i+c-1}^{(k)} - r) \left(\frac{1}{1-\frac{r+\delta}{\gamma}} \right)^{2(k-i-c+1)} \\
&< \frac{\delta^{2c-1}}{(r-\delta)^{2c-1}} \left(\frac{1}{1-\frac{r+\delta}{\gamma}} \right)^{2(k-c)-1} . \qquad (3.23)
\end{aligned}$$

Da die Identität (3.22) für alle $\delta < r < \gamma - \delta$ gilt, müssen wir noch den Radius r finden, der die Schranke (3.23) minimiert; mit anderen Worten wir müssen das Maximum der Funktion $f(r) = (r-\delta)^\alpha (1 - \frac{r+\delta}{\gamma})^\beta$ bestimmen. Eine einfache Rechnung zeigt

$$\max_{\delta<r<\gamma-\delta} f(r) = \frac{\alpha^\alpha \beta^\beta}{(\alpha+\beta)^{\alpha+\beta}} \gamma^\alpha \left(1 - \frac{2\delta}{\gamma}\right)^{\alpha+\beta}$$

und daraus ergibt sich (3.18). □

Wächst die Größe des Clusters um einen Ritzwert, so gilt

Satz 3.8 *Angenommen, es liegen $c-1 \geq 1$ Nullstellen von ψ_{k-1} und c Nullstellen von ψ_k in einem Cluster mit Durchmesser δ und Spektrallücke γ, d.h.*

$$\delta = \theta_{i+c-1}^{(k)} - \theta_i^{(k)}, \quad \gamma = \min\{\theta_i^{(k)} - \theta_{i-1}^{(k-1)}, \theta_{i+c-1}^{(k-1)} - \theta_{i+c-1}^{(k)}\}.$$

Ist $2\delta < \gamma$, so gilt

$$\left|\sum_{l=1}^{c} m_{i+l-1}^{(k)} - \sum_{l=1}^{c-1} m_{i+l-1}^{(k-1)}\right| < \left(\frac{\delta}{\gamma}\right)^{2c-2} \frac{1}{\left(1-\frac{2\delta}{\gamma}\right)^{2k-2}} \frac{(2k-2)^{2k-2}}{\left((2c-2)^{(2c-2)}\right)\left((2k-2c)^{(2k-2c)}\right)} \tag{3.24}$$

Beweis: Der Beweis unterscheidet sich nur unwesentlich vom vorherigen. Zunächst ist (3.19) durch

$$\left|\psi_{2,k-1}(\theta_{i+c-2}^{(k-1)})\right| < \delta^{c-2} \prod_{j=1}^{i-1} (\theta_{i+c-1}^{(k)} - \theta_j^{(k-1)}) \prod_{j=i+c-1}^{k-1} |\theta_{i+c-2}^{(k-1)} - \theta_j^{(k-1)}|,$$

zu ersetzen, so daß wir für die obere Schranke des Produktes der Subdiagonalelemente statt (3.20) nun

$$\prod_{\nu=2}^{k} \beta_\nu^2 < \delta^{2c-2} \prod_{j=1}^{i-1} (\theta_{i+c-1}^{(k)} - \theta_j^{(k-1)}) \prod_{j=i+c-1}^{k-1} |\theta_{i+c-2}^{(k-1)} - \theta_j^{(k-1)}| \prod_{j=1}^{i-1} (\theta_{i+c-1}^{(k)} - \theta_j^{(k)}) \prod_{j=i+c}^{k} |\theta_{i+c-2}^{(k-2)} - \theta_j^{(k)}|$$

bekommen. Für $\delta < r < \gamma - \delta$ ist wieder $B = B_r(\theta_{i+c-1}^{(k)})$. Indem man nun wieder unterscheidet zwischen den Ritzwerten, die rechts vom Cluster liegen und denen, die links davon bzw. im Cluster liegen, erhält man erneut mit Lemma 3.6

$$\left|\frac{1}{2\pi i} \prod_{\nu=2}^{k} \beta_\nu^2 \int_{\partial B} \frac{dz}{\psi_{k-1}(z)\psi_k(z)}\right| < \frac{\delta^{2c-2}}{(r-\delta)^{2c-1}} \frac{1}{\left(1-\left(\frac{r+\delta}{\gamma}\right)\right)^{2k-2c}}.$$

Man sucht also wieder den Radius r, für den diese obere Schranke minimal wird und damit ist dann (3.24) bewiesen. □

In Satz 3.8 lassen sich die Rollen von ψ_{k-1} und ψ_k vertauschen, d.h. die Aussage hat auch Gültigkeit, falls sich der Cluster um einen Ritzwert auf $c-1$ im Schritt k verkleinert.

Bemerkung 3.9 *Liegt der Cluster von Ritzwerten am rechten oder linken Ende des Spektrums, so kann man den Faktor $\frac{1}{\left(1-\frac{2\delta}{\gamma}\right)}$ in (3.18) und (3.24) durch $\frac{1}{\left(1-\frac{\delta}{\gamma}\right)}$ ersetzen. Das stellt allerdings nur eine geringfügige Verbesserung dar, da wir grundsätzlich den Fall $\delta \ll \gamma$ betrachten.*

Wir haben gezeigt, dass sich die Gewichte zu einem Cluster von Ritzwerten stabilisieren. Außerdem wissen wir aus dem vorherigen Kapitel, dass sich ein Cluster von Ritzwerten nur in der Nähe eines Eigenwertes bilden kann. Die Ergebnisse diese Kaptitels legen nun eine Heuristik für folgendes Problem[1] nahe:

Fragestellung 3.10 *Man betrachtet die Eigenwertgleichung*

$$Au = \lambda u, \quad u^*u = 1.$$

Läßt sich - und wenn ja, wie - das Skalarprodukt

$$y^*u, \quad y^*y = 1,$$

abschätzen? Man wende den Lanczosalgorithmus mit Startvektor $q_1 = y$ an, denn $(y^*u)^2$ ist das Gewicht zum Eigenwert λ. Wir wissen, dass sich die Gewichte stabilisieren und man daher den Wert $(y^*u)^2$ numerisch kennt.

Ausblick
Einleitend zu diesem Kapitel haben wir die Parallelität von *cg*-Verfahren und Lanczosalgorithmus erwähnt und aufgezeigt. In [40] wird gezeigt, dass man die Fehler der *cg*-Iterierten x_k in der Energienorm, also $||\tilde{x} - x_k||_A$, über die Gaußquadraturapproximationen an das Integral (3.6) messen kann, wobei $f(\lambda) = \frac{1}{\lambda}$ ist, $0 < \zeta \leq \lambda$:

$$||\tilde{x} - x_0||_A^2 = ||r_0||^2 \int_\zeta^\xi \frac{1}{\lambda}\, dm^{(k)}(\lambda) \; + ||\tilde{x} - x_k||_A^2$$

Die mögliche Konvergenzverzögerung in endlicher Arithmetik erklärt sich daraus, dass sowohl *cg*-Verfahren als auch Lanczosalgorithmus eben nicht Gaußquadraturapproximationen an

$$\int_\zeta^\xi \frac{1}{\lambda}\, dm(\lambda)$$

berechnen sondern an ein Integral

$$\int_\zeta^\xi \frac{1}{\lambda}\, dm_1(\lambda).$$

Dabei ergibt sich die Gewichtsfunktion $m_1(\lambda)$ aus $m(\lambda)$, indem die einfachen Stützstellen λ_j - die Eigenwerte von A - ersetzt werden, z.B. durch einen (endlichen) Cluster in einem kleinen Intervall um λ_j.

[1] Von G. Golub an Z. Strakoš; wurde mir am 11. Dezember 2003 in Prag mitgeteilt.

Als nächstes gilt es die folgende Fragestellung zu untersuchen: Hat man zwei (eventuell sogar differenzierbare) Gewichtsfunktionen $m_1(\lambda)$ und $m_2(\lambda)$, die sich in einem noch näher zu definierendem Sinn nur gering voneinander unterscheiden, ist es dann richtig, dass für $k \in \mathbb{N}$

$$\text{Gaußquadratur}_k \left(\int_{\zeta}^{\xi} \frac{1}{\lambda} dm_1(\lambda) \right) \approx \text{Gaußquadratur}_k \left(\int_{\zeta}^{\xi} \frac{1}{\lambda} dm_2(\lambda) \right)$$

gilt?

Kapitel 4

Unitärer Lanczos-ähnlicher Algorithmus

Der symmetrische Lanczosalgorithmus orthogonalisiert (in exakter Arithmetik) sukzessive die Basen von Krylowunterräumen. Ist nun die Matrix, deren Eigenwerte berechnet werden sollen, nicht symmetrisch, so liefert beispielsweise das Arnoldiverfahren (siehe z.B. [34] S. 172 ff) zwar eine orthogonale Basis eines Krylowunterraumes, allerdings ist der Rayleighquotient der Matrix dann keine tridiagonale Matrix sondern eine (obere) Hessenbergmatrix. Zwar erzeugt auch der zweiseitige Lanczosalgorithmus eine Tridiagonalmatrix, doch geht dieser Algorithmus nicht auf die Struktur des Eigenwertproblems ein und leidet zudem unter dem Problem des (möglichen) *breakdowns* (siehe z.B. [30]).

Durch Ausnutzung der Eigenschaft *unitär* gelingt es Elsner und Bunse-Gerstner in [5] ein Verfahren zu entwickeln, das eine unitäre Matrix auf eine blocktridiagonale und pentadiagonale Matrix transformiert, den unitären Lanczos-ähnlichen Algorithmus. Wird dabei die zugrunde liegenden Ähnlichkeitstransformation berechnet, indem man den zugehörigen verallgemeinerten Krylowunterraum (siehe Definition 4.11) mit dem Verfahren nach Gram-Schmidt orthogonalisiert, so zeigt dieser Algorithmus numerisch ein ähnliches Verhalten wie der symmetrische Lanczosalgorithmus.

Auch aufgrund der engen Beziehungen zwischen unitären und hermiteschen Matrizen über die Cayleytransformation stellte sich die Frage, ob die im vorigen Abschnitt benutzten Ideen auch auf den Lanczos-ähnlichen Algorithmus für unitäre Matrizen übertragen werden können: Lassen sich die Residuen allein aus Informationen aus den Ritzwerten abschätzen und gibt es einen ähnlichen Zusammenhang zwischen Orthogonalitätsverlust und Konvergenz von Ritzwerten?

4.1 Unitäre Reduktion auf ein Schurparameter Pencil

Seien $A, Q_0 \in \mathbb{C}^{N\times N}$ zwei unitäre Matrizen, so dass $H = Q_0^* A Q_0$ eine obere Hessenbergmatrix ist, deren Einträge auf der ersten Nebendiagonalen reell und nichtnegativ sind. Dann ist H das Produkt von N Givensmatrizen (siehe z.B. [5]):

$$H = G_1(c_1)G_2(c_2)\dots G_N(c_N). \tag{4.1}$$

Dabei ist $G_j(c_j) \in \mathbb{C}^{N\times N}$ für $j = 1, \dots, N-1$ durch

$$G_j(c_j) = \begin{bmatrix} I_{j-1} & & & \\ & -c_j & s_j & \\ & s_j & \bar{c}_j & \\ & & & I_{N-j-1} \end{bmatrix}, \quad \text{mit} \quad \begin{matrix} c_j \in \mathbb{C},\ s_j \geq 0 \\ |c_j|^2 + s_j^2 = 1 \end{matrix} \tag{4.2}$$

und für $j = N$ durch

$$G_N = \begin{bmatrix} I_{N-1} & \\ & c_N \end{bmatrix}, \quad \text{mit} \quad |c_N| = 1. \tag{4.3}$$

definiert.

Definition 4.1 ([5]) *Die Parameter $c_1, \dots, c_N$ der Darstellung (4.1) - (4.3) heißen normierte Schurparameter von H, und die Zahlen $s_1, \dots, s_{N-1}$ werden normierte komplementäre Schurparameter von H genannt.*

Aufgrund der Bedingung $s_j \geq 0$ in (4.2) findet das Adjektiv *normiert* Eingang in diese Definition (siehe [5]).

Ebenfalls in der Arbeit [5] wird nun so vorgegangen, dass die elementaren unitären Matrizen der Darstellung (4.1) entsprechend der geraden und ungeraden Indices sortiert werden, d.h. in das Produkt der *ungeraden* Givensmatrizen

$$G_o^{(N)} = \prod_{j=1}^{\left[\frac{N+1}{2}\right]} G_{2j-1}(c_{2j-1}) \tag{4.4}$$

und in das Produkt der *geraden* Givensmatrizen

$$G_e^{(N)} = \prod_{j=1}^{\left[\frac{N}{2}\right]} G_{2j}(c_{2j}). \tag{4.5}$$

Das Pencil

$$G_o^{(N)} - \lambda \left(G_e^{(N)}\right)^*, \quad \lambda \in \mathbb{C},$$

heißt Schurparameter-Pencil und ist tridiagonal, denn $G_o^{(N)}$ und $G_e^{(N)}$ haben die folgende Form:

$$G_o^{(N)} = \begin{bmatrix} -c_1 & s_1 & & & & \\ s_1 & \bar{c}_1 & & & & \\ & & -c_3 & s_3 & & \\ & & s_3 & \bar{c}_3 & & \\ & & & & \ddots & \end{bmatrix}, \quad G_e^{(N)} = \begin{bmatrix} 1 & & & & & \\ & -c_2 & s_2 & & & \\ & s_2 & \bar{c}_2 & & & \\ & & & -c_4 & s_4 & \\ & & & s_4 & \bar{c}_4 & \\ & & & & & \ddots \end{bmatrix}$$

Nun ist eine obere Hessenbergmatrix der Form (4.1) unitär ähnlich zu (vgl. [1])

$$T = G_o^{(N)} G_e^{(N)}, \tag{4.6}$$

und das heißt insbesondere, dass H und T die gleichen Eigenwerte besitzen. Die Transformation einer unitären Matrix A auf ein Schurparameter Pencil $G_o - \lambda G_e^*$ und damit auf (4.6) ist außerdem im Wesentlichen eindeutig.

Lemma 4.2 ([5]) *Zu $A \in \mathbb{C}^{N\times N}$ unitär gibt es eine unitäre Matrix $Q \in \mathbb{C}^{N\times N}$, so dass*

$$Q^* A Q = G_o^{(N)} G_e^{(N)}$$

gilt. Sind alle komplementären Schurparameter positiv, so ist die Transformation bei gegebener erster Spalte q_1 von Q eindeutig.

4.2 Der unitäre Lanczos-ähnliche Algorithmus

Wir wiederholen kurz die Herleitung des Verfahrens aus [5] und [4] (dort S.62). Wie Lemma 4.2 schon andeutet, verläuft die Herleitung des unitären Lanczos-ähnlichen Verfahrens analog zum symmetrischen Fall. Angenommen, wir haben eine unitäre Matrix $Q \in \mathbb{C}^{N\times N}$, so dass

$$Q^* A Q = G_o^{(N)} G_e^{(N)} \tag{4.7}$$

gilt. Es ist (4.7) äquivalent zu

$$AQ \left(G_e^{(N)}\right) * = Q G_o^{(N)} \tag{4.8}$$

sowie

$$A^* Q G_o^{(N)} = Q \left(G_e^{(N)}\right)^*. \tag{4.9}$$

Für k ungerade erhalten wir die $k$te Spalte von (4.8) zu

$$A(s_{k-1} q_{k-1} + c_{k-1} q_k) = -c_k q_k + s_k q_{k+1},$$

wobei $q_j = Qe_j$ ist. Daraus läßt sich aufgrund der Orthogonalität der Spalten von Q c_k wie folgt bestimmen

$$c_k = -q_k^* A(s_{k-1}q_{k-1} + c_{k-1}q_k),$$

und man bekommt s_k zu

$$w_{k+1} = A(s_{k-1}q_{k-1} + c_{k-1}q_k) + c_k q_k, \quad s_k = ||w_{k+1}||.$$

Ist dann $s_k \neq 0$, so erhält man den neuen Lanczosvektor schließlich aus der Normierung von w_{k+1}. Ist k gerade, so gilt für die $k$te Spalte von (4.9)

$$A^*(s_{k-1}q_{k-1} + \bar{c}_{k-1}q_k) = -\bar{c}_k q_k + s_k q_{k+1}$$

und es berechnen sich c_k, s_k und q_{k+1} ganz analog. Zusammengefaßt erhalten wir

Algorithmus 4.3 (Unitärer Lanczos-ähnlicher Algorithmus, [5]) *Es seien eine unitäre Matrix $A \in \mathbb{C}^{N\times N}$ und ein Vektor $q_1 \in \mathbb{C}^N$ mit $q_1^* q_1 = 1$ gegeben. Wir setzen $s_0 = 0$, $c_0 = 1$ und $q_0 = 0 \in \mathbb{C}^N$ und berechnen für $k = 1, 2, \ldots$*

Ist k ungerade:

$$\begin{aligned} z_k &= A(s_{k-1}q_{k-1} + c_{k-1}q_k) \\ c_k &= -(q_k^* z_k) \\ w_{k+1} &= z_k + c_k q_k \\ s_k &= ||w_{k+1}|| \\ q_{k+1} &= \frac{w_{k+1}}{s_k} \end{aligned}$$

Ist k gerade:

$$\begin{aligned} z_k &= A^*(s_{k-1}q_{k-1} + \bar{c}_{k-1}q_k) \\ c_k &= -\overline{(q_k^* z_k)} \\ w_{k+1} &= z_k + \bar{c}_k q_k \\ s_k &= ||w_{k+1}|| \\ q_{k+1} &= \frac{w_{k+1}}{s_k} \end{aligned}$$

Sind dann k Schritte von Algorithmus 4.3 durchgeführt worden, so definieren wir

$$T_k = G_o^{(k)} G_e^{(k)},$$

dabei sind $G_o^{(k)}$ und $G_e^{(k)}$ die in (4.4) und (4.5) definierten Matrizen, mit dem einzigen Unterschied, dass die $k$te Matrix nicht mehr zwangsläufig unitär ist, $G_k(c_k) = \begin{bmatrix} I_{k-1} & \\ & -c_k \end{bmatrix}$, falls der Betrag von c_k kleiner als 1 ist. Analog

zum reell symmetrischen Lanczos Algorithmus definieren wir die Matrix der Lanczosvektoren

$$Q_k = [q_1, q_2, \ldots, q_k] \in \mathbb{C}^{N \times k},$$

also ist T_k der Rayleighquotient der Matrix A mit Q_k:

$$T_k = Q_k^* A Q_k.$$

4.3 Eigenvektor- Eigenwert Struktur von T_k

Zunächst einmal haben die von Algorithmus 4.3 erzeugten Matrizen T_k ein bestimmtes Nullenmuster, das von Bohnhorst in [4] *Doppeltreppe* genannt wurde.

Definition 4.4 ([4] S. 30) *Eine Matrix $T = [t_{l,j}]_{1 \le l,j \le n}$ mit der Eigenschaft*

$$t_{l,j} = 0, \quad \textit{falls} \begin{cases} l \textit{ gerade und} & \begin{cases} j \ge l+2 & \textit{oder} \\ j \le l-3 & , \end{cases} \\ l \textit{ ungerade und} & \begin{cases} j \ge l+3 & \textit{oder} \\ j \le l-2 & \end{cases} \end{cases}$$

heißt Doppeltreppen-Matrix. Gilt zusätzlich

$$\begin{array}{lll} t_{2,1} > 0 & & \\ t_{l,l+2} > 0 & 1 \le l \le n-2 & \textit{und } l \textit{ ungerade} \\ t_{l+2,l} > 0 & 2 \le l \le n-2 & \textit{und } l \textit{ gerade,} \end{array}$$

so nennen wir T kurz DT-Matrix.

Das Nullenmuster stellen wir für eine 8×8 Matrix dar:

$$\begin{bmatrix} x & x & + & 0 & 0 & 0 & 0 & 0 \\ + & x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & x & x & x & + & 0 & 0 & 0 \\ 0 & + & x & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & x & + & 0 \\ 0 & 0 & 0 & + & x & x & x & 0 \\ 0 & 0 & 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & 0 & + & x & x \end{bmatrix}$$

Das x und das $+$ stehen für mögliche Einträge der Matrix, die ungleich Null sind. Bei einer DT-Matrix wird verlangt, dass sich an den durch $+$ gekennzeichneten Stellen positive Zahlen befinden; das bedeutet in Zusammenhang mit Algorithmus 4.3, dass die normierten komplementären Schurparameter s_j positiv sind und der unitäre Lanczos-ähnliche Algorithmus nicht zusammenbricht. Mit anderen Worten, Algorithmus 4.3 erzeugt eine Folge T_k von

DT-Matrizen.

Werden die Rechnungen des unitären Lanczos-ähnlichen Algorithmus ohne Rundungsfehler durchgeführt, so bricht das Verfahren für ein $d \leq N$ mit $s_d = 0$ ab. Der dann erzeugte Schurparameter Pencil $G_o^{(d)}$, $G_e^{(d)}$ liefert folglich eine unitäre DT-Matrix. Für jedes $\lambda \in \mathbb{C}$ gilt dann

$$\operatorname{rang}(\lambda I - T_d) = \operatorname{rang}\left(\lambda \left(G_o^{(d)}\right)^* - G_e^{(d)}\right) \geq d - 1, \tag{4.10}$$

da $\lambda \left(G_o^{(d)}\right)^* - G_e^{(d)}$ für $\lambda \neq 0$ eine tridiagonale Matrix ist, bei der alle Nebendiagonalelemente ungleich Null sind. Da T_d außerdem unitär ist, sind alle Eigenwerte von T_d einfach.

Allerdings sind für $k < d$ die Hauptabschnittsmatrizen T_k von T_d nicht unitär, denn der $k$te normierte Schurparameter hat einen Betrag kleiner als 1, $|c_k| < 1$. Trotzdem ist auch T_k eine DT-Matrix. Sei

$$S_k = \operatorname{diag}(I_{k-1}, -c_k)$$

dann gilt

$$T_k = \begin{cases} G_o^{(k)} \tilde{G}_e^{(k)} S_k & \text{, falls k gerade ist} \\ S_k \tilde{G}_o^{(k)} G_e^{(k)} & \text{, falls k ungerade ist} \end{cases}$$

mit $e_k^* \tilde{G}_e^{(k)} e_k = 1$ und $e_k^* \tilde{G}_o^{(k)} e_k = 1$ und ansonsten stimmen diese Matrizen mit $G_e^{(k)}$ bzw. $G_o^{(k)}$ überein. Wie in (4.10) sieht man, dass der Rang von $(\lambda I - T_k)$ auch größer oder gleich $k-1$ ist, und T_k deshalb auch geometrisch einfache Eigenwerte hat. Es stellt sich aber die Frage, ob die algebraische Vielfachheit der Eigenwerte ebenfalls gleich eins ist?

Vermutung 4.5 *Ist T_d eine unitäre DT-Matrix, so dass für die komplementären Schurparameter $0 < s_j < 1$ gilt, $1 \leq j < d$, dann haben die führenden Hauptabschnitte T_k von T_d algebraisch einfache Eigenwerte, $1 \leq k \leq d$.*

Wir diskutieren diese Fragestellung kurz. Es sei k gerade und wir setzen

$$T_+ = T_k, \quad T = G_o^{(k)} \tilde{G}_e^{(k)}.$$

Das bedeutet, dass T_+ eine unitäre Matrix plus einer Rang-1 Matrix ist. Mit $S = S_k$ und $c = c_k \neq 0$ gilt also

$$T_+ = TS = T + T e_k e_k^* (c - 1).$$

Da

$$\det(T_+ - \lambda I) = 0 \Leftrightarrow \det\left(T - \lambda S^{-1}\right) = 0$$

gilt, läßt sich die Vielfachheit der Eigenwerte von T_+ eventuell über das allgemeine Eigenwertproblem $T - \lambda S^{-1}$ untersuchen. Wir wenden dazu die verallgemeinerte Schurzerlegung (siehe [15] S.396) an. Es sei

$$W^*(TS)W = R_1$$

die Schur'sche Normalform von T_+ und

$$SW = ZR_2 \tag{4.11}$$

die QR-Zerlegung von SW, so dass R_2 positive Diagonaleinträge hat. Da außerdem W, T und Z unitäre Matrizen sind, gilt zusätzlich noch

$$W^*TZ = R_1R_2^{-1} = D = \operatorname{diag}\left(e^{i\beta_1}, \ldots, e^{i\beta_k}\right). \tag{4.12}$$

Wir bezeichnen die Diagonalelemente von R_2 als $r_j = e_j^* R_2 e_j$. T_+ hätte dann einen degenerierten Eigenwert, falls es mindestens zwei Indices $1 \leq l < j \leq k$ gibt, für die

$$\beta_l = \beta_j \quad \text{und} \quad r_l = r_j$$

gilt. Um zu zeigen, dass T_+ einfache Eigenwerte hat, genügt es also, zu demonstrieren, dass entweder $\beta_l \neq \beta_j$ oder $r_k \neq r_j$ für alle $l \neq j$ gilt. Das Nullenmuster von T_+ liefert uns das folgende

Lemma 4.6 *Es sei T_d eine unitäre DT-Matrix, und T_k eine führende Hauptabschnittsmatrix $1 \leq k \leq d$. Ist x ein Rechtseigenvektor und y ein Linkseigenvektor von T_k zu einem Eigenwert $\theta \neq 0$, so gilt*

$$0 \neq \begin{cases} e_k^* x & , \textit{falls } k \textit{ gerade} \\ e_k^* y & , \textit{falls } k \textit{ ungerade} \end{cases}$$

Beweis: Wie nehmen an, k sei gerade und setzen $x_j = e_j^* x$, $1 \leq j \leq k$. Sei θ ein Eigenwert von T_k. Dann gilt:

$$\tilde{G}_e^{(k)} S_k x = \theta \left(G_o^{(k)}\right)^* x. \tag{4.13}$$

Die $l$te Zeile von (4.13) ist

$$\begin{cases} \theta\left(-\bar{c}_1 x_1 + s_1 x_2\right) &=& x_1 & , l = 1 \\ \theta\left(s_{l-1}x_{l-1} + c_{l-1}x_l\right) &=& -c_l x_l + s_l x_{l+1} & , 1 < l < k \quad \text{und l gerade} \\ \theta\left(-\bar{c}_l x_l + s_l x_{l+1}\right) &=& s_{l-1}x_{l-1} + \bar{c}_l x_l & , 1 < l < k \quad \text{und l ungerade} \\ \theta\left(s_{k-1}x_{k-1} + \bar{c}_{k-1}x_k\right) &=& -c_k x_k & , l = k \end{cases}$$

Wäre $x_k = 0$, so würde $x_k = x_{k-1} = \ldots = x_1 = 0$ folgen, da θ und alle komplementären Schurparameter ungleich Null sind. Jedoch ist $x = 0$ ein Widerspruch dazu, dass x Eigenvektor ist. Für k ungerade läuft der Beweis ganz analog ab. □

Satz 4.7 *Ist T_k führender Hauptabschnitt der unitären DT-Matrix T_d, $k < d$, so gilt:*

$$\Lambda(T_k) \subset \{ z : \ |c_k| \leq |z| < 1 \}.$$

Beweis: Es sei $c_k \neq 0$, dann ist auch $0 \notin \Lambda(T_k)$. Wir betrachten wieder den Fall, dass k eine gerade Zahl ist. Falls k ungerade ist, verläuft der Beweis identisch. Für $0 \neq \theta \in \Lambda(T_k)$ mit Eigenvektor x gilt:

$$|\theta| ||x|| = ||G_o^{(k)} \tilde{G}_e^{(k)} S_k x|| = ||S_k x|| = \sqrt{\sum_{j=1}^{k-1} |x_j|^2 + |c_k|^2 |x_k|^2} < ||x||, \quad (4.14)$$

denn nach Lemma 4.6 ist x_k ungleich Null, und es gilt $|c_k| < 1$. Die untere Abschätzung ist nur für $c_k \neq 0$ interessant. Nach (4.12) gilt für die Eigenwerte θ_j von T_k gerade $\theta_j = r_j e^{i\beta_j}$, d.h. dass die Beträge der Eigenwerte von T_k auf der Diagonalen von R_2 stehen. Aus

$$R_2^* R_2 = W^* S^* S W$$

folgt

$$\prod_{j=1}^{k} r_j^2 = (\det(R_2))^2 = \det(R_2 R_2^*) = |c_k|^2$$

und da wir mit (4.14) gezeigt haben, dass $r_j < 1$ ist, erhalten wir $r_j \geq |c_k|$ für $j = 1, \ldots, k$. □

Ist $c_k \neq 0$, so folgt aus (4.11)

$$R_2 - R_2^{-*} = Z^* \left(S - S^{-*} \right) W = \frac{|c|^2 - 1}{\bar{c}} Z^* e_k e_k^* W,$$

und mit Hilfe von $0 < |c| \leq r_j < 1$ erkennt man, dass die Diagonaleinträge von $R_2 - R_2^{-*}$ alle ungleich Null sind, so dass auch die letzten Zeilen der unitären Matrix Z und W keine Nullen enthalten: $e_k^* Z e_j \neq 0 \neq e_k^* W e_j$ für $j = 1, \ldots, k$.

Um zu zeigen, dass die Eigenwerte der führenden Hauptabschnitte T_k einfach sind, dachten wir zunächst, es sei möglich $r_j \neq r_l$ in der Zerlegung (4.11) und (4.12) nachzuweisen. Aber das ist nicht möglich, wie anhand des nachfolgenden Beispiels demonstriert wird.

Beispiel 4.8 *Angenommen, es seien alle komplementären Schurparameter gleich eins: $s_j = 1$ für $j = 1, \ldots, k-1$. Dann ist $T_k S_k^{-1}$ (falls k gerade ist und sonst $S_k^{-1} T_k$) eine Permutationsmatrix und es gilt:*

$$\det(TS - \lambda I) = \begin{matrix} + \\ - \end{matrix} \left\{ \begin{array}{l} \det\left(G_e^{(k)} S - \lambda G_o^{(k)} \right) \\ \det\left(S G_o^{(k)} - \lambda G_e^{(k)} \right) \end{array} \right\} = \begin{matrix} + \\ - \end{matrix} \left(c - \lambda^k \right).$$

Somit haben alle Eigenwerte von T_k den gleichen Betrag und für die Diagonale von R_2 in (4.11) gilt $r_1 = r_2 = \ldots = r_k = (|c_k|)^{1/k}$.

Als Möglichkeit bleibt also, zu zeigen, dass in (4.12) die Winkel alle verschieden sind, also $\beta_l \neq \beta_j$ für $j \neq l$ und $0 \leq \beta_l, \beta_j < 2\pi$. Allerdings ist uns dieses bisher nicht gelungen.

Als weitere Bezeichnungen benutzen wir wie im vorherigen Kapitel für die charakteristischen Polynome

$$\begin{aligned} \psi_k(\lambda) &= \det(\lambda I - T_k) \\ \psi_{2,k}(\lambda) &= \det(\lambda I - T_{2,k}). \end{aligned}$$

Lemma 4.9 *Es sei T_k ein Hauptabschnitt einer unitären DT-Matrix T_d, $k \leq d$, und habe einfache Eigenwerte. Wir bezeichnen mit X die Matrix der Rechtseigenvektoren und mit Y die Matrix der Linkseigenvektoren, $Y = X^{-*}$, d.h. $T_k = X\,diag(\theta_1, \ldots, \theta_k)Y^*$. Dann gilt für jedes Eigentripel (θ, x, y) von T_k:*

$$\begin{aligned} x_k\bar{y}_k\psi_k'(\theta) &= \psi_{k-1}(\theta) \\ x_1\bar{y}_1\psi_k'(\theta) &= \psi_{2,k}(\theta) \end{aligned}$$

Ist k gerade:

$$\begin{aligned} x_1\bar{y}_k\psi_k'(\theta) &= c_k\theta^{\frac{k-2}{2}}\prod_{j=1}^{k-1} s_j \\ x_1\bar{y}_{k-1}\psi_k'(\theta) &= (\theta + c_k\bar{c}_{k-1})\theta^{\frac{k-2}{2}}\prod_{j=1}^{k-2} s_j \end{aligned}$$

und falls k ungerade ist ($k \geq 3$):

$$\begin{aligned} x_k\bar{y}_1\psi_k'(\theta) &= -\theta^{\frac{k-3}{2}}c_k\prod_{j=1}^{k-1} s_j \\ x_{k-1}\bar{y}_1\psi_k'(\theta) &= -(\theta + \bar{c}_{k-1}c_k)\theta^{\frac{k-3}{2}}\prod_{j=1}^{k-2} s_j. \end{aligned}$$

Beweis: Die Formeln lassen sich, ähnlich wie in [26] auf Seite 129, aus einer Identität für die adjungierte Matrix herleiten:

$$\mathrm{adj}(\theta I - T_k) = \psi_k'(\theta)xy^*$$

Man vergleicht nun die entsprechenden Einträge der beiden Matrizen. □

Wäre hingegen θ ein algebraisch l-facher (aber geometrisch einfacher) Eigenwert, was wir aber für DT-Matrizen nicht beobachten konnten und daher nicht unterstellen (Vermutung 4.5), so würde stattdessen

$$\mathrm{adj}(\theta I - T_k) = \frac{1}{l!}\psi_k^{(l)}(\theta)xy^*$$

gelten, woraus problemlos analoge Resultate zu Lemma 4.9 gewonnen werden könnten.

Eine besonders schöne Eigenschaft der diversen Lanczosverfahren (siehe z.B.[3], [12]) ist die Berechenbarkeit des Residuums aus Größen des kleinen Eigenwertproblems für T_k. Möchte man das auch hier machen, muß man in Analogie zu (2.4) wissen, wie die ersten k Spalten der Reduktion (4.7) von A auf Doppeltreppenform aussehen. Es gilt

Satz 4.10 ([4] S. 72/73) *Nach $1 \leq k \leq d-1$ Schritten des Algorithmus 4.3 gilt, falls k gerade ist*

$$A^*Q_k = Q_k T_k^* + s_{k-1}s_k q_{k+1}e_{k-1}^* + c_{k-1}s_k q_{k+1}e_k^* \tag{4.15}$$

und falls k ungerade ist

$$AQ_k = Q_k T_k + s_{k-1}s_k q_{k+1}e_{k-1}^* + \bar{c}_{k-1}s_k q_{k+1}e_k^* \tag{4.16}$$

Wir betrachten nun für ungerades k das Ritzpaar $(\theta_j^{(k)}, z_j^{(k)})$, d.h. es gilt $T_k x_j^{(k)} = \theta_j^{(k)} x_j^{(k)}$ und $z_j^{(k)} = Q_k x_j^{(k)}$; damit folgt aus (4.16)

$$||Az_j^{(k)} - \theta_j^{(k)} z_j^{(k)}|| = |s_{k-1}s_k x_{k-1,j}^{(k)} + s_k \bar{c}_{k-1} x_{k,j}^{(k)}| =: \delta_{k,j} \tag{4.17}$$

und für gerades k liefert (4.15) zusammen mit einem Linkseigenvektor $T_k^* y_j^{(k)} = \bar{\theta}_j^{(k)} y_j^{(k)}$ sowie $\tilde{z}_j^{(k)} = Q_k y_j^{(k)}$:

$$||A^*\tilde{z}_j^{(k)} - \bar{\theta}_j^{(k)}\tilde{z}_j^{(k)}|| = |s_{k-1}s_k \bar{y}_{k-1,j}^{(k)} + s_k \bar{c}_{k-1} y_{k,j}^{(k)}| =: \tilde{\delta}_{k,j}. \tag{4.18}$$

Wie beim symmetrischen Lanczosverfahren kann man damit die Konvergenz von Ritzpaaren über die Größen $\delta_{k,j}$, $\tilde{\delta}_{k,j}$ kontrollieren.

Mit Lemma 4.9 versuchen wir nun ebenfalls in Analogie zum symmetrischen Lanczosalgorithmus, die Größen $\delta_{k,j}$ bzw. $\tilde{\delta}_{k,j}$ nur aus Informationen aus Ritzwerten $\theta_j^{(k)}$ zu bestimmen, so dass auch hier Eigenvektoren von T_k erst dann berechnet werden müssen, wenn die Konvergenz hinreichend gut ist. Ein Problem allerdings ist, dass T_k im allgemeinen weder unitär noch normal ist, dass also Links- und Rechtseigenvektoren verschieden sind. Daher bekommen wir auch nur

$$|y_{1,j}^{(k)}|\delta_{k,j} = \prod_{j=1}^{k} s_j \left| \frac{(\theta_j^{(k)})^{(k-3)/2}(\theta_j^{(k)} + 2\bar{c}_{k-1}c_k)}{\psi_k'(\theta_j^{(k)})} \right|, \quad \text{k ungerade} \tag{4.19}$$

$$|x_{1,j}^{(k)}|\tilde{\delta}_{k,j} = \prod_{j=1}^{k} s_j \left| \frac{(\theta_j^{(k)})^{(k-2)/2}(\theta_j^{(k)} + 2\hat{c}_{k-1}c_k)}{\psi_k'(\theta_j^{(k)})} \right|, \quad \text{k gerade.} \tag{4.20}$$

4.4 Ein numerisches Experiment

Wir wollen anhand eines Beispiels schauen, ob der Algorithmus 4.3 bezüglich Konvergenz von Ritzwerten, Clusterbildung und Orthogonalitätsverlust in endlicher Arithmetik ein ähnliches Verhalten an den Tag legt wie der symmetrische Lanczosalgorithmus. Dazu eine Vorüberlegung.

Definition 4.11 ([4] S. 63) *Zu $k \in \mathbb{N}$, $A \in \mathbb{C}^{N\times N}$ unitär und $v \in \mathbb{C}^N$ definieren wir den* verallgemeinerten Krylowraum *als*

$$\tilde{\mathcal{K}}_k(A,v) = \begin{cases} span\{v, Av, A^*v, A^2v, (A^*)^2v, \ldots, (A^*)^{\frac{k-1}{2}}v\}, & k \quad ungerade \\ span\{v, Av, A^*v, A^2v, (A^*)^2v, \ldots, A^{\frac{k}{2}}v\}, & k \quad gerade \end{cases}$$

Dann überzeugt man sich leicht davon, dass sich jeder Vektor w aus $\tilde{\mathcal{K}}_k(A,v)$ in der Form Polynom in A (bzw. in A^*) mal einem Vektor v darstellen läßt:

$$w = \begin{cases} p(A)\,(A^*)^{\frac{k-1}{2}}\,v = \tilde{p}\,(A^*)\,A^{\frac{k-1}{2}}v & , \; k \quad \text{ungerade} \\ q(A)\,(A^*)^{\frac{k}{2}-1}\,v = \tilde{q}\,(A^*)\,A^{\frac{k}{2}}v & , \; k \quad \text{gerade} \end{cases}$$

für gewisse Polynome $p, \tilde{p}, q, \tilde{q}$ vom Grad kleiner oder gleich $k-1$.

Als eine wesentliche Größe bei der Herleitung oberer a priori Schranken zur Abschätzung der Konvergenz der Ritzpaare betrachtet Saad (vgl. [34] S.204) den Abstand eines Eigenvektors u_i von A zu dem Unterraum, aus dem man die Ritzapproximationen gewinnt, hier also $\tilde{\mathcal{K}}_k(A,v)$. Sei $\mathcal{P}_k = Q_kQ_k^*$ die orthogonale Projektion auf $\tilde{\mathcal{K}}_k(A,v)$ und für den Startvektor q_1 gelte $q_1 = \sum_{j=1}^{N} \alpha_j u_j$ mit $\alpha_i \neq 0$ und $q_1^*q_1 = 1$. Zudem bezeichnen wir mit M das Spektrum von A ohne λ_i, $M = \Lambda(A) \setminus \{\lambda_i\}$. Es gilt:

$$\begin{aligned} ||(I-\mathcal{P}_k)\alpha_i u_i||^2 &= \min_{degp\leq k-1} ||\alpha_i u_i - p(A)(A^*)^{\frac{k-1}{2}}q_1||^2 \\ &= \min_{degp\leq k-1} ||\alpha_i A^{\frac{k-1}{2}}u_i - p(A)q_1||^2 \\ &\leq \min_{\substack{degp\leq k-1 \\ p(\lambda_i)=\lambda_i^{\frac{k-1}{2}}}} ||\alpha_i(\lambda_i^{\frac{k-1}{2}} - p(\lambda_i))u_i - \sum_{\substack{j=1 \\ j\neq i}}^{N} \alpha_j p(\lambda_j)u_j||^2 \\ &= \min_{\substack{degp\leq k-1 \\ p(\lambda_i)=\lambda_i^{\frac{k-1}{2}}}} \sum_{\substack{j=1 \\ j\neq i}}^{N} |\alpha_j|^2|p(\lambda_j)|^2 \\ &< \sum_{\substack{j=1 \\ j\neq i}}^{N} |\alpha_j|^2 \min_{\substack{degp\leq k-1 \\ p(\lambda_i)=\lambda_i^{\frac{k-1}{2}}}} \max_{z\in M} |p(z)|^2. \end{aligned}$$

Also haben wir das Approximationsproblem

$$\min_{\substack{p(\lambda_i)=\lambda_i^{\frac{k-1}{2}} \\ degp\leq k-1}} \max_{z\in M} |p(z)|.$$

zu untersuchen. Zunächst einmal gilt

$$\min_{\substack{p(\lambda_i)=\lambda_i^{\frac{k-1}{2}} \\ degp\leq k-1}} \max_{z\in M} |p(z)| = \min_{\substack{p(1)=1 \\ degp\leq k-1}} \max_{z\in \lambda_i M} |p(z)|. \tag{4.21}$$

Das bedeutet, wir können $\lambda_i = 1 \notin M$ annehmen. Die beiden benachbarten Eigenwerte von λ_i seien $e^{i\beta_-} = \lambda_{i-1} \neq \lambda_i \neq \lambda_{i+1} = e^{i\beta_+}$. Wir definieren $\phi = 2\pi - (\beta_+ - \beta_-)$, $a = e^{i\left(\pi + \frac{\beta_1+\beta_{N-1}}{2}\right)}$ und $\gamma = \left(\cos\left(\frac{\phi}{4}\right)\right)^{-1}$. Die Vorgehensweise ist nun wie folgt. Man löst nicht das Approximationsproblem (4.21) über der diskreten Menge M, sondern betrachtet das Problem auf einer kontinuierlichen Obermenge, in diesem Fall einem Kreisbogen $\Omega_0 \supset M$. Die folgenden Ergebnisse stammen aus [21], S.57, finden sich auch in [22].

$$\Omega_0 = \{ae^{i\beta} \ : \ \frac{\phi}{2} \leq \beta \leq 2\pi - \frac{\phi}{2}\}$$

Wir bezeichnen mit $C_n(z)$ das n-te Tschebyscheffpolynom für $z \in \mathbb{C}$ (siehe z.B. [34] S.144). Für $n \geq k-1$ stellt sich dann das $n$te Faberpolynom für Ω_0 dar als (siehe [21], Theorem 4.1, S.30):

$$F_n(z) = 2(az)^{n/2} C_n\left(\gamma \frac{a^{1/2}z^{-1/2} + a^{-1/2}z^{1/2}}{2}\right) - \left(\frac{a}{\gamma}\right)^n, \quad z \neq 0$$

sowie

$$F_n(0) = a^n(\gamma^n - \gamma^{-n}),$$

und das schätzt man zu

$$|F_n(z)| \leq 2 + \gamma^{-n}, \quad \text{für alle} \quad z \in \Omega_0$$

und

$$|F_n(1)| \geq 2\left|C_n\left(\gamma \frac{a^{-1/2} + a^{1/2}}{2}\right)\right| - \gamma^{-n}.$$

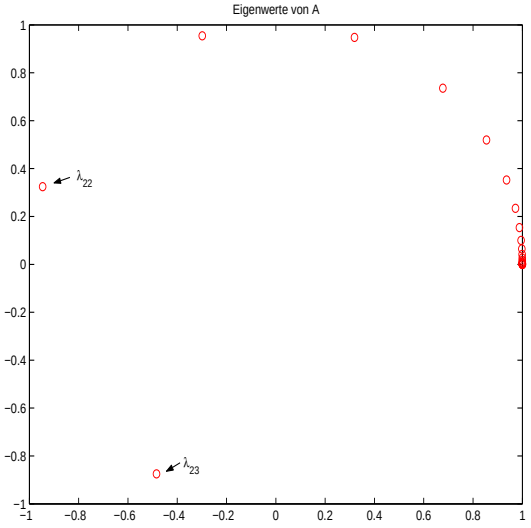
ab. Damit bekommen wir insgesamt

$$||(I - \mathcal{P}_k)u_i|| \leq \sqrt{\frac{1}{|\alpha_i|^2} - 1}\, \frac{2 + \gamma^{-k+1}}{2\left|C_{k-1}\left(\gamma \frac{a^{-1/2}+a^{1/2}}{2}\right)\right| - \gamma^{-k+1}}. \tag{4.22}$$

Die Werte von $|C_{k-1}(x)|$ werden für $|x| > 1$ schnell groß (vgl. [34] S.143 oder[26] S.331), somit besagt (4.22), dass die Ritzwerte um so schneller gegen einen Eigenwert λ_i von A konvergieren, je größer die Lücke ϕ auf dem

Einheitskreis zwischen den beiden benachbarten Eigenwerten von λ_i ist. Ein nicht überraschendes Ergebnis, denn auch im symmetrischen Lanczosalgorithmus konvergieren die Ritzwerte um so schneller gegen einen Eigenwert je größer die Spektrallücke ist. Dementsprechend konstruieren wir ein Beispiel mit hinreichend großer Lücke auf dem Einheitskreis.

Beispiel 4.12 *Die Zahlen $b_1, \ldots, b_{24}$ mögen die Verteilung (2.17) mit $N = 24$ und $\rho = 0.7$ haben und es sei $\mu_j = \frac{2\pi}{b_{24}-b_1}(b_j - b_1)$, $\lambda_j = e^{i\mu_j}$ und $A = diag(\lambda_1, \ldots, \lambda_{24})$.*



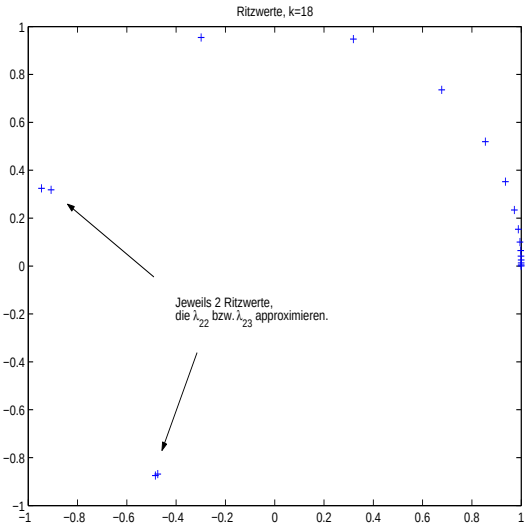
Wir wenden nun den Algorithmus 4.3 mit Startvektor $q_1 = e$ an und beobachten:

- Der Algorithmus bricht in endlicher Arithmetik nicht nach $N = 24$ Schritten mit $s_N = 0$ ab.

- Die Orthogonalität der Lanczosvektoren geht noch vor $k = N$ verloren. Kondition der Matrix Q_k:

k	17	18	19	20
$\text{cond}_2(Q_k)$	41.914	1.8225×10^4	1.3365×10^7	1.7086×10^{10}

- Es bilden sich Cluster von Ritzwerten. Zum Beispiel approximieren im Schritt $k = 18$ jeweils zwei Ritzwerte die Eigenwerte λ_{22} und λ_{23}:



4.5 Rundungsfehleranalyse

In der Arbeit [5] wird eine Stabilitätsanalyse der Reduktion einer unitären Matrix A auf Schurparameterform G_oG_e durchgeführt. Es wird gezeigt, dass die Reduktion rückwärts stabil ist, wenn die Transformation mit Householdermatrizen implementiert wird. Allerdings wird nicht Algorithmus 4.3 analysiert, der eine Gram-Schmidt Orthogonalisierung der Basen der verallgemeinerten Krylowräume $\tilde{\mathcal{K}}_k(A, q_1)$ darstellt. Der nun folgende erste Satz dieses Abschnitts besagt, dass die durch den unitären Lanczos ähnlichen Algorithmus in endlicher Arithmetik berechneten Größen spaltenweise die Gleichungen (4.8) und (4.9) wiedergeben plus einem Fehler, der von der Ordnung der Maschinengenauigkeit ϵ ist. Die maximale Anzahl an Nichtnullelementen in den Zeilen bzw. Spalten von A wird erneut mit $m \leq N$ bezeichnet. Ergebnisse von Gleitkommaoperationen werden hier mit einem $\hat{}$ gekennzeichnet.

Satz 4.13 *Sei* $\epsilon_{33} = \left((m+3)\sqrt{2}+3\right)\epsilon$ *und* $\epsilon_{22} = \left(2+2\sqrt{2}\right)\epsilon$. *Wurden* j *Schritte des unitären Lanczos ähnlichen Algorithmus in endlicher (oder exakter) Arithmetik durchgeführt, ohne dass der Algorithmus zusammenbricht (d.h.* $s_l \neq 0$ *für* $l = 1, \ldots, j$*), dann gilt für die berechneten Größen*

$$A\hat{Q}_j(\hat{G}_e^{(j)})^* e_j = \hat{Q}_j\hat{G}_o^{(j)}e_j + \hat{s}_j\hat{q}_{j+1} + \tilde{f}_j$$

$$\left|\tilde{f}_j\right| \leq |A||\hat{Q}_j||(\hat{G}_e^{(j)})^*|e_j\epsilon_{33} + |\hat{Q}_j||\hat{G}_o^{(j)}|e_j\epsilon_{22} + \mathcal{O}\left(\epsilon^2\right),$$

falls j *eine gerade Zahl ist und andernfalls*

$$A^*\hat{Q}_j\hat{G}_o^{(j)}e_j = \hat{Q}_j(\hat{G}_e^{(j)})^* e_j + \hat{s}_j\hat{q}_{j+1} + \tilde{f}_j$$

$$|\tilde{f}_j| \leq |A^*||\hat{Q}_j||\hat{G}_o^{(j)}|e_j\epsilon_{33} + |\hat{Q}_j||(\hat{G}_e^{(j)})^*|e_j\epsilon_{22} + \mathcal{O}\left(\epsilon^2\right).$$

Beweis: Wir führen die Rechnungen für den Fall, dass $j \in \mathbb{N}$ eine gerade Zahl ist, durch. Die erste Operation des Algorithmus ist dann die Multiplikation einer reellen Zahl mit einem komplexen Vektor[1]

$$\hat{p}_1 = fl\left(\hat{s}_{j-1}\hat{q}_{j-1}\right) = \hat{s}_{j-1}\hat{q}_{j-1} + \delta p_1, \quad |\delta p_1| \leq \hat{s}_{j-1}|\hat{q}_{j-1}|\epsilon,$$

es folgt eine Saxpy Operation

$$\begin{aligned} \hat{p}_2 &= fl\left(\hat{p}_1 + \hat{\bar{c}}_{j-1}\hat{q}_j\right) = \hat{p}_1 + \hat{\bar{c}}_{j-1}\hat{q}_j + \delta\tilde{p}_2 \\ &= \hat{s}_{j-1}\hat{q}_{j-1} + \hat{\bar{c}}_{j-1}\hat{q}_j + \delta p_2, \end{aligned}$$

mit

$$\begin{aligned} |\delta\tilde{p}_2| &\leq \left((1+2\sqrt{2})|\hat{\bar{c}}_{j-1}||\hat{q}_j| + |\hat{p}_1|\right)\epsilon + \mathcal{O}(\epsilon^2) \\ |\delta p_2| &\leq |\delta p_1| + |\delta\tilde{p}_2| \\ &\leq 2\hat{s}_{j-1}|\hat{q}_{j-1}|\epsilon + (1+2\sqrt{2})|\hat{\bar{c}}_{j-1}||\hat{q}_j|\epsilon + \mathcal{O}(\epsilon^2). \end{aligned}$$

Anschließend wird der Vektor $\hat{z}_j$ durch eine Matrix-Vektor Multiplikation bestimmt

$$\hat{z}_j = fl\left(A^*\hat{p}_2\right) = A^*\hat{p}_2 + \delta\tilde{z}_j, \quad |\delta\tilde{z}_j| \leq (m+1)\sqrt{2}|A^*||\hat{p}_2|\epsilon + \mathcal{O}(\epsilon^2),$$

so dass wir

$$\hat{z}_j = A^*\left(\hat{s}_{j-1}\hat{q}_{j-1} + \hat{\bar{c}}_{j-1}\hat{q}_j\right) + \delta z_j$$

mit

$$\begin{aligned} |\delta z_j| &\leq |\delta\tilde{z}_j| + |A^*||\delta p_2| \\ &\leq |A^*|\left(2\hat{s}_{j-1}|\hat{q}_{j-1}| + (1+2\sqrt{2})|\hat{\bar{c}}_{j-1}||\hat{q}_j|\right)\epsilon \\ &\quad +\sqrt{2}(m+1)|A^*||\hat{p}_2|\epsilon + \mathcal{O}(\epsilon^2) \\ &\leq |A^*|\left(2\hat{s}_{j-1}|\hat{q}_{j-1}| + (1+2\sqrt{2})|\hat{\bar{c}}_{j-1}||\hat{q}_j|\right)\epsilon \\ &\quad +\sqrt{2}(m+1)|A^*|\left(\hat{s}_{j-1}|\hat{q}_{j-1}| + |\hat{\bar{c}}_{j-1}||\hat{q}_j|\right)\epsilon + \mathcal{O}(\epsilon^2) \\ &\leq \left((m+3)\sqrt{2}+1\right)|A^*|\left(\hat{s}_{j-1}|\hat{q}_{j-1}| + |\hat{\bar{c}}_{j-1}||\hat{q}_j|\right)\epsilon + \mathcal{O}(\epsilon^2) \end{aligned}$$

haben. Der nächste berechnete Schurparameter ist $\hat{c}_j$ und folgt aus einem Skalarprodukt

$$\hat{c}_j = -\overline{fl\left(\hat{q}_j^*\hat{z}_j\right)} = -\overline{\hat{q}_j^*\hat{z}_j} + \delta c_j, \quad |\delta c_j| \leq \sqrt{2}\gamma_{N+1}(\epsilon)|\hat{q}_j^*||\hat{z}_j|.$$

Eine weiter Saxpy Operation liefert

$$\hat{w}_{j+1} = fl\left(\hat{z}_j + \hat{\bar{c}}_j\hat{q}_j\right) = \hat{z}_j + \hat{\bar{c}}_j\hat{q}_j + \delta\tilde{w}_{j+1},$$

[1] siehe Anhang A

wo

$$|\delta\tilde{w}_{j+1}| \leq (1+2\sqrt{2})\left(|\hat{\bar{c}}_j||\hat{q}_j| + |\hat{z}_j|\right)\epsilon + \mathcal{O}(\epsilon^2).$$

Also gilt

$$\hat{w}_{j+1} = A^*\left(\hat{s}_{j-1}\hat{q}_{j-1} + \hat{\bar{c}}_{j-1}\hat{q}_j\right) + \hat{\bar{c}}_j\hat{q}_j + \delta w_{j+1}$$

mit

$$\begin{aligned}
|\delta w_{j+1}| &\leq |\delta\tilde{w}_{j+1}| + |\delta z_j| \\
&\leq \left((m+3)\sqrt{2}+1\right)|A^*|\left(\hat{s}_{j-1}|\hat{q}_{j-1}| + |\hat{\bar{c}}_{j-1}||\hat{q}_j|\right)\epsilon \\
&\quad +(1+2\sqrt{2})|\hat{\bar{c}}_j||\hat{q}_j|\epsilon + \mathcal{O}(\epsilon^2).
\end{aligned}$$

Für den berechneten komplementären Schurparameter $\hat{s}_j$ gilt

$$\begin{aligned}
\hat{b}_j &= fl\left(\hat{w}_{j+1}^*\hat{w}_{j+1}\right) = \hat{w}_{j+1}^*\hat{w}_{j+1}(1+\delta b_j), \quad |\delta b_j| \leq \sqrt{2}\gamma_{N+1}(\epsilon) \\
\hat{s}_j &= fl\left(\sqrt{\hat{b}_j}\right) = \sqrt{\hat{w}_{j+1}^*\hat{w}_{j+1}(1+\delta b_j)}(1+\delta s_j), \quad |\delta s_j| \leq \epsilon.
\end{aligned}$$

Schließlich bekommt man den neuen (berechneten) Lanczosvektor aus einer Division

$$\hat{q}_{j+1} = fl\left(\frac{\hat{w}_{j+1}}{\hat{s}_j}\right) = \frac{\hat{w}_{j+1}}{\hat{s}_j} + \delta q_{j+1}, \quad |\delta q_{j+1}| \leq \left|\frac{\hat{w}_{j+1}}{\hat{s}_j}\right|\epsilon,$$

und das bedeutet

$$\hat{s}_j\hat{q}_{j+1} = \hat{w}_{j+1} + \hat{s}_j\delta q_{j+1} = A^*(\hat{s}_{j-1}\hat{q}_{j-1} + \hat{\bar{c}}_{j-1}\hat{q}_j) + \hat{\bar{c}}_j\hat{q}_j - \tilde{f}_j$$

mit

$$\begin{aligned}
|\tilde{f}_j| &\leq |\hat{s}_j\delta q_{j+1}| + |\delta w_{j+1}| \\
&\leq |\hat{w}_{j+1}|\epsilon + \left((m+3)\sqrt{2}+2\right)|A^*|\left(\hat{s}_{j-1}|\hat{q}_{j-1}| + |\hat{\bar{c}}_{j-1}||\hat{q}_j|\right)\epsilon \\
&\quad +(1+2\sqrt{2})|\hat{\bar{c}}_j||\hat{q}_j|\epsilon + \mathcal{O}(\epsilon^2) \\
&\leq \left((m+3)\sqrt{2}+3\right)|A^*|\left(\hat{s}_{j-1}|\hat{q}_{j-1}| + |\hat{\bar{c}}_{j-1}||\hat{q}_j|\right)\epsilon \\
&\quad +(2+2\sqrt{2})|\hat{\bar{c}}_j||\hat{q}_j|\epsilon + \mathcal{O}(\epsilon^2).
\end{aligned}$$

Für den Fall, dass j ungerade ist, ändert sich an den Rechnungen nichts Wesentliches. □

Wie auch für den symmetrischen Lanczosalgorithmus treffen wir zwei zusätzlich Annahmen. Die erste ist die, dass die berechneten Lanczosvektoren lokal zueinander orthogonal sind, d.h.

$$\hat{q}_j^*\hat{q}_{j-1} = 0, \quad \text{für} \quad j = 2, \ldots k, \tag{4.23}$$

was in Matrixnotation

$$\hat{Q}_k^* \hat{Q}_k - I_k = (C_k^* + \Delta_k + C_k)\,, \tag{4.24}$$

bedeutet und hierbei ist C_k eine strikte obere Dreiecksmatrix, die zusätzlich auch auf der ersten Nebendiagonalen nur Nulleinträge besitzt. Die zweite Annahme betrifft das kleine Eigenwertproblem für $\hat{T}_k = \hat{G}_o^{(k)} \hat{G}_e^{(k)}$. Wir nehmen an, dass dieses exakt gelöst wird:

$$\hat{T}_k x_j^{(k)} = \theta_j^{(k)} x_j^{(k)}, \quad (\tilde{y}_j^{(k)})^* \hat{T}_k = \theta_j^{(k)} (\tilde{y}_j^{(k)})^*. \tag{4.25}$$

Ganz grob gesprochen haben wir in Satz 4.13 nachgerechnet, dass in endlicher Arithmetik (4.15) und (4.16) plus jeweils einem Fehler in der Größenordnung der Maschinengenauigkeit gilt, dass es also $F_k, G_k \in \mathbb{C}^{N\times k}$ gibt, mit

$$\begin{aligned} A\hat{Q}_k &= \hat{Q}_k \hat{T}_k + \hat{r}_k + F_k & (4.26)\\ A^*\hat{Q}_k &= \hat{Q}_k \hat{T}_k^* + \hat{\tilde{r}}_k + G_k, & (4.27) \end{aligned}$$

wo

$$\hat{r}_k = \begin{cases} \hat{s}_{k-1}\hat{s}_k\hat{q}_{k+1}e_{k-1}^* + \hat{\bar{c}}_{k-1}\hat{s}_k\hat{q}_{k+1}e_k^* & , k \text{ ungerade} \\ \left(-\hat{s}_k\hat{c}_{k+1}\hat{q}_{k+1} + \hat{s}_k\hat{s}_{k+1}\hat{q}_{k+2}\right)e_k^* & , k \text{ gerade} \end{cases}$$

und

$$\hat{\tilde{r}}_k = \begin{cases} (-\hat{s}_k\hat{\bar{c}}_{k+1}\hat{q}_{k+1} + \hat{s}_k\hat{s}_{k+1}\hat{q}_{k+2})e_k^*, & k \text{ ungerade} \\ \hat{s}_{k-1}\hat{s}_k\hat{q}_{k+1}e_{k-1}^* + \hat{c}_{k-1}\hat{s}_k\hat{q}_{k+1}e_k^*, & k \text{ gerade} \end{cases}.$$

Der nachfolgende Satz liefert das zu (2.13) analoge Ergebnis, es wird der Zusammenhang zwischen dem Orthogonalitätsverlust und der Konvergenz von Ritzpaaren aufgezeigt. In den Formeln (4.28) und (4.29) werden zudem die analogen Bestandteile wie bei (2.13) verwendet, nämlich die Residuen $\delta_{k,j}$, $\tilde{\delta}_{k,j}$ in exakter Arithmetik, Rundunsgfehlereinflüsse $\epsilon_{jj}^{(k)}$, $\tilde{\epsilon}_{jj}^{(k)}$ sowie der jeweils neu berechnete Lanczosvektor $\hat{q}_{k+1}$ und die Ritzvektoren $\hat{z}_j^{(k)}$ (und $\hat{\tilde{z}}_j^{(k)}$. Dabei ist zu beachten, dass - entsprechend der Herleitung des unitären Lanczos-ähnlichen Algorithmus - die Orthogonalität jeweils zwischen neu berechnetem Lanczosvektor und Rechts- bzw. Linksritzvektor gemessen wird, je nachdem ob der Index k gerade oder ungerade ist. Die Interpretation von (4.28) und (4.29) ist dann identisch zu der von Satz 2.4.

Satz 4.14 *Der unitäre Lanczos-ähnliche Algorithmus in endlicher Arithmetik genüge (4.24), (4.25), (4.26) und (4.27). Es sei*

$$\begin{aligned} \Delta_k\hat{T}_k - \hat{T}_k\Delta_k &= K_k - L_k \\ \hat{Q}_k^* F_k - G_k^*\hat{Q}_k &= (N_k + \Delta_{x,0}) - (M_k + \Delta_{0,x})\,, \end{aligned}$$

wobei K_k und N_k strikte untere Dreiecksmatrizen sind, L_k und M_k sind strikte obere Dreiecksmatrizen und $\Delta_{x,0}$ sowie $-\Delta_{0,x}$ sind Diagonalmatrizen,

auf deren Diagonale sich abwechselnd Null- und Nichtnulleinträge befinden[2]*. Für die berechneten Links- und Rechtsritzvektoren* $\hat{z}_j^{(k)} = \hat{Q}_k x_j^{(k)}$ *und* $\hat{\tilde{z}}_j^{(k)} = \hat{Q}_k \tilde{y}_j^{(k)}$ *gilt:*

$$\left|\hat{q}_{k+1}^* \hat{z}_j^{(k)}\right| = \frac{\left|\tilde{\epsilon}_{jj}^{(k)}\right|}{\tilde{\delta}_{k,j}}, \quad k \quad \textit{gerade} \tag{4.28}$$

$$\left|\hat{q}_{k+1}^* \hat{\tilde{z}}_j^{(k)}\right| = \frac{\left|\epsilon_{jj}^{(k)}\right|}{\delta_{k,j}}, \quad k \quad \textit{ungerade} \tag{4.29}$$

wobei

$$\tilde{\epsilon}_{jj}^{(k)} = \left(y_j^{(k)}\right)^* (N_k + \Delta_{x,0} + K_k)\, x_j^{(k)}, \qquad \epsilon_{jj}^{(k)} = \left(y_j^{(k)}\right)^* (M_k + \Delta_{0,x} + L_k)\, x_j^{(k)}$$
$$\tilde{\delta}_{k,j} = \left|\hat{s}_{k-1}\hat{s}_k y_{k-1,j}^{(k)} + \hat{c}_{k-1}\hat{s}_k y_{k,j}^{(k)}\right|, \qquad \delta_{k,j} = \left|\hat{s}_{k-1}\hat{s}_k x_{k-1,j}^{(k)} + \hat{\tilde{c}}_{k-1}\hat{s}_k x_{k,j}^{(k)}\right|.$$

Beweis: Wir führen den Beweis erst für den Fall, dass k eine gerade Zahl ist, durch. Zunächst multiplizieren wir (4.26) von links mit $\hat{Q}_k^*$ und erhalten

$$\hat{Q}_k^* A \hat{Q}_k = \hat{Q}_k^* \hat{Q}_k \hat{T}_k + \hat{Q}_k(-\hat{s}_k \hat{c}_{k+1} \hat{q}_{k+1} + \hat{s}_k \hat{s}_{k+1} \hat{q}_{k+2}) e_k^* + \hat{Q}_k^* F_k, \tag{4.30}$$

und transponieren von $\hat{Q}_k^* \times$ (4.27) liefert

$$\hat{Q}_k^* A \hat{Q}_k = \hat{T}_k \hat{Q}_k^* \hat{Q}_k + (\hat{s}_{k-1}\hat{s}_k e_{k-1} + \hat{c}_{k-1}\hat{s}_k e_k)\hat{q}_{k+1}^* \hat{Q}_k + G_k^* \hat{Q}_k \tag{4.31}$$

Man subtrahiert (4.31) von (4.30) und bekommt

$$\begin{aligned} 0 \;=\; & \hat{Q}_k^* \hat{Q}_k \hat{T}_k - T_k \hat{Q}_k^* \hat{Q}_k + \hat{Q}_k^*(-\hat{s}_k \hat{c}_{k+1} \hat{q}_{k+1} + \hat{s}_k \hat{s}_{k+1} \hat{q}_{k+2}) e_k^* \\ & -(\hat{s}_{k-1}\hat{s}_k e_{k-1} + \hat{c}_{k-1}\hat{s}_k e_k)\hat{q}_{k+1}^* \hat{Q}_k + \hat{Q}_k^* F_k - G_k^* \hat{Q}_k, \end{aligned}$$

was äquivalent zu

$$\begin{aligned} & (\hat{s}_{k-1}\hat{s}_k e_{k-1} + \hat{c}_{k-1}\hat{s}_k e_k)\hat{q}_{k+1}^* \hat{Q}_k - \hat{Q}_k^*(-\hat{s}_k \hat{c}_{k+1} \hat{q}_{k+1} + \hat{s}_k \hat{s}_{k+1} \hat{q}_{k+2}) e_k^* \\ & = C_k^* \hat{T}_k - \hat{T}_k C_k^* + \Delta_k \hat{T}_k - \hat{T}_k \Delta_k + C_k \hat{T}_k - \hat{T}_k C_k + \hat{Q}_k^* F_k - G_k^* \hat{Q}_k \end{aligned} \tag{4.32}$$

ist. Jetzt betrachtet man das Nullenmuster der einzelnen Summanden in (4.32). Es hat $\hat{Q}_k^*(-\hat{s}_k \hat{c}_{k+1} \hat{q}_{k+1} + \hat{s}_k \hat{s}_{k+1} \hat{q}_{k+2}) e_k^*$ nur Einträge ungleich Null in der letzten Spalte und ist folglich eine obere Dreiecksmatrix. Aufgrund der lokalen Orthogonalität (4.23) ist die letzte Spalte von $(\hat{s}_{k-1}\hat{s}_k e_{k-1} + \hat{c}_{k-1}\hat{s}_k e_k)\hat{q}_{k+1}^* \hat{Q}_k$ gleich Null und diese Matrix kann nur in den letzten beiden Zeilen Einträge ungleich Null besitzen, ist also eine untere Dreiecksmatrix.

[2] $(\Delta_{0,x})_{1,1} = 0$, $(\Delta_{x,0})_{1,1} \neq 0$

Die Diagonaleinträge von $\Delta_k\hat{T}_k - \hat{T}_k\Delta_k$ sind ebenfalls gleich Null. Es ist außerdem

$$C_k\hat{T}_k - \hat{T}_kC_k = \begin{bmatrix} 0 & x & \dots & \dots & x \\ & x & \ddots & & \vdots \\ & & 0 & \ddots & \vdots \\ & & & \ddots & x \\ & & & & x \end{bmatrix}$$

und

$$C_k^*\hat{T}_k - \hat{T}_kC_k^* = \begin{bmatrix} x & & & & \\ x & 0 & & & \\ \vdots & \ddots & x & & \\ \vdots & & \ddots & \ddots & \\ x & \dots & \dots & x & 0 \end{bmatrix}$$

Mit $\Delta_{x,0}$ und $\Delta_{0,x}$ wird die Diagonale von $\hat{Q}_k^*F_k - G_k^*\hat{Q}_k$ geteilt:

$$\begin{aligned} \Delta_{x,0} &= \operatorname{diag}\left(e_1^*(\hat{Q}_k^*F_k - G_k^*\hat{Q}_k)e_1, 0, e_3^*(\hat{Q}_k^*F_k - G_k^*\hat{Q}_k)e_3, 0, \dots, 0\right), \\ \Delta_{0,x} &= -\operatorname{diag}\left(0, e_2^*(\hat{Q}_k^*F_k - G_k^*\hat{Q}_k)e_2, 0, \dots, e_k^*(\hat{Q}_k^*F_k - G_k^*\hat{Q}_k)e_k\right). \end{aligned}$$

Man trennt nun in (4.32) die oberen und unteren Dreiecksmatrizen wie folgt:

$$\begin{aligned} \left(\hat{s}_{k-1}\hat{s}_ke_{k-1} + \hat{c}_{k-1}\hat{s}_ke_k\right)\hat{q}_{k+1}\hat{Q}_k &= C_k^*\hat{T}_k - \hat{T}_kC_k^* \\ &\quad + (N_k + \Delta_{x,0} + K_k) \qquad (4.33) \\ \hat{Q}_k^*\left(-\hat{s}_k\hat{c}_{k+1}\hat{q}_{k+1} + \hat{s}_k\hat{s}_{k+1}\hat{q}_{k+2}\right)e_k^* &= C_k\hat{T}_k - \hat{T}_kC_k \\ &\quad - (M_k + \Delta_{0,x} + L_k). \end{aligned}$$

Bildet man $\left(y_j^{(k)}\right)^* \times$ (4.33) $\times x_j^{(k)}$, so ergibt sich

$$\left(\hat{s}_{k-1}\hat{s}_ky_{k-1,j}^{(k)} + \hat{c}_{k-1}\hat{s}_ky_{k,j}^{(k)}\right)\hat{q}_{k+1}^*z_j^{(k)} = \left(y_j^{(k)}\right)^*(N_k + \Delta_{x,0} + K_k)\,x_j^{(k)},$$

wovon man den Betrag nimmt und (4.28) erhält. Für den Fall, dass k eine ungerade Zahl ist, zeigen analoge Rechnungen

$$\begin{aligned} e_k(-\hat{c}_{k+1}\hat{s}_k\hat{q}_{k+1}^* + \hat{s}_k\hat{s}_{k+1}\hat{q}_{k+2}^*)\hat{Q}_k &= C_k^*\hat{T}_k - \hat{T}_kC_k^* \\ &\quad + (N_k + \Delta_{x,0} + K_k) \\ -\hat{Q}_k^*\hat{q}_{k+1}\left(\hat{s}_{k-1}\hat{s}_ke_{k-1}^* + \hat{\bar{c}}_{k-1}\hat{s}_ke_k^*\right) &= C_k\hat{T}_k - \hat{T}_kC_k \\ &\quad - (M_k + \Delta_{0,x} + L_k) \qquad (4.34) \end{aligned}$$

und aus $\left(y_j^{(k)}\right)^* \times$ (4.34) $\times x_j^{(k)}$ folgt schließlich

$$-\left(\tilde{z}_j^{(k)}\right)^*\hat{q}_{k+1}\left(\hat{s}_{k-1}\hat{s}_kx_{k-1,j}^{(k)} + \hat{\bar{c}}_{k-1}\hat{s}_kx_{k,j}^{(k)}\right) = -y_j^{(k)}(M_k + \Delta_{0,x} + L_k)\,x_j^{(k)}.$$

□

Anhang A

Arithmetik

Wir geben kurz ein paar Bemerkungen zum Rechnen in Gleitkommaarithmetik entsprechend des Modells (1.4). Für natürliche Zahlen $n \in \mathbb{N}$ mit $n\,\epsilon < 1$ definieren wir die Zahlen

$$\gamma_n(\epsilon) = \frac{n\epsilon}{1 - n\epsilon}. \tag{A.1}$$

Um den Einfluß der Rundungsfehler bei mehreren hintereinander ausgeführten Rechnungen in Gleitkommaarithmetik abzuschätzen, sind die beiden folgenden Lemmata hilfreich. Diese finden sich beispielsweise in dem Buch [18] ab Seite 69.

Lemma A.1 *([18] S. 69) Es sei $|\delta_j| \leq \epsilon$ und $\rho_j \in \{-1, +1\}$ für $1 \leq j \leq n$, zudem sei $n\,\epsilon < 1$. Dann gilt*

$$\prod_{j=1}^{n}(1 + \delta_j)^{\rho_j} = (1 + \theta_n), \quad \textit{mit} \quad |\theta_n| \leq \gamma_n(\epsilon) \tag{A.2}$$

Für das nächste Lemma nehmen wir an, dass die Zahlen $\gamma_k(\epsilon)$ gemäß (A.1) immer definiert seien.

Lemma A.2 *([18] S. 74) Für eine natürliche Zahlen n sei θ_n wie in (A.2) definiert. Es gilt:*

$$\begin{aligned}
(1 + \theta_k)(1 + \theta_j) &= 1 + \theta_{k+j} \\
\frac{1 + \theta_k}{1 + \theta_j} &= \begin{cases} 1 + \theta_{k+j}, & j \leq k \\ 1 + \theta_{k+2j}, & j > k \end{cases} \\
\gamma_k(\epsilon)\gamma_j(\epsilon) &\leq \gamma_{\min\{j,k\}}(\epsilon) \\
j\gamma_k(\epsilon) &\leq \gamma_{jk}(\epsilon) \\
\gamma_k(\epsilon) + \epsilon &\leq \gamma_{k+1}(\epsilon) \\
\gamma_k(\epsilon) + \gamma_j(\epsilon) + \gamma_k(\epsilon)\gamma_j(\epsilon) &\leq \gamma_{k+j}(\epsilon)
\end{aligned}$$

A.1 Komplexe Zahlen

Sämtliche Berechnungen dieser Arbeit wurden mit **MATLAB** durchgeführt, und in MATLAB ist eine komplexe Arithmetik bereits implementiert. Anders als bei einer skalaren Programmiersprache wie **C** muß der Anwender in MATLAB nicht erst noch Routinen entwickeln, um auch komplexe Zahlen bearbeiten zu können. Das ist zumindest keine vollständig triviale Aufgabe. Deshalb werden kurz die Probleme erwähnt, die auftreten, wenn Rechnungen mit komplexen Zahlen in endlicher Arithmetik durchgeführt werden.

Selbstverständlich lassen sich die Grundrechenarten für komplexe Zahlen in der naheliegenden Art und Weise durchführen, d.h. zu $x = a + ib$ und $y = c + id$ rechnen wir

$$x + y = (a + c) + i(b + d) \tag{A.3}$$

$$xy = (ac - bd) + i(ad + bc) \tag{A.4}$$

$$\frac{x}{y} = \frac{ac + bd}{c^2 + d^2} + i\frac{bc - ad}{c^2 + d^2} \tag{A.5}$$

Um möglichen *Exponentenüber-* oder *Exponentenunterlauf* (d.h. dass das Ergebnis einer Rechnung die größte darstellbare Zahl übersteigt bzw. kleiner als die kleinste darstellbare Maschinenzahl ist) vorzubeugen, sollte man bei der Division zweier komplexer Zahlen den folgenden Standardtrick (vgl. [18] S. 503) verwenden:

$$\frac{x}{y} = \begin{cases} \frac{\left(a+b\left(\frac{d}{c}\right)\right)+i\left(b-a\left(\frac{d}{c}\right)\right)}{c+d\left(\frac{d}{c}\right)}, & |c| \geq |d| \\ \frac{\left(a\left(\frac{c}{d}\right)+b\right)+i\left(b\left(\frac{c}{d}\right)-a\right)}{c\left(\frac{c}{d}\right)+d}, & |c| < |d| \end{cases} \tag{A.6}$$

Es gilt:

Lemma A.3 ([18] S. 79) *Es seien $x, y \in \mathbb{C}$. Werden die Grundrechenarten in Gleitkommaarithmetik gemäß (A.3), (A.4) und (A.5) unter dem Standardmodell (1.4) durchgeführt, so gilt*

$$\begin{aligned} fl(x + y) &= (x + y)(1 + \delta), & |\delta| \leq \epsilon \\ fl(xy) &= xy(1 + \delta), & |\delta| \leq \sqrt{2}\gamma_2(\epsilon) \\ fl\left(\frac{x}{y}\right) &= \frac{x}{y}(1 + \delta), & |\delta| \leq \sqrt{2}\gamma_4(\epsilon) \end{aligned}$$

Wird die Division gemäß (A.6) durchgeführt, so gilt

$$fl\left(\frac{x}{y}\right) = \frac{x}{y}(1 + \delta), \qquad |\delta| \leq \sqrt{2}\gamma_7(\epsilon)$$

Grundlegend für jede Rundungsfehleranalyse in der Numerischen Linearen Algebra ist die Analyse des gewöhnlichen Skalarproduktes in $\mathbb{C}^N$ (bzw.

in $\mathbb{R}^N$). Seien $v, w \in \mathbb{C}^N$, mit den gleichen Rechnungen wie in [15] auf Seite 68 erhält man

$$fl(v^*w) = \bar{v}_1 w_1(1+\tilde{\theta}_{N+1}) + \bar{v}_2 w_2(1+\tilde{\theta}'_{N+1}) + \bar{v}_3 w_3(1+\tilde{\theta}_N) + \ldots + \bar{v}_N w_n(1+\tilde{\theta}_3),$$

wobei $|\tilde{\theta}_j| \leq \sqrt{2}\gamma_j(\epsilon)$ für $j = 3, \ldots, N+1$ und $|\tilde{\theta}'_{N+1}| \leq \sqrt{2}\gamma_{N+1}(\epsilon)$ gilt. Also folgt für das in Gleitkommaarithmetik berechnete Skalarprodukt zweier komplexer Vektoren

$$fl(v^*w) = v^*w + \delta vw, \quad |\delta vw| \leq \sqrt{2}\gamma_{N+1}(\epsilon)|v^*|\,|w|.$$

Mit diesem Ergebnis sieht man auch leicht die Resultate für die weiteren grundlegenden Rechenoperationen der linearen Algebra ein:

Ist $\alpha \in \mathbb{R}$, $x \in \mathbb{C}^N$, so gilt

$$fl(\alpha x) = \alpha x(1+\delta), \quad |\delta| \leq \epsilon,$$

wohingegen für $\alpha \in \mathbb{C}$

$$fl(\alpha x) = \alpha x(1+\delta), \quad |\delta| \leq \sqrt{2}\gamma_2(\epsilon)$$

richtig ist. Die komplexe Version einer Saxpy[1] Operation ist dann

$$fl(\alpha x + y) = \alpha x + y + \delta e$$

mit

$$|\delta e| \leq \Big((1+2\sqrt{2})|\alpha x| + |y|\Big)\,\epsilon + \mathcal{O}(\epsilon^2).$$

Schließlich gilt für die Multiplikation einer Matrix mit einem Vektor, wenn $m \leq N$ die maximale Anzahl an Einträgen ungleich Null in den Zeilen von $A \in \mathbb{C}^{N\times N}$ ist:

$$fl(Ax) = Ax + \delta Ax,$$

mit

$$|\delta Ax| \leq (m+1)\sqrt{2}|A||x|\epsilon + \mathcal{O}(\epsilon^2).$$

[1] Die Abkürzung *Saxpy* steht für **S**ingle precision **a**lpha x **p**lus y

Literaturverzeichnis

[1] G.S. Ammar, W.B. Gragg, L. Reichel, *On the Eigenproblem for Orthogonal Matrices*, Proc. 25th IEEE Conference on Decision and Control, Athens, 1986.

[2] L.V. Ahlfors, *Complex Analysis*, McGraw-Hill, 3. ed., New York, 1979.

[3] Zhajoun Bai, *Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem.*, Mathematics of Computation 62 (1994), no. 205, 209–226.

[4] B. Bohnhorst, *Beiträge zur numerischen Behandlung des unitären Eigenwertproblems*, Dissertation, Fakultät für Mathematik, Universität Bielefeld, Bielefeld, 1993.

[5] A. Bunse-Gerstner, L. Elsner, *Schur Parameter Pencils for the Solution of the Unitary Eigenproblem*, Linear Algebra and Its Applications, 154–156 (1991), 741–778.

[6] J.K. Cullum, R.A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. I Theory*, Birkhäuser Boston, 1985.

[7] P.J. Davis, P. Rabinowitz, *Methods of numerical integration*, Academic Press, 1975.

[8] V.L. Druskin, L.A. Knizhnerman, *Error bounds in the simple Lanczos Procedure for computing functions of symmetric matrices and eigenvalues*, Comput. Maths. Math. Phys. 31, No. 7, (1991), 20–30.

[9] S. Elhay, G.M.L. Gladwell, G.H. Golub, Y.M. Ram, *On some eigenvector-eigenvalue relations*, SIAM J. Matrix Anal. Appl. 20, No. 3 (1999), 563–574.

[10] L. Elsner, Kh. D. Ikramov, *On a condensed Form for Normal Matrices Under Finite Sequences of Elementary Unitary Similarities*, Linear Algebra and Its Applications 254 (1997), 79–98.

[11] T. Ericsson, *On the Eigenvalues and Eigenvectors of Hessenberg Matrices.*, Technical Report 10, Chalmers University of Technology and of University Göteburg, June 1990, Göteborg.

[12] H. Fassbender, *Error analysis of the symplectic Lanczos method for the symplectic eigenvalue problem.*, BIT 40 (2000), no.3, 471–496.

[13] G. Freud, *Orthogonal Polynomials*, Pergamon Press, 1971.

[14] G. Golub, H.A. van der Vorst, *Eigenvalue computation in the 20th century.*, Numerical Analysis, Vol III. Linear algebra. J. Comput. Appl. Math. 123 (2000), no. 1-2, 35–66.

[15] G. Golub, C. van Loan, *Matrix Computations*, The Johns Hopkins University Press, 2nd ed. 1989.

[16] A. Greenbaum, *Behavior of Slightly Perturbed Lanczos and Conjugate-Gradient Recurrences*, Linear Algebra and Its Applications, 113 (1989), 7–63.

[17] O.H. Hald, *Inverse Eigenvalue Problems for Jacobi Matrices*, Linear Algebra and Its Applications, 14 (1976), 63–85.

[18] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM Publications, Philadelphia, PA, 1996.

[19] L.A. Knizhnerman, *The quality of approximations to an isolated eigenvalue and the distribution of Ritz numbers in the simple Lanczos procedure*, Comput. Maths. Math. Phys. 35, No. 10, (1995), 1175–1187.

[20] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards 45 (1950), 255–282.

[21] J. Liesen, *Construction and Analysis of Polynomial Iterative Methods for Non-Hermitian Systems of Linear Equations*, Dissertation, Fakultät für Mathematik, Universität Bielefeld, Bielefeld, 1998.

[22] J. Liesen, *Computable Convergence Bounds for GMRES*, SIAM J. Matrix Anal. Appl. 21, No. 3 (2000), 882–903.

[23] C.C. Paige, *The computation of eigenvalues and eigenvectors of very large sparse matrices*, Ph.D. Thesis, London University, Institute of Computer Science, London, 1971.

[24] C.C. Paige, *Error Analysis of the Lanczos Algorithm for Tridiagonalizing a Symmetric Matrix*, J. Inst. Maths. Appl. 18 (1976), 341–349.

[25] C.C. Paige, *Accuracy and Effectiveness of the Lanczos Algorithm for the Symmetric Eigenproblem*, Linear Algebra and Its Applications 34 (1980), 235–258.

[26] B.N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, N.J. 1980.

[27] B.N. Parlett, *Misconvergence in the Lanczos Algorithm*, Res. Report PAM-404, University of California, Berkeley (1987).

[28] B.N. Parlett, *Do we fully understand the symmetric Lanczos algorithm yet?*, Proceedings of the Cornelius Lanczos International Centenary Conference (Raleigh, NC, 1993), 93–107, SIAM, Philadelphia, PA, 1994.

[29] B.N. Parlett, *Invariant subspaces for tightly clustered eigenvalues of tridiagonals.*, BIT 36, No. 3 (1996), 542–562.

[30] B.N. Parlett, D.R. Taylor, Z.A. Liu, *A look-ahead Lanczos Algorithm for unsymmetric matrices*, Mathematics of Computation 44 (1985), no. 169, 105–124.

[31] T.J. Rivlin, *The Chebyshev Polynomials*, Wiley, New York [u.a.], 1974.

[32] Y. Saad, *Variations on Arnoldi's Method for Computing Eigenelements of Large Unsymmetric Matrices*, Linear Algebra and Its Applications, 34 (1980), 269–295.

[33] Y. Saad, *Projection Methods for solving large sparse eigenvalue problems*, in Matrix Pencil Proceedings, B. Kågström und A. Ruhe, Herausgeber, Springer-Verlag, Berlin, 1982, 121–144.

[34] Y. Saad, *Numerical Methods for Large Eigenvalue Problems.*, Manchester University Press, Manchester; Halsted Press, New York, 1992.

[35] D.S. Scott, *How to Make the Lanczos Algorithm Converge Slowly*, Mathematics of Computation 33 (1979), no. 145, 239–247.

[36] D.C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method.*, SIAM J. Matrix Anal. Appl. 13 (1992), no. 1, 357–385.

[37] J. Stoer, *Numerische Mathematik 1*, Springer-Verlag, 7. Auflage 1994.

[38] Z. Strakoš, A. Greenbaum, *Open questions in the convergence analysis of the Lanczos process for the real symmetric eigenvalue problem*, IMA Research Report 934, March 1992.

[39] Z. Strakoš, A. Greenbaum, *Predicting behaviour of finite precision Lanczos and conjugate gradient computations*, SIAM J. Matrix Anal. Appl. 13 (1992), 121–137.

[40] Z. Strakoš, P. Tichý, *On error estimation in the conjugate gradient method and why it works in finite precision computations.*, Electronic Transactions on Numerical Analysis 13 (2002), 56–80.

[41] R.C. Thompson, P. McEnteggert, *Principal Submatrices II: The Upper and Lower Quadratic Inequalities.*, Linear Algebra and Its Applications 1 (1968), 211–243.

[42] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[43] J.H. Wilkinson, *Rundungsfehler*, Springer, Berlin [u.a.], 1969.

[44] K. Wu, H. Simon, *Thick-restart Lanczos method for large symmetric eigenvalue problems.* SIAM J. Matrix Anal. Appl. 22 (2000), no. 2, 602–616.

[45] Qiang Ye, *On close eigenvalues of tridiagonal matrices*, Numerische Mathematik 70 (1995), 507–514.

[46] J.P.M. Zemke, *Krylov subspace methods in finite precision: A unified approach.*, Dissertation, Technische Universität Hamburg-Harburg, Hamburg, 2003.